

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Навчально-науковий інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

До захисту ДОПУЩЕНО
Завідувач кафедри
_____ Оксана ТИМОЩУК
«__» _____ 20__ р.

**ДИПЛОМНА РОБОТА
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Системний аналіз і управління»
спеціальності 124 «Системний аналіз»
на тему: «Розробка системи автоматизованого керування рекламними
кампаніями в соціальних мережах»**

Виконав:

Студент IV курсу, групи КА-93
Хижняк Максим Олександрович _____

Керівник:

старший викладач, PhD Гуськова Віра Геннадіївна _____

Консультант з економічного розділу:

доцент, к.е.н, Рощина Надія Василівна _____

Консультант з нормоконтролю:

к.ф-м.н. Статкевич Віталій Михайлович _____

Рецензент:

проф., д.т.н. Бідюк Петро Іванович _____

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

Студент _____

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

Рівень вищої освіти – перший(бакалаврський)

Спеціальність – 124 «Системний аналіз»

Освітньо-професійна програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Оксана ТИМОЩУК

«__» _____ 20__ р.

Завдання

на дипломну роботу студенту

Хижняку Максиму Олександровичу

1. Тема роботи «Розробка системи автоматизованого керування рекламними кампаніями в соціальних мережах», керівник роботи старший викладач, PhD Гуськова Віра Геннадіївна, затверджені наказом по університету від «30» травня 2023р. № 2065-с.
2. Строк подання студентом роботи 13.06.2023 р.
3. Вихідні дані до роботи: набір даних про роботу рекламних кампаній у соціальних мережах
4. Зміст роботи: аналіз предметної області, огляд існуючих підходів, навчання моделей, створення програмного продукту, аналіз результатів
5. Перелік ілюстративного матеріалу: презентація для захисту дипломної роботи, яка складається з 12 слайдів, які містять у собі постановку задачі, огляд предметної області, актуальність, огляд роботи і результати роботи моделей, а також технологічний стек необхідний для створення програмного продукту

6. Консультанти розділів роботи:

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

7. Дата видачі завдання: 17.04.2023

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Затвердження теми бакалаврської дипломної роботи, огляд літературних джерел.	17.04.2023 – 23.04.2023	Виконав
2.	Перший розділ. Дослідження предметної області, аналіз існуючих підходів.	24.04.2023 – 30.04.2023	Виконав
3.	Другий розділ. Опис математичних основ роботи, огляд різних регресійних моделей.	01.05.2023 – 07.05.2023	Виконав
4.	Третій розділ. Розробка програмного продукту, проведення аналізу отриманих результатів.	08.05.2023 – 14.05.2023	Виконав
5.	Четвертий розділ. Проведення функціонально-вартісного аналізу розробленої практичної частини.	15.05.2023 – 21.05.2023	Виконав
6.	Оформлення роботи.	22.05.2023 – 28.05.2023	Виконав

Студент

Максим ХИЖНЯК

Керівник

Віра ГУСЬКОВА

РЕФЕРАТ

Тема: “Розробка системи автоматизованого керування рекламними кампаніями в соціальних мережах”

Дипломна робота: 97 с., 25 формул, 10 табл., 29 рис., 2 додатки., 25 джерел.

МАШИННЕ НАВЧАННЯ, ПОЛІНОМІАЛЬНА РЕГРЕСІЯ, ЗАДАЧА ПРОГНОЗУВАННЯ, ПРОГНОЗУВАННЯ ЕФЕКТИВНОСТІ РОБОТИ РЕКЛАМНИХ КАМПАНІЙ

Об’єкт дослідження: впровадження програмного рішення яке дозволить компаніям ефективно управляти рекламними кампаніями в соціальних мережах

Предмет дослідження: розробка і дослідження системи автоматизованого керування рекламними кампаніями в соціальних мережах

Мета дослідження: відновити залежність між маркетинговими метриками що характеризують рекламні кампанії, на основі цього розробити систему яка оптимізує доходи компанії приймаючи рішення щодо подальшої роботи рекламної кампанії

Використані моделі: поліноміальна модель регресії

Отримані результати: метрика $R_squared$ у створеної моделі має значення 94% на тестових даних. Також метрика MSE дорівнює 96.5.

ABSTRACT

Topic: "Development of a system for automated management of advertising campaigns in social networks"

The thesis contains 97 p., 25 formulas., 10 tab., 29 pict., 2 appendices, 25 sources.

MACHINE LEARNING, THE PROBLEM OF FORECASTING, FORECASTING THE EFFICIENCY OF ADVERTISING CAMPAIGNS

The object of the research: implementation of a software solution that will allow companies to effectively manage advertising campaigns in social networks

The main subject of the research development and research of an automated management system for advertising campaigns in social networks

The purpose of the study: to restore the dependency between marketing metrics that characterize advertising campaigns and, based on this, to develop a system that optimizes the company's revenue when deciding on the further operation of the advertising campaign

Used models: polynomial regression model

The results obtained: the R_squared metric of the created model has a value of 94% on the test data. Also, the MSE metric is 96.5.

ЗМІСТ

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ	8
ВСТУП.....	9
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1 Актуальність проблеми.....	12
1.2 Існуючі випадки реалізації автоматизації маркетингу в соціальних мережах.....	16
1.3 Інструменти для роботи з рекламними кампаніями.....	17
1.4 Оцінка ефективності рекламних кампаній.....	18
1.4.1 Пожиттєва цінність клієнта	19
1.4.2 Ціна за покупку	20
1.5 Постановка задачі	21
1.6 Висновки до першого розділу	22
2 МАТЕМАТИЧНІ ОСНОВИ РОБОТИ.....	23
2.1 Алгоритми машинного навчання	23
2.2 Навчання з учителем	24
2.3 Типи регресій	27
2.4 Початковий аналіз даних	29
2.4.1 Аналіз даних. Огляд, візуалізація і кореляція.....	31
2.5 Поліноміальна регресія	32
2.6 Перенавчання	33
2.7 Оцінка ефективності роботи моделі	35
2.9 Висновки до другого розділу.....	39
3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ І АНАЛІЗ РЕЗУЛЬТАТІВ ...	40

	7
3.1 Проблематика задачі	40
3.2 Архітектура системи	42
3.3 Реалізація прийняття рішень	43
3.4 Реалізація системи сповіщень	44
3.5 Робота з даними	44
3.6 Віддалений сервер	45
3.7 Програмна оболонка.....	48
3.8 Результати роботи.....	51
3.9 Висновки до третього розділу	53
4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ	54
4.1 Постановка задачі проектування.....	55
4.2 Обґрунтування функцій програмного продукту.....	55
4.3 Обґрунтування системи параметрів програмного продукту	58
4.4 Аналіз експертного оцінювання параметрів	61
4.5 Аналіз рівня якості варіантів реалізації функцій.....	65
4.6 Економічний аналіз варіантів розробки ПП	67
4.7 Вибір кращого варіанту ПП техніко-економічного рівня	72
4.8 Висновки до четвертого розділу	73
ВИСНОВКИ	75
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	76
ДОДАТОК А ІЛЮСТРАТИВНІ МАТЕРІАЛИ	79
ДОДАТОК Б ЛІСТИНГ ПРОГРАМИ.....	86

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

ФВА – функціонально-вартісний аналіз;

ПП – програмний продукт;

МН – машинне навчання;

ВМ – виконавчий маркетинг;

ВСТУП

У сучасному світі соціальні мережі відіграють вельми вагому роль як у нашому повсякденному житті, так і в бізнес-середовищі. Вони представляють собою потужний інструмент для комунікації, спілкування та обміну інформацією між користувачами, але їх значимість лише зростає з часом.

Особливо цікавим аспектом соціальних мереж для безлічі організацій є можливість проведення рекламних кампаній, спрямованих на просування власних товарів і послуг, залучення нових клієнтів і підвищення свого впливу на цифровому ринку. Відтак, соціальні мережі стають не тільки майданчиком для взаємодії з аудиторією, але і ефективним засобом побудови бренду та збільшення його відомості.

Завдяки широкому охопленню соціальних мереж, компанії мають можливість досягти значно більшої кількості людей, ніж за допомогою традиційних каналів маркетингу. Крім того, соціальні мережі дозволяють точно налаштувати таргетування, спрямовуючи рекламу на конкретну аудиторію за інтересами, поведінкою, демографічними характеристиками та багатьма іншими параметрами.

Використання соціальних мереж у бізнесі вимагає наявності стратегії, професійного підходу та постійного моніторингу результатів. Ефективне управління соціальними медіа дозволяє створити позитивний імідж компанії, збільшити впізнаваність бренду і залучити активну аудиторію, яка може стати лояльними клієнтами. Зокрема, відповідність контенту, взаємодія зі споживачами та підтримка цікавих дискусій є ключовими факторами успіху в соціальних мережах.

Отже, не можна недооцінювати вагу соціальних мереж у контексті сучасного світу. Вони виконують низку важливих функцій, що сприяють зміцненню зв'язку з аудиторією, просуванню брендів і залученню нових

клієнтів. Відтак, активне і компетентне використання соціальних мереж є необхідністю для сучасних організацій, які прагнуть успіху в цифровій епосі.

З введенням соціальних медіа реклами, організації стикаються з викликами, що вимагають значних зусиль, часу та ресурсів для планування, виконання та аналізу рекламних кампаній. Однак, зі зростанням обсягів даних та складності кампаній виникає необхідність розробки систем, які забезпечують автоматизоване керування рекламними кампаніями в соціальних мережах.

Цей процес вимагає ще більших зусиль та ресурсів, оскільки організації повинні забезпечити оптимальні результати в умовах постійно зростаючої конкуренції і складності соціальних мереж. Але завдяки автоматизації керування рекламними кампаніями, організації можуть зосередитися на стратегічному плануванні, творчому процесі та взаємодії зі своєю аудиторією.

Автоматизовані системи для керування рекламою в соціальних мережах забезпечують багатофункціональний підхід до управління кампаніями, включаючи розміщення рекламних матеріалів, спрямоване таргетування, моніторинг результатів та аналіз ефективності. Вони дозволяють автоматично оптимізувати рекламні бюджети, встановлювати стратегії пропозицій, вимірювати показники продуктивності та реагувати на зміни у реальному часі.

Застосування автоматизованих систем управління рекламою в соціальних мережах допомагає організаціям ефективніше використовувати свої ресурси, збільшувати ефективність кампаній та досягати бажаних результатів. Вони дозволяють зосередитися на стратегічних завданнях, спілкуванні зі споживачами та створенні цінного контенту, забезпечуючи успішну присутність організації в соціальних медіа.

Метою даної дипломної роботи було не лише проаналізувати наявні системи автоматизованого керування рекламними кампаніями в соціальних мережах, але й розробити оптимальну систему, що максимально сприяє організаціям у ефективному використанні можливостей соціальних мереж для досягнення своїх маркетингових цілей. Запропонована система не лише надає

зручний вигляд інформації про свою роботу, але й допомагає прискорити процес прийняття рішень щодо планування майбутніх заходів, спрямованих на збільшення доходів компанії, яка обирає використання цієї системи.

Розроблена система автоматизованого керування рекламними кампаніями забезпечує комплексний підхід до управління рекламою в соціальних мережах. Вона забезпечує зручний інтерфейс для моніторингу та аналізу рекламних кампаній, надаючи детальну статистику та звіти, які допомагають організаціям зрозуміти ефективність їхніх зусиль і внести необхідні корективи для досягнення кращих результатів.

Крім того, система пропонує функції автоматичного оптимізування рекламних кампаній на основі зібраної інформації та аналізу даних. Вона використовує розумні алгоритми та технології, щоб підкреслити потенційно успішні напрями реклами, забезпечуючи максимальний результат за вкладеними зусиллями.

Загалом, запропонована система автоматизованого керування рекламними кампаніями в соціальних мережах сприяє підвищенню ефективності маркетингових зусиль організацій, забезпечуючи оптимальний використання можливостей соціальних мереж та спрощуючи процес прийняття стратегічних рішень.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність проблеми

Продуктові компанії постійно витрачають мільйони на рекламу як у Інтернеті, так і у фізичному світі. Всі ці рекламні кампанії залежать від маркетологів, але виникають проблеми, коли через обмежену кількість даних маркетологи не можуть оцінити ефективність своїх зусиль.

Існують три ключові мотиваційні ідеї для проведення досліджень у галузі рекламних кампаній. По-перше, було б навчально дізнатися, як саме маркетологи ведуть свої рекламні кампанії. Оскільки часто вони стверджують, що маркетинг - це мистецтво, вони приймають рішення на основі інтуїції.

По-друге, було б фантастично мати машину, яка може виконувати ті самі завдання з рекламними кампаніями, що і людина. Це відкрило б потенціал для звільнення маркетологів від рутинних завдань і надало їм можливість більше зосередитися на творчості та експериментах.

Нарешті, система управління рекламними кампаніями може маєть потенціал краще розуміти ефективність реклами, ніж людина. Вона може аналізувати величезну кількість історичних даних і виявляти закономірності, які можуть залишатися непоміченими для маркетологів, тим самим ефективніше керуючи бюджетом.

У сучасному світі, для створення продукту, яким користуватимуться мільйони людей, необхідні не лише ідея та реалізація, але й маркетингові зусилля. Тому команда маркетингу відіграє важливу роль у продуктових компаніях. З більшими масштабами проекту збільшується обсяг даних, а разом з цим і ймовірність людських помилок. Щоб вирішити цю проблему, все більше компаній звертаються до розвитку автоматизації в галузі маркетингу.

Найбільш актуальною стає проблема прогнозування ефективності реклами після випуску операційної системи iOS 14.5. [1] Після цього оновлення всі додатки почали запитувати у користувачів, чи бажають вони

ділитись своїми рекламними вподобаннями (див. рисунок 1.1). Значить, користувачі мають право самостійно вирішувати, чи надавати свій рекламний ідентифікатор (IDFA).

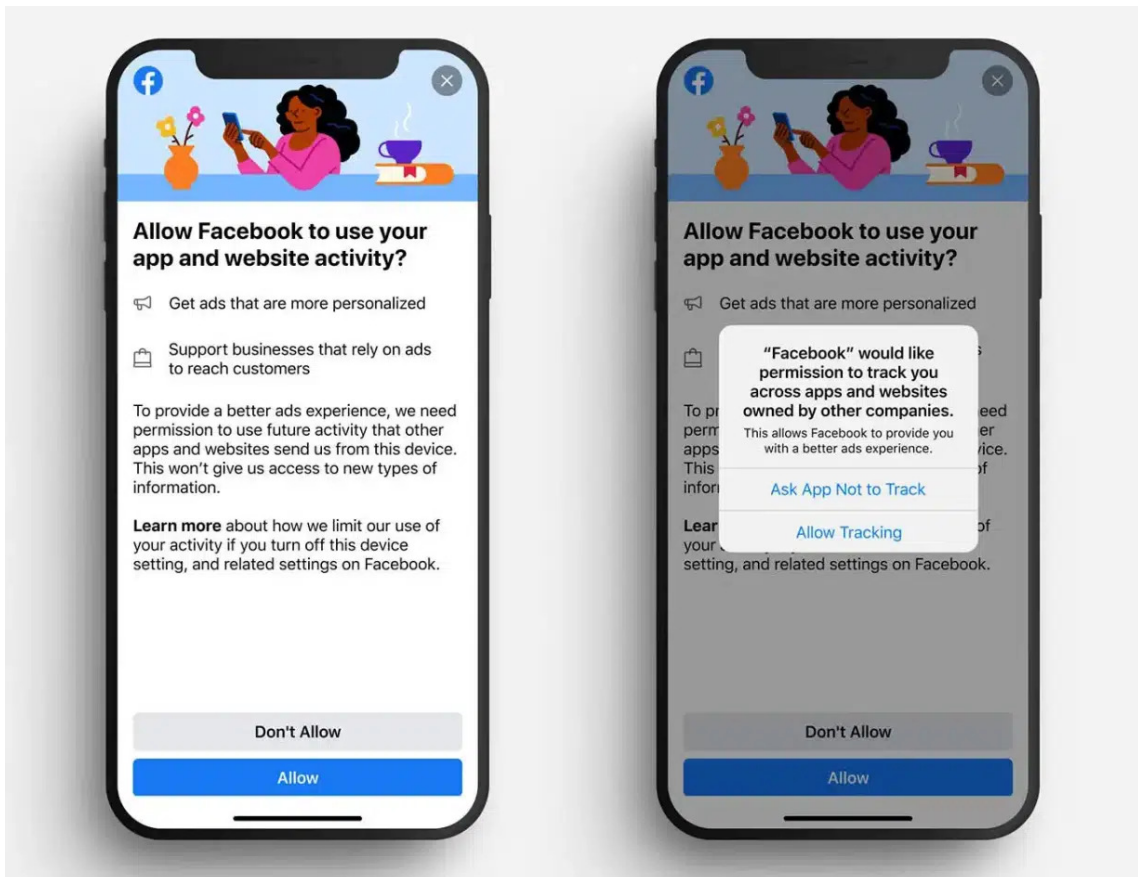


Рисунок 1.1 - Сповіщення у iOS 14.5

Згідно з даними видання [2] *roirevolution* у 2023 році лише 25% користувачів погоджуються на поширення рекламних уподобань. Таким чином лише кожен четвертий користувач дозволяє коректно відслідковувати свою рекламну історію і надає змогу коректно оцінювати якість рекламних кампаній у соціальних мережах з яких цей користувач був залучений.

Але ці дані не допомагають зрозуміти наскільки ситуація погіршилась. Для цього слід розглянути динаміку різних показників, які визначають ефективність маркетингу у компанії або бізнесі.

Для більш глибокого розуміння потрібно знати що таке метрика. Метрикою називається числова характеристика якості певного бізнес-процесу або бізнес-активності.

За даними ресурсу Insense [3] у періоді з 16.05.2021 до 12.06.2021 та з 16.05.2022 до 12.06.2022 (ідентичні проміжки у різні роки) ROAS(рентабельність витрат на рекламу) рекламних кампаній знизився на 1.69% (рис. 1.2)

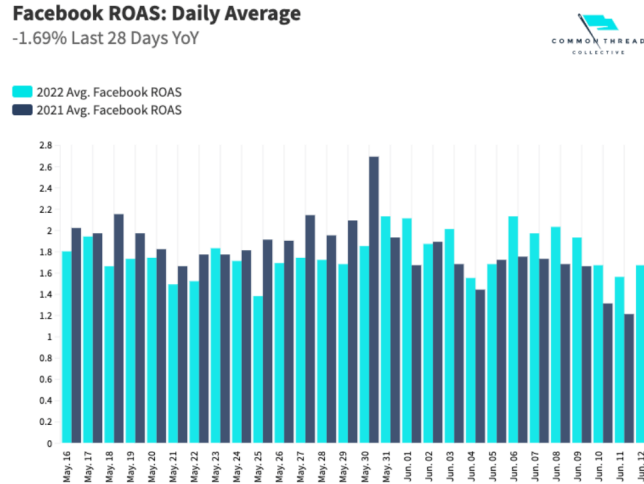


Рисунок 1.2 - Рентабельність рекламних кампаній

З того ж ресурсу дізнаємось, що середня щоденна конверсія у цільову дію знизилась на 5.87% (рис. 1.3)

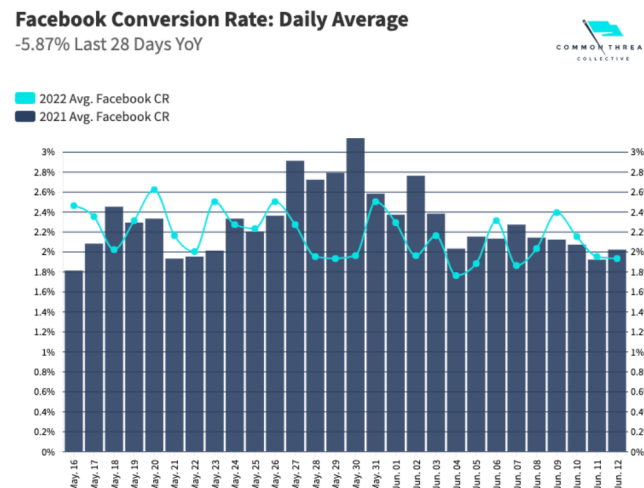


Рисунок 1.3 – Щоденна конверсія у цільову дію

Також на 21.94% зросла ціна за тисячу показів реклами цільовим користувачам соціальних мереж(рис. 1.4)

Facebook CPM: Daily Average

+21.94% Last 28 Days YoY

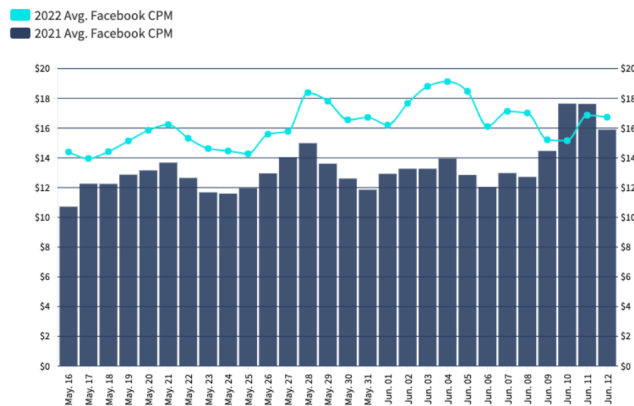


Рисунок 1.4 – Ціна за тисячу показів

Найбільше за все знизилась виручка з реклами. Середня виручка за день знизилась на 34.87% (рис. 1.5)

Facebook Revenue: Daily Average

-34.87% Last 28 Days YoY

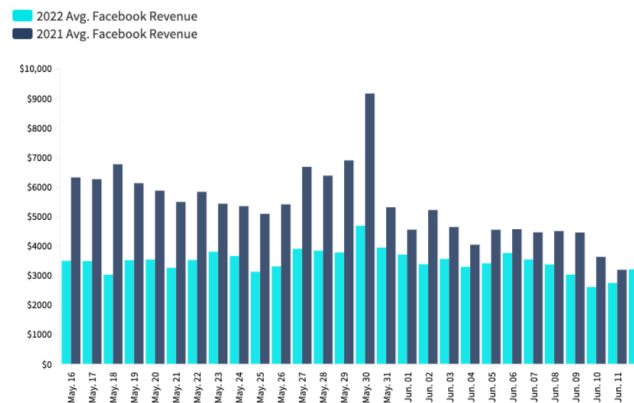


Рисунок 1.5 – Середня виручка за день

Як можна побачити, цифровий бізнес сильно постраждав від нововведень у iOS 14.5. З ресурсу Insense дізнаємось, що сфера цифрової комерції постраждала не так сильно. Виручка у ніші знизилась лише на 2.06% (рис. 1.6). Причиною цього є те, що цифрова комерція не залежить від надання користувачем своїх рекламних вподобань у застосунках. Звичайно існують винятки, як от застосунки онлайн магазинів одягу. Проте більшість інтернет магазинів існують у формі веб-сайтів, а не мобільних застосунків. Таким

чином нововведення у iOS 14.5 цілком очікувано не мали великого впливу на цей ринок.

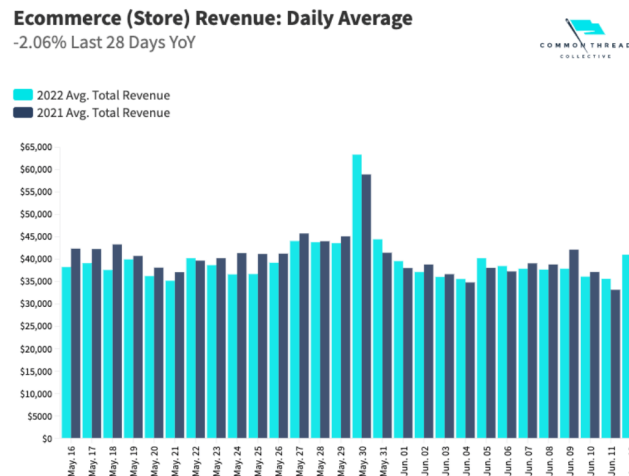


Рисунок 1.6 – Виручка у ніші цифрової комерції

1.2 Існуючі випадки реалізації автоматизації маркетингу в соціальних мережах

Чудовим прикладом успішної реалізації автоматизації роботи рекламних кампаній у соціальних мережах є історія компанії McAfee [4]. Не дивлячись на досить непогані метрики перед впровадженням автоматизації, компанія вирішила їх покращити. McAfee спеціалізується на створенні систем захисту та аналізу шкідливого програмного забезпечення. Тобто основним продуктом компанії є цифрові додатки на особисті комп'ютери і мобільні пристрої.

Головним недоліком існуючої маркетингової стратегії компанії McAfee була якість користувачів залучених з рекламних кампаній. Відповідальні за маркетинг у компанії висунули припущення, що імплементація автоматизації роботи рекламних кампаній допоможе підвищити якість користувачів.

Отже, після прийнятого рішення було зроблене наступне:

1. Проведено аналіз залучених користувачів.
2. Досліджено, що кількість користувачів цілком задовільна, але якість користувачів недостатня.
3. Створено ПЗ що виділяє найбільш якісний сегмент залучених користувачів.
4. Рекламні кампанії налаштовуються так, щоб шукати користувачів найбільш схожих на тих, що належать до сегменту з попереднього пункту.

Після введення такого рішення компанія McAfee втратила суттєву частку користувачів, які раніше залучались з рекламних кампаній, проте значно підвищила якість нових користувачів. Таким чином абсолютна кількість користувачів зменшилась, проте виручка з кожного користувача збільшилась у 4 рази згідно з даними з ресурсу [webpresence](#). [4]

Цей кейс підкреслює важливість автоматизації маркетингу у соціальних мережах для досягнення успіху в цифровому середовищі та підвищення ефективності рекламних кампаній компаній.

1.3 Інструменти для роботи з рекламними кампаніями

Найпоширенішим інструментом для створення, поширення і аналізу рекламних кампаній є Facebook Business Manager. [5] Це інструмент створений компанією Meta(в минулому Facebook), який дозволяє:

1. Створювати рекламні кампанії, слідкувати за результатами роботи рекламних кампаній.
2. Просувати рекламу з використанням моделей машинного навчання.
3. Представити рекламу великій аудиторії (рис. 1.7).

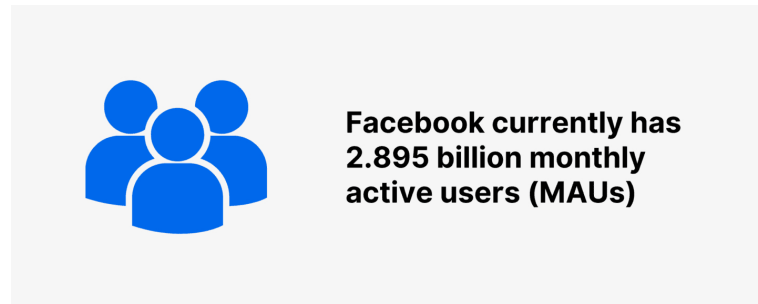


Рисунок 1.7 – Аудиторія Facebook

Окрім створення та поширення реклами, Facebook Business Manager дозволяє отримати дані у зручному форматі шляхом використання API. Це дозволяє мінімізувати ручну роботу зі збору, очистки та зведення даних. Також це дозволяє автоматизувати процес збору даних, а також надає можливість використання моделей машинного навчання у різноманітних цілях.

1.4 Оцінка ефективності рекламних кампаній

У виконавчому маркетингу є безліч метрик(змінних, які характеризують кампанію), зокрема:

- CPC – cost per click (ціна за клік).
- CPA – cost per acquisition (ціна за залучення).
- CPP – cost per purchase (ціна за покупку).
- LTV – lifetime value (пожиттєва цінність клієнта).
- CPM – cost per mille (ціна за тисячу показів реклами).
- тощо.

Кожна з цих метрик відображає характер роботи кампанії на певному етапі. Також доцільність вибору ключової метрики залежить від цілі

рекламної кампанії. Наприклад CPM відображає наскільки якісна аудиторія бачить рекламу. CPP відображає середню кількість ресурсів, які будуть витрачені на отримання однієї продажі. LTV дає змогу оцінити скільки доходу в середньому приносить один залучений клієнт. Отже, маємо великий набір метрик. Тепер слід визначити які з них є потенційно залежними, щоб приймати вигідні для бізнесу рішення.

Через те що більшість бізнесів обирають модель монетизації, яка передбачає покупку(пробної версії чи продукту), однією з шуканих змінних я вирішив обрати CPP(cost per purchase). Ця змінна дає змогу розуміти і прогнозувати скільки ресурсів в майбутньому буде витрачено на залучення одного клієнта. Іншими словами, скільки грошей повинен витратити бізнес, щоб було зроблено один продаж. Друга шукана змінна має описувати якість користувачів яких було залучено з допомогою рекламної кампанії. З точки зору бізнесу якість користувача це кількість грошей, які принесе останній. Тому другою залежною метрикою я обрав LTV. Отже залишилось знайти залежність між CPP та LTV.

1.4.1 Пожиттєва цінність клієнта

Раніше було визначено, що ми шукатимемо залежність між ціною за продажу і цінністю користувача. Проте класична метрика LTV не повністю характеризує якість залученого користувача. Найбільша соціальна мережа Facebook розподіляє кошти, що витрачаються на рекламу, між користувачами з різних країн. Щоб найбільш коректно оцінювати якість користувача, я вирішив модифікувати метрику LTV і використовувати wLTV(weighted LTV або зважена цінність користувача). Алгоритм обрахування wLTV наведений на рис 1.8

campaign_name	country_code	spend by country	country arps	total spend	spend share	weighted ARPS formula	weighted ARPS
[A]/campaign/text US		50	40	100	0.5	=country_arps * spend_share	20
[A]/campaign/text UA		30	20	100	0.3	=country_arps * spend_share	6
[A]/campaign/text RU		20	19	100	0.2	=country_arps * spend_share	3.8
						resulting ARPS	29.8

Рисунок 1.8 – Алгоритм обрахування wLTV

Формула обрахунку wLTV наведена у формулі 1.1

$$wLTV = \sum_{i=0}^n LTV_i * share_i \mid n = N(campaigns); \quad (1.1)$$

$$share_i = \frac{spend_i}{Total Spend}, i = 0..n, n = N(campaigns);$$

1.4.2 Ціна за покупку

Метрика CPP у маркетингу вказує на вартість, яку компанія понесла за одне здійснене придбання або конверсію. Вона вимірює ефективність рекламних кампаній та дозволяє оцінити, скільки коштує компанії залучення нового клієнта або виконання певної дії, наприклад, реєстрації на сайті, здійснення покупки або заповнення форми. CPP розраховується за формулою 1.2

$$CPP = \frac{Total Spend}{Purchases} \quad (1.2)$$

Метрика CPP є важливим інструментом у маркетинговому аналізі, оскільки дозволяє визначити ефективність витрат на рекламу та здійснити розрахунок рентабельності маркетингових кампаній.

1.5 Постановка задачі

Перед нами постає задача із вхідними даними у вигляді таблиці. Полями цієї таблиці є дані про реальні рекламні кампанії, що витрачали реальні ресурси.

Метою дипломної роботи є створення автономної і автоматичної системи з використанням алгоритмів машинного навчання що буде припиняти роботу нерентабельних рекламних кампаній у соціальних мережах

Задача вирішувалась у три етапи:

1. Збір даних для забезпечення можливості знаходження залежності між змінними про які йдеться у підрозділі 1.4.
2. Попередній аналіз даних(EDA) для попередньої обробки даних та знаходження аномалій які потрібно усунути і застосування різних алгоритмів регресії та порівняльний аналіз.
3. З обраною оптимальною моделлю регресії, створюється інфраструктура: база даних, віддалений сервер, система сповіщень. Тобто потрібно “зібрати” систему в єдине ціле.

Програмний продукт запроваджуватиметься у процесі ТОВ “Айті Ленд”. Таким чином компанія матиме змогу:

1. збільшити доходи;
2. матиме можливість спрямувати ресурс маркетинг менеджерів на стратегічні задачі замість ручного моніторингу;
3. своєчасно дізнаватись інформацію про ефективність реклами.

1.6 Висновки до першого розділу

Задача зводиться до знаходження залежності між ціною за продажу і цінністю залученого користувача. Іншими словами ми шукатимемо залежність між CPP і wLTV. Це дозволить компанії зберігати стабільність роботи відділу маркетингу, а також прогнозувати свої доходи.

Задача вирішуватиметься у три етапи:

1. Збір даних.
2. Дослідження даних.
3. Пошук залежності.
4. Створення системи з оповіщення про роботу рекламних кампаній.

2 МАТЕМАТИЧНІ ОСНОВИ РОБОТИ

2.1 Алгоритми машинного навчання

Алгоритми машинного навчання (Machine Learning algorithms) - це математичні моделі і методи, які дозволяють комп'ютерам автоматично "навчатися" на основі даних і виконувати завдання без явного програмування. Вони дозволяють комп'ютерам виявляти закономірності та шаблони в даних, а також здійснювати прогнози та приймати рішення на основі цих закономірностей.

Алгоритми машинного навчання можуть бути розділені на декілька категорій залежно від типу завдання, яке вони вирішують:

1. Навчання з учителем (Supervised Learning): Цей тип навчання використовує мітки або класифікації для навчання моделі. Алгоритми навчання з учителем навчаються на попередньо розмічених прикладах, де кожен приклад складається зі вхідних даних та відповідного вихідного значення. Деякі популярні алгоритми навчання з учителем включають лінійну регресію, логістичну регресію, дерева рішень, метод опорних векторів (SVM), навчання нейронних мереж і багато інших.
2. Навчання без вчителя (Unsupervised Learning): У цьому типі навчання немає міток або класифікацій. Алгоритми навчання без вчителя виявляють приховані структури та закономірності у наборі даних без попереднього визначення цільових значень. Приклади алгоритмів навчання без вчителя включають кластеризацію, зменшення розмірності (dimensionality reduction), правила асоціацій і ієрархічні моделі.
3. Підсилене навчання (Reinforcement Learning): Цей тип навчання використовується для розв'язання задач, де агент взаємодіє з оточенням і отримує винагороду або покарання за кожну дію. Агент повинен

навчитися знаходити оптимальну стратегію дій, яка максимізує отриману винагороду в довгостроковій перспективі.

4. Парні навчання (Semi-Supervised Learning): Цей тип навчання поєднує елементи навчання з учителем та навчання без вчителя. Використовуються навчальні дані з мітками, але також використовуються немарковані дані для покращення якості моделі.

Алгоритми машинного навчання застосовуються в різних галузях, таких як комп'ютерне зору, обробка природної мови, рекомендаційні системи, фінанси, медицина, транспорт, розпізнавання мови, графічне мистецтво та багато інших. Вони дозволяють використовувати великі обсяги даних для отримання цінної інформації та автоматизації складних завдань.

2.2 Навчання з учителем

Машинне навчання з учителем (Supervised Learning) - це тип машинного навчання, де модель навчається на основі попередньо розмічених прикладів, які містять як вхідні дані, так і відповідні вихідні мітки або класифікації.

У машинному навчанні з учителем, модель навчається встановлювати залежність між вхідними даними та цільовими вихідними значеннями. Приклади таких задач включають класифікацію, регресію та прогнозування.

1. Класифікація: У задачах класифікації модель навчається призначати вхідним даним одну з попередньо визначених категорій або класів. Наприклад, модель може класифікувати електронні листи як спам або

- не спам, розпізнавати зображення як кішку або собаку або прогнозувати, чи пацієнт має певну хворобу на основі медичних ознак.
2. Регресія: У задачах регресії модель навчається побудувати функцію, яка передбачає неперервне числове значення відповіді на основі вхідних даних. Наприклад, модель може передбачати ціну будинку на основі його характеристик, прогнозувати продажі на основі рекламних витрат або передбачати час доставки замовлення на основі різних факторів.

У процесі навчання з учителем, модель прогнозує вихідні значення для нових, невиданих даних, і ці прогнози порівнюються з правильними відповідями для оцінки її точності та продуктивності. Метою є побудова моделі, яка може генералізувати залежності між вхідними і вихідними даними і здатна коректно передбачати відповіді для нових прикладів.

У машинному навчанні з учителем використовуються різні алгоритми, включаючи лінійну регресію, логістичну регресію, дерева рішень, метод опорних векторів (SVM), навчання нейронних мереж та інші.

Для успішного застосування машинного навчання з учителем необхідні такі кроки:

1. Збір та підготовка даних: Спочатку потрібно зібрати вхідні дані, які включають вхідні функції (ознаки) та відповідні мітки або класифікації. Дані можуть бути структуровані (таблиці) або неструктуровані (текст, зображення, аудіо тощо). Після цього проводиться попередня обробка даних, така як очищення, видалення відсутніх значень, нормалізація тощо.
2. Вибір моделі: Другий крок - вибір алгоритму або моделі, яка буде використовуватися для навчання на даних. Вибір моделі залежить від типу задачі (класифікація або регресія), характеру даних та величини набору даних. Кожен алгоритм має свої переваги, обмеження та параметри, які можуть бути налаштовані.

3. Навчання моделі: У цьому кроці модель навчається на навчальних даних шляхом підбору оптимальних значень параметрів. Метою є мінімізація функції втрат, яка вимірює різницю між прогнозованими значеннями моделі та справжніми значеннями в навчальних даних. Для цього використовуються методи оптимізації, такі як градієнтний спуск.
4. Оцінка та валідація моделі: Після навчання моделі важливо оцінити її продуктивність та здатність до узагальнення на нові дані. Для цього використовуються метрики якості, такі як точність, розподіл класів, середньоквадратична помилка тощо. Модель може бути перевірена на тестовому наборі даних, які не використовувалися під час навчання.
5. Застосування моделі: Після успішної оцінки моделі вона може бути використана для прогнозування або класифікації нових невиданих даних. Модель може бути імплементована у виробничому середовищі для автоматизованої обробки даних та прийняття рішень.

Важливо відзначити, що машинне навчання з учителем потребує достатньо великого обсягу якісних та репрезентативних даних для ефективного навчання моделі. Також необхідно ретельно контролювати якість та збалансованість даних, уникати перенавчання (*overfitting*) та підлаштування (*underfitting*) моделі.

2.3 Типи регресій

У машинному навчанні існує кілька типів регресій, які використовуються для моделювання залежностей між вхідними та вихідними змінними. Основними типами регресій є:

1. Лінійна регресія (Linear Regression): Це один з найпростіших та найпопулярніших видів регресії. Вона моделює лінійну залежність між вхідними змінними та вихідним значенням. Лінійна регресія прагне знайти оптимальну залежність виду $y = k_1x_1 + k_2x_2 + \dots + k_px_p + \epsilon$, що найкращим чином підходить до даних. Вона може бути одновимірною (з однією вхідною змінною) або багатовимірною (з кількома вхідними змінними).
2. Логістична регресія (Logistic Regression): Цей тип регресії використовується для моделювання бінарних або категоріальних вихідних змінних. Логістична регресія використовує логістичну функцію для прогнозування ймовірності належності до певного класу.
3. Поліноміальна регресія (Polynomial Regression): Це розширення лінійної регресії, де використовуються поліноміальні функції вхідних змінних для моделювання нелінійних залежностей. Поліноміальна регресія дозволяє більш гнучко підлаштовуватися до даних і моделювати більш складні залежності.
4. Регресія з використанням дерев рішень (Decision Tree Regression): Цей тип регресії використовується для моделювання нелінійних та неоднорідних залежностей. Він використовує дерева прийняття рішень, де вхідні дані поділяються на декілька регіонів залежно від їх характеристик та виробляються вихідні значення на основі середніх або медіанних значень в кожному регіоні.
5. Регресія з використанням методу опорних векторів (Support Vector Regression): Цей тип регресії використовується для моделювання як

- лінійних, так і нелінійних залежностей. Він використовує метод опорних векторів, що шукає оптимальну гіперплощину, яка найкраще апроксимує дані.
6. Регресія на основі нейронних мереж (Neural Network Regression): Цей тип регресії використовується для моделювання складних нелінійних залежностей за допомогою штучних нейронних мереж. Він використовує шари нейронів та зв'язки між ними для прогнозування вихідного значення.
 7. Регресія з використанням методу k-найближчих сусідів (k-Nearest Neighbors Regression): Цей метод використовується для моделювання нелінійних залежностей, він шукає k найближчих сусідів для нового прикладу та використовує їх значення для прогнозування вихідного значення.
 8. Регресія з використанням методу градієнтного бустингу (Gradient Boosting Regression): Цей метод використовує ансамбль дерев рішень, що побудовані послідовно, кожне з яких спрямоване на покращення попереднього дерева шляхом зведення помилок.
 9. Регресія з використанням методу випадкового лісу (Random Forest Regression): Цей метод також використовує ансамбль дерев рішень, але кожне дерево будується незалежно за допомогою випадкових підвибірок даних та випадкових підвибірок ознак.
 10. Регресія з використанням методу рекурентних нейронних мереж (Recurrent Neural Network Regression): Цей метод використовується для моделювання послідовних даних, де вхідні дані залежать від попередніх часових кроків. Рекурентні нейронні мережі можуть зберігати стан та інформацію про попередні обробки.
 11. Байєсівська регресія (Bayesian Regression): Цей метод використовує байєсівські статистичні методи для моделювання регресійної залежності. Він дає можливість враховувати апріорні знання та несприятливі впливи шуму в даних.

12.Регресія з використанням методу еластичної сітки (Elastic Net Regression): Цей метод поєднує лінійну регуляризацію та лінійну регресію, щоб моделювати залежності з урахуванням як лінійних, так і нелінійних змінних.

Це лише декілька типів регресій, які можуть бути використані в машинному навчанні. Вибір конкретного методу регресії залежить від характеристик даних, типу залежності та конкретної задачі, яку потрібно вирішити.

2.4 Початковий аналіз даних

EDA (Exploratory Data Analysis) - це процес вивчення та аналізу даних для отримання важливих інсайтів перед використанням їх для моделювання в машинному навчанні. Нижче перераховані основні кроки, які можна виконати під час EDA аналізу даних у машинному навчанні:

1. Завантаження даних: дані завантажуються у відповідну структуру(яка зазвичай має вигляд таблиці) із місця зберігання даних. При завантаженні даних можуть бути прибрані непотрібні колонки даних(поля у базі даних). Також можуть бути оброблені індекси, пропущені значення тощо.
2. Огляд даних: робиться огляд перших кількох рядків даних, щоб отримати загальне уявлення про їх структуру та формат. Також виводиться загальна інформація про дані, така як кількість рядків та стовпців, типи даних кожного стовпця.
3. Візуалізація: дані краще сприймаються людиною(незалежно від того аналітик це чи замовник) у вигляді зображень, аніж великого масиву

чисел. Наприклад, можна побудувати гістограми, діаграми розсіювання, box-plot графіки та інші візуалізації.

4. Обробка даних: виконується очищення та попередню обробку даних. Це може включати заповнення пропущених значень, видалення дублікатів, масштабування даних та інші операції для підготовки даних до моделювання.
5. Аналіз статистичних характеристик: досліджуються основні статистичні характеристики даних, такі як середнє значення, медіана, стандартне відхилення, мінімум та максимум. Це дозволить отримати більш детальне розуміння розподілу даних та виявити потенційні аномалії або викиди.
6. Кореляційний аналіз: досліджуються кореляції між різними стовпцями даних. Використовується матриця кореляцій та графіки теплової карти для візуалізації залежностей. Це допомагає виявити сильні та слабкі взаємозв'язки між змінними.
7. Виявлення аномалій: використовуються статистичні методи, візуалізації та алгоритми виявлення аномалій для виявлення потенційних аномалій або викидів у даних.
8. Вибір важливих ознак: проводиться аналіз важливості ознак, щоб визначити, які змінні найбільше впливають на вихідну змінну. Це може допомогти вибрати найбільш важливі ознаки для подальшого моделювання.

Ці кроки EDA надають глибоке розуміння нашого набору даних, дозволяють виявити потенційні проблеми та допомагають нам вибрати та підготувати правильні ознаки для побудови моделі машинного навчання.

2.4.1 Аналіз даних. Огляд, візуалізація і кореляція

Поглянемо на дані і побудуємо гістограму з розподілом даних. Нагадаю, що ми досліджуватимемо залежність між зваженою пожиттєвою цінністю користувача(wLTV) і середньою ціною за одну продаж(CPP) (рис 2.1).

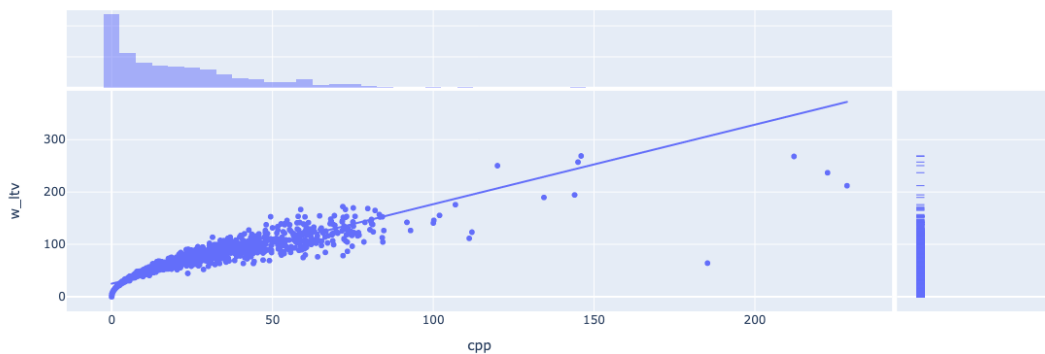


Рисунок 2.1 – Залежність між wLTV і CPP

Можемо бачити, що залежність не лінійна, тому одразу відкидаємо один з можливих варіантів пошуку залежності між змінними, а саме лінійну регресію.

Наступним кроком мав бути пошук кореляції, але це не обов'язково робити, адже здоровий глузд підказує, що чим більше ресурсів витрачено на залучення користувача, то більшою буде його пожиттєва цінність. Тим не менш, перевіримо наше припущення. Формула за якою будемо знаходити кореляцію наведена на рис. 2.2.

$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E(X)^2} \cdot \sqrt{E(Y^2) - E(Y)^2}}$$

Рисунок 2.2 – Формула кореляції

Кореляція між двома змінними wLTV і CPP приблизно дорівнює 0.91, що підтверджує наше припущення. Таким чином дійсно доцільно шукати залежність між цими змінними.

Слід також звернути увагу на розподіл кампаній за змінною CPP. Бачимо що велика кількість значень знаходиться на проміжку $[0; 2.5)$. Отже робимо висновок що більшість рекламних кампаній або оптимізуються алгоритмами соціальної мережі де вони були запущені, або створені задля залучення неякісних користувачів, якщо це передбачено у маркетинговій стратегії. Проте, як зазначено раніше, є сильна кореляція між “ціною” залученого користувача і його зваженою пожиттєвою цінністю.

Потенційні залежності між іншими змінними шукати не має сенсу, адже середня ціна продажі є функцією від кількості покупок і об’ємом ресурсів витрачених на роботу кампанії, отже очевидно, що кореляція там точно знайдеться.

2.5 Поліноміальна регресія

Поліноміальна регресія - це варіант лінійної регресії, в якому використовуються поліноміальні функції вхідних змінних для побудови моделі. У поліноміальній регресії вхідні змінні підносяться до степеня, що вищий за 1, що дозволяє моделі враховувати нелінійні залежності між змінними.

Формула поліноміальної регресії має вигляд:

$$y = \beta_0 + \beta_1x + \beta_2x^2 + \beta_3x^3 + \dots + \beta_nx^n,$$

де:

- y – вихідна змінна. Тобто змінна яку ми намагаємось передбачити,
- x – вхідна змінна,
- $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ – коефіцієнти моделі, які впливають на вихідну змінну.

Поліноміальна регресія дозволяє моделювати нелінійні залежності між вхідними та вихідними змінними шляхом використання степеневих функцій. Наприклад, якщо маємо лише одну вхідну змінну x , поліноміальна регресія другого ступеня буде виглядати наступним чином:

$$y = \beta_0 + \beta_1x + \beta_2x^2$$

Таким чином, поліноміальна регресія може моделювати складніші взаємозв'язки між змінними, ніж проста лінійна регресія. Однак варто враховувати, що вищі степені поліномів можуть призвести до перенавчання моделі, тому важливо збалансувати складність моделі та її здатність узагальнювати на нові дані.

2.6 Перенавчання

Перенавчання (overfitting) є проблемою, яка може виникнути при побудові моделі поліноміальної регресії, коли модель занадто точно підлаштовується під тренувальні дані, але недостатньо узагальнюється на нові, невідомі дані. Це може призвести до поганої прогнозної здатності моделі.

Одна з причин перенавчання в поліноміальній регресії - використання занадто високих степенів полінома. Якщо ми використовуємо поліном

високого ступеня, модель може відтворити шум або випадкові варіації в тренувальних даних, що не мають суттєвого впливу на справжній зв'язок між змінними. Це може призвести до "перекрученості" моделі та неправильних прогнозів на нових даних.

Іншою причиною перенавчання можуть бути недостатні дані для тренування моделі. Якщо тренувальний набір даних дуже малий або не представляє достатньої різноманітності, модель може "запам'ятати" кожен окремий приклад, замість виявлення загальних закономірностей. Це може призвести до перенавчання та відсутності вміння узагальнювати на нові дані.

Щоб запобігти перенавчанню в поліноміальній регресії, можна застосовувати наступні стратегії:

1. Збільшення тренувального набору даних: додавання більше репрезентативних даних може допомогти моделі краще зрозуміти загальну закономірність та уникнути перенавчання.
2. Використання регуляризації: додавання штрафу на великі значення коефіцієнтів в поліноміальній регресії може допомогти контролювати складність моделі та зменшити перенавчання.
3. Використання крос-валідації: використання методу крос-валідації дозволяє оцінити прогнозу здатність моделі на незалежних даних. Це допомагає виявити, чи відбувається перенавчання та налаштувати модель відповідним чином.
4. Використання вибірки моделей: вибір моделей з різними ступенями полінома та вибір кращої моделі на основі метрик якості може допомогти уникнути перенавчання та забезпечити кращу узагальнюючу здатність.

Ці стратегії допоможуть контролювати перенавчання та покращити прогнозу здатність поліноміальної регресії на нових даних.

2.7 Оцінка ефективності роботи моделі

Оцінка ефективності моделі поліноміальної регресії залежить від кількох факторів. Основні метрики, які можуть бути використані для оцінки ефективності, включають наступні:

1. Середня квадратична помилка (Mean Squared Error, MSE): Це одна з найпоширеніших метрик для оцінки точності регресійних моделей. Вона обчислює середнє значення квадратів різниць між прогнозованими значеннями моделі і справжніми значеннями. Чим менше значення MSE, тим краще модель прогнозує дані. MSE обчислюється за формулою $MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$, де Y є вектором реальних даних спостережуваної змінної, а \hat{Y} є вектором передбачених значень.
2. Коефіцієнт детермінації (R-squared): Цей показник відображає, наскільки добре модель пояснює варіацію у вихідних даних. Значення R-squared може бути в діапазоні від 0 до 1, де 0 означає, що модель не пояснює жодної варіації, а 1 означає, що модель повністю пояснює варіацію. Чим ближче значення R-squared до 1, тим краще модель підлаштовується під дані. R-squared знаходиться за формулою $R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2}$, $\bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$, де y – фактичне значення, а \hat{y} – оціночне значення.

Отже оцінюватимемо ефективність роботи моделі за допомогою метрик MSE і R-squared. Обидві метрики широко використовуються для оцінки точності та пояснювальної здатності регресійних моделей, включаючи поліноміальну регресію.

Застосовуючи MSE і R-squared разом, ми отримаємо інформацію про точність та пояснювальну здатність поліноміальної регресії. Однак, слід

пам'ятати, що ці метрики не вичерпно оцінюють всі аспекти ефективності моделі.

2.8 Навчання моделі

Ми вже дослідили, що дані мають залежність, проте не лінійну. Звісно, можна спробувати знайти найбільш точну лінійну залежність між змінними, проте це не буде відображати об'єктивну залежність між змінними. Тому розглянемо модель поліноміальної регресії.

Найбільшою “складністю” у навчанні поліноміальної регресії є знаходження коефіцієнтів моделі, які впливають на вихідну змінну. Тобто знаходження степені поліному, який буде наближуватись до справжньої залежності.

На перший погляд здається, що поліному третього степеню буде достатньо. Перевіримо це навчивши модель і оцінимо ефективність роботи побудованої моделі.

Дані поділено у співвідношенні 8:2. Тобто тренувальних даних 80%, даних для тестування 20%. Навчивши модель поліноміальної регресії на тренувальній вибірці отримали наступні показники ефективності роботи моделі (табл. 2.1). Також візуалізуємо відновлену залежність (рис 2.2).

Таблиця 2.1 – Показники ефективності роботи моделі

Метрика	Значення
MSE	139.65
R_squared	0.91

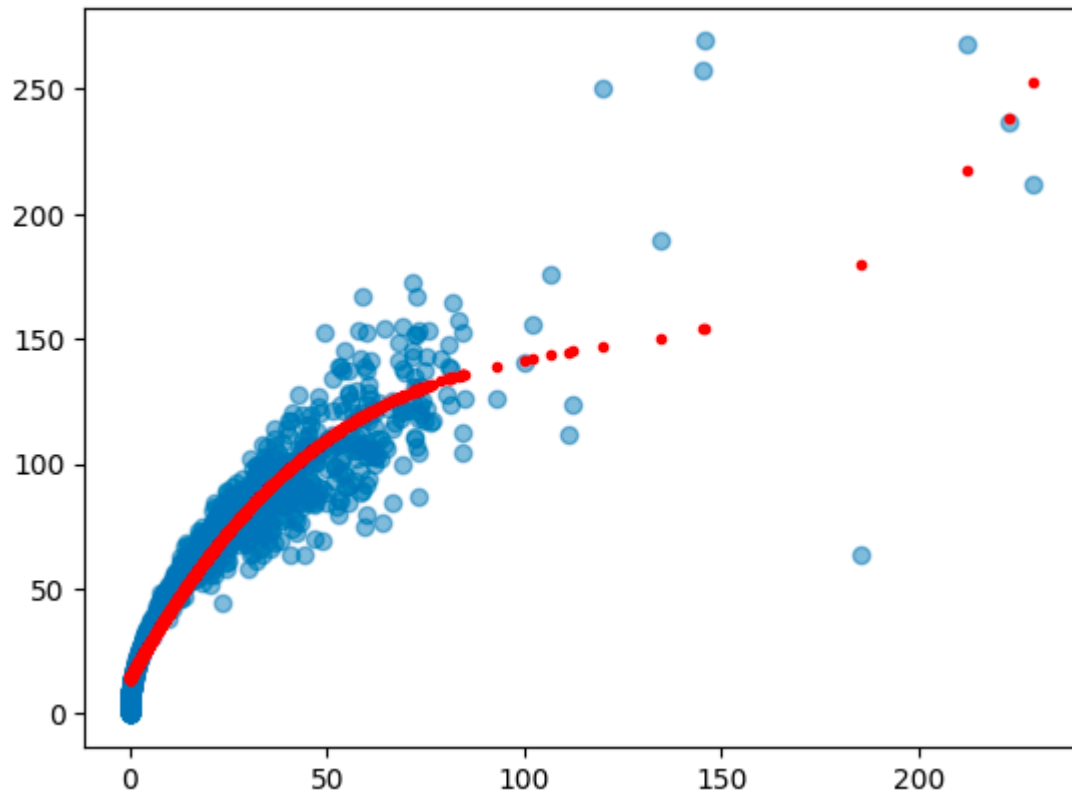


Рисунок 2.2 – Відновлена залежність

Тепер перевіримо ефективність роботи моделі на тестових даних. Тестові дані складають 20% від початкової вибірки і вибрані випадковим чином. Результати роботи моделі наведено у таблиці 2.2, а відновлена залежність візуалізована на рис 2.3

Таблиця 2.2 – Результати роботи моделі

Метрика	Значення
MSE	96.49
R_squared	0.94

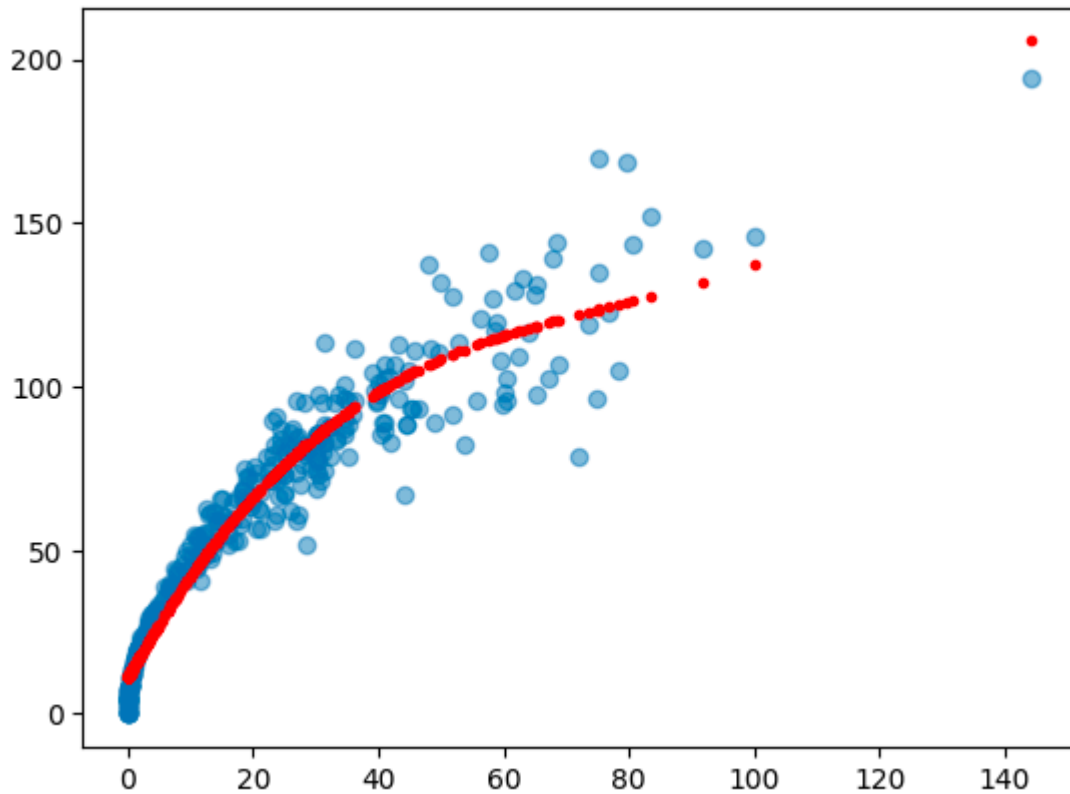


Рисунок 2.3 – Відновлена залежність

Отже маємо досить гарні результати роботи моделі. Судячи з метрики $R_squared$. Проте незрозуміло наскільки “гарною” є метрика MSE . Вона пояснює наскільки точно модель передбачає значення. Тобто чим менша MSE тим краща точність моделі. Проте для кожного набору даних MSE є унікальною. Тому порівнюємо метрики отримані на тренувальних даних при використанні поліноміальної регресії і при використанні лінійної регресії. Отримані значення наведені у таблиці 2.3

Таблиця 2.3 – Значення MSE залежно від степеню поліному

Степінь поліному	MSE
1	277.12
3	139.65

Отже з використанням поліноміальної регресії отримали зменшення MSE у 1.98 рази. Звичайно можна було б отримати навіть менше значення метрики, проте потрібно було б більшу кількість даних і більшу кількість часу аби створити і навчити модель. Тому я вважаю що отриманих результатів роботи моделі достатньо.

2.9 Висновки до другого розділу

У другому розділі я зробив поверхневий аналіз даних. Зокрема аналіз з використанням візуалізацій. Це дозволило звузити набір потенційних моделей машинного навчання. Після поверхневого аналізу я визначився з моделлю машинного навчання. Зупинився на поліноміальній регресії з параметром $degree=3$. Після навчання отримали непогані результати на тестовій вибірці. Також розглянули відмінності у метриках між лінійною та поліноміальними регресіями. Отримали покращення метрики MSE майже у два рази, що свідчить про те, що обрана модель майже у два рази точніша.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ І АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Проблематика задачі

Потрібно створити програмний продукт, який відповідатиме наступним вимогам:

- Швидкодійність.
- Наявність черги.
- Цілодобова робота.
- Наявність моніторингу стану.
- “Віддалена” робота.
- Масштабованість.

Найбільшою складністю у створенні системи автоматизованого керування рекламними кампаніями є створення швидкодійної та масштабованої системи. У соціальних мережах відсутній єдиний алгоритм для створення API. Тобто під кожен систему потрібно створити окремий модуль який буде з нею взаємодіяти. Отже потрібно одразу розуміти з якими соціальними мережами буде працювати система. В нашому випадку ми обмежимося мережею Facebook.

Швидкодійність і масштабованість залежать від архітектури системи, тому над архітектурою слід працювати в першу чергу. Також швидкодійність залежить від потужностей апаратного забезпечення, тому потрібно обирати “з запасом”, щоб уникнути ситуацій з перевантаженням системи і, як наслідок, зниженням швидкодії. На щастя запропонована система не потребує значних потужностей, тому її розробку та тестування можна проводити на локальних потужностях. Це дозволить зберегти значну кількість ресурсів, які можна буде спрямувати на збільшення рекламних бюджетів. Це дозволить збільшувати потенційні доходи компанії.

Усі алгоритми будуть створені використовуючи мову python версії 3.10. Такий вибір обумовлений поширеністю та легкістю сприйняття мови. Також python має велику кількість безкоштовних бібліотек які дозволяють створювати різноманітні рішення. В тому числі автоматизовані системи у маркетингу. Також більшість соціальних мереж мають свої SDK створені спеціально для мови python. Це значно збільшує швидкість створення та тестування програмних продуктів

Потрібно створити систему, яка працюватиме на віддалених потужностях. Найпершим рішенням в голову приходить використання хмарних сервісів, які надає сервіс AWS(Amazon Web Servers). Також компанія Amazon безкоштовно надає рік користування їхніми сервісами, тому зупинимось на них.

Також потрібно вибрати сервіс, який дозволить створити середовище роботи системи. Можна розгорнути систему безпосередньо на віддаленому сервері, проте це завадить масштабованості системи. Наприклад потрібно буде перенести всю систему на інший віддалений сервер. Це складно зробити безпосередньо з сервера, тому також використовуватимемо Docker. Docker – це програмне забезпечення, що дозволяє робити контейнеризацію додатків. Це ПЗ дозволяє упаковувати програмні продукти та їх залежності разом в контейнери, що забезпечує переносимість та незалежність від операційної системи де запускаються контейнери. Також контейнери Docker забезпечують репродукованість середовища. Це означає що можна створити контейнер з точною копією налаштувань та залежностей, що забезпечує однакову поведінку програмного продукту на будь-якій системі.

При створенні системи буде реалізовано збір даних із використанням API. Дані будуть зберігатись у реляційній базі даних. Системою управління реляційними базами даних було обрано MySQL версії 8

3.2 Архітектура системи

Кожен раз система збиратиме актуальні дані з соціальної мережі з використанням АРІ що надається цією соціальною мережею. Після збору актуальні дані будуть замінювати поточні дані. Тобто в рамках одного дня будуть видалятися і додаватися нові записи до бази даних. Після актуалізації даних завантажуватимуться дані за весь період часу і відбуватиметься агрегація. На агрегованих даних виконуватиметься перевірка ефективності роботи рекламної кампанії. Після перевірки буде сформований список рекламних кампаній які потрібно зупинити. Наступним кроком буде зупинка роботи рекламних кампаній у соціальній мережі з використанням АРІ що надається соціальною мережею. Якщо зупинка роботи рекламних кампаній була успішною, про це буде повідомлено, а також буде надіслано список вимкнених рекламних кампаній. Якщо зупинка успішна, то дані про зупинені кампанії також будуть оновлені. Якщо зупинка неуспішна, про це також буде повідомлено текстовим повідомленням. Така логіка роботи може бути представлена у вигляді схеми на рис. 3.1

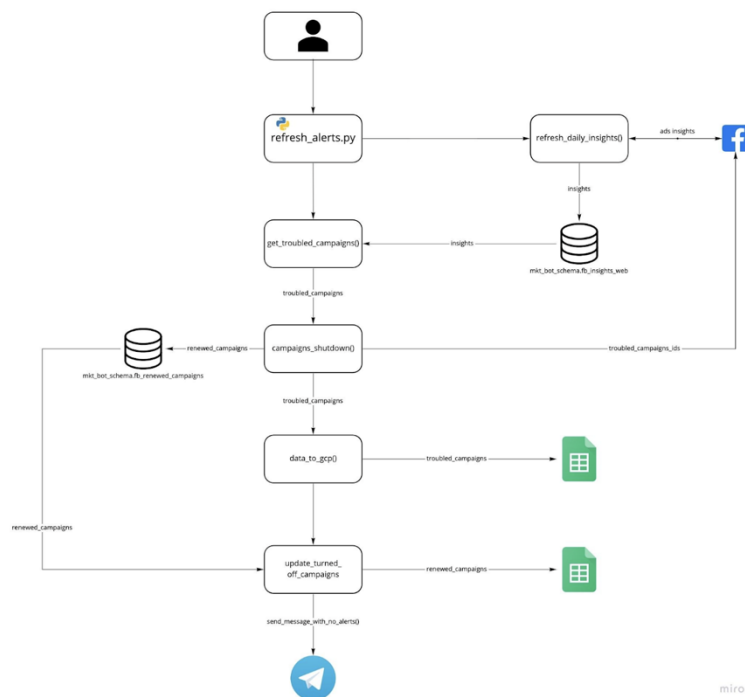


Рисунок 3.1 – Схема роботи системи

3.3 Реалізація прийняття рішень

У попередньому(2) розділі було описано математичне підґрунтя для прийняття рішень. Тепер перейдемо до програмної реалізації. Маємо агреговані дані про роботу рекламної кампанії: ідентифікатор кампанії, ціну за одну продажу(CPP) і середню зважену позитивну цінність користувача(wLTV). Також маємо навчену модель, яку розглянули у розділі 2 і відновлену залежність. Тобто маємо функцію, яка прийматиме на вхід CPP і wLTV, а на вихід даватиме рішення щодо припинення роботи кампанії. Всередині функції буде реалізовано перевірку поточного значення wLTV: якщо воно більше або рівне за $f(CPP_{current})$ то рекламна кампанія може працювати далі. Інакше роботу кампанії слід зупинити. Проте приймаючи рішення таким чином ми ризикуємо допустити помилку 2 роду. Тобто ми вимкнемо рекламну кампанію яку в майбутньому залишили б. Це відбувається через процес оптимізації рекламних кампаній безпосередньо всередині соціальної мережі. Єдиним виходом з цієї ситуації є дати кампанії час “навчитись”. Тобто дати їй витратити достатню кількість ресурсів. Покладаючись на власний досвід вважатимемо, що після витрати 20\$ ми можемо приймати рішення про вимкнення рекламної кампанії. Тобто тепер ми робимо всі описані вище кроки, отримуємо список рекламних кампаній що мають бути вимкнені, а після цього прибираємо зі списку рекламні кампанії, що витратили менш ніж 20\$.

3.4 Реалізація системи сповіщень

Потрібно надсилати сповіщення про роботу системи і, у випадку наявності вимкнених кампаній, надсилати ідентифікатори цих кампаній. Ці сповіщення мають надходити швидко, у зручному вигляді і у місце до якого мають доступ відповідальні люди. Велика кількість комунікації відбувається у месенджері Telegram, тому вирішено надсилати сповіщення саме туди. Також Telegram дозволяє надсилати широкий спектр повідомлень, таких як: текст, зображення, аудіо, відео, посилання тощо. Тобто ми матимемо змогу зручно відформатувати повідомлення щоб запобігти втраті фокусу і втраті важливих даних. Наприклад маркетинг менеджер розуміє, що кампанія мала бути вимкнена, проте її немає в списках вимкнених кампаній. Тоді маркетинг менеджер може оперативно сповістити розробника системи про потенційну помилку в роботі. Найбільшою перевагою Telegram є простота використання API. Це також зменшить час потрібний на розробку модулю сповіщень.

3.5 Робота з даними

Будемо отримувати дані про рекламні кампанії з соціальної мережі Facebook. Розробники Facebook називають ці дані insights. Для отримання даних знадобиться ключ доступу. Іншими словами access key. Задля уникнення неприємностей я прибрав цей ключ, замінивши його на коментар. Так як даних може бути досить багато, слід обмежити кількість завантажуваних даних за один запит. Або створити функцію, що буде завантажувати дані частинами. На щастя Facebook має такий функціонал, тому завантажувати дані будемо саме так. Формат завантажених даних наведено на рисунку 3.2

	account_currency	campaign_name	spend	impressions	campaign_id	date_start	date_stop	country_code	account_id
0	USD	[B]/Vadym/BetterKids/Pixel2/Keiki_Web_3_Benipo...	0.820000	55.0	23852917195960009	2023-03-01	2023-03-01	AU	6.289765e+14
1	USD	[B]/Vadym/BetterKids/Pixel2/Keiki_Web_3_Benipo...	0.500000	84.0	23852917195960009	2023-03-01	2023-03-01	CA	6.289765e+14
2	USD	[B]/Vadym/BetterKids/Pixel2/Keiki_Web_3_Benipo...	0.020000	6.0	23852917195960009	2023-03-01	2023-03-01	IL	6.289765e+14
3	USD	[B]/Vadym/BetterKids/Pixel2/Keiki_Web_3_Benipo...	0.010000	56.0	23852917195960009	2023-03-01	2023-03-01	MV	6.289765e+14
4	USD	[B]/Vadym/BetterKids/Pixel2/Keiki_Web_3_Benipo...	0.160000	49.0	23852917195960009	2023-03-01	2023-03-01	NZ	6.289765e+14
...
79787	USD	[B]/Daryna/GrowthKidsLearning/Pixel3/Keiki_Web...	16.820978	9497.0	23854331875330319	2023-05-25	2023-05-25	MX	1.348667e+15
79788	USD	[B]/Daryna/GrowthKidsLearning/Pixel2/Keiki_Web...	0.130000	161.0	23854381799500319	2023-05-25	2023-05-25	MX	1.348667e+15
79789	USD	[B]/Daryna/GrowthKidsLearning/Pixel2/Keiki_Web...	0.000000	21.0	23854381919630319	2023-05-25	2023-05-25	CR	1.348667e+15
79790	USD	[B]/Daryna/GrowthKidsLearning/Pixel2/Keiki_Web...	0.000000	59.0	23854381919630319	2023-05-25	2023-05-25	DO	1.348667e+15
79791	USD	[B]/Daryna/GrowthKidsLearning/Pixel2/Keiki_Web...	0.000000	21.0	23854381919630319	2023-05-25	2023-05-25	EC	1.348667e+15

Рисунок 3.2 – Формат завантажених даних

Наведені раніше дані будуть агреговані. Під час агрегації дані групуються по ідентифікатору рекламної кампанії і матимуть вигляд наведений на рисунку 3.3. Також на цих даних була навчена модель з розділу 2.

	campaign_name	campaign_id	spend	impressions	purchases	date_start	date_stop	weighted_arps	cpp	w_ltv
0	[B]/Daryna/BetterKids/Pixel2/Keiki_Web_3_Benipo...	23853169920860009	186.840836	10488.0	11	2023-03-05	2023-03-08	43.77	16.99	55.95
1	[B]/Daryna/BetterKids/Pixel2/Keiki_Web_3_Benipo...	23853189014640009	45.330000	2192.0	-1	2023-03-07	2023-03-07	43.85	45.33	93.53
2	[B]/Daryna/BetterKids/Pixel2/Keiki_Web_3_Benipo...	23853189029420009	43.040000	2780.0	1	2023-03-07	2023-03-12	43.46	43.04	91.34
3	[B]/Daryna/BetterKids/Pixel2/Keiki_Web_3_Benipo...	23853189039570009	16.690001	1175.0	1	2023-03-07	2023-03-15	43.56	16.69	60.79
4	[B]/Daryna/BetterKids/Pixel2/Keiki_Web_3_Benipo...	23853189048400009	56.510000	3427.0	2	2023-03-07	2023-03-09	43.78	28.26	79.37
...
1819	[T]/Daria/GrowthKidsLearning/Pixel2/HCPM2/en/0...	23854095107320070	30.380000	2848.0	1	2023-03-14	2023-03-17	52.26	30.38	84.99
1820	[T]/Daria/GrowthKidsLearning/Pixel2/HCPM2/en/0...	23854095107210070	30.350002	2105.0	-1	2023-03-14	2023-03-17	51.84	30.35	91.68
1821	[T]/Daria/GrowthKidsLearning/Pixel2/HCPM2/en/0...	23854106042730070	30.480063	3227.0	1	2023-03-15	2023-03-18	52.11	30.48	80.28
1822	[T]/Daria/GrowthKidsLearning/Pixel2/HCPM2/en/0...	23854106043030070	30.130001	2037.0	-1	2023-03-15	2023-03-18	50.42	30.13	86.22
1823	[T]/Daria/GrowthKidsLearning/Pixel2/HCPM2/en/0...	23854106042980070	29.969999	2071.0	1	2023-03-15	2023-03-18	52.38	29.97	77.50

Рисунок 3.3 – Формат агрегованих даних

3.6 Віддалений сервер

Як було зазначено раніше, будемо користуватись сервісом AWS. Створення віддаленого серверу гарно описане у документації Amazon[7]. Зосередимось на використанні самого серверу. Після створення remote instance ми маємо зберегти ключі для доступу до серверу. Файл з ключем має розширення .pem. Отже спочатку треба запустити сервер. Алгоритм дій наведено на рисунку 3.4

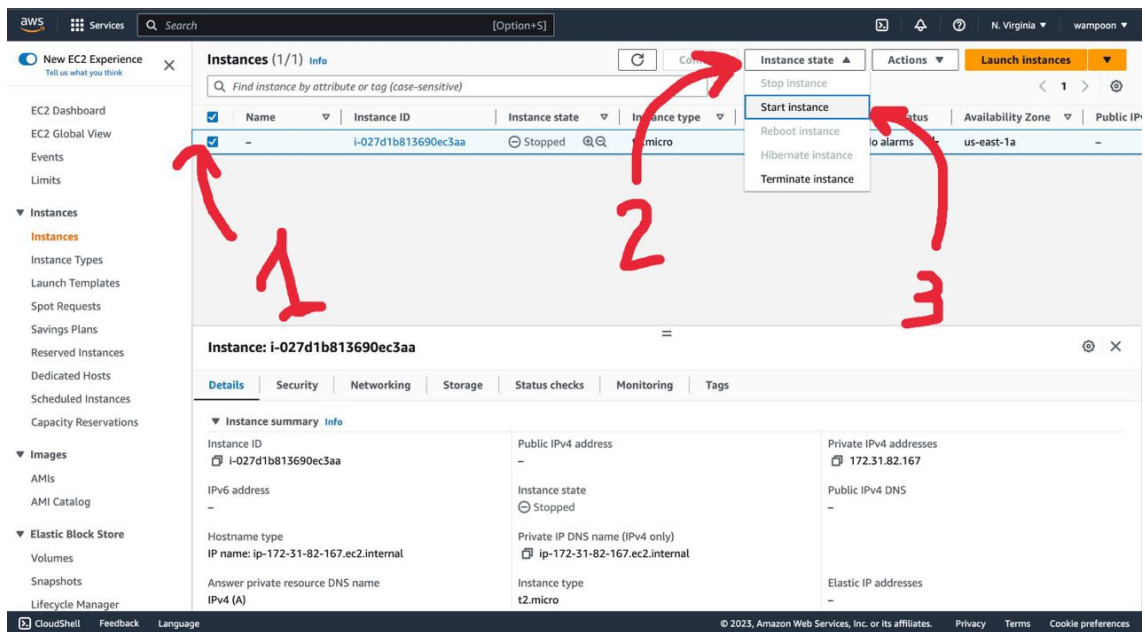


Рисунок 3.4 – Алгоритм дій для запуску сервера

Після цього статус віддаленого серверу має змінитись на “Running”. Рядок з інформацією про віддалений сервер має виглядати так як це показано на рисунку 3.5

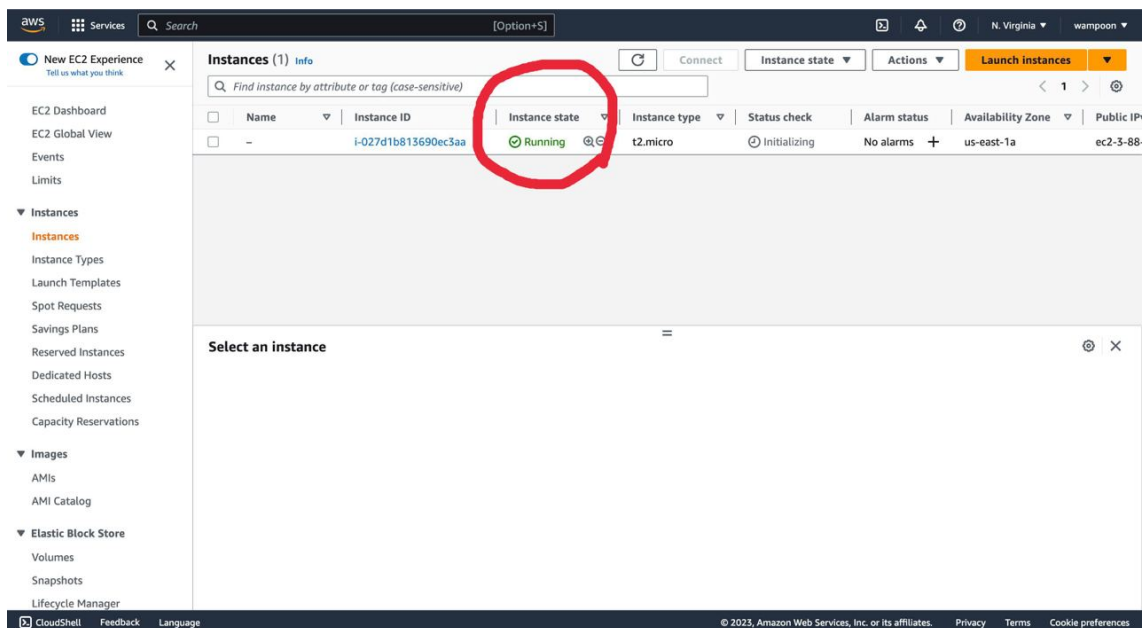


Рисунок 3.5 – Вигляд рядку з інформацією про віддалений сервер

Після цього потрібно перейти у налаштування сервера натиснувши на Instance ID. Перед нами відкриється вікно налаштувань сервера. Тут нас цікавить підключення до серверу(рис. 3.6). Робити це будемо через SSH. SSH

– це мережевий протокол який забезпечує безпечно з'єднання через ненадійну мережу, таку як Інтернет.

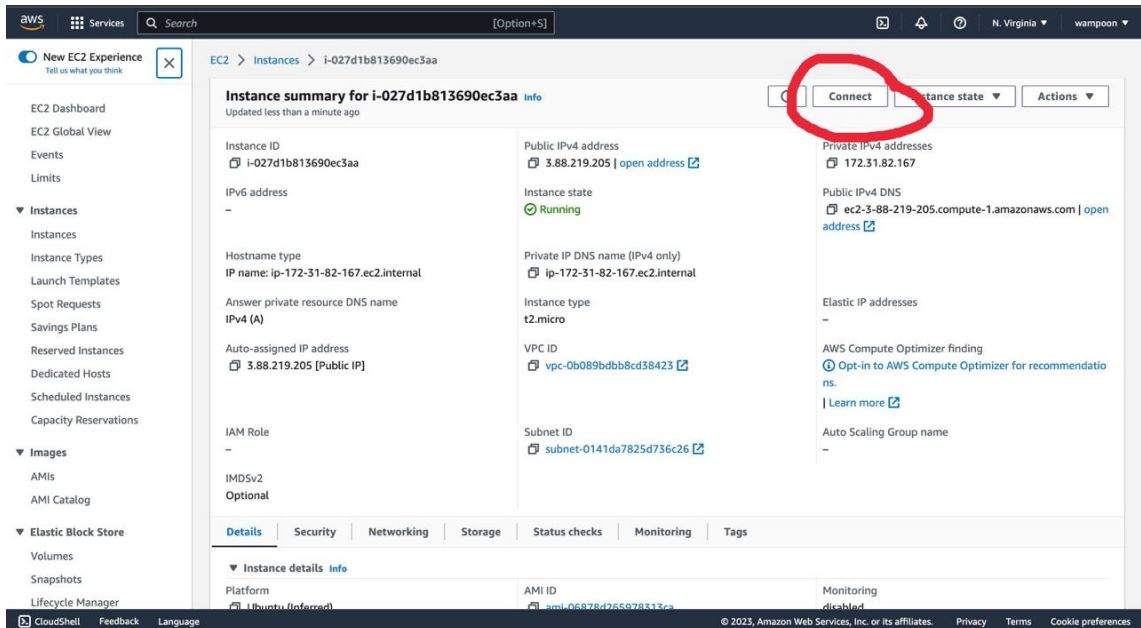


Рисунок 3.6 – Вікно налаштувань сервера

Після цього потрібно підключитись до сервера з локальної машини використовуючи отримані раніше файли ключів. На щастя AWS дає можливість копіювати команду потрібну для виконання підключення (рис 3.7).

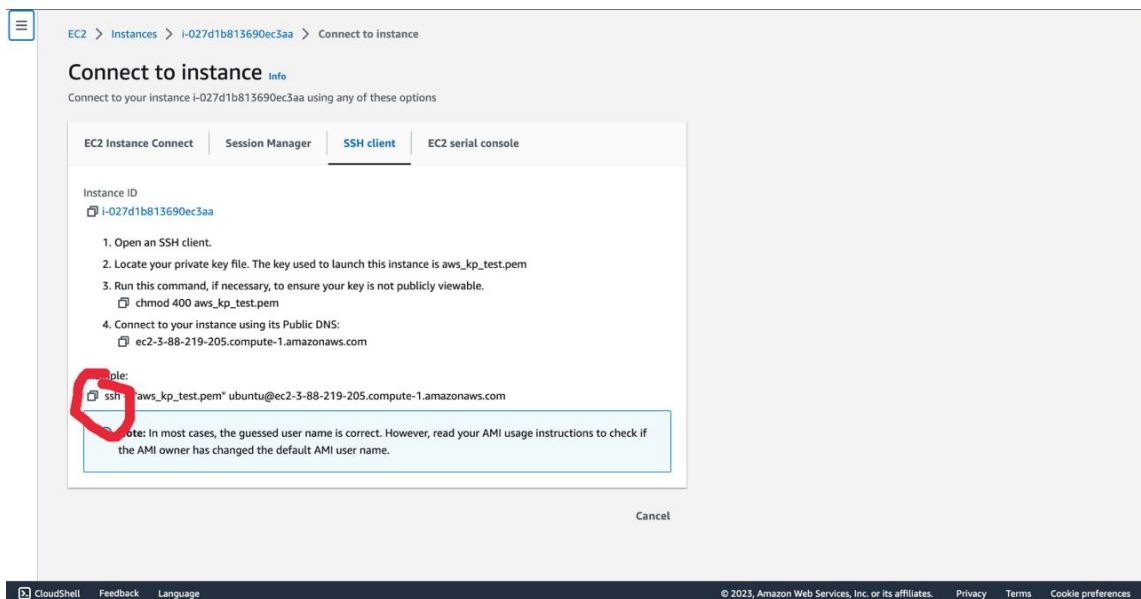


Рисунок 3.7 – Команда для підключення до сервера

Після виконання цих операцій потрібно зайти в термінал на локальній машині(PowerShell для Windows, Terminal для Mac/Linux). Після успішного підключення до серверу у терміналі має зв'явитись інформація подана на рисунку 3.8

```
aws — ubuntu@ip-172-31-82-167: ~ — ssh -i aws_kp_test.pem ubuntu@ec2-3-88-219-205.compute-1.amazonaws.com — 104x37
mkhyz@Maksyms-MacBook-Air aws % ssh -i "aws_kp_test.pem" ubuntu@ec2-3-88-219-205.compute-1.amazonaws.com
The authenticity of host 'ec2-3-88-219-205.compute-1.amazonaws.com (3.88.219.205)' can't be established.
ED25519 key fingerprint is SHA256:XstJ8JsgmVeBzGjSOXiMdDj89qU3DuP565swiuVaiao.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: ec2-44-211-67-50.compute-1.amazonaws.com
  ~/.ssh/known_hosts:4: ec2-54-80-11-159.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-88-219-205.compute-1.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1026-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue Jun  6 20:41:34 UTC 2023

System load:  0.02490234375      Processes:            102
Usage of /:   18.1% of 24.05GB   Users logged in:     0
Memory usage: 67%              IPv4 address for docker0: 172.17.0.1
Swap usage:  0%                IPv4 address for eth0:  172.31.82.167

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Mon Jun  5 17:51:58 2023 from 176.36.11.129
ubuntu@ip-172-31-82-167:~$
```

Рисунок 3.8 – Інформація після успішного підключення

3.7 Програмна оболонка

Після того як створено віддалений сервер і виконано підключення до нього, слід запусити Docker і створити контейнер для запуску системи. Завантаження і запуск Docker відбувається за декілька простих команд:

- sudo apt update && sudo apt upgrade -y;
- sudo apt install ca-certificates curl gnupg lsb-release;
- sudo mkdir -p /etc/apt/keyrings;

- `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg;`
- `echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >/dev/null;`
- `sudo apt update;`
- `sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin;`
- `docker run hello-world.`

Після запуску останньої команди маємо отримати наступне повідомлення(рис. 3.9)

```
ubuntu@ip-172-31-82-167:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

ubuntu@ip-172-31-82-167:~$ █
```

Рисунок 3.9 – Сповідання про успішну інсталяцію Docker

Після успішної установки сервісу Docker маємо створити контейнер для роботи системи. Для цього необхідно створити файл-конфігурацію для Docker (рис. 3.10).

```

ubuntu@ip-172-31-82-167:~/diploma$ cat Dockerfile
FROM python:3.9

ENV PYTHONPATH "${PYTHONPATH}"
COPY ./requirements.txt / .

RUN pip3 install -r requirements.txt

CMD ["python", "bot.py"]

```

Рисунок 3.10 – Вміст Dockerfile

Після створення конфігурації потрібно власне створити контейнер. На щастя робиться це за одну команду `docker build`. В аргументи команди передаємо бажану назву контейнера (рис. 3.11).

```

ubuntu@ip-172-31-82-167:~/diploma$ docker build -t diploma_1 .
[+] Building 146.4s (8/8) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 180B                                             0.0s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/python:3.9                   0.1s
=> [internal] load build context                                                0.5s
=> => transferring context: 22.07MB                                             0.4s
=> CACHED [1/3] FROM docker.io/library/python:3.9@sha256:6e9c45244cf9ef21fe9310f6c7bab754c5e4d23c819421f91 0.0s
=> [2/3] COPY ./requirements.txt / . .                                          0.3s
=> [3/3] RUN pip3 install -r requirements.txt                                    89.7s
=> exporting to image                                                            55.3s
=> => exporting layers                                                            55.1s
=> => writing image sha256:41f5b261d0b5fdaa471a70a888e4d174633c32596fef772a97ec9b2691074e0b 0.0s
=> => naming to docker.io/library/diploma_1                                     0.0s

```

Рисунок 3.11 – Створення контейнера

Залишається зробити останній крок: потрібно зробити так, щоб система робила оновлення регулярно. В реальних умовах цей інтервал має бути 15-30 хвилин, проте в нашому випадку можемо виставити інтервал 8 годин. Зробити це дозволить утиліта `crontab` яка йде в комплекті з встановленою операційною системою на віддаленому сервері. Перший запуск і налаштування відбувається командою `crontab -e`. Далі відкриється вікно редактора `crontab` куди ми маємо ввести частоту виконання команд і, власне, команду (рис. 3.12).

```

GNU nano 6.2                               /tmp/crontab.rNWoif/crontab
Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 */8 * * * sudo docker run -i diploma_1 bash -c "python main.py" >/dev/null 2>&1

```

[^]G Help [^]O Write Out [^]W Where Is [^]K Cut [^]T Execute [^]C Location ^M-^U Undo
[^]X Exit [^]R Read File [^]\ Replace [^]U Paste [^]J Justify [^]/ Go To Line ^M-^E Redo

Рисунок 3.12 – Редактор crontab

3.8 Результати роботи

У попередньому підрозділі ми створили програмну оболонку для автономного функціонування системи. Основною функцією цього програмного продукту є зупинка роботи неефективних рекламних кампаній, а також сповіщення про відключені рекламні кампанії. Раніше було зазначено, що сповіщення приходитимуть у месенджер Telegram (рис. 3.13).

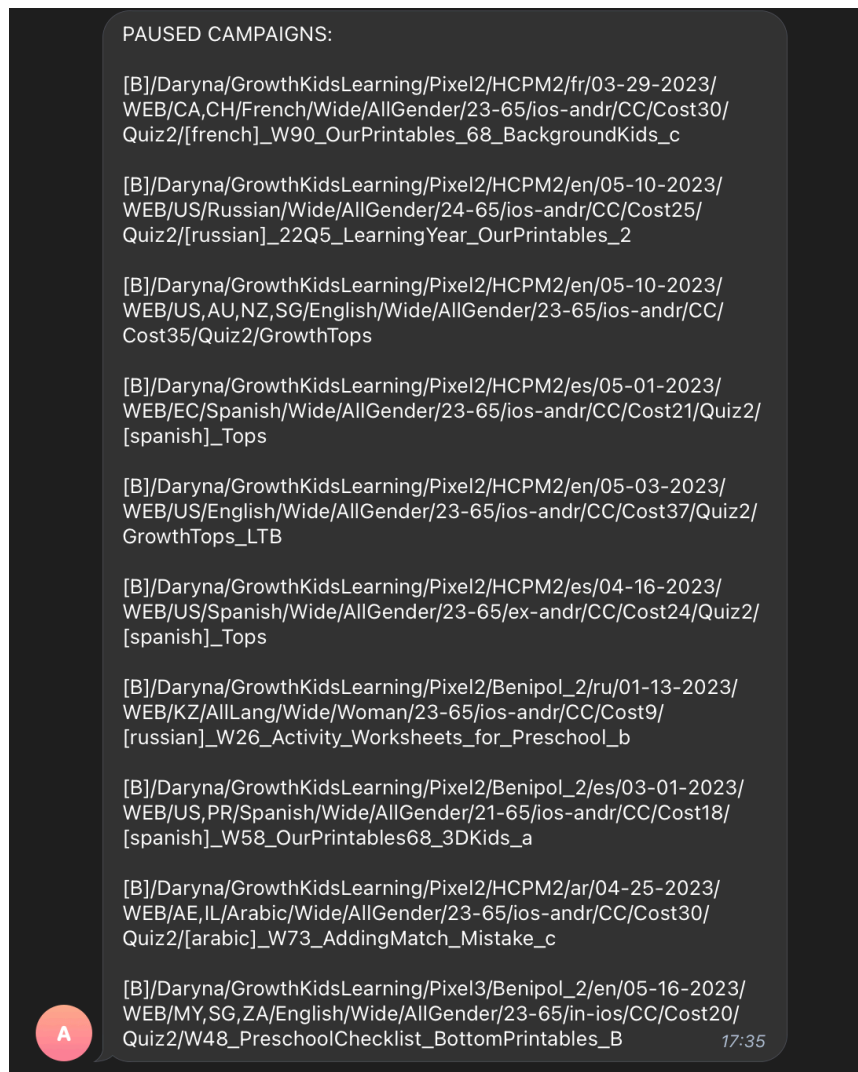


Рисунок 3.13 – Сповіщення про зупинені рекламні кампанії

Також зупинені рекламні кампанії записуються у базу даних для уникнення помилок пов'язаних з повторною зупинкою. Формат запису відключених кампаній наведено на рисунку 3.14

	campaign_name	campaign_id
1	[B]/Daryna/GrowthKidsLearning/Pixel2/BenipoL_2...	23854393875640459
2	[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/ar/...	23854681963260070
3	[B]/Daryna/BetterKids/Pixel2/Keiki_Web_3_Benip...	23853263511200009
4	[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/es/...	23854579002650070
5	[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/ar/...	23854185126810070
6	[B]/Daryna/GrowthKidsLearning/Pixel2/Keiki_Web...	23854272503300009
7	[B]/Daryna/GrowthKidsLearning/Pixel2/BenipoL_2...	23855367006960459
8	[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/ar/...	23854093528690070
9	[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/en/...	23854994960310070

Рисунок 3.14 – Зберігання даних про відключені рекламні кампанії

3.9 Висновки до третього розділу

Було створено програмний продукт а також програмну оболонку для автономного функціонування програмного продукту. Для створення ПП і ПО були використані: python, MySQL, AWS, Docker, crontab, Telegram. Такий стек технологій дозволяє максимально швидко розгортати та тестувати програмний продукт. Також дана конфігурація дозволяє в режимі реального часу відстежувати правильність рішень прийнятих системою.

4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

В заданому розділі буде проведено оцінювання основних характеристик для майбутнього програмного продукту, що створений для підвищення ефективності роботи рекламних кампаній у соціальних мережах

Дана реалізація буде сприяти проведенню усіх необхідних досліджень, що дасть змогу якісно дослідити питання не лише в Україні, проте у всьому світі.

Також в даному дослідженні показано різні варіанти реалізації для забезпечення найбільш коректної та оптимальної стратегії вибору, що має вплив на економічні фактори та сумісність з майбутнім програмним продуктом. Для цього застосовувався апарат функціонально-вартісного аналізу.

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

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

4.1 Постановка задачі проектування

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки системи прогнозу стійкості фінансових показників. Оскільки рішення стосовно проектування та реалізації компонентів, що розробляється, впливають на всю систему, кожна окрема підсистема має її задовольняти. Тому фактичний аналіз представляє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення аналізу даних по компанії.

Технічні вимоги до програмного продукту є наступні:

- функціонування на персональних комп'ютерах із стандартним набором компонентів;
- зручність та зрозумілість для користувача;
- швидкість обробки даних та доступ до інформації в реальному часі;
- можливість зручного масштабування та обслуговування;
- мінімальні витрати на впровадження програмного продукту.

4.2 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка можливого програмного продукту, яка дозволяє аналізувати різні характеристики, що безпосередньо впливають на стійкість підприємства. Беручи за основу цю функцію, можна виділити наступні:

F_1 – вибір мови програмування.

F_2 – вибір провайдера віддаленого сервера.

F_3 – вибір середовища розробки.

Кожна з цих функцій має декілька варіантів реалізації:

Функція F_1 :

- а) python.
- б) C#.

Функція F_2 :

- а) AWS.
- б) MS Azure.

Функція F_3 :

- а) Visual Studio Code.
- б) Google Colab.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1).

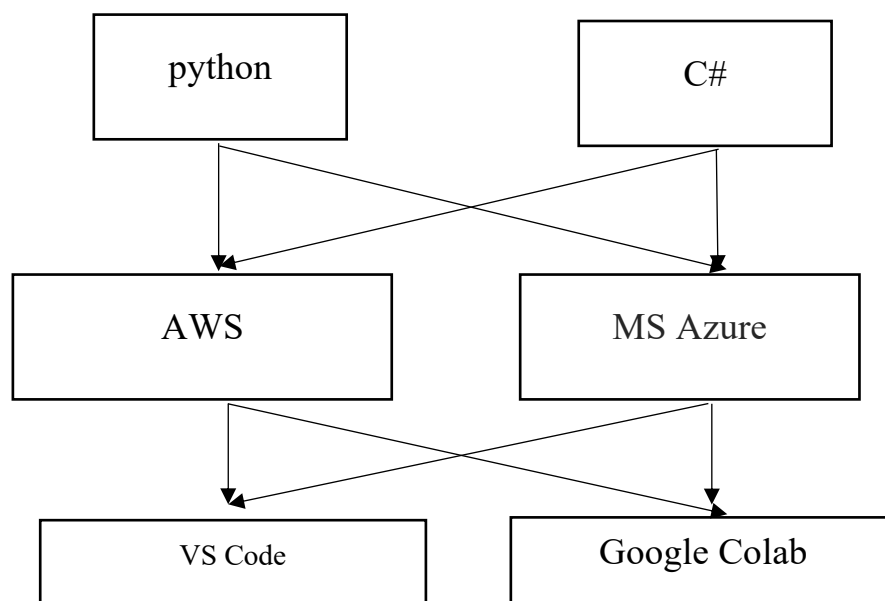


Рисунок 4.1 – Морфологічна карта

Морфологічна карта відображає множину всіх можливих варіантів основних функцій. Позитивно-негативна матриця показана в таблиці 4.1.

Таблиця 4.1 - Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
F_1	А	Зрозумілий інтерфейс, велика кількість інтеграцій, широкий спектр бібліотек	Нижча швидкодія, використання великих об'ємів пам'яті
	Б	ООП-орієнтована мова програмування, легкі інтеграції з продуктами Microsoft	Потрібне специфічне середовище виконання, обмежена кросплатформеність, складність синтаксису
F_2	А	Оплата on-demand, open-source продукт, наявна підтримка Linux-подібних операційних систем	Відсутність прозорого пояснення цін, складний інтерфейс
	Б	Легкість з'єднання внутрішніх сервісів, оплата за хвилину	Відсутність підтримки Linux, «дружелюбність» лише до продуктів Microsoft
F_3	А	Наявність великої кількості інтеграцій, наявність вбудованої системи версіонування, кросплатформеність	Складність налаштування, застарілий інтерфейс
	Б	Робота на віддалених потужностях, інтеграція з продуктами Google	Обмеження на обчислювальні ресурси, обмеження на довжину сесії, залежність від якості Інтернету

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F_1 :

Перевагу даємо зрозумілості і масштабованості. Для спрощення роботи з API варіант Б має бути відкинтий.

Функція F_2 :

Програма допускає обрання обох варіантів. Можливо використати варіанти один з двох варіантів: А чи Б.

Функція F_3 :

Реалізація першого варіанту є сприйнятливою для зменшення витрат часу на реалізацію програми. Це варіант А.

Таким чином, будемо розглядати такі варіанти реалізації ПП:

$$F_1a - F_2a - F_3a$$

$$F_1a - F_2б - F_3a$$

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.3 Обґрунтування системи параметрів програмного продукту

На основі даних, розглянутих вище, визначаються основні параметри вибору, які будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- $X1$ – швидкодія мови програмування;
- $X2$ – об'єм пам'яті для завантаження даних;
- $X3$ – час обробки даних;
- $X4$ – потенційний об'єм програмного коду.

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію програмного продукту, як показано у таблиці 4.2.

Таблиця 4.2 - Основні параметри програмного продукту

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	X1	оп/мс	10000	14000	19000
Об'єм пам'яті	X2	Мб	1024	500	100
Час обробки даних	X3	мс	60000	10000	3000
Потенційний об'єм програмного коду	X4	кількість рядків коду	500	250	100

За даними таблиці 4.3 будуються графічні характеристики параметрів – рис. 4.2 – рис. 4.5.

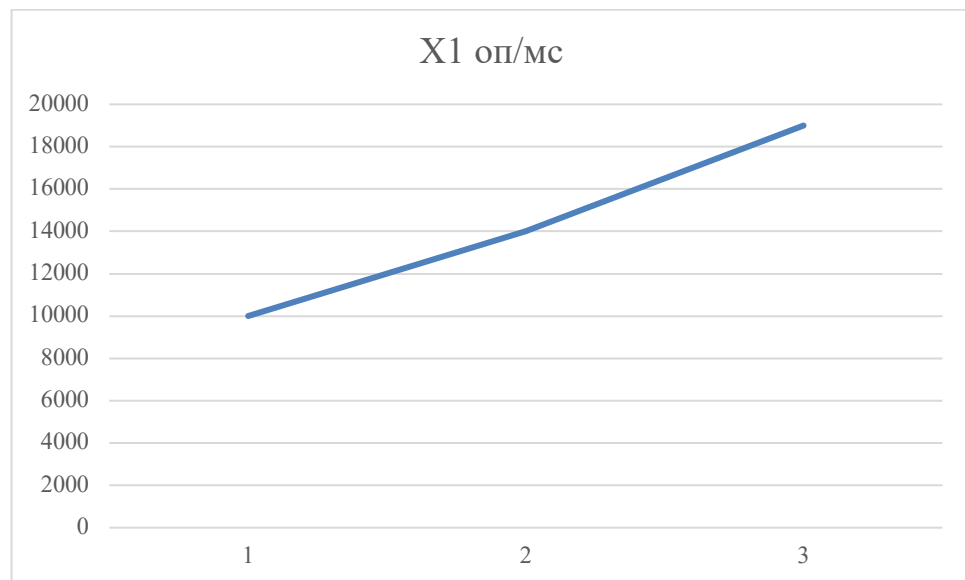


Рисунок 4.2 – X1, швидкодія мови програмування

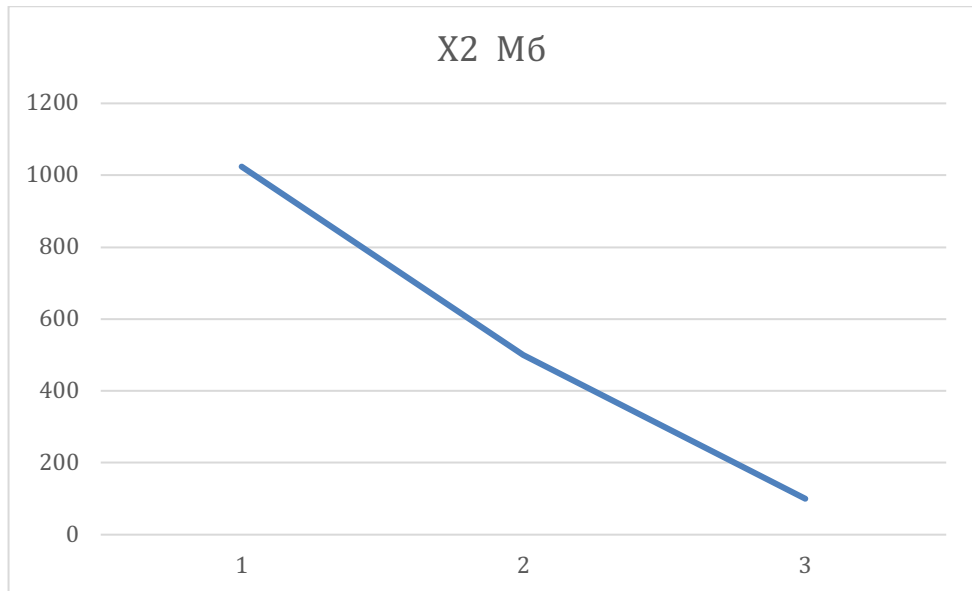


Рисунок 4.3 – X2, об'єм пам'яті

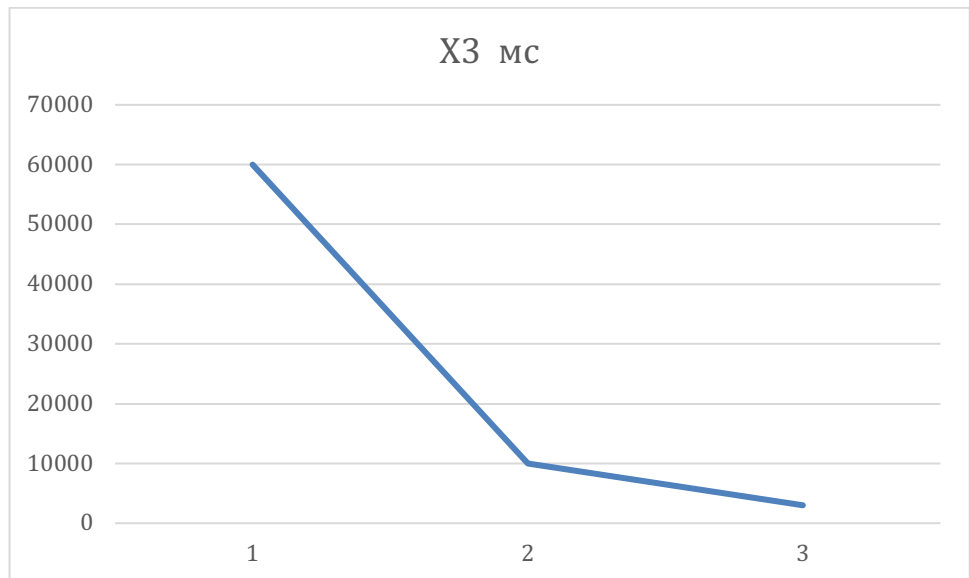


Рисунок 4.4 – X3, час попередньої обробки даних

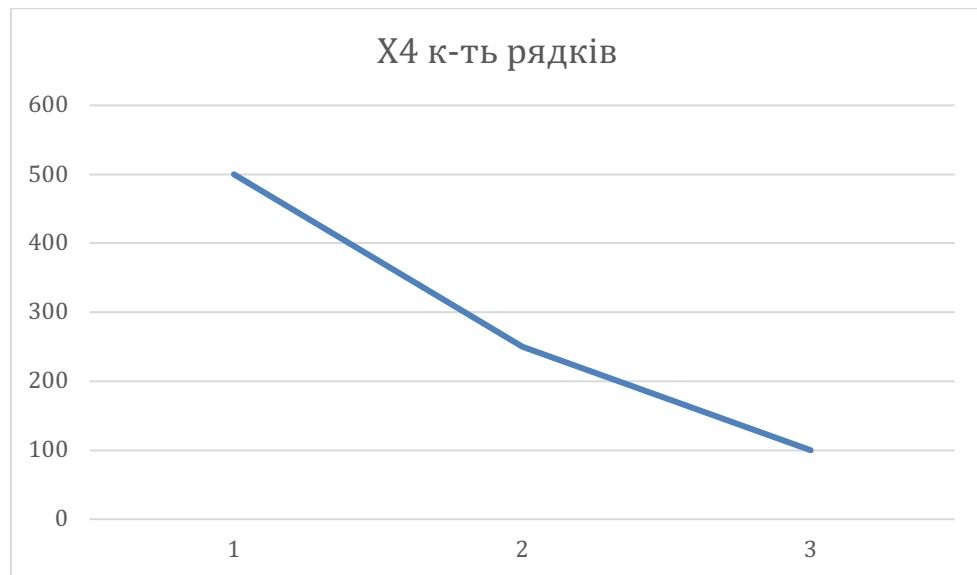


Рисунок 4.5 – X4, потенційний об'єм програмного коду

4.4 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

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

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

Результати експертного ранжування наведені у таблиці 4.3.

Таблиця 4.3 - Результати ранжування параметрів

Назва параметра	Позначення параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
Швидкість мови програмування	$X1$	Оп/мс	2	2	2	1	1	1	2	11	-6,5	42,25
Об'єм пам'яті	$X2$	Мб	3	4	3	3	4	3	4	24	6,5	42,25
Час попередньої обробки даних	$X3$	мс	1	1	1	2	2	2	1	10	-7,5	56,25
Потенційний об'єм програмного коду	$X4$	Кількість рядків коду	4	3	4	4	3	4	3	25	7,5	56,25
Разом			10	10	10	10	10	10	10	70	0	197

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

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

де N – число експертів,

n – кількість параметрів;

б) середня сума рангів:

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

в) відхилення суми рангів кожного параметра від середньої суми рангів:

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

Сума відхилень по всіх параметрам повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^N \Delta_i^2 = 197. \quad (4.4)$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 197}{7^2(4^3 - 4)} = 0,754 > W_k = 0,67. \quad (4.5)$$

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

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.4.

Таблиця 4.4 - Попарне порівняння параметрів.

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	<	<	<	<	<	<	<	<	0,5
X1 і X3	>	>	>	<	<	<	>	>	1,5
X1 і X4	<	<	<	<	<	<	<	<	0,5
X2 і X3	>	>	>	>	>	>	>	>	1,5
X2 і X4	<	>	<	<	>	<	<	<	0,5
X3 і X4	<	<	<	<	<	<	<	<	0,5

Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається по формулі:

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

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

Для кожного параметра зробимо розрахунок вагомості K_{vi} за наступними формулами:

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

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

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

$$K_{vi} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \quad (4.9)$$

$$b'_i = \sum_{j=1}^N a_{ij} b_j \quad (4.10)$$

Як видно з таблиці 4.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 - Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.		Третя ітер.	
	X1	X2	X3	X4	b_i	K_{Bi}	b_i^1	K_{Bi}^1	b_i^2	K_{Bi}^2
X1	1	0.5	1.5	0.5	3.5	0.22	12,25	0,21	41,875	0,2
X2	1.5	1	1.5	0.5	4.5	0.28	16,25	0,28	59,125	0,28
X3	0.5	0.5	1	0.5	2.5	0.16	9.25	0.16	34.125	0.16
X4	1.5	1.5	1.5	1	5.5	0.34	21,25	0,35	77,875	0,36
Всього:					16	1	59	1	213	1

4.5 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів $X2$ (Об'єм пам'яті), $X3$ (час обробки даних) та $X4$ (потенційний об'єм програмного коду) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра $X1$ (швидкодія мови програмування) обрано не найгіршим.

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

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

де n – кількість параметрів;

K_{gi} – коефіцієнт вагомості i -го параметра;

B_i – оцінка i -го параметра в балах.

Таблиця 4.6 - Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	A	X1	15000	9	0.2	1.8
F2	A	X2	100	10	0.28	2.8
	B	X3	4000	6	0.16	0.96
F3	A	X4	150	7	0.36	2.52

За даними з таблиці 4.6 за формулою:

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

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 1.8 + 2.8 + 2.52 = 7.12 ;$$

$$K_{K2} = 1.8 + 0.96 + 2.52 = 5.28 .$$

Як видно з розрахунків, кращим є 1 варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.6 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту.
2. Розробка програмної оболонки.

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

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Загальна трудомісткість обчислюється як:

$$T_0 = T_P \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}, \quad (4.13)$$

де T_P – трудомісткість розробки ПП;

K_{Π} – поправочний коефіцієнт.

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

K_M – коефіцієнт рівня мови програмування.

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

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

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: $T_P = 32$ людино-дні. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_{\Pi} =$

1.5. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1: $K_{СК} = 1$. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ} = 0.7$. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 32 \cdot 1.5 \cdot 0.7 = 33.6 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто $T_P = 25$ людино-днів, $K_{П} = 0.95$, $K_{СК} = 1$, $K_{СТ} = 0.6$:

$$T_2 = 25 \cdot 0.95 \cdot 0.6 = 14.25 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (33.6 + 14.25 + 4.8 + 14.25) \cdot 8 = 535.2 \text{ людино-годин.}$$

$$T_{II} = (33.6 + 14.25 + 6.91 + 14.25) \cdot 8 = 552.08 \text{ людино-годин.}$$

Найбільш високу трудомісткість має варіант II.

В розробці беруть участь один програміст з окладом 18000 грн., один аналітик в області даних з окладом 25000 і один тестувальник з окладом 12000. Визначимо середню зарплату за годину за формулою:

$$C_{ч} = \frac{M}{T_m \cdot t} \text{ грн.,} \quad (4.14)$$

де M – місячний оклад працівників;

T_m – кількість робочих днів тиждень;

t – кількість робочих годин в день.

$$C_{\text{ч}} = \frac{18000 + 25000 + 12000}{3 \cdot 21 \cdot 6} = 145,5 \text{ грн.} \quad (4.15)$$

Тоді, розрахуємо заробітну плату за формулою:

$$C_{\text{зп}} = C_{\text{ч}} \cdot T_i \cdot K_{\text{д}}, \quad (4.16)$$

де $C_{\text{ч}}$ – величина погодинної оплати праці програміста;

T_i – трудомісткість відповідного завдання;

$K_{\text{д}}$ – норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

$$\text{I. } C_{\text{зп}} = 145,5 \cdot 535,2 \cdot 1,2 = 93445,92 \text{ грн.}$$

$$\text{II. } C_{\text{зп}} = 145,5 \cdot 552,08 \cdot 1,2 = 96393,168 \text{ грн.}$$

Відрахування на єдиний соціальний внесок становить 22%:

$$\text{I. } C_{\text{вд}} = C_{\text{зп}} \cdot 0,22 = 93445,92 \cdot 0,22 = 20558,1 \text{ грн.}$$

$$\text{II. } C_{\text{вд}} = C_{\text{зп}} \cdot 0,22 = 96393,168 \cdot 0,22 = 21206,5 \text{ грн.}$$

Тепер визначимо витрати на оплату однієї машино-години. ($C_{\text{м}}$)

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

$$C_{\text{г}} = 12 \cdot M \cdot K_3 = 12 \cdot 18000 \cdot 0,2 = 43200 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{3П} = C_{Г} \cdot (1 + K_3) = 43200 \cdot (1 + 0.2) = 51840 \text{ грн.}$$

Відрахування на соціальний внесок:

$$C_{Від} = C_{3П} \cdot 0.22 = 51840 \cdot 0.22 = 11404,8 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 25000 грн.

$$C_A = K_{ТМ} \cdot K_A \cdot Ц_{ПР} = 1.4 \cdot 0.25 \cdot 25000 = 8750 \text{ грн.,}$$

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

K_A – річна норма амортизації;

$Ц_{ПР}$ – договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = K_{ТМ} \cdot Ц_{ПР} \cdot K_P = 1.4 \cdot 25000 \cdot 0.08 = 2800 \text{ грн.,}$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$\begin{aligned} T_{ЕФ} &= (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = (365 - 104 - 12 - 16) \cdot 8 \cdot 0.35 = \\ &= 1383,2 \text{ години,} \end{aligned}$$

де D_K – календарна кількість днів у році;

D_B, D_C – відповідно кількість вихідних та святкових днів;

D_P – кількість днів планових ремонтів устаткування;

t – кількість робочих годин в день;

K_B – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_C \cdot K_3 \cdot C_{\text{ЕН}} = 1383.2 \cdot 0.3 \cdot 0.7 \cdot 4.87 = 1414.6 \text{ грн.},$$

де N_C – середньо-споживча потужність приладу;

K_3 – коефіцієнтом зайнятості приладу;

$C_{\text{ЕН}}$ – тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$C_H = C_{\text{ПР}} \cdot 0.67 = 25000 \cdot 0.67 = 16750 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_A + C_P + C_{\text{ЕЛ}} + C_H, \quad (4.17)$$

$$C_{\text{ЕКС}} = 51840 + 11404.8 + 8750 + 2800 + 1414.6 + 16750 = 92959,40 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{М-Г}} = C_{\text{ЕКС}} / T_{\text{ЕФ}} = 92959,4 / 1383,2 = 67.2 \text{ грн/год.}$$

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

$$C_M = C_{M-\Gamma} \cdot T, \quad (4.18)$$

$$\text{I. } C_M = 67.2 \cdot 535.2 = 35965.44 \text{ грн.}$$

$$\text{II. } C_M = 67.2 \cdot 552.08 = 37099,77 \text{ грн.}$$

Накладні витрати складають 67% від заробітної плати:

$$C_H = C_{ЗП} \cdot 0,67, \quad (4.19)$$

$$\text{I. } C_H = 93445.92 \cdot 0,67 = 62608.8 \text{ грн.}$$

$$\text{II. } C_H = 96393,168 \cdot 0,67 = 64583,4 \text{ грн.}$$

Отже, вартість розробки ПП за варіантами становить:

$$C_{ПП} = C_{ЗП} + C_{ВІД} + C_M + C_H, \quad (4.20)$$

$$\text{I. } C_{ПП} = 93445.92 + 20558.1 + 35965.44 + 62608.8 = 212578,26 \text{ грн.}$$

$$\text{II. } C_{ПП} = 96393,168 + 21206.5 + 37099,77 + 64583,4 = 219282,84 \text{ грн.}$$

4.7 Вибір кращого варіанту ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{TEPj} = K_{Kj} / C_{\Phi j}, \quad (4.21)$$

$$K_{TEP1} = 7.12 / 212578,26 = 3.349354e-05,$$

$$K_{TEP2} = 5.28 / 219282,84 = 2.407849e-05.$$

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $K_{\text{TEP1}} = 3.349354e-05$.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишилися після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $K_{\text{TEP}} = 3.349354e-05$.

Цей варіант реалізації програмного продукту має такі параметри:

- Вибір мови програмування – python;
- Провайдер віддаленого серверу – AWS;
- Використання VS Code як середовища розробки.

Даний варіант виконання програмного комплексу дає користувачу зручний інтерфейс, швидку реалізацію програми та доступний функціонал для роботи.

4.8 Висновки до четвертого розділу

В даній частині було проведено повний функціонально-вартісний аналіз програмного продукту. Також було знайдено оцінку основних функцій програмного продукту.

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

На основі аналізу вибрано перший варіант реалізації програмного продукту.

ВИСНОВКИ

У результаті дослідження та розробки системи автоматизованого керування рекламними кампаніями у соціальних мережах було проведено комплексний аналіз даних, виявлено залежності між ціною за продажу і цінністю залученого користувача (CPP і wLTV), а також визначено ефективну модель машинного навчання – поліноміальну регресію з параметром $\text{degree}=3$. Застосування такої моделі привело до покращення метрики MSE майже у два рази порівняно з лінійною регресією.

Для розробки програмного продукту та його автономного функціонування було використано широкий стек технологій, зокрема Python, MySQL, AWS, Docker, crontab та Telegram. Це дозволило ефективно розгортати та тестувати програмний продукт, а також в режимі реального часу відстежувати правильність рішень, прийнятих системою.

Загальною метою дипломної роботи було створення програмного продукту та визначення основних функцій, що відповідають вимогам рекламного бізнесу. Проведений у роботі аналіз дозволяє зрозуміти параметри, характеризуючі програмний продукт, та вибрати найоптимальніший варіант його реалізації.

В результаті дипломної роботи було розроблено програмний продукт та програмну оболонку, що забезпечують автономне функціонування системи автоматизованого керування рекламними кампаніями у соціальних мережах. Цей продукт дозволяє зберігати стабільність роботи відділу маркетингу, прогнозувати доходи та забезпечує зручність управління рекламними кампаніями.

В цілому розробка системи автоматизованого керування рекламними кампаніями у соціальних мережах дозволила покращити ефективність маркетингових стратегій, збільшити точність прогнозування доходів та підвищити рівень автоматизації управління рекламними кампаніями.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Mobile attribution in iOS 14.5+. Mighty 2021. URL: <https://www.mighty.digital/blog/mobile-attribution-in-ios-14-5>
2. Apple iOS 14.5 Update: Impact on Digital Marketing & Ecommerce Brands. Roirevolution 2021. URL: <https://roirevolution.com/blog/apple-ios-14-update-impact-on-ecommerce-brands/>
3. All you need to know about the iOS14 update for Facebook Ads. Insense 2023. URL: <https://insense.pro/blog/all-you-need-to-know-about-the-ios-14-update-for-facebook-ads#>
4. 3 Case Studies That Show The Power Of Marketing Automation. Webpresence 2016. URL: <https://webpresence.digital/uk-blog/case-studies-power-marketing-automation/>
5. Facebook Demographic Statistics: How Many People Use Facebook in 2023? Backlinko 2023. URL: <https://backlinko.com/facebook-users>
6. 10 essential performance marketing metrics to monitor. Funnel.io 2022. URL: <https://funnel.io/blog/performance-marketing-metrics>
7. Facebook Ads Manager. URL: <https://adsmanager.facebook.com/adsmanager/manage/>
8. Aggarwal C. C. Social Network Data Analytics. New York: Springer, 2016. 568 p.
9. Mitchell T. M. Machine Learning. New York: McGraw-Hill Education, 1997. 432 p.
10. Shalev-Shwartz S., Ben-David S. Understanding Machine Learning: From Theory to Algorithms. Cambridge: Cambridge University Press, 2014. 449 p.
11. Gelman A., Hill J. Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge: Cambridge University Press, 2006. 648 p.

12. Cook R. D., Weisberg S. Applied Regression Including Computing and Graphics. New Jersey: Wiley , 1999. 632 p.
13. Facebook Marketing API. Insights API. URL: <https://developers.facebook.com/docs/marketing-api/insights>
14. What is Amazon EC2? Amazon 2023. URL: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>
15. David A. F. Statistical Models: Theory and Practice. Cambridge: Cambridge University Press, 2009. 458 p.
16. Oracle Cloud Infrastructure Documentation. URL: <https://docs.oracle.com/en-us/iaas/mysql-database/doc/getting-started.html>
17. Aiogram Documentation. URL: <https://docs.aiogram.dev/en/latest/>
18. Bradlow E. T., Fader P. S., Hardie B. G. Handbook of Marketing Analytics: Methods and Applications in Marketing Management, Public Policy, and Litigation Support. Cheltenham: Edward Elgar Publishing, 2010. 712 p.
19. Köksal M. Marketing Analytics: Data-Driven Techniques with Microsoft Excel. New Jersey: Wiley, 2017. 354 p.
20. Docker Documentation. Docker Inc. 2023. URL: <https://docs.docker.com/>
21. List of Probability and Statistics Symbols. Math Vault 2020. URL: <https://web.archive.org/web/20200818113919/https://mathvault.ca/hub/higher-math/math-symbols/probability-statistics-symbols/>
22. Lewis R., Dart J. The New Rules of Marketing and PR: How to Use Content Marketing, Podcasting, Social Media, AI, Live Video, and Newsjacking to Reach Buyers Directly. New Jersey: Wiley, 2018. 448 p.
23. Murphy K. P. Machine Learning: A Probabilistic Perspective. Cambridge: MIT Press 2012. 1104 p.
24. What is Performance Marketing: How it Works, Channels, and Benefits. Simplilearn 2023. URL: <https://www.simplilearn.com/what-is-performance-marketing-article>

25. Understanding Polynomial Regression Model. Analytics Vidhya 2021. URL:
<https://www.analyticsvidhya.com/blog/2021/10/understanding-polynomial-regression-model/>

ДОДАТОК А ІЛЮСТРАТИВНІ МАТЕРІАЛИ

Розробка системи автоматизованого керування рекламними кампаніями в соціальних мережах

Виконав: Хижняк Максим КА-93
Керівник: Гуськова Віра Геннадіївна

Дослідження

Мета: встановити залежність між маркетинговими метриками що характеризують рекламні кампанії, на основі цього розробити систему яка оптимізує доходи компанії приймаючи рішення щодо подальшої роботи рекламної кампанії

Предмет: розробка і дослідження системи автоматизованого керування рекламними кампаніями в соціальних мережах

Об'єкт: впровадження програмного рішення яке дозволить компаніям ефективно управляти рекламними кампаніями в соціальних мережах

 **Актуальність дослідження**

- сучасна задача
 - зростання конкуренції
 - зростання об'ємів даних
 - клієнти очікують персоналізації
 - ефективність витрат
 - відстеження результативності
-

3

 **Постановка задачі**

1. Збір даних
 2. Попередній аналіз, порівняння моделей регресії
 3. Створення програмного продукту і програмної оболонки
-

4

📀 Вхідні дані

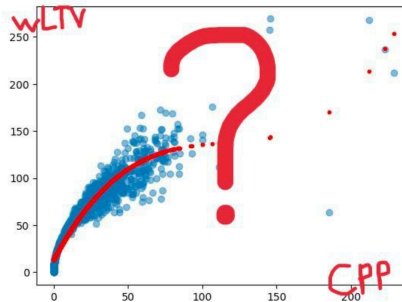
На вхід подавались дані наступного формату:

	account_currency	campaign_name	spend	impressions	campaign_id	date_start	date_stop	country_code	account_id
0	USD	[B]Vadym(BetterKids)Pixel2/Keiki_Web_3_Benipo...	0.820000	55.0	23852917195960009	2023-03-01	2023-03-01	AU	6.289705e+14
1	USD	[B]Vadym(BetterKids)Pixel2/Keiki_Web_3_Benipo...	0.500000	84.0	23852917195960009	2023-03-01	2023-03-01	CA	6.289705e+14
2	USD	[B]Vadym(BetterKids)Pixel2/Keiki_Web_3_Benipo...	0.020000	6.0	23852917195960009	2023-03-01	2023-03-01	IL	6.289705e+14
3	USD	[B]Vadym(BetterKids)Pixel2/Keiki_Web_3_Benipo...	0.010000	56.0	23852917195960009	2023-03-01	2023-03-01	MV	6.289705e+14
4	USD	[B]Vadym(BetterKids)Pixel2/Keiki_Web_3_Benipo...	0.160000	49.0	23852917195960009	2023-03-01	2023-03-01	NZ	6.289705e+14
...
79787	USD	[B]Daryna(GrowthKidsLearning)Pixel2/Keiki_Web...	16.820978	9497.0	23854381919630319	2023-05-25	2023-05-25	MX	1.348867e+15
79788	USD	[B]Daryna(GrowthKidsLearning)Pixel2/Keiki_Web...	0.190000	161.0	23854381919630319	2023-05-25	2023-05-25	MX	1.348867e+15
79789	USD	[B]Daryna(GrowthKidsLearning)Pixel2/Keiki_Web...	0.200000	21.0	23854381919630319	2023-05-25	2023-05-25	CR	1.348867e+15
79790	USD	[B]Daryna(GrowthKidsLearning)Pixel2/Keiki_Web...	0.200000	59.0	23854381919630319	2023-05-25	2023-05-25	DO	1.348867e+15
79791	USD	[B]Daryna(GrowthKidsLearning)Pixel2/Keiki_Web...	0.200000	21.0	23854381919630319	2023-05-25	2023-05-25	EC	1.348867e+15

Дані взяті з реального продукту що створений для навчання дітей дошкільного і шкільного віку

5

🤔 Що шукатимемо?



$$CPP_{total} = \frac{spend_{total}}{N(purchases)}$$

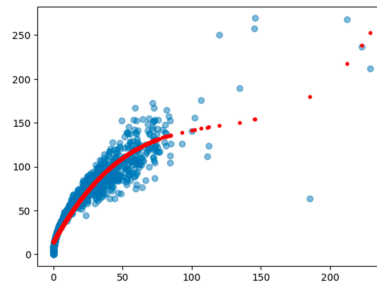
$$wLTV = \sum_{i=0}^n LTV_i * share_i \quad | \quad n = N(campaigns);$$

$$share_i = \frac{spend_i}{Total Spend}, i = 0..n, n = N(campaigns);$$

6

🦄 Модель

Після завантаження і попереднього аналізу спробував знайти залежність:

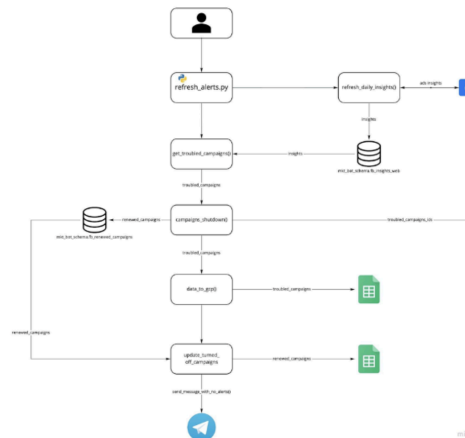


Метрика	Значення
MSE	96.49
R_squared	0.94

Степінь поліному	MSE
1	277.12
3	96.49

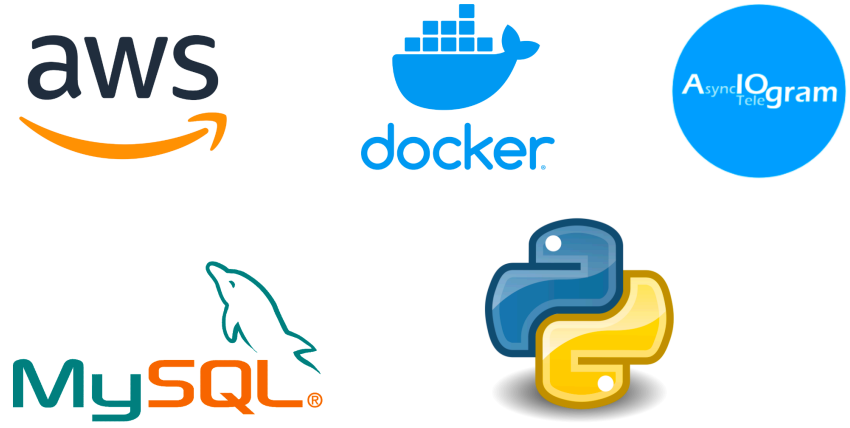
7

🏛️ Архітектура системи



8

📱 Стек технологій



✅ Результат роботи ПП

```

PAUSED CAMPAIGNS:
[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/fr/03-29-2023/
WEB/CA_CN/French/WideAllGender/23-65/ios-andr/CC/Cost30/
Quiz2/french_W90_OurPrintables_98_BackgroundKId_c

[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/en/05-10-2023/
WEB/US/Russian/WideAllGender/24-65/ios-andr/CC/Cost25/
Quiz2/russian_Z2Q9_LearningYear_OurPrintables_2

[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/en/05-10-2023/
WEB/US/AU/NZ/25/English/WideAllGender/23-65/ios-andr/CC/
Cost35/Quiz2/GrowthTops

[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/es/05-01-2023/
WEB/ES/Spanish/WideAllGender/23-65/ios-andr/CC/Cost2/Quiz2/
[Spanish]_Tops

[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/en/05-03-2023/
WEB/US/English/WideAllGender/23-65/ios-andr/CC/Cost17/Quiz2/
GrowthTops_LTB

[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/es/04-16-2023/
WEB/US/Spanish/WideAllGender/23-65/ios-andr/CC/Cost24/Quiz2/
[Spanish]_Tops

[B]/Daryna/GrowthKidsLearning/Pixel2/Benipol_2/ru/01-19-2023/
WEB/KZ/Akang/Wide/Woman/23-65/ios-andr/CC/Cost9/
[russian]_W26_Activity_Worksheets_for_Preschool_b

[B]/Daryna/GrowthKidsLearning/Pixel2/Benipol_2/es/03-01-2023/
WEB/US_PR/Spanish/WideAllGender/21-65/ios-andr/CC/Cost18/
[Spanish]_W18_OurPrintables18_39KId_c

[B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/ar/04-25-2023/
WEB/AE_IL/Arabic/WideAllGender/23-65/ios-andr/CC/Cost30/
Quiz2/arabic_W73_AddingMatch_Mistake_c

[B]/Daryna/GrowthKidsLearning/Pixel2/Benipol_2/en/05-16-2023/
WEB/MY_SG_ZA/English/WideAllGender/23-65/ios-andr/CC/Cost20/
Quiz2/W18_PreschoolChecklist_BottomPrintables_b
  
```

campaign_name	campaign_id
1 [B]/Daryna/GrowthKidsLearning/Pixel2/Benipol_2...	23854393875649459
2 [B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/ar/...	23854681963260070
3 [B]/Daryna/BetterKids/Pixel2/Keiki_Web_3_Benip...	23853263511280009
4 [B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/es/...	23854579002650070
5 [B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/ar/...	23854185126810070
6 [B]/Daryna/GrowthKidsLearning/Pixel2/Keiki_Web...	23854272503300009
7 [B]/Daryna/GrowthKidsLearning/Pixel2/Benipol_2...	23855367806969459
8 [B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/ar/...	238540913528690070
9 [B]/Daryna/GrowthKidsLearning/Pixel2/HCPM2/en/...	238549949460310070

Подальші плани

- Оцінка роботи ПП з точки зору доходів компанії.
- Моніторинги.
- Оптимізація роботи системи.
- Дослідження нових моделей.
- Додавання нових змінних.
- Розробка графічного інтерфейсу.

11

Висновки

Розробка системи автоматизованого керування рекламними кампаніями у соціальних мережах включала:

- Комплексний аналіз даних та виявлення залежностей між ціною за продажу і цінністю залученого користувача.
- Використання ефективної моделі машинного навчання - поліноміальної регресії з параметром $\text{degree}=3$.
- Розробку програмного продукту та його автономну функціональну оболонку з використанням широкого стеку технологій.
- Функціонально-вартісний аналіз програмного продукту та визначення основних функцій.
- Забезпечення автономного функціонування системи автоматизованого керування рекламними кампаніями.
- Підвищення ефективності маркетингових стратегій, точнішого прогнозування доходів та автоматизації управління рекламними кампаніями.

Таким чином, розробка системи автоматизованого керування рекламними кампаніями в соціальних мережах має на меті покращити ефективність маркетингу, забезпечити точніше прогнозування доходів та підвищити рівень автоматизації управління рекламою.

12

Дякую за увагу

ДОДАТОК Б ЛІСТИНГ ПРОГРАМИ

bot.py

```

# core aiogram tools
import aiogram.utils.markdown as md
from aiogram import Bot, Dispatcher, types
from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.types import ParseMode, InlineKeyboardMarkup, InlineKeyboardButton
from aiogram.utils import executor

# async tools
import asyncio

# custom data
from config import *

# init bot
bot = Bot(token=BOT_TOKEN)
storage = MemoryStorage()
dp = Dispatcher(bot, storage=storage)

# "/"start" command handling
@dp.message_handler(commands=['start'])
async def cmd_start(message: types.Message):

    # getting user's Telegram ID
    user_id = message.chat.id

    # reply to message
    await message.reply(f"your telegram id is {user_id}. Welcome")

# make bot start receiving commands and messages
executor.start_polling(dp, skip_updates=True)

main.py
# imports for etl
import pandas as pd

```

```

import numpy as np
from datetime import datetime, timedelta
import pytz
import traceback

# imports for telegram bot
import aiogram.utils.markdown as md
from aiogram import Bot, Dispatcher, types
from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.types import ParseMode
from aiogram.utils import executor
import asyncio

# import custom
from config import *
from utils import *
from tools import *

# logs wrapping
try:

    # update data
    refresh_data('dev')

    # get underperforming campaigns
    bad_campaigns = get_troubled_campaigns()

    # get list of paused campaigns
    paused_campaigns = campaigns_shutdown(bad_campaigns)

    # send alert if some campaigns were turned off
    if (len(paused_campaigns)>0):

        # construct message
        msg = 'PAUSED CAMPAIGNS:\n\n' + '\n\n'.join(paused_campaigns)

        # send message
        asyncio.run(
            send_message(
                ALERTING_CHAT_ID,
                msg
            )
        )

```

```

# send success message
else:
    asyncio.run(
        send_message(
            ALERTING_CHAT_ID,
            'Success'
        )
    )

except Exception as e:

    error_msg = f'\n*ERROR:* \n{str(e)} \n{traceback.format_exc()}'

    asyncio.run(
        send_message(
            ALERTING_CHAT_ID,
            error_msg
        )
    )

```

tools.py

```

import pandas as pd
import numpy as np
import pytz
from config import *
from utils import *

# importing facebook sdk
from facebook_business.api import FacebookAdsApi
from facebook_business.adobjects.adaccount import AdAccount
from facebook_business.adobjects.campaign import Campaign
from facebook_business.adobjects.adset import AdSet

# telegram dependencies
import aiogram.utils.markdown as md
from aiogram import Bot, Dispatcher, types
from aiogram.contrib.fsm_storage.memory import MemoryStorage
from aiogram.types import ParseMode
from aiogram.utils import executor
import asyncio

```

```

def get_min_ltv(cpp):

    min_ltv = 13.451391397904423 + 2.9844908719945047*cpp -
0.02472134947354339*cpp**2 + 0.00007111933272524996*cpp**3

    return min_ltv

def get_max_date_insights(table='insights') -> str:

    max_date = '2023-03-01'
    try:
        temp_date = pd.read_sql(f'select max(date_start) as date from {table}',
con=engine()).iloc[0].date
        max_date = temp_date if pd.notnull(temp_date)==True else max_date

    except Exception as e:
        print(e)
    return max_date

def get_insights():
    return pd.DataFrame()

def refresh_data(env='dev'):

    # try refreshing data
    try:

        # try getting max date from existing data
        max_date_from_db = get_max_date_insights('fb_insights_web')

        # logging date parameters for retrieving insights
        print(f"GETTING WEB FB INSIGHTS:\ndaily from: {max_date_from_db}\ndaily
to:
{(datetime.now(tz=pytz.timezone('Asia/Vladivostok'))).strftime('%Y-%m-%d')}")

        # set parameters for retrieving insights
        insights_params = {
            'level': 'campaign',

```

```

        'time_range':{'since':max_date_from_db,
'until':(datetime.now(tz=pytz.timezone('Asia/Vladivostok'))).strftime('%Y-%m-%d')},
        # 'time_increment':1,
        'breakdowns':['country'],
        'fields':[
            'account_name',
            'account_currency',
            'campaign_name',
            'spend',
            'impressions',
            'campaign_id'
        ]
    }

# if production environment
if env=='prod':

    # get insights from specified accounts with specified parameters
    insights = get_insights(accounts=accounts['web'],
parameters=insights_params)

# if development environment
else:

    # get loaded insights
    insights = pd.read_sql('insights', engine())

# logging amount of retrieved data
print(f'\n\nROWS OF DATA RETRIEVED: {insights.shape[0]}')

# get minimal date from retrieved insights to delete all existing data
with this date and replace with relevant data
date_to_delete = insights.date_start.min()

# deleting last not full day of insights
db_con = engine()
db_con.execute(f"delete          from          insights          where
date_start>='{date_to_delete}'")

# inserting new relevant data to db
insights.to_sql('insights',          engine(),          chunksize=10000,
if_exists='append', index=False)

```

```

        # return good result
        return 1

except Exception as e:

    # log exception
    print(e)

    # return fail
    return 0

def get_troubled_campaigns() -> pd.DataFrame:

    # read data
    insights = pd.read_sql('insights', engine())

    # calculate minimal LTV for every campaign
    insights['min_ltv'] = insights['w_ltv'].apply(lambda x: get_min_ltv(x))

    # get campaigns where weighted LTV is lower than min LTV and where spend
    is at least 20
    bad_campaigns = insights[
        (insights['w_ltv'] < insights['min_ltv'])
        & (insights['spend'] >= 20)
    ]

    # return suspicious campaigns
    return bad_campaigns

def campaigns_shutdown(bad_campaigns: pd.DataFrame) -> list:

    # get paused campaigns
    paused_campaigns = pd.read_sql('paused_campaigns', engine())

    # remove Test campaigns and already paused campaigns
    insights = bad_campaigns[
        ~(bad_campaigns['campaign_name'].str.contains('[T]', regex=False))

```

```

&(~bad_campaigns['campaign_name'].isin(paused_campaigns['campaign_name'].unique().tolist()))
    ].sample(10)

    # set new status for campaigns
    insights['status'] = 'PAUSED'

    # get ids of campaigns
    shutdown_ids = insights['campaign_id'].unique().tolist()

    # get names of paused campaigns
    shutdown_names = insights['campaign_name'].unique().tolist()

    # iterate through accounts and pause campaigns
    for i in range(len(accounts)):

        # init FB API
        FacebookAdsApi.init(app_id=accounts[i]['id'],
access_token=tokens[accounts[i]['business']])

        # iterate through campaigns and pause
        for id in shutdown_ids:

            # init campaign instance
            campaign = Campaign(id)

            # update campaign status
            campaign.api_update(
                params={'status':'PAUSED'}
            )

            # update paused campaigns
            insights[['campaign_name','campaign_id']].to_sql('paused_campaigns',
engine(), if_exists='append', index=False)

        # return paused campaigns' names
        return shutdown_names

async def send_message(chat_id, message) -> None:

    #init bot

```

```

bot = Bot(BOT_TOKEN)

# send message
await bot.send_message(
    chat_id,
    message,
    parse_mode=ParseMode.HTML
)

# close connector
await bot.close()

```

utils.py

```

from config import *

from datetime import datetime, timedelta
from tqdm import tqdm
from operator import add

def engine(schema='diploma', dialect = 'mysql', con = db_con_str, charset =
charset):

    from sqlalchemy import create_engine as engine

    return engine('/'.join([con, schema])+charset)

```

sandbox_diploma.ipynb

```

# %%
import pandas as pd
from datetime import datetime, timedelta
import plotly.express as px
import matplotlib.pyplot as plt
import numpy as np

from config import *
from utils import *

pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)

%load_ext autoreload
%autoreload 2

```

```

import warnings
warnings.filterwarnings("ignore")

# %% [markdown]
# ### rules for w_ltv = f(X)
# X = X<br>
# y = 15*X**0.5 + 0.3 * np.random.randn(X.shape[0],1) * X

# %% [markdown]
# # Getting Data

# %%
insights = pd.read_sql('insights', engine())

ltv = pd.read_sql('ltv', engine())

purchases = pd.read_sql('purchases', engine())

# %%
insights

# %% [markdown]
# # Working with Data

# %%
# # get purchases
#
#             insights['purchases']
insights.reset_index(drop=True).merge(purchases.reset_index(drop=True),
how='left',      left_on=['date_start', 'campaign_name',      'country_code'],
right_on=['install_date', 'campaign_name', 'country_code'])[['purchases']]
# insights.purchases.fillna(0, inplace=True)
# insights['purchases'] = insights.purchases.astype(int)

# # group data
#
#             insights
insights.groupby(by=['campaign_name', 'campaign_id', 'country_code'], as_index=False).agg({'spend': 'sum', 'impressions': 'sum', 'purchases': 'sum', 'date_start': 'min', 'date_stop': 'max'})

# # get total spend for camapaign
#
#             insights['total_spend']
insights.groupby(by=['campaign_name', 'campaign_id']).spend.transform(sum)

```

```

# # get spend share for every geo
# insights['spend_share'] = insights['spend']/insights['total_spend']

# # merge LTV
# insights = insights.merge(ltv, how='left',on=['country_code'])

# # get wLTV
#
#                               insights['weighted_arps']
round(insights['spend_share']*insights['arps_expected'], 2)

# # group data
# insights = insights.groupby(by=['campaign_name','campaign_id'],
as_index=False).agg({'spend':'sum','impressions':'sum','purchases':'sum','dat
e_start':'min','date_stop':'max','weighted_arps':'sum'})

# # prevent division by zero
# insights['purchases'] = insights['purchases'].replace(to_replace=0, value=-
1)

# # get cpp for campaign
# insights['cpp'] = round(abs(insights['spend']/insights['purchases']), 2)

# # purchase classes depending on purchase count
# # insights['purchase_class'] = insights['purchases'].apply(lambda x: '0' if
x<1 else '1' if x==1 else '2-4' if (x>=2 and x<=4) else '5-10' if (x>=5 and
x<=10) else '11+')

# %%
# visualise data to provide EDA
px.scatter(insights, x='cpp',y='w_ltv',marginal_x="histogram",
marginal_y="rug", trendline='ols')

# %%
# convert data to np arrays
X = insights['cpp'].to_numpy().reshape((-1,1))
y = insights['w_ltv'].to_numpy().reshape((-1,1))

# %%
# get correlation between CPP and wLTV
np.corrcoef(X.reshape(-1),y.reshape(-1))[0][1]

# %%

```

```

from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split

# %%
# splitting data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# check train and test sets sizes
print(f"Train size: {X_train.shape[0]}\nTest size: {X_test.shape[0]}")

# %%
# Instance
poly2 = PolynomialFeatures(degree=3, include_bias=False)
X_poly = poly2.fit_transform(X_train)

# Fit Linear model with poly features on train data
poly_model = LinearRegression().fit(X_poly,y_train)
prediction_train = poly_model.predict(X_poly)

# Plot
plt.scatter(X_train,y_train, alpha=0.5)
plt.plot(X_train, prediction_train, color='red', linestyle='', marker='.',
lw=0.1);

# %%
# calculate evaluation metrics
label_mean = np.mean(y_train)
mse = mean_squared_error(y_train, prediction_train)
r_2 = r2_score(y_train, prediction_train)

# print evaluation
print(f"Train data:\nlabel mean: {label_mean}\nMSE: {mse}\nR2: {r_2}")

# %%
# get coeffs
print(f"y = {poly_model.intercept_[0]} + ({poly_model.coef_[0][0]})*X +
({poly_model.coef_[0][1]})*X^2 + ({poly_model.coef_[0][2]})*X^3")

# print(f"b_0: {poly_model.intercept_[0]}\ncoeffs: {poly_model.coef_[0]}")

# %%

```

```
7.111933272524995 / 10000

# %%
13.5+2.98*50 - 0.0247*(50**2) + 0.00007112 * (50**3)

# %%
poly_model.predict(poly2.fit_transform([[50.0]]))

# %%
# Instance
X_poly_test = poly2.fit_transform(X_test)

# Fit Linear model with poly features on train data
poly_model = LinearRegression().fit(X_poly_test,y_test)
prediction_test = poly_model.predict(X_poly_test)

# Plot
plt.scatter(X_test,y_test, alpha=0.5)
plt.plot(X_test, prediction_test, color='red', linestyle='', marker='.',
lw=0.1);

# %%
# calculate evaluation metrics
label_mean = np.mean(y_test)
mse = mean_squared_error(y_test, prediction_test)
r_2 = r2_score(y_test, prediction_test)

# print evaluation
print(f"Test data:\nlabel mean: {label_mean}\nMSE: {mse}\nR2: {r_2}")

# %%
```