НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут прикладного системного аналізу

Кафедра штучного інтелекту

До захисту допущено:

В. о. завідувачки кафедри

_____ Ірина ДЖИГИРЕЙ

«___»_____20__ р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системи і методи штучного інтелекту»

спеціальності 122 «Комп'ютерні науки»

на тему: «Класифікація профілів користувачів на основі характеристик та поведінки»

Виконала:
студентка IV курсу, групи КІ-13
Бабич Маргарита Юріївна _____

Керівник:
доцент кафедри ШІ, д.ф.,
Гуськова Віра Геннадіївна _____

Консультант з економічного розділу:
доцент кафедри економічної кібернетики, к.е.н., доцент,
Рощина Надія Василівна _____

Консультант з нормоконтролю:
фахівець першої категорії кафедри ШІ, к.т.н., доцент,
Комариста Богдана Миколаївна _____

Рецензент:
старший викладач кафедри ММСА, д.ф.,
Левенчук Людмила Борисівна _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студентка _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра штучного інтелекту

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки»

Освітньо-професійна програма «Системи і методи штучного інтелекту»

ЗАТВЕРДЖУЮ
В. о. завідувачки кафедри
_____ Ірина ДЖИГИРЕЙ
«15» січня 2025 р.

ЗАВДАННЯ
на дипломну роботу студенту
Бабич Маргариті Юріївні

1. Тема роботи «Класифікація профілів користувачів на основі характеристик та поведінки», керівник роботи, керівник роботи Гуськова Віра Геннадіївна доцент кафедри ММСА, PhD., затверджені наказом по НН ІПСА від «26» травня 2025 р. № 1759-с.

2. Термін подання студентом роботи «09» червня 2025 року.

3. Вихідні дані до роботи – дані про транзакції клієнтів роздрібного онлайн-магазину, що містять інформацію про замовлення, товари, дати покупок, ціну, кількість, країну та ідентифікатор клієнта.

4. Зміст роботи – аналіз предметної області електронної комерції, огляд сучасних підходів до класифікації та кластеризації користувачів, підготовка та обробка даних, побудова гібридної моделі сегментації та прогнозування профілю клієнта, реалізація програмного рішення, а також оцінка якості моделі та інтерпретація результатів для бізнес-застосування.

5. Перелік ілюстративного матеріалу: електронна презентація.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Надія Василівна, доцент, к. е. н.		

7. Дата видачі завдання «03» лютого 2025 року.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за темою	21.04.2025	Виконано
2	Підготовка першого розділу	28.04.2025	Виконано
3	Підготовка другого розділу	05.05.2025	Виконано
4	Розробка програмного продукту	12.05.2025	Виконано
5	Підготовка третього розділу	19.05.2025	Виконано
6	Оформлення дипломної роботи	02.06.2025	Виконано
7	Підготовка презентації доповіді	09.06.2025	Виконано

Студент Маргарита БАБИЧ

Керівник Віра ГУСЬКОВА

РЕФЕРАТ

Дипломна робота: 87 с., 11 рис., 9 табл., 17 посилань, додаток.

КЛАСИФІКАЦІЯ, КЛАСТЕРИЗАЦІЯ, СЕГМЕНТАЦІЯ, ПОВЕДІНКОВІ ДАНІ, АНАЛІЗ ТРАНЗАКЦІЙНИХ ДАНИХ, МЕТОДИ МАШИННОГО НАВЧАННЯ, RFM АНАЛІЗ

Об’єкт дослідження – процеси взаємодії клієнтів з платформою електронної комерції, зокрема їх транзакційна активність.

Предметом дослідження – методи кластеризації та класифікації для поведінкових даних з метою автоматизованої сегментації та прогнозування профілю для нових користувачів в сфері електронної комерції.

Метою роботи – розробка моделі, що поєднує кластеризацію існуючих клієнтів та класифікацію нових користувачів, для побудови інструменту персоналізованого аналізу клієнтської бази.

Результати – проведено дослідження ефективності різних методів кластеризації та класифікації для сегментації клієнтів у сфері електронної комерції. Побудовано комбіновану модель, яка дозволяє автоматично визначати сегменти клієнтів на основі транзакційних даних, а також класифікувати нових користувачів до відповідних груп.

Шляхи подальшого розвитку предмету дослідження – інтеграція демографічних, поведінкових і інформаційних даних з інших каналів, використання більш складних алгоритмів глибинного навчання для покращення точності класифікації, а також розширення моделі для прогнозування життєвої цінності клієнта і ймовірності відтоку.

ABSTRACT

Bachelor's thesis: 87 p., 11 figures, 9 tables, 17 references, appendix.

CLASSIFICATION, CLUSTERING, SEGMENTATION, BEHAVIORAL DATA, TRANSACTION DATA ANALYSIS, MACHINE LEARNING METHODS, RFM ANALYSIS

Object of the study – the processes of customer interaction with the e-commerce platform, particularly their transactional activity.

Subject of the study – clustering and classification methods for behavioral data aimed at automated segmentation and profile prediction of new users in the e-commerce domain.

Purpose of the work – to develop a model that combines clustering of existing customers and classification of new users in order to build a tool for personalized customer base analysis.

Results – the study assessed the effectiveness of various clustering and classification methods for customer segmentation in e-commerce. A combined model was developed that allows automatic identification of customer segments based on transactional data, as well as classification of new users into the corresponding groups.

Directions for further research are integration of demographic, behavioral, and informational data from other channels; use of more advanced deep learning algorithms to improve classification accuracy; and expansion of the model to predict customer lifetime value and churn probability.

ЗМІСТ

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

CRM – Customer Relationship Management.

ETL – Extract, Transform, Load.

RFM – Recency, Frequency, Monetary.

ROC – Receiver Operating Characteristic.

AUC – Area Under the Curve.

DBSCAN – Density-Based Spatial Clustering of Applications with Noise.

CART – Classification and Regression Trees.

ID3 – Iterative Dichotomiser 3.

ВСТУП

Онлайн-магазини[1], банки, провайдери зв’язку, додатки на телефоні – сьогодні майже всі бізнеси збирають інформацію про своїх клієнтів. В сучасному світі підприємства схильні до прийняття рішень, керуючись даними.

Зібрана інформація про клієнтів допомагає створювати персоналізований досвід для кожного окремого користувача, пропонувати йому продукти подібні до того, що він купував раніше, запропонувати переглянути фільм чи прочитати книгу, яка пов’язана з контентом, що користувач вже вживав раніше. Аналітичні інструменти допомагають розділити користувачів на кластери, які характеризують поведінку кожної когорти людей.

Саме завдяки такій сегментації підприємство може сконцентруватись на кожній когорті користувачів окремо. Наприклад, заохочувати лояльних користувачів новими пропозиція чи персональними знижками, щоб змусити їх повертатись знову і знову. Таке розділення також допомагає активувати нелояльних користувачі за допомогою, наприклад, поштової розсилки з новими пропозиціями чи нагадуванням про продукти в користувацькому кошику.

Можна стверджувати, що такий персоналізований підхід – це потужний інструмент маркетингу, який, в певній мірі, дозволяє заощаджувати компанії на залученні нових клієнтів, за допомогою взаємодії зі старими клієнтами.

Ціль моєї роботи полягає в тому, щоб показати, як з невеликої кількості даних можна дізнатись надзвичайно багато нової інформації про користувачів, та за допомогою методів машинного навчання, виокреми кластер клієнтів і класифікувати кожного клієнта в кластер до якого він підходить.

---

[1]Тут і нижче використано такий інструмент штучного інтелекту як чат-бот з генеративним штучним інтелектом ChatGPT виключно для корегування та редагування тексту, створеного автором цієї дипломної роботи, на основі автоматизованої перевірки граматики, структури та стилю, що відповідає Політиці використання штучного інтелекту для академічної діяльності в КПІ ім. Ігоря Сікорського (протокол №11 Вченої ради КПІ ім. Ігоря Сікорського від 11 грудня 2023 р.).

Результат роботи можна використати для: прогнозу відтоку клієнтів, побудові маркетингових стратегій для кожного класу користувачів, оптимізації витрат на маркетингові кампанії.

РОЗДІЛ 1 ОГЛЯД ЗАСТОСУВАННЯ КЛАСТЕРИЗАЦІЙНИХ ТА КЛАСИФІКАЦІЙНИХ ПІДХОДІВ

1.1 Мета і актуальність задачі кластеризації та класифікації

Згідно з даними McKinsey & Company, компанії, які адаптують свої пропозиції під кожен сегмент клієнтів, мають дохід на 10-15% вище, ніж компанії, які не персоналізують свою комунікацію [1] . Близько 80% людей, спілкуються з тими брендами, які мають персоналізований підхід до кожного клієнта, а 70% маркетологів користуються сегментацією аудиторії [2].

Класифікація клієнтів, до певного сегменту дозволяє компаніям покращити свою комунікацію з покупцями і така співпраця є вигідною для обох сторін. Оскільки, для клієнтів теж важливо отримувати релевантні для себе пропозиції, і не витрачати час на перегляд нецікавого контенту.

Широке застосування класифікація знаходить у роздрібній торгівлі, як онлайн так і офлайн. Це проявляється в акціях, які різняться залежно від клієнта до клієнта, в пропозиція на карті лояльності та в рекламі, яку бачить кожен користувач онлайн. Завдяки такому розділу на сегменти, можна зрозуміти, які патерни поведінки об’єднують клієнтів, та до якої поведінки схильний кожен кластер.

Щоденно, користуючись своїми гаджетами, ми бачимо безліч реклами, але в більшості своїй, ця реклама пов’язана з нашими інтересами, тобто потенційно ми можемо скористатися з кожної пропозиції. Чи є це добре? Скоріше так, ніж ні. Завдяки такому підходу бізнес заощаджує на маркетингу, закуповуючи трафік тільки для кластерів, люди з який, потенційно стануть їх клієнтами.

Найбільш розповсюджене використання класифікація знаходить в маркетингу. Як вже було сказано, близько 70% маркетологів користуються розділом клієнтів на сегменти, завдяки чому росте ефективність їхньої роботи.

Розглянемо справу української мережі кінотеатрів Multiplex. Сегментація клієнтів дала позитивні результати і в продажах: листи з великою кількістю

пропозицій згенерували на 51,3% більше продажів при зовсім незначних відмінностях у трафіку [3]. Головним рушієм підвищення продаж стала персоналізація взаємодії зі своїми клієнтами.

Компанія збирала дані з декількох джерел, таких як: мобільний додаток, веб-сайт, підписка на старт продажів білетів. Після заповнення всіх цих даних, починається поділ клієнтів на сегменти, хоча може здатись що інформації відносно не багато, тобто це тільки дані про місце знаходження клієнта, кінотеатр, яким він цікавиться та фільм і час сеансу. Але цього вже достатньо, щоб почати персоналізовану комунікацію. Ця комунікація складалась з: мобільних пуш-повідомлень, повідомлень через месенджер Viber, SMS повідомлень та комунікацію через пошту. Відповідно люди дізнавались про новини, акції, тощо в кінотеатрах свого міста, та саме в тих кінотеатрах куди вони ходили. В цьому випадку, сегментація була заснована на геолокації. Також, варто додати, що в цьому випадку сегментація це попередній етап перед кластеризацією. Тобто, спочатку визначаються сегменти користувачів, а далі вже існуючі кластери присвоюють новим клієнтам.

Інший приклад використання профіля клієнта демонструє український бізнес – мережа супермаркетів Сільпо. За допомогою програми “Власний рахунок” мережа збирає інформацію, записуючі дані про покупки людей та дату транзакції [4]. За допомогою аналізу, компанія прогнозує майбутні покупки для кожного клієнтського сегменту і будує профіль клієнта. Найбільше такі алгоритми використовуються в маркетингу, де алгоритм будує пропозиції на основі вподобань клієнтів, наприклад пропонує знижки для людей які найчастіше ними користуються, чи екзотичні товари для людей з такими смаками. За допомогою сегментації, компанії також вдалось заощадити на випуску пластикових карт “Власний рахунок”. Для цього було проаналізовано який сегмент зацікавиться новою системою і виявлено, що ця пропозиція потенційно зацікавить людей з сім’ями, які користуються безготівковою оплатою. Завдяки цьому, було скорочено кількість виробництва карток.

Своє застосування кластеризація знаходить також в політиці на прикладі виборчої кампанії Володимира Зеленського. Під час підготовки команда провела анкетування та виділила 32 сегменти виборців. Завдяки цьому в подальшому були обрані цільові класи людей, та втілено таргетовану рекламу на платформі Facebook. Пізніше журналісти проаналізували кампанії двох головних кандидатів на посаду президента на цій платформі та виявили, що лозунги в підтримку Петра Порошенка були запущені на широкі сегменти людей, в той час як кампанія Володимира Зеленського була запущена точково, і отримала більший відклик.

Виявлення шахрайства в фінансових установах, ще один не менш популярний напрямок використання класифікації та кластеризації. На основі історичних даних, тобто транзакцій які вже пройшли, банки можуть прогнозувати з якою ймовірністю кожна транзакція є шахрайством. Для виявлення шахрайства використовуються класичні підходи як логістична регресія або ж набагато об’ємніші та складні системи на базі нейронних мереж, але принцип залишається незмінним. Виокремлення типової поведінки на основі щоденних транзакцій та пошук нетипових, які відрізняються від більшості транзакцій в установі. Надалі для нетипових транзакцій ймовірність виявитись шахрайством росте.

Підсумовуючи, бачимо, що класифікація та кластеризація це потужний інструмент в маркетингу і не тільки. Завдяки цьому, можна виділити конкретні сегменти людей, та використати персоналізований підхід для більшого відклику зі сторони вже існуючих або ж потенційних клієнтів чи користувачів, також можна фільтрувати шахрайські дії та блокувати їх до моменту реалізації транзакції.

1.2 Вплив профілів користувачі на бізнес-процеси

Типові підходи до сегментації виділяють декілька основних клієнтських груп, на основі яких прогнозуються класи клієнтів та будуються бізнес-процеси для взаємодії з кожним класом. Прогнозування безпосередньо впливає на те як бізнес будує стратегію продажів та обслуговує своїх користувачів. Розглянемо декілька бізнес-процесів та те як на них впливає класифікація та кластеризація.

Запуск маркетингової кампанії та її аналіз в сучасному бізнесі один з найважливіших етапів залучення клієнтів. В межах цього процесу застосовується безліч моделей машинного навчання, а саме моделі класифікації та кластеризації.

Після проведення рекламних кампаній, за допомогою аналізу, можна сегментувати більш і менш успішні кампанії. Таким чином можна виявити, до якої реклами аудиторія бізнесу прихильніша та в подальшому створенні реклами орієнтуватись на ці дані. Наприклад, виявити, які кампанії приваблюють користувачі здатних до відтоку, чи як реагують різні вікові категорії на різні рекламні банери. Класифікація допомагає прогнозувати як той чи інший клієнт буде взаємодіяти з маркетинговими методами. Це дозволяє оцінити який відсоток користувачів принесе прибуток, завдяки певному маркетинговому прийому. Ці знання дозволяють заощаджувати на рекламних кампаніях, за допомогою їх таргетування на активну аудиторію.

Відділ продажів розподіляє навантаження на клієнтів згідно з їх потенціалом, ймовірністю здійснити покупку чи реєстрацію. Перед цим необхідно визначити, який умовний потенціал має кожен клієнт. Існують різні техніки які дозволяють провести оцінку потенціалу завдяки знанням про клієнта. Однією з таких технік є потенційна оцінка клієнтів. Методологія, яку використовують відділи продажів і маркетингу для оцінки "якості" потенційних клієнтів шляхом присвоєння їм балів залежно від їхньої поведінки, що свідчить про зацікавленість у продуктах або послугах [6] . Такий підхід дозволяє сконцентрувати зусилля відділу на клієнтах, які мають більшу ймовірність

зробити покупку чи стати клієнтом компанії, що підвищує ефективність роботи та дозволяє ліпше розподіляти ресурси.

Відтік клієнтів – процес, коли клієнти схиляються до того, щоб перестати користуватись сервісом або здійснювати покупки, це типова проблема підприємств. Оскільки залучення нових клієнтів коштує дорожче ніж утримання вже наявних, постає проблема виявлення клієнтів які схильні до того щоб перестати робити покупки чи закінчити користування сервісом. Моделі прогнозування відтоку дозволяють, при наявності історичних даних, спрогнозувати, які клієнти більш схильні до відтоку. Завдяки передчасному виявленню таких клієнтів, компанія може завчасно звернути на них свою увагу, і за допомогою маркетингових прийомів та пропозицій утримати їх.

Об’єднання класифікаційних та кластеризаційних підходів, дозволяє краще зрозуміти цільову аудиторію, оптимізувати комунікацію та приймати ефективні рішення за допомогою даних. Такі підходи мають безпосередній вплив на якість маркетингових заходів та збільшення прибутковості.

1.3 Вплив набору ознак

Перед побудовою моделей класифікації та кластеризації необхідно зібрати дані про клієнтів. Розглянемо найпопулярніші набори ознак, та яку інформацію завдяки ним можна отримати.

Демографічні дані – це інформація яка описує вік, стать, дохід, місце проживання тощо. Така інформація зазвичай є сталою і не змінюється. Знаючи стать і вік можна провести первинну сегментацію людей. Використати це для побудови вподобань в розрізі місця проживання, або ж в розрізі величини від доходу. В подальшому, персональні дані дають змогу присвоювати нових клієнтів до раніше сформованих класів, і будувати взаємодію з ними на минулому досвіді.

Поведінкові метрики – дані, які описують поведінку користувача на сайті чи додатку. До уваги береться кількість сесій на сторінці, час проведений в сервісі, кількість переглянутих товарів. Найважливішими показниками є частота покупок та здійснення транзакцій. За допомогою цих даних, можна вирахувати цінність клієнта. Ця інформація є основним джерелом даних для такого класичного підходу до сегментації клієнтів, як RFM аналіз – розділення на сегменти в залежності від часу останньої покупки, частоти здійснення покупок, та виручки кожного клієнта.

Канали взаємодії – джерела, з яких клієнт потрапляє до нашого сервізу. Можна виокремити певні підкатегорії: тип девайсу, канал комунікації, джерело трафіка.

Знання про тип девайсу, допомагає визначити, які клієнти частіше здійснюють покупки чи переглядають веб-сторінку в розрізі пристроїв: персонального комп’ютера, планшету, смартфона. До цього можна додати тип операційної системи, яким користується клієнт, це може бути Windows, MacOS, Android, iOS. Знання про різницю цін на певні типи девайсі, дозволяють будувати припущення про дохід, якщо така інформація не була надана раніше.

Канали комунікації це поштові розсилки, SMS повідомлення, взаємодії з соцмережами компанії та пуш-повідомлення. Для формування характеристик важливо знати як клієнт реагує на кожен канал комунікації, тобто чи переходить по посиланням з розсилок, чи реагує на сповіщення в додатку.

Джерела трафіку допомагають зрозуміти звідки користувач прийшов. Джерелами трафіку можуть бути пошукові системи, соціальні мережі. Користувачі з різних класів можуть приходити з різних джерел. В подальшому, це дає зможу таргетувати рекламу для кожного класу клієнтів окремо, тобто розміщувати рекламу в соціальних мережах, для користувачів, які більше реагують на повідомлення там, і так само для реклами в пошукових системах.

Набір характеристик в електронній комерції вирішує багато проблем та допомагає персоналізувати підхід до клієнтів. Характеристики можуть бути сталими (стать і місце проживання), так і динамічними (кількість покупок, час

проведений на сторінці). Вдале поєднання цих атрибутів дає можливість отримати повний профіль клієнта, для його подальшого аналізу.

1.4 Вплив поведінкових шаблонів на диференційну здатність алгоритмів

Поведінкові шаблони – це типові патерни поведінки користувачів на сервісі, закономірності та послідовності, які дозволяють визначити наміри та інтереси користувачів. Розглянемо декілька технік, які дозволять зібрати інформацію для побудови поведінкових шаблонів.

Послідовність кліків (Clickstream) – характеристика того, як людина користується додатком чи веб-сайтом. До цього відносять послідовність кліків на сторінці, час проведений на сайті, глибину перегляду сторінки, заповнення реєстраційних форм, використання кнопки повернення на попередню сторінку, точку входу та виходу зі сторінки. Записуючи та аналізуючи кожен клік клієнта, компанії можуть виявляти популярний контент або місця, де користувачі залишають сайт, і використовувати цю інформацію для оптимізації користувацьких шляхів, маркетингових стратегій і воронок конверсії [8].

Шлях, який проходить клієнт дає інформація про те, чи це потенцій покупець чи просто оглядач. Кліки можуть вказувати на те, чи здійснить людина покупку. Якщо користувач цілеспрямовано шукає товар, ймовірність успіху зростає, і таку поведінку можна виокремити в патерн поведінки потенційного покупця. При хаотичному блуканні зі сторінки на сторінку не варто розраховувати на те, що покупка буде здійснена. Ще одним важливим патерном є швидкість переходу до форми покупки, відповідно, якщо користувач це робить швидко, його поведінку можна характеризувати як цілеспрямовану.

Сучасні моделі, які обробляють дані про послідовність кліків, дозволять класифікувати клієнта вже за перші декілька кліків, тобто присвоїти йому певний патерн поведінки [10] .

Підхід аналізу кліків доволі потужний інструмент, який дозволяє робити класифікації поведінки людини, а також для прогнозу конверсії.

Аналіз корзини – спосіб виявлення патернів поведінки згідно активності додавання товарів до кошику. Це ще один потужний інструмент диференціації клієнтів.

Для визначення патернів поведінки необхідно мати інформацію про те чи додає людина товари до свого кошику та які її наступні дії. Тобто, додавання до кошика товару та перехід до оформлення замовлення вже може охарактеризувати людину як цілеспрямованого покупці. В той час, якщо людина додає товари до кошику та повертається до покупок не оформлюючи замовлення, це може говорити про те, що такий клас покупців потребує додаткової стимуляції для здійснення замовлення.

Також кошик розглядається в розрізі типу продуктів які туди потрапляють. Наприклад, якщо в корзині знаходяться тільки товари зі знижкою, вже можна говорити про те, що такі клієнти будуть купувати тільки за знижками. Якщо ж товари в корзині знаходяться без подальшого оформлення замовлення, можна припускати, що клієнти очікують знижку або не впевнені в своєму замовленні.

Диференціація на різні профілі дозволяє застосовувати маркетингові прийоми щоб стимулювати покупців. Такими прийомами може бути нагадування про речі в кошику, пропозиція персональної знижки тощо.

Розрізнення профілів користувачі, завдяки послідовності кліків та аналізу клієнтського кошика показують високий рівень диференціації поведінки. Вони дозволяють не тільки визначати профіль клієнта, тобто сегментувати, а і в реальному часі прогнозувати до якого профілю буде належати користувач, використовуючи для цього короткі сценарії поведінки людини. Це дає змогу зробити платформу продажу динамічною, тобто додавати промо-акції чи прибирати їх в залежності від намірів користувача. Тобто, якщо модель

прогнозує сценарій поведінки людини як клієнта з сумнівами, йому можна пропонувати спеціальні пропозиції. Якщо ж клієнт є цілеспрямованим покупцем пропустити крок стимуляції до покупки, щоб не відволікати його. Такий підхід дозволяє динамічно будувати поведінку платформи для кожного класу клієнтів, що в свою чергу може підвищити охоплення платформи та конверсії.

1.5 Приклади використання моделей класифікації

Класифікація з попередньою сегментацію знаходить широке використання в електронній комерції. Моделі використовуються для: класифікації клієнтів по когортам в залежності від типу поведінки або часу, коли людина стає покупцем; виявленні шахрайства на платформі; прогнозування клієнтів схильних до відтоку; кластеризація на основі поведінкових векторів для виявлення груп клієнтів, яких не можна виявити вручну. Розглянемо декілька прикладних випадків застосування класифікації в торгівлі.

Американська корпорація Mastercard поділилась, як компанія вираховує шахрайські транзакції за допомогою штучного інтелекту. Компанія в рік обробляє близько 160 мільярдів транзакцій, які генерують дані для створення і навчання нової системи. Була створена система розпізнавання шахрайських транзакцій за допомогою присвоєння в реальному часі балу кожній транзакції, щоб в подальшому визначати рівень довіри до кожної транзакції. Для визначення того, чи транзакція є шахрайською, оцінюється попередній досвід користувача та його характеристики. За допомогою цього система здатна класифікувати транзакцію протягом 50 секунд. Це допомагає запобігти як шахрайським діям так і попередити транзакції коли покупці неправомірно їх оскаржують. Оскільки класичні моделі класифікації не є достатньо складними для даної задачі, система використовує композицію моделей, хоча в ній і присутня класифікація як така. Тобто на виході отримується бінарний результат чи була транзакція

шахрайською чи ні. За допомогою такого підходу компанія покращує рівень безпеки для користувачі, частково за допомогою алгоритму класифікації. [11]

Розглянемо також наукову публікацію “A Comparative Analysis of Churn Prediction Models: A Case Study in Bank Credit Card”, де автори порівнюють моделі машинного навчання для вирішення задачі прогнозування відтоку. У дослідженні було використано реальні дані одного з банків. Характеристики з якими працювала модель описувати їх вік, місце проживання, кредитний рейтинг, баланс рахунку, кількість продуктів в банку, наявність кредитної карти, активність. Оскільки, дані відтоку зазвичай не є збалансованими, тобто меншість клієнтів схильні до відтоку, такі дані є незбалансованими. Переде подачею даних в модель було використано техніки збалансування даних, щоб підвищити чутливість моделі до меншого класу. Також було відібрано найбільш релевантні атрибути для опису кожного клієнта. Автори розглядали два різних підходи: класичний підхід за допомогою логістичної регресії та моделі машинного навчання такі як випадковий ліс та XGBoost. Найкраще себе показав ансамблевий алгоритм XGBoost, який працює на основі дерев рішень та градієнтного бустингу. Модель показала точність 97%, що є гарним показником при розпізнаванні відтоку.

Підсумовуючи, бачимо, що моделі класифікації є популярним засобом щоб успішно вирішувати задачі різного спрямування. Такий підхід дозволяє компаніям оптимізувати свої ресурси, покращити клієнтський досвід, та вберегти себе від шахрайства. При об’єднанні класифікації з кластеризацію можна отримати потужний інструмент для вирішення прикладних задач у бізнесі.

1.6 Постановка проблеми

У сучасній електронній комерції, важливим є не тільки залучення нових користувачів, а й ефективна робота з тим, хто вже користується сервісом. Персоналізований підхід допомагає підвищувати якість маркетингових стратегій, збільшувати цінність клієнта протягом його життєвого циклу, зменшувати відтік та заощаджувати ресурси в середині різних бізнес процесів.

Оскільки дані, які збираються бізнесом, не містять готових категорій користувачів, виникає необхідність в побудові гібридної моделі, яка спочатку створює категорії користувачів, а потім дозволяє класифікувати нових клієнтів до відповідних груп.

Варто врахувати і те, що менші бізнеси не мають такого об’єму даних як технічні гіганти, наприклад Amazon чи Netflix. Проте більшість компаній не залежно від розміру володіють трансакційною інформацією, тобто дані про те коли були здійснені певні покупки, в якому обсязі та якого роду купівля.

Саме тому виникає потреба у побудові моделі, яка ґрунтуються на базових, але доступних усім бізнесам метриках. Така модель дозволить отримати цінну інформацію про поведінкові шаблони клієнтів, формувати персоналізовані пропозиції, ефективніше управляти маркетинговими кампаніями та підвищувати загальну ефективність взаємодії з аудиторією без необхідності у великомасштабному зборі даних.

Таким чином, необхідно розробити алгоритм, який виділити клієнтські сегменти на основі даних про транзакції здійснені користувачами, та побудувати класифікатор для прогнозу сегменту нових клієнтів .

Висновки до розділу 1

В цьому розділі було розглянуто актуальність класифікації в умовах сучасного бізнесу. Встановлено, що головною метою класифікації є персоналізованих підхід до кожного користувача і запобігання шахрайським діям. За допомогою алгоритмів бізнеси покращують користувацький досвід та заощаджують бюджет за допомогою розумного таргетування своєї реклами.

Досліджено профілі користувачі та який вплив вони мають на різні бізнес-задачі. Визначено, що за допомогою класифікації та кластеризації процеси в маркетингу, а саме таргет, будується на окремих профілях користувачі, що в свою чергу дозволяє проводити кампанії ефективніше та отримувати якісніший відгук від аудиторії. В продажах, за допомогою кластеризації можна визначити профілі клієнтів і акцентувати увагу на тих, людях, які скоріш за все будуть користуватись послугами, купувати товар. Не менш важливою виявилась також задача відтоку, яка дозволяє компаніям при вчасному визначенні таких людей, застосувати міри для їх утримання.

Описано різні набори даних про користувачів, такі як демографічні дані, поведінкові метрики та канали взаємодії. Після збору таких даних, компанія має можливість побудувати потужні моделі в різних розрізах, наприклад вік, стать, локація, чи соцмережа з якої прийшов клієнт.

Інший потужний інструмент визначення класу користувачів це поведінкові шаблони. Їх формування можливе на основі аналізу послідовності кліків на платформі, аналізі користувацького кошику. Це дає змогу визначити чи є клієнт рішучим чи потребує додаткової стимуляції щоб здійснити бажану дію.

В результаті попереднього аналізу, було розглянуто декілька прикладних застосувань кластеризації в потребах бізнесу, де моделі вирішують багато бізнес проблем.

РОЗДІЛ 2 МЕТОДИ ПОБУДОВИ ЗАДАЧІ КЛАСИФІКАЦІЇ ТА КЛАСТЕРИЗАЦІЇ

Класифікація це приклад задачі навчання з вчителем. Це означає, що для навчання таких моделей використовуються як вхідні ознаки так і правильні вихідні значення. Тобто, модель має визначити до якого класу належить об’єкт попередньо знаючи кількість класів та притаманні їм значення атрибутів. Наприклад, задача визначення шахрайства є задачею класифікації, оскільки модель має бінарних вихід, існує тільки два можливих варіанти розвитку подій, або об’єкт є шахрайською транзакцією або ні. Знаходить своє застосування класифікація і в електронній комерції. Завдяки таким алгоритмам, можна прогнозувати відтік клієнтів, визначити чи здійсніть клієнт покупку. Існує також багатокласова або мультикласова, якщо модель має вибір між трьома і більше класами на виході.

В свою чергу кластеризація є прикладом навчання без вчителя. Головна мета використання моделей такого типу – визначення кластерів або груп об’єктів, які подібні між собою. Зазвичай такі підходи використовуються в задачах, де треба виявити структуру в даних без міток. Розглядаючи приклад з шахрайськими транзакція, якщо ми не маємо інформацію про те, які дії можуть бути шахрайськими, доречно використати кластеризацію, щоб виявити які групи транзакцій відрізняються від інших. В електронній комерції задача кластеризації вирішує проблему сегментації клієнтів за поведінкою, інтересами тощо.

В цьому розділі буде розглянуто теоретичні підходи для вирішення задачі класифікації. Оскільки метою роботи є не тільки реалізація моделей класифікації і їх порівняння, а й реалізація моделей кластеризація, як попередньої обробки даних, розглянемо на яких засадах створені дані алгоритми. Розглянемо, за допомогою яких метрик можна виміряти якість роботи обох типів моделей.

Дослідимо існуючі методи очистки та фільтрації даних, роботу з атрибутами в умовах дисбалансу класів і пропущених значень.

2.1. Традиційні алгоритми класифікації

В цьому пункті буде розглянуто прості лінійні та деревоподібні алгоритми для вирішення проблеми класифікації. Їх перевага полягає в інтерпретованості моделі та відносно простій реалізації. Такі моделі не потребують великих обчислень і добре працюють на малих об’ємах даних та не потребують налаштування багатьох гіперпараметрів.

2.1.1 Логістична регресія

Логістична регресія – фундаментальний метод класифікації у статистиці та машинному навчанні. Логістична регресія належить до сімейства лінійних класифікаторів, використовує підхід навчання з вчителем. Регресія визначає ймовірність настання події, яка має дискретну природу. Вже згадувалось раніше, що класифікація може бути як бінарною так і багатокласовою. В класичному підході логістична регресія є методом бінарної класифікації, тобто прогнозує ймовірність до якого з двох класів належить об’єкт.

Модель логістичної регресії вираховує ймовірність приналежності об’єкту до певного класу, у випадку бінарної регресії моделюється ймовірність приналежності до 1 класу, на далі позитивний клас.

Оскільки, в лінійній регресії вихід функції не є обмеженим значеннями [0;1] в моделі логістичної регресії використовується логістична функція яка здійснює нелінійне перетворення результату лінійної моделі та має наступний вигляд:

$$P = (y = 1|\, x\,) = \frac{1}{1 + e^{-(w^T + b)}}, \tag{2.1}$$

де P – функція логістичної регресії;

w – вектор ваг;

x – вектор атрибутів;

b – зміщення.

На рисунку 2.1 зображено функцію логістичної регресії, яка показує ймовірність скласти екзамен в залежності від годин витрачених на навчання. Бачимо, що в проміжку від двох до чотирьох годин витрачених на навчання, крива різко зростає. Це називається зона переходу, тобто функція стає чутливою до незначних змін у вхідних ознаках і ймовірність віднести об'єкт до позитивного класу зростає. Це обумовлено експоненціальною природою функції.

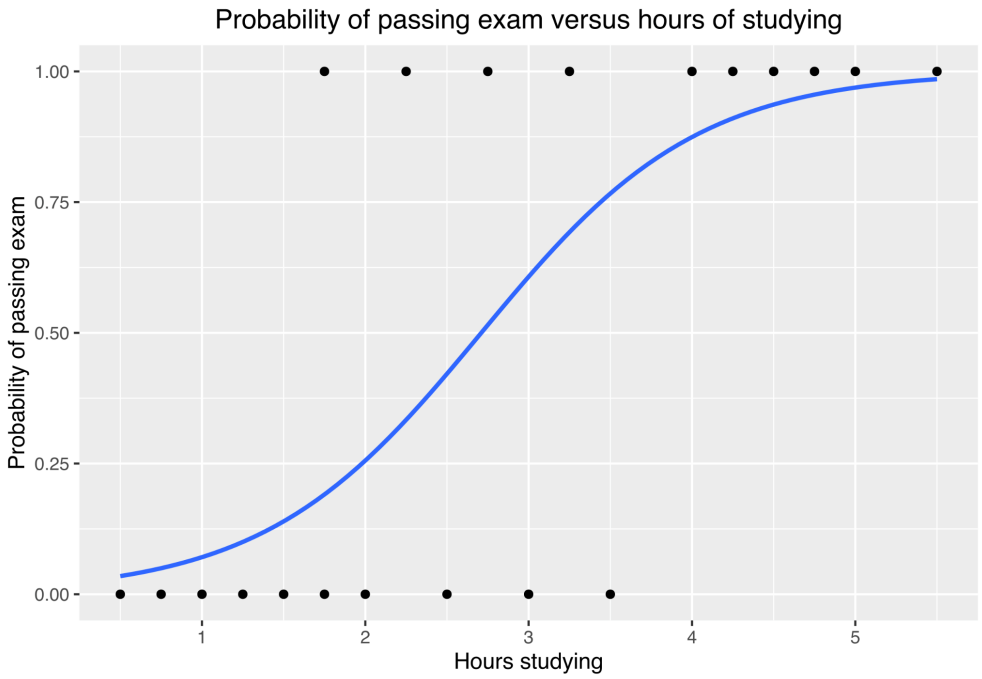

Рисунок 2.1 – Крива логістичної регресії [13]

Головними перевагами логістичної регресії є простота моделі та інтерпретованість результатів. За допомогою коефіцієнтів моделі можна визначити як кожна окрема ознака впливає на ймовірність приналежності до

позитивного класу. Логістична регресія добре працює в тому випадку, якщо класи розділені лінійно.

Оскільки логістична регресія оцінює вплив кожної ознаки, при наявності мультиколінеарності характеристик, модель буде працювати нестабільно, тому вимагається видалення ознак з високою кореляцією. Іншою проблемою є різний масштаб ознак. Це обумовлено методом оптимізації функції втрат, який відбувається завдяки градієнтному спуску. Для кращого перформансу рекомендовано попередньо стандартизувати всі вхідні ознаки. Оскільки модель заснована на побудові математичної функції, це унеможливлює роботу з категоріальними ознаками, без їх попереднього шифрування.

2.1.2 К-найближчих сусідів

К – найближчих сусідів ще один підхід до вирішення задач класифікації. Головна ідея алгоритму полягає в тому, щоб присвоїти клас об’єкту базуючись на його відстані до найближчих прикладів. Тобто алгоритм порівнює відстань між об’єктом для якого треба визначити клас і прикладами з навчальної вибірки. Найчастіше використовують евклідову відстань:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2}, \tag{2.2}$$

де d – евклідова відстань;

x – вектор ознак.

Визначається k об’єктів з найменшою відстанню і шляхом голосування шуканому об’єкту приписується той клас, в якого найбільше представників серед k об’єктів. На рисунку 2.2 зображено приклад присвоєння класу об’єкту коло.

Бачимо, що в залежно від того, як буде обране k, коло буде приписане до різного класу. Наприклад, якщо $k = 3$ колу буде присвоєний клас трикутника, у випадку $k = 5$ буде присвоєно клас прямокутника.

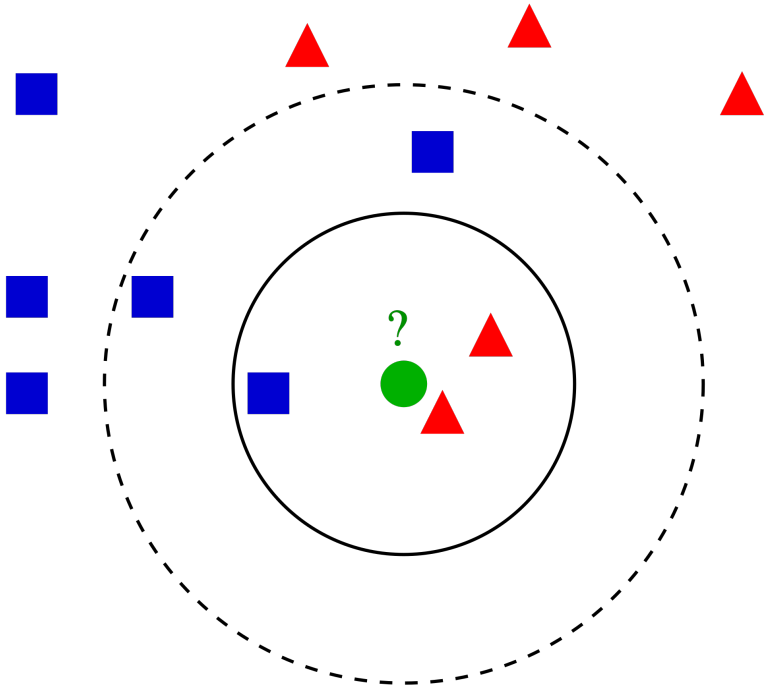

Рисунок 2.2 – Приклад роботи алгоритму к-найближчих сусідів. [14]

Такий підхід є непараметричним і алгоритм під час навчання не будує явну модель, а просто запам’ятовує дані. Тобто алгоритм не має функції втрат і не відбувається жодної оптимізації. Через це к-найближчих сусідів є представником “ледачого навчання” (lazy learning), в якому алгоритми під час навчання тільки завчають дані, а саме обрахування для кожного представника відбувається незалежно.

Це обумовлює повільність алгоритму, особливо на великих даних, оскільки для кожного об’єкту треба обраховувати відстані до всіх точок в навчальному наборі. Крім того, важливим параметром є метрика відстані, яка при неправильно виборі, може погіршити результати класифікації. Велика кількість ознак є ще однією перешкодою для використання цього алгоритму, оскільки саме вона сприяє виникненню прокляття розмірності, ситуації в якій

об’єкти розріджуються в простоті, об’єкти можуть стати майже однаково далекі, з’являється потреба в набагато більшій кількості даних. Як і логістична регресія, к-найближчих сусідів не може працювати з категоріальними ознаками, оскільки між ними неможливо виміряти відстань.

У випадку низької розмірності характеристик і правильному виборі метрики відстані, к-найближчих сусідів є ефективним алгоритмом класифікації, оскільки така модель є достатньо простою і здатна показувати високу точність. Також використання цього алгоритму підходить, коли є необхідність виявити аномалії, такі об’єкти, який знаходяться занадто дала він всіх інших.

2.1.3 Дерево прийняття рішень

Дерево рішень – типовий представник моделей машинного навчання. Такий алгоритм здатний працювати як з класифікацією так і з регресією, тобто передбачати неперервні значення. Розглянемо як працює дерево рішень в задачі класифікації.

Алгоритм базується на правилі “якщо-то”, тобто розщеплює простір ознак на підпростори. На виході ми маємо навчену модель, яка керуючись правилами визначає до якого класу належить об’єкт. На відміну від к-найближчих сусідів, дерево рішень довго навчається, але робить швидкий прогноз, оскільки вже має певний набір правил. На рисунку 2.3 зображено спрощений вигляд дерева рішень для прогнозування ризику серцевого нападу. Бачимо, що вузли дерева містять умови, а листки кінцеві передбачення. Кожне розгалуження дерева відбувається за певною ознакою, наприклад як вага, чи паління. На кожному кроці алгоритм обирає оптимальну характеристику та її поріг для розщеплення даних.

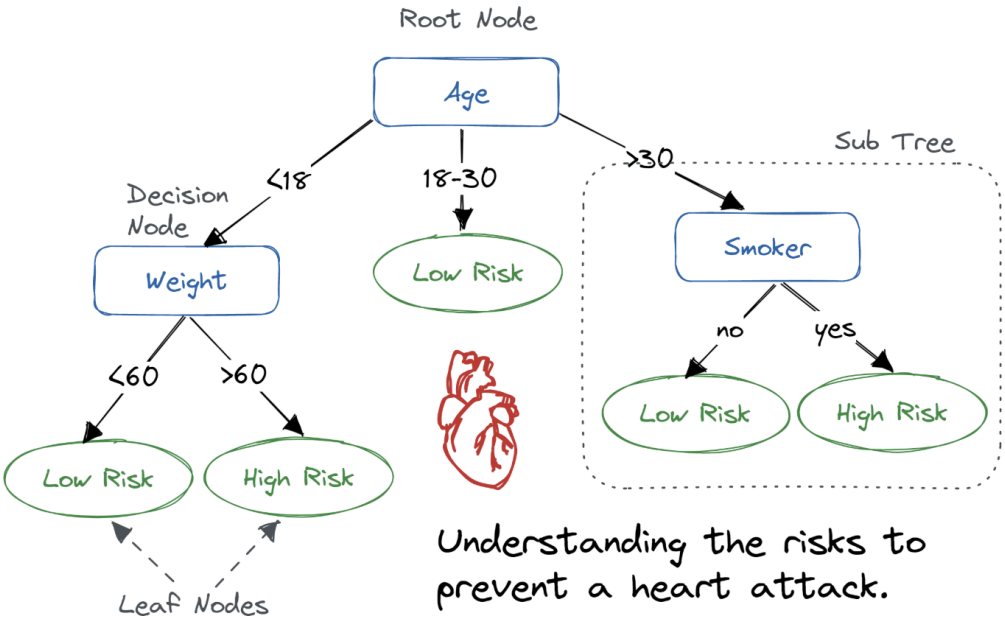

Рисунок 2.3 – Навчальний приклад дерева рішень для класифікації [15]

Для розділення ознак використовуються метрики невизначеності. Їх мета зменшити невизначеність в кожному вузлі, тобто зробити підмножини однорідними, де більшість об’єктів належить до одного класу. Для цього використовуються такі метрики як індекс Джині чи ентропія. Визначення індексу Джині:

$$G(t) = 1 - \sum_{k=1}^{K} p_k^2 , \tag{2.3}$$

де p_k частка об’єктів класу. Для розділення вузла обираються ті характеристики, які мінімізують міру невизначеності у підвузлах.

Кожна метрика визначає алгоритм побудови дерева, наприклад в алгоритмі CART основною метрикою розділення є індекс Джині, а для алгоритму ID3 використовується ентропія.

Завдяки такому підходу до розгалуження, дерева можуть ставати надто гнучкими і підлаштовуватись до навчальних даних. Це є головною причиною

перенавчання, тобто модель запам’ятовує навчальні дані та показує гарну точність, але точність на тестовому наборі даних значно зменшується. Щоб цього уникнути потрібно правильно налаштувати гіперпараметри моделі, тобто задати коректну глибину, мінімальну кількість прикладів в кожному вузлі, обрізати дерево після максимального поділу. Також проблему навчання можна вирішити завдяки ансамблевим моделям, наприклад випадковий ліс, який буде розглянуто далі.

Підсумовуючи, можна стверджувати, що дерево рішень ефективний та швидкий алгоритм, в якому дані не потребують масштабування, оскільки навчання не залежить від відстані між характеристиками. Крім того передбачення здійснюється швидко оскільки відбувається прохід по навченому раніше дереву.

2.2 Методи машинного навчання

Оскільки, вже було визначено, що класичні підходи потребують багато часу для роботи з великим об’ємом даних і вони не завжди можуть виявити складні патерни між класами, далі буде розглянуто сучасні підходи, такі як ансамблеві моделі та базову архітектуру нейронної мережі. Наступні моделі показують вищу точність та кращу узагальнюючу здатність у складних умовах.

2.2.1 Випадковий ліс

Випадковий ліс – ансамблевий метод машинного навчання. Такий підхід створює модель на основі “слабких учнів”, тобто базових і простих моделей. Ідея полягає в тому, що множина моделей буде працювати краще, ніж одна, з

врахуванням того, що кожна модель буде різна, узагальнююча здатність фінальної моделі теж буде кращою.

Для створення випадкового лісу використовують техніку бутстрапу (bootstrap), тобто кожне дерево навчається на випадковій підвибірці з початкового набору даних. Такий підхід додає випадковість і сприяє декореляції дерев, що покращує узагальнюючу здатність моделі. Навчанням можна керувати за допомогою таких гіперпараметрів як: глибина базової моделі, кількість базових моделей, мінімальна кількість прикладів для розщеплення тощо. Усереднення прогнозів для задачі класифікації відбувається за допомогою голосування, де перемагає мажоритарний клас.

Випадковий ліс добре працює з багатовимірними характеристика, як і дерево рішень, не потребує масштабування ознак та не є чутливим до викидів.

Попри вищевказані переваги моделі, її складніше інтерпретувати без додаткових бібліотек, на відміну від одного випадкового дерева. На великих обсягах даних навчання буде сповільнене.

2.2.2 XGBoost і LightGBM

XGBoost і LightGBM є прикладами ансамблевих моделей, навчання яких відбувається за допомогою градієнтного спуску. Обидва алгоритми використовують послідовність випадкових дерев, де кожне наступне дерево створюється з метою виправлення помилок попередніх представників. Алгоритми здатні швидко працювати на великих об’ємах даних з високою точністю результату. На сьогодні ці моделі є сучасними та стандартними при вирішенні задач класифікації та регресії.

XGBoost (Extreme Gradient Boosting) як було вище сказано, є ансамблевим алгоритм, який працює за принципом послідовної побудови моделей, де кожна наступна модель навчається на помилках попередньої. Ідея полягає в тому, щоб

за допомогою градієнтного спуску мінімізувати функцію витрат, удосконалюючи прогноз. Однією з особливостей алгоритму є вбудована регуляризація, яка дозволяє контролювати складність моделі і знижує ймовірність перенавчання. Для боротьби з перенавчання в моделі також вбудований параметр ранньої зупинки навчання, тобто тренування зупиняється якщо значення функції втрат перестає змінюватись протягом певної кількості інтеграцій. Алгоритм не вимагає масштабування ознак та працює з пропущеними значеннями. Швидке навчання моделі є наслідком паралельного виконання обчислень, завдяки чому алгоритм можна масштабувати на великі датасети та дані з високою розмірністю. XGBoost використовує техніку побудови дерев рівень за рівнем, тобто на кожному етапі дерево рівномірно розширюється на один рівень. Всі вузли поточного рівня дерева розбиваються рівномірно, незалежно від того як змінюється функція втрат. Завдяки такому підходу, дерево є симетричним, що в свою чергу зменшує перенавчання. Важливість ознак обраховується автоматично, що дає можливість зрозуміти які ознаки найбільше впливають на результат моделі.

LightGBM (Light Gradient-Boosting Machine) в порівнянні до XGBoost має інший принцип побудови дерев. Кожне наступне дерево будується в глибину, тобто розглядається той листок, для якого функція втрат є найбільшою. Це спричиняє побудову занадто глибоких дерев, що призводить до перенавчання, тому глибину дерева в цьому алгоритмі необхідно корегувати. LightGBM не оперує всіма ознаками на відміну від XGBoost, алгоритм використовує спеціальний механізм який вибирає ті атрибути, які є найбільш значущими, а атрибути без великої значущості об’єднує, зменшуючи розмірність даних. Це дозволяє зменшити кількість обчислень і робить алгоритм швидшим за XGBoost. З розглянути алгоритмів, LightGBM є першим, дозволяє працювати з категоріальними змінними без попереднього кодування.

2.2.3 Перцептрон

Перцептрон – одна із перших моделей штучних нейронних мереж. Вона стала основою для розвитку штучних мереж і машинного навчання загалом. Перцептрон є лінійним класифікатором, який ухвалює рішення на основі лінійної комбінації вхідних ознак. Модель складається з вхідного шару (вектор ознак), вагових коефіцієнтів та виходу. Класичний перцептор не містить прихованих шарів, його архітектура зображена на рисунку 2.4.

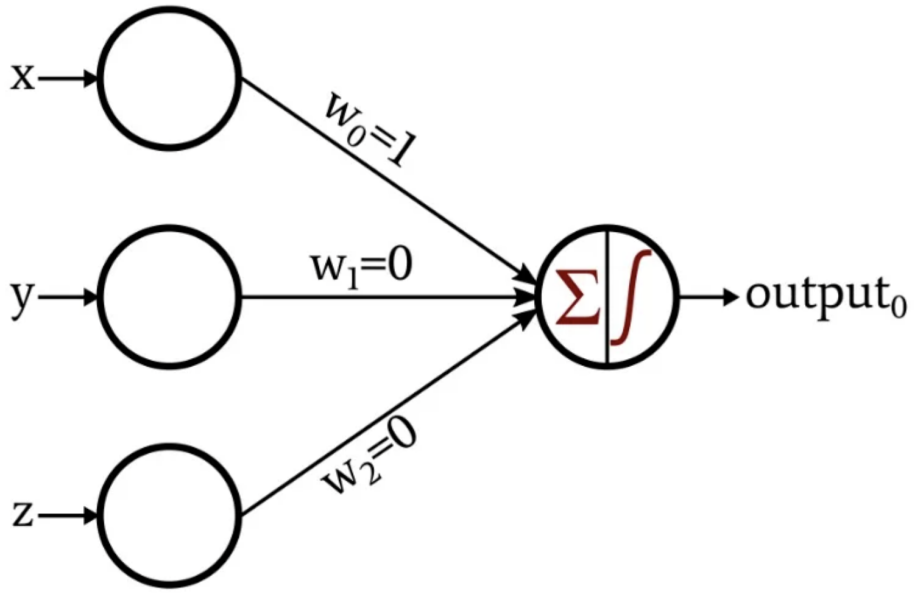

Рисунок 2.4 – Класичний перцептрон [16]

У класичному перцептроні кожен вхідний сигнал (x, y, x) множиться на відповідний ваговий коефіцієнт ($w0$, $w1$, $w2$) і обраховується зважена сума входів. Щоб отримати вихідний сингал, до зваженої суми входів застосовую функцію активації, наприклад ступеневу функцію, яка повертає значення 1 чи 0 залежно від того чи перевищує зважена сума входів певний поріг.

Навчання відбувається за допомогою коригування ваг, тобто якщо приклад проходить через перцептрон і клас було визначено неправильно відбувається коригування ваг згідно з величиною помилки.

Через свою простоту, перцептрон працює коректно тільки у випадку лінійно розділених даних, для яких існує гіперплощина, що розділяє дані на класи без помилок.

Щоб обійти це обмеження, пізніше було створено багатошаровий перцептрон, тобто модель, яка має хоча б один прихований шар нейронів. На кожному шарі застосовується функція активації і результат передається далі по мережі. В моделях цього класу кожен нейрон зв’язаний з усіма нейронами наступного шару, модель є повнозв’язною.

На відміну від класичного перцептрона, багатошаровий перцептрон використовує нелінійні функції активації (сигмоїдна, ReLU, гіперболічний тангенс), це дозволяє моделювати нелінійні залежності, і моделі цього роду є більш практичними.

Навчання відбувається завдяки алгоритму зворотного поширення. Після проходження прикладу по мережі, обраховується значення функції втрат на виході, тоді для кожного шару починаючи з вихідного обраховується градієнт функції втрат за вагами. Після цього змінюються ваги в напрямку протилежному до градієнта, щоб мінімізувати функцію втрат.

2.3 Кластеризаційні підходи як частина попередньої обробка даних

Раніше було визначено поняття та методи класифікації. Варто додати, що для навчання такої моделі необхідно мати класи. Оскільки при роботі з сирими даними класи попередньо не будуть визначені, їх необхідно створити самостійно. Це головна умова для класифікації, і також допоміжний інструмент для аналізу класі та факторів які їх утворюють в бізнесі.

Це можливо зробити за допомогою алгоритмів сегментації, результати роботи яких в подальшому стануть основою для класифікації. У цьому підрозділі буде розглянуто різні за природою алгоритми сегментації, а також типи задач, які вони дозволяють вирішувати в практичних бізнес-сценаріях.

2.3.1 K-середніх

Кластеризація методом К-середніх є класичним алгоритмом сегментації, популярний завдяки простій логіці та реалізації. К-середніх розділяє кластери за допомогою відстані між об’єктами, щоб мінімізувати дисперсію всередині кожного кластеру і максимізувати її між кластерами. Принцип роботи методу відбувається за схемою, представленою нижче.

1. Попередньо визначається кількість кластерів k.
2. Випадкового ініціалізується k центроїд – середні значення кожного майбутнього кластеру.
3. Кожен об’єкт призначається до кластеру, центроїда якого знаходиться на найменшій відстані до об’єкту.
4. Відбувається оновлення центроїд. Обчислюється нове середнє положення для всіх точок, щоб належить до відповідного кластеру.

Усі кроки повторюються ітераційно, до моменту стабілізації центроїд, тобто при наступних ітераціях, центроїд не змінюватимуть свого положення.

Такий підхід добре працю на випуклих і однорідних кластерах. Головними перевагами є простота та швидкість роботи.

Попри це треба завчано визначити кількість кластерів k.Також, результат роботи алгоритму залежить від початкової ініціалізації центроїд, і є чутливим до викидів, які можуть викривлювати середнє значення.

Для вирішення недоліків базового алгоритму були розроблені його модифікації, зокрема ті, що покращують процес початкової ініціалізації

центроїд. Деякі з них дозволяють враховувати ймовірність належності об'єкта до кількох кластерів, а інші – оновлюють центроїди не за всіма об'єктами, а на основі вибіркових підмножин даних, що суттєво пришвидшує роботу алгоритму на великих вибірках.

Також існують підходити, які дозволяють за допомогою евристики оцінити кількість класів k Один із найпоширеніших – метод ліктя, який заснований на визначені суми квадратів відстаней точок до центроїд для різних значень k. В такому підході оптимальною є кількістю кластерів k при якій сума квадратів відстаней точок до центроїд перестає різко спадати. Ця точка на графіку суми квадратів від кількості k є перегином кривої.

2.3.2 DBSCAN

DBSCAN (Density-based spatial clustering of applications with noise) належить до класу алгоритмів, які проводять сегментацію на основі щільності розподілу об’єктів. Головна ідея полягає в формуванні кластерів в областях високої щільності точок.

Формування кластерів відбувається за допомогою двох головних гіперпараметрів алгоритму: радіус пошуку точок (esp) та мінімальна кількість точок для формування щільної області (min_samples). Точка вважається ядровою, якщо в межах заданого радіуса esp вона має не менше, ніж min_samples сусідів. У цьому випадку вона виступає центром щільної області, яка потенційно формує кластер.

Точки які не мають необхідної кількості сусідів і не відносяться до жодного кластеру вважаються аномальними і відносяться до -1 кластеру. Таким чином в алгоритм вбудовано автоматичний пошук нетипових об’єктів.

Перевагою є відсутність необхідності задавати кількість кластерів на початку роботи алгоритму. Також завдяки роботі з щільністю, DBSCAN може формувати кластери довільної форми та розміру.

Недоліками алгоритму є чутливість до вибору параметрів, яку можна визначити експериментально чи за допомогою евристичних підходів і висока складність алгоритму, що сповільнює роботу з великими наборами даних.

2.3.3 Ієрархічна кластеризація

Сімейство методів ієрархічної кластеризації описує алгоритми, що формують вкладену структуру кластерів, яка зображаться у вигляді дендрограми – дерева на основі матриці подібності. За допомогою такого підходу можна утримати уявлення про ступінь подібності між кластерами на різних рівнях ієрархії.

Існує два підходи до створення кластерів.

1. Агломеративна кластеризація – кластери створюються знизу, тобто на початку роботи кожен об’єкт розглядається як кластер, після чого кластери зливаються за подібністю, доки не буде створено один загальний кластер.
2. Дивізивна кластеризація – підхід зверху вниз, на початку всі об’єкти належать до одного класу і потім рекурсивно розділяються на менші кластериде

Наступні кластери створюються за допомогою методів виміру відстані між кластерами. Найпопулярніші підходи до виміру відстаней є: мінімальна відстань між елементами класу, максимальна відстань між елементами кластеру, середня відстань, об’єднання кластерів при злитті яких найменше зростає внутрішньокластерна дисперсія.

Головна перевага – створення дендограми ієрархії класів, за допомогою якої можна самостійно визначити оптимальну кількість кластерів.

Недоліками таких алгоритмів є висока обчислювальна складність, неможливість роз’єднати раніше об’єднані класи, нестійкість до викидів, які можуть спотворити кластери.

Ієрархічна кластеризація підходить для задач де необхідно відслідкувати структур подібності або якщо є необхідність в багаторівневій сегментації.

2.4 Метрики якості моделей

Щоб обрати оптимальні алгоритми для вирішення проблеми, необхідно оцінити точність і здатність до узагальнення. Це можливо зробити за допомогою метрик оцінювання якості алгоритмів, які дозволяють порівнювати різні підходи. У цьому підрозділі будуть розглянути типові метрики для оцінки алгоритмів класифікації і кластеризації.

2.4.1 Метрики класифікації

Перед визначення метрик варто спочатку ввести такі поняття:

- TP (True Positive) – кількість випадків, коли модель визначила позитивний клас правильно;
- TN (True Negative) – кількість випадків, коли модель визначила негативний клас правильно;
- FP (False Positive) – кількість випадків, коли модель помилково визначила клас як позитивний, хоча об’єкт належав до негативного класу;
- FN (False Negative) – кількість випадків, коли модель помилково визначила клас як негативний, хоча об’єкт належав до позитивного класу.

У випадку багатокласової класифікації, метрики обраховуються для кожного класу окремо, тоді шуканий клас вважається позитивним, а всі інші класи – негативні.

У задачі класифікації існують два головних види помилок, помилка I роду і помилка II роду.

Помилка I роду визначається завдяки FP, коли шуканий клас був визначений неправильно, наприклад транзакція була шахрайською, а модель її такою не визначила.

Помилка II роду визначається завдяки FN, коли негативний клас був визначений неправильно, наприклад транзакція не була шахрайською, але модель її вважає такою.

Хоча на перший погляд і здається, що завжди варто мінімізувати помилку I роду це не так. В залежності від задачі, розробник моделі має визначити мінімізація якої помилки буде пріоритетна. Наприклад в задачах класифікації хвороб по зображенню гістограми пріоритетом є мінімізація помилки II роду, яка дозволяє не пропустити справжній випадок хвороби.

Головною метрикою оцінки класифікації є точність (accuracy), вона рахується як відношення кількості правильно визначених об’єктів до загальної кількості прикладів:

$$Accuracy = \frac{\sum_{i=1}^{K} TP_i}{N}, \tag{2.4}$$

де K – кількість класів;

N – кількість прикладів.

Не завжди точності достатньо для оцінки роботи моделі, особливо у випадку дисбалансу класів. Таким чином, якщо більшість прикладів належать до одного класу який добре класифікується, загальна точність буде високою попри низьку точність класифікації менших класі. Для визначення якості класифікації

в розрізі кожного класу використовують метрики точності (presicion) та повноти (recall).

Точність (presicion), не варто плутати з accuraccy, – показує, яка частина прикладів класифікованих до позитивного класу є насправді позитивними і обраховується за наступною формулою:

$$Precision_i = \frac{TP_i}{TP_i + FP_i}, \tag{2.5}$$

де *i*-номер класу. При обрахунку вважаємо *i*-й клас позитивним, а всі інші негативними.

Повнота (recall) – показує, яку частку позитивних класів модель виявила правильно:

$$Recall_i = \frac{TP_i}{TP_i + FN_i}. \tag{2.5}$$

Оптимальною буде модель з високою точністю і повнотою, проте часто збільшення однієї метрики призводить до зменшення іншої.

Для комплексної оцінки потрібно використовувати метрику, яка є зваженим повноти і точності – F1-метрика. Вона є особливо корисною у задачах з дисбалансом класів, оскільки дозволяє оцінити, наскільки добре модель одночасно виявляє позитивні приклади та уникає хибних спрацьовувань. F-метрику можна обрахувати за допомогою формули 2.6.

$$F_\beta = (1 + \beta^2)\frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}, \tag{2.6}$$

де β ваговий коефіцієнт. Якщо β дорівнює 1, формула набуває стандартного вигляду F1-метрика. Якщо ваговий коефіцієнт більше 1, то ми надаємо перевагу повноті, навпаки – точності.

Всі метрики в цьому підрозділі можуть мати значення в інтервалі [0,1], де 1 показує ідеальний результат. Такі метрики дозволяють оцінити якість роботи моделі класифікації і порівняти її з іншими.

2.4.2 Метрики кластеризації

Оскільки в задачі кластеризації ми не маємо правильних міток для навчання моделі, оцінка роботи алгоритму проводиться завдяки метрикам, які оцінюють кожен результуючий кластер, його щільність, віддаленість від інших кластерів та якість розділення.

Коефіцієнт силуету (Silhouette Coefficient) оцінює як добре об’єкти вписуються у всіх кластер, порівняно з найближчим сусіднім кластером:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(s)\}}, \qquad -1 \leq s(i) \leq 1, \tag{2.7}$$

де $a(i)$ – середня відстань між об’єктом і та іншими точками в його кластері;
$b(i)$ – найменша середня відстань між і та точками сусіднього кластера.

При значенні силуетного коефіцієнта, близькому до 1, можна зробити висновок, що об’єкт добре вписується у свій кластер і значно віддалений від сусідніх кластерів. Значення, наближене до 0, свідчить про те, що об’єкт розміщений поблизу межі між двома кластерами, а від’ємне значення вказує на ймовірну класифікацію об’єкта до неправильного кластеру, тобто ближчого до іншого центру.

Індекс Девіса–Болдіна (Davies–Bouldin Index) – метрика, яка оцінює ступінь подібності між кластерами, їх внутрішню щільність та віддаленість від інших кластерів. Тобто ідеальні кластери мають бути щільними та віддаленими один від одного.

$$DN = \frac{1}{K}\sum_{i=1}^{K} max_{j\neq i}\left(\frac{S_i + S_j}{M_{i,j}}\right), \quad (2.7)$$

де S_i – внутрішньокластерна дисперсія кластера i;

$M_{i,j}$ – відстань між центроїдами кластерів i та j.

Низьке значення індексу Девіса-Болдіна означає кращу класифікацію, метрика визначена в інтервалі від 0 до безкінечності.

Індекс Калінські-Харабаш (Calinski–Harabasz Index) – метрика, яка виміряє відношення міжкластерної дисперсії до внутрішньокластерної дисперсії. Таким чином, вище значення індексу означає, що кластери добре розділені і знаходяться на певній відстані один до одного, а їх внутрішня дисперсія мала, тобто кластери щільні.

За допомогою вищевказаних індексів можна оцінити якість кластеризації, а також провести кластеризацію для різної кількості кластерів k, та визначити найоптимальніше число k.

2.5 Джерела даних

Перед початком вирішення задачі класифікації та кластеризації є необхідність зібрати дані, від їхньої якості буде залежати якість роботи моделей. Для задачі побудови профіля клієнта, дані мають відображати поведінку людей та їх активність. Розглянемо далі декілька типових джерел даних для такої задачі.

Реєстраційні дані надають первинну інформацію про користувача, його стать, вік, регіон, тип девайсу з якого відбулась реєстрація. Ці дані можуть бути зібрані, як і під час першої реєстрації користувача так і під час реєстрації в певних

внутрішніх кампанія. Такі дані є основою для демографічної сегментації та дозволяють будувати первинну класифікацію клієнтів за профілями.

Лог-файли та журнали активності користувача містять інформацію про взаємодію користувача з сервісом. Як вже згадувалось раніше, ця інформація допомагає визначити сценарії поведінки клієнтів, що є важливим для прогнозу, наприклад, конверсії.

Записи транзакцій надають інформацію про активність клієнтів в сервісі, це можуть бути як підписки на сервісі, так і покупки. Знаючи про суми транзакцій, їх частоту, час замовлень, можна побудувати моделі цінності клієнта, наприклад RFM (Recency, Frequency, Monetary Value) модель, яка ділить користувачів на лояльних, активних та сплячих.

Маркетингові джерела такі як джерело трафіку, відклик на поштову розсилку, переходи на пуш-повідомленнями, дозволяють моделювати профілі клієнтів на основі їх реакцій на джерела комунікації.

Дані з зовнішніх систем (CRM системи, Google Analytics, Meta Pixel, Amplitude) дозволяють доповнити дані за допомогою зовнішніх джерел. Вони дають змогу простежити шлях клієнта до моменту входу на сервіс, зібрати інформацію про попередні контакти з сервісом, реакції на рекламні кампанії і відслідкувати зовнішні джерела трафіку. CRM системи зберігають в собі історію взаємодії з клієнтами, звернення в підтримку, покупки, дзвінки тощо. Google Analytics дозволяє оцінити середню тривалість сесії, географічну приналежність користувачів, що може бути корисним при подальшій сегментації. Meta Pixel записує реакції на рекламні кампанії в соціальних мережах, а Amplitude допомагає зрозуміти, як користувач взаємодіє з функціональністю сервісу. Інтеграція даних з цих джерел, допомагає побудувати більш точне представлення про профілі клієнтів характерні для конкретних бізнес сегментів.

Підсумовуючи, використання даних із різних джерел допомагає побудувати більш чітку характеристику клієнтів, оцінити їх цінність для бізнесу та виявити нетипові закономірності в поведінці користувачів.

2.5 Інтеграція даних з різних джерел

В попередньому підрозділі було розглянуто різні джерела даних, за допомогою яких можна побудувати потужніші профілі клієнтів. Проте велика кількість джерел створює проблему інтеграції даних між собою. Зазвичай для її вирішення використовують ETL-процеси.

На першому етапі дані з різних завантажуються в озеро даних, сховище, яке зберігає інформацію в різних форматах без необхідності перетворення. Далі дані трансформують до одного вигляду, створюють нові атрибути, очищують та нормалізують. На останньому етапі відбувається вивантаження даних до сховища, наприклад бази даних, що дозволяє в подальшому їх використання для аналітики та побудови моделей.

Також існує підхід ELT, обробка даних в іншому порядку, де трансформація даних відбувається вже після завантаження в цільове сховище.

Такий механізм дає змогу об’єднати дані з різних джерел, зробити їх узгодженими та придатними для подальшого аналізу. Ефективна реалізація цього процесу значно покращує вихідні дані, які є важливим фактором якості роботи будь-яких моделей побудованих на даних.

2.6 Попередня обробка даних

Дані отримані з бізнес-систем можуть містити шум, аномальні значення чи бути неструктурованими. Саме через це вимагається їх попередня обробка перед навчанням моделі.

В першу чергу аналізують кількість відсутніх даних в атрибутах. З відсутніми даними можна працювати по різному, в залежності від задачі.

Типовим підходом є видалення даних. Якщо ж дані суттєво впливають на залежну змінну, рекомендується заповнення таких пропусків, наприклад медіаною. Також на цьому етапі розглядають аномальні дані, викиди, які теж або видаляють або замінюються типовими значеннями, медіанами, для кожної характеристики.

На другому етапі змінні кодують. Раніше вже згадувалось, що багато моделей, які працюють з відстанями між об’єктами не можуть працювати з категоріальними даними. Для цього використовують бінарне кодування (one-hot encoding), тобто створюють характеристики відповідно до назв категорій. Для тих категорій що відповідають об’єкту проставляють значення 1, для всіх інших 0. Інший підхід до кодування категоріальних змінних кодування за мітками, тобто кожній категорії відповідає певне число від 0, і відповідно в стовбці категоріальні змінні замінюються неперервними. Така техніка є оптимальнішою у випадку, якщо бінарне кодування створює багато зайвих стовбців.

Після кодування дані також можуть масштабувати, якщо модель цього вимагає. Тобто значення неперервних змінних приводять до одного діапазону в межах атрибутів.

Після такого форматування, генерують нові атрибути, в залежності від даних та проблеми, яку модель має вирішити.

Для проблеми класифікації також актуальною є проблема дисбалансу класів. Для її вирішення можна застосувати техніки генерації об’єктів малих класів, або ж навпаки зменшувати розмір більших класів.

Підсумовуючи, попередня обробка даних є важливою умовою якісного аналізу. Цей етап забезпечує коректну роботу моделей, знижує похибку прогнозування, підвищує інтерпретованість результатів.

Висновки до розділу 2

Підходи класифікації та кластеризації в задачах класифікації профілів клієнтів демонструє значний потенціал для покращення та підвищення ефективності багатьох бізнес процесів. Завдяки здатності моделей машинного навчання знаходити зв’язки в даних, можна будувати моделі, що прогнозують схильність клієнтів до покупки, прогнозувати відток, аналізувати реакції на маркетингові кампанії.

Особливу увагу було приділеною комбінуванню моделей, використання кластеризації як попередньої обробки даних для подальшої кластеризації клієнтів. Такий підхід дозволяє спочатку поділити клієнтів та сегменти, а в подальшому призначати певний сегмент кожному новому клієнту.

Серед розглянутих підходів, високу ефективність для задачі класифікації показали ансамблеві моделі засновані на деревах рішень такі як XGBoost і LightGBM, а для задачі класифікації K-середніх та DBSCAN. Було проаналізовано метрики, які дозволяють оцінити роботи моделей та підібрати найоптимальніші рішення для різних типів даних.

Проведений аналіз формує теоретичну та прикладну основу для реалізації практичного етапу, який передбачає побудову моделей на реальних даних з урахуванням особливостей бізнес-середовища.

РОЗДІЛ 3 ПОБУДОВА МОДЕЛІ ПРОГНОЗУВАННЯ ПРОФІЛЯ КОРИСТУВАЧА НА ОСНОВІ ХАРАКТЕРИСТИК ТА ПОВЕДІНКИ

3.1 Опис набору даних та очистка

Для побудови гібридної моделі класифікації було обрано датасет “Online Retail II”, який складається з транзакцій покупок Британського онлайн рітейл магазину. Дані містять інформацію про покупки в період з 1 грудня 2009 до 9 грудня 2010 року, тобто один за один календарний рік.

Атрибути початкового датасету:

- InvoiceNo – номер транзакції, C на початку є поверненням;
- StockCode – код товару;
- Description – опис товару;
- Quantity – кількість одиниць;
- InvoiceDate – дата замовлення;
- UnitPrice – ціна одиниці;
- CustomerID – код клієнта;
- Country – країна, резидентом якої є клієнт.

Було досліджено, що дані містять пропуски та аномальні значення. З метою очищення, було видалено всіх клієнтів без ідентифікаційного коду, видалено транзакції, які описують доставку, платну інструкцію, знижку та транзакції на благодійність.

Дані є репрезентативний прикладом транзакцій активності користувачів в електронній комерції, що дозволяє виконати кластеризацію поведінкових шаблонів та побудову моделей прогнозування для нових клієнтів, що є метою даної роботи.

3.2 Розвідувальний аналіз

Розвідувальний аналіз є ключовим етапом перед побудовою моделей машинного навчання. Розглянемо структуру і особливості даних.

Загальна кількість транзакцій становить 404921, покупки здійснило 4364 користувачі. В ході аналізу, також було виявлено, що частка транзакцій є транзакціями повернення. На рисунку 3.1 зображено відсоткове відношення транзакцій. Бачимо, що близько 4% від всіх здійснених покупок, було повернуто.

Рисунок 3.1 – Розподіл транзакцій за типом

На рисунку 3.2 зображено розподіл покупок за днями тижня, можна зробити висновок, що в суботу платформа не приймала жодних замовлень.

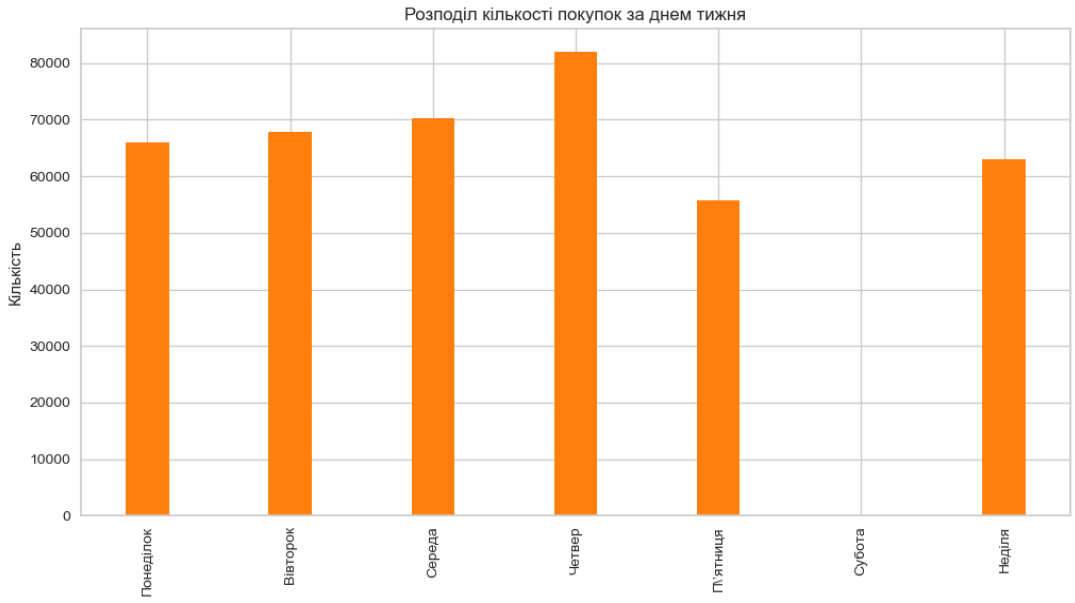

Рисунок 3.2 – Розподіл транзакцій за днем тижня

У ході розвідувального аналізу було встановлено, що замовлення надходили не лише від резидентів Великобританії, але й від клієнтів з понад 30 інших країн. Однак понад 90% усіх транзакцій були здійснені саме британськими клієнтами, що свідчить про переважно внутрішній характер ринку для даного інтернет-магазину.

3.3 Генерування допоміжних атрибутів

Оскільки дані містять тільки опис транзакцій, для побудови клієнтського профілю необхідно було створити додаткові атрибути.

Для базової сегментації було обрано стратегію RFM аналізу, тобто розділення клієнтів згідно частоти покупок, давності покупок, кількості витрачених грошей.

Створено наступні атрибути:

- Recency – час від останньої покупки в днях;

- Frequency – кількість транзакцій;
- Monetary – сума витрат споживача, не враховуючи повернення;
- NetMonetary – сумарна вартість всіх покупок, враховуючи повернення;
- TotalPrice – загальна ціна покупки, кількість товару помножена на ціну одиниці товару;
- FirstPurchase – кількість днів з моменту першої покупки;
- PurchaseInterval – середня кількість днів між покупками, 0 в разі однієї покупки;
- PurchaseDayOfWeek – день тижня в який було здійснено транзакцію, від 1 до 7 відповідно;
- ReturnRate – коефіцієнт повернень, кількість повернутих товарів відносно всіх транзакцій.

Оскільки подальша сегментація виконується на рівні індивідуального клієнта, всі поведінкові та транзакційні атрибути було зведено до агрегованого представлення для кожного клієнта згідно його ідентифікаційного коду.

Для зменшення впливу аномальних значень було прологарифмовано наступні атрибути: Frequency, Monetary, NetMonetary, AvgOrderPrice, PurchaseInterval.

Також було змінено атрибут PurchaseDayOfWeek, який відображає день тижня покупки у категоріальному форматі від 1 до 7. У такому представленні неділя та понеділок сприймаються як максимально віддалені значення, хоча насправді між ними лише один день. Щоб уникнути такого спотворення було створено дві додаткові змінні Day_sin та Day_cos, які перетворюють значення ознаки на двовимірне представлення на одиничному колі. Це дозволяє зберегти циклічну природу дня тижня, де близькі за календарем дні залишаються близькими і в числовому представленні.

Саме ці атрибути будуть в подальшому використовуватись для створення моделей.

3.4 Побудова моделей кластеризації

Оскільки для побудови класифікації необхідно мати мітки класів, було побудовано модель кластеризації та визначено і описано кластери клієнтів.

На початковому етапі, для побудови кластерів було обрано розширений вектор ознак, який включав в себе не тільки Recency, Frequency, Monetary а і декілька додаткових атрибутів, таких як FirstPurchase, PurchaseInterval, ReturnRate. Проте за результатами роботи моделей не вдалось чітко відокремити кластери, через варіативність ознак.

З огляду на це, для побудови моделі кластеризації було обрано лише три ознаки: Recency, Frequency, Monetary. Для визначення кластерів обрано три алгоритми, які відрізняються за своєю природою: K-середніх, DBSCAN, та агломеративна кластеризації.

Найкращий результат показав алгоритм K-середніх, результати метрик для порівняння алгоритмів зображено в таблиці 3.1.

Таблиця 3.1 – Результати кластеризації

Модель	*Коефіцієнт силуету*	*Індекс Девіса–Болдіна*	*Індекс Калінські-Харабаш*	*Кількість кластерів*
K-середніх	0.4290	0.7853	4662.48	4
DBSCAN	0.2405	1.4963	779.80	3
Агломеративна кластеризація	0.3818	0.7960	4071.66	4

Бачимо, що відносно інших алгоритмів, K-середніх продемонстрував: найвище значення коефіцієнта силуету, що вказує на чітке розмежування між кластерами та високу внутрішню однорідність; найменше значення індексу

Девіса–Болдіна, що свідчить про якісне формування кластерів з мінімальним перекриттям; найбільше значення індексу Калінські–Харабаша підтверджує, що кластери суттєво відрізняються один від одного.

Кількість кластерів було обрано за допомогою метода ліктя та силуетного аналізу, для різної кількості к.

На рисунку 3.3 зображено результати кластеризації клієнтів у тривимірному просторі за трьома ключовими ознаками: давність, частота покупок та сума витрат. Кожна точка відповідає окремому клієнту, а колір – належності до певного кластера, визначеного алгоритмом K-середніх.

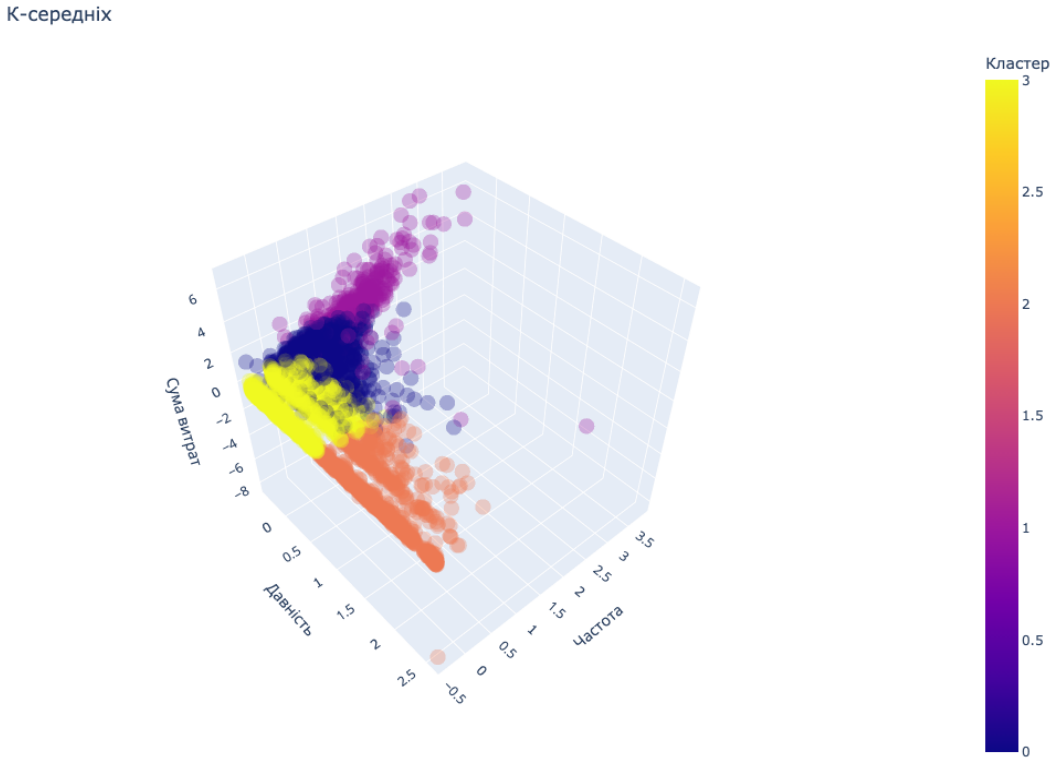

Рисунок 3.3 – Візуалізація утворених кластерів

Як видно з візуалізації, кластери утворюють відносно компактні та розділені області, що свідчить про наявність структури в даних. Для опису профіля кожного кластеру було взяло медіанне значення по кожному атрибуту та отримано наступне:

- кластер 0: Лояльні, середньоцінні клієнти, які не витрачають забагато, але купують регулярно й переважно не повертають товари;
- кластер 1: Найбільш активні та вигідні клієнти, з великими витратами;
- кластер 2: Пасивні клієнти, остання покупка була зроблена більше 6 місяців тому, робили покупки 1-2 рази;
- кластер 3: Нові клієнти, останні покупки були зроблені приблизно 2 місяці тому, здійснювали декілька покупок та робили мало повернень.

За результатами кластеризації було визначено відсоткове співвідношення кількості клієнтів у кожному кластері, що зображено на рисунку 3.2. Найбільшу частку склали нові клієнти, а найменшу частку складає кластер, який описує найвигідніших і найактивніших клієнтів.

Рисунок 3.4 – Співвідношення кластерів

В результаті кластеризації за допомогою методу К-середніх, було отримано добре розмежовані кластери, які мають чітку бізнес інтерпретацію. В наступному розділі розглянемо побудову моделей класифікації для прогнозу профіля клієнта з використанням отриманих міток.

3.5 Побудова моделей класифікації

Оскільки класи виявились таким, що добре розмежовують для розробки моделі прогнозу було обрано три прості моделі з різними алгоритмами обробки даних: логістична регресія, дерево рішень, перцептрон з одним прихованим шаром.

Для навчання моделей були використані всі доступні атрибути: Recency, FirstPurchase, Frequency, Monetary, NetMonetary, AvgOrderPrice, PurchaseInterval, ReturnRate, ReturnFlag, Day_sin, Day_cos. В якості цільової ознаки використовувались мітки, знайдені за допомогою кластеризації.

Для підбору гіперпараметрів кожної моделі, окрім перцептрону, було використано пошук по сітці або GridSearchCV, в яку вже інтегрована перехресна перевірка.

У таблиці 3.2 представлені результати роботи моделі логістичної регресії (LogReg), дерева рішень (Tree), та багатошарового перцептрону з одним прихованим шаром (MLP).

Таблиця 3.1 – Метрики моделей

Модель	*Accuracy*	*F1*	*Precision*	*Recall*	*AUC*
LogReg	0.98	0.98	0.98	0.98	0.99
Tree	0.96	0.96	0.96	0.96	0.98
MLP	0.94	0.94	0.94	0.94	0.99

Логістична модель показала найкращі результати класифікації, оскільки дані були добре лінійно роздільні.

Дерево рішень показало гірші результати, оскільки метою було побудувати просту модель, для дерева рішень була обрана глибина 5, щоб уникнути перенавчання і спростити її.

Найгіршою моделлю виявився багатошаровий перцептрон з одним прихованим шаром з 10 нейронами в ньому. Така модель була обрана для порівняння з іншими, оскільки вона здатна знаходити нелінійні зв’язки, проте в даному випадку такої необхідності не було.

Отже, логістична регресія продемонструвала найкраще співвідношення точності, збалансованості метрик, інтерпретованості та стабільності. Саме тому вона була обрана як основна модель для прогнозування належності нових клієнтів до кластерів.

У результаті перехресного підбору параметрів було обрано конфігурацію логістичної регресії з середньою силою регуляризації C=1 та L2-нормою, що забезпечує стабільне згладжування коефіцієнтів без їх занулення. Як алгоритм оптимізації використано lbfgs – ефективний та рекомендований для багатокласових задач. Обмеження max_iter = 100 забезпечило швидку та надійну збіжність при попередньо масштабованих ознаках.

На рисунку 3.3 зображено матрицю невідповідностей, з якої можна зробити висновок, що модель справилась з задачею прогнозу.

Також було побудовано ROC-криву, зображену на рисунку 3.4, яка показує як змінюється чутливість (TPR) відносно хибно позитивної кількості результатів (FPR) для кожного класу.

Рисунок 3.5 – Матриця невідповідностей

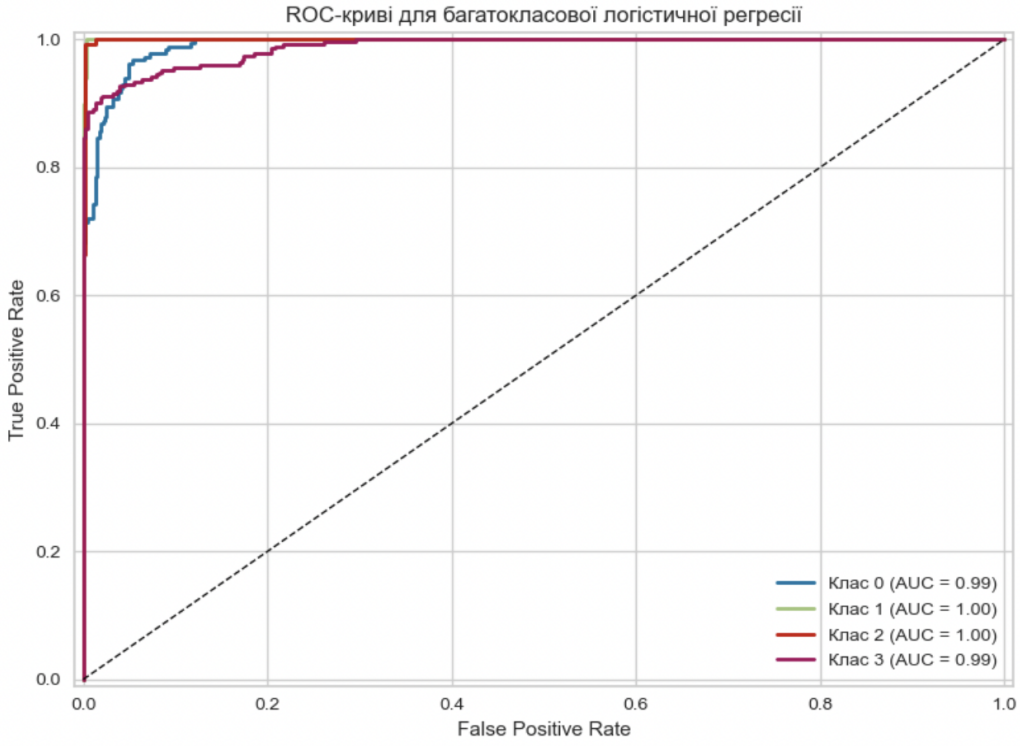

Рисунок 3.6 – ROC-крива

Класи 0 і 3 мають значення площі під кривою AUC = 0.99, що свідчить про дуже високу якість класифікації – модель майже завжди правильно розпізнає ці класи. Для класів 1 і 2 показник AUC досяг 1.00, тобто модель ідеально відділяє ці класи від інших, без жодної плутанини. Усі ROC-криві лежать значно вище діагоналі випадкових рішень, що вказує на високу дискримінативну здатність логістичної регресії в умовах багатокласового класифікаційного завдання.

Оскільки логістична регресіє є добре інтерпретованою моделлю, розглянемо коефіцієнти моделі, наведені в таблиці 3.3, які дозволяють оцінити вплив кожної ознаки на ймовірність належності клієнта до певного кластеру.

Таблиця 3.3 – Коефіцієнти логістичної регресії

Class	*Class 0*	*Class 1*	*Class 2*	*Class 3*
Recency	-1.793454	-1.121621	4.443332	-1.528257
NetMonetary	0.469657	3.928092	-1.443435	-2.954314
Frequency	1.156771	2.810703	-1.731709	-2.235765
Monetary	1.383799	2.128432	-1.918352	-1.593880
FirstPurchase	-0.291267	0.819682	0.588440	-1.116855
Purchase Interval	0.172738	-0.114832	-0.754739	0.696833
ReturnRate	-0.318584	-0.083755	-0.207015	0.609355
AvgOrderPrice	-0.081781	0.158126	-0.442457	0.366112
Day_sin	-0.130531	-0.259856	0.360933	0.029454

Кінець таблиці 3.3

Class	*Class 0*	*Class 1*	*Class 2*	*Class 3*
ReturnFlag	-0.204059	0.355970	-0.032629	-0.119283

Клас 0 демонструє помірні значення витрат і частоти, а також негативні коефіцієнти для ReturnRate, що дозволяє трактувати цей сегмент як лояльних, стабільних клієнтів, які роблять покупки без суттєвих коливань у поведінці.

Найбільший позитивний вплив на ймовірність потрапляння до Класу 1 мають частота покупок, загальна сума витрат NetMonetary та сума без урахування повернень Monetary, що свідчить про те, що саме цей клас об’єднує найактивніших і найвигідніших клієнтів. Також важливою є ознака FirstPurchase, яка показує, що ці користувачі співпрацюють із платформою вже тривалий час.

Клас 2, навпаки, характеризується найвищим значенням Recency – тобто давністю останньої покупки, а також низькими значеннями частоти й суми витрат, що дозволяє інтерпретувати цей сегмент як неактивних або втрачених клієнтів.

Клас 3 має негативний коефіцієнт для FirstPurchase, що свідчить про те, що це нові користувачі. Водночас у них спостерігається позитивний вплив з боку PurchaseInterval і ReturnRate, тобто ці клієнти нещодавно почали купувати, здійснили кілька покупок і часом здійснюють повернення.

Таким чином, аналіз коефіцієнтів підтверджує змістовну кластеризацію клієнтів і дозволяє інтерпретувати поведінкові патерни кожної групи.

В результаті порівняння трьох моделей класифікації – логістичної регресії, дерева рішень та багатошарового перцептрона, найкращі результати продемонструвала логістична регресія.

Головною причиною обрати саме цю модель є те, що класи у задачі виявились лінійно відокремленими, завдяки попередній обробці та масштабуванню ознак. У таких умовах лінійна модель працює не лише

ефективно, а й стабільно, на відміну від більш складних алгоритмів, які можуть перенавчатися або бути надмірно чутливими до параметрів.

Через те, що класи є лінійно роздільними, альтернативою до логістичної регресії може бути побудова ручних правил класифікації, заснований на результатах кластеризації, проте такий підхід погано масштабується на великі обсяги даних і не дає гнучких меж між класами.

Натомість логістична регресія дає систематичне та оптимізоване рішення, автоматично визначаючи вагу кожної ознаки, і дозволяє легко масштабувати модель та адаптувати її до нових умов. Крім того, вона дозволяє чітко інтерпретувати вплив кожної ознаки на результат, що особливо важливо для бізнес-застосувань, де обґрунтованість рішень має велике значення.

Висновки до розділу 3

В цьому розділі було реалізовано повний цикл побудови гібридної моделі класифікації клієнтів згідно їхнього профіля поведінки. Було розглянуто та порівняно декілька моделей для задачі класифікації і задачі кластеризації, як частини попередньої обробки даних та визначено найкращу комбінацію алгоритмів.

Перед реалізацією алгоритмів, було здійснено розвідувальний аналіз обраного датасету, виявлено аномалії в даних та згенеровано допоміжні атрибути на основі яких в подальшому було побудовано моделі.

Для сегментації клієнтів найкращим алгоритмом визнано класичний підхід К-середніх, які кластеризував користувачів на 4 кластери, відповідно найактивніші, лояльні, неактивні та нові клієнти.

На основі міток кластеризації було реалізовано моделі кластеризації, та протестовано три різні підходи до класифікації: логістична регресія, дерево рішень, багатошаровий перцептрон з одним прихованим шаром. Найкращі

результати показала модель логістичної регресії, яка класифікує користувачів з точністю 98%.

Також, під час розробки моделей, було виявлено, що класи є лінійно роздільними, що дає можливість побудувати ручні правила класифікації, засновані на результатах кластеризації. Проте зважаючи на мінуси такого підходу оптимальним варіантом залишається логістична регресія.

Таким чином, побудована гібридна модель успішно реалізує завдання класифікації клієнтів у системі електронної комерції та може бути використана для подальшої персоналізації пропозицій або прогнозування поведінки користувачів.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

4.1 Постановка задачі проектування

В заданому розділі буде проведено оцінювання основних характеристик для майбутнього програмного продукту, що спеціалізується на побудові прогнозу класу профілів користувачів на основі даних транзакцій.

Функціонально-вартісний аналіз (ФВА) передбачає собою технологію, що дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. ФВА проводиться з метою виявлення резервів зниження витрат за рахунок ефективніших варіантів виробництва, кращого співвідношення між споживчою вартістю виробу та витратами на його виготовлення. Для проведення аналізу використовується економічна, технічна та конструкторська інформація.

Алгоритм функціонально-вартісного аналізу включає в себе визначення послідовності етапів розробки продукту, визначення повних витрат (річних) та кількості робочих часів, визначення джерел витрат та кінцевий розрахунок вартості програмного продукту.

Основні технічні вимоги до програмного продукту:

– сумісність зі стандартними компонентами персональних комп'ютерів;

– зрозумілість та зручність інтерфейсу для користувача;

– швидке опрацювання даних та миттєвий доступ до інформації;

– можливість легко масштабувати та підтримувати програмний продукту;

– мінімізація витрат на впровадження програмного продукту

4.2 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка програмного продукту, який дозволяє здійснювати сегментацію клієнтів на основі даних транзакцій, навчати моделі класифікації та здійснювати автоматичне віднесення нових клієнтів до визначених сегментів для покращення бізнес-рішень у сфері електронної комерції. Взявши за основу дану функцію, можна виокремити наступні підфункції:

F_1 – вибір мови програмування.

1. Python.

2. R.

F_2 – вибір модуля побудови моделей.

1. Scikit-learn.

2. TensorFlow & Keras.

F_3 – вибір середовища розробки.

1. PyCharm.

2. Kaggle Notebook.

3. Jupyter Notebook.

Кожна функція має декілька варіантів реалізації. Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1).

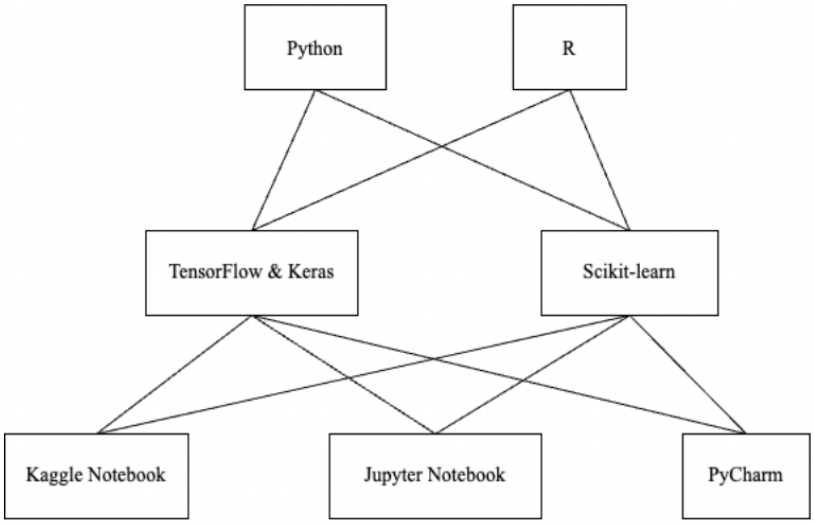

Рисунок 4.1 – Морфологічна карта системи

На основі морфологічної карти побудуємо позитивно-негативну матрицю, яка наведена у таблиці 4.1.

Таблиця 4.1 – Позитивно-негативна матриця

Функція	*Варіант реалізації*	*Переваги*	*Недоліки*
F_1	1	Простий синтаксис, велика кількість модулів з реалізованими структурами моделей машинного та глибокого навчання	Нижча швидкість навчання, потреба у великій кількості пам’яті для інтеграції методів та моделей із модулів
	2	Вдосконалені пакети статистичних методів, висока якість візуалізацій	Спеціалізований синтаксис, що не схожий на інші мови програмування, нижча інтеграція у веб-додатки

Кінець таблиці 4.1

Функція	*Варіант реалізації*	*Переваги*	*Недоліки*
F_3	1	Інтегрований для роботи з великими наборами даних та складними моделями, широкий вибір методів	Нижча гнучкість роботи, особливо з низькорівневими операціями
	2	Зручний у використанні, динамічний граф обчислень	Обмеженість функціоналу для попередньої обробки даних
F_3	1	Потужне середовище розробки IDE, ефективний функціонал для розробки, налагодження і тестування	Низька інтерактивність, нижча ефективність для дослідження даних і побудови візуалізацій
	2	Інтерактивний додаток, швидка результативність	Обмеженість та складність роботи над великими проектами, відсутність потужного IDE
	3	Інтерактивне середовище, зручне для експериментів та візуалізації	Обмежені можливості для роботи з великими проектами, відсутність функціоналу потужного IDE

При аналізі позитивно-негативної матриці, можна помітити, що при розробці програмного продукту деякі варіанти реалізації функцій необхідно відкинути, оскільки вони не відповідають поставленим перед програмним продуктом задачам.

Функція F_1: варіант 1 має вагому перевагу – широкий вибір модулів, що призначені для роботи із задачами машинного та глибокого навчання, а також інтеграція різних методів для досягнення ефективності у навчанні та тестуванні моделей, тоді як варіант 2 є більш обмеженим та менш універсальним через синтаксис. Тому варіант 2 не буде враховуватись надалі.

Функція F_2: варіант 2 відкидається, оскільки Scikit-learn забезпечує необхідний функціонал для традиційних методів машинного навчання, є більш гнучким для попередньої обробки даних та краще підходить для задач нашого дослідження.

Функція F_3: варіант 2 відкидається через свою обмежену інтерактивність та складність роботи над великими проектами. Таким чином, залишаються варіанти 1 та 3. PyCharm забезпечує потужне середовище розробки з ефективним функціоналом для розробки, налагодження і тестування, тоді як Jupyter Notebook є зручним для експериментів та візуалізації даних.

Таким чином, будуть розглянуті наступні варіанти реалізації програмного продукту (ПП):

F_1 1 – F_2 1 – F_3 1

F_1 1 – F_2 1 – F_3 3

Ці варіанти реалізації забезпечують максимальну ефективність та функціональність для розробки програмного продукту, здатного вирішувати завдання здійснювати сегментацію клієнтів на основі даних транзакцій, навчати моделі класифікації та здійснювати автоматичне віднесення нових клієнтів до визначених сегментів для покращення бізнес-рішень у сфері електронної комерції.

4.3 Обґрунтування системи параметрів програмного продукту

На основі даних, розглянутих вище, визначаються основні параметри вибору, які будуть використані для розрахунку коефіцієнта технічного рівня. Для того, щоб охарактеризувати програмне забезпечення, будуть використовуватись наступні параметри:

– $X1$: ефективність виконання коду;

– *X*2: об’єм оперативної пам’яті;

– *X*3: час навчання моделей;

– *X*4: обсяг необхідного програмного коду.

Гірші, середні і кращі значення параметрів вибираються на основі вимог потреб для навчання моделей та умов, що характеризують експлуатацію програмного продукту, як показано у таблиці 4.2.

Таблиця 4.2 – Оцінка параметрів

Назва параметра	*Умовні позначення*	*Одиниці виміру*	*Значення параметра*		
			Гірші	*Середні*	*Кращі*
Швидкодія мови програмування	*X*1	оп/мс	4000	10000	16000
Об’єм пам’яті	*X*2	Гб	4	8	16
Час навчання моделей	*X*3	с	1200	600	200
Обсяг програмного коду	*X*4	Кількість рядків коду	1600	1000	600

4.4 Аналіз експертного оцінювання

Після детального обговорення та аналізу кожний експерт оцінює ступінь важливості кожного параметра для конкретної цілі – розробка програмного продукту, який забезпечує найбільш точні результати при прогнозуванні класів користувачів у сфері електронної комерції та обчисленні прогнозованих значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія з 7 осіб. Визначення коефіцієнтів значимості передбачає:

– визначення рівня значимості параметра шляхом присвоєння різних рангів;
– перевірку придатності експертних оцінок для подальшого використання;
– визначення оцінки попарного пріоритету параметрів;
– обробку результатів та визначення коефіцієнта значимості.

Результати експертного ранжування наведені у таблиці 4.3.

Таблиця 4.3 – Результати експертного ранжування

Позначення параметра	*Назва параметра*	*Ранг параметра за оцінкою експерта*							*Сума рангів* R_i	*Відхилення* Δ_i	Δ_i^2
		1	2	3	4	5	6	7			
X1	Швидкодія мови програмування	2	2	1	3	2	1	2	13	-4.5	20.25
X2	Об’єм оперативної пам’яті	1	1	2	1	1	2	2	10	-7.5	56.25
X3	Час навчання моделей	3	4	3	3	4	4	3	24	6.5	42.25
X4	Обсяг програмного коду	4	3	4	3	3	3	3	23	5.5	30.25
Разом		10	10	10	10	10	10	10			149

Для перевірки степені достовірності експертних оцінок визначаються наступні параметри:

1. Сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^{N} r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70, \quad (4.1)$$

де N – число експертів;

n – кількість параметрів.

2. Середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17{,}5. \quad (4.2)$$

3. Відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T. \quad (4.3)$$

Сума відхилень по всім параметрам повинна дорівнювати 0.

4. Загальна сума квадратів відхилення:

$$S = \sum_{i=1}^{N} \Delta_i^2 = 137. \quad (4.4)$$

Коефіцієнт узгодженості обчислюється за формулою:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 197}{7^2(4^3 - 4)} = 0{,}754 > W_k = 0{,}56. \quad (4.5)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжування, буде проведено попарне порівняння всіх параметрів та занесення результатів у таблиці 4.4.

Таблиця 4.4 – Попарне порівняння параметрів

Параметри	*Експерти*							*Кінцева оцінка*	*Числове значення*
	1	2	3	4	5	6	7		
X1 і X2	>	>	<	>	>	<	=	>	1.5
X1 і X3	<	<	<	=	<	<	<	<	0.5
X1 і X4	<	<	<	=	<	<	<	<	0.5
X2 і X3	<	<	<	<	<	<	<	<	0.5
X2 і X4	<	>	<	<	<	<	<	<	0.5
X3 і X4	<	<	<	=	>	>	=	>	1.5

Числове значення, що визначає ступінь переваги і-го параметра над j-м визначається по формулі:

$$a_{ij} = \begin{cases} 1.5 \text{ при } X_i > X_j \\ 1.0 \text{ при } X_i = X_j \\ 0.5 \text{ при } X_i < X_j \end{cases}. \quad (4.6)$$

З отриманих числових оцінок переваги складемо матрицю А=$\| a_{ij} \|$.

Для кожного параметра зробимо розрахунок вагомості за наступними формулами:

$$K_{\text{ві}} = \frac{b_i}{\sum_{i=1}^{n} b_i}, \quad (4.7)$$

$$b_i = \sum_{i=1}^{N} a_{ij} . \quad (4.8)$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{\text{Bi}} = \frac{b_i'}{\sum_{i=1}^{n} b_i'}, \quad (4.9)$$

$$b_i' = \sum_{i=1}^{N} a_{ij} b_j . \quad (4.10)$$

Зобразимо результати розрахунків вагомості параметрів у таблиці 4.5.

Таблиця 4.5 – Результати розрахунків вагомості параметрів

Параметри X_i	*Параметри* X_j				*Перша ітер.*		*Друга ітер.*	
	X1	X2	X3	X4	b_i	K_{Bi}	$b^1{}_i$	$K^1{}_{Bi}$
X1	1	1,5	0,5	0,5	3,5	0,22	12,25	0,21
X2	0,5	1	0,5	0,5	2,5	0,16	9,25	0,15
X3	1,5	1,5	1	1,5	5,5	0,34	21,25	0,36
X4	1,5	1,5	0,5	1	4,5	0,28	16,25	0,28
Всього					16	1	59	1

Після другої ітерації різниця значень коефіцієнтів вагомості не перевищує 2%, тому подальші ітерації не потребуються.

4.5 Аналіз рівня варіантів реалізації функцій

Рівень якості кожного варіанту виконання основних функцій визначається окремо. Абсолютні значення параметрів X1 (швидкодія мови 62 програмування), X2 (об'єм оперативної пам'яті), X3 (час навчання моделей), X4 (об'єм програмного коду) відповідають технічним вимогам умов функціонування даного програмного продукту (ПП).

Коефіцієнт технічного рівня для кожного варіанту реалізації ПП розраховується так (таблиця 4.6):

Таблиця 4.6 – Розрахунок рівня якості

Основні функції	*Варіант реалізації функції*	*Параметри*	*Абсолютне значення параметра*	*Бальна оцінка параметра*	*Коефіцієнт вагомості параметра*	*Коефіцієнт рівня якості*
F1	А	X1	12 000	16	0,21	3,36
F2	А	X2	8	24	0,15	3,6
	Б	X3	300	21	0,36	7,56
F3	А	X4	800	13	0,28	3,64

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 4.6):

$$K_K(j) = \sum_{i=1}^{n} K_{вi,j} B_{i,j}\,, \qquad (4.11)$$

де n – кількість параметрів;

$K_{вi}$– коефіцієнт вагомості i–го параметра;

B_i – оцінка i–го параметра в балах.

За даними з таблиці 4.6. за формулою:

$$K_K = K_{\text{ТУ}}[F_{1k}] + K_{\text{ТУ}}[F_{2k}] + \ldots + K_{\text{ТУ}}[F_{zk}], \quad (4.12)$$

Рівень якості кожного з варіантів:

$$K_{K1} = 3{,}36 + 3{,}6 + 3{,}64 = 10{,}6,$$
$$K_{K2} = 3{,}36 + 7{,}56 + 3{,}64 = 14{,}56.$$

Аналізуючи розрахунки, можна побачити, що другий варіант є кращим, адже коефіцієнт технічного рівня має найбільше значення.

4.6 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку необхідно провести розрахунок трудомісткості.

Всі варіанти охоплюють два окремих завдання.

1. Розробка проекту програмного продукту.
2. Навчання моделей та проведення досліджень.

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 – до групи Б. За складністю алгоритми, що використовуються у завданні 1 належать до групи 1, а в завданні 2 – до групи 3.

Далі буде проведено розрахунок норм часу на розробку та програмування для кожного із завдань.

Загальна трудомісткість обчислюється як:

$$\mathrm{Т_О} = \mathrm{Т_Р} \cdot \mathrm{К_П} \cdot \mathrm{К_{СК}} \cdot \mathrm{К_М} \cdot \mathrm{К_{СТ}} \cdot \mathrm{К_{СТ.М}}, \quad (4.13)$$

де T_P – трудомісткість розробки ПП;

$K_П$ – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

$K_М$ – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення.

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1, трудомісткість становить $T_p = 36$ людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання становить $K_{\text{П}} = 1{,}8$. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань становить $K_{CK} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, тоді застосовується коефіцієнт $K_{CT} = 0{,}9$. Тоді загальна трудомісткість програмування першого завдання дорівнює $T_1 = 36 \cdot 1{,}8 = 58{,}32$ людино-днів.

Далі будуть проведені аналогічні розрахунки інших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто $T_{\text{Р}} = 28$ людино-днів, $K_{\text{П}} = 0{,}9$, $K_{\text{СК}} = 1$, $K_{\text{СТ}} = 0{,}8$: $T_2 = 28 \cdot 0{,}9 \cdot 0{,}8 = 20,16$ людино-днів.

Складається трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їхню трудомісткість:

$$\mathrm{Т}_I = (58{,}32 + 20{,}16 + 11{,}82) \cdot 8 = 722{,}4 \text{ людино} - \text{години},$$
$$\mathrm{Т}_{II} = (58{,}32 + 20{,}16 + 14{,}1) \cdot 8 = 740{,}64 \text{ людино} - \text{години}.$$

Найбільш високу трудомісткість має варіант №2.

У розробці братимуть участь один інженер з машинного навчання та один програміст для розробки та оптимізації коду. Оклад першого становить 20 000

грн, оклад другого – 18 000 грн. Середня зарплата за годину розраховується за формулою:

$$С_ч = \frac{М}{T_m \cdot t} грн, \quad (4.14)$$

де М – місячний оклад працівників;

T_m – кількість робочих днів на місяць;

t – кількість робочих годин в день.

Тоді, розрахуємо заробітну плату за формулою:

$$С_ч = \frac{20000+18000}{3 \cdot 23 \cdot 8} = 68{,}84.$$

Заробітна плата розраховується за формулою:

$$С_{ЗП} = С_ч \cdot Т_i \cdot К_Д, \quad (4.16)$$

де $С_ч$– величина погодинної оплати праці програміста;

T_i – трудомісткість відповідного завдання;

$К_Д$ – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

I. $С_{ЗП} = 68{,}84 \cdot 722{,}4 \cdot 1.2 = 59\,676{,}01$ грн.

II. $С_{ЗП} = 68{,}84 \cdot 740{,}64 \cdot 1.2 = 61\,182{,}78$ грн.

Варто розрахувати єдиний соціальний внесок, що становить 22%:

I. $С_{ВІД} = С_{ЗП} \cdot 0.22 = 59\,676{,}01 \cdot 0.22 = 13\,128{,}72$ грн.

II. $С_{ВІД} = С_{ЗП} \cdot 0.22 = 61\,182{,}78 \cdot 0.22 = 13\,460{,}211$ грн.

Тепер визначимо витрати на оплату однієї машино-години. ($С_М$).

Так як одна ЕОМ обслуговує одного програміста з окладом 18000 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$С_Г = 12 \cdot М \cdot К_З = 12 \cdot 18000 \cdot 0,2 = 43\ 200 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$С_{ЗП} = С_Г \cdot (1 + К_З) = 43\ 200 \cdot (1 + 0.2) = 51\ 840 \text{ грн.}$$

Відрахування на соціальний внесок:

$$С_{ВІД} = С_{ЗП} \cdot 0.22 = 51840 \cdot 0,22 = 11\ 404,8 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 12 000 грн.

$$С_А = К_{ТМ} \cdot К_А \cdot Ц_{ПР} = 1,2 \cdot 0,25 \cdot 12000 = 3\ 600 \text{ грн.},$$

де $К_{ТМ}$– коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

$К_А$– річна норма амортизації;

$Ц_{ПР}$– договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$С_Р = К_{ТМ} \cdot Ц_{ПР} \cdot К_Р = 1,2 \cdot 0,09 \cdot 12000 = 1\ 296 \text{ грн},$$

де $К_Р$– відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$Т_{ЕФ} = (Д_К – Д_В – Д_С – Д_Р) \cdot t_З \cdot К_В = (365 – 104 – 12 – 14) \cdot 8 \cdot 0.36 =$$

$$= 676,8 \text{ години},$$

де $Д_К$ – календарна кількість днів у році;

$Д_В$, $Д_С$ – відповідно кількість вихідних та святкових днів;

$Д_Р$ – кількість днів планових ремонтів устаткування;

t –кількість робочих годин в день;

$К_В$– коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$С_{ЕЛ} = Т_{ЕФ} \cdot N_С \cdot К_З \cdot Ц_{ЕН} = 676,8 \cdot 0,2 \cdot 0,35 \cdot 9,43 = 446,75 \text{ грн},$$

де $N_С$ – середньо-споживча потужність приладу;

$К_З$ – коефіцієнтом зайнятості приладу;

$Ц_{ЕН}$ – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$С_Н = Ц_{ПР} \cdot 0.67 = 12\,000 \cdot 0,67 = 8040 \text{ грн}.$$

Тоді, річні експлуатаційні витрати будуть:

$$С_{ЕКС} = С_{ЗП} + С_{ВІД} + С_А + С_Р + С_{ЕЛ} + С_Н, \qquad (4.17)$$

$$С_{ЕКС} = 51840 + 11404,8 + 3600 + 1\,296 + 446,75 + 8040 = 76\,627,55 \text{ грн}.$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$С_{М\text{-}Г} = С_{ЕКС} / Т_{ЕФ} = 76\ 627{,}55 / 676{,}8 = 113,\ 22 \quad \text{грн/год.}$$

Оскільки в даному випадку всі роботи, які пов‘язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$С_М = С_{М-Г} \cdot Т, \qquad (4.18)$$

I. $С_М = 113,\ 22 \cdot 722{,}4 = 81\ 790{,}12$ грн.

II. $С_М = 113,\ 22 \cdot 740{,}64 = 83\ 855{,}26$ грн.

Накладні витрати складають 67% від заробітної плати:

$$С_Н = С_{ЗП} \cdot 0{,}67, \qquad (4.19)$$

I. $С_Н = 59\ 676{,}01 \cdot 0{,}67 = 39\ 982{,}92$ грн.

II. $С_Н = 61\ 182{,}78 \cdot 0{,}67 = 40\ 992{,}46$ грн.

Отже, вартість розробки ПП за варіантами становить:

$$С_{ПП} = С_{ЗП} + С_{ВІД} + С_М + С_Н, \qquad (4.20)$$

I. $С_{ПП} = 59\ 676{,}01 + 13\ 128{,}72 + 81\ 790{,}12 + 39\ 982{,}92 = 194\ 577{,}77$ грн.

II. $С_{ПП} = 61\ 182{,}78 + 13\ 460{,}211 + 83\ 855{,}26 + 40\ 992{,}46 = 199\ 490{,}71$ грн.

4. 7 Вибір кращого варіанту ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$К_{ТЕР\,j} = К_{Кj} / С_{Фj}, \quad (4.21)$$

I. $К_{ТЕР1} = 10,6 / 194\,577,77 = 5,44 \cdot 10^{-5}$

II. $К_{ТЕР2} = 14,56 / 199\,490,71 = 7,3 \cdot 10^{-5}$

Найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $К_{ТЕР1} = 5,44 \cdot 10^{-5}$.

Після виконання функціонально-вартісного аналізу програмного комплексу, що розроблюється, можна зробити висновок, що з альтернатив, які залишились після першого відбору двох варіантів виконання програмного комплексу, оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості.

Цей варіант реалізації програмного продукту має такі параметри:

– мова програмування для реалізації: Python;

– використання комбінації модулів Scikit-lean;

– використання Jupyter Notebook у якості середовища розробки.

Цей варіант виконання програмного комплексу дає користувачеві можливість ефективно проводити навчання моделей нейронних мереж для задачі розробки системи підтримки прийняття рішень для прогнозування трендів у галузі соціальних медіа у зручному середовищі.

Висновки до розділу 4

В цій частині проведено повний функціонально-вартісний аналіз програмного забезпечення. Також було знайдено оцінку основних функцій програмного продукту. В результаті виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, було визначено та проведено оцінку основних функцій програмного забезпечення, а також знайдено параметри, які його характеризують.

ВИСНОВКИ

У ході виконання дипломної роботи було здійснено аналіз методів машинного навчання, які використовуються для сегментації клієнтів у сфері електронної комерції. Основну увагу приділено методам кластеризації (зокрема, алгоритму K-середніх) та класифікації (логістична регресія), що дозволяють автоматизовано групувати клієнтів за схожими патернами поведінки та прогнозувати належність нових користувачів до відповідних сегментів.

Для дослідження було використано реальний датасет транзакцій, на основі якого виконано попередню обробку, побудовано поведінкові метрики а також проведено генерацію нових ознак. Було проведено масштабування даних, усунення викидів і аналіз розподілу значень.

У процесі кластеризації було випробувано декілька методів, з яких K-середніх продемонстрував найкраще групування користувачів за показником силуету. На основі результатів кластеризації побудовано моделі класифікації, які дозволяють визначити до якого сегменту належить новий користувач. Найвищі показники точності, recall та F1-score були отримані за допомогою логістичної регресії, що пояснюється лінійною відокремленістю сегментів після масштабування.

Отримані результати показують високу ефективність поєднання кластеризації з подальшою класифікацією для вирішення завдань поведінкової сегментації. Розроблений підхід дозволяє автоматизувати аналіз клієнтської бази, підвищити точність персоналізованих маркетингових стратегій та покращити прогнозування поведінки користувачів. Запропонована модель може бути адаптована до інших завдань у сфері електронної комерції, зокрема для прогнозування відтоку клієнтів, підбору індивідуальних пропозицій або оцінки життєвої цінності клієнта.

Запропонована модель може бути адаптована до інших завдань у сфері e-commerce, зокрема для прогнозування відтоку клієнтів, підбору індивідуальних пропозицій або оцінки життєвої цінності клієнта.

ПЕРЕЛІК ПОСИЛАНЬ

1. What is customer segmentation? URL: https://www.businessnewsdaily.com/15973-what-is-customer-segmentation.html (дата звернення: 04.06.2025)
2. Segmentation statistics. URL: https://www.notifyvisitors.com/blog/segmentation-statistics/ (дата звернення: 04.06.2025)
3. Як мультиплекс збільшив продажі на 50% завдяки сегментації. URL: https://esputnik.com/uk/blog/keys-multiplex-na-50-bilshe-prodazhiv (дата звернення: 04.06.2025)
4. Які дані збирає «Сільпо» про покупців. URL: https://rau.ua/novyni/kakie-dannye-sobiraet-silpo/ (дата звернення: 04.06.2025)
5. 8 технологічних революцій України: четверта – Big Data. URL: https://techiia.com/ua/news/8-tehnolognih-revolyucij-ukrayini-revolyuciya-chetverta-big-data (дата звернення: 04.06.2025)
6. Lead scoring definition. URL: https://www.techtarget.com/searchcustomerexperience/definition/lead-scoring (дата звернення: 04.06.2025)
7. Erevelles S., Fukawa N., Swayne L. Big data consumer analytics and the transformation of marketing. Journal of Business Research, 2016, Vol. 69, Issue 2. P. 897–904. URL: https://www.mdpi.com/1999-5903/14/5/144
8. What is clickstream data? URL: https://www.rudderstack.com/blog/clickstream-data/ (дата звернення: 04.06.2025)
9. Lăzăroiu G., Neguriţă O., Grecu I., Grecu G., Mitran P.C. Consumer behavior prediction using big data analytics. Journal of Self-Governance and Management Economics, 2020, Vol. 8, Issue 1. P. 24–29. URL: https://www.mdpi.com/0718-1876/20/1/28

10. Carcillo F., Le Borgne Y.A., Caelen O., Bontempi G. AI-based fraud detection in financial transactions: Review of recent advances. *Scientific Reports*, 2020, Vol. 10, Article 1. URL: https://www.nature.com/articles/s41598-020-73622-y
11. How Mastercard uses AI to detect credit card fraud. URL: https://www.businessinsider.com/mastercard-ai-credit-card-fraud-detection-protects-consum (дата звернення: 04.06.2025)
12. Customer segmentation in e-commerce. URL: https://www.csupom.com/uploads/1/1/4/8/114895679/n20p7formatted.pdf
13. Logistic regression. URL: https://en.wikipedia.org/wiki/Logistic_regression
14. K-nearest neighbors algorithm. URL: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
15. Decision tree classification in Python. URL: https://www.datacamp.com/tutorial/decision-tree-classification-python (дата звернення: 04.06.2025)
16. How to train a basic perceptron neural network. URL: https://www.allaboutcircuits.com/technical-articles/how-to-train-a-basic-perceptron-neural-network/ (дата звернення: 04.06.2025)
17. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. Springer, 2009. 745 p. URL:https://www.sas.upenn.edu/~fdiebold/NoHesitations/BookAdvanced.pdf

ДОДАТОК А ЛІСТИНГ ОСНОВНИХ СКЛАДНИКІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

```
import datetime as dt
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sb
import plotly.express as px
from yellowbrick.cluster import SilhouetteVisualizer

from pandas.api.types import CategoricalDtype

from sklearn.preprocessing import RobustScaler, label_binarize
from sklearn.cluster import KMeans, DBSCAN, AgglomerativeClustering
from sklearn.neighbors import NearestNeighbors
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.metrics import silhouette_score, calinski_harabasz_score,
davies_bouldin_score
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score, roc_curve, auc, precision_score, recall_score, f1_score,
roc_auc_score
from sklearn.metrics import RocCurveDisplay

from imblearn.over_sampling import SMOTE
import warnings

warnings.filterwarnings('ignore')

random_state = 42
```

```
df = pd.read_excel('Online_Retail.xlsx')
df.dropna(subset=['CustomerID'], inplace=True)
df['Description_clean'] =
df['Description'].astype(str).str.strip().str.lower()

# Фільтрація службових товарів
mask = df['StockCode'].astype(str).str.len() < 5
df_short_codes = df[mask]
support_items_list = df_short_codes['StockCode'].value_counts().index
df = df[~df['StockCode'].isin(support_items_list)]

# Генерація нових ознак для RFM-аналізу
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
snapshot_date = df['InvoiceDate'].max() + pd.Timedelta(days=1)
rfm = df.groupby('CustomerID').agg({
    'InvoiceDate': lambda x: (snapshot_date - x.max()).days,
    'InvoiceNo': 'nunique',
    'TotalSum': 'sum'
})
rfm.rename(columns={'InvoiceDate': 'Recency',
                    'InvoiceNo': 'Frequency',
                    'TotalSum': 'Monetary'}, inplace=True)

# Масштабування даних
features_for_clustering = ['Recency', 'Frequency', 'Monetary']
scaler = RobustScaler()
X_scaled = scaler.fit_transform(rfm[features_for_clustering])

# Кластеризація K-середніх
kmeans = KMeans(n_clusters=4, random_state=random_state)
rfm['Cluster'] = kmeans.fit_predict(X_scaled)

# Підготовка даних для класифікації
X = rfm.drop('Cluster', axis=1)
y = rfm['Cluster']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=random_state)

# Логістична регресія
```

```
log_reg = LogisticRegression(max_iter=1000)
log_reg.fit(X_train, y_train)
y_pred = log_reg.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("F1 Score:", f1_score(y_test, y_pred, average='weighted'))
print("Classification Report:\n", classification_report(y_test, y_pred))
```