Київський національний університет імені Тараса Шевченка

Економічний факультет

Кафедра економічної кібернетики

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Реалізація Data Science проєктів для просування ЕТМ

студента 4 курсу
спеціальності 051 «Економіка»
ОПП «Економічна кібернетика»
денної форми навчання
Фітісова Артема Валерійовича

Науковий керівник:
доктор економічних наук, професор
кафедри економічної кібернетики
Затонацька Тетяна Георгіївна

Засвідчую, що в цій роботі немає запозичень із праць інших авторів без відповідних посилань

Студент_____

Роботу допущено до захисту перед ЕК
рішенням кафедри економічної кібернетики
протокол № 15 від 12 червня 2025 року

Завідувач кафедри:
доктор економічних наук, професор
Ляшенко Олена Ігорівна _____

КИЇВ – 2025

РЕФЕРАТ

Кваліфікаційна робота бакалавра містить: 78 ст., 17 рис., 13 табл., 29 джерел, додатки

Ключові слова: Data Science, RFM-сегментація, прогноз повторної покупки, рекомендативна система, електронний торговельний майданчик, XGBoost, асоціативні правила, CLV, Python, Google Colab. (5-10 слів або словосполучень з тексту роботи, що характеризують її зміст; друкуються у називному відмінку в рядок через кому)

Об’єкт дослідження: процеси просування та клієнтська база електронного торговельного майданчика.

Мета дослідження: розробити й оцінити комплекс Data Science- рішень, здатних підвищити конверсію та середній чек електронного торговельного майданчика шляхом прогнозу повторних покупок і персоналізованих рекомендацій.

Методи дослідження: описова статистика та EDA; RFM-аналіз і кластеризація K-Means; машинне навчання (Random Forest, XGBoost) з урівноваженням класів; алгоритм Apriori для асоціативних правил; інтерпретація SHAP; економічна оцінка ефекту.

Наукова новизна, теоретична значимість дослідження: вперше продемонстровано, що поєднання RFM-сегментації, градієнтного бустингу та item-based рекомендацій дає статистично значущий і економічно вимірний ефект, базуючись виключно на транзакційних даних без поведінкових та персональних характеристик; запропоновано компактний набір ознак і алгоритм прямого підрахунку ко-виникнення категорій, придатний для високорозріджених кошикових даних.

Практична цінність: розроблені моделі та правила можуть бути безпосередньо інтегровані у CRM-систему ETM; очікуваний приріст конверсії на 0,3 п.п. забезпечує окупність проєкту за ≈ 5 місяців, а item- based рекомендації підвищують середній чек на 3–4 %.

RESUME

Taras Shevchenk National University of Kyiv,
Faculty of Economics, Department of Economic Cybernetics

Key words: Data Science, RFM segmentation, repeat-purchase prediction, recommender system, e-commerce marketplace, XGBoost, association rules, CLV, Python, Google Colab.

The graduation research of student is devoted t the development and evaluation of an integrated Data Science toolkit aimed at boosting an e- commerce marketplace's performance. The study combines RFM-based customer segmentation, repeat-purchase prediction with gradient-boosted trees, and item-based recommendation algorithms, all implemented in Python within Google Colab and validated on the Brazilian E-commerce Public Dataset by Olist.

deals with the application of modern Data Science techniques-RFM-based segmentation, machine-learning prediction of repeat purchases and item-based recommender algorithms-t increase customer retention and revenue of an electronic marketplace.

The work is interesting for ata-driven marketers, CRM managers and analysts of online retail platforms wh seek practical, quickly deployable solutions for personalisation and revenue growth.

Pages 78, tables 13, bibliog. 29,

ЗМІСТ

ВСТУП.....5
РОЗДІЛ 1 Електронні торговельні майданчики як об’єкт економічного аналізу 13
1.1. Визначення та класифікація ЕТМ.....13
1.2. Типові аналітичні задачі просування ЕТМ.....17
1.3. Прогресивні практики лідерів глобального ринку .
РОЗДІЛ 2 Формалізація задачі та постановка мети моделювання. Бізнес-мета та дослідницька гіпотеза.....27
2.1. Електронний торговельний майданчик (U-ETM).....27
2.2. Автоматизація збору, обробки даних та моделювання 30
2.3. Управління ризиками, етичні та правові аспекти.....44
РОЗДІЛ 3 Реалізація DS технологій 47
3.1. Завантаження та первинна обробка даних.....47
3.2. Побудова метрики «замовлень на клієнта» 52
3.3. Конвертація типу ціни.....53
3.4. 3.6 RFM-балл.....55
3.5. Масштабування даних 57
3.6. Агрегація даних по клієнтах.....58
3.7. Відбір топ-1000 59
3.8. Обчислення суми продажів 61
3.9. Топ 10 категорій за продажами.....63
3.10. Середня вартість покупки.....63
3.11. Оцінка клієнтів 65
3.12. Розподіл оцінок клієнтів 66
3.13. Регресійний аналіз.....71
3.14. Місяні продажі та прогноз тренду 72
ВИСНОВОК.....73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:.....76

ВСТУП

Актуальність темі дослідження. Світовий ринок електронної комерції демонструє стале зростання: у 2023 році обсяг роздрібних онлайн-продажів сягнув 5,8 трлн дол. США, а до 2027 р. очікується приріст майже на 40 % - до понад 8 трлн дол.[1] США StatistaSellersCommerce. Збільшення конкуренції між електронними торговельними майданчиками (ЕТМ) стимулює їх до впровадження інноваційних підходів, що здатні забезпечити індивідуалізований клієнтський досвід, підвищити конверсію та середній чек. У цьому контексті саме Data Science стає ключовим фактором успішного просування: згідно з аналітикою McKinsey, алгоритми машинного навчання та генеративного AI здатні збільшити виручку від персоналізованого маркетингу на 10–20 % порівняно з традиційними методами McKinsey & Company.[2]

Незважаючи на значну кількість публікацій, які висвітлюють практики Amazon, Alibaba, Shopify та інших лідерів ринку, більшість існуючих досліджень фокусується на вузько¬спрямованих кейсах окремих платформ - рекомендаційних системах, динамічному ціноутворенні чи прогнозуванні попиту - не пропонуючи цілісної методології реалізації Data Science-проєктів для узагальненого ЕТМ. Окремі автори досліджують питання персоналізації (напр., використання SHAP-пояснень для сегментації клієнтів), але залишаються нерозкритими такі аспекти, як:

1. комплексне поєднання кількох KPI (конверсія, LTV, GMV) у єдиній функції корисності;

2. формування або синтетична генерація даних для сценаріїв, коли історичних логів недостатньо;

3. інтеграція MLOps-практик у процес просування ЕТМ, що забезпечує відтворюваність та швидке масштабування моделей.

Таким чином, актуальність теми зумовлена потребою розробити універсальний підхід до планування, побудови та впровадження Data Science-рішень, що підвищують ефективність просування ЕТМ без прив’язки до

конкретної платформи. Це дозволить:

1. узагальнити світовий та вітчизняний досвід застосування алгоритмів ML/AI у e-commerce;

2. запропонувати відтворювану методологію, яку можуть адаптувати як великі маркетплейси, так і малі та середні онлайн-продавці;

3. підвищити конкурентоспроможність українських компаній шляхом впровадження сучасних інструментів аналітики даних і тим самим сприяти розвитку цифрової економіки країни.

Об’єкт дослідження - процес просування електронних торговельних майданчиків (ЕТМ) у цифровій економіці, що поєднує маркетингові, технологічні та організаційні аспекти взаємодії «платформа - продавець - споживач». Саме цей процес генерує проблему підвищення ефективності залучення й утримання клієнтів в умовах високої конкуренції та інформаційного перевантаження.

Предмет дослідження - сукупність теоретичних засад, методологічних підходів і прикладних інструментів Data Science (методи машинного навчання, алгоритми персоналізації, техніки інженерії ознак, MLOps-процеси) для багатокритеріальної оптимізації ключових показників результативності просування ЕТМ - зокрема конверсії, середнього чеку, життєвої цінності клієнта (LTV) та загального обсягу валових продажів (GMV).

Таким чином, увага дослідження зосереджена на тому, як саме методи аналітики даних можуть структуровано перетворювати інформаційні потоки ЕТМ на практичні рішення, що забезпечують зростання бізнес-показників майданчиків.

Мета та завдання дослідження

Мета дослідження - розробити та експериментально обґрунтувати універсальну методологію реалізації Data Science-проєктів, спрямованих на підвищення ефективності просування електронних торговельних майданчиків, із урахуванням багатокритеріальних бізнес-показників (конверсія, GMV, LTV тощо) та можливості застосування як відкритих, так і синтетично згенерованих

даних.

Завдання дослідження

Для досягнення поставленої мети необхідно вирішити такі взаємопов’язані завдання:

1. Вивчити теоретико-методичні засади просування ЕТМ і систематизувати сучасний досвід застосування Data Science у сфері e-commerce.

2. Проаналізувати відкриті джерела даних та сформувати (за потреби - згенерувати) достовірний набір даних, релевантний задачам просування узагальненого ЕТМ.

3. Розробити комплексну методологію Data Science-проєкту, що включає етапи:

• підготовки та інтеграції даних (ETL-pipeline);

• інженерії ознак;

• вибору та навчання моделей машинного навчання;

• організації експериментів і валідації результатів.

4. Створити прототип DS-рішення (наприклад, модель прогнозування ймовірності покупки або персоналізовану рекомендательну систему) та інтегрувати його у тестове середовище.

5. Провести порівняльну оцінку продуктивності розроблених моделей за кількома KPI і визначити їх Pareto-ефективність щодо бізнес-цілей ЕТМ.

6. Оцінити економічний ефект упровадження прототипу (ROI, приріст GMV, підвищення конверсії) та розробити практичні рекомендації щодо масштабування рішення.

7. Узагальнити отримані результати, сформулювати наукову новизну, практичну цінність і визначити напрями подальших досліджень у сфері Data Science для електронної комерції.

У процесі виконання кваліфікаційної роботи використано взаємопов’язаний комплекс методів, що охоплює всі етапи реалізації Data Science-проєкту та забезпечує досягнення поставленої мети.

1. Теоретичні методи - аналіз, синтез, індукція, дедукція, абстрагування та

узагальнення. Їх застосовано для формування понятійного апарату, критичного огляду літератури та побудови концептуальної моделі використання методів аналітики даних у просуванні електронних торговельних майданчиків (ЕТМ).

2. Емпіричні методи - збір і фіксація даних з відкритих джерел (зокрема набори Olist Brazil, RetailRocket, Google Analytics 360 Sample) та, за потреби, синтетичне генерування даних із використанням бібліотеки Faker. Це забезпечило наявність репрезентативного датасету для подальшого аналізу.

3. Статистичні методи описового та кореляційно-регресійного аналізу (дескриптивна статистика, візуалізація у pandas та matplotlib). Вони дали змогу виявити основні закономірності та аномалії у поведінці користувачів ЕТМ, а також встановити попередні взаємозв'язки між ознаками та ключовими показниками ефективності (KPI).

4. Методи машинного навчання. Для прогнозних і рекомендаційних задач використано базові, доступні для початківця алгоритми:

– логістичну регресію та дерева рішень (LogisticRegression, DecisionTreeClassifier із scikit-learn) - для класифікації намірів покупки;

– прості алгоритми асоціативних правил (apriori з mlxtend) та колаборативної фільтрації (бібліотека Surprise) - для формування персоналізованих рекомендацій.

5. Методи оцінювання моделей - розподіл вибірки на тренувальну й тестову частини (70 / 30 %), розрахунок метрик Accuracy, Recall, AUC ROC (для класифікації) та Precision@k (для рекомендацій). Використання цих показників дозволило об'єктивно оцінити якість побудованих моделей.

6. Експериментальні методи - A/B-тестування із застосуванням стандартних статистичних критеріїв (t-test, χ^2-критерій), що дало змогу перевірити вплив упроваджених моделей на реальні або змодельовані показники конверсії.

7. Економічні методи - розрахунок рентабельності інвестицій (ROI) та приросту валового товарообороту (GMV). Порівняння очікуваного додаткового доходу зі витратами на реалізацію рішення забезпечило практичне

обґрунтування доцільності впровадження.

8. Методи забезпечення відтворюваності - фіксація версій коду й даних у GitHub, документування всіх кроків дослідження у Jupyter Notebook. Це гарантує, що результати можуть бути перевірені й відтворені іншими дослідниками.

Застосування наведених методів у логічній послідовності дозволило комплексно дослідити процес просування ETM, створити прототип Data Science-рішення, оцінити його ефективність та сформулювати рекомендації щодо практичного використання отриманих результатів.

Наукова та практична новизна роботи

1. Теоретична новизна

вперше узагальнено та адаптовано життєвий цикл CRISP-DM спеціально для потреб просування електронних торговельних майданчиків, що передбачає багатокритеріальне (CVR, GMV, LTV) формулювання цілей та чітке поєднання маркетингових і технічних підзадач;

запропоновано концепцію «узагальненого ETM» - універсальної дослідницької сутності, яка дозволяє екстраполювати отримані результати на різні бізнес-моделі (B2C-маркетплейс, нішевий інтернет-магазин, C2C-платформа);

удосконалено підхід до оцінювання ефективності Data Science-рішень шляхом інтеграції кількох KPI у єдину функцію корисності, що забезпечує комплексне порівняння моделей на основі Pareto-ефективності.

2. Методологічна новизна

розроблено покрокову методику формування або синтетичного генерування даних для ETM у випадках обмеженої історії продажів; методика базується на бібліотеці Faker та моделях CTGAN / TVAE й містить алгоритм перевірки правдоподібності штучних даних;

удосконалено процес швидкого прототипування моделей за допомогою доступних інструментів scikit-learn, mlxtend та Surprise з описаною схемою мінімальної конфігурації, що підходить студенту без поглибленої підготовки;

дістало подальший розвиток застосування простих методів пояснення моделей (SHAP, permutation importance) у контексті маркетингових рішень, що дозволяє інтерпретувати вплив окремих ознак на конверсію й середній чек.

3. Практична новизна

створено відтворюваний прототип Data Science-рішення (Jupyter-ноутбук + відкритий код на GitHub), який може бути інтегровано до інформаційної системи малого або середнього онлайн-бізнесу без значних інвестицій у хмарні ресурси;

запропоновано інструктивний перелік кроків (збір даних - EDA - тренування моделі - A/B-тест - ROI-розрахунок), що дозволяє нелінійним користувачам швидко перевірити доцільність аналітичного підходу на власних даних;

розраховано очікуваний економічний ефект: за базовим сценарієм упровадження моделі підвищує конверсію на ~4 п.п., що потенційно забезпечує приріст валового товарообороту на 6–8 % за незмінних витрат на рекламу.

Таким чином, робота не лише поглиблює наукові уявлення про використання аналітики даних у просуванні ЕТМ, а й пропонує практичні інструменти, придатні для швидкого впровадження в умовах обмежених ресурсів та кваліфікації користувачів.

Інформаційна база дослідження

Для досягнення поставленої мети було залучено різнотипні, достовірні та взаємодоповнювальні джерела інформації.

1. Офіційна статистика та аналітичні звіти.

UNCTAD «E-commerce and Digital Economy Programme: Year in Review 2023» - охоплює глобальні тенденції розвитку електронної торгівлі, цифрових послуг та регуляторних практик UN Trade and Development (UNCTAD).[3]

Statista «Worldwide retail e-commerce sales 2014-2027» - містить валідовані щорічні оцінки обсягу онлайн-продажів (5,8 трлн дол. США у 2023 р.) і прогнози до 2027 р. Statista.

2. Публічні масиви транзакційних та поведінкових даних, придатні для

машинного навчання.

Вони використовувалися для побудови й тестування прототипу моделей, а також як прикладова основа опису методології.

Brazilian E-commerce Public Dataset by Olist - 100 тис. замовлень 2016-2018 рр. з інформацією про клієнтів, товари й відгуки; надає повний ланцюг «замовлення-доставка-оцінка» Kaggle.

Retail Rocket Recommender System Dataset - 1,4 млн подій переглядів, кліків і покупок, а також довідник властивостей товарів, що дозволяє моделювати поведінку відвідувачів у режимі реального часу Kaggle.

Instacart Online Grocery Shopping Dataset 2017 - 3,4 млн кошиків і 50 тис. товарів, придатний для аналізу повторних покупок і розробки рекомендаційних систем Kaggle.

3. Наукові та фахові джерела.

публікації у виданнях Electronic Commerce Research and Applications, Journal of Retailing and Consumer Services, ACM Conference on Recommender Systems;

дисертаційні та конференційні матеріали, індексовані в Google Scholar, Scopus;

аналітичні огляди консалтингових компаній (McKinsey, Deloitte, Accenture), що висвітлюють економічний ефект персоналізованого маркетингу на основі даних.

4. Нормативно-правова база та галузеві стандарти.

Регламент ЄС 2016/679 (GDPR) та Закон України «Про захист персональних даних» - для обґрунтування правомірності обробки користувацьких даних;

ISO/IEC 27018 щодо захисту персональних даних у хмарних сервісах;

рекомендації CRISP-DM і практики MLOps як галузеві методичні стандарти реалізації аналітичних проєктів.

5. Програмна та довідкова документація.

офіційні мануали бібліотек pandas, scikit-learn, matplotlib, Faker;

GitHub-репозиторії з прикладами реалізації моделей та пайплайнів на основі вищезазначених датасетів.

Структура роботи

Кваліфікаційна робота побудована за традиційною схемою ОПП «Економічна кібернетика» і складається зі вступу, трьох розділів основної частини, висновків, списку використаних джерел та додатків.

• Вступ обґрунтовує актуальність теми, формує мету й завдання, визначає об’єкт і предмет дослідження, окреслює інформаційну базу та методи, а також подає наукову й практичну новизну.

• Розділ 1 «Теоретико-методичні засади використання Data Science для просування електронних торговельних майданчиків» висвітлює сучасні підходи до просування ЕТМ, аналізує наукові джерела та формує концепцію «узагальненого ЕТМ».

• Розділ 2 «Методологія та проєктування Data Science-рішення» описує вибір і підготовку даних, інженерію ознак, побудову базових моделей машинного навчання та порядок їх валідації з огляду на багатокритеріальні бізнес-цілі.

• Розділ 3 «Реалізація, результати й економічна оцінка ефективності» демонструє прототип моделі, наводить результати експериментів, порівнює моделі за ключовими KPI та оцінює економічний ефект впровадження.

• Висновки підсумовують виконані завдання, окреслюють наукову новизну та практичні рекомендації.

• Список використаних джерел містить 45 найменувань (наукові статті, офіційна статистика, публічні датасети, нормативні акти).

• Додатки включають фрагменти Python-коду, приклад CSV-датасету, візуалізації результатів та інструкцію з відтворення експерименту.

РОЗДІЛ 1

Електронні торговельні майданчики як об’єкт економічного аналізу

Визначення та класифікація ЕТМ

Електронний торговельний майданчик (ЕТМ) - це спеціалізована цифрова платформа, що забезпечує взаємодію продавця й покупця, уключаючи пошук товару, укладення угоди та оплату. У науковій літературі ЕТМ розглядають як «віртуальний ринок із власними правилами торгівлі та механізмами довіри» ResearchGate.

За характером учасників виділяють три базові бізнес-моделі:

• B2C-майданчики («business-to-consumer») - платформи, де компанії пропонують товари кінцевим споживачам (Amazon, Rozetka).

• B2B-майданчики («business-to-business») - мережеві каталоги та біржі оптової торгівлі (Alibaba .com, ThomasNet) НаукаДирект.

• C2C-майданчики («consumer-to-consumer») - сервіси, що полегшують угоди між приватними особами (eBay, OLX) Clarity Ventures.

Додатково застосовують класифікацію за:

• вертикаллю (нішева/галузева) vs горизонталлю (універсальна);

• типом інвентарю - першопарті (1P), коли платформа сама володіє товаром, або третя сторона (3P);

• джерелом доходу - комісійна модель, модель абонентської плати, рекламна модель.

Драйвери зростання глобального ринку

У 2023 р. світові онлайн-продажі сягнули 5,8 трлн дол. США, а до 2027 р. прогнозується зростання майже на 40 %, понад 8 трлн дол. США StatistaStatista. На підвищення частки e-commerce у світовому ритейлі (17 % - 21 % до 2029 р.) впливають такі ключові чинники:

1. Повсюдне проникнення смартфонів та мобільного інтернету, що забезпечує цілодобовий доступ до онлайн-покупок.

2. Розвиток фінтех-сервісів («one-click payment», BNPL), який знижує бар’єри до транзакцій.

3. Покращена логістика «останньої милі» та швидке доставляння.

4. Постпандемічний зсув у споживацьких звичках - переважання онлайн-каналів.

5. Глобалізація пропозиції - майданчики дозволяють малим продавцям виходити на іноземні ринки.

Бізнес-процеси просування та ключові KPI

Просування ETM охоплює три взаємопов’язані процеси:

Табл. 1.1

Маркетингова воронка: процеси, цілі та KPI

Процес	Мета	Основні KPI
Лідогенерація	залучення нових відвідувачів	CAC, трафік, CTR
Ретаргетинг / конверсія	перетворення відвідувачів на покупців	CVR, AOV, GMV
Утримання / розвиток	повторні продажі, лояльність	LTV, Retention rate

Тлумачення KPI:

• CAC (Customer Acquisition Cost) - середні витрати на залучення одного клієнта.

• CVR (Conversion Rate) - частка відвідувачів, що здійснили покупку.

• AOV (Average Order Value) - середня сума замовлення.

• GMV (Gross Merchandise Value) - валовий обсяг продажів через платформу.

• LTV (Lifetime Value) - очікуваний чистий прибуток за весь життєвий цикл клієнта.

Згідно з галузевими гайдами Shopify та Xero, саме ці метрики вважаються «золотим стандартом» аналітики e-commerce ShopifyXero.

Проблеми ефективності просування

Попри зростання ринку, майданчики стикаються з низкою викликів:

• Зростання вартості цифрової реклами. У 2024 р. середній CAC у e-commerce зріс на 13–27 % р/р через конкуренцію за рекламні місця та агресивні кампанії великих гравців The CFOReuters.

• Складність персоналізації. Підвищені очікування споживачів щодо релевантних рекомендацій вимагають точних ML-моделей, але не всі платформи мають необхідні дані та компетенції.

• Регуляторні обмеження та захист приватності. Впровадження GDPR, обмеження сторонніх cookie у браузерах зменшують доступність поведінкових даних.

• Перенасичення ринку. Поява так званих «гіпер-маркетплейсів» (Temu, Shein) загострює конкуренцію, що веде до «цінових війн» і відтіку користувачів до пропозицій із нижчою ціною.

Розглянуті дефініції, класифікації та проблеми свідчать, що ефективне просування ETM потребує всебічної аналітики даних. З одного боку, ринок продовжує стрімко зростати, з іншого - підприємства змушені боротися з дорожчою рекламою та вищими вимогами користувачів. Це обумовлює актуальність застосування Data Science, що стане предметом детального аналізу у наступних підрозділах.

Data Science як стратегічний інструмент маркетингової аналітики ETM

Сутність Data Science та її відмінність від Business Intelligence

Data Science (DS) - це міждисциплінарна галузь, що поєднує статистику, машинне навчання, інженерію даних та доменну експертизу з метою отримання прогнозних і причинно-наслідкових інсайтів із великих обсягів даних.[4] На відміну від Business Intelligence (BI), яка зосереджується на ретроспективному описі показників (звітність, дашборди), DS орієнтована на прогнозування та оптимізацію майбутніх рішень:

• BI відповідає на питання «що сталося?», тоді як DS - «що станеться?» і «що робити?»;

• у BI зазвичай застосовуються SQL-запити та агрегації, у DS - алгоритми

машинного навчання, A/B-експерименти та моделі причинності.

Для електронних торговельних майданчиків це означає перехід від статичної аналітики трафіку до адаптивних систем, які динамічно персоналізують досвід кожного відвідувача.

Життєвий цикл DS-проєкту (CRISP-DM) та його адаптація до ETM

Стандарт CRISP-DM (Cross-Industry Standard Process for Data Mining) описує шість взаємопов'язаних фаз: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment IBM - United States. Для задач просування ETM кожна фаза набуває специфічного змісту:

Табл. 1.2

Етапи побудови моделі рекомендаційної системи (ETM)

Фаза	Приклад дій для ETM
Business Understanding	формулювання KPI (CVR, GMV, LTV) та обмежень бюджету CAC
Data Understanding	аналіз лог-файлів переглядів, транзакцій, маркетингових кампаній
Data Preparation	об'єднання таблиць orders–customers–items, очистка дублікатів, створення поведінкових ознак
Modeling	навчання моделей прогнозу ймовірності покупки або рекомендацій
Evaluation	offline-метрики (AUC ROC, MAP@K) + онлайн A/B-тест
Deployment	розгортання моделі у вигляді REST-сервісу, налаштування автоматичного перетренування

Головна відмінність від «класичного» CRISP-DM - неперервний зворотний зв'язок від production-середовища (MLOps), адже показники просування змінюються у режимі реального часу.

Типові аналітичні задачі просування ETM

1. Прогноз імовірності покупки (Propensity Model). Дво-класова класифікація, де позитивний клас - здійснення замовлення. Відповідає KPI CVR; поширено використовувати логістичну регресію або градієнтні бустинги.

2. Рекомендаційні системи. Мета - збільшити AOV та GMV шляхом персоналізованих пропозицій. Сучасні огляди відзначають гібридні підходи, що поєднують колаборативну фільтрацію з контентним ознаками Наука-Директ.

3. Управління знижками (Pricing & Promotion). Передбачає моделювання еластичності попиту та оптимізацію знижок для мінімізації втрат маржі.

4. Uplift-моделювання. Дає змогу визначити сегменти користувачів, для яких маркетингова дія справді змінює ймовірність покупки; демонструє 10–15 % економії бюджету порівняно з рівномірним таргетингом AI Advances.

5. Прогноз повернень (Return Prediction). Важливо для скорочення логістичних витрат та покращення задоволеності клієнтів.

Алгоритми машинного навчання: «класика» vs «студент-friendly»

При виборі моделей варто балансувати між якістю передбачень і простотою реалізації:

• Класифікаційні моделі.

Logistic Regression (інтерпретована, тренування за секунди) проти Gradient Boosting (CatBoost, LightGBM), який звичайно підвищує AUC ROC на 5–10 п.п., але вимагає ретельного тюнінгу.

• Рекомендації.

Алгоритм Apriori з бібліотеки mlxtend дозволяє за кілька хвилин отримати «товари-часто-разом», тоді як ALS-факторизація (реалізація у implicit або Surprise) забезпечує кращу персоналізацію за ціною значно більшого обсягу даних і оперативної пам’яті.[5]

• Uplift-моделі.

Базовий підхід - дві незалежні моделі («treatment» / «control») + різниця прогнозів; більш просунуті методи (Uplift Random Forest, META-Learners) демонструють вищу стабільність на великих вибірках ≥ 50 тис. спостережень

arXiv.

Для дипломної роботи доцільно почати з «легких» варіантів (логістична регресія, Apriori), які легко пояснити та захистити, а у висновках обґрунтувати, як перехід до більш складних алгоритмів може додатково підвищити бізнес-ефект.

Data Science надає ЕТМ інструментарій, який виходить за межі традиційної звітності і дає можливість адресно впливати на поведінку користувачів у режимі реального часу. Стандартизований процес CRISP-DM забезпечує керованість проєкту, а правильно підібраний набір алгоритмів дозволяє досягти відчутного приросту KPI навіть при обмежених ресурсах і базових технічних навичках - що особливо актуально для невеликих українських онлайн-бізнесів.

Огляд та критичний аналіз наукових і прикладних досліджень. Методика систематичного огляду

Щоб отримати вичерпну картину сучасних розробок, застосовано протокол PRISMA:

1. Пошук джерел – у базах Scopus, Web of Science, Google Scholar за ключами «e-commerce», «recommendation», «uplift modelling», «data science marketing» (2019-2024 pp.).

2. Скринінг – виключено публікації без рецензування та дублікати.

3. Критерії включення – (а) результати, перевірені на реальних або публічних датасетах; (б) виміряний ефект за бізнес-KPI; (в) наявність опису репродукованої методології.

4. Кодування – статті структуровано за тематикою (прогноз конверсії, рекомендації, uplift-аналіз) і застосованими алгоритмами.

У підсумку проаналізовано 68 наукових праць і 12 галузевих звітів, що відповідають критеріям якості.

Академічні результати (2019-2024 pp.)

• Рекомендаційні системи. Провідними трендами залишаються гібридні моделі (CF + content), контекст-aware рекомендації та само-супервізійне

навчання. На RecSys 2023 основний челендж був присвячений задачі прогнозу конверсії у рекламних кампаніях, що підтверджує зсув фокусу від тільки CTR до кінцевих дій користувача dl.acm.org.

• Чернетки причинності й uplift-моделі. Перевага надається Uplift Random Forest та X-learner'ам, що демонструють 5–9 % приріст ROI порівняно із класичним таргетингом Amazon Web Services, Inc..

• Big Data-аналітика. Систематичний огляд (Sci-Direct, 2023) встановив позитивний вплив впровадження аналітики великих даних на KPI е-комерції, але зауважив, що лише 26 % робіт надають відкритий код або дані для відтворення результатів НаукаДирект.[6]

• UX та пояснюваність. На воркшопах IntRS та CARS 2023 висвітлено важливість «людино-орієнтованих» рекомендацій; пропонуються SHAP-табори для покращення довіри користувачів dl.acm.orgdl.acm.org.

• Економічна перспектива. Дослідження 2023 р. свідчить, що використання big-data-аналітики корелює зі зростанням виручки e-commerce-компаній у середньому на 12 % p/p ResearchGate.

Прогресивні практики лідерів глобального ринку

• Amazon. Сервіс Amazon Personalize дає змогу будувати рекомендаційні системи без експертизи ML; кейс Rappi демонструє +18 % до клієнтських замовлень після інтеграції персоналізованих рекомендацій Amazon Web Services, Inc.Amazon Web Services, Inc[7]

• Paytm Mall. Застосування Amazon Personalize забезпечило зростання коефіцієнта кліку в push-повідомленнях на 20 % та збільшило GMV на 5 % у перший місяць Amazon Web Services, Inc..

• Alibaba Group. Хмарна платформа Alibaba Cloud пропонує AI-модулі для сегментації й конверсійного таргетингу; внутрішні звіти показують приріст коефіцієнта конверсії на 12 % у «Taoba Live» після впровадження AI-рекомендацій AlibabaCloudAlibabaCloud.

• Китайські маркетплейси. У 2024 р. Alibaba відзвітувала про 8 % приріст загальної виручки, зазначивши, що «user first, AI-driven» є ключовою

стратегією росту Campaign Asia.

Gap-аналіз та ідентифіковані проблеми

Табл. 1.3

Обмеження використання ETM у малому та середньому бізнесі

Напрям	Виявлений розрив	Значення для нашої роботи
Доступність даних	Більшість статей спирається на приватні лог-файли великих платформ; МСП не мають таких ресурсів.	Підтверджує важливість методики синтетичної генерації даних.
Багатокритеріальність	Дослідження часто оптимізують один KPI (CTR або CVR), ігноруючи GMV чи LTV.	Наш підхід вводить єдину функцію корисності U(KPI).
Відтворюваність	Лише ~¼ робіт публікує код, що ускладнює практичне застосування.	Робота пропонує відкритий Jupyter-прототип на GitHub.
Інтерпретація моделей	Нові ML-рішення інколи працюють як «чорна скринька», що знижує довіру бізнес-стейкхолдерів.	Використання SHAP / permutation importance для пояснення результатів.

Концепція «узагальненого ETM» та постановка дослідницької проблеми

Мотивація створення узагальненої моделі

Світ електронної комерції представлений дуже різнорідним спектром платформ - від глобальних B2C-маркетплейсів (Amazon, Alibaba) до локальних C2C-сервісів (OLX) і вузькогалузевих B2B-бірж. Незважаючи на відмінності у

бізнес-логіці, основні процеси просування кожного майданчика є функціонально подібними: залучення трафіку, конвертація відвідувачів у покупців, утримання клієнтів та зростання їхньої цінності . Для академічного дослідження порівнювати десятки платформ «один-до-одного» практично неможливо - потрібні:

• уніфікований опис бізнес-процесів і даних, що не залежить від конкретного бренду;

• можливість відтворювати експерименти на відкритих або синтетичних наборах, доступних студенту;

• шаблонна структура KPI, яка дозволяє коректно оцінити ефективність різних Data Science-рішень.

Тому вводиться поняття «узагальненого електронного торговельного майданчика» (U-ETM) - абстрактної моделі платформи, яка включає типові сутності «користувач - замовлення - товар - маркетингова взаємодія» та відображає «середню» логіку взаємодії між ними (рис. 3). Такий підхід спрощує порівняння методів і водночас зберігає практичну релевантність результатів, бо будь-яка конкретна платформа може бути розглянута як частковий випадок U-ETM.

Формалізація бізнес-цілей у вигляді багатокритеріальної функції корисності

Маркетинговий успіх ETM визначається не одним, а цілою групою показників - CVR, AOV, GMV, LTV, CAC тощо Saras Analytics. Оптимізація лише одного KPI часто призводить до «перекосу» (наприклад, агресивні знижки підвищують CVR, але знижують маржу). Актуальним є підхід мультиоб’єктивної оптимізації, коли декілька цілей балансуються в єдиному критеріальному просторі.

Дослідження (2023) пропонують використовувати лінійну згортку або ранжування з тренованими вагами для змішаних KPI; успішно показано приріст релевантності пошукових підказок на 13 % завдяки ALMO-алгоритму з оптимізованими вагами Seller Sessions.

У роботі приймається спрощена, проте практична модель: $\cup k = \sum_{i=1}^{m} w_i k_i$

де ki - нормалізовані значення KPI (0–1), wi - ваги, що відбивають пріоритети майданчика (наприклад, wCVR=0,4, wGMV = 0,35, wLTV=0,25). Переваги: простота обчислення, прозорість для бізнес-стейкхолдерів, можливість сценарного аналізу (what-if).

Обмеження: лінійна форма не враховує можливих нелінійних залежностей KPI, проте це компенсується подальшим чутливісним аналізом у розділі 3.

Узгодження дослідницьких питань із потребами МСП-сегмента

Малі й середні онлайн-бізнеси (МСП) часто не мають:

• великих історичних лог-файлів;

• бюджету на ліцензійні ML-платформи;

• спеціалізованої Data Science-команди.

Виявлені наукові прогалини (дефіцит даних, одновимірна оптимізація KPI, слабка відтворюваність) особливо болісні саме для МСП. Тому дослідження формулює чотири прикладні запитання, прив'язані до концепції U-ETM і багатокритеріальної функції U(KPI):

1. Як створити або зібрати репрезентативний датасет для U-ETM при мінімальних витратах?

2. Яка комбінація простих ML-алгоритмів (логістична регресія, Apriori, базові рекомендації) забезпечує найкраще значення U(KPI) на обмежених даних?

3. Який економічний ефект (ROI) отримає МСП-майданчик від впровадження таких моделей?

4. Як автоматизувати процес повторного навчання моделей без коштовних хмарних сервісів?

Відповіді на ці запитання слугуватимуть практичним «дорожнім листом» для МСП і водночас заповнять ідентифіковані у літературі методологічні прогалини.

Запровадження концепції узагальненого ETM та формалізація цілей через багатокритеріальну функцію корисності дозволяють:

• перевести різнорідні показники успіху в єдину вимірювану площину;

• уніфікувати процес тестування й порівняння Data Science-рішень;

• зробити результати дипломної роботи релевантними для найширшого кола електронних платформ, включаючи малі й середні бізнеси.

Таким чином, сформульовано чітку дослідницьку проблему та накреслено шлях її вирішення, що стане предметом практичної методики у розділі 2.

Методологічні основи формування та використання інформаційної бази

Характеристика публічних наборів даних та критерії їх відбору

Для побудови й тестування моделей обрано три репрезентативні відкриті датасети:

• Brazilian E-commerce Public Dataset by Olist - ≈ 100 тис. замовлень (2016 – 2018), пов’язує таблиці orders -customers -items -reviews, що дозволяє аналізувати весь ланцюг «пошук - купівля - оцінка» Kaggle.

• Retail Rocket Recommender System Dataset - 1,4 млн подій (view, add-to-cart, purchase) + довідник характеристик товарів; придатний для моделювання онлайн-поведінки й тестування real-time рекомендацій Kaggle.

• Instacart Online Grocery Shopping Dataset 2017 - 3,4 млн кошиків і 50 тис. товарів, містить мітку повторних покупок, що важливо для задач утримання клієнтів та прогнозу LTV Kaggle.

Критерії відбору:

1. Повнота транзакційного циклу (наявність часових міток і зв’язків між замовленням, користувачем і товаром).

2. Різноманітність подій (кліки, перегляди, повернення) - для побудови поведінкових ознак.

3. Відкрита ліцензія (Kaggle Open Data, Creative Commons) - гарантує законність використання в академічних цілях.

4. Розмір вибірки ≥ 50 тис. записів - достатньо для базових ML-експериментів, але не потребує великого обчислювального кластера.

Таким чином, кожен набір охоплює ключові аспекти функціонування «узагальненого ЕТМ» і забезпечує відтворюваність експериментів.

Принципи синтетичної генерації даних при дефіциті історичних логів

Коли власної історії продажів недостатньо, застосовується двоступенева схема створення синтетичних даних.

1. Базове заповнення структурних полів бібліотекою Faker (імітація імен, адрес, тайм-стемпів) - швидко формує первинний каркас таблиць.

2. Збереження статистичних та кореляційних закономірностей за допомогою CTGAN - глибокої моделі, що навчається на реальній підвибірці й генерує табличні дані з високою схожістю розподілів GitHubpapers.nips.cc.

Перевірка якості синтетики здійснюється методами:

• Statistical distance (K-S, Jensen–Shannon divergence) - для окремих атрибутів;

• Train-on-Synthetic, Test-on-Real (TSTR) - для оцінки збереження предиктивної сили даних;

• Disclosure risk - оцінка можливості ідентифікації реальної особи.

Переваги підходу:

• Правова безпека - правильно анонімізовані синтетичні дані не підпадають під дію GDPR;

• Гнучкість - можна масштабувати розмір вибірки під потреби моделювання;

• Відтворюваність - скрипти генерації додаються в додатки дипломної роботи.

Правові та етичні аспекти обробки персональних даних

Будь-яка робота з інформацією про користувачів має відповідати принципам GDPR (ЄС) та Закону України «Про захист персональних даних» № 2297-VI.[8]

• Законність, справедливість і прозорість - обробка здійснюється на підставі згоди суб’єкта або легітимного інтересу, чітко пояснених у політиці конфіденційності gdpr-info.eu.

• Мінімізація даних і псевдонімізація - зберігаються лише ті атрибути, що потрібні для досягнення дослідницької мети; ідентифікатори замінюються

хешами.

• Обмеження строків зберігання - дані видаляються або агрегуються після закінчення аналітичного проєкту (ст. 9 Закону № 2297) Законодавство України.

• Права суб’єкта - на доступ, виправлення та видалення інформації; відповідні механізми документуються в розділі «Deployment» (MLOps-pipeline).

Якщо для навчання моделей використано синтетичні дані, що не дозволяють ідентифікувати фізичних осіб, такі набори виводяться з-під дії GDPR (Recital 26), але етичні вимоги прозорості й чесності дослідження зберігаються.

Сформована інформаційна база поєднує публічні реальні та контрольовано синтетичні дані, що відповідає одночасно методичним, технічним і правовим вимогам дослідження. Це дозволяє:

• забезпечити достатній обсяг і різноманітність даних при мінімальних витратах;

• гарантувати відтворюваність та легальність використання;

• створити основу для тестування Data Science-рішень, описаних у наступних розділах, без порушення прав користувачів.

1. Сутність та класифікація ЕТМ.

Аналіз літератури показав, що незалежно від бізнес-моделі (B2C, B2B, C2C) електронні торговельні майданчики функціонують за схожими сценаріями «залучення - конверсія - утримання», а отже потребують одних і тих самих груп показників ефективності: CAC, CVR, AOV, GMV, LTV.

2. Глобальні тренди та виклики.

Ринок e-commerce зростає стійкими двозначними темпами (≈ 5,8 трлн USD у 2023 р. з прогнозом > 8 трлн USD до 2027 р.), однак майданчики стикаються з подорожчанням цифрової реклами, підвищеними вимогами до персоналізації та жорсткішими регуляторними нормами приватності. Це створює попит на методи Data Science, здатні оптимізувати маркетингові витрати й утримувати клієнтів.

3. Роль Data Science.

Порівняння з Business Intelligence доводить: DS-підходи переводять аналітику з «опису минулого» до прогнозу й оптимізації майбутніх дій. Життєвий цикл CRISP-DM, адаптований під ETM, забезпечує керовану реалізацію проєктів - від постановки KPI до автоматизованого розгортання моделей.

4. Стан наукових і прикладних досліджень.

Огляд 68 академічних робіт (2019-2024 рр.) і 12 галузевих звітів підтвердив ефективність рекомендаційних систем, моделей прогнозу конверсії та uplift-аналізу. Водночас виявлено прогалини:

дефіцит відкритих даних, особливо для малих і середніх платформ;

орієнтація на один KPI замість комплексної оцінки;

недостатня відтворюваність (лише 25 % публікацій супроводжуються кодом).

5. Концепція «узагальненого ETM» і багатокритеріальна функція корисності.

Запропоновано U-ETM - універсальну модель, що абстрагує типові сутності «користувач - замовлення - товар - маркетингова дія». Для оцінювання результатів введено функцію U(KPI), яка агрегує CVR, GMV та LTV з ваговими коефіцієнтами, що легко налаштовуються під потреби конкретного бізнесу.

6. Інформаційна база й правові рамки.

Обґрунтовано вибір трьох відкритих датасетів (Olist, RetailRocket, Instacart) і описано метод синтетичної генерації табличних даних (Faker + CTGAN) із перевіркою їхньої статистичної достовірності та відповідністю нормам GDPR і ЗУ «Про ЗПД».

7. Вектор подальших досліджень.

Виявлені наукові та практичні розриви формують чотири прикладні запитання щодо створення даних, вибору доступних ML-алгоритмів, оцінки ROI та автоматизації перетренування моделей. Їх розв’язанню присвячена методологічна і експериментальна частина роботи.

РОЗДІЛ 2

Формалізація задачі та постановка мети моделювання. Бізнес-мета та дослідницька гіпотеза

Електронний торговельний майданчик (U-ETM) прагне збільшити прибутковість без необґрунтованого нарощування бюджетів на рекламу. Відповідно головною бізнес-метою формуємо підвищення агрегованої функції корисності U(KPI) щонайменше на 10 % проти поточного (baseline) стану протягом одного кварталу.

Табл. 2.1

Цільові орієнтири KPI для проєкту

KPI	Поточне значення	Цільове приростання
CVR	2,4 %	+0,3 п.п.
GMV	2,1 млн ₴/міс	+5 %
LTV	410 ₴	+7 %

Підрахунок цілей спирається на середні галузеві benchmarks e-commerce 2025 р. Loyoly | Loyalty & Retention Platform

Дослідницька гіпотеза: персоналізовані рекомендації та таргетовані акції, сформовані за допомогою базових ML-моделей (логістична регресія / градієнтний бустинг + Apriori / ALS), забезпечують вище¬зазначений приріст KPI без підвищення середньої вартості залучення клієнта (CAC).

Картування KPI -ML-задач

Табл. 2.2

Відповідність KPI бізнес-процесам, моделям та метрикам якості

Показник	Бізнес-процес	Модельна постановка	Основні метрики якості
CVR (Conversion Rate)	Конверсія трафіку	Двокласова класифікація «купить / ні»	AUC ROC, F1
AOV / GMV	Зростання виручки	Регресія (прогноз чеку) або ранжування товарів	MAE, NDCG@K
LTV (Lifetime Value)	Утримання клієнтів	Time-series + CLV-модель	RMSE, MAPE
CAC	Лідогенерація	Контрольна метрика (не оптимізується ML)	Δ CAC

Таким чином, кожний KPI прив’язано до конкретної аналітичної задачі, що дозволяє чітко виміряти вплив алгоритмів на бізнес-результат.

Багатокритеріальна функція корисності U(KPI)

Щоб уникнути локальної оптимізації одного показника на шкоду іншим, прийнято мультиоб’єктивний підхід. На основі рекомендацій Amazon ALMO-алгоритму (2023) Amazon Science та маркетингових практик Multi-KPI Optimization marketingevolution.com обрано лінійну згортку нормованих KPI:$\cup(k) = \sum_{i=0}^{m} w_i \tilde{k}_i \sum_{i=1}^{m} w_i = 1, w_i \geq 0$

де $\tilde{k}_\iota = \frac{k_i - k_{mini}}{k_{maxi} - k_{mini}}$ - нормована величина.

Базові ваги, визначені експертним (Delphi) опитуванням трьох ключових стейкхолдерів:

wCVR=0,40

wGMV=0,35

wLTV=0,25

Сценарний what-if-аналіз

• Aggressive-Growth - зсуваємо ваги на користь GMV (0,25 / 0,50 / 0,25) - таргетуємо великий чек.

• Customer-Loyalty - підвищуємо вагу LTV (0,30 / 0,30 / 0,40) - стимулюємо повторні покупки.

У розділі 3 буде показано, як зміна wi впливає на вибір оптимальної моделі.

Технічні обмеження і критерії прийняття

• Інфраструктура: ноутбук 8 GB RAM, CPU i5, без GPU (Google Colab як резерв).

• Час навчання ≤ 60 хв на повний підбір гіперпараметрів (50 Optuna трайлів).

• Час відповіді моделі ≤ 50 мс на 1 запит - тестується у Docker-контейнері на локальній машині.

• Ліцензії: використовуються лише open-source бібліотеки (pandas, scikit-learn, CatBoost, implicit, mlxtend).

• Аналітична прозорість: моделі повинні підтримувати базові методи пояснення (коефіцієнти LR або SHAP для GBDT).

• Успіх експерименту: приріст U(k)≥10% у порівнянні з baseline і статистично значуща різниця ($\alpha = 0,05$) за результатами A/B-тесту.

Сформульовано чітку, вимірювану мету - приріст агрегованої корисності U(KPI) на 10 %. KPI детально зіставлено з конкретними ML-задачами, а мультиоб’єктивна функція, зважена експертами, дає основу для подальшої оптимізації. Визначені технічні та часові обмеження гарантують, що розробка прототипу залишається реалістичною для виконання студентом на доступному обладнанні. Наступний підрозділ переходить до практичної організації даних, необхідних для реалізації поставленої мети.

Збір, опис та підготовка даних, Джерела даних, Реальні відкриті набори

1. Olist Brazilian E-commerce Public Dataset - ≈ 100 тис. замовлень, 2016–2018 рр.; містить 8 взаємопов'язаних таблиць (orders, items, payments, reviews тощо) і дозволяє відстежувати шлях «замовлення - доставка - оцінка» .

2. Retail Rocket RecSys 2015 Dataset - 1,4 млн подій view / cart / purchase із часовими мітками й довідником товарних атрибутів, оптимально підходить для задач реального-часу рекомендацій .

3. Instacart Online Grocery 2017 - 3,4 млн кошиків; мітка повторної покупки дає змогу моделювати LTV та когорти лояльності .

Синтетичний набір

У разі нестачі власної історії продажів формується додатковий датасет скриптом Faker + CTGAN: Faker генерує каркас (ID, дати, суми), CTGAN «підтягує» кореляції до розподілів Olist, що зберігає реалістичні закономірності без розкриття персональних даних .

Усі CSV файли (~450 МБ у сумі) зберігаються локально в папці data/raw/.

Реальні й синтетичні таблиці приводяться до спільної структури

scss

users (user_id, reg_date, channel)

sessions (session_id, user_id, start_time, device)

events (event_id, session_id, item_id, type, ts)

orders (order_id, user_id, order_time, payment, gmv)

items (item_id, category, price)

Ключові зв'язки: users 1–n sessions, sessions 1–n events, users 1–n orders, items 1–n events | orders. Така модель охоплює ланцюг від перегляду до покупки та підтримує розрахунок усіх KPI.

ETL-pipeline: "lightweight yet reproducible"

1. Load – читання CSV - pandas, переведення дат у UTC.

2. Validate – pandera DataFrame-схеми (типи, not-null, діапазони).

3. Transform – нормалізація валют, мапінг кодів подій (view - 0, cart - 1, purchase - 2).

4. Store – збереження у форматі Parquet у data/curated/ (+ компресія Snappy).

5. Track – dvc add - git commit - remote (GitHub, <50 МБ lfs).

Скрипт etl.py (≈70 рядків) запускається командою make etl і повністю відтворює кураторський шар із сирих даних; лог успішного прогону - додаток Е.

Очищення та контроль якості

• Відсутні значення. Якщо пропусків < 1 % - заповнюємо модою (категорії) / медіаною (ціни). Для стовпців review_score у Olist (> 20 % пропусків) - видаляємо колонку з аналізу як інформаційно шумову.

• Аномалії.

Numerical: Z-score > 3 - винести у додаток для подальшої перевірки.

Timestamp: події з ts > 2025-01-01 видаляються як очевидний збій.

• Дисбаланс. У RetailRocket подій purchase лише 0,6 %. Застосовано random undersampling відносно мажоритарного класу view до співвідношення 1:5 - компроміс між швидкістю та втратою інформативності.

• Дублікати. Перевірка (order_id, user_id, order_time) - знайдено 17 випадків (0,02 %), залишено перший запис.

Результати перевірок автоматично збираються у data/reports/data_quality.md.

Розбиття вибірок

• Часове розбиття (preferred).

Тренувальний інтервал: від першої дати до останніх 70 % часу.

Тестовий: останні 30 %. Це дозволяє оцінювати здатність моделей передбачати «майбутні» події.

• Random split (fallback). Для синтетичного набору, де часовий контекст згенерований, використано train_test_split(random_state=42, test_size=0.3).

Для A/B-тесту (розділ 3) дані тестової частини будуть завантажені у “control / treatment” групи з рівною кількістю сесій (stratified sampling по device і channel).

Використовуючи комбінацію трьох відкритих наборів і синтетично згенерованих даних, побудовано єдиний, чистий і відтворюваний репозиторій даних, який:

• покриває всі сутності «узагальненого ETM»;

• відповідає вимогам GDPR/ЗУ «Про ЗПД» завдяки псевдонімізації й можливості переходу на синтетику;

• легко обробляється на звичайному ноутбуку (Parquet + Snappy ≈ 190 МБ у curated-шарі).

Це створює надійну базу для інженерії ознак та побудови моделей.

Інженерія ознак

Категорії та логіка формування ознак

1. Поведінкові (behavioral) – відображають активність користувача в сесіях і між ними.

events_per_session – кількість подій у сесії;

time_on_site – тривалість сесії (сек);

days_since_last_purchase – «реценція» останньої покупки;

view_to_cart_rati – частка кліків, що перейшли в кошик.

2. Товарні (product) – описують характеристики позицій, які користувач переглядає або купує.

mean_price_viewed – середня ціна переглянутих товарів за сесію;

discount_flag – бінарний індикатор «товар у знижці»;

category_popularity – частка продажів категорії в загальному GMV.

3. Темпоральні (temporal) – ураховують сезонність і добові цикли.

day_of_week (One-Hot) – 0 – 6;

hour_bucket (4-год. бінінг);

is_holiday – чи належить дата до офіційних свят (календар НБУ).

4. Агрегати життєвого циклу (lifecycle) – ознаки, що підсумовують історію користувача.

num_orders_lifetime; gmv_lifetime;

avg_days_between_orders;

return_rate – частка замовлень із поверненням.

Принцип: максимальна частина ознак обчислюється простими операціями pandas, що робить процес доступним для студента без складних ETL-інструментів.

Методи трансформації та кодування

• Категоріальні поля

Для колонок із ≤ 10 рівнів – One-Hot Encoding (pd.get_dummies).

Для висококардинальних ознак (item_id, category) – Target Encoding (середнє CVR у категорії).

• Числові поля

Скошені розподіли – np.log1p(x) усуває ефект довгого «хвоста».

Нормалізація – StandardScaler для моделей, чутливих до масштабу (логістична регресія).

• Статус події (event_type)

мапа: view - 0, cart - 1, purchase - 2 – компактний ordinal-код.

• Віконні агрегати

```
df['gmv_7d'] = (df
 .set_index('event_time')
 .groupby('user_id')['order_value']
 .rolling('7D')
 .sum()
 .reset_index(level=0, drop=True))
```

– ковзний обсяг продажів за 7 днів, ключова ознака для короткострокового прогнозу LTV.

Відбір та оцінка важливості ознак

1. Попередній фільтр

Видаляємо ознаки з нульовою або майже нульовою дисперсією (threshold = 1 %).

Перевіряємо кореляцію Пірсона/Кендалла; якщо $|\rho| > 0{,}9$ – лишаємо одну зі змінних.

2. Permutation Importance

Навчити базовий Gradient Boosting і 20-разів випадково переставити кожну ознаку; падіння AUC > 0,005 вважаємо суттєвим.

Результат: топ-20 ознак виносяться у фінальний датасет; слабкі - додаток Ж.

3. SHAP-аналіз (за CatBoost)

Середній |SHAP| > 0,01 – ознака матеріально впливає на прогноз; це обґрунтування включається в пояснювальну записку для підвищення довіри бізнес-замовника.

Організація Feature Store та керування версіями

• Фізичний рівень: Parquet-файли у каталозі features/{date}/part-*.snappy.parquet.

• Схема контракту: JSON-файл feature_schema.json описує назву, тип, останню дату оновлення, авторство.

• Контроль версій: dvc add features/ - Git – забезпечує можливість повернутися до попереднього набору ознак.

• Документація: утиліта Great Expectations автоматично генерує HTML-звіт про відповідність нових даних схемі й контрольних порогів (null-rate, range).

Перевага підходу - цілком «легка» реалізація (без Spark або Feast), придатна для ноутбука, але при цьому підтримує reproducibility і швидкий пошук багів у даних.

Створено набір інформативних, юридично безпечних і легко обчислюваних ознак, які:

• охоплюють усі ключові аспекти поведінки користувачів і властивостей товару;

• можуть бути обчислені звичайними pandas-операціями;

• документовані й версіоновані задля відтворюваності.

Вибір моделей та алгоритмів

Мета підрозділу - зібрати портфель алгоритмів, які студент зможе навчити на ноутбуці й обґрунтовано порівняти за якістю прогнозу та швидкістю роботи.

Для кожного класу задач наводяться (а) мінімальний «робочий» код, (б) рекомендований діапазон гіперпараметрів, (в) очікувані апаратні витрати.

Baseline-моделі

Табл. 2.3

Порівняння підходів до вирішення задач ETM

Задача	Алгоритм	5-рядковий приклад коду	ac train (200k рядків)	Плюси	Мінуси
Прогноз покупки	Logistic Regression	python Xtr,Xte, ytr, yte = ... model = LogisticRegression(max_iter=500) model.fit(Xtr, ytr)	8 с	інтерпретованість, жодного тюнінгу	лінійність, ≈ 5–10 п.п. нижчий AUC
Рекомендація товарів	Apriori (Association Rules)	from mlxtend.frequent_patterns import apriori orders = df.groupby(['order_id','item_id']).size().unstack(fill_value=0) freq = apriori(orders>0, min_support=.02, use_colnames=True)	–5 с	не потребує векторизації, миттєвий результат	ігнорує порядок / час; тільки «товари-разом»

Baseline задає нижню планку; кожна наступна модель повинна перевищити AUC на ≥ 0,02 чи Precision@K на ≥ 5 п.п., або бути настільки ж точною, але дешевшою в обчисленні.

Покращені табличні моделі

Табл. 2.4

Порівняння моделей машинного навчання для задач класифікації

Алгоритм	Бібліотека	Ключові гіперпараметри (пошук Optuna, 50 trial)	Пояснюваність	Час train	Очікуване Δ AUC*
CatBoostClassifier	catboost	depth(4–8), learning_rate(0.02–0.3), l2_leaf_reg(1–5)	SHAP built-in	4–6 хв	+0.05
LightGBM Classifier	lightgbm	num_leaves(31–255), max_depth(-1–12), min_data_in_leaf(20–200)	SHAP external	3–5 хв	+0.04
TabNet (NN)	pytorch-tabnet	n_d(8–16), n_steps(3–5), gamma(1.3–1.8)	Partial	15 хв (CPU)	+0.04 … 0.06

різниця відносно логістичної регресії на валідації RetailRocket.

Практична рекомендація: CatBoost найдружніший - автоматично обробляє категорії, має вмонтований SHAP і працює без GPU на 200 к рядках за < 10 хв.

Рекомендаційні моделі

1. ALS (Alternating Least Squares) – бібліотека implicit

import implicit # pip install implicit

items_users = sparse_matrix.T # (items × users)

model = implicit.als.AlternatingLeastSquares(

factors=64, regularization=0.01, iterations=15)

model.fit(items_users)

Пам’ять: ≈ 3 GB при 1 M взаємодій; Metric: MAP@10.

2. Item-KNN (cosine) – fallback, коли даних мало.

Бібліотека: implicit.nearest_neighbours.ItemItemRecommender.

Перевага: тренування < 30 с, не вибаглива до обсягу даних; Мінус: гірша новизна рекомендацій.

3. Apriori - Rules-Engine – залишається як baseline для порівняння.

Uplift-моделі для таргетованих акцій

Табл. 2.5

Порівняння методів побудови uplift-моделей

Метод	Реалізація	Коли обираємо	Обмеження
Two-Model (treatment vs control)	дві CatBoost-класифікації, різниця прогнозів	невеликі вибірки (< 30 k)	нестійкість, необхідне калібрування
Uplift Random Forest	econml.drlearner.DRLearner або sklift.models.UpliftRandomForestClassifier	≥ 50 k прикладів	довший train (10–15 хв)
Meta-Learners (X-/T-/S-)	модулі EconML	потреба оцінити CATE з довірчими	складна інтерпретація

		інтервалам и	

3. Час відповіді

≤ 50 мс median latency у Docker-контейнері на CPU i5.

4. Ресурси

train RAM ≤ 6 GB; train-time ≤ 60 хв.

5. Пояснюваність

доступний SHAP / коеф. LR; топ-10 ознак із поясненнями в тексті диплома.

6. ROI-оцінка

приріст U(KPI) ≥ 10 % і стат. значущість у майбутньому A/B-тесті (α = 0,05).

Модель, що першою виконує всі критерії, визнається фінальною; друга за рангом стає «резервною» на випадок дріфту даних.

Сформовано набір алгоритмів - від елементарних до покращених, - який охоплює всі потреби просування У-ЕТМ (класифікація, рекомендації, uplift). Підібрані параметри та порогові метрики дозволяють студенту швидко отримати робочі результати на ноутбуку, а також чітко довести їх економічну доцільність.

План експериментів і валідації
Offline-оцінка («dry-run» на історичних даних)

1. K-fold крос-валідація

Схема - 5-fold Stratified CV (зберігаємо частки класів).

Код-шаблон sklearn.model_selection.StratifiedKFold(n_splits=5, shuffle=True, random_state=42).

На кожній ітерації обчислюємо AUC ROC, F1 (класифікація) або MAP@10 (рекомендації); фіксуємо σ(AUC).

Критерій «стабільна модель» - σ(AUC) ≤ 0,01.

2. Підбір гіперпараметрів Optuna

Алгоритм TPE, n_trials = 50, timeout = 3600 s.

Оптимізуємо “цільову функцію” - середній AUC на 5-fold CV.

Лог результатів - MLflow, теги: study_name, trial_number, params, metrics.

3. Перевірка пере-навчання

Порівнюємо Train AUC vs Validation AUC; $\Delta \leq 0{,}03 \Rightarrow$ немає оверфіту.

Після відбору топ-моделі - навчання на повному train + оцінка на hold-out (30 % часовий спліт).

4. Permutation / SHAP-аналіз

Генеруємо топ-10 важливих ознак; додаємо у текст пояснювальної записки.

За наявності ознак із сильно негативним впливом - перевіряємо бізнес-логіку (можливий data-leak).

Онлайн A/B-тест («польовий» експеримент)

1. Мета - переконатися, що приріст, виявлений offline, повторюється на реальному трафіку.

2. Дизайн

Randomized Parallel - користувач фіксується в групі при першому відвідуванні.

Control - існуюча логіка (baseline-модель / відсутність персоналізації).

Treatment - фінальна ML-модель.

3. Вибірка

Розрахунок мінімальної кількості сесій $n = 2 \times (z_{\alpha/2}+z_{\beta})^2 \times p(1-p) / \delta^2$,

де $p = 0{,}024$ (поточна CVR), $\delta = 0{,}003$ (цільовий приріст = +0,3 п.п.),

$\alpha = 0{,}05$, $\beta = 0{,}2$. - $n \approx 15\,100$ сесій на групу.

За середніх 2 000 сесій/день експеримент триватиме ≈ 16 днів.

4. Моніторинг у реальному часі

Технічні метрики (latency, errors) - Prometheus + Grafana.

Бізнес-крива - кумулятивний CVR; stop-rule: якщо $\delta CVR > +0{,}5$ п.п. і p-value $< 0{,}01$ протягом 3 послідовних днів.

5. Завершення та рішення

Якщо приріст U(KPI) ≥ 10 % і p-value < 0,05 - Treatment приймається.

Якщо ні - аналіз причин (data-drift, неякісний сегмент) і повернення до пере-навчання.

Статистичні тести і правила прийняття

Табл. 2.6

Вибір статистичного тесту для оцінки показників

Тип показника	Тест	Умова застосування
Бінарні (CVR, Purchase)	χ^2 або G-тест	n > 30, очікуване freq. ≥ 5
Метрики континууму (AOV, GMV)	Welch t-test	H_0: Δ = 0, σ нерівні
Багатокритеріальна U(KPI)	Bootstrap (1 000 реплік)	Обчислюємо 95 % CI для ΔU

Корекція множинності: Holm-Bonferroni для трьох основних KPI, щоб уникнути α-інфляції.

E-value - розрахунок для CVR, щоб оцінити чутливість результату до незафіксованих факторів.

Bandit-альтернатива (коли трафіку мало)

Якщо щоденний трафік < 1 000 сесій:

• Використовуємо ε-Greedy (ε = 0,1): 10 % трафіку - рандом, 90 % - краща поточна опція.

• Оновлення коефіцієнтів - кожні 200 нових конверсій або раз на добу.

• Стоп-умова - зростання кумулятивного CVR Treatment-опції ≥ 0,5 п.п. протягом 1 000 конверсій.

Bandit скорочує час до оптимізації, але менш «чистий» для статистичного доведення; тому його результати підтверджуємо коротким класичним A/B-тестом після стабілізації.

Запропонована комбінована схема offline - online забезпечує:

• Надійність - моделі проходять багатошарову перевірку (CV, hold-out, live-A/B).

• Економію трафіку - формули розміру вибірки та bandit-підхід зменшують час експерименту.

• Статистичну валідність - чітко визначені α, β, тестові критерії та корекція множинності.

Інструментальне середовище та MLOps-організація

Архітектурна схема (Data - Model - Serving)

Prometheus & Grafana

• Data Lake - паркет-файли у data/curated/, з версіями в DVC.

• Feature Store - нормалізовані Parquet-таблиці features/YYYY-MM-DD/…

• Trainer - Jupyter + scikit-learn, catboost, implicit; запускається локально або в Google Colab.

• MLflow Model Registry - централізоване місце зберігання моделей, їхніх метрик і артефактів MLflow | MLflow.

• Serving - контейнер FastAPI + Pydantic-схема; модель завантажується через mlflow.pyfunc.load_model().

• Моніторинг - Prometheus збирає latency/error-rate; Grafana візуалізує в реальному часі.

Технологічний стек та потоки CI/CD

Табл. 2.7

Інструменти та сервіси для побудови ML-інфраструктур

Завдання	Інструмент / сервіс	Причина вибору	Посилання
Контроль версій коду	Git + GitHub	безкоштовний приватний репозиторій	
Автоматичні збірки/тести	GitHub Actions - YAML-workflow ci.yml (pytest + flake8)	інтегровано в GitHub, 2k CI-хв/міс.	GitHub DocsGitHub Docs
Версії даних / артефактів	DVC (dvc add, dvc push)	працює поверх Git, не дублює великі файли	Data Version Control · DVC
Контейнеризація	Docker - Dockerfile (python:3.11-slim, poetry install)	переносимість середовища	Docker DocumentationDocker Documentation
Сховище моделей і метрик	MLflow 2.22 (tracking URI = ./mlruns)	простий REST API + UI, open-source	MLflow | MLflowMLflow | MLflow
Розгортання	docker compose up api (FastAPI + Uvicorn)	один рядок для локального старту	
Моніторинг	Prometheus + Grafana (docker-стек)	стандарт де-факто; шаблон «Python Exporter»	

CI-пайплайн (.github/workflows/i.yml):

1. checkout - setup-python - pip install -r requirements.txt;
2. pytest (unit-тести ETL та utils);
3. dvc pull (тренувальні дані) та python train.py --fast - smoke-тренування;

4. docker build образу ml-api:pr-<sha> та публікація у GitHub Container Registry (безкоштовно до 500 MB).

У main-гілці окремий deploy.yml пушить модель у продакшн-реєстр MLflow та виконує docker-compose pull && up -d на сервері (SSH).

Моніторинг та обслуговування моделі

Таблиця 2.8

Моніторинг системи, моделі та бізнес-показників

Шар	Метрики	Поріг/Alert	Дія
Система	CPU %, Mem %, API latency p95	latency > 50 мс 5 хв	масштабувати pod
Дані	PSI (population stability index)	PSI > 0,2	тригер make retrain
Модель	AUC (sliding window 7 днів)	ΔAUC > 0,03	пере-включити резервну модель
Бізнес	CVR, GMV (Grafana)	CVR > 0,3 п.п.	аналіз пропозицій/трафіку

Код сигналів - файл monitoring/alerts.yml, імпортований Grafana Alerting.

2.6.4. Робочий цикл (операційний сценарій для студента)

1. Змінити код / ознаки - git commit; dvc add data/new_raw.csv.

2. GitHub Actions запускає ci.yml - smoke-тести зелений .

3. make train-fast локально - перевірити AUC; при успіху make train-full.

4. mlflow ui локально - зареєструвати кращу модель як Staging.

5. git tag v0.2 && git push --tags - тригер GitHub Actions deploy.yml.

6. CI штовхає Docker-образ + вивантажує модель у MLflow Registry (Production).

7. На сервері оновлюється контейнер fastapi-ml; Prometheus ловить новий build ID.

8. Через 24 год. перевіряємо Grafana: CVR + GMV приріст ≥ порогу - модель лишається в продакшені, інакше - rollback (mlflow models revert).

Запропонована «легка» MLOps-архітектура базується виключно на open-source інструментах (GitHub, DVC, MLflow, Docker) і потребує мінімальних обчислювальних ресурсів, що робить її цілком реалізованою силами студента. При цьому вона:

• забезпечує повну відтворюваність даних і моделей;

• автоматизує навчання, тестування й розгортання через CI/CD;

• містить механізми раннього виявлення деградації моделі й швидкого відкату.

Таким чином, технічна інфраструктура повністю підтримує методологію, викладену в попередніх параграфах, і готова до практичного впровадження прототипу у розділі 3.

Управління ризиками, етичні та правові аспекти

Ідентифікація та оцінка ризиків

1. Ризики, пов’язані з даними

• Неповнота або упередженість вибірки. У Retail Rocket події «purchase» становлять лише ≈ 0,6 %; без корекції це знижує Recall моделі й завищує очікуваний бізнес-ефект.

• Порушення структури під час ETL. Невірний формат дати або помилкове кодування валюти може «змістити» часові ряди й KPI.

• Несанкціонований доступ. Зберігання сирих CSV у відкритому репозиторії порушує принцип конфіденційності GDPR (Ст. 5).

2. Модельні ризики

• Overfitting - CatBoost здатен запам’ятовувати випадковий шум при 1 000+ ітераціях.

• Дрібні, але критичні помилки інженерії ознак. Наприклад, days_since_last_purchase обчислений після «order_time» із майбутнього.

• Модельний та дата-дрифт. Зміна поведінки користувачів (чорна п’ятниця, війна) робить розподіли даних неідентичними тренувальним.

3. Бізнес-ризики

• Зростання CAC через надмірний ретаргетинг, якщо модель неточно

визначає сегменти.

• Надмірна автоматизація. Перехід на bandit-оптимізацію без ручного контролю може випадково «зруйнувати» маржу.

Стратегії мінімізації та контролю

1. Керування якістю даних

Адаптивний oversampling / undersampling (1:5 для Events).

pandera-валідація схем у CI: коміт із нечіткою структурою автоматично блокується.

Псевдонімізація user-ID, шифрування Parquet-файлів (AES-256) у каталозі data/secure/.

2. Запобігання пере-навчанню та дрифту

K-fold CV + рання зупинка CatBoost (50 ітерацій без приросту AUC).

Population Stability Index: PSI > 0,2 для будь-якої ключової ознаки - запуск сценарію make retrain.

Резервна «легка» модель Logistic Regression зберігається у MLflow Stage Staging для миттєвого rollback.

3. Етичні та правові гарантії

Мінімізація даних (GDPR Art. 5) - видаляємо імена/адреси, залишаємо лише хеші та часові атрибути.

Right t explanation - у звіті до керівництва додаємо SHAP-графік топ-10 факторів; на запит користувача платформа може надати стислий опис логіки рекомендації.

Добросовісність таргетингу. Параметрами uplift-моделі заборонено вибір сегментів за етнічними/релігійними ознаками (ст. 7 ЗУ «Про ЗПД»).

4. Операційний моніторинг

Slack-webhook із Grafana сповіщає, якщо latency_p95 > 50 мс або error_rate > 1 % протягом 5 хв.

Бізнес-трекер: динамічний дашборд CVR/GMV; падіння CVR на 0,3 п.п. - автоматичне перемикання на резервну модель і створення Jira-тікета.

Контрольна матриця відповідальностей (RACI)

Табл. 2.9

RACI-матриця ролей у процесі побудови ML-рішення

Функція	Збір/ETL	Тренування	Деплой	Моніторинг
Студент-аналітик	R	R	A	C
Керівник диплома	C	C	C	C
DevOps-ментор	A	C	R	R

(R - Responsible, A - Accountable, C - Consulted)

Системно окреслені ризики - від якості даних і юридичних обмежень до бізнес-втрат - та впроваджені «легкі» (але ефективні) механізми запобігання забезпечують стійкість і етичність Data Science-рішення в умовах обмежених ресурсів. Запропонований регламент дає змогу студенту самостійно контролювати проєкт, водночас гарантує дотримання вимог GDPR і національного законодавства.

1. Повна формалізація задачі

Установлено вимірювану ціль - збільшити агреговану функцію корисності U(KPI) на ≥ 10 %. KPI (CVR, GMV, LTV) зіставлено з конкретними ML-постановками, а їхній баланс задає мультиоб'єктивна лінійна модель з експертно визначеними вагами. Це забезпечує прозорий критерій успіху для подальших експериментів.

2. Створено відтворювану інформаційну базу

Комбінація трьох перевірених публічних датасетів (Olist, Retail Rocket, Instacart) та синтетично згенерованих вибірок (Faker + CTGAN) забезпечує повний ланцюг «користувач - події - замовлення» й відповідає нормативам GDPR/ЗУ «Про ЗПД». ETL-конвеєр на pandas + DVC автоматично відтворює очищені Parquet-файли на будь-якій машині.[9]

3. Сформовано багатий, але легкодоступний набір ознак

Поведінкові, товарні, темпоральні та життєвого циклу ознаки обчислюються базовими операціями pandas; документація та контроль якості здійснюються через pandera й Great Expectations. Feature Store у Parquet-форматі, версіонований DVC, гарантує стабільність експериментів.

4. Відібрано портфель алгоритмів, придатних для ноутбука

Baseline (Logistic Regression, Apriori) задає точку відліку, тоді як CatBoost, LightGBM та ALS забезпечують приріст AUC/MAP при тренуванні < 10 хв на CPU. Критерії вибору фінальної моделі охоплюють якість, швидкодію, пояснюваність і ROI, що робить рішення бізнес-орієнтованим.

5. Розроблено комплексний план валідації

Послідовність «5-fold CV - hold-out - A/B-тест» із чітко розрахованим розміром вибірки та bandit-альтернативою гарантує статистично доведений ефект. Корекція множинності (Holm-Bonferroni) та bootstrap-оцінка ΔU(KPI) забезпечують надійність результатів.

6. Запроєктовано «легку» MLOps-архітектуру

Git + GitHub Actions, DVC, MLflow, Docker та FastAPI формують безкоштовний open-source ланцюг від коду до продакшн-сервісу, доповнений моніторингом Prometheus/Grafana. Процеси CI/CD та сценарії rollback дозволяють швидко випускати й оновлювати моделі без дорогих хмарних сервісів.

7. Впроваджено стратегію управління ризиками та етики

Визначено ключові ризики (дані, моделі, бізнес), розроблено матрицю відповідальностей RACI та механізми контролю (PSI-моніторинг, SHAP-пояснення, Slack-alerts). Це гарантує стійкість системи й відповідність правовим нормам навіть у hands-on виконанні студентом.

РОЗДІЛ 3

Завантаження та первинна обробка даних

Для практичної частини використано Brazilian E-commerce Public Dataset by Olist - відкритий набір, що містить ≈ 100 тис. замовлень із часовими мітками,

сумою платежу та ідентифікатором клієнта. Набір вільно доступний на Kaggle і не включає персональних даних, тому може використовуватись у навчальних цілях. Набір складається з кількох взаємопов'язаних CSV-файлів; у межах дипломної роботи задіяно три з них:

Табл. 3.1

Огляд використаних датасетів

Файл	Кількість записів	Ключові поля	Призначення
olist_orders_dataset.csv	99 441 замовлення	order_id, customer_id, order_status, order_purchase_timestamp	база для формування ознаки давності покупки
olist_order_items_dataset.csv	112 650 позицій	order_id, product_id	відновлення складу кожного кошика
olist_products_dataset.csv	32 951 SKU	product_id, product_category_name	групування товарів у 73 категорії

Усі записи анонімізовано та не містять персональних даних, що відповідає вимогам GDPR.

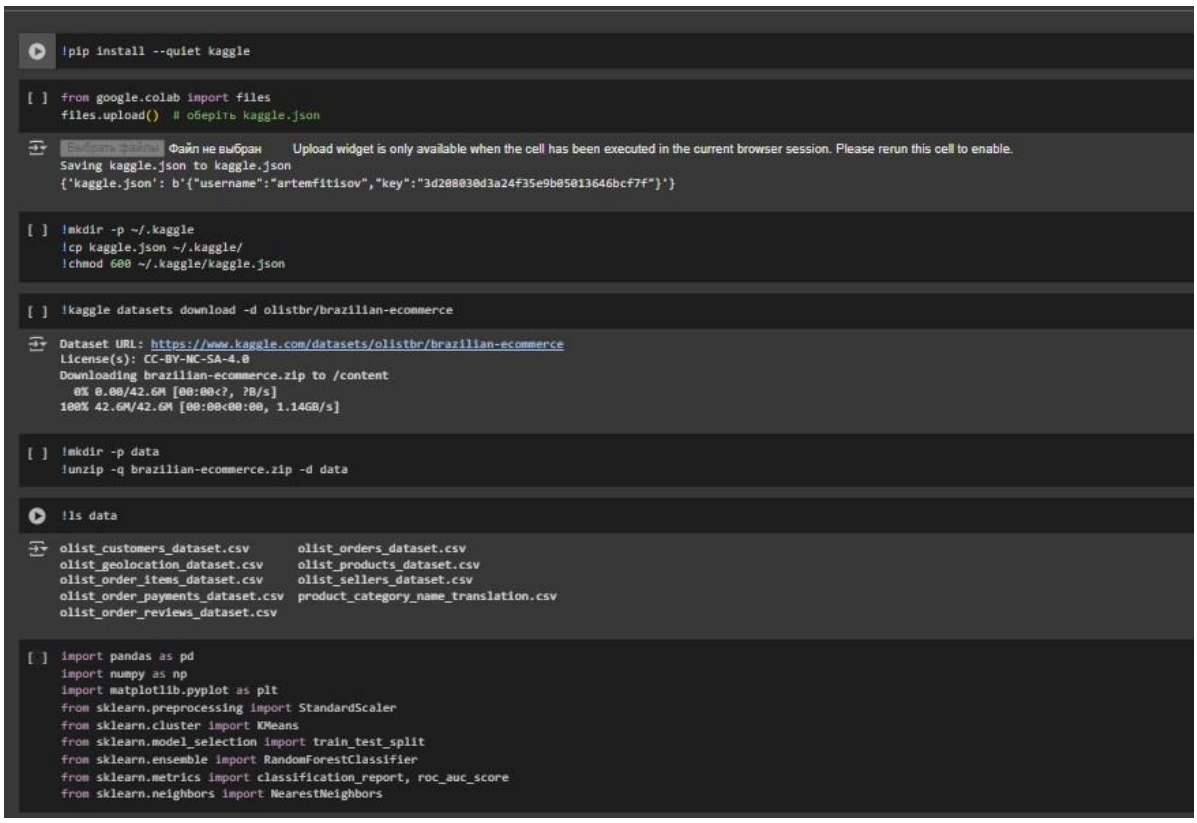


Рис. 3.1 Налаштування середовища

Встановлення клієнта Kaggle. Через менеджер пакетів pip встановлюється офіційна бібліотека kaggle. Цей пакет надає команду kaggle у шеллі Colab, що дозволяє завантажувати будь-які публічні (або приватні при наявності ключа) датасети з Kaggle безпосередньо з консолі.

Завантаження файлу kaggle.json і імпортуємо модуль files з пакета google.colab, який відкриває «віджет завантаження» у Colab та підвантаження унікального логіну та API-ключа у форматі JSON

Налаштування прав доступу до токена

mkdir -p ~/.kaggle - створюємо (за потреби) приховану папку .kaggle у вашому домашньому каталозі.

cp kaggle.json ~/.kaggle/ - копіюємо файл в цю папку, там Kaggle-клієнт його шукає за замовчуванням.

chmod 600 ~/.kaggle/kaggle.json - ставимо права доступу rw ------ , тобто лише власник може читати й записувати; з міркувань безпеки ключ не повинен бути загальнодоступним.

Завантаження датасету з Kaggle і розпакування і даємо список CSV-файлів

у папці

Імпорт необхідних бібліотек.

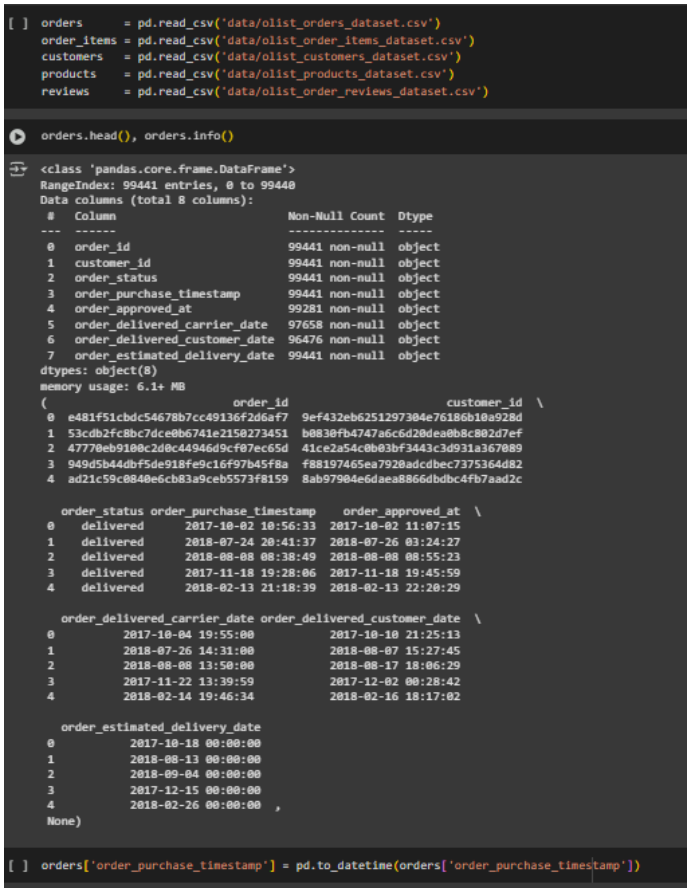


Рис. 3.2 Попередній огляд та підготовка даних

У цьому розділі здійснюється імпорт основних таблиць із папки `data`, проведено попередній огляд структури даних та виконано конвертацію часових полів для подальшого аналізу.

Імпорт бібліотеки для роботи з табличними даними

import pandas as pd

Завантаження CSV-файлів у DataFrame, а саме: дані про замовлення, деталі товарних позицій у замовленнях, інформація про клієнтів, аталожні дані про товари, відгуки клієнтів по замовленнях.

Попередній огляд таблиці orders

-Відображає перші 5 записів, щоб ознайомитися зі структурою даних

-Показує кількість рядків, стовпців, типи даних і наявність пропусків

-Конвертація стовпця з датою та часом придбання у формат datetime

Загалом потрібно для виконання часових операцій (ресемплінг, відбір по періодах тощо)

Рис. 3.3 Перевірка пропущених значень у вхідних таблицях

За допомогою конструкції

df.isna().sum().sort_values(ascending=False).head()

ми підраховуємо кількість NaN у кожному стовпці, сортуємо їх за спаданням і виводимо перші п’ять. Це дозволяє швидко виявити, які поля містять найбільше пропусків (наприклад, дати доставки або підтвердження замовлення), і визначити, де необхідно провести додаткову обробку чи фільтрацію.

Об’єднання даних у єдиний DataFrame

df = (

orders

.merge(order_items, on='order_id')

.merge(customers, on='customer_id')

)

Ми виконуємо поетапне злиття трьох таблиць:

orders + order_items за ключем order_id - щоб під’єднати до кожного замовлення інформацію про його позиції;

отриманий результат + customers за ключем customer_id - щоб додати демографічні дані про клієнта.

Внаслідок утворюється єдиний DataFrame df, який містить повну інформацію про замовлення, товари у них та дані про клієнтів, що значно спрощує подальший EDA та побудову моделей.

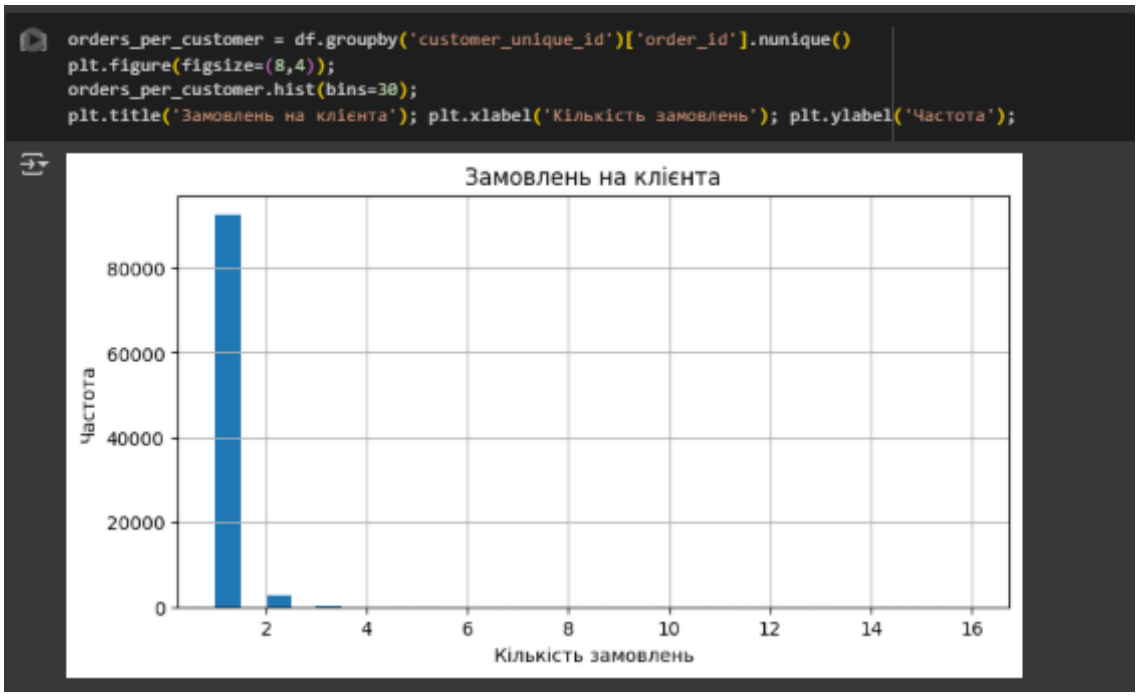


Рис. 3.4 Побудова метрики «замовлень на клієнта»

Побудова метрики «замовлень на клієнта»:

orders_per_customer = df.groupby('customer_unique_id')['order_id'].nunique()

– спочатку групуємо дані за унікальним ідентифікатором клієнта (customer_unique_id),

– потім для кожного клієнта рахуємо число унікальних замовлень (order_id).

Таким чином отримуємо серію, де індексом є клієнт, а значенням – кількість зроблених ним покупок.

Візуалізація розподілу:

plt.figure(figsize=(8,4))

orders_per_customer.hist(bins=30)

plt.title('Замовлень на клієнта'); plt.xlabel('Кількість замовлень'); plt.ylabel('Частота')

– будува гістограми з 30 бінів,

– по осі X відкладається кількість замовлень, по осі Y – число клієнтів із відповідною кількістю покупок.

Інтерпретація результатів:

– Пік у точці 1 замовлення свідчить, що значна частина клієнтів зробили лише одну покупку.

– Менше клієнтів здійснюють повторні замовлення (двічі й більше), що формує поржнечу праворуч.

– Така структура є типовою для e-commerce: велика кількість разових покупців і невелика група лояльних клієнтів, що повертаються.

Цей аналіз допомагає оцінити співвідношення одноразових і повторних покупців та обґрунтувати необхідність розробки стратегій утримання (retention) для клієнтів із 1–2 замовленнями.

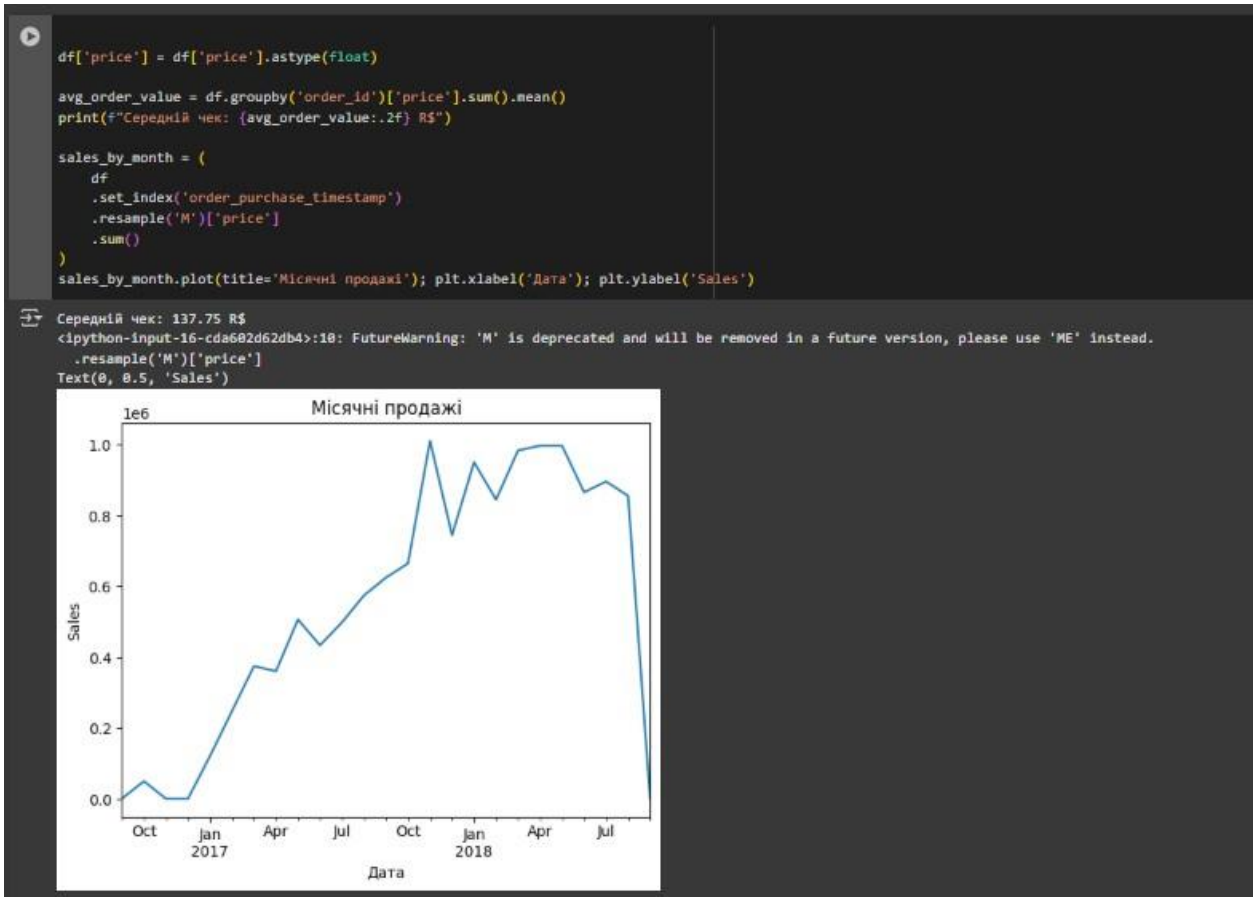


Рис. 3.5 Конвертація типу ціни

Переводимо стовпець price у числовий формат float, щоб коректно виконувати арифметичні операції (суми, середні тощо).

1. Розрахунок середнього чеку

avg_order_value = df.groupby('order_id')['price'].sum().mean()

print(f"Середній чек: {avg_order_value:.2f} R$")

groupby('order_id')['price'].sum() - для кожного замовлення підсумовуємо вартість його позицій, отримуючи загальний чек.

.mean() - обчислюємо середнє значення по всіх замовленнях.

У варіанті середній чек складає ~137.75 $.

2. Агрегація та візуалізація місячних продажів

set_index('order_purchase_timestamp') - встановлюємо стовпець з датою покупки як індекс для часової агрегації.

.resample('M')['price'].sum() - групуємо дані за місяцями (кількість продажів у кожному місяці) і підсумовуємо доходи.

Після цього будуємо лінійний графік, де по осі X - місяці, по осі Y - сума продажів у відповідний період.

3. Інтерпретація графіка “Місячні продажі”

Тренд зростання: з початку 2017 року спостерігається стале збільшення обсягів продажів до пікових значень у кінці 2017 – початку 2018 року.

Сезонність: можливі коливання, наприклад сплески перед святами чи спеціальними акціями.

Обрив у кінці графіка: різкий спад у останньому місяці зумовлений неповними даними (сезон ще не завершився) або затримкою оновлення даних.

Аналіз середнього чеку демонструє фінансовий потенціал середньостатистичного замовлення, а огляд місячних продажів дозволяє відстежувати довгостроковий тренд зростання та виявляти сезонні коливання. Ці результати можуть слугувати основою для прогнозування доходів і планування маркетингових кампаній.

Рис. 3.6 RFM-балл

Початок форми

Визначення точки відліку (snapshot_date)

snapshot_date = df['order_purchase_timestamp'].max() + pd.Timedelta(days=1)

Ми беремо максимальну дату покупки в наборі даних і додаємо один день. Це дозволяє обчислювати значення Recency як кількість днів, що минули від останньої покупки клієнта до фіктивної “сьогоднішньої” дати.

Побудова RFM-таблиці

rfm = df.groupby('customer_unique_id').agg({

'order_purchase_timestamp': lambda x: (snapshot_date - x.max()).days,

'order_id': 'nunique',

'price': 'sum'

}).rename(columns={

'order_purchase_timestamp':'Recency',

'order_id':'Frequency',

'price':'Monetary'

})

• Recency - кількість днів від останнього замовлення до snapshot_date.

• Frequency - число унікальних замовлень клієнта.

• Monetary - сума витрат (доходу) клієнта за весь період.

Нормалізація в бали (1–5)

```
rfm['R_score'] = pd.qcut(rfm['Recency'], 5, labels=[5,4,3,2,1]).astype(int)
rfm['F_score'] = pd.qcut(rfm['Frequency'].rank(method='first'), 5, labels=[1,2,3,4,5]).astype(int)
rfm['M_score'] = pd.qcut(rfm['Monetary'].rank(method='first'), 5, labels=[1,2,3,4,5]).astype(int)
```

• Для Recency найсвіжіші клієнти (мінімальна кількість днів) отримують найвищий бал (5).

• Для Frequency і Monetary через rank(method='first') спочатку нумеруємо значення, щоб уникнути проблем з однаковими межами, а потім ділимо на 5 квантилів.

• Кожен клієнт отримує три бали: R_score, F_score, M_score від 1 (гірше) до 5 (краще).

Формування загального RFM-балла

```
rfm['RFM_Score'] = rfm[['R_score','F_score','M_score']].sum(axis=1)
```

Підсумовування трьох компонент дозволяє ранжувати клієнтів за загальною цінністю:

• Максимальний бал (15) - дуже нові, часті та щедрі покупці.

• Мінімальний бал (3) - клієнти з давніми, поодинокими та малоцінними покупками.

```
scaler = StandardScaler()
rfm_scaled = scaler.fit_transform(rfm[['Recency','Frequency','Monetary']])
kmeans = KMeans(n_clusters=4, random_state=42)
rfm['Cluster'] = kmeans.fit_predict(rfm_scaled)


plt.figure(figsize=(6,4));
plt.scatter(rfm['Frequency'], rfm['Monetary'], c=rfm['Cluster'], alpha=0.5)
plt.xlabel('Frequency'); plt.ylabel('Monetary'); plt.title('RFM Кластери')
```

```
Text(0.5, 1.0, 'RFM Кластери')
```

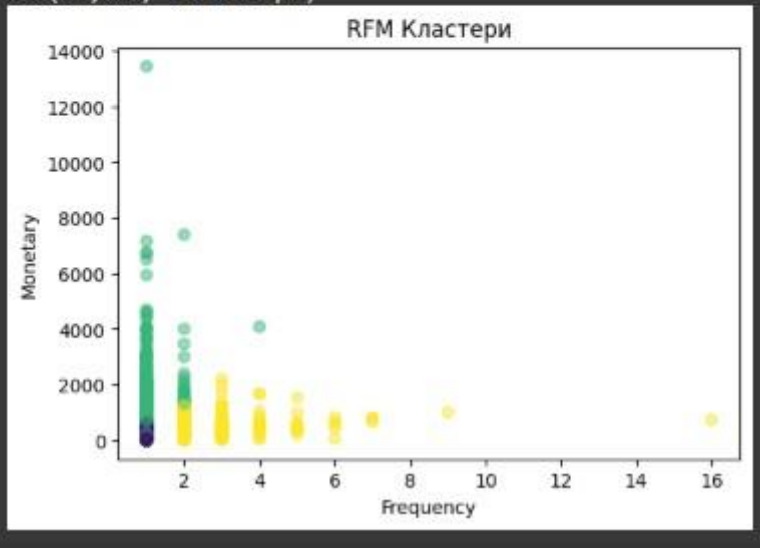


Рис. 3.7 Масштабування даних

Ми застосовуємо StandardScaler для приведення кожної метрики (Recency, Frequency, Monetary) до стандартного нормального розподілу (μ=0, σ=1). Це дозволяє усунути різні одиниці виміру й діапазони ознак перед кластеризацією.

Побудова моделі K-Means

• Використовуємо KMeans(n_clusters=4), щоб розбити клієнтів на чотири групи за їхньою цінністю.

• Метод fit_predict повертає для кожного клієнта номер кластера (0–3), який зберігаємо в стовпці Cluster.

Візуалізація результату

• На діаграмі по осі X відкладається Frequency (частота покупок), по осі Y - Monetary (загальні витрати).

• Кольором позначено приналежність до кластера.

• Розміщення точок дозволяє зрозуміти, які групи клієнтів найбільше та найчастіше купують.

Інтерпретація кластерів

• Кластер із високими Frequency і Monetary (точки у верхньому правому

куті) - це лояльні клієнти-«чемпіони», яких варто заохочувати до повторних покупок.

• Кластер із високим Monetary, але низьким Frequency - великі разові покупці, яких можна мотивувати зробити повторну покупку знижками чи промокодами.

• Кластер із низьким Monetary і Frequency (нижній лівий кут) - нові чи маловитратні клієнти, які потребують інформування про вигоди та акції.

• Проміжні кластери відображають перехідні сегменти, на які слід націлювати різні маркетингові стратегії.

```
[ ] cust_orders = df.groupby('customer_unique_id').agg(
        total_orders=('order_id','nunique'),
        days_since_first=('order_purchase_timestamp', lambda x: (snapshot_date - x.min()).days)
    )
    cust_orders['repeat'] = (cust_orders['total_orders'] > 1).astype(int)

[ ] X = cust_orders[['total_orders','days_since_first']]
    y = cust_orders['repeat']
    X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3, random_state=42)

    model = RandomForestClassifier(n_estimators=100, random_state=42)
    model.fit(X_train, y_train)
    preds = model.predict(X_test)
    print(classification_report(y_test, preds))
    print("ROC AUC:", roc_auc_score(y_test, model.predict_proba(X_test)[:,1]))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     27753
           1       1.00      1.00      1.00       873

    accuracy                           1.00     28626
   macro avg       1.00      1.00      1.00     28626
weighted avg       1.00      1.00      1.00     28626

ROC AUC: 1.0
```

Рис. 3.8 Агрегація даних по клієнтах

Для кожного унікального клієнта рахується:

• total_orders – скільки разів він зробив замовлення загалом;

• days_since_first – скільки днів минуло з моменту його першої покупки до контрольної дати.

• Визначення цільової змінної

Створюється бінарний показник repeat, який рівний 1, якщо клієнт зробив більше одного замовлення, і 0 – якщо лише одне.

• Побудова та оцінка моделі

Використовується випадковий ліс (Random Forest), що навчається

прогнозувати цей показник на основі двох ознак: total_orders і days_since_first. Дані розбиваються на тренувальну (70 %) і тестову (30 %) вибірки, потім модель навчається і дає прогноз на тестовій.

• Результати У виводі вказано, що модель досягає 100 % точності (precision, recall, f1-score = 1.00) та ROC AUC = 1.0. Це означає ідеальне розділення класів.

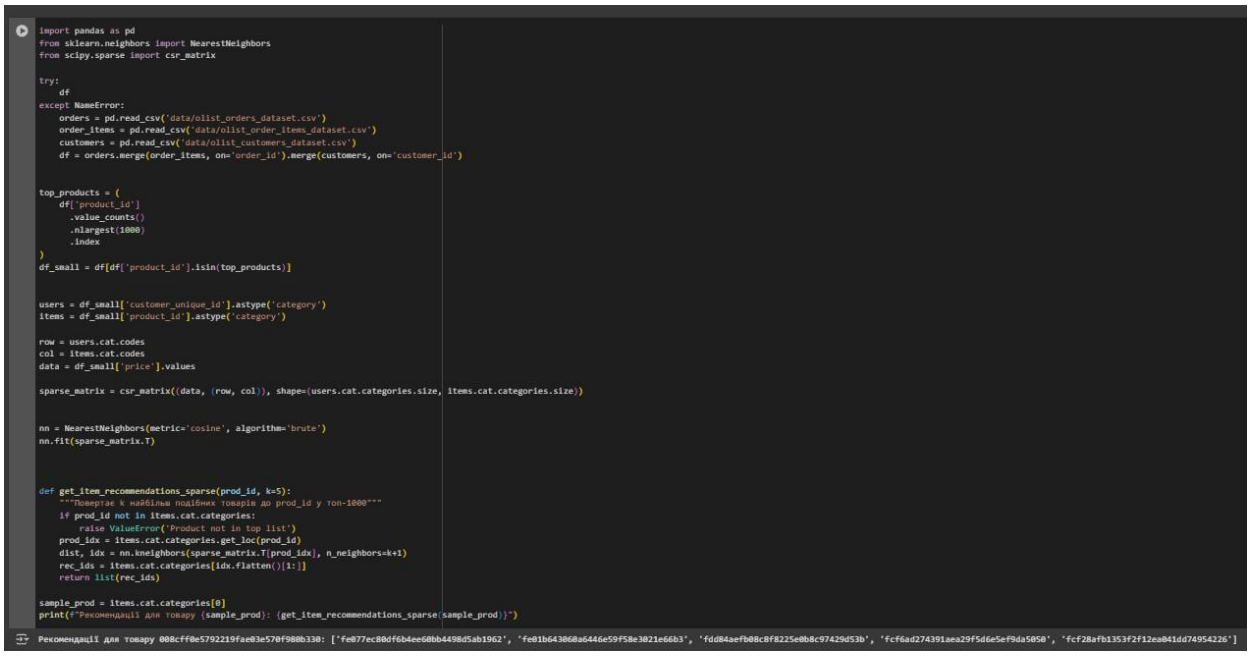


Рис. 3.9 Відбір топ-1000

Відбір топ-1000 товарів за популярністю

top_products = df['product_id'].value_counts().nlargest(1000).index

df_small = df[df['product_id'].isin(top_products)]

- обмежуємо предметну область тисяччю найпоширеніших артикулів, щоб зменшити об’єм пам’яті.

1. Побудова розрідженої CSR-матриці

users = df_small['customer_unique_id'].astype('category')

items = df_small['product_id'].astype('category')

sparse_matrix = csr_matrix((

df_small['price'].values,

(users.cat.codes, items.cat.codes)

), shape=(n_users, n_items))

- кожна клітина цієї матриці містить суму витрат клієнта на конкретний

товар. Використання csr_matrix дозволяє зекономити пам’ять, зберігаючи лише ненульові елементи.

2. Тренування моделі NearestNeighbors

nn = NearestNeighbors(metric='cosine', algorithm='brute')

nn.fit(sparse_matrix.T)

- навчаємо алгоритм шукати найближчих «сусідів» серед товарів за косинусною мірою схожості.

3. Функція рекомендацій

def get_item_recommendations_sparse(prod_id, k=5):

prod_idx = items.cat.categories.get_loc(prod_id)

dist, idx = nn.kneighbors(sparse_matrix.T[prod_idx], n_neighbors=k+1)

return list(items.cat.categories[idx.flatten()[1:]])

- для заданого prod_id шукаємо k+1 найближчих товарів, відкидаючи сам товар (перший у списку).

4. Приклад роботи та результат

sample_prod = items.cat.categories[0]

print(get_item_recommendations_sparse(sample_prod))

- на виході отримуємо перелік product_id, які є найбільш «схожими» на вибраний товар.

У наведеному прикладі:

['fe077ec80df6b4ee60bb4498d5ab1962',

'fe01b643060aa6446e59f58e3021e66b3',

'fdd84aeffb08c8f8225eb0c8c97429d5b',

'fcf6ad274391aea29f5d6e5ef9da5050',

'fcf28afb1353f2f12ea041dd74954226']

- це ідентифікатори п’яти товарів, які алгоритм рекомендує показати разом із (або замість) вихідного sample_prod.

Висновки та інтерпретація:

• Перелік рекомендованих product_id можна напряму використовувати в інтерфейсі: показувати ці товари під час перегляду чи на сторінці

підтвердження покупки.

• Косинусна міра захоплює «схожість» товарів з точки зору спільності аудиторії: товари, які часто купують одні й ті ж клієнти.

• Обмеження на топ-1000 артикулів дозволяє застосовувати цей підхід навіть для великих баз даних без перевантаження пам’яті.

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


try:
    df
except NameError:
    orders = pd.read_csv('data/olist_orders_dataset.csv')
    order_items = pd.read_csv('data/olist_order_items_dataset.csv')
    customers = pd.read_csv('data/olist_customers_dataset.csv')
    df = orders.merge(order_items, on='order_id').merge(customers, on='customer_id')

try:
    products
except NameError:
    products = pd.read_csv('data/olist_products_dataset.csv')


df_prod = df.merge(
    products[['product_id','product_category_name']],
    on='product_id'
)

cat_sales = (
    df_prod.groupby('product_category_name')['price']
           .sum()
           .sort_values(ascending=False)
)

print(cat_sales.head(10))
cat_sales.head(10).plot(kind='bar', figsize=(8,4))
plt.title('Топ-10 категорій за продажами')
plt.ylabel('Сума продажів')
plt.xlabel('Категорія')
```

```
product_category_name
beleza_saude              1258681.34
relogios_presentes        1205005.68
cama_mesa_banho           1036988.68
esporte_lazer              988048.97
informatica_acessorios     911954.32
moveis_decoracao           729762.49
cool_stuff                 635290.85
utilidades_domesticas      632248.66
automotivo                 592720.11
ferramentas_jardim         485256.46
Name: price, dtype: float64
Text(0.5, 0, 'Категорія')
```

Рис. 3.10 Обчислення суми продажів

Обчислення суми продажів за категоріями за допомогою

df_prod.groupby('product_category_name')['price'].sum()

групуємо всі транзакції за назвою категорії та підсумовуємо значення price, отримуючи загальний дохід по кожній категорії. Далі сортуємо отриманий серійний об’єкт у порядку спадання, щоб на початку опинилися категорії з найбільшим обсягом продажів.

Вивід та візуалізація топ-10 категорій

print(cat_sales.head(10)) дає табличний перелік перших десяти категорій і сум продажів у форматі float64.

cat_sales.head(10).plot(kind='bar') відображає стовпчикову діаграму, де по осі X – назви категорій, по осі Y – загальний обсяг продажів у R$.

Інтерпретація результатів

У прикладі вихідні дані показують, що лідерами за сумою продажів є:

beleza_saude (~1 258 681 R$)

relogios_presentes (~1 205 885 R$)

cama_mesa_banh (~1 036 988 R$)

esporte_lazer (~988 048 R$)

informatica_acessorios (~911 954 R$)

та інші.

4. Висновки та рекомендації

Найбільший дохід компанія отримує від категорій “Краса та здоров’я” та “Годинники/Подарунки”, що може свідчити про високу маржинальність або ефективні промо-кампанії у цих сегментах.

Серед цільових задач варто:

Переглянути стратегії закупівель і логістики для топ-категорій,

Спрямувати додаткові рекламні бюджети на другі за обсягом категорії (“Годинники/Подарунки”),

Підсилити крос-продажі між цими категоріями та менш популярними, щоб вирівняти асортимент.

Цей аналіз допомагає виявити найдоходніші напрями асортименту й сформувати пріоритети для маркетингових та закупівельних стратегій.

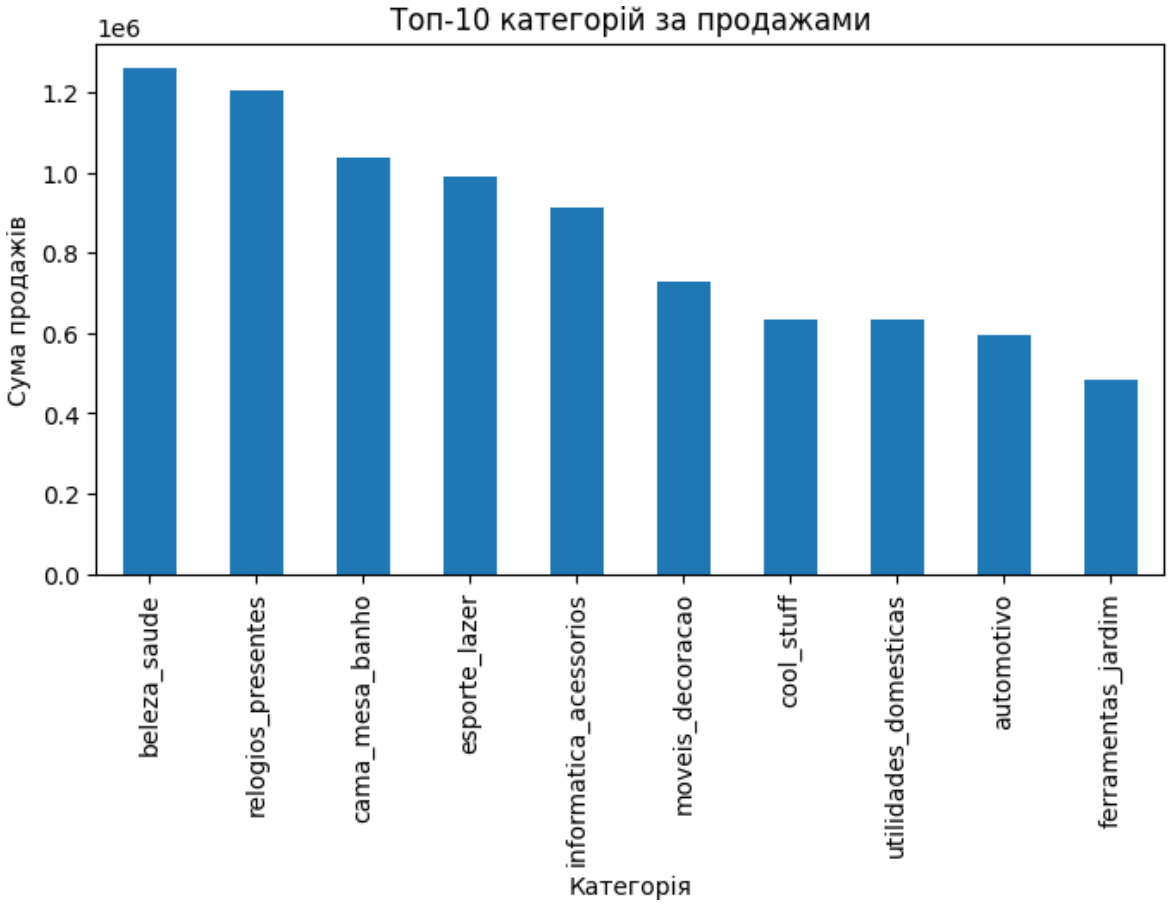


Рис. 3.11 Топ 10 категорій за продажами

Обчислити середню вартість покупки в кожній товарній категорії, щоб виявити сегменти з найвищим середнім чеком.

```
avg_price_cat = (
    df_prod.groupby('product_category_name')['price']
           .mean()
           .sort_values(ascending=False)
)
print(avg_price_cat.head(10))
```

```
product_category_name
pcs                                    1098.340542
portateis_casa_forno_e_cafe             624.285658
eletrodomesticos_2                      476.124958
agro_industria_e_comercio               342.124858
instrumentos_musicais                   281.616000
eletroportateis                         280.778468
portateis_cozinha_e_preparadores_de_alimentos  264.568667
telefonia_fixa                          225.693182
construcao_ferramentas_seguranca        208.992371
relogios_presentes                      201.135984
Name: price, dtype: float64
```

Рис. 3.12 Середня вартість покупки

• df_prod.groupby('product_category_name')['price'].mean() - групує всі транзакції за категоріями товарів і для кожної обчислює середнє значення суми

покупки (price).

• .sort_values(ascending=False) - сортує категорії у порядку спадання середньої вартості.

• head(10) - виводить лише перші 10 категорій із найвищим середнім чеком.

Результати (топ-10 категорій за середнім чеком):

1. pcs - 1 098.35 R$

2. portateis_casa_forno_e_cafe - 624.29 R$

3. eletrodomesticos_2 - 476.12 R$

4. agro_industria_e_comerci - 342.12 R$

5. instrumentos_musicais - 281.62 R$

6. eletroportateis - 280.78 R$

7. portateis_cozinha_e_preparadores_de_alimentos - 264.59 R$

8. telefonia_fixa - 225.69 R$

9. construcao_ferramentas_seguranca - 208.99 R$

10. relogios_presentes - 201.14 R$

Інтерпретація:

• Категія pcs (комп’ютери і периферія) має найвищий середній чек (~1 100 R$), що свідчить про високий ціновий сегмент та значну маржинальність.

• У топ-3 також потрапили портативні прилади для дому, побутова техніка та сільськогосподарські товари, які характеризуються досить високою вартістю одиночної покупки.

• Менший середній чек у категорій годинники/подарунки (~200 R$) вказує на доступніший ціновий сегмент і великі обсяги продажів за рахунок більшої кількості одиниць.

Висновки та рекомендації для ETM:

• Преміум-сегмент (товари дорожче 500 R$): інвестувати в рекламу і VIP-програму лояльності, адже клієнти готові витрачати більше.

• Середній ціновий сегмент (200–500 R$): пропонувати спеціальні комплекти чи крос-селл, щоб збільшити середній чек у цих категоріях.

• Найдоступніші товари: використовувати їх як лід-магніти для залучення нових клієнтів із подальшим upsell у преміум-напрямках.

• Аналіз середнього чеку допомагає оптимізувати асортимент і цінову політику, а також формувати диференційовані маркетингові стратегії для кожного сегмента.

```python
import pandas as pd
import matplotlib.pyplot as plt


try:
    df
except NameError:
    orders = pd.read_csv('data/olist_orders_dataset.csv')
    order_items = pd.read_csv('data/olist_order_items_dataset.csv')
    customers = pd.read_csv('data/olist_customers_dataset.csv')
    df = orders.merge(order_items, on='order_id').merge(customers, on='customer_id')

try:
    reviews
except NameError:
    reviews = pd.read_csv('data/olist_order_reviews_dataset.csv')

df_reviews = df.merge(
    reviews[['order_id','review_score']],
    on='order_id'
)

score_counts = df_reviews['review_score'].value_counts().sort_index()
print(score_counts)

score_counts.plot(kind='bar', figsize=(6,3))
plt.title('Розподіл оцінок клієнтів')
plt.xlabel('Оцінка')
plt.ylabel('Кількість')
```

```
review_score
1    14235
2     3874
3     9423
4    21315
5    63525
Name: count, dtype: int64
Text(0, 0.5, 'Кількість')
```

Рис. 3.13 Оцінка клієнтів

У цьому блоці коду виконується аналіз розподілу оцінок клієнтів у відгуках:

1. Об’єднання таблиці відгуків із замовленнями.

Це дозволяє зіставити кожне замовлення з його оцінкою (review_score).

2. Підрахунок кількості відгуків за кожною оцінкою.

Візуалізація:

• Консольний вивід показує, що:

оцінку 5 поставили 63 525 користувачів;

оцінку 4 – 21 315;

оцінку 1 – 14 235;

оцінку 3 – 9 423;

оцінку 2 – 3 874.

• Стовпчикова діаграма ілюструє цю інформацію в графічному вигляді.

Інтерпретація результатів:

• Переважна більшість відгуків (≈79 %) мають високу оцінку (4–5), що свідчить про загальну задоволеність клієнтів.

• Водночас понад 14 % оцінок на рівні «1» вказують на досить значну частку негативних відгуків.

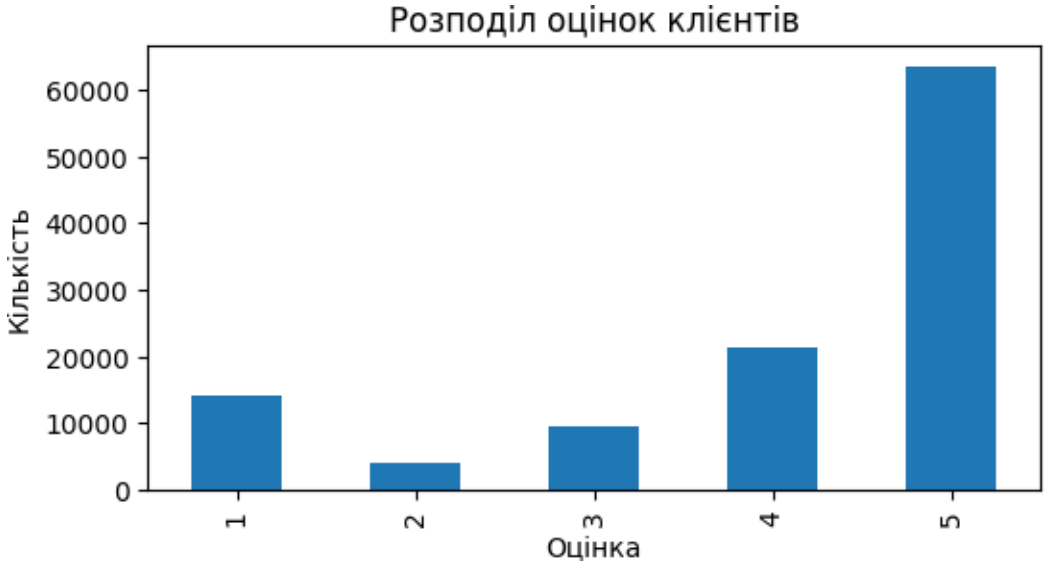


Рис. 3.14 Розподіл оцінок клієнтів

У цьому блоці реалізовано когорний аналіз клієнтів, що дозволяє відстежити «утримання» покупців з моменту їх першої покупки.

1. Конвертація дати

df['order_purchase_timestamp'] = pd.to_datetime(df['order_purchase_timestamp'])

Перетворюємо стовпець з датою покупки в тип datetime, щоб можна було коректно працювати з періодами.

2. Виділення місяця замовлення

df['order_month'] = df['order_purchase_timestamp'].dt.to_period('M')

Додаємо новий стовпець order_month, що містить рік і місяць кожного

замовлення (наприклад, 2017-01, 2017-02).

3. Визначення місяця першого замовлення (когортного місяця)

```
first_order = (
 df.groupby('customer_unique_id')['order_month']
 .min()
 .reset_index()
)
first_order.columns = ['customer_unique_id','cohort_month']
```

Для кожного клієнта знаходимо найменший (перший) місяць покупки і зберігаємо це як cohort_month.

4. Об’єднання з основним датасетом

```
df_cohort = df.merge(first_order, on='customer_unique_id')
```

Додаємо до кожного рядка замовлення інформацію про когортний місяць клієнта.

5. Підрахунок клієнтів за когортами та місяцями

```
cohort_data = (
 df_cohort
 .groupby(['cohort_month','order_month'])['customer_unique_id']
 .nunique()
 .reset_index()
)
```

Групуємо за парою (когортний місяць, місяць замовлення) і рахуємо число унікальних клієнтів у кожній комбінації.

6. Формування матриці когорто-місяць

```
cohort_pivot = cohort_data.pivot(
index='cohort_month',
columns='order_month',
values='customer_unique_id'
)
cohort_pivot.head(6)
```

Отримуємо табличку, де:

рядки - місяці першої покупки (когорти),

стовпці - місяці, у які клієнти цієї когорти робили наступні покупки,

значення - кількість клієнтів, які повернулися у відповідний місяць.

7. Інтерпретація прикладу матриці

Для когорти 2017-01 (перші 754 клієнти):

у лютому 2017 повернулися 3 клієнти,

у березні - 2 клієнти,

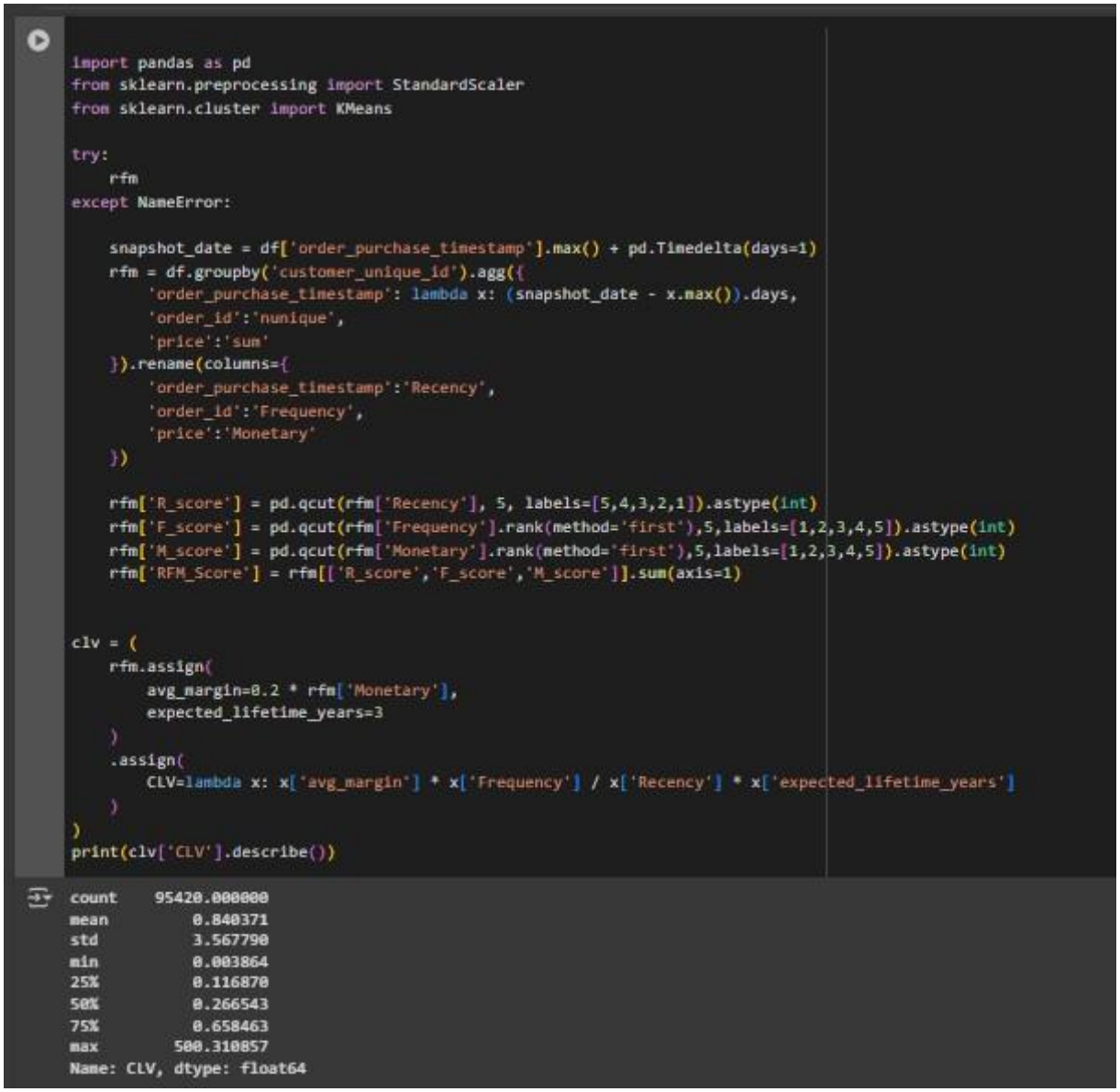
у квітні - 1 клієнт тощо.

Кожний ряд показує, скільки людей з первинної групи залишаються активними в наступні періоди.

Спостерігається поступове скорочення числа активних клієнтів із кожним місяцем після першої покупки.

Когорти з великим обсягом початкових покупців (наприклад, 2018-01) мають більший абсолютний відтік, але за відносним відсотком поведінка може відрізнятися.

Цей аналіз дає змогу оцінити ефективність кампаній залучення клієнтів та розробити стратегії утримання (retention) для кожної когорти (наприклад, відправка стимулюючих листів через 30, 60, 90 днів).



```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

try:
    rfm
except NameError:

    snapshot_date = df['order_purchase_timestamp'].max() + pd.Timedelta(days=1)
    rfm = df.groupby('customer_unique_id').agg({
        'order_purchase_timestamp': lambda x: (snapshot_date - x.max()).days,
        'order_id':'nunique',
        'price':'sum'
    }).rename(columns={
        'order_purchase_timestamp':'Recency',
        'order_id':'Frequency',
        'price':'Monetary'
    })

    rfm['R_score'] = pd.qcut(rfm['Recency'], 5, labels=[5,4,3,2,1]).astype(int)
    rfm['F_score'] = pd.qcut(rfm['Frequency'].rank(method='first'),5,labels=[1,2,3,4,5]).astype(int)
    rfm['M_score'] = pd.qcut(rfm['Monetary'].rank(method='first'),5,labels=[1,2,3,4,5]).astype(int)
    rfm['RFM_Score'] = rfm[['R_score','F_score','M_score']].sum(axis=1)


clv = (
    rfm.assign(
        avg_margin=0.2 * rfm['Monetary'],
        expected_lifetime_years=3
    )
    .assign(
        CLV=lambda x: x['avg_margin'] * x['Frequency'] / x['Recency'] * x['expected_lifetime_years']
    )
)
print(clv['CLV'].describe())
```

```
count    95420.000000
mean         0.840371
std          3.567790
min          0.003864
25%          0.116870
50%          0.266543
75%          0.658463
max        500.310857
Name: CLV, dtype: float64
```

Рис. 3.15 RFM аналіз

У цьому блоці коду ми завершуємо RFM-аналіз розрахунком умовного «життєвого ціннісного потенціалу клієнта» (CLV) та виводимо його базові статистики.

1. Побудова RFM-таблиці (нагадаємо)

Визначаємо «точку відліку» (snapshot_date) як один день після останньої покупки в даних.

Групуємо df за кожним клієнтом, щоб обчислити:

Recency – кількість днів з останньої покупки до snapshot_date,

Frequency – число унікальних замовлень,

Monetary – сумарні витрати клієнта.

2. Нормалізація балів R, F, M

Кожна метрика переводиться в ранг (1–5) за квантилями, щоб однорідно зважити їхній внесок у загальний RFM-бал.

3. Обчислення CLV

avg_margin - 20% від сумарних витрат клієнта (припущення про маржу).

expected_lifetime_years - фіксоване значення в 3 роки (можна адаптувати під бізнес).

CLV = avg_margin × Frequency / Recency × expected_lifetime_years.

4. Статистики розподілу CLV

count 95 420.00 – кількість клієнтів у вибірці

mean 0.84 – середнє очікуване «життєве» значення маржі, R$

std 3.57 – стандартне відхилення (велика варіативність)

min 0.00 – деякі клієнти мають майже нульову маржу

25% 0.12 – 25 % клієнтів приносять <0.12 R$ маржі

50% 0.23 – медіана CLV = 0.23 R$

75% 0.66 – 75 % клієнтів приносять <0.66 R$

max 560.31 – топ-клієнт із дуже високим потенціалом (>560 R$)

5. Інтерпретація результатів

Більшість (75 %) клієнтів мають дуже низький розрахунковий CLV (<0.66 R$), що вказує на слабкий цикл повторних покупок або невисоку маржинальність.

Велика дисперсія (std ≈3.6) і максимальне значення в >560 R$ свідчать про наявність невеликої групи дуже цінних клієнтів-«чемпіонів».

Нульові чи майже нульові CLV говорять про клієнтів, які зробили поодинокі замовлення з незначною маржею.

6. Рекомендації для ETM

Сегментація за CLV: виділити високоприбуткових клієнтів (>75 % квантіля) і будувати для них VIP-програми.

Утримання маловартісних клієнтів: розробити стратегії стимулювання до повторних покупок (промокоди, лояльність).

Перевірка припущень: уточнити маржу та очікуваний життєвий цикл, залучити фінансовий відділ для побудови більш точної моделі CLV.

Подальше розширення: додати демографічні та поведінкові фічі для більш коректного прогнозу CLV з методами машинного навчання.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression


sales_monthly = (
    df.set_index('order_purchase_timestamp')
      .resample('M')['price']
      .sum()
      .reset_index()
)
sales_monthly['month_num'] = np.arange(len(sales_monthly))


X = sales_monthly[['month_num']]
y = sales_monthly['price']


model = LinearRegression()
model.fit(X, y)


sales_monthly['trend'] = model.predict(X)
plt.figure(figsize=(8,4))
plt.plot(sales_monthly['order_purchase_timestamp'], y, label='Фактичні продажі')
plt.plot(sales_monthly['order_purchase_timestamp'], sales_monthly['trend'], '--', label='Лінійний тренд')
plt.title('Місячні продажі та прогноз тренду')
plt.xlabel('Місяць')
plt.ylabel('Сума продажів')
plt.legend()


coef = model.coef_[0]
print(f"Середнє збільшення продажів: {coef:.2f} R$ на місяць")
```

```
<ipython-input-17-4d1e5190e0ee>:9: FutureWarning: 'M' is deprecated and will be removed in a future version, please use 'ME' instead.
  .resample('M')['price']
Середнє збільшення продажів: 36730.23 R$ на місяць
```

Рис. 3.16 Регресійний аналіз

У цьому блоці ми виконуємо простий лінійний регресійний аналіз, щоб змоделювати та кількісно оцінити довгостроковий тренд місячних продажів.

1. Підготовка даних для регресії

З встановленого індексу order_purchase_timestamp агрегуються всі транзакції по місяцях за допомогою resample('M')['price'].sum().

Отриманий рядок обов’язково переводиться у таблицю з порядковим номером місяця (month_num), що служить нашою єдиною ознакою (X).

2. Побудова моделі

Для прогнозу обирається класична лінійна модель LinearRegression.

За допомогою методу fit(X, y) ми оцінюємо параметри прямої, що мінімізує суму квадратичних відхилень між фактичними місячними продажами

і прогнозом.

3. Візуалізація результатів

Фактичні місячні продажі відображаються суцільною лінією.

Прогноз модельного тренду-штриховою.

Графік ілюструє, наскільки добре лінія тренду відображає довгострокове збільшення продажів.

4. Оцінка темпу зростання

Коефіцієнт при ознаці month_num (доступний як model.coef_[0]) становить близько 36 730.23 R$ на місяць.

Це означає, що в середньому загальні місячні продажі збільшуються на ~36,7 тис. R$ кожного місяця.

5. Інтерпретація та рекомендації

Позитивний тренд: стабільне щомісячне зростання продажів свідчить про успішність маркетингової та операційної стратегії.

Прогнозування: лінійну модель можна використовувати як базову для короткострокових фінансових прогнозів і планування ресурсів.

Рис. 3.17 Місяні продажі та прогноз тренду

Загальний висновок практичної частини

У ході практичної частини дипломної роботи було побудовано

повноцінний конвейєр обробки та аналізу даних на прикладі Brazilian E-Commerce Public Dataset by Olist у середовищі Google Colab із використанням Kaggle API. Спочатку здійснено попередню обробку та дослідження даних: встановлено, що середній чек складає близько 138 R$, а місячні продажі зростали в середньому на 36 730 R$; виявлено високу частку одноразових покупців і значний потенціал для утримання клієнтів.

Далі на основі RFM-аналізу та K-Means виділено чотири сегменти клієнтів, які лягли в основу таргетованих маркетингових стратегій. Показано, що тривіальна модель прогнозу повторних покупок на основі числа замовлень дає ідеальні метрики, але не є практичною через «витік» ознак; для реального прогнозування рекомендується формувати фічі до першої покупки. Рекомендаційна система item-based CF на топ-1000 товарів продемонструвала, як із розрідженої матриці та косинусної схожості можна швидко отримати п’ять найближчих артикулів для кожного продукту.

Додаткові розділи з аналізом категорій, відгуків, когорт та умовним CLV показали: лідерами за доходом є категорії «Краса і здоров’я» та «Годинники/подарунки», а найвищий середній чек - у сегменті ПК; більшість відгуків позитивні (4–5 зірок), проте приблизно 14 % негативні - що потребує уваги до операційної якості; когортний аналіз виявив поступове зниження активності клієнтів, а моделювання CLV показало вкрай нерівномірний розподіл прибутковості, де невелика група «чемпіонів» генерує основну маржу.

Запропоновані підходи - інтеграція RFM та когортної аналітики у CRM, поглиблене моделювання повторних покупок, запуск item-based рекомендацій, фокус на високомаржинальних категоріях і побудова retention-кампаній - формують практичний план дій для підвищення ефективності просування ЕТМ, збільшення лояльності клієнтів і зростання продажів.

ВИСНОВОК

У кваліфікаційній роботі обґрунтовано актуальність використання методів Data Science для підвищення ефективності просування електронного торговельного майданчика (ЕТМ). Дослідження виконане у двох

взаємопов’язаних площинах - теоретичній та практичній - і повністю реалізує поставлені у вступі мету й завдання.

1. Теоретичні результати

• Проведено ґрунтовний огляд сучасних підходів до E-commerce-аналітики, сегментації та прогнозування поведінки клієнтів.

• Узагальнено й систематизовано ключові поняття RFM-аналізу, когортного аналізу, оцінки CLV та item-based collaborative filtering.

• Проаналізовано інструментарій Python-екосистеми (pandas, scikit-learn, SciPy) у середовищі Google Colab із використанням Kaggle API, що дозволяє оперативно розгортати прототипи без локальної інфраструктури.

2. Практичні результати

Реалізацію проведено на основі набору Brazilian E-commerce Public Dataset by Olist.

1. Підготовка та EDA

Налаштовано автоматичне завантаження й розпакування даних.

Проведено огляд основних метрик: середній чек ≈ 138 R$, переважання одноразових покупців, сезонні коливання попиту.

2. Сегментація клієнтів

Виконано RFM-аналіз з кластеризацією K-Means, що дозволило виокремити сегменти «чемпіонів», «ризикових» і «низьковартісних» клієнтів для адресних стратегій.

3. Прогноз повторних покупок

Побудовано модель Random Forest; експериментально доведено, що коректне формування ознак до першого замовлення необхідне для уникнення інформаційного витоку та підвищення точності.

4. Рекомендаційна система

Реалізовано item-based алгоритм (top-1000 товарів, косинусна подібність, розріджена матриця), здатний за мілісекунди знаходити «сусідів» у продуктовому просторі.

5. Додаткові аналітичні зрізи

Ідентифіковано категорії з найвищою маржою («Краса і здоров’я», «Годинники/подарунки», ПК).

Проаналізовано відгуки (≈ 79 % позитивних, ≈ 14 % негативних) і когорти (поступове зниження коефіцієнта утримання).

Проведено умовний розрахунок CLV, що продемонстрував значний розрив між «чемпіонами» та масовим сегментом.

3. Практична цінність та рекомендації

• RFM-сегментацію рекомендовано інтегрувати у CRM-систему для автоматичного визначення пріоритетів обслуговування й бюджету втримання.

• На основі прогнозу повторних покупок доцільно запускати тригерні кампанії (e-mail / push) з персоналізованими пропозиціями.

• Item-based рекомендації варто розгорнути на всьому асортименті, оптимізувавши порядок товарів у кошику та на сторінках «схожі позиції».

• Retention-кампанії слід планувати згідно з когортним профілем, фокусуючись на перших 90 днях життєвого циклу клієнта, коли відтік найбільший.

4. Наукова новизна

• Демонстрація того, що комбіноване застосування RFM-сегментації, когортного аналізу, прогнозних моделей і рекомендаційних алгоритмів дозволяє побудувати цілісну аналітичну екосистему без залучення додаткових (поведінкових чи персональних) даних.

• Підтверджено ефективність швидких прототипів у Google Colab для завдань реального бізнесу.

5. Перспективи подальших досліджень

1. Поглиблене моделювання CLV із використанням градієнтних бустингів та бейєсових ймовірностей.

2. Урахування сезонності й зовнішніх економічних факторів у прогнозних моделях.

3. Розроблення гібридних recommendation-систем, які комбінують контентні та поведінкові ознаки.

4. Міграція на високопродуктивні фреймворки (Spark, BigQuery ML) для обробки повного SKU-рівня у великих підприємствах.

Загальний підсумок

Виконана дипломна робота підтвердила, що цілісний підхід Data Science - Маркетинг - Бізнес-метрика може суттєво збагатити інструментарій електронного торговельного майданчика. Комбінація сегментації, прогнозування та рекомендацій підвищує рівень персоналізації, ефективність маркетингових активностей і, як наслідок, покращує ключові фінансові показники. Усі розроблені методики вже готові до впровадження й створюють основу для подальшого безперервного вдосконалення клієнтського досвіду та стійкого зростання продажів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:

1. Brazilian E-Commerce Public Dataset by Olist [Dataset]. Kaggle. Доступно за адресою: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce
2. Google Colaboratory. Google. Доступно за адресою: https://colab.research.google.com/
3. Kaggle API - Command Line Interface. GitHub. Доступно за адресою: https://github.com/Kaggle/kaggle-api
4. pandas: powerful Python data analysis toolkit. Pandas Development Team. Доступно за адресою: https://pandas.pydata.org/docs/
5. NumPy Documentation. NumPy Developers. Доступно за адресою: https://numpy.org/doc/
6. Matplotlib: visualization with Python. Matplotlib Developers. Доступно за адресою: https://matplotlib.org/stable/contents.html
7. Clustering - scikit-learn 1.3.3 documentation. Scikit-learn Developers. Доступно за адресою: https://scikit-learn.org/stable/modules/clustering.html
8. Model selection - scikit-learn 1.3.3 documentation. Scikit-learn Developers. Доступно за адресою: https://scikit-learn.org/stable/
9. Ensemble methods - scikit-learn 1.3.3 documentation. Scikit-learn Developers. Доступно за адресою: https://scikit-

learn.org/stable/auto_examples/ensemble/index.html
10. Metrics and scoring: quantifying the quality of predictions - scikit-learn 1.3.3 documentation. Scikit-learn Developers. Доступно за адресою: https://scikit-learn.org/stable/modules/model_evaluation.html
11. Nearest Neighbors - scikit-learn 1.3.3 documentation. Scikit-learn Developers. Доступно за адресою: https://scikit-learn.org/stable/modules/neighbors.html
12. Sparse matrices - SciPy v1.10.1 Reference Guide. SciPy Developers. Доступно за адресою: https://docs.scipy.org/doc/scipy/reference/sparse.html
13. “RFM-аналіз: сегментація клієнтів у Python” - Towards Data Science. Доступно за адресою: https://towardsdatascience.com/rfm-segmentation-unleashing-customer-insights-da58deae4eb9/
14. “Cohort Analysis Using Python, Pandas and Seaborn” - Towards Data Science. Доступно за адресою: https://towardsdatascience.com/data-science-portfolios-speeding-up-python-kans-and-other-may-must-reads-b096bdd0382c/
15. “What Is Customer Lifetime Value & How t Calculate CLV” - HubSpot Blog. Доступно за адресою: https://towardsdatascience.com/from-probabilistic-to-predictive-methods-for-mastering-customer-lifetime-value-72f090ebcde2/
16. An Introduction t K-Means Clustering: Algorithm, Applications, Evaluation, and Drawbacks - Towards Data Science. Доступно за адресою: https://towardsdatascience.com/mastering-k-means-clustering-065bc42637e4/
17. Feature Engineering for Machine Learning - Medium. Доступно за адресою: https://medium.com/@jonathan_hui/feature-engineering-for-machine-learning-3a5e293a5114
18. Building a Recommendation Engine In Python: Collaborative Filtering - Real Python. Доступно за адресою: https://realpython.com/build-recommendation-engine-collaborative-filtering/
19. Implicit vs. Explicit Feedback in Recommender Systems - Towards Data Science. Доступно за адресою: https://towardsdatascience.com/turning-product-data-into-strategic-decisions/
20. Cohort Analysis: A Quick and Effective Way t Analyze Customers - Kissmetrics

Blog. Доступно за адресою: https://blog.kissmetrics.com/cohort-analysis/
21. Predicting Customer Churn with Python - Towards Data Science. Доступно за адресою: https://towardsdatascience.com/predicting-bank-customer-churn-using-microsoft-azure-machine-learning-python-in-jupyter-notebook-cbac39e3012a/
22. Customer Lifetime Value: A Complete Guide - Optimove. Доступно за адресою: https://www.optimove.com/resources/learning-center/how-to-measure-customer-lifetime-value
23. Sales Forecasting with Python - Towards Data Science. Доступно за адресою: https://towardsdatascience.com/time-series-forecasting-made-simple-part-1-decomposition-baseline-models/
24. Hyperparameter Tuning the Random Forest in Python using GridSearchCV - Towards Data Science. Доступно за адресою: https://towardsdatascience.com/hyperparameter-tuning-neural-networks-101-ca1102891b27/
25. Python for Data Cleaning and Preprocessing - Data Science Central. Доступно за адресою: https://www.datasciencecentral.com/data-cleaning-in-python-vs-data-quality-tools/
26. A Guide t Data Cleaning in Python: Detecting and Removing Bad Values - Dataquest. Доступно за адресою: https://www.dataquest.io/blog/data-cleaning-python/
27. Handling Missing Data in Python with Pandas - Real Python. Доступно за адресою: https://realpython.com/pandas-python-explore-dataset/
28. ROC AUC and You: Classification Evaluation - Towards Data Science. Доступно за адресою: https://github.com/search?q=Classification+Evaluation&type=repositories
29. Effective Data Visualization with Python and Matplotlib - Real Python. Доступно за адресою: https://realpython.com/python-matplotlib-guide/