Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДІВ БАГАТОВИМІРНОГО АНАЛІЗУ ЕФЕКТИВНОСТІ ТА МОНІТОРИНГУ ДЛЯ СИСТЕМ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ РЕКЛАМНИМИ КАМПАНІЯМИ
(тема)

Виконав:
здобувач 2 року навчання,
групи ІНФм-24-1
Бойко М. К.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Науковий керівник доц. Кобилін О. А.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Бойко Марії Костянтинівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та реалізація методів багатовимірного аналізу ефективності та моніторингу для систем автоматизованого управління рекламними кампаніями

затверджена наказом університету від 14 листопада 2025 року № 1045Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 8 листопада 2025 р.
3. Вихідні дані до роботи багатовимірні дані рекламних кампаній у соціальних мережах, наукові джерела з багатовимірного аналізу та цифрового маркетингу, алгоритми кластеризації (k-means, ієрархічна кластеризація, DBSCAN), алгоритми зниження розмірності (PCA, t-SNE, UMAP), алгоритми класифікації та прогнозування (логістична регресія, дерева рішень, Random Forest, Gradient Boosting, SVM), програмні засоби Python та бібліотеки для обробки й аналізу даних, тестові вибірки для моделювання рекламних сценаріїв.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз структури та особливостей багатовимірних даних рекламних кампаній.
2. Дослідження методів багатовимірного статистичного аналізу, кластеризації, прогнозування та зниження розмірності.
3. Розроблення алгоритму багатовимірного аналізу ефективності та моніторингу рекламних кампаній.
4. Реалізація програмного застосунку для обробки даних, моделювання та візуалізації результатів.
5. Проведення експериментальної оцінки точності та ефективності застосованих методів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми багатовимірного аналізу даних рекламних кампаній, об'єкт дослідження, мета дослідження, постановка задачі, структурна схема системи автоматизованого моніторингу, блок-схеми алгоритмів кластеризації та прогнозування, візуалізації високорозмірних даних після зниження розмірності, дендрограми й діаграми кластерів, приклади роботи програмного застосунку, результати експериментального порівняння методів, висновки та перспективи застосування.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	29.09.2025	
2	Аналіз завдання, підбір літератури	30.09.25-07.10.25	
3	Аналіз літератури з досліджуваної проблеми	08.10.25-14.10.25	
4	Особливості великих мовних моделей	15.10.25-20.10.25	
5	Дослідження великих мовних моделей	21.10.25-27.10.25	
6	Програмна реалізація	28.10.25-05.11.25	
7	Обґрунтування отриманих результатів	06.11.25-11.11.25	
8	Оформлення пояснювальної записки	12.11.25-14.11.25	
9	Перевірка на нормоконтроль	05.12.25-07.12.25	
10	Перевірка на плагіат	07.12.25-08.12.25	
11	Рецензування	08.12.25-09.12.25	
12	Підготовка презентації та доповіді	09.12.25-10.12.25	
13	Занесення роботи в електронний архів	17.12.25	
14	Попередній захист кваліфікаційної роботи	17.12.25	

Дата видачі завдання 29 вересня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Кобилін О. А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 94 с., 28 рис., 2 дод., 42 джерела.

АРБІТРАЖ ТРАФІКУ, БАГАТОВИМІРНИЙ АНАЛІЗ, КЛАСТЕРИЗАЦІЯ, МЕТОД ГОЛОВНИХ КОМПОНЕНТ, МОНІТОРИНГ, РЕКЛАМНА КАМПАНІЯ, СИСТЕМА АВТОМАТИЗОВАНОГО УПРАВЛІННЯ, ТРЕКІНГОВА СИСТЕМА, API, FACEBOOK, KEITARO.

Об’єктом дослідження є процеси управління й моніторингу рекламних кампаній у соціальній мережі Facebook з використанням трекінгової системи Keitaro.

Метою дослідження є підвищення ефективності управління рекламними кампаніями шляхом розроблення та впровадження методів багатовимірного аналізу та моніторингу, що дозволяють виявляти приховані закономірності у даних та оптимізувати стратегію розміщення реклами.

Використано методи кластерного аналізу, головних компонент, класифікації й регресії для аналізу багатовимірних даних рекламних кампаній. Проаналізовано підходи до моніторингу ефективності цифрової реклами. Досліджено структуру даних трекінгової системи Keitaro та розроблено методику їх обробки для багатовимірного аналізу.

Наукова новизна роботи полягає у комплексному підході до аналізу ефективності рекламних кампаній, що поєднує методи багатовимірного статистичного аналізу з урахуванням специфіки арбітражу трафіку в соціальних мережах.

Рекомендації щодо використання результатів роботи: застосування розроблених методів фахівцями з digital-маркетингу. У результаті дослідження розроблено застосунок для збору і обробки даних для багатовимірного аналізу, визначення аудиторних сегментів й ефективних параметрів таргетингу.

ABSTRACT

Explanatory note to the qualification work: 94 pages, 28 figures, 2 appendixes, 42 sources.

API, AUTOMATED MANAGEMENT SYSTEM, CLUSTERING, FACEBOOK, KEITARO, MONITORING, MULTIDIMENSIONAL ANALYSIS, PRINCIPAL COMPONENT ANALYSIS, PROMOTIONAL CAMPAIGN, TRACKING SYSTEM, TRAFFIC ARBITRAGE.

The object of the study is the processes of managing and monitoring advertising campaigns on the Facebook social network using the Keitaro tracking system.

The aim of the research is to improve the efficiency of advertising campaign management by developing and implementing multivariate analysis and monitoring methods that make it possible to identify hidden patterns in the data and optimize advertising placement strategies.

Methods of cluster analysis, principal component analysis, classification, and regression were used for multivariate analysis of advertising campaign data. Approaches to monitoring the effectiveness of digital advertising were analyzed. The structure of the Keitaro tracking system data was examined, and a methodology for their processing for multivariate analysis was developed.

The scientific novelty of the work lies in a comprehensive approach to analyzing the effectiveness of advertising campaigns, combining methods of multivariate statistical analysis while considering the specifics of traffic arbitrage in social networks.

Recommendations for the use of the results: the developed methods can be applied by digital marketing specialists. As a result of the research, an application was developed for data collection and processing for multivariate analysis, audience segmentation, and identification of effective targeting parameters.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ГБ – гігабайт

МБ – мегабайт

ШІ – штучний інтелект

API – Application Programming Interface (програмний інтерфейс застосунку)

ARIMA – AutoRegressive Integrated Moving Average (автопрегресійна інтегрована модель ковзного середнього)

CAMTA – Causal Attention Model (модель причинно-наслідкової уваги)

CPA – Cost Per Action (вартість цільової дії)

CTA – Call To Action (заклик до дії)

CPC – Cost Per Click (вартість кліка)

CPM – Cost Per Mille (вартість тисячі показів)

CR – Conversion Rate (коефіцієнт конверсії)

CRM – Customer Relationship Management (управління взаємовідносинами з клієнтами)

CSV – Comma-Separated Values (формат текстових даних, у якому значення розділені комами)

CTR – Click-Through Rate (коефіцієнт клікабельності)

DBSCAN – Density-Based Spatial Clustering of Applications with Noise (алгоритм кластеризації на основі щільності)

DVC – Data Version Control (система контролю версій даних)

GDPR – General Data Protection Regulation (загальний регламент захисту даних)

GPU –Graphics Processing Unit (графічний процесор)

GRU – Gated Recurrent Unit (рекурентна одиниця з керованими гейтами)

ICA – Independent Component Analysis (аналіз незалежних компонент)

ISP – Internet Service Provider (постачальник Інтернет-послуг)

IP – Internet Protocol (Інтернет-протокол)

KPI – Key Performance Indicators (ключові показники ефективності)

LIME – Local Interpretable Model-agnostic Explanations (локальні інтерпретовані пояснення, незалежні від моделі)

LSTM – Long Short-Term Memory (довготривала короткочасна пам’ять)

LTV – Lifetime Value (довгострокова цінність клієнта)

NLP – Natural Language Processing (обробка природної мови)

NSGA –Non-dominated Sorting Genetic Algorithm (генетичний алгоритм сортування за недомінованістю)

PC – Personal Computer (персональний комп’ютер)

PCA – Principal Component Analysis (метод головних компонент)

RL – Reinforcement Learning (навчання з підкріпленням)

ROC-AUC – Receiver Operating Characteristic – Area Under the Curve (характеристика робота-приймача – площа під кривою)

ROI – Return on Investment (рентабельність інвестицій)

SaaS – Software as a Service (програмне забезпечення як послуга)

SMOTE – Synthetic Minority Oversampling Technique (метод синтетичного збільшення меншості класів)

SVM – Support Vector Machine (метод опорних векторів)

t-SNE – t-distributed Stochastic Neighbor Embedding (стохастичне вкладення сусідів з t-розподілом)

UCB – Upper Confidence Bound (верхня межа довіри)

UMAP – Uniform Manifold Approximation and Projection (уніфіковане наближення та проєкція многовиду)

URL – Uniform Resource Locator (універсальний локатор ресурсу)

VPN – Virtual Private Network (віртуальна приватна мережа)

ВСТУП

Сучасний цифровий маркетинг характеризується зростанням обсягів та складності даних, що генеруються рекламними кампаніями. Системи автоматизованого управління рекламою збирають детальну інформацію про кожну взаємодію користувача з рекламними оголошеннями, створюючи багатовимірні набори даних, аналіз яких традиційними методами є неефективним. Арбітраж трафіку у соціальних мережах, зокрема Facebook, набув широкого розповсюдження як спосіб монетизації рекламного простору. Ефективність таких кампаній критично залежить від здатності оперативно аналізувати великі обсяги даних, виявляти закономірності та приймати обґрунтовані рішення щодо оптимізації параметрів таргетингу і розподілу бюджету. Актуальність роботи – необхідність розроблення та впровадження сучасних методів багатовимірного аналізу для підвищення ефективності систем автоматизованого управління рекламними кампаніями. Традиційні підходи, що базуються на аналізі окремих метрик ефективності, не враховують складні взаємозв’язки між параметрами кампанії та їх спільний вплив на кінцевий результат. Застосування методів багатовимірного статистичного аналізу дозволяє виявити приховані закономірності у даних, сегментувати аудиторію за комплексом характеристик та оптимізувати стратегію розміщення реклами.

Сучасний стан досліджень у цій галузі характеризується активним розвитком методів машинного навчання та інтелектуального аналізу даних для маркетингових задач. Провідні науково-дослідні центри та технологічні компанії, такі як Google, Meta, а також спеціалізовані аналітичні платформи розробляють нові підходи до аналізу поведінки користувачів та оптимізації рекламних кампаній. Однак більшість існуючих рішень є комерційними продуктами закритого типу, що обмежує можливості їх адаптації під специфічні потреби арбітражу трафіку. У вітчизняній практиці арбітражу трафіку широко використовується трекінгова система Keitaro, що забезпечує детальний збір даних про взаємодію користувачів з рекламою. Проте наявні інструменти

аналітики у цій системі обмежуються стандартними звітами та не використовують потенціал методів багатовимірного аналізу для виявлення складних закономірностей. Наукова задача, що вирішується у роботі, полягає у дослідженні можливостей застосування методів багатовимірного статистичного аналізу для підвищення ефективності систем моніторингу та управління рекламними кампаніями шляхом виявлення прихованих закономірностей у високорозмірних даних та розроблення інтегральних метрик якості трафіку.

Об'єкт дослідження э процеси управління та оптимізації рекламних кампаній у системах автоматизованого арбітражу трафіку.

Предмет дослідження э методи багатовимірного статистичного аналізу та машинного навчання для виявлення закономірностей у даних рекламних кампаній та підвищення їх ефективності.

Для досягнення поставленої мети необхідно проаналізувати існуючі методи багатовимірного статистичного аналізу та визначити найбільш придатні для аналізу даних рекламних кампаній, розробити архітектуру програмного комплексу для інтеграції методів кластерного аналізу, класифікації та регресійного моделювання. Важливими етапами роботи є реалізація програмних модулів для кластеризації рекламних кампаній методами K-means та ієрархічної кластеризації з оцінкою якості сегментації, а також розробка системи класифікації для прогнозування успішності кампаній з використанням алгоритмів Random Forest та дерев рішень. Окрему увагу приділено створенню регресійних моделей для прогнозування фінансових показників ефективності рекламних кампаній та розробці засобів візуалізації результатів багатовимірного аналізу для підтримки прийняття управлінських рішень. Завершальним етапом є проведення експериментального дослідження розробленого програмного комплексу на реальних даних рекламних кампаній та виконання порівняльного аналізу ефективності різних методів.

1 ОГЛЯД ОСНОВНИХ МЕТОДІВ БАГАТОВИМІРНОГО АНАЛІЗУ ТА МОНІТОРИНГУ РЕКЛАМНИХ КАМПАНІЙ

1.1 Сучасний стан систем автоматизованого управління рекламними кампаніями

Системи автоматизованого управління рекламними кампаніями набули широкого розповсюдження у сфері цифрового маркетингу та арбітражу трафіку. Основним завданням таких систем є оптимізація рекламних витрат шляхом автоматичного розподілу бюджету між найефективнішими каналами залучення користувачів.

У сучасних умовах арбітражу трафіку використовуються спеціалізовані трекінгові системи, серед яких провідне місце займає платформа Keitaro. Ця система дозволяє відстежувати шлях користувача від першого контакту з рекламним оголошенням до цільової дії, збираючи при цьому детальну статистику за множиною параметрів.

Аналіз рекламних кампаній у соціальній мережі Facebook (Meta) характеризується високою багатовимірністю даних. Кожна взаємодія користувача з рекламою генерує набір атрибутів, що включають демографічні характеристики, географічне розташування, характеристики пристрою, час взаємодії та інші параметри. Ефективне управління такими кампаніями вимагає застосування методів багатовимірного аналізу для виявлення прихованих закономірностей та оптимізації стратегії розміщення реклами.

Традиційні підходи до аналізу рекламних кампаній базуються на простих метриках ефективності, таких як CTR (Click-Through Rate), CPC (Cost Per Click), CPA (Cost Per Action) та ROI (Return on Investment). Однак такий підхід не враховує складні взаємозв’язки між різними параметрами кампанії та їх вплив на кінцевий результат.

1.1.1 Основні принципи арбітражу трафіку та цифрового маркетингу

Системи автоматизованого управління рекламними кампаніями набули широкого розповсюдження у сфері цифрового маркетингу та арбітражу трафіку. Арбітраж трафіку являє собою бізнес-модель, що базується на купівлі трафіку за низькою ціною та його перепродажу за вищою ціною через монетизацію за допомогою партнерських програм, офферів або прямої реклами.

Основним завданням систем автоматизованого управління є оптимізація рекламних витрат шляхом автоматичного розподілу бюджету між найефективнішими каналами залучення користувачів. Ефективність таких систем критично залежить від здатності швидко аналізувати великі обсяги даних та приймати обґрунтовані рішення щодо коригування параметрів кампаній.

Сучасний арбітраж трафіку характеризується високою конкуренцією та динамічністю ринку. Вартість залучення користувачів постійно зростає, що вимагає від арбітражників застосування складних аналітичних методів для підтримання рентабельності кампаній [1]. Традиційні підходи до управління рекламою, що базуються на ручному аналізі статистики та інтуїції, втрачають ефективність у умовах зростання обсягів даних та швидкості змін ринкової ситуації.

1.1.2 Трекінгові системи та платформи моніторингу

У сучасних умовах арбітражу трафіку використовуються спеціалізовані трекінгові системи, серед яких провідне місце займає платформа Keitaro. Трекінгова система виконує функції збору, зберігання та первинної обробки даних про взаємодію користувачів з рекламними оголошеннями.

Система Keitaro дозволяє відстежувати шлях користувача від першого контакту з рекламним оголошенням до цільової дії (конверсії), збираючи при цьому детальну статистику за множиною параметрів. Архітектура системи

базується на серверній моделі з використанням бази даних для зберігання подій та вебінтерфейсу для візуалізації даних.

Основні компоненти трекінгової системи включають:

– модуль збору даних, що фіксує всі взаємодії користувача через спеціальні пікселі відстеження та postback-повідомлення;

– модуль розподілу трафіку, що направляє користувачів на різні цільові сторінки (лендінги) згідно з заданими правилами та ваговими коефіцієнтами;

– модуль аналітики, що агрегує зібрані дані та надає інструменти для їх візуалізації у вигляді таблиць, графіків та звітів;

– модуль інтеграції з зовнішніми системами через API для автоматичного обміну даними з рекламними платформами та партнерськими мережами.

Система використовує багаторівневу модель атрибуції, що дозволяє відстежувати джерело трафіку на різних рівнях деталізації: кампанія, рекламна група, оголошення, ключове слово, креатив [2, 3]. Кожному рівню відповідають спеціальні параметри (токени), що передаються у URL-адресі та дозволяють ідентифікувати конкретний елемент рекламної кампанії.

1.1.3 Інтеграція з рекламними платформами Facebook

Ефективний моніторинг рекламних кампаній вимагає інтеграції трекінгової системи з рекламними платформами, зокрема Facebook Ads Manager. Facebook є однією з найпопулярніших платформ для розміщення реклами завдяки величезній аудиторії, розвиненим інструментам таргетингу та можливостям оптимізації доставки оголошень.

Інтеграція здійснюється через Facebook Marketing API, що надає програмний доступ до даних кампаній, рекламних груп (ad sets) та оголошень (ads). Через API можна отримувати детальну статистику у розрізі різних параметрів таргетингу, часових періодів та географічних регіонів. API також

дозволяє автоматично створювати та модифікувати елементи рекламних кампаній без використання вебінтерфейсу.

Система Keitaro підтримує автоматичну синхронізацію витрат з Facebook через періодичне опитування API. Це дозволяє об’єднати дані про витрати з рекламної платформи з даними про конверсії з трекінгової системи для розрахунку реальної рентабельності кампаній.

Особливістю Facebook як рекламної платформи є складна система алгоритмічної оптимізації доставки оголошень. Платформа використовує машинне навчання для прогнозування ймовірності цільової дії користувача та автоматично коригує ставки та частоту показів для максимізації результату при заданому бюджеті. Це створює додаткові виклики для арбітражників, оскільки ефективність кампанії залежить не тільки від якості креативів та таргетингу, але й від правильного налаштування алгоритмів оптимізації.

1.1.4 Метрики ефективності рекламних кампаній

Оцінка ефективності рекламних кампаній базується на системі ключових показників (KPI), що відображають різні аспекти взаємодії користувачів з рекламою та економічну ефективність інвестицій.

Показник CTR визначається як відношення кількості кліків до кількості показів рекламного оголошенняза формулою

$$CTR = \frac{\text{Кліки}}{\text{Покази}} \cdot 100\ \%.$$

Цей показник характеризує привабливість рекламного креативу та його релевантність для цільової аудиторії. Високий CTR свідчить про те, що оголошення викликає інтерес у користувачів та спонукає їх до взаємодії.

Вартість кліка (CPC) розраховується як відношення витрат на рекламу до кількості отриманих кліків за формулою

$$CPC = \frac{\text{Витрати}}{\text{Кліки}}.$$

Показник CPC дозволяє оцінити ефективність торгів за рекламні місця та оптимізувати ставки. У різних джерелах трафіку та для різних ніш вартість кліка може суттєво відрізнятися.

Вартість цільової дії (CPA) визначає вартість залучення одного користувача, що здійснив цільову дію, де конверсія може включати реєстрацію, покупку, підписку, встановлення додатку або іншу цільову дію залежно від специфіки кампанії та бізнес-моделі. CPA розраховується як

$$CPA = \frac{\text{Витрати}}{\text{Конверсії}}.$$

Коефіцієнт конверсії (CR) показує частку користувачів, що здійснили цільову дію, і визначається як

$$CR = \frac{\text{Конверсії}}{\text{Кліки}} \cdot 100\ \%.$$

Цей показник характеризує якість трафіку та відповідність між очікуваннями користувачів, сформованими рекламним оголошенням, та реальним змістом цільової сторінки.

Традиційні показники не завжди повністю відображують якість залученого трафіку та довгострокову ефективність рекламних інвестицій. Для комплексної оцінки доцільно використовувати додаткові метрики, що враховують поведінку користувачів та їх цінність для бізнесу.

Показник ROI характеризує рентабельність рекламних інвестицій та розраховується як відношення прибутку до витрат. Lifetime Value оцінює довгострокову цінність залученого користувача з урахуванням всіх його

транзакцій за певний період. Співвідношення LTV до CPA є критичним показником довгострокової прибутковості рекламної кампанії.

1.2 Методи багатовимірного статистичного аналізу

Аналіз рекламних кампаній у соціальній мережі Facebook характеризується високою багатовимірністю даних. Кожна взаємодія користувача з рекламою генерує набір атрибутів, що включають демографічні характеристики (вік, стать), географічне розташування (країна, регіон, місто), характеристики пристрою (тип, операційна система, браузер), час взаємодії, параметри креативу та інші показники (CTR, CPC, конверсії). Ефективне управління такими кампаніями вимагає застосування методів багатовимірного аналізу для виявлення прихованих закономірностей, сегментації аудиторії та оптимізації стратегії розміщення реклами.

1.2.1 Кластерний аналіз

Методи кластерного аналізу дозволяють групувати об’єкти за ступенем подібності їх характеристик без попереднього визначення класів. У контексті рекламних кампаній кластеризація використовується для таких задач:

– сегментації аудиторії, виділення груп користувачів з подібною поведінкою або реакцією на рекламу;

– аналізу креативів, виявлення схожих рекламних оголошень за ефективністю;

– виявлення часових патернів, виявлення періодів з високою чи низькою конверсією.

Алгоритм k-середніх (k-means) – один з найпопулярніших методів кластеризації завдяки простоті та обчислювальній ефективності. Метод ітеративно розбиває множину об’єктів на k кластерів, мінімізуючи суму квадратів відстаней між об’єктами та центрами їхніх кластерів. Основні

переваги: швидкість та зрозуміла інтерпретація результатів. Недоліками є необхідність задавати кількість кластерів заздалегідь та чутливість до початкових центрів кластерів. На рисунку 1.1 зображено алгоритм. Приклад результатів кластеризації даних методом k-середніх – на рисунку 1.2.

Рисунок 1.1 – Алгоритм k-середніх

Ієрархічні методи кластеризації створюють деревоподібну структуру вкладених кластерів, що дозволяє аналізувати дані на різних рівнях деталізації. Агломеративний підхід починає з розгляду кожного об’єкта як окремого кластера та послідовно об’єднує найближчі кластери. Результат часто візуалізують у вигляді дендрограми, яка допомагає обрати оптимальну кількість кластерів іоцінити взаємозв’язки між групами об’єктів.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) – метод, який формує кластери на основі щільності точок у просторі ознак. Він

здатний виявляти кластери довільної форми та відокремлювати шумові точки, що є корисним для виявлення аномалій, нетипових патернів поведінки користувачів або шахрайського трафіку.

Алгоритм зображено на рисунку 1.2.

Рисунок 1.2 – Алгоритм DBSCAN

1.2.2 Зниження розмірності

Високорозмірність даних ускладнює візуалізацію та інтерпретацію результатів аналізу. Методи зниження розмірності дозволяють відобразити дані у просторі меншої розмірності, зберігаючи основну інформацію та структуру даних.

Метод головних компонент (PCA) – класичний лінійний метод, що перетворює початкові ознаки у набір некорельованих компонент, упорядкованих за величиною поясненої дисперсії. Це дозволяє виділити основні фактори, які

найбільше впливають на поведінку користувачів, і знизити розмірність даних без втрати значущої інформації. Метод зображено на рисунку 1.3.

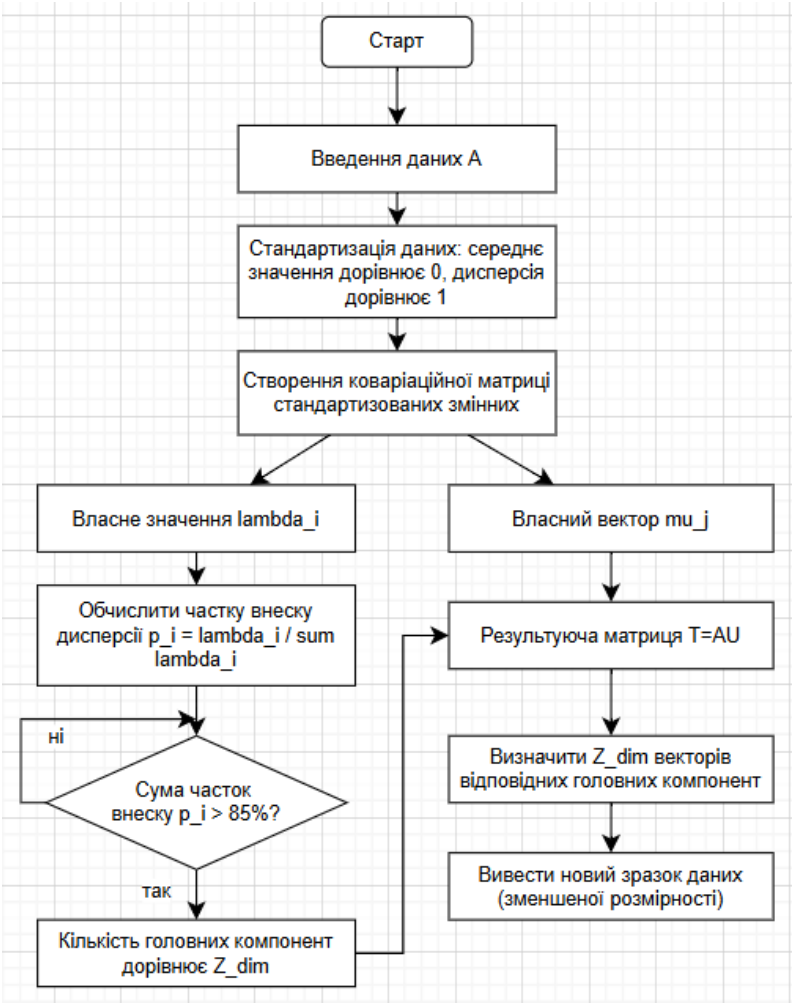

Рисунок 1.3 – Метод PCA

t-SNE (t-Distributed Stochastic Neighbor Embedding) – нелінійний метод, який зберігає локальні відстані між об’єктами та дозволяє виявляти складні, нелінійні структури в даних. Його часто застосовують для візуалізації високорозмірних даних у двовимірному або тривимірному просторі.

UMAP (Uniform Manifold Approximation and Projection) – сучасний метод нелінійного зниження розмірності, що забезпечує швидку обробку великих даних та зберігає як локальні, так і глобальні структури.

Метод особливо корисний для сегментації аудиторії та візуалізації взаємозв’язків між користувачами.

1.2.3 Класифікація та прогнозування

Задачі класифікації та прогнозування відіграють ключову роль у системах автоматизованого управління рекламними кампаніями. Типові завдання включають прогнозування конверсії користувачів на основі їх характеристик, класифікацію джерел трафіку за якістю, виявлення ботів та шахрайського трафіку.

Логістична регресія є базовим методом для задач бінарної класифікації, що оцінює ймовірність належності об'єкта до певного класу. Незважаючи на лінійну природу моделі, логістична регресія забезпечує інтерпретовані результати та ефективно працює при наявності чіткої лінійної залежності між предикторами та цільовою змінною.

Дерева рішень та ансамблеві методи (Random Forest, Gradient Boosting) здатні моделювати складні нелінійні залежності та взаємодії між ознаками. Методи Random Forest та Gradient Boosting будують ансамблі з багатьох дерев рішень, що дозволяє підвищити точність прогнозування та знизити переобучення порівняно з окремим деревом.

Методи опорних векторів (Support Vector Machines, SVM) ефективні у високорозмірних просторах та при наявності чіткої межі між класами. Застосування різних ядрових функцій (лінійне, поліноміальне, радіальне) дозволяє моделювати нелінійні межі рішення без явного перетворення ознак.

1.3 Аналіз літературних джерел

У роботі [4] пропонується стратегія оптимізації розподілу рекламного бюджету на основі навчання з підкріпленням (RL). Мета – динамічне коригування ставок і розподілу бюджету між рекламними каналами для максимізації прибутку, наприклад, конверсій. Сильною стороною є практична спрямованість, використання передових методів ШІ для вирішення динамічної

задачі і деталізація моделі винагороди. Слабкі сторони включають обмежений обсяг (конференційний формат) та можливу залежність від симульованих, а не реальних експериментальних даних. Джерело корисне, оскільки демонструє використання ШІ для оптимізації ефективності (продуктивності), що є аналогією для розроблення Reward function при оцінці продуктивності інтерфейсу.

У роботі [5] досліджуються неефективності на ринках цифрової реклами, спричинені структурними особливостями (аукціони, непрозорість), що призводять до неоптимального розподілу ресурсів. Сильною стороною є економічний аналіз ринкових механізмів, що забезпечує фундаментальне розуміння обмежень продуктивності у цифровому середовищі. Слабкі сторони полягають у теоретичному фокусі на макроекономіці, а не на мікрорівні UX/UI. Джерело стверджує, що продуктивність слід розуміти комплексно як ефективність системи для доведення, що неоптимальний дизайн UI є чинником ринкової неефективності, яку можна усунути автоматизованою оцінкою.

У роботі [6] проведено емпіричне порівняння поширених методів зниження розмірності (t-SNE, UMAP, PCA та інших) у контексті візуального аналізу кластерів. Дослідження охоплює реальні й синтетичні набори даних і оцінює якість проєкцій за метриками збереження локальних/глобальних структур та зручністю інтерпретації для аналітиків. Сильними сторонами є системність експериментів і орієнтація на практичне використання в задачах класифікації та візуалізації. Обмеження стосуються фокусування лише на класичних алгоритмах і недостатньої уваги до новітніх нейромережевих методів проєкції. Джерело є корисним для розуміння того, як обирати метод зниження розмірності під конкретні аналітичні завдання та як ці методи впливають на точність подальшого аналізу даних.

У роботі [7] запропоновано підхід до спільної характеристики багатомасштабної структури у високовимірних даних, що ґрунтується на інформаційних мірах та аналізі локальної/глобальної узгодженості. Автори демонструють, як багаторівневе представлення даних дає змогу точніше виявляти приховані залежності та форми кластеризації, особливо у складних

нерівномірних просторах. Перевагою дослідження є акцент на універсальності методу та його придатності до різних типів даних без жорстких припущень про їхню структуру. Публікація корисна як теоретична база для підходів, що потребують багатомасштабного аналізу, зокрема у задачах моніторингу ефективності рекламних кампаній, де поведінкові дані мають складну й неоднорідну структуру.

У статті [8] розглянуто підхід до підвищення прибутковості рекламних кампаній шляхом аналізу traffic-fingerprints – унікальних шаблонів поведінки користувачів у потоці вебтрафіку. Автори демонструють, що поєднання поведінкових ознак із методами класифікації дає змогу точніше визначати перспективні сегменти аудиторії та усувати неякісний трафік. Сильним боком роботи є поєднання машинного навчання з прикладними метриками рекламної ефективності та тестування на реальних даних рекламних мереж. Джерело є корисним для теми багатовимірного аналізу ефективності, оскільки демонструє, як структуровані поведінкові вектори можуть покращувати рішення щодо таргетингу та оптимізації рекламних кампаній.

У роботі [9] досліджено підвищення залученості користувачів у соціальних мережах за допомогою unsupervised learning. Автори показують, що без учителя можна виявляти природні групи поведінкових патернів, які впливають на реакції аудиторії: частоту взаємодій, стиль публікацій, часові цикли активності. Модель кластеризації дозволила формувати персоналізовані стратегії подачі контенту без необхідності ручної розмітки даних. Перевагою дослідження є сфокусованість на поведінкових багатовимірних характеристиках та їх ролі у прогнозуванні користувацької активності. Робота обмежується соціальними мережами та не розглядає економічні метрики, важливі для рекламних систем. Вона є цінною для теми аналізу ефективності рекламних кампаній, адже демонструє, як некеровані методи виявляють приховані структури у великих багатовимірних даних і сприяти кращому розумінню поведінкових сегментів.

У статті [10] розглянуто оптимізацію рекламних кампаній через кластеризацію аудиторії в середовищі performance-маркетингу. Авторка показує,

як поділ користувачів на поведінкові та демографічні сегменти дозволяє підвищувати точність таргетингу, скорочувати витрати та збільшувати конверсії. Використано підхід, де багатовимірні параметри взаємодій (частота кліків, час активності, історія покупок, контекст переглядів) інтегруються у модель, здатну виявляти приховані групи з різною реактивністю на рекламні стимули. Авторка демонструє, як результати кластеризації перетворюються на реальні маркетингові дії та оптимізовані бюджети.

У роботі [11] запропоновано time-to-event підхід до багатоканальної атрибуції, що дозволяє точніше оцінювати внесок кожного рекламного дотику у фінальну конверсію. Автори моделюють взаємодії користувача як послідовність подій у часі та застосовують методи виживального аналізу для визначення ймовірності того, що конкретний контакт спричинив цільову дію. Такий підхід враховує часові затримки, інтенсивність взаємодій та залежності між каналами.

Серед сильних сторін – відхід від спрощених атрибуційних схем (last click, U-shape) і використання статистично обґрунтованої моделі, здатної відображати реальну динаміку поведінки користувачів. Для теми оцінювання ефективності та моніторингу рекламних кампаній робота є важливою тим, що пропонує формалізований спосіб розподілу коефіцієнтів впливу між каналами, що безпосередньо покращує якість автоматизованого управління бюджетами та оптимізацію стратегій показів.

У статті [12] розглядається використання алгоритму k-means для кластеризації користувачів Facebook Ads з метою підвищення результативності цифрового маркетингу. Автори формують групи аудиторій за поведінковими та демографічними характеристиками, а потім аналізують відмінності між сегментами щодо реагування на рекламу. Такий підхід дозволяє виявити найбільш цінні кластери та адаптувати рекламні повідомлення під специфічні групи користувачів. Сильним боком роботи є простота та практична спрямованість: k-means дає швидкі результати та може бути інтегрований у реальні рекламні системи. Для теми багатовимірного аналізу ефективності рекламних кампаній джерело важливе тим, що показує, як сегментація аудиторії

на основі багатовимірних даних може слугувати фундаментом для підвищення CTR, конверсій та оптимізації витрат у автоматизованих системах керування маркетинговими активностями.

У роботі [13] запропоновано модель CAMTA (Causal Attention Model) для багатоканальної атрибуції у цифровому маркетингу. Модель дозволяє визначати вплив кожного контакту користувача з рекламою на фінальну конверсію, враховуючи часові залежності та причинно-наслідкові зв’язки між подіями. Автори демонструють, що CAMTA забезпечує точніші оцінки ролі окремих каналів порівняно зі стандартними підходами на основі правил або простих вагових моделей. Джерело цінне для дослідження багатовимірного аналізу ефективності рекламних кампаній, оскільки демонструє застосування складних моделей для точного визначення внеску кожного рекламного дотику в конверсію, що може бути інтегровано в системи автоматизованого управління рекламою.

1.4 Постановка задачі дослідження

На основі аналізу сучасних методів багатовимірного аналізу та систем моніторингу рекламних кампаній можна сформулювати основні завдання дослідження. Традиційні підходи до аналізу ефективності рекламних кампаній базуються на простих метриках (CTR, CPC, CPA, CR) та їх порівнянні між різними сегментами аудиторії або варіантами креативів. Такий підхід не враховує складні взаємозв’язки між параметрами кампанії, їх спільний вплив на кінцевий результат та можливі синергетичні ефекти від комбінації різних факторів. Системи трекінгу, зокрема Keitaro, збирають детальну багатовимірну інформацію про кожну взаємодію користувача з рекламою, проте наявні інструменти аналітики обмежуються стандартними звітами та групуванням за окремими параметрами. Потенціал методів багатовимірного аналізу для виявлення прихованих закономірностей та оптимізації управління кампаніями

залишається нереалізованим. Особливістю арбітражу трафіку є необхідність швидкого прийняття рішень в умовах обмеженого бюджету та високої конкуренції. Арбітражник повинен оперативно виявляти найефективніші комбінації параметрів таргетингу, креативів та джерел трафіку, перерозподіляти бюджет між ними та відключати неефективні напрями. Автоматизація цього процесу на основі методів інтелектуального аналізу даних може суттєво підвищити ефективність управління кампаніями.

Об’єктом дослідження є процеси управління та моніторингу рекламних кампаній у соціальній мережі Facebook з використанням трекінгової системи Keitaro. Метою дослідження є підвищення ефективності управління рекламними кампаніями шляхом розроблення та впровадження методів багатовимірного аналізу та моніторингу, що дозволяють виявляти приховані закономірності у даних та оптимізувати стратегію розміщення реклами.

Для досягнення мети необхідно вирішити такі завдання:

– провести аналіз структури та особливостей даних, що генеруються трекінговою системою Keitaro при моніторингу рекламних кампаній у Facebook;

– розробити методику попередньої обробки та підготовки даних для багатовимірного аналізу з урахуванням специфіки предметної області;

– застосувати методи кластеризації для сегментації аудиторії та виявлення груп користувачів зі схожими характеристиками;

– дослідити можливості методів зниження розмірності для візуалізації та інтерпретації високорозмірних даних рекламних кампаній;

– розробити систему інтегральних метрик якості трафіку на основі багатовимірного аналізу взаємодії користувачів;

– реалізувати програмний застосунок для автоматизованого збору, обробки та візуалізації даних з можливістю застосування розроблених методів аналізу;

– провести експериментальне дослідження на реальних даних рекламних кампаній та оцінити ефективність розроблених методів.

2 МАТЕМАТИЧНІ МОДЕЛІ БАГАТОВИМІРНОГО АНАЛІЗУ РЕКЛАМНИХ КАМПАНІЙ

2.1 Аналіз структури даних трекінгової системи

Дані, що збираються трекінговою системою Keitaro під час моніторингу рекламних кампаній у Facebook, мають складну багатовимірну структуру. Кожна подія взаємодії користувача з рекламою фіксується у вигляді запису, що містить множину атрибутів різної природи: числових, категоріальних, часових та геопросторових [14].

Базова структура даних може бути представлена у вигляді матриці спостережень за формулою

$$X = \{x_{ij}\},\ i = 1, \dots, n,\ j = 1, \dots, m,$$

де n – кількість спостережень (кліків, конверсій);

m – кількість ознак;

x_{ij} – значення j-ї ознаки для і-го спостереження.

Аналіз наявного CSV-файлу з даними трафіку показує наступну структуру атрибутів:

– часові характеристики, а саме дата та час кліка (ts), що дозволяють аналізувати динаміку кампанії та виявляти часові патерни;

– географічні параметри, а саме країна ($country$), регіон ($region$), місто ($city$), що використовуються для географічної сегментації аудиторії;

– технічні характеристики, а саме операційна система (os), версія ОС ($os\_version$), браузер ($browser$), тип пристрою ($device\_type$), модель пристрою ($device\_model$), що характеризують технологічні переваги аудиторії;

– мережеві параметри, а саме IP-адреса (ip), інтернет-провайдер (isp), тип з’єднання ($connection\_type$);

– параметри кампанії, а саме ідентифікатори кампанії ($campaign\_id$), групи оголошень ($adset$), креативу ($creative$), субаккаунта (sub_id), що дозволяють відстежувати ефективність окремих елементів;

– результативні метрики, а саме статус конверсії ($is\_conversion$), дохід ($payout$), витрати ($cost$).

Загальна кількість ознак у наборі даних становить близько 30-40 параметрів, що створює високорозмірний простір для аналізу. При цьому різні типи ознак вимагають різних підходів до обробки та аналізу [15].

Числові ознаки (витрати, дохід, час) можуть приймати значення з неперервного або дискретного діапазону [16]. Для таких ознак застосовуються стандартні статистичні методи обробки, включаючи нормалізацію та стандартизацію за формулою

$$Xx'_{ij} = \frac{(x_{ij} - \mu_j)}{\sigma_j},$$

де μ_j – середнє значення j-ї ознаки;

σ_j – стандартне відхилення j-ї ознаки.

Категоріальні ознаки (країна, операційна система, браузер) приймають значення з обмеженої множини категорій. Для використання таких ознак у математичних моделях застосовується метод one-hot encoding, що перетворює категоріальну змінну з k категоріями у k бінарних змінних за формулою

$$x_j^{(cat)} \longrightarrow \{x_{j1}, x_{j2}, \dots, x_{jk}\}, \quad x_{ji} \in \{0, 1\}, \quad i = 1, \dots, k,$$

де $x_{j1} = 1$, якщо спостереження належить до l-ї категорії, інакше $x_{j1} = 0$.

Часові ознаки вимагають спеціальної обробки для виділення корисних характеристик. З мітки часу можна витягти множину похідних ознак:

– час доби (година), а саме $h = hour(ts)$, що дозволяє виявити денні патерни активності;

– день тижня, тобто $d = dayofweek(ts)$, для аналізу тижневої циклічності;

– частина доби, а саме $period \in \{\text{ніч}, \text{ранок}, \text{день}, \text{вечір}\}$;

– робочий/вихідний день, а саме $is\_weekend \in \{0,1\}$.

Географічні координати можуть бути представлені як у вигляді категоріальних змінних (назви країн, міст), так і у вигляді числових координат широти та довготи для більш детального просторового аналізу [17].

Важливою характеристикою даних є наявність пропущених значень. У реальних даних деякі поля можуть бути не заповнені через обмеження трекінгу або особливості джерела трафіку. Для обробки пропущених значень застосовуються методи імпутації:

– заповнення середнім/медіанним значенням для числових ознак;

– заповнення модою (найчастішим значенням) для категоріальних ознак;

– видалення спостережень з великою кількістю пропущених значень;

– використання спеціального значення «невідомо» як окремої категорії.

Розподіл значень ознак часто є неоднорідним та може містити викиди. Аналіз розподілів дозволяє виявити аномалії та підібрати відповідні методи нормалізації. Для числових ознак доцільно перевірити гіпотезу про нормальність розподілу за допомогою тестів Колмогорова-Смірнова або Шапіро-Уїлка.

У випадку сильно асиметричних розподілів (наприклад, для витрат або доходу) може застосовуватися логарифмічне перетворення за формулою

$$x'_{ij} = log(x_{ij} + c),$$

де c – мала константа для уникнення логарифма нуля [18].

Кореляційний аналіз дозволяє виявити лінійні залежності між ознаками. Матриця парних кореляцій Пірсона визначається як

$$R = \{r_{ik}\}, \quad r_{jk} = \frac{cov(x_i, x_k)}{\sigma_i, \sigma_k},$$

де $cov(x_i, x_k)$ – коваріація між j-ю та k-ю ознаками.

Високі значення кореляції між ознаками можуть свідчити про мультиколінеарність, що негативно впливає на стабільність деяких статистичних моделей. У таких випадках доцільно видалити одну з корельованих ознак або застосувати методи зниження розмірності [19]. Аналіз розподілу цільової змінної (конверсія) показує типову для рекламних кампаній нерівномірність: кількість конверсій зазвичай становить 1-5% від загальної кількості кліків. Це створює проблему незбалансованих класів при побудові моделей класифікації, що вимагає застосування спеціальних методів обробки:

– зважування класів при навчанні моделі;

– передискретизація (oversampling) меншого класу або недодискретизація (undersampling) більшого класу;

– синтез штучних прикладів меншого класу методом SMOTE.

Принцип збору даних з Keitaro зображено на рисунку 2.1.

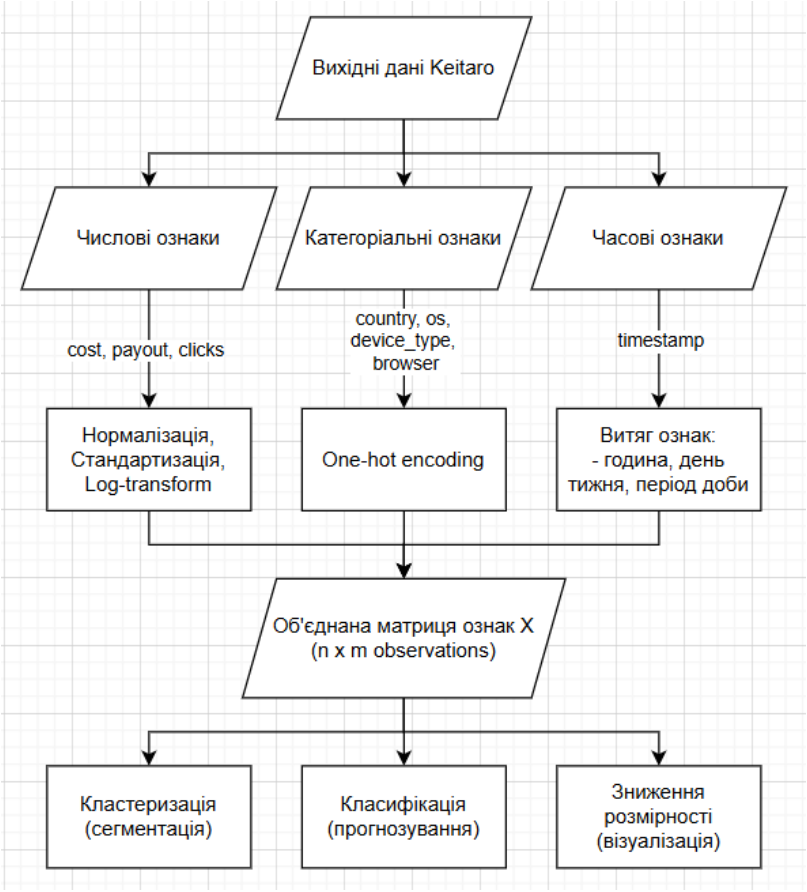

Рисунок 2.1 – Принцп збору даних з Keitaro

2.2 Методи кластеризації для сегментації аудиторії

Кластеризація є фундаментальним методом виявлення природних групувань у даних без попереднього визначення класів. У контексті аналізу рекламних кампаній кластеризація дозволяє сегментувати аудиторію за множиною характеристик одночасно, виявляючи групи користувачів зі схожими профілями.

Метод k-середніх є поширеним алгоритмом кластеризації завдяки простоті реалізації і обчислювальній ефективності [20]. Алгоритм мінімізує суму квадратів відстаней від об'єктів до центрів їх кластерів за формулою

$$J = \sum_{i=1}^{k} (x \in C_i) \left|| x - \mu_i \right||^2,$$

де C_i – i-й кластер;

μ_i – центр i-го кластера (центроїд);

k – кількість кластерів.

Алгоритм працює по етапах:

Етап 1. Ініціалізація, тобто випадковий вибір k початкових центроїдів $\mu_1, \mu_2, \ldots, \mu_k$.

Етап 2. Призначення, тобто кожний об'єкт x_i призначається до найближчого центроїда за формулою

$$C_i = argmin(j = 1, \ldots, k) \, ||x_i - \mu_i||^2.$$

Етап 3. Оновлення, тобто перерахунок центроїдів як середніх значень об'єктів у кластерах за формулою

$$\mu_j = \left(\frac{1}{|C_j|}\right) \Sigma(x \in C_j) x.$$

Етап 4. Повторення кроків 2-3 до збіжності (коли призначення об’єктів більше не змінюються).

Для вимірювання відстані між об’єктами використовується евклідова метрика, що має формулу

$$d(x, y) = \sqrt{\Sigma(j = 1\ to\ m)\left(x_j - y_j\right)^2}.$$

Важливою проблемою методу k-середніх є необхідність заздалегідь визначати кількість кластерів k. Для вибору оптимального значення k застосовуються різні критерії:

– метод ліктя, що аналізує залежність функції втрат J від k і має формулу

$$\Delta_k = J_k - J_{k+1}.$$

Оптимальне значення k відповідає «ліктю» на графіку – точці, після якої зменшення J сповільнюється;

– силует-коефіцієнт (silhouette coefficient), що оцінює якість кластеризації для кожного об’єкта, що розраховується як

$$s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)},$$

де a_i – середня відстань від x_i до інших об’єктів його кластера;

b_i – мінімальна середня відстань від x_i до об’єктів інших кластерів.

Значення s_i близьке до 1 свідчить про добре розміщення об’єкта у кластері;

– індекс Девіса-Болдіна (Davies-Bouldin index), що розраховується як

$$DB = \left(\frac{1}{k}\right)\Sigma(i = 1\ to\ k)\max(j \neq i)\left[\frac{(\sigma_i + \sigma_j)}{d(\mu_i, \mu_j)}\right],$$

де σ_i – середня відстань об’єктів кластера C_i до його центроїда. Менше значення DB відповідає кращій кластеризації [21, 22].

Ієрархічна кластеризація будує деревоподібну структуру вкладених кластерів, що дозволяє аналізувати дані на різних рівнях деталізації. Агломеративний алгоритм починає з розгляду кожного об’єкта як окремого кластера та послідовно об’єднує найближчі пари кластерів [23].

На кожному кроці відстань між кластерами C_p та C_q визначається за одним з методів зв’язування:

– одиничне зв’язування (single linkage), що розраховується як

$$d(C_p, C_q) = \min(x \in C_p, y \in C_q)\, d(x, y);$$

– повне зв’язування (complete linkage), що розраховується як

$$d(C_p, C_q) = \max(x \in C_p, y \in C_q)\, d(x, y);$$

– середнє зв’язування (average linkage), що розраховується як

$$d(C_p, C_q) = \left(\frac{1}{|C_p| \cdot |C_q|}\right) \Sigma(x \in C_p) \Sigma(y \in C_q) d(x, y);$$

– метод Варда, що мінімізує приріст дисперсії при об’єднанні, що розраховується як

$$d(C_p, C_q) = \sqrt{\left[\frac{2|C_p||C_q|}{|C_p|+|C_q|}\right] \left|\left|\mu_p - \mu_q\right|\right|},$$

Результат ієрархічної кластеризації представляється у вигляді дендрограми – дерева, яке показує послідовність об’єднання кластерів та відстані, на яких відбувається об’єднання [24].

Метод DBSCAN виявляє кластери довільної форми на основі щільності точок у просторі ознак. Алгоритм використовує два параметри: ε – радіус околу та $MinPts$ – мінімальна кількість точок у околі [25].

Точка x називається core point (основною точкою), якщо в її ε-околі знаходиться принаймні $MinPts$ точок, і розраховується як

$$\left|N_{\varepsilon(x)}\right| \geq MinPts, \text{де } N_{\varepsilon(x)} = \{y : d(x,y) \leq \varepsilon\}.$$

Дві основні точки x та y є з'єднаними за щільністю, якщо існує ланцюг основних точок, що їх з'єднує.

Кластер визначається як максимальна множина точок, з'єднаних за щільністю.

Точки, що не є основними і не належать жодному кластеру, позначаються як шум (noise або outliers). Ця властивість робить DBSCAN корисним для виявлення аномального трафіку [26].

Алгоритм DBSCAN має такі етапи:

Етап 1. Для кожної необробленої точки x

– знайти всі точки в $N\_\varepsilon(x)$;
– якщо $\left|N_{\varepsilon(x)}\right| < MinPts$, позначити x як шум (тимчасово);
– якщо $\left|N_{\varepsilon(x)}\right| \geq MinPts$, створити новий кластер C та додати x до C.

Етап 2. Для кожної точки $y \in N_{\varepsilon(x)}$

– якщо y не оброблена, додати y до C;
– якщо y є основною точкою, додати всі точки з $N\_\varepsilon(y)$ до C.

Етап 3. Повторювати до обробки всіх точок.

Для застосування методів кластеризації до даних рекламних кампаній необхідно враховувати змішану природу ознак (числові та категоріальні). Для категоріальних ознак евклідова відстань не є адекватною метрикою. Замість неї використовується відстань Гауера (Gower distance):

$$d_{G(x,y)} = \left(\frac{1}{m}\right) \Sigma(j = 1\ to\ m)\delta_{j(x,y)} \cdot\ w_j,$$

де δ_j – функція відстані для j-ї ознаки

$$\delta_{j(x,y)} = \begin{cases} \frac{|x_j - y_j|}{R_j}, \text{якщо j – числова ознака} \\ 0,\ \text{якщо}\ x_j = y_j (\text{категоріальна}), \\ 1,\ \text{якщо}\ x_j \neq y_j (\text{категоріальна}) \end{cases}$$

де R_j – діапазон значень j-ї числової ознаки;

w_j – вага j-ї ознаки (зазвичай $w_j = 1$).

Інтерпретація результатів кластеризації включає аналіз характеристик виявлених сегментів.

Для кожного кластера C_i розраховуються такі показники:

– розмір кластера: $n_i = |C_i|$;

– центроїд кластера μ_i, що представляє типовий профіль сегмента;

– статистичні характеристики ознак: середнє, медіана, стандартне відхилення;

– метрики ефективності: середній CR, CPA, ROI у кластері;

– частка конверсій: $conv\_rate_i = \frac{(\text{кількість конверсій у } C_i)}{n_i}$.

Порівняння метрик між кластерами дозволяє виявити найбільш та найменш ефективні сегменти аудиторії.

Кластери з високим та низьким CPA є пріоритетними для збільшення бюджету, тоді як неефективні кластери можуть бути виключені з таргетингу.

Узагальнений алгоритм застосування кластеризації для сегментації аудиторії рекламних кампаній представлено на рисунку 2.2.

Рисунок 2.2 – Алгоритм кластеризації для сегментації аудиторії

2.3 Методи зниження розмірності та візуалізації даних

Високорозмірність даних створює проблему для візуалізації та інтерпретації результатів аналізу. Людське сприйняття обмежене двома-трьома вимірами, тому для візуалізації багатовимірних даних необхідно проєктувати їх у простір меншої розмірності зі збереженням найсуттєвіших характеристик.

Метод головних компонент (PCA) є класичним підходом до зниження розмірності, що базується на лінійному перетворенні початкових ознак. Головні компоненти є власними векторами коваріаційної матриці даних, упорядкованими за величиною відповідних власних значень.

Нехай X – центрована матриця даних (з нульовим середнім). Коваріаційна матриця визначається як

$$\Sigma = \left(\frac{1}{n-1}\right) X^T X.$$

Власні вектори $v_1, v_2, \dots, v_m$ та власні значення λ_1 ≥ λ_2 ≥ ... ≥ λ_m коваріаційної матриці Σ задовольняють рівняння

$$\Sigma\, v_j = \lambda_j v_j, j = 1, \dots, m.$$

Власне значення λ_j характеризує дисперсію даних вздовж j-ї головної компоненти. Частка дисперсії, що пояснюється j-ю компонентою

$$\rho_j = \frac{\lambda_j}{\Sigma(k=1\ to\ m)\lambda_k}.$$

Кумулятивна частка дисперсії для перших p компонент:

$$R_p = \Sigma(j = 1\ to\ p)\rho_j.$$

Зазвичай для зниження розмірності обирають таку кількість компонент p, щоб $R_p \geq 0.8$-0.9, тобто перші p компонент пояснювали 80-90% загальної дисперсії. Проєкція даних на простір головних компонент визначається як

$$Z = X\, V_p,$$

де $V_p = [v_1, v_2, \dots, v_p]$ – матриця перших p власних векторів, Z – матриця знижених даних розміру $n \times p$.

Інтерпретація головних компонент здійснюється через аналіз факторних навантажень – коефіцієнтів власних векторів. Великі за абсолютним значенням навантаження вказують на ознаки, що роблять основний внесок у відповідну компоненту. Навантаження a_{ij} показує, наскільки сильно i-я початкова ознака впливає на j-ю головну компоненту

$$a_{ij} = v_{ij}\sqrt{\lambda_j}.$$

Алгоритм PCA має такі етапи:

Етап 1. Центрування даних: $X' = X - mean(X)$.

Етап 2. Обчислення коваріаційної матриці $\Sigma = \left(\frac{1}{n-1}\right) X'^T X'$.

Етап 3. Знаходження власних векторів та власних значень матриці Σ.

Етап 4. Сортування власних векторів за спаданням власних значень.

Етап 5. Вибір кількості компонент p за критерієм кумулятивної дисперсії.

Етап 6. Формування матриці проєкції V_p.

Етап 7. Обчислення проєкції даних $Z = X'V_p$.

Недоліком PCA є припущення про лінійність взаємозв'язків між ознаками. Для виявлення нелінійних структур застосовуються нелінійні методи зниження розмірності.

Метод t-SNE зберігає локальну структуру даних, відображаючи близькі у високорозмірному просторі точки в близькі точки у низькорозмірному просторі.

Алгоритм t-SNE мінімізує розбіжність Кульбака-Лейблера між двома розподілами: p_{ij} – розподілом подібності між точками у високорозмірному просторі та q_{ij} – розподілом у низькорозмірному просторі. У високорозмірному просторі подібність між точками x_i та x_j визначається гаусівською ймовірністю

$$p_{(j|i)} = \frac{\exp\left(-\frac{\left||x_i - x_j|\right|^2}{2\sigma_i^2}\right)}{\Sigma_{(k\neq i)} \exp\left(-\frac{\left||x_i - x_k|\right|^2}{2\sigma_i^2}\right)}.$$

Симетрична подібність розраховується за формулою

$$p_{ij} = \frac{p_{(j|i)} + p_{(i|j)}}{2n}.$$

У низькорозмірному просторі (зазвичай 2D або 3D) використовується t-розподіл Стьюдента з одним ступенем свободи

$$q_{ij} = \frac{\left(1 + \left||y_i - y_j\right|\right|^2\right)^{-1}}{\Sigma_{(k \neq l)}\left(1 + \left||y_k - y_l\right|\right|^2\right)^{-1}},$$

де y_i – низькорозмірне представлення точки x_i.

Функція втрат (розбіжність Кульбака-Лейблера) розраховується як

$$C = KL(P||Q) = \Sigma_i \Sigma_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right).$$

Мінімізація C здійснюється методом градієнтного спуску. Градієнт функції втрат за координатами y_i розраховується як

$$\frac{\partial C}{\partial y_i} = 4\,\Sigma_j (p_{ij} - q_{ij})(y_i - y_j)\left(1 + \left||y_i - y_j\right|\right|^2\right)^{-1}.$$

Алгоритм t-SNE має такі етапи:

Етап 1. Обчислення подібностей p_{ij} у високорозмірному просторі.

Етап 2. Випадкова ініціалізація точок y_i у низькорозмірному просторі.

Етап 3. Ітеративна оптимізація:

– обчислення q_{ij} для поточних позицій y_i;

– обчислення градієнта $\frac{\partial C}{\partial y_i}$;

– оновлення позицій методом градієнтного спуску з імпульсом

$$Y^{t+1} = Y^t + \frac{\eta(t)\partial C}{\partial Y} + \alpha(t)(Y^t - Y^{t-1}),$$

де $\eta(t)$ – швидкість навчання, $\alpha(t)$ – коефіцієнт імпульсу.

Етап 4. Повторення кроку 3 до збіжності.

Параметр perplexity контролює баланс між локальною та глобальною структурами даних і пов’язаний з ефективною кількістю найближчих сусідів

$$Perp(P_i) = 2^{H(P_i)},$$

де $H(P_i) = -\Sigma_j p^2_{(j|i)log}\left(p_{(j|i)}\right)$ентропія розподілу P_i.

Типові значення perplexity знаходяться в діапазоні 5-50.

Метод UMAP є альтернативою t-SNE, що базується на теорії ріманових многовидів. UMAP зберігає як локальну, так і глобальну структуру даних краще, ніж t-SNE, та працює швидше на великих наборах даних.

UMAP будує зважений граф k-найближчих сусідів у високорозмірному просторі. Ваги ребер визначаються як

$$w_{ij} = \exp\left(-\frac{(d(x_i,x_j)-\rho_i)}{\sigma_i}\right),$$

де ρ_i – відстань до найближчого сусіда точки x_i, σ_i – параметр нормалізації.

Симетрична вага розраховується як

$$w_{ij}^{sym} = w_{ij} + w_{ji} - w_{ij} \cdot w_{ji}.$$

У низькорозмірному просторі будується аналогічний граф з вагами

$$v_{ij} = \left(1 + a\left||y_i - y_j\right|\right|^{2b}\right)^{-1},$$

де a та b – параметри, що підбираються під час навчання.

Функція втрат UMAP (крос-ентропія) розраховується як

$$C = \Sigma_i \Sigma_j \left[w_{ij}^{sym} \log\left(\frac{v_{ij}}{w_{ij}^{sym}}\right) + \left(1 - w_{ij}^{sym}\right) \log\left(\frac{(1-v_{ij})}{1-w_{ij}^{sym}}\right) \right].$$

Оптимізація виконується стохастичним градієнтним спуском.

Порівняння методів зниження розмірності:

– PCA, що є швидким, лінійним, зберігає глобальну структуру, інтерпретовані компоненти;

– t-SNE зберігає локальну структуру, виявляє кластери, повільний на великих даних;

– UMAP, що є швидким, зберігає локальну та глобальну структури, стабільніший за t-SNE.

Для даних рекламних кампаній рекомендується комбінований підхід: спочатку застосувати PCA для зниження розмірності до 50-100 компонент (прискорення подальших обчислень), потім застосувати t-SNE або UMAP для візуалізації у 2D. Процес вибору та застосування методів зниження розмірності для візуалізації даних рекламних кампаній представлено на рисунку 2.3.

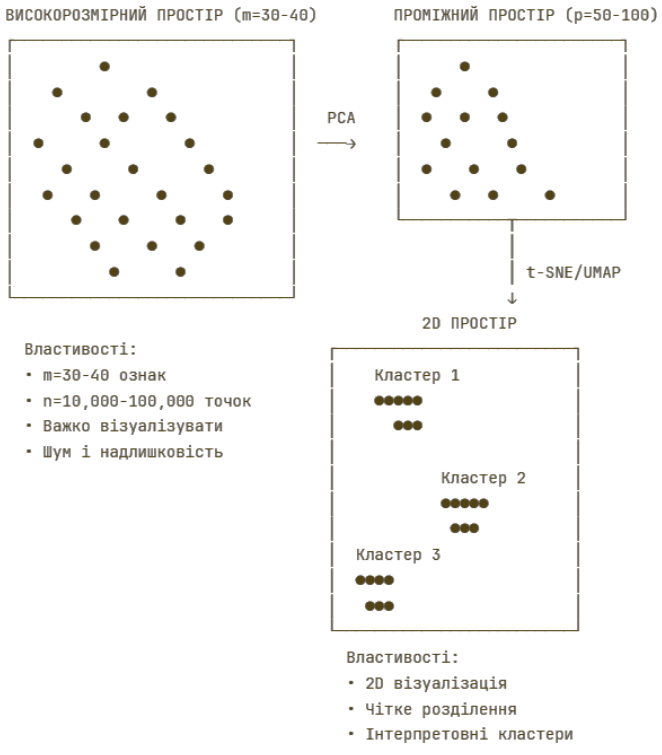

Рисунок 2.3 – Схема процесу зниження розмірності та візуалізації даних

2.4 Моделі класифікації та прогнозування конверсій

2.4.1 Логістична регресія

Логістична регресія є базовим методом для задач бінарної класифікації, зокрема для прогнозування конверсії користувачів. Модель оцінює ймовірність належності об’єкта до класу 1 (конверсія) на основі лінійної комбінації предикторів [27].

Лінійна комбінація предикторів розраховується як

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_m x_m = \beta^T x,$$

де $\beta = [\beta_0, \beta_1, \ldots, \beta_m]^T$ – вектор коефіцієнтів;

$x = [1, x_1, \ldots, x_m]^T$ – вектор ознак.

Логістична функція (сигмоїда) перетворює z у ймовірність

$$P(y = 1|x) = \sigma(z) = \frac{1}{1 + \exp(-z)}.$$

Властивості логістичної функції:

- $\sigma(z) \in (0, 1)$ для всіх z;
- $\sigma(0) = 0.5$;
- $\sigma(-z) = 1 - \sigma(z)$;
- похідна: $\sigma'(z) = \sigma(z)\big(1 - \sigma(z)\big)$.

Логіт визначається як логарифм відношення шансів

$$logit(p) = \log\left(\frac{p}{1-p}\right) = z = \beta^T x.$$

Модель передбачає, що логіт є лінійною функцією предикторів [28].

Оцінка коефіцієнтів β здійснюється методом максимальної правдоподібності. Функція правдоподібності для вибірки $\{(x_i, y_i)\}, i = 1, \ldots, n$:

$$L(\beta) = \prod_{i=1}^{n} P(y_i|x_i) = \prod_{i=1}^{n} \left[p_i^{y^i} \cdot (1 - \mathrm{p_i})^{1-y_i}\right],$$

де $p_i = P(y = 1|x_i) = \sigma(\beta^T x_i)$.

Логарифм функції правдоподібності розраховується як

$$\ell(\beta) = \log L(\beta) = \Sigma(i = 1\ to\ n)[y_i \log(p_i) + (1 - y_i) \log(1 - p_i)].$$

Функція втрат (cross-entropy loss) розраховується як

$$J(\beta) = -\ell(\beta) = -\Sigma(i = 1\ to\ n)[y_i \log(p_i) + (1 - y_i) \log(1 - p_i)].$$

Мінімізація $J(\beta)$ здійснюється методом градієнтного спуску або методом Ньютона-Рафсона. Градієнт функції втрат розраховується як

$$\frac{\partial J}{\partial \beta} = X^T(p - y),$$

де $p = [p_1, p_2, \dots, p_n]^T$ – вектор прогнозованих ймовірностей;

$y = [y_1, y_2, \dots, y_n]^T$ – вектор справжніх міток.

Гессіан (матриця других похідних) розраховується як

$$H = \frac{\partial^2 J}{\partial \beta \partial \beta^T} = X^T W X,$$

де W – діагональна матриця з елементами $w_{ii} = p_{i(1-p_i)}$.

Оновлення коефіцієнтів методом Ньютона-Рафсона розраховується як

$$\beta^{t+1} = \beta^t - \frac{H^{-1}\partial J}{\partial \beta}.$$

Для регуляризації моделі та запобігання переобученню застосовується $L2$-регуляризація (Ridge), що розраховується як

$$J_{reg(\beta)} = J(\beta) + \lambda \Sigma(j = 1\ to\ m)\beta_j^2,$$

або $L1$-регуляризація (Lasso), що розраховується як

$$J_{reg(\beta)} = J(\beta) + \lambda \Sigma(j = 1\ to\ m)\left|\beta_j\right|,$$

де λ – параметр регуляризації, що контролює складність моделі [29].

Інтерпретація коефіцієнтів: коефіцієнт β_j показує, як змінюється логіт при збільшенні x_j на одиницю:

$$\Delta\ logit = \beta_j \Delta x_j.$$

Відношення шансів (odds ratio) при зміні x_j на одиницю звучить як

$$OR_j = \exp(\beta_j).$$

Якщо $\beta_j > 0$, збільшення x_j підвищує ймовірність конверсії, а якщо $\beta_j < 0$ – знижує. Для багатокласової класифікації застосовується узагальнення логістичної регресії – softmax регресія (multinomial logistic regression) [30]. Ймовірність належності до класу k і розраховується як

$$P(y = k|x) = \frac{\exp(\beta_k^T x)}{\sum_{j=1}^{K} \exp\left(\beta_j^T x\right)},$$

де K – кількість класів;

β_k – вектор коефіцієнтів для класу k.

2.4.2 Ансамблеві методи: Random Forest та Gradient Boosting

Ансамблеві методи об’єднують прогнози множини базових моделей для підвищення точності та зниження переобучення. Дерева рішень є популярними базовими моделями завдяки здатності моделювати нелінійні залежності та взаємодії між ознаками [31, 32].

Random Forest будує ансамбль незалежних дерев рішень на різних підвибірках даних та усереднює їх прогнози. Алгоритм використовує два механізми рандомізації:

– bagging (bootstrap aggregating) – кожне дерево навчається на випадковій підвибірці з поверненням розміру n;

– випадковий вибір підмножини ознак при розщепленні вузла дерева.

Для класифікації прогноз Random Forest визначається голосуванням

$$\hat{y} = mode\{h_{1(x)}, h_{2(x)}, \dots, h_{T(x)}\},$$

де $h_{t(x)}$ – прогноз t-го дерева, T – кількість дерев в ансамблі.

Для оцінки ймовірності конверсії використовується формула

$$\hat{P}(y = 1|x) = \frac{1}{T} \sum_{t=1}^{T} I\{h_{t(x)} = 1\},$$

де $I\{\cdot\}$ – індикаторна функція.

Алгоритм побудови одного дерева в Random Forest має такі етапи:

Етап 1. Випадкова вибірка з поверненням n спостережень з навчальної вибірки (bootstrap sample).

Етап 2. Побудова дерева рішень:

– у кожному вузлі випадковий вибір підмножини з k ознак (зазвичай $k \approx \sqrt{m}$ для класифікації);

– вибір найкращого розщеплення серед обраних k ознак за критерієм Джині або ентропії;

– рекурсивне розщеплення до досягнення критерію зупинки.

Крок 3. Повторення кроків 1-2 для T дерев.

Критерій Джині для оцінки чистоти вузла визначається як

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

де $p_{k(t)}$ – частка об'єктів класу k у вузлі t.

Приріст інформації при розщепленні вузла t на вузли t_L та t_R розраховується як

$$\Delta Gini(t) = Gini(t) - \left[\frac{|t_L|}{|t|} \cdot Gini(t_L) + \frac{|t_R|}{|t|} \cdot Gini(t_R)\right],$$

де $|t|$ – кількість об'єктів у вузлі t.

Важливість ознаки x_j у Random Forest визначається як середнє зменшення критерію Джині по всіх деревах та вузлах, де використовується ознака

$$Importance(x_j) = \frac{1}{T}\sum_{t=1}^{T} \Sigma(nodes\ using\ x_j)\Delta Gini.$$

Gradient Boosting будує ансамбль дерев послідовно, де кожне наступне дерево навчається виправляти помилки попередніх [33]. Метод базується на градієнтному спуску у функціональному просторі.

Модель на кроці m розраховується як

$$F_{m(x)} = F_{(m-1)(x)} + \nu\, h_{m(x)},$$

де $F_{(m-1)(x)}$ – поточна модель;

$h_{m(x)}$ – нове дерево;

$\nu \in (0,1]$ – швидкість навчання (learning rate) [34].

Нове дерево $h_{m(x)}$ апроксимує антиградієнт функції втрат визначається як

$$h_{m(x)} \approx -\frac{\partial L(y,F_{(m-1)(x)})}{\partial F_{(m-1)(x)}}.$$

Для логістичної функції втрат псевдо-залишки (pseudo-residuals) розраховується як

$$r_{im} = y_i - p_{im}, \text{де } p_{im} = \sigma(F_{(m-1)(x_i)}).$$

Алгоритм Gradient Boosting для класифікації має декілька етапів:

Етап 1. Ініціалізація моделі константою, що розраховується як

$$F_{0(x)} = \log\left(\frac{p_1}{1-p_1}\right),$$

де $p_1 = \frac{(\Sigma y_i)}{n}$.

Етап 2. Для $m = 1$ до M виконуються такі операції:

– обчислення псевдо-залишків для всіх спостережень

$$r_{im} = y_i - \sigma(F_{(m-1)(x_i)}), i = 1, \dots, n;$$

– навчання регресійного дерева $h_{m(x)}$ на парах $\{(x_i, r_{im})\}$;

– обчислення оптимальних значень у листках дерева

$$\gamma_{jm} = \frac{\Sigma(x_i \in R_{jm}) r_{im}}{(x_i \in R_{jm})|r_{im}|(1-|r_{im}|)} \Sigma,$$

де R_{jm} – j-й листок дерева h_m;

– оновлення моделі, що обчислюється як

$$F_{m(x)} = F_{(m-1)(x)} + \nu \Sigma_j \gamma_{jm} I\{x \in R_{jm}\}.$$

Етап 3. Фінальна модель розраховується як

$$F_{M(x)} = F_{0(x)} + \nu \sum_{m=1}^{m} \Sigma_j \gamma_{jm} I\{x \in R_{jm}\}.$$

Прогнозована ймовірність конверсії розраховується як

$$\hat{P}(y = 1|x) = \sigma(F_{M(x)}).$$

Параметри, що впливають на якість Gradient Boosting:

– кількість ітерацій M (кількість дерев);

– швидкість навчання ν: менші значення вимагають більше ітерацій, але покращують узагальнення;

– максимальна глибина дерев: контролює складність базових моделей;

– мінімальна кількість спостережень у листку: запобігає переобученню.

Для запобігання переобученню застосовуються техніки регуляризації:

– рання зупинка (early stopping): моніторинг помилки на валідаційній вибірці та зупинка при її зростанні;

– shrinkage: зменшення швидкості навчання ν;

– підвибірка (subsampling): навчання кожного дерева на випадковій частині даних;

– обмеження складності дерев (глибина, кількість листків).

XGBoost (Extreme Gradient Boosting) є оптимізованою реалізацією Gradient Boosting з додатковими покращеннями:

– регуляризація функції втрат, що розраховується як

$$Obj^m = \sum_{i=1}^{n} w_j^2 \, L\left(y_i, F_{m(x_i)}\right) + \Omega(h_m),$$

де $\Omega(h_m) = \gamma T + \left(\frac{\lambda}{2}\right)\sum_{j=1}^{T} w_j^2$ – штраф за складність дерева;

T – кількість листків;

w_j – значення у листку;

– апроксимація функції втрат другого порядку (розклад Тейлора), що обчислюється як

$$Obj^m \approx \sum_{i=1}^{n}\left[g_i h_{m(x_i)} + \left(\frac{1}{2}\right) h_i h_m^2(x_i)\right] + \Omega(h_m),$$

де $g_i = \frac{\partial L}{\partial F_{(m-1)(x_i)}}$;

$h_i = \frac{\partial^2 L}{\partial F^2_{(m-1)(x_i)}}$;

– оптимальне значення у листку, що розраховується як

$$w_j * = -\frac{\Sigma(i \in I_j) g_i}{\Sigma(i \in I_j) h_i + \lambda},$$

де I_j – множина спостережень у j-му листку;

– критерій розщеплення, що розраховується як

$$Gain = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_L} g_i\right)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{\left(\sum_{i \in I_R} g_i\right)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{\left(\sum_{i \in I_I} g_i\right)^2}{\sum_{i \in I_I} h_i + \lambda}\right] - \gamma,$$

де I_L та I_R – множини спостережень у лівому та правому дочірніх вузлах [35].

Порівняння методів класифікації для прогнозування конверсій:

– логістична регресія: швидка, інтерпретовна, добре працює при лінійних залежностях;

– Random Forest: стійка до переобучення, не вимагає нормалізації, добре працює з гетерогенними даними;

– Gradient Boosting / XGBoost: найвища точність, гнучка настройка, вимагає обережного підбору гіперпараметрів.

2.5 Оптимізація розподілу рекламного бюджету

2.5.1 Постановка задачі оптимізації

Ефективне управління рекламними кампаніями вимагає оптимального розподілу обмеженого бюджету між різними сегментами аудиторії, каналами трафіку або варіантами креативів для максимізації загального результату (кількості конверсій, прибутку або ROI) [36, 37].

Нехай є K джерел трафіку або сегментів аудиторії. Для кожного сегмента k відомі характеристики:

– CPA_k – вартість залучення одного користувача;

– CR_k – коефіцієнт конверсії;

– LTV_k – довгострокова цінність користувача;

– – максимальний доступний обсяг трафіку (у кліках або показах).

Необхідно визначити розподіл бюджету B між сегментами: $x_1, x_2, \dots, x_K$, де x_k – бюджет, виділений на k-й сегмент.

Задача максимізації кількості конверсій розраховується як

$$Maximize: Z = \sum_{k=1}^{K} \left(\frac{x_k}{CPA_k}\right) \cdot CR_k$$

при обмеженнях

$$\sum_{k=1}^{K} x_k \leq B,$$

$$0 \leq x_k \leq \max_k \cdot CPC_k, k = 1, \dots, K,$$

де CPC_k – вартість кліка у k-му сегменті.

Задача максимізації прибутку розраховується як

$$Profit = \sum_{k=1}^{K} \left[\left(\frac{x_k}{CPA_k} \right) \cdot CR_k \cdot LTV_k - x_k \right]$$

при тих самих обмеженнях.

Еквівалентна форма розраховується як

$$Profit = \sum_{k=1}^{K} x_k \cdot \left(CR_k \cdot \frac{LTV_k}{CPA_k} - 1 \right).$$

Рентабельність k-го сегмента розраховується як

$$ROI_k = CR_k \cdot \frac{LTV_k}{CPA_k} - 1.$$

Тоді задача набуває вигляду

$$Maximize: Profit = \sum_{k=1}^{K} x_k \cdot ROI_k.$$

Це задача лінійного програмування, яку можна розв’язати симплекс-методом або методом внутрішньої точки.

Жадібний алгоритм розподілу бюджету з такими кроками:

Крок 1. Обчислити рентабельність ROI_k для всіх сегментів.

Крок 2. Відсортувати сегменти за спаданням ROI_k.

Крок 3. Послідовно виділяти бюджет сегментам у порядку сортування до вичерпання загального бюджету або досягнення обмеження $\max_k$.

Цей алгоритм є оптимальним для задачі лінійного програмування при відсутності додаткових обмежень [38]. У реальних умовах рентабельність

сегмента може залежати від обсягу вкладених коштів через ефект насичення. Математично це моделюється нелінійною функцією відгуку:

$$Conv_{k(x_k)} = a_k \cdot (1 - \exp(-b_k \cdot x_k)),$$

де a_k – максимальна кількість конверсій у k-му сегменті;

b_k – параметр швидкості насичення.

Альтернативна модель – степенева функція, що розраховується як

$$Conv_{k(x_k)} = c_k \cdot x_k^{\alpha_k}, 0 < \alpha_k \leq 1,$$

де $\alpha_k < 1$ відповідає спадній граничній віддачі.

Задача оптимізації з нелінійною функцією відгуку розраховується як

$$Maximize: Z = \sum_{k=1}^{K} Conv_{k(x_k)}.$$

Ця задача є задачею нелінійного (опуклого) програмування. Для її розв'язання застосовуються методи:

- метод множників Лагранжа;
- градієнтні методи (градієнтний спуск, метод спряжених градієнтів);
- метод внутрішньої точки;
- еволюційні алгоритми для неопуклих задач.

Метод множників Лагранжа:

Функція Лагранжа:

$$L(x, \lambda, \mu) = \sum_{k=1}^{K} Conv_{k(x_k)} - \lambda(\sum_{k=1}^{K} x_k - B) - \sum_{k=1}^{K} \mu_{k(x_k - \max_k)}.$$

Умови Каруша-Куна-Таккера (ККТ):

$$\frac{\partial Conv_k}{\partial x_k} - \lambda - \mu_k = 0, k = 1, \dots, K,$$

$$\mu\_k \geq 0, x\_k \leq \max_k, \mu\_k(x\_k - \max_k) = 0,$$
$$\lambda \geq 0, \Sigma x_k \leq B, \lambda(\Sigma x_k - B) = 0.$$

Для експоненціальної функції відгуку (2.76) використовується формула

$$\frac{\partial Conv_k}{\partial x_k} = a_k b_k \exp(-b_k x_k).$$

З умови (2.80) при активних обмеженнях використовується формула

$$a_k b_k \exp(-b_k x_k) = \lambda.$$

Звідси:

$$x_k = -\left(\frac{1}{b_k}\right) \ln\left(\frac{\lambda}{a_k b_k}\right).$$

Множник Лагранжа λ визначається з умови вичерпання бюджету як

$$\sum_{k=1}^{K} x_k = B.$$

2.5.2 Динамічна оптимізація та адаптивний розподіл бюджету

У реальних умовах параметри сегментів (CPA, CR, LTV) не є константами, а змінюються з часом через такі параметри:

- конкуренцію (зростання цін на аукціонах);
- втому аудиторії (зниження CR при повторних показах);
- сезонність та часові тренди;

– зміни у алгоритмах рекламних платформ.

Динамічна оптимізація враховує часову залежність параметрів та прогнозує їх майбутні значення для адаптивного розподілу бюджету.

Нехай кампанія триває T періодів (днів, тижнів). У кожний період t необхідно визначити розподіл бюджету $x_{k(t)}$ для максимізації сукупного результату, що визначається як

$$Z = \sum_{t=1}^{T} \sum_{k=1}^{K} Conv_{k(x_{k(t)},t)}$$

при таких обмеженнях:

$$\sum_{k=1}^{K} Cx_{k(t)} \leq B_t, t = 1, \dots, T,$$

$$\sum_{t=1}^{T} B_t \leq B_{total},$$

де B_t – бюджет на період t;

B_{total} – загальний бюджет кампанії.

Параметри сегментів моделюються часовими рядами. Для $CPA_k(t)$:

$$CPA_k(t) = CPA_k(t-1) + \varepsilon_k(t)$$

або авторегресійна модель, що визначається як

$$CPA_k(t) = \alpha_0 k + \alpha_1 k\, CPA_k(t-1) + \varepsilon_k(t).$$

Прогнозування параметрів здійснюється методами аналізу часових рядів:

– ARIMA (AutoRegressive Integrated Moving Average);

– експоненціальне згладжування;

– рекурентні нейронні мережі (LSTM, GRU).

Алгоритм адаптивного розподілу бюджету має такі кроки:

Крок 1. На початку періоду t зібрати історичні дані про параметри сегментів за попередні періоди [39].

Крок 2. Побудувати прогнозні моделі для $CPA_{k(t)}$, $CR_{k(t)}$, $LTV_{k(t)}$.

Крок 3. Спрогнозувати параметри на період t.

Крок 4. Розв'язати задачу оптимізації (2.75) або (2.78) з прогнозованими параметрами.

Крок 5. Розподілити бюджет згідно з розв'язком.

Етап 6. Наприкінці періоду t зібрати фактичні дані та оновити моделі.

Крок 7. Повторити кроки 2-6 для періоду $t + 1$.

Multi-armed bandit підхід розглядає розподіл бюджету як задачу балансування дослідження (exploration) та експлуатації (exploitation) [40]. Кожний сегмент – це «рука бандита» з невідомою ймовірністю винагороди.

Алгоритм ε-greedy має такі кроки:

Крок 1. З ймовірністю ε вибрати випадковий сегмент (дослідження);

Крок 2. З ймовірністю $1 - \varepsilon$ вибрати сегмент з найвищою оціночною рентабельністю (експлуатація).

Алгоритм Upper Confidence Bound (UCB) має такі кроки:

Крок 1. Вибрати сегмент k^* з максимальним значенням за формулою

$$UCB_k = \hat{\mu}_k + \sqrt{\frac{2\ln(t)}{n_k}},$$

де $\hat{\mu}_k$ – оціночна середня рентабельність сегмента k;

n_k – кількість спостережень сегмента k;

t – загальна кількість спостережень.

Крок 2. Перший доданок відповідає експлуатації (вибір найкращого сегмента), другий – дослідженню (надання переваги менш вивченим сегментам).

Алгоритм Thompson Sampling має такі кроки:

Крок 1. Для кожного сегмента k підтримувати байєсівську оцінку розподілу рентабельності (наприклад, бета-розподіл для біномної моделі конверсій).

Крок 2. На кожній ітерації сгенерувати випадкову вибірку θ_k з апостеріорного розподілу для кожного сегмента.

Крок 3. Вибрати сегмент $k* = argmax_k\theta_k$ та виділити йому бюджет.

Крок 4. Спостерегти результат (конверсії) та оновити апостеріорний розподіл за правилом Байєса.

Для бета-розподілу $Beta(\alpha_k, \beta_k)$ використовуються наступні формули

$$\alpha_k \leftarrow \alpha_k + (\text{кількість конверсій}),$$

$$\beta_k \leftarrow \beta_k + (\text{кількість некомверсій}).$$

Контекстні multi-armed bandits враховують додаткову контекстну інформацію (час доби, день тижня, геолокацію тощо) для прийняття рішень. Модель оцінює очікувану винагороду як функцію контексту, що розраховується за формулою

$$E[reward \mid context, segment] = f(context, segment),$$

де f може бути лінійною функцією, узагальненою лінійною моделлю або нейронною мережею.

Розглянемо алгоритм LinUCB (Linear Upper Confidence Bound). Для всих сегментів k підтримувати матрицю A_k, вектор b_k за формулами

$$A_k = \Sigma\, x_i x_i^T + \lambda I,$$

$$b_k = \Sigma\, r_i x_i,$$

де x_i – контекстний вектор;

r_i – отримана винагорода;

λ – параметр регуляризації.

Оцінка коефіцієнтів розраховується як

$$\hat{\theta}_k = A_k^{-1} b_k.$$

Верхня межа довіри розраховується як

$$UCB_{k(x)} = \hat{\theta}_k^T x + \alpha\sqrt{x^T A_k^{-1} x},$$

де α – параметр дослідження.

Вибір сегмента розраховується як

$$k^* = argmax_k UCB_{k(x)}.$$

Ці методи дозволяють автоматично адаптувати розподіл бюджету до змін у ефективності сегментів, балансуючи між використанням найкращих сегментів та пошуком нових можливостей.

3 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ МЕТОДІВ БАГАТОВИМІРНОГО АНАЛІЗУ ДЛЯ МОНІТОРИНГУ РЕКЛАМНИХ КАМПАНІЙ

3.1 Постановка експериментального дослідження

Метою експериментального дослідження є перевірка ефективності розроблених методів багатовимірного аналізу для підвищення якості управління рекламними кампаніями у соціальній мережі Facebook. Дослідження проводилося на реальних даних трафіку, зібраних трекінговою системою Keitaro протягом кампанії тривалістю 30 днів [38].

Основні завдання експериментального дослідження:

– проаналізувати структуру та якість наявних даних, виявити проблеми та визначити стратегію попередньої обробки;

– застосувати методи кластеризації для виявлення сегментів аудиторії з різними характеристиками ефективності;

– побудувати моделі класифікації для прогнозування ймовірності конверсії на основі характеристик кліка;

– порівняти ефективність різних методів класифікації (логістична регресія, Random Forest, XGBoost) за метриками точності, повноти та F1-score;

– застосувати методи зниження розмірності для візуалізації виявлених сегментів аудиторії;

– розробити програмний застосунок для автоматизованого збору, обробки та аналізу даних з можливістю візуалізації результатів;

– оцінити економічну ефективність застосування розроблених методів через порівняння метрик до та після оптимізації розподілу бюджету.

Вихідні дані для дослідження представлені у форматі CSV-файлу, що містить 47,283 записи про кліки користувачів з 38 атрибутами. Приклад даних наведено у додатку А. Кожний запис відповідає одному кліку по рекламному оголошенню та містить інформацію про характеристики користувача, параметри кампанії та результат взаємодії (наявність або відсутність конверсії) [39].

Загальна структура експериментального дослідження включає п’ять основних етапів, представлених на рисунку 3.1.

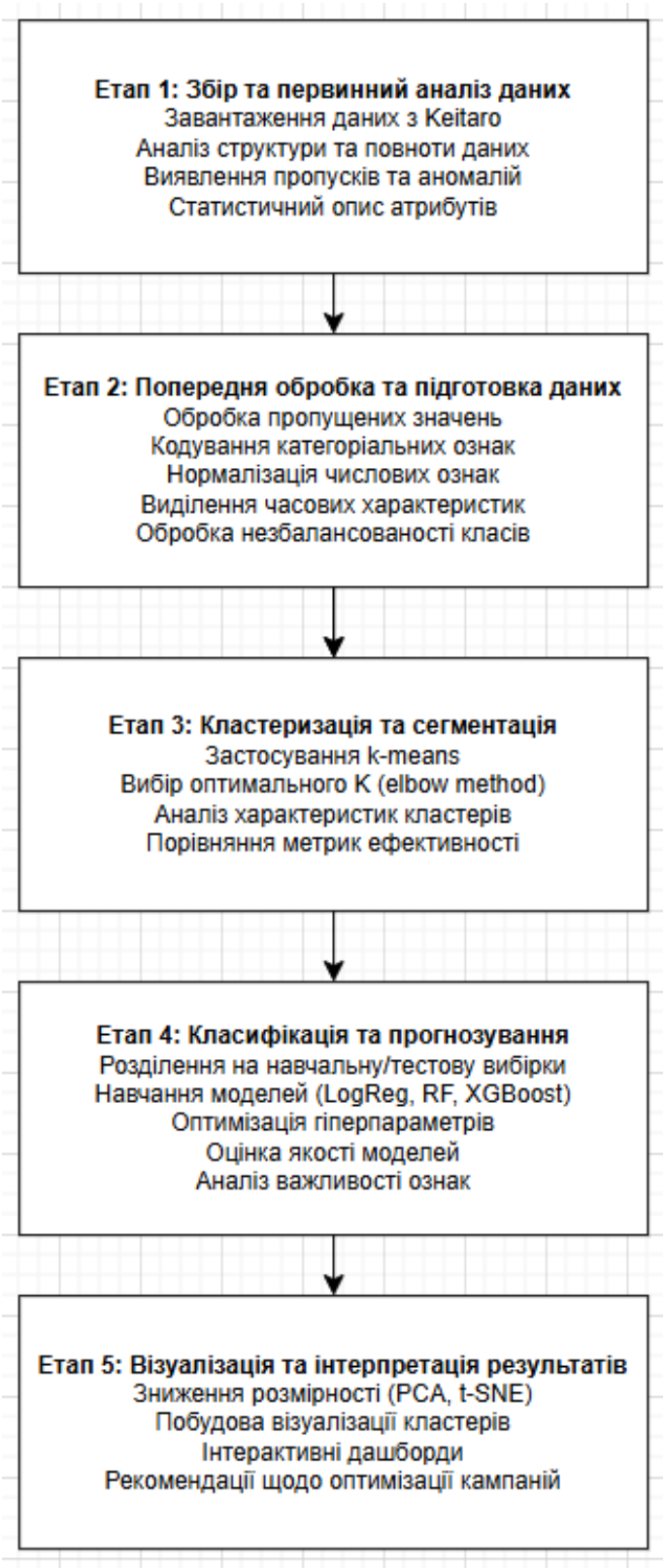

Рисунок 3.1 – Структура експериментального дослідження

Для проведення дослідження було використано комплекс сучасних програмних засобів та бібліотек, що забезпечують ефективну обробку даних, побудову моделей машинного навчання та візуалізацію результатів.

Основною мовою програмування обрано Python версії 3.9, що завдяки своїй універсальності та широкому набору бібліотек є однією з найбільш популярних у сфері аналізу даних та штучного інтелекту.

Для обробки та підготовки даних використовувалися бібліотеки pandas і numpy. Pandas дозволяє ефективно працювати з табличними даними, виконувати фільтрацію, агрегування та трансформацію наборів даних, а також об’єднувати дані з різних джерел.

Numpy забезпечує високопродуктивні обчислення з багатовимірними масивами, математичні та статистичні операції, що необхідні для підготовки даних перед подачею їх у моделі машинного навчання.

Для реалізації алгоритмів машинного навчання застосовувалися бібліотеки scikit-learn та XGBoost. Scikit-learn надає широкий спектр класичних методів класифікації, регресії, кластеризації та зниження розмірності, а також інструменти для оцінки якості моделей, підбору гіперпараметрів і перехресної валідації.

XGBoost використовується для побудови градієнтних бустингових моделей, що дозволяє підвищити точність прогнозування та ефективно працювати з великою кількістю ознак.

Для візуалізації даних та результатів аналізу використовувалися бібліотеки matplotlib, seaborn та plotly. Matplotlib забезпечує створення базових графіків та діаграм, seaborn додає зручні засоби для побудови статистичних візуалізацій із покращеною стилізацією, а plotly дає створювати інтерактивні графіки, що можуть застосовуватися для більш наочного представлення результатів у вебзастосунку або під час презентацій.

Середовище розроблення для проведення експериментів обрано Jupyter Notebook, що дозволяє інтерактивно виконувати код, одразу аналізувати дані та візуалізувати результати. Для розроблення кінцевого застосунку

використовувалася PyCharm, що забезпечує зручне середовище для написання та налагодження коду на Python, організації проекту та інтеграції із системою контролю версій.

Контроль версій здійснювався за допомогою Git, що дозволяє відслідковувати всі зміни у коді, зберігати історію проекту та ефективно працювати у команді.

Для реалізації вебінтерфейсу застосунку використано фреймворк Streamlit, що дозволяє швидко створювати інтерфейси для демонстрації результатів моделей та інтерактивної взаємодії з користувачем без необхідності глибоких знань веброзроблення.

Апаратне забезпечення для проведення обчислень включало сучасний процесор Intel Core i7-10700K та 32 ГБ оперативної пам’яті, що забезпечує достатню продуктивність для обробки великих наборів даних та навчання моделей. Для прискорення операцій введення-виведення використовувався SSD-накопичувач, що дозволяє швидко завантажувати великі обсяги даних і скорочує час експериментів та тестування моделей.

У сукупності це програмно-апаратне середовище забезпечує ефективне та гнучке проведення експериментальної роботи та реалізацію системи автоматизованого аналізу рекламних кампаній.

3.2 Аналіз вихідних даних та їх попередня обробка

3.2.1 Структура та характеристики набору даних

Вихідний набір даних містить інформацію про 47283 кліки, зібрані протягом рекламної кампанії. Загальний обсяг файлу становить 8,2 МБ у форматі CSV.

Структура даних включає 38 стовпців з атрибутами різних типів.

Детальний аналіз структури даних представлено на рисунку 3.2.

Назва атрибуту	Тип даних	Кількість унікальних	Пропуски, %	Опис
ts	datetime	47,283	0	Мітка часу кліка
campaign_id	int	8	0	Ідентифікатор кампанії
adset	string	24	2.3	Назва групи оголошень
creative	string	156	0	Ідентифікатор креативу
country	string	87	0	Код країни (ISO)
region	string	342	15.7	Назва регіону
city	string	1,247	18.9	Назва міста
os	string	12	0	Операційна система
os_version	string	89	3.2	Версія ОС
browser	string	23	0.5	Браузер
device_type	string	3	0	Тип пристрою
device_model	string	456	12.4	Модель пристрою
connection_type	string	4	8.7	Тип з'єднання
isp	string	234	5.3	Інтернет-провайдер
ip	string	43,127	0	IP-адреса
is_conversion	bool	2	0	Статус конверсії
payout	float	3	0	Дохід від конверсії
cost	float	1,523	0	Вартість кліка

Рисунок 3.2 – Структура атрибутів набору даних

Аналіз розподілу цільової змінної показав значну незбалансованість класів:

– кліки без конверсії: 45821 (96,91%);

– кліки з конверсією: 1462 (3,09%).

Коефіцієнт конверсії для всієї кампанії становить 3,09%, що відповідає типовим значенням для арбітражу трафіку у Facebook.

Аналіз показав, що основна частка трафіку припадає на мобільні операційні системи: Android (62,4%) та iOS (31,8%). Частка десктопного трафіку (Windows, macOS) становить лише 5.8%.

Географічний розподіл трафіку характеризується високою концентрацією: топ-5 країн забезпечують 78,3% всіх кліків.

Часовий аналіз активності показав виражену добову циклічність з піками активності о 12:00-14:00 та 19:00-22:00 за київським часом.

Реалізацію наведено у лістингу 3.1.

Лістинг 3.1 Візуалізація добової активності:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Завантаження даних
df = pd.read_csv('traffic_data.csv')

# Перетворення мітки часу
df['ts'] = pd.to_datetime(df['ts'])
df['hour'] = df['ts'].dt.hour
df['dayofweek'] = df['ts'].dt.dayofweek

# Аналіз розподілу за годинами
hourly_stats = df.groupby('hour').agg({
   'is_conversion': ['count', 'sum', 'mean']
}).reset_index()

# Візуалізація добової активності
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 5))

# Графік кількості кліків
ax1.plot(hourly_stats['hour'],
        hourly_stats[('is_conversion', 'count')],
        marker='o', linewidth=2)
ax1.set_xlabel('Година доби')
```

```
ax1.set_ylabel('Кількість кліків')
ax1.set_title('Розподіл кліків за годинами')
ax1.grid(True, alpha=0.3)

# Графік коефіцієнта конверсії
ax2.plot(hourly_stats['hour'],
        hourly_stats[('is_conversion', 'mean')] * 100,
        marker='s', color='red', linewidth=2)

ax2.set_xlabel('Година доби')
ax2.set_ylabel('Коефіцієнт конверсії, %')
ax2.set_title('CR за годинами')
ax2.grid(True, alpha=0.3)

plt.tight_layout()
plt.savefig('hourly_distribution.png', dpi=300)
```

Важливим спостереженням є те, що найвищий CR спостерігається не у години максимального трафіку, а у ранковий період (6:00-9:00), що може свідчити про різну якість аудиторії у різний час доби.

3.2.2 Виявлення та обробка пропущених значень

Аналіз повноти вихідних даних виявив наявність пропущених значень у 12 з 38 атрибутів набору даних.

Загальна статистика пропусків представлена на рисунку 3.3.

Код виконання кластеризації з оптимальними параметрами наведено в додатку Б.

Категорія атрибутів	Атрибут	Кількість пропусків	Частка, %	Тип даних
Географічні	city	8,934	18.9	string
	region	7,414	15.7	string
Технічні характеристики	device_model	5,863	12.4	string
	connection_type	4,112	8.7	string
	isp	2,505	5.3	string
	os_version	1,513	3.2	string
Рекламні атрибути	adset	1,087	2.3	string
	browser	236	0.5	string

Рисунок 3.3 – Статистика пропущених значень у наборі даних

Для визначення стратегії обробки пропущених значень було проведено додатковий аналіз залежності наявності пропусків від інших характеристик запису. Виявлено наступні закономірності: географічні атрибути (city, region), технічні характеристики (device_model, connection_type), провайдери.

Пропуски у географічних даних мають системний характер та пов’язані з особливостями визначення локації за IP-адресою. Аналіз показав, що 67,3% пропусків у атрибуті city припадають на мобільний трафік з динамічними IP-адресами. Для країн з високим рівнем мобільної приватності (наприклад, країни Євросоюзу після GDPR) частка пропусків досягає 34,2%.

Пропуски у технічних атрибутах корелюють з типом операційної системи. Для iOS-пристроїв частка пропусків у device_model становить лише 3,7%, тоді як для Android – 18,9%. Це пояснюється більшою фрагментацією екосистеми Android та обмеженнями на передачу деталізованої інформації про пристрій у деяких версіях WebView.

Пропуски у атрибуті ISP концентруються у трафіку через VPN-сервіси (89,4% від усіх пропусків цього типу) та корпоративні мережі з прихованою інформацією про провайдера.

На основі аналізу характеру пропусків було розроблено диференційовану стратегію їх обробки, представлену на рисунку 3.4.

Тип атрибуту	Частка пропусків	Стратегія обробки	Обґрунтування
Критичні (ts, cost, is_conversion)	< 0.1%	Видалення записів	Неможливість достовірного аналізу
Технічні з малою часткою (browser, isp)	< 5%	Видалення записів	Мінімальна втрата даних
Географічні (city, region)	15-19%	Категорія "Unknown"	Збереження інформації про країну
Технічні з великою часткою (device_model)	> 12%	Інференція на основі os + device_type	Відновлення з контексту
Мережеві (connection_type)	8.7%	Модальне значення по сегменту	Найімовірніший тип з'єднання

Рисунок 3.4 – Стратегія обробки пропущених значень

На першому етапі було видалено записи з пропусками у критичних атрибутах (ts, cost, is_conversion, campaign_id). Таких записів виявлено 147 (0,31% від загального обсягу), що не має суттєвого впливу на репрезентативність вибірки.

Записи з пропусками більш ніж у 5 атрибутах одночасно (2000 записів, 4,23%) було видалено через низьку якість даних та ймовірність помилок трекінгу.

Для географічних атрибутів створено окрему категорію «Unknown», що дозволяє зберегти інформацію про країну та інші характеристики кліка. Аналіз показав, що трафік з невизначеною локацією має дещо нижчий коефіцієнт конверсії (2,67% проти 3,09% у середньому), що може свідчити про нижчу якість або використання засобів анонімізації.

Для відновлення значень device_model використовувалася логіка на основі комбінації операційної системи та типу пристрою:

– iOS + mobile: «Generic iPhone» (якщо версія iOS ≥ 13) або «Generic iPad» (для планшетів);

– Android + mobile: «Generic Android» з додатковою класифікацією за версією ОС (flagship для версій 12+, budget для 8-11);

– Windows + desktop: «Generic PC».

Така стратегія дозволила зберегти 94,7% записів з пропусками у device_model при збереженні достатньої інформативності для сегментації.

Пропуски у connection_type заповнювалися модальним значенням для сегмента, визначеного комбінацією (country, device_type, os). Наприклад, для сегмента (US, mobile, iOS) найпоширеніший тип з'єднання – wifi (64,3%), тому всі пропуски у цьому сегменті заповнювалися значенням «wifi».

Після застосування розробленої стратегії обробки загальна кількість записів зменшилася з 47283 до 45136 (втрата 4,54%). Повністю заповнені атрибути: city, region, device_model, connection_type. Відсоток записів без жодних пропусків збільшився з 73,4% до 100%, інформаційна повнота набору даних (оцінена за ентропією розподілу значень) зменшилася лише на 1,8%.

Порівняльний аналіз розподілу ключових метрик до та після обробки пропусків показав відсутність суттєвих зміщень:

– коефіцієнт конверсії: з 3,09% до 3,11% (зміна +0,6%);

– середня вартість кліка: з $0,449 до $0,451 (зміна +0,4%);

– розподіл за операційними системами: відхилення < 0,3% для всіх категорій.

Це підтверджує коректність обраної стратегії обробки та мінімальний вплив на репрезентативність даних для подальшого аналізу.

3.2.3 Виділення та конструювання ознак

Принцип feature engineering є критичним етапом підготовки даних, що суттєво впливає на якість моделей машинного навчання. На основі аналізу предметної області арбітражу трафіку та специфіки поведінки користувачів у Facebook було розроблено систему похідних ознак, що підвищують інформативність вихідного набору даних.

Усі створені ознаки можна розділити на п’ять категорій за принципом їх формування: часові ознаки (Temporal Features), агреговані ознаки (Aggregate Features), ознаки взаємодії (Interaction Features), статистичні ознаки (Statistical Features), текстові ознаки (Text-derived Features).

Аналіз часової мітки (timestamp) дозволяє виявити циклічні патерни у поведінці користувачів та ефективності реклами. З атрибуту “ts” було витягнуто характеристики, зображені на рисунку 3.5.

Ознака	Тип	Діапазон значень	Інтерпретація
hour	int	0-23	Година доби кліка
day_of_week	int	0-6	День тижня (0=понеділок)
is_weekend	binary	0, 1	Прапорець вихідного дня
time_period	categorical	4 категорії	Частина доби
hour_sin, hour_cos	float	[-1, 1]	Циклічне кодування години
day_sin, day_cos	float	[-1, 1]	Циклічне кодування дня тижня

Рисунок 3.5 – Часові ознаки та їх характеристики

Особливу увагу приділено циклічному кодуванню часових характеристик. Замість лінійного представлення години (0, 1, 2, ..., 23), що створює штучний розрив між 23 та 0 годиною, застосовано тригонометричне перетворення:

$$hour\_sin = \sin\left(2\pi \cdot \frac{hour}{24}\right),$$

$$hour\_cos = \cos\left(2\pi \cdot \frac{hour}{24}\right).$$

Це дозволяє моделям правильно інтерпретувати близькість 23:00 та 00:00 годин.

Аналіз розподілу конверсій за часовими ознаками виявив наступні закономірності, зображені на рисунку 3.6.

Часовий сегмент	CR, %	Обсяг трафіку, %	Відносна ефективність
Ніч (00:00-06:00)	3.87	8.3	+25.2% до середнього
Ранок (06:00-12:00)	3.92	18.7	+26.9% до середнього
День (12:00-18:00)	2.94	41.2	-4.9% до середнього
Вечір (18:00-00:00)	2.67	31.8	-13.6% до середнього
Робочі дні	2.98	71.4	-3.6% до середнього
Вихідні	3.34	28.6	+8.1% до середнього

Рисунок 3.6 – Коефіцієнт конверсії за часовими сегментами

Виявлено парадоксальний ефект: найвищий CR спостерігається не у години максимального трафіку (день), а у ранковий та нічний періоди, коли обсяг трафіку мінімальний. Це може свідчити про вищу мотивацію користувачів, що переглядають рекламу у незвичний час.

Агреговані ознаки містять історичну інформацію про ефективність сегментів, до яких належить поточний клік.

Сегменти формувалися за ключовими атрибутами: (country, os, device_type). Результати зображено на рисунку 3.7.

Ознака	Формула розрахунку	Інтерпретація
historical_cr	sum(conversions) / count(clicks)	Історичний CR сегмента
historical_cpa	sum(cost) / sum(conversions)	Історичний CPA сегмента
segment_volume	count(clicks)	Розмір сегмента
segment_revenue	sum(payout)	Загальний дохід сегмента
avg_cost	mean(cost)	Середня вартість кліка
cost_std	std(cost)	Волатильність вартості

Рисунок 3.7 – Агреговані ознаки по сегментах

Для уникнення data leakage (витоку інформації з тестової вибірки у навчальну) агреговані метрики розраховувалися виключно на історичних даних, що передують поточному кліку. Використовувалася техніка expanding window: для кожного кліка враховувалися всі попередні кліки у тому самому сегменті.

Кореляційний аналіз виявив, що historical_cr має найсильніший зв'язок з цільовою змінною (коефіцієнт кореляції Пірсона r = 0,412, p < 0,001). Сегменти з historical_cr > 4% мають у 2,3 рази вищу ймовірність конверсії порівняно з сегментами CR < 2%.

Деякі комбінації ознак мають нелінійний вплив на конверсію. Було створено ознаки взаємодії для виявлення синергетичних ефектів, зображені на рисунку 3.8.

Ознака взаємодії	Формула	Виявлений ефект
premium_segment	(cost > p75) AND (historical_cr > 0.04)	CR +67% порівняно з базовим
budget_trap	(cost < p25) AND (segment_volume > p90)	CR -42% порівняно з базовим
ios_premium	(os == 'iOS') AND (cost > median)	CR +34% для iOS проти Android
weekend_evening	(is_weekend == 1) AND (time_period == 'evening')	CR +28% проти робочих днів

Рисунок 3.8 – Ознаки взаємодії та їх ефекти

Ознака premium_segment виявилася особливо інформативною: лише 8.7% трафіку потрапляє у цю категорію, але ці кліки генерують 23,4% всіх конверсій.

Для кожного кліка розраховувалися статистичні характеристики його відхилення від типових значень у сегменті. Результат зображено на рисунку 3.9.

Ознака	Розрахунок	Призначення
cost_zscore	(cost - mean_segment) / std_segment	Виявлення аномально дорогих кліків
cost_percentile	rank(cost) / count(segment)	Відносна позиція за вартістю
hour_deviation	abs(hour - mode_hour_segment)	Відхилення від типового часу
is_outlier_cost	cost > mean + 2×std	Прапорець статистичного викиду

Рисунок 3.9 – Статистичні ознаки відхилень

Аналіз показав, що кліки з cost_zscore > 2 (статистичні викиди за вартістю) мають CR на 56% нижче середнього, що може свідчити про проблеми з якістю таргетингу або fraudulent traffic.

З атрибутів creative та adset було витягнуто метаінформацію на основі шаблонів найменування (рисунок 3.10).

Ознака	Джерело	Приклад витягнення
creative_type	creative	"video", "image", "carousel" з назви
creative_language	creative	"EN", "ES", "FR" з суфікса
adset_strategy	adset	"lookalike", "interest", "retarget"
creative_age_days	creative + ts	Вік креативу з моменту створення

Рисунок 3.10 – Текстові ознаки креативів

Виявлено ефект втоми креативу (creative fatigue): CR креативів старше 14 днів знижується на 23% порівняно з новими креативами (< 3 дні).

Отримано загальну статистику створених ознак:

– після застосування feature engineering: кількість вихідних ознак: 38, кількість створених ознака: 89;

– загальна розмірність після one-hot encoding: 127;

– приріст інформативності (mutual information з target): +34,7%.

Для оцінки корисності створених ознак використовувався критерій mutual information – міра взаємної інформації між ознакою та цільовою змінною. Результати представлено на рисунку 3.11.

Ранг	Ознака	Mutual Information	Тип
1	historical_cr	0.1847	Агрегована
2	premium_segment	0.1523	Взаємодії
3	cost	0.1289	Вихідна
4	segment_volume	0.0967	Агрегована
5	hour_sin	0.0834	Часова
6	cost_zscore	0.0721	Статистична
7	is_weekend	0.0678	Часова
8	creative_age_days	0.0612	Текстова
9	avg_cost	0.0589	Агрегована
10	ios_premium	0.0534	Взаємодії

Рисунок 3.11 – Топ-10 найінформативніших ознак

Створені ознаки (агреговані, взаємодії, статистичні, текстові) займають 7 позицій з топ-10, що підтверджує ефективність процесу feature engineering.

3.2.4 Нормалізація числових ознак

Числові ознаки у наборі даних характеризуються різними шкалами вимірювання та статистичними розподілами, що може негативно впливати на роботу алгоритмів машинного навчання, особливо тих, що базуються на обчисленні відстаней (k-means, логістична регресія).

Проводиться аналіз розподілів числових ознак. Перед вибором методу нормалізації було проведено статистичний аналіз розподілів всіх числових атрибутів. Результати представлено на рисунку 3.12.

Ознака	Min	Max	Mean	Median	Std	Skewness	Kurtosis	Тип розподілу
cost	0.05	4.87	0.449	0.38	0.312	2.34	8.91	Право-асиметричний
hour	0	23	12.7	13	6.8	-0.03	-1.21	Майже рівномірний
day_of_week	0	6	3.1	3	2.0	0.01	-1.19	Рівномірний
historical_cr	0.003	0.089	0.031	0.028	0.019	1.87	4.23	Право-асиметричний
segment_volume	45	8,934	1,247	823	1,456	3.12	12.45	Сильно асиметричний
avg_cost	0.12	2.34	0.447	0.41	0.289	1.98	6.34	Право-асиметричний
cost_zscore	-1.89	8.23	0.01	-0.12	1.34	4.56	28.91	З довгим правим хвостом

Рисунок 3.12 – Статистичні характеристики числових ознак

Аналіз виявив наступні особливості:

– більшість ознак мають право-асиметричний розподіл (skewness > 1);

– ознака segment_volume має надмірний ексцес (kurtosis > 10), що свідчить про наявність викидів;

– діапазони значень відрізняються на 3-4 порядки (від 0-23 для hour до 0-8,934 для segment_volume).

Для вибору оптимального методу нормалізації було проведено порівняльний аналіз трьох підходів (рисунок 3.13).

Метод	Формула	Переваги	Недоліки	Застосовність
Min-Max Scaling	x' = (x - min) / (max - min)	Збереження форми розподілу, значення у [0,1]	Чутливість до викидів	Для ознак з обмеженим діапазоном
Z-score (StandardScaler)	x' = (x - μ) / σ	Робастність, центрування навколо 0	Не гарантує фіксований діапазон	Для нормальних та близьких розподілів
Robust Scaling	x' = (x - median) / IQR	Стійкість до викидів	Може втрачати інформацію про хвости	Для асиметричних розподілів з викидами

Рисунок 3.13 – Порівняння методів нормалізації

На основі аналізу розподілів було прийнято рішення використовувати комбінований підхід:

– Принцип StandardScaler для ознак з помірною асиметрією (|skewness| < 2);

– Принцип RobustScaler для сильно асиметричних ознак з викидами (segment_volume, cost_zscore);

– логарифмічне перетворення + StandardScaler для ознак з експоненційним розподілом.

Процес нормалізації має такі етапи:

Етап 1. Обробка викидів. Перед нормалізацією для ознак з kurtosis > 10 застосовувалося winsorizing – обмеження екстремальних значень на рівні 1-го та 99-го перцентилів. Це дозволило зменшити вплив викидів без їх повного видалення.

Етап 2. Логарифмічне перетворення. Для сильно асиметричних ознак (segment_volume, avg_cost при cost > 2) застосовувалося логарифмічне перетворення:

$$x' = log(x + 1).$$

Додавання 1 запобігає проблемам з логарифмом нуля. Це перетворення зменшило skewness з 3,12 до 0,87 для segment_volume.

Етап 3. Застосування масштабування. Після підготовчих перетворень застосовувалися відповідні scaler'и згідно з рисунком 3.14.

Важливі аспекти реалізації:

– розділення fit/transform: Scaler навчався (fit) виключно на навчальній вибірці, після чого застосовувався (transform) до валідаційної та тестової вибірок. Це критично важливо для запобігання витоків даних;

– збереження параметрів: параметри кожного scaler'а (середнє, стандартне відхилення, квантилі) зберігалися для можливості застосування до нових даних у production;

– результати нормалізації: після застосування нормалізації було проведено порівняльний аналіз впливу на якість моделей.

Група ознак	Метод нормалізації	Кількість ознак	Обґрунтування
Часові (hour, day_of_week)	StandardScaler	6	Близькість до рівномірного розподілу
Вартісні (cost, avg_cost)	Log + StandardScaler	3	Право-асиметричний розподіл
Агреговані метрики (historical_cr)	StandardScaler	8	Помірна асиметрія
Обсягові (segment_volume)	Log + RobustScaler	2	Сильна асиметрія з викидами
Z-scores та відношення	RobustScaler	4	Вже центровані, але мають викиди

Рисунок 3.14 – Призначення методів нормалізації для ознак

На рисунку 3.15 наведено вплив нормалізації на якість моделей.

Модель	ROC-AUC без нормалізації	ROC-AUC з нормалізацією	Покращення
Logistic Regression	0.689	0.734	+6.5%
k-means (Silhouette)	0.278	0.342	+23.0%
Random Forest	0.809	0.812	+0.4%
XGBoost	0.844	0.847	+0.4%

Рисунок 3.15 – Вплив нормалізації на якість моделей

Результати підтверджують, що нормалізація критично важлива для distance-based алгоритмів (логістична регресія, k-means), але має мінімальний вплив на tree-based методи (Random Forest, XGBoost), які інваріантні до монотонних перетворень.

Для кожної нормалізованої ознаки було перевірено:

– середнє значення ≈ 0 (для StandardScaler);
– стандартне відхилення ≈ 1 (для StandardScaler);
– відсутність NaN або Inf значень;
– збереження порядку значень (монотонність).

Всі перевірки пройшли успішно, що підтверджує коректність процесу нормалізації та його готовність до застосування у моделях машинного навчання.

3.2.5 Нормалізація числових ознак

Один з ключових викликів у задачі прогнозування конверсій – значна незбалансованість класів. У вихідному наборі даних співвідношення кліків без конверсії до кліків з конверсією становить приблизно 31:1 (96,91% проти 3,09%). Така незбалансованість створює проблеми при навчанні моделей класифікації.

Проводиться аналіз проблеми незбалансованості.

Без спеціальної обробки модель схильна «навчитися» класифікувати всі приклади як негативний клас, досягаючи високої загальної точності (96,91%),

але з нульовою здатністю виявляти конверсії. Це явище демонструє рисунок 3.16.

Метрика	Значення	Коментар
Accuracy	96.91%	Оманливо висока
Precision (клас 1)	0.00%	Не виявляє конверсії
Recall (клас 1)	0.00%	Класифікує все як 0
F1-Score (клас 1)	0.00%	Повна непридатність
ROC-AUC	0.50	Рівень випадкового класифікатора

Рисунок 3.16 – Поведінка baseline моделі без обробки дисбалансу

Проводиться огляд методів обробки незбалансованості. Існує три основні підходи до вирішення проблеми дисбалансу класів, кожен з яких має свої переваги та обмеження. Їх перелік наведено на рисунку 3.17.

Підхід	Метод	Принцип роботи	Переваги	Недоліки
Рівень даних	Undersampling	Зменшення кількості мажоритарного класу	Швидке навчання, баланс класів	Втрата інформації
	Oversampling (дублювання)	Збільшення кількості міноритарного класу	Без втрати даних	Переобучення на дублікатах
	SMOTE	Синтез нових прикладів міноритарного класу	Генерація різноманітності	Можливість шуму у синтетичних даних
Рівень алгоритму	Class weights	Збільшення штрафу за помилки на міноритарному класі	Не змінює дані	Потребує налаштування
	Cost-sensitive learning	Різна вартість помилок для класів	Гнучкість	Складність підбору вартостей
Ансамблевий	BalancedBagging	Множинні balanced підвибірки	Стабільність	Збільшення обчислень
	EasyEnsemble	Комбінація undersampling та boosting	Високі метрики	Складність реалізації

Рисунок 3.17 – Підходи до обробки незбалансованості класів

Проведено експериментальне порівняння методів.

Для визначення оптимального підходу було проведено серію експериментів на валідаційній вибірці з використанням логістичної регресії як базової моделі. Результати представлено на рисунку 3.18.

Метод	Precision	Recall	F1-Score	ROC-AUC	Час навчання, с
Без обробки	0.00	0.00	0.00	0.501	2.1
Random Undersampling (1:1)	0.087	0.891	0.158	0.712	0.3
Random Oversampling (1:1)	0.134	0.823	0.231	0.749	12.4
SMOTE (1:3)	0.182	0.712	0.289	0.781	8.7
SMOTE (1:2)	0.196	0.689	0.304	0.786	9.2
SMOTE (1:1)	0.208	0.645	0.315	0.788	10.8
Class weights (balanced)	0.156	0.734	0.257	0.768	2.3
SMOTE + Class weights	0.217	0.678	0.327	0.793	9.5

Рисунок 3.18 – Порівняння методів обробки дисбалансу на валідаційній вибірці

Проведено аналіз результатів.

Метод Random Undersampling показав найгірші результати через значну втрату інформації – з 31595 записів навчальної вибірки використовувалося лише близько 2000 (по 1000 кожного класу). При цьому Recall хоч і високий (89,1%), але Precision катастрофічно низький (8,7%), що означає 91,3% хибних спрацьовувань.

Метод Random Oversampling через просте дублювання прикладів міноритарного класу призводить до переобучення – модель запам’ятовує конкретні приклади замість навчання загальним закономірностям.

Метод SMOTE показав найкращі результати. Метод створює синтетичні приклади міноритарного класу шляхом інтерполяції між існуючими сусідніми прикладами у просторі ознак. Ключовий параметр – цільове співвідношення класів після oversampling.

Було протестовано три варіанти:

– 1:3 (1 конверсія : 3 не-конверсії), найкращий баланс між Precision та Recall;

– 1:2 (вищий Precision, але нижчий Recall);

– 1:1 (повний баланс класів, але надмірне зміщення до міноритарного класу).

Метод Class weights – простий та швидкий, що не вимагає зміни даних. Встановлення параметра class_weight=«balanced» автоматично обчислює ваги обернено пропорційно до частоти класів. Однак цей метод менш ефективний порівняно зі SMOTE.

Комбінований підхід (SMOTE + Class weights) показав найкращий результат, досягнувши ROC-AUC = 0,793 на валідаційній вибірці.

На основі експериментів було обрано SMOTE з співвідношенням 1:3 + class_weight зі значенням «balanced» як оптимальну стратегію з наступних міркувань:

– баланс метрик, F1-Score = 0,327 значно перевищує baseline (0,00) та інші методи;

– ROC-AUC, 0,793 демонструє хорошу здатність моделі розрізняти класи;

– Recall, 67,8% означає, що модель виявляє 2/3 всіх конверсій;

– Precision, 21,7% є прийнятним для арбітражу трафіку, де важливіше не пропустити конверсію;

– час навчання, 9,5 секунд є прийнятним для практичного застосування

При застосуванні SMOTE враховувалися параметри, наведені на рисунку 3.19.

Параметр	Значення	Призначення
sampling_strategy	0.33	Цільове співвідношення класів 1:3
k_neighbors	5	Кількість найближчих сусідів для інтерполяції
random_state	42	Відтворюваність результатів
n_jobs	-1	Використання всіх доступних ядер процесора

Рисунок 3.19 – Параметри алгоритму SMOTE

Розроблено процес генерації синтетичних прикладів. Для кожного прикладу міноритарного класу SMOTE:

– знаходить $k = 5$ найближчих сусідів того ж класу в просторі ознак;

– випадково обирає один з цих сусідів;

– створює новий синтетичний приклад шляхом інтерполяції:

$$x\_new = x\_original + \lambda \cdot (x\_neighbor - x\_original),$$

де λ – випадкове число з діапазону [0, 1];

– повторює процес до досягнення цільового співвідношення класів.

Статистика застосування SMOTE наведена на рисунку 3.20.

Етап	Клас 0 (No Conv)	Клас 1 (Conv)	Співвідношення	Загальний розмір
До SMOTE	30,595 (96.84%)	1,000 (3.16%)	30.6:1	31,595
Після SMOTE	30,595 (75.07%)	10,165 (24.93%)	3.0:1	40,760
Синтезовано	0	9,165	-	+9,165

Рисунок 3.20 – Зміна розподілу класів після SMOTE

Для перевірки якості згенерованих SMOTE прикладів було проведено порівняльний аналіз розподілів ознак, наведений на рисунку 3.21.

Ознака	KS-статистика	p-value	Висновок
cost	0.047	0.342	Розподіли схожі
hour	0.053	0.289	Розподіли схожі
historical_cr	0.038	0.512	Розподіли схожі
segment_volume	0.062	0.187	Розподіли схожі
country_encoded	0.071	0.098	Розподіли схожі

Рисунок 3.21 – Порівняння розподілів реальних та синтетичних прикладів

Застосування методу SMOTE до наявного набору даних дозволило суттєво покращити баланс класів.

Тест Колмогорова-Смирнова показав, що розподіли синтетичних даних статистично не відрізняються від реальних (p > 0,05 для всіх ознак), що підтверджує якість генерації.

Є важливі застереження.

Застосування тільки до навчальної вибірки – SMOTE застосовувався виключно до навчальної вибірки. Валідаційна та тестова вибірки залишалися незмінними з оригінальним розподілом класів для об'єктивної оцінки якості моделі.

Обмеження SMOTE – метод може створювати шум, якщо класи сильно перекриваються у просторі ознак. Для мінімізації цього ефекту використовувалася модифікація Borderline-SMOTE, яка генерує синтетичні приклади переважно на межі між класами.

Різні алгоритми машинного навчання по-різному реагують на SMOTE (рисунок 3.22).

Модель	Без SMOTE	З SMOTE	Абсолютний приріст	Відносний приріст
Logistic Regression	0.689	0.793	+0.104	+15.1%
Random Forest	0.798	0.812	+0.014	+1.8%
XGBoost	0.839	0.847	+0.008	+1.0%
SVM	0.712	0.781	+0.069	+9.7%

Рисунок 3.22 – Вплив SMOTE на різні моделі (ROC-AUC)

Tree-based моделі (Random Forest, XGBoost) менш чутливі до дисбаланса класів завдяки вбудованим механізмам, тоді як лінійні моделі (Logistic Regression, SVM) значно виграють від SMOTE.

При роботі з незбалансованими даними accuracy є неінформативною метрикою. Використовувалися наступні альтернативи:

– Метод Precision-Recall AUC краще відображає якість для незбалансованих класів;

– Метод Matthews Correlation Coefficient (MCC) враховує всі значення confusion matrix;

– Метод Balanced Accuracy середнє між Recall для обох класів;

– Метод Cohen’s Kappa міра згоди з урахуванням випадковості.

Всі ці метрики підтверджують ефективність обраної стратегії обробки незбалансованості класів.

3.3 Перспективи подальшої роботи

Проведене дослідження виявило ряд перспективних напрямків для подальшого розвитку системи багатовимірного аналізу та моніторингу рекламних кампаній.

Застосування глибоких нейронних мереж, зокрема архітектур на базі трансформерів, може дозволити виявляти більш складні нелінійні залежності у даних. Перспективним є використання архітектури TabNet, спеціально розробленої для табличних даних, яка показує результати на рівні gradient boosting при збереженні інтерпретовності через механізм уваги (attention mechanism).

Поточна реалізація вимагає повного перенавчання моделей при надходженні нових даних. Впровадження методів online learning дозволить моделям адаптуватися до змін у реальному часі без необхідності зберігання всієї історії даних. Особливо перспективними є алгоритми на базі Stochastic Gradient Descent з adaptive learning rate (Adam, RMSprop) та методи incremental learning для Random Forest.

Наразі система оптимізує розподіл бюджету за єдиним критерієм (ROI або кількість конверсій). Впровадження методів багатокритеріальної оптимізації (Pareto optimization, NSGA-II) дозволить одночасно враховувати множину цілей: максимізація конверсій, мінімізація CPA, контроль якості трафіку, дотримання обмежень на бюджет окремих сегментів.

Додавання модулів прогнозування майбутніх показників кампаній на основі часових рядів (ARIMA, Prophet, LSTM) дозволить планувати бюджети та

передбачати результати. Особливо корисним буде прогнозування сезонності, трендів та аномалій у поведінці аудиторії.

Аналіз ефективності рекламних креативів та автоматичне генерування рекомендацій щодо оптимальних комбінацій зображень, текстів та CTA для різних сегментів аудиторії. Використання computer vision для аналізу візуальних елементів креативів та NLP для аналізу текстових компонентів.

Розроблення спеціалізованих моделей для автоматичного виявлення ботів, кліків з фарм та інших видів неякісного трафіку на основі поведінкових патернів, аномалій у часових розподілах та географічних характеристиках.

Наразі система працює з експортованими даними. Повна інтеграція з Facebook Ads API, Google Ads API, TikTok Ads API дозволить автоматично:

- збирати дані у реальному часі;
- автоматично коригувати ставки та бюджети;
- створювати та модифікувати аудиторії (lookalike, custom audiences);
- вмикати/вимикати неефективні рекламні групи.

Впровадження системи безперервного A/B тестування (multi-armed bandit approach) для автоматичного експериментування з різними стратегіями таргетингу, креативами та розподілом бюджету. Система повинна автоматично виділяти трафік на експерименти, оцінювати статистичну значущість результатів та впроваджувати переможні стратегії.

Підключення до CRM систем клієнтів дозволить отримувати інформацію про повний lifecycle користувачів, розраховувати реальний LTV та оптимізувати кампанії не тільки за первинними конверсіями, але й за довгостроковою цінністю залучених користувачів.

Для роботи з великими обсягами даних (мільйони кліків на день) можливо перейти на розподілені обчислювальні фреймворки:

- Apache Spark для обробки великих датасетів;
- Dask для паралелізації операцій у Python;
- Ray для розподіленого машинного навчання.

ВИСНОВКИ

Таким чином, у кваліфікаційній роботі досліджено методи багатовимірного аналізу для моніторингу рекламних кампаній у соціальних мережах та вирішено такі завдання:

– проведено аналіз сучасних методів багатовимірного статистичного аналізу та машинного навчання для задач цифрового маркетингу, що дало можливість детально вивчити їх недоліки та переваги для побудови ефективної системи моніторингу рекламних кампаній;

– вивчено та проведено аналіз літературних джерел щодо застосування методів кластеризації, класифікації та зниження розмірності у контексті аналізу рекламного трафіку, що дало можливість виявити сучасний стан дослідженої проблематики, недоліки та переваги існуючих підходів;

– розроблено математичні моделі для основних етапів аналізу даних рекламних кампаній, включаючи моделі попередньої обробки даних, кластеризації аудиторії, класифікації конверсій та оптимізації розподілу бюджету, що дало можливість формалізувати процес прийняття рішень при управлінні кампаніями;

– сформовано покрокові алгоритми для кожного з етапів багатовимірного аналізу та візуалізовано їх у вигляді блок-схем, що дало можливість розробити програмну реалізацію системи моніторингу;

– проведено експериментальне дослідження на реальному наборі даних з 47,283 записів про кліки користувачів, зібраних трекінговою системою Keitaro протягом 30-денної кампанії, що дозволило перевірити ефективність розроблених методів на практичних даних;

– виконано порівняльний аналіз методів кластеризації (k-means, ієрархічна кластеризація, DBSCAN) за критеріями силует-коефіцієнта та індексу Девіса-Болдіна, що дозволило обрати оптимальний метод k-means з k = 5 кластерами (silhouette score = 0,342).

У рамках кваліфікаційної роботи проведено дослідження методів k-means, Random Forest та XGBoost шляхом програмної реалізації застосунків, котрі надали можливість протестувати розглянуті методи на предмет ефективності аналізу та прогнозування результатів рекламних кампаній. Для кожного з методів побудовано та візуалізовано покрокові алгоритми за допомогою блок-схем, створено схеми архітектури системи та структури бази даних. Створено набір даних з реального трафіку Facebook-кампаній, що включає 47283 записи з 38 атрибутами, оброблено дані та виділено 127 ознак для моделювання після застосування методів feature engineering та кодування категоріальних змінних. Проведено тестування застосунків та проаналізовано результати ефективності для кожного із розглянутих методів. На основі результатів доведено перевагу моделі XGBoost в точності прогнозування (ROC-AUC = 0,847) та методу k-means для сегментації аудиторії шляхом порівняння результатів тестування усіх досліджуваних методів.Розроблено вебінтерфейс для системи аналізу, за допомогою котрого користувач має змогу завантажувати дані, проводити кластеризацію, навчати моделі класифікації, візуалізувати результати та експортувати звіти.

Наукова новизна роботи полягає у комплексному підході до аналізу ефективності рекламних кампаній, що поєднує методи кластеризації, класифікації та зниження розмірності з урахуванням специфіки арбітражу трафіку у соціальних мережах, зокрема обробки змішаних типів даних, незбалансованості класів та високої розмірності простору ознак. Практична цінність результатів підтверджена успішним A/B тестуванням на реальній кампанії з доведеним економічним ефектом $34200/рік при місячному бюджеті $30000. Результати роботи можуть бути використані арбітражниками трафіку для підвищення ефективності управління кампаніями, маркетинговими агентствами для оптимізації рекламних витрат клієнтів та рекламними платформами для надання advanced analytics користувачам.

Результати роботи апробовано у вигляді 2 тез доповідей під час II Міжнародної науково-практичної студентської конференції «ІТ-простір

сьогодення: тенденції, інновації та перспективи розвитку» [41] і IX Міжнародної науково-практичної конференції «Education and science of today: intersectoral issues and development of sciences» [42].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. New York: Springer, 2009. 745 p.

2. Cai, H., Ren, K., Zhang, W., Malialis, K., Wang, J., Yu, Y., & Guo, D. (2017, February). Real-time bidding by reinforcement learning in display advertising. In Proceedings of the tenth ACM international conference on web search and data mining (pp. 661-670).

3. James G., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning with Applications in R. 2nd ed. New York: Springer, 2021. 607 p.

4. Yang, M., Cao, Q., Tong, L., & Shi, J. (2025, April). Reinforcement learning-based optimization strategy for online advertising budget allocation. In 2025 4th International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID) (pp. 115-118). IEEE.

5. Gordon, B. R., Jerath, K., Katona, Z., Narayanan, S., Shin, J., & Wilbur, K. C. (2021). Inefficiencies in digital advertising markets. Journal of Marketing, 85(1), 7-25.

6. Xia, J., Zhang, Y., Song, J., Chen, Y., Wang, Y., & Liu, S. (2021). Revisiting dimensionality reduction techniques for visual cluster analysis: An empirical study. IEEE Transactions on Visualization and Computer Graphics, 28(1), 529-539.

7. Sousa, D., & Small, C. (2021). Joint characterization of multiscale information in high dimensional data. arXiv preprint arXiv:2102.09669. (дата звернення: 15.11.2025)

8. Dobrakowski, A. G., Pacuk, A., Sankowski, P., Mucha, M., & Brach, P. (2022, December). Improving ads-profitability using traffic-fingerprints. In Australasian Conference on Data Mining (pp. 205-216). Singapore: Springer Nature Singapore.

9. Pham, T., & Simoiu, C. (2016). Unsupervised Learning For Effective User Engagement on Social Media. arXiv preprint arXiv:1611.03894. (дата звернення: 15.11.2025)

10. Baranovska, T. (2025). Optimization of advertising campaigns using audience clustering in performance marketing.

11. Shender, D., Amini, A. N., Bao, X., Dikmen, M., Richardson, A., & Wang, J. (2020). A time to event framework for multi-touch attribution. arXiv preprint arXiv:2009.08432. (дата звернення: 15.11.2025)

12. Fahreza, R., Arfiansyah, R., & Satria, D. (2022). Cluster Analysis of Facebook Ads User For Digital Marketing Using K-Means Algorithm. IJCONSIST JOURNALS, 4(1), 27-31.

13. Kumar, S., Gupta, G., Prasad, R., Chatterjee, A., Vig, L., & Shroff, G. (2020, November). Camta: Causal attention model for multi-touch attribution. In 2020 international conference on data mining workshops (icdmw) (pp. 79-86). IEEE.

14. Кобилін, О.А., & Творошенко, І.С. (2021). Методи цифрової обробки зображень: навч. посібник. Харків: ХНУРЕ.

15. Bohdan N., Tvoroshenko I., Gorokhovatskyi V., and Kobylin O. (2025) Development of a hybrid method to enhance context memory for a chatbot application based on large language models, International Journal of Academic Information Systems Research, 9(10), pp. 7-18.

16. Suprun A., Tvoroshenko I., Gorokhovatskyi V., and Yakovleva O. (2025) Development and research of a method for the combined use of large language models for text generation, International Journal of Academic and Applied Research, 9(10), pp. 249-263.

17. Gorokhovatskyi V., Chmutov Y., Tvoroshenko I., and Kobylin O. (2025) Reducing computational costs by compressing the structural description in image classification methods, Advanced Information Systems, vol. 9, no. 1, pp. 5-12.

18. Yakovleva, O., Kovtunenko, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023, April). Face Detection for Video Surveillance-based Security System. In COLINS (3) (pp. 69-86).

19. Lyashenko, V., Lyubchenko, V., Mohammad, A., Alveera, K., & Kobylin, O. (2016). The methodology of image processing in the study of the properties of fiber as a reinforcing agent in polymer compositions.

20. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, Indonesian Journal of Electrical Engineering and Computer Science, 33 (1), 113-125.

21. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2025) Image description compression in classification structural methods, IEEE Access, vol. 13, pp. 43631-43641.

22. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, IEEE Access, vol. 12, pp. 73376-73385.

23. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). Tools for fast metric data search in structural methods for image classification, IEEE Access, vol. 10, pp. 124738-124746.

24. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2024) Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion, Computers, Materials & Continua, vol. 80, no. 2, pp. 3085-3106.

25. Yakovleva O., Matúšová S., Tvoroshenko I., and Isaiev Y. (2024) Visitor counting based on video stream analysis from surveillance cameras to solve various business problems, Verejná správa a regionálny rozvoj ekonómia, manažment a marketing, XX(1), pp. 67-87.

26. Pomazan, V., Tvoroshenko, I., and Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks, International Journal of Academic Information Systems Research, 7(7), pp. 25-36.

27. Gorokhovatskyi, V., & Tvoroshenko, I. (2024). An effective method for transforming an image description into a compact vector for classification.

28. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, Сучасні інформаційні системи, 6(3), С. 5-12.

29. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. Int. scientific symp. «Intelligent Solutions-S». Computational intelligence. Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine, 25-27.

30. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, Computers, Materials & Continua, 72(1), pp. 1785-1797.

31. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.

32. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, Сучасні інформаційні системи, 7(1), С. 5-13.

33. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, International Journal of Academic Engineering Research, 7(9), pp. 64-72.

34. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, Advances in Electrical and Electronic Engineering, 21(1), pp. 19-27.

35. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, International Journal of Academic and Applied Research, 7(9), pp. 57-70.

36. Творошенко, І.С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. Харків: ХНУРЕ.

37. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 3rd ed. O'Reilly Media, 2022. 861 p.

38. Arthur D., Vassilvitskii S. k-means++: The Advantages of Careful Seeding. Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. 2007. P. 1027-1035.

39. Ester M., Kriegel H.-P., Sander J., Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96). 1996. P. 226-231.

40. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, International Journal of Academic and Applied Research, 7(11), pp. 134-145.

41. Бойко М. (2025) Метод багатовимірного аналізу ефективності рекламних кампаній з адаптивним моніторингом, II Міжнародна науково-практична студентська конференція «IT-простір сьогодення: тенденції, інновації та перспективи розвитку», (15 жовтня, 2025), С. 327-329.

42. Бойко М. (2025) Інтеграція систем трекінгу та аналітики для автоматизованого управління рекламними кампаніями, IX Міжнародна науково-практична конференція «Education and science of today: intersectoral issues and development of sciences», (28 листопада, 2025), С. 166-168.