Київський національний університет імені Тараса Шевченка
Економічний факультет
Кафедра економічної кібернетики

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

«Моделювання та аналіз даних користувачів веб-сайту на прикладі компанії «Outpost Club»»

студента 4 курсу
спеціальності 051 «Економіка»
ОПП «Економічна кібернетика»
денної форми навчання
Руденка Андрія Андрійовича

Науковий керівник:
кандидат економічних наук, доцент
Шпирко Віктор Васильович

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

Роботу допущено до захисту перед ЕК
рішенням кафедри економічної кібернетики
від 05.06.2024 р., протокол № 15

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

КИЇВ – 2024

РЕФЕРАТ

Кваліфікаційна робота бакалавра містить: 59 ст., 15 рис., 13 табл., 27 джерел.

Ключові слова: KMeans, XGBoost, SQL, Python, бази даних, дата аналітика, візуалізація даних, моделі прогнозування.

Об'єкт дослідження: поведінка користувачів веб-сайту компанії «Outpost Club».

Мета дослідження: розробка та застосування комплексного підходу до моделювання та аналізу даних користувачів веб-сайту компанії «Outpost Club» для оптимізації їх взаємодії з сайтом, підвищення конверсії та, як наслідок, збільшення ефективності бізнесу за допомогою інструментів дата аналітики.

Методи дослідження: індукції та дедукції, історичні, структурні, системного аналізу, кількісного та якісного порівняння, логічного та економічного аналізу, прогнозування, порівняння, економетрики.

Наукова новизна, теоретична значимість дослідження: розроблено комплекс економіко-математичних моделей для аналізу даних користувачів веб-сайту компанії «Outpost Club».

Практична цінність: обґрунтування науково-практичних підходів, спрямованих на вдосконалення процесів оптимізації та мінімізації використання маркетингових ресурсів та оптимізації процесів продажів компанії «Outpost Club»

RESUME

Taras Shevchenko National University of Kyiv,

Faculty of Economics, Department of Economic Cybernetics

Key words: KMeans, XGBoost, SQL, Python, data bases, data analysis, data visualizations, forcasting models.

The graduation research of student describes the current state and opportunities of using data analytics tools in the analysis of user behaviour data.

The work is interesting for researchers and students who study the prospects of applying data analytics methods in B2C segment, as well as for entrepreneurs and marketers who are interested in improving of marketing and sales activities of the enterprise through data analysis and new technologies.

Pages 59, pictures 15, tables 13, bibliog. 27.

ЗМІСТ

ВСТУП

Актуальність даної теми полягає в зростаючій потребі бізнесу в оптимізації своєї онлайн-присутності та підвищенні ефективності взаємодії з користувачами. У сучасному цифровому світі веб-сайти відіграють вирішальну роль у комунікації між компаніями та їх потенційними клієнтами. Збір, аналіз та моделювання даних користувачів дозволяють глибше зрозуміти потреби та поведінку відвідувачів, оптимізувати структуру та контент сайту, підвищити конверсію та, як наслідок, збільшити прибуток.

Дата аналітика є ключовим елементом в успішній роботі компанії, оскільки дозволяє отримати важливі дані та інсайти щодо поведінки споживачів та ефективності роботи різних департаментів так каналів трафіку.

У цьому контексті використовуються різні методи та алгоритми, серед яких особливо виділяються KMeans та XGBoost.

Моделі KMeans широко використовуються для кластеризації даних. Одним із відомих науковців, які активно досліджували та застосовували метод KMeans, є Макс Веллінг, який зробив значний внесок у розвиток методів кластеризації та машинного навчання. Також такі зарубіжні науковці, як Філіп Гетсман [24], Каялвілі Табіанан [25], Шубашіні Велу [25], Вінаякумар Раві [25] Досліджували сегментацію клієнтів у електронній комерції.

Алгоритм XGBoost, який є одним з найпотужніших методів для задач машинного навчання, був розроблений Тенці Чженом [23] та Карлосом Гестрінгом [23]. Цей метод став популярним завдяки своїй високій продуктивності та здатності працювати з великими наборами даних. XGBoost дозволяє ефективно прогнозувати майбутні значення на основі великих масивів даних. Також Альфонсо Монако [26] та Нітіка Шарма [27] Використовували алгоритм XGBoost для прогнозування продажів.

Потрібно зазначити, що дана курсова робота створена на основі реальних завдань, які були виконані працюючи у компанії «Outpost Club».

Метою роботи є розробка та застосування комплексного підходу до моделювання та аналізу даних користувачів веб-сайту компанії «Outpost Club» для оптимізації їх взаємодії з сайтом, підвищення конверсії та, як наслідок, збільшення ефективності бізнесу за допомогою інструментів дата аналітики.
Для досягнення цієї мети у роботі поставлено та вирішено такі завдання:

1. Аналіз поведінки користувачів.
2. Кластеризація користувачів.
3. Дослідження факторів впилву на прибуток.
4. Прогнозування прибутку на наступний рік.

Об'єктом дослідження є поведінка користувачів веб-сайту компанії «Outpost Club».

Предметом дослідження є методи та алгоритми моделювання і аналізу даних, для розуміння поведінки користувачів та оптимізації взаємодії з ними.

Методи дослідження. В основу дипломної роботи покладені загальнонаукові та спеціальні методи дослідження, а саме індукції та дедукції, історичні, структурні, системного аналізу, кількісного та якісного порівняння, теорії ймовірностей та математичної статистики, логічного та економічного аналізу, методи вибіркових досліджень, прогнозування.

Практичне значення одержаних результатів дослідження полягає в обґрунтуванні науково-практичних підходів, спрямованих на вдосконалення процесів оптимізації та мінімізації використання маркетингових ресурсів та оптимізації процесів продажів компанії «Outpost Club» для підвищення ефективності бізнесу та прибутковості.

Структура роботи складається з трьох основних розділів. Перший розділ присвячений дослідженню бази даних компанії та розвідувальному аналізу користувачів.

В другому розділі виконується кластеризація користувачів за допомогою моделі K-means.

У третьому розділі досліджуються фактори, що впливають на прибуток та прогнозуються продажі на наступний рік за допомогою моделі XGBoost.

РОЗДІЛ 1. РОЗВІДУВАЛЬНИЙ АНАЛІЗ ДАНИХ КОРИСТУВАЧІВ

1.1 Ознайомлення з середовищем Hubspot

HubSpot є американською компанією-розробником програмного забезпечення, яка спеціалізується на інструментах інбаунд-маркетингу, продажів і обслуговування клієнтів.

Основна ідея інбаунд-маркетингу, яку просуває HubSpot, полягає в тому, щоб приваблювати клієнтів за допомогою корисного контенту та взаємодій, які є значущими та персоналізованими, замість традиційних методів реклами, таких як холодні дзвінки, спам або прямий маркетинг. Це підхід, що фокусується на створенні та розподілі вартісного контенту, який приваблює потенційних клієнтів, перетворює їх у ліди, замовників та, врешті-решт, прихильників бренду.

HubSpot пропонує широкий спектр продуктів і послуг, включаючи:

- CRM (система управління відносинами з клієнтами): безкоштовний інструмент, що дозволяє компаніям керувати своїми базами даних клієнтів, взаємодіями, угодами та завданнями.
- Маркетинг Hub: набір інструментів для приваблення відвідувачів на сайт, перетворення їх у ліди та клієнтів. Включає інструменти для роботи з електронною поштою, соціальними медіа, SEO, створенням контенту та автоматизацією маркетингу.
- Sales Hub: інструменти для автоматизації продажів та кращого управління воронкою продажів, зокрема, шаблони електронних листів, відстеження взаємодій із потенційними клієнтами та автоматизоване ведення документації.
- Service Hub: інструменти для підвищення ефективності обслуговування клієнтів, включаючи тікети, базу знань, інструменти для проведення опитувань задоволеності клієнтів та більше.

- CMS Hub: платформа для управління контентом, яка дозволяє легко створювати та оптимізувати веб-сайти з метою залучення та конверсії відвідувачів.

Ключовою перевагою продуктів HubSpot є їх інтегрованість: всі інструменти працюють разом уніфіковано, що дозволяє компаніям мати єдину систему для всіх аспектів маркетингу, продажів і обслуговування.

Функціонал Hubspot:

1. CRM (Customer Relationship Management)
   - Управління контактами: Централізоване зберігання даних про контакти, компанії, угоди та завдання.
   - Відстеження взаємодій: Автоматичне відстеження електронних листів, дзвінків та зустрічей.
   - Керування воронкою продажів: Візуалізація та управління етапами продажів у реальному часі.
   - Автоматизація завдань: Мінімізація рутинних завдань за допомогою автоматизації.
2. Marketing Hub
   - Приваблення трафіку: Інструменти SEO, контент-маркетингу, соціальних медіа, та рекламних кампаній для залучення відвідувачів.
   - Конверсія відвідувачів у ліди: Створення лендінгів, форм, CTA (викликів до дії), та поп-апів.
   - Автоматизація маркетингу: Автоматизовані емейл кампанії, lead nurturing та scoring.
   - Аналітика та звіти: Детальні звіти про ефективність маркетингових кампаній та ROI.

3. Sales Hub

- Автоматизація продажів: Шаблони електронних листів, послідовності дій, та автоматизація задач.
- Управління угодами: Персоналізовані воронки продажів для кращого відстеження угод.
- Відстеження електронної пошти та дзвінків: Інтеграція з електронною поштою та можливість здійснення та відстеження дзвінків безпосередньо через платформу.
- Звіти та прогнози продажів: Аналітика продажів для кращого розуміння продуктивності та прогнозування.

4. Service Hub

- Управління запитами клієнтів: Створення та управління тікетами, автоматизація сервісних процесів.
- База знань: Створення та управління базою знань для самообслуговування клієнтів.
- Інструменти для збору відгуків: Опитування задоволеності клієнтів, NPS (Net Promoter Score), збір відгуків.
- Автоматизація обслуговування клієнтів: Розподіл тікетів, створення автоматичних відповідей та ескалаційних процедур.

5. CMS Hub

- Управління веб-контентом: Легке створення та редагування веб-сторінок без необхідності знання коду.
- SEO рекомендації: Інтегровані SEO інструменти для оптимізації контенту.
- Безпека веб-сайту: Автоматичне SSL шифрування та захист від загроз.
- Адаптивність та персоналізація: Створення адаптивних веб-сайтів з персоналізованим контентом для різних груп користувачів.

HubSpot забезпечує інтегровану платформу, яка дозволяє компаніям оптимізувати свої процеси залучення клієнтів, взаємодії з ними та управління відносинами, поєднуючи різноманітні функції в єдиному інтерфейсі.

Для початку дослідження була створена вибірка контактів за допомогою інструменту «списки» у Hubspot.

NAME	LEAD STATUS	CREATE DATE	LIFECYCLE STAGE	ARE YOU LOOKING FOR A ROOM OR FULL APARTMENT?
YASMINE Filali	Unqualified (to be removed)	May 24, 2024 1:57 AM GMT+3	Customer	Room
La'Treil Allen	New	May 23, 2024 2:29 PM GMT+3	Customer	Room
Cian Ahern	New	May 21, 2024 6:39 PM GMT+3	Customer	Room
Shaun Waul	New	May 13, 2024 3:13 PM GMT+3	Customer	Room
Olayemi Adeniran	New	May 6, 2024 3:12 AM GMT+3	Opportunity	Room
Jasonie Walker	Unqualified (to be removed)	May 5, 2024 1:39 PM GMT+3	Opportunity	Room
Manpreet Khaira	Unqualified (to be removed)	Apr 27, 2024 10:30 PM GMT+3	Opportunity	Room
Diane Le	Unqualified (to be removed)	Apr 23, 2024 5:06 PM GMT+3	Opportunity	Room
Pierre j Jean	Unqualified (to be removed)	Apr 15, 2024 4:35 AM GMT+3	Opportunity	Room
Shervon Williams	Unqualified (to be removed)	Apr 15, 2024 3:22 AM GMT+3	Opportunity	Room
Kiki Gikunda	Unqualified (to be removed)	Apr 11, 2024 2:17 AM GMT+3	Opportunity	Room
Monica Setaruddin	Unqualified (to be removed)	Apr 8, 2024 5:13 AM GMT+3	Opportunity	Room
Alissa Schumacher	Unqualified (to be removed)	Apr 4, 2024 3:17 PM GMT+3	Opportunity	Room
Phillip White	Unqualified (to be removed)	Apr 3, 2024 8:08 AM GMT+3	Opportunity	Room
Camille Truong	Unqualified (to be removed)	Apr 3, 2024 2:37 AM GMT+3	Opportunity	Room

Рис. 1.1. Вибірка контактів з Hubspot

Джерело: створено автором на основі [4]

Для фільтрування контактів був використаний фільтр по даті створення контакту починаючи з першого січня 2023 року. Також були обрані наступні поля для відображення: Record ID – Contact, First Name, Last Name, email, Lead Status, Create Date, Lifecycle Stage, Are you looking for a ROOM or FULL APARTMENT?, What is your budget?, What is your budget? (FULL APT), Original Source

У результаті було отримано 20000 записів, та збережено у файл clustering-data.csv (див табл. 1.1).

Таблиця 1.1. Огляд файлу clustering-data.csv

Record ID - Contact	First Name	Last Name	email	Create Date	Lifecycle Stage	Are you looking for a ROOM or FULL APARTMENT?	What is your budget?	Original Source
58655551	Merlin Lucas	Heiser	merlinlheiser@gmail.com	10/15/2023	Opportunity	Room	$1750 - $2000	Paid Search
58852351	Antoine	Clark	antoinelclark@gmail.com	10/22/2023	Opportunity	Room	$1250 - $1500	Paid Search
60293901	Catarina	Amendoeira	catarinacunhamendoeira@gmail.com	12/6/2023	Opportunity	Room	$1500 - $1750	Paid Search
59638551	Kate	Barkley	katelynbarkley@icloud.com	11/16/2023	Opportunity	Room	$1500 - $1750	Organic Search
58000351	Olive	Ok	liveokol@hotmail.com	9/30/2023	Opportunity	Room	$1250 - $1500	Paid Search
59376451	Kyra	Phillips	cras.kphillips@gmail.com	11/11/2023	Opportunity	Room	$1250 - $1500	Organic Search

Джерело: розрахунки автора на основі [4]

1.2 Ознайомлення з базою даних компанії

База даних компанії «Outpost Club» створена у середовищі MySQL та налійчує 86 таблиць з даними про користувачів, платіжною інформацією та даними про вебсайт компанії.

MySQL — це популярна система управління базами даних, яка використовує мову SQL для обробки даних. Вона є однією з найвідоміших реляційних баз даних і використовується в широкому спектрі застосунків, від невеликих місцевих сайтів до великих, високонавантажених веб-платформ і корпоративних систем.

Основні характеристики MySQL:

- Відкритий код: MySQL — це система з відкритим вихідним кодом, що означає, що кожен може її завантажити, використовувати та модифікувати без вартості ліцензій.
- Переносимість: MySQL підтримує різноманітні операційні системи, включаючи Linux, Windows, OS X, Solaris та інші.
- Надійність і стабільність: MySQL відома своєю надійністю. Вона має механізми для забезпечення цілісності даних, включаючи транзакції з підтримкою ACID, бекапи та відновлення.
- Гнучкість: MySQL підтримує різноманітні типи даних, що робить її придатною для різних типів додатків, від веб-сайтів до аналітичних систем.
- Простота використання: MySQL має простий синтаксис і легку до розуміння документацію, що робить її доступною навіть для початківців.

- Підтримка спільноти та комерційна підтримка: Через велику спільноту користувачів і розробників, для MySQL доступна велика кількість ресурсів для навчання та підтримки. Oracle, компанія, яка володіє MySQL, також пропонує комерційну підтримку і додаткові сервіси.
- Безпека: MySQL включає розширені можливості безпеки, такі як шифрування даних, управління доступом на основі ролей і аутентифікація користувачів.

Прикладні сфери застосування MySQL:

- Веб-розробка: MySQL є основою для багатьох веб-додатків і платформ, включаючи WordPress, Joomla, та Drupal.
- Електронна комерція: Великі та малі магазини використовують MySQL для обробки транзакцій, управління клієнтськими даними та аналітики.
- Фінанси: Банки та фінансові інститути використовують MySQL для аналізу даних, управління активами та виконання транзакцій.
- Корпоративні рішення: MySQL використовується для керування внутрішніми базами даних, від службових записів до клієнтських баз даних.

Після підключення до бази даних напишемо SQL запит для формування таблиці з такими полями: імейл, вік, стать, громадянство США, тип кімнати.

Запустимо наступний код:

```
SELECT DISTINCT email, YEAR(CURDATE()) - YEAR(birthday)
AS age, gender, us_citizen, s_bookings.type
FROM s_users
         JOIN s_bookings_users on
s_bookings_users.user_id = s_users.id
         JOIN s_bookings on s_bookings_users.booking_id =
s_bookings.id
WHERE YEAR(s_users.created) > 2022;
```

Рис. 1.2. SQL запит для вибірки даних про бронювання

Джерело: створено автором на основі [2]

Цей SQL-запит використовує три таблиці: s_users, s_bookings_users і s_bookings. Запит з'єднує ці таблиці за допомогою ідентифікаторів користувачів та бронювань, щоб отримати різні поля. Він вибирає електронну адресу, вік користувача обрахований як різниця між поточним роком та роком народження, стать, інформацію про громадянство США та тип бронювання. Умова в запиті обмежує вибірку лише користувачами, які були створені після 2022 року. Ключове слово DISTINCT гарантує, що кожен рядок в результаті буде унікальний за комбінацією вказаних полів. Таким чином, цей запит надає змогу переглянути унікальну інформацію про нових користувачів та їхні бронювання.

Збережемо цей запит у вигляді таблиці з назвою s_users.csv. Ця таблиця налічує 5500 записів (див табл. 1.2).

Таблиця 1.2. Огляд файлу s_users.csv

email	age	gender	us_citizen	type
	29	2	1	1
	31	1	2	1
	26	2	2	1
	30	2	1	1
	23	2	2	1
	23	2	2	2
	36	2	1	1
	22	2	2	1
	28	2	1	1
	25	2	2	1

Джерело: розрахунки автора на основі [2]

Для виконання оглядового аналізу потрібно об’єднати таблиці s_users.csv та clustering-data. Для цього скористаємось середовищем Python

На початку коду виконується імпорт бібліотеки pandas. Pandas - це популярна бібліотека для обробки і аналізу даних, яка надає структури даних та функції для роботи з табличними даними.

Дані з двох CSV файлів, clustering-data.csv та s_users.csv, що знаходяться на робочому столі, зчитуються та зберігаються в змінних hubspot та s_users відповідно. Потім ці два датасети об'єднуються по спільному стовпцю email, за допомогою функції pd.merge(), причому залишаються лише ті рядки, які присутні в обох датасетах. Результат злиття зберігається в змінну data.

У результаті отримуємо таблицю, яка містить 3000 записів (див. табл. 1.3)

Таблиця 1.3. Огляд файлу KMeans_data.csv

First Name	Last Name	email	Lead Status	Lifecycle Stage	Original Source	age	gender	us_citizen	type
Viola	Albright		Customer	Application Fee Paid	Paid Search	21	1	1	1
India	Boonen		Alumni	Customer	Paid Search	23	1	2	1
Alena	Mauger		Alumni	Customer	Offline Sources	31	1	2	1
Theresa	Harper-Harris			Lead	Offline Sources	40	1	1	1
Aditya Kumar	Gudimella Tirumala		Alumni	Customer	Direct Traffic	32	2	2	1
Audrey	E Moreland		Alumni	Customer	Offline Sources	25	1	1	1
Duran	Crooks		Customer	Customer	Direct Traffic	27	2	1	1
Nathaniel	Stratton			Lead	Offline Sources	23	2	1	2
Norin	Ouch		Customer	Customer	Organic Search	21	2	2	1
Sophia	Calvert		Customer	Customer	Direct Traffic	21	1	1	1
			Alumni	Lead	Direct Traffic	23	1	2	1

Джерело: розрахунки автора на основі [5]

1.3 Візуалізація даних

Для візуалізації даних користувачів скористаємося програмою Tableu.

Tableau - це потужний інструмент візуалізації даних, який широко використовується у сфері бізнес-аналітики для створення наочних інтерактивних звітів та дашбордів. Цей інструмент дозволяє користувачам перетворювати сирі дані в легко інтерпретовану форму, що допомагає компаніям приймати обґрунтовані рішення на основі даних. Ось кілька ключових аспектів Tableau:

1. Інтуїтивно зрозумілий інтерфейс: Tableau пропонує дружній та інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам легко перетягувати різні компоненти для створення візуалізацій. Це робить Tableau доступним не тільки для аналітиків даних, але й для бізнес-користувачів.
2. Потужні можливості візуалізації: За допомогою Tableau можна створювати широкий спектр візуалізацій, від простих гістограм та кругових діаграм до складних географічних карт і тривимірних графіків.
3. Інтеграція даних: Tableau може інтегруватися з різними джерелами даних, включаючи Excel, SQL бази даних, хмарні сховища даних і багато інших. Це забезпечує гнучкість при роботі з різними форматами даних.
4. Динамічні дашборди: Користувачі можуть створювати інтерактивні дашборди, які оновлюються у реальному часі. Це дозволяє користувачам взаємодіяти з даними та проводити глибокий аналіз.
5. Співпраця та ділення: Tableau підтримує співпрацю, дозволяючи користувачам ділитися своїми візуалізаціями та дашбордами з колегами або вбудовувати їх у веб-сторінки.
6. Розширені аналітичні можливості: Крім стандартних візуалізацій, Tableau пропонує продвинуті аналітичні функції, такі як прогнозування, аналіз тенденцій і статистичний аналіз.
7. Навчання та спільнота: Tableau має обширну спільноту користувачів і навчальні ресурси, включаючи онлайн-курси, форуми та вебінари, які допомагають новим користувачам освоїтися з інструментом.

Tableau особливо цінується за свою здатність швидко перетворювати складні набори даних у зрозумілу і візуально привабливу форму, що робить його незамінним інструментом у арсеналі сучасного аналітика.

Створимо візуалізації у Tableau для розвідувального аналізу користувачів.

Відобразимо розподіл користувачів за гендером (див. рис. 1.3).

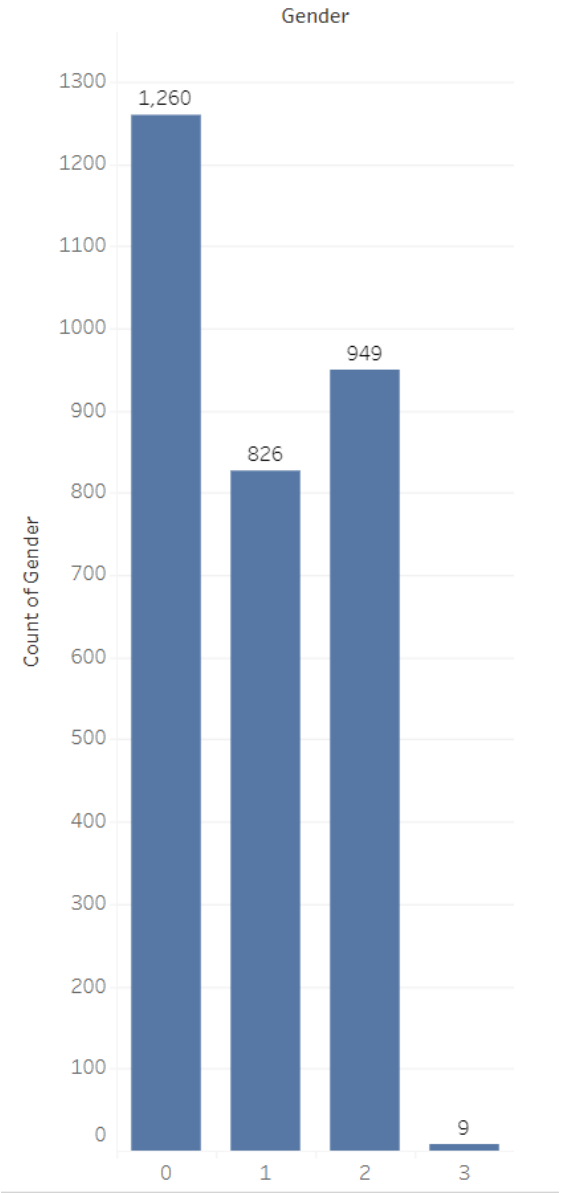

Рис. 1.3. Розподіл користувачів за гендером

Джерело: побудовано автором на основі [3]

На рис. 1.2 гендер був позначений цифрами, відповідно: 0 – невідомий, 1 – жінки, 2 – чоловіки, 3 – інше.

Відобразимо розподіл користувачів за каналом трафіку (див. рис. 1.4).

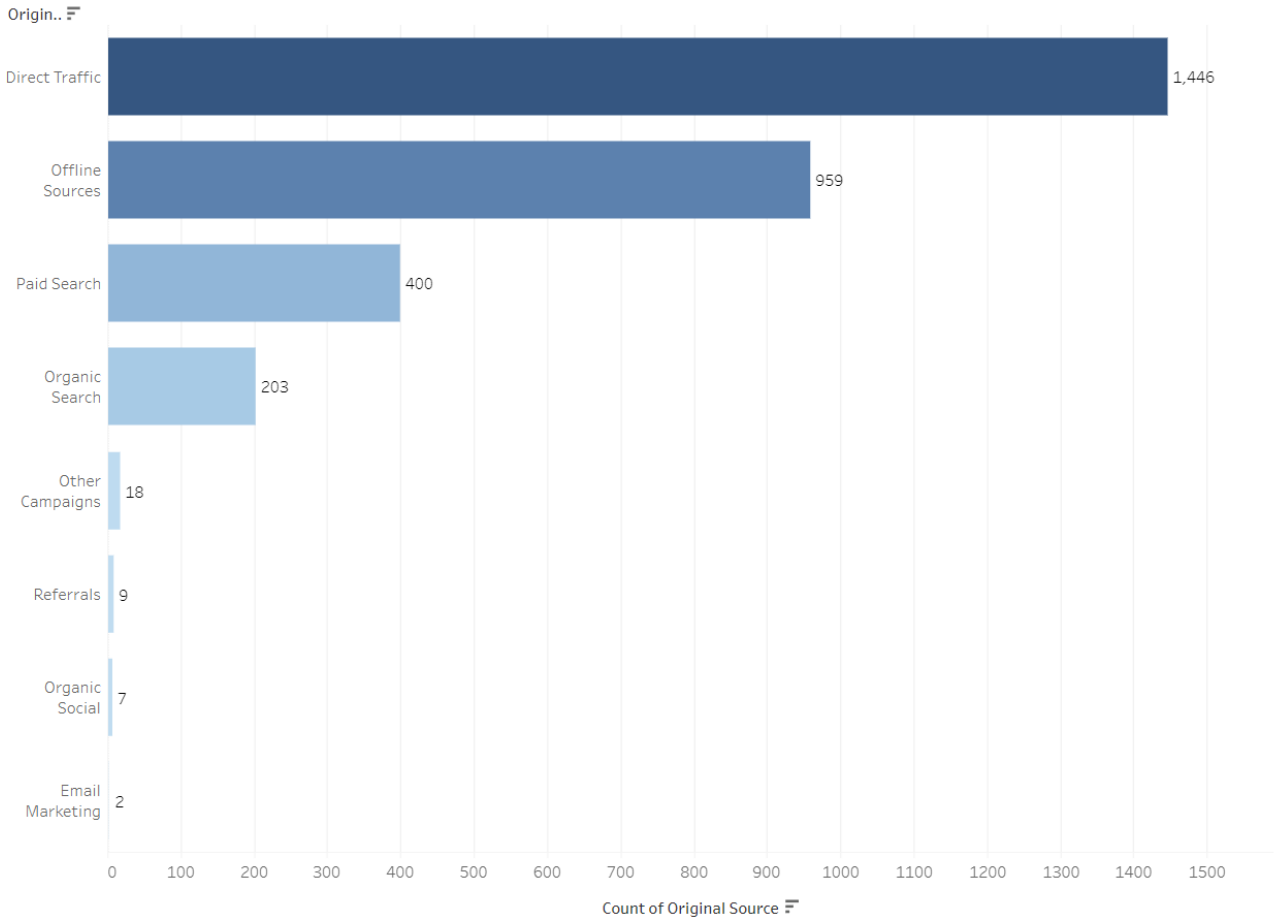

Рис. 1.4. Розподіл користувачів за каналом трафіку

Джерело: побудовано автором на основі [3]

Відобразимо розподіл користувачів за бюджетом (див. рис. 1.5).

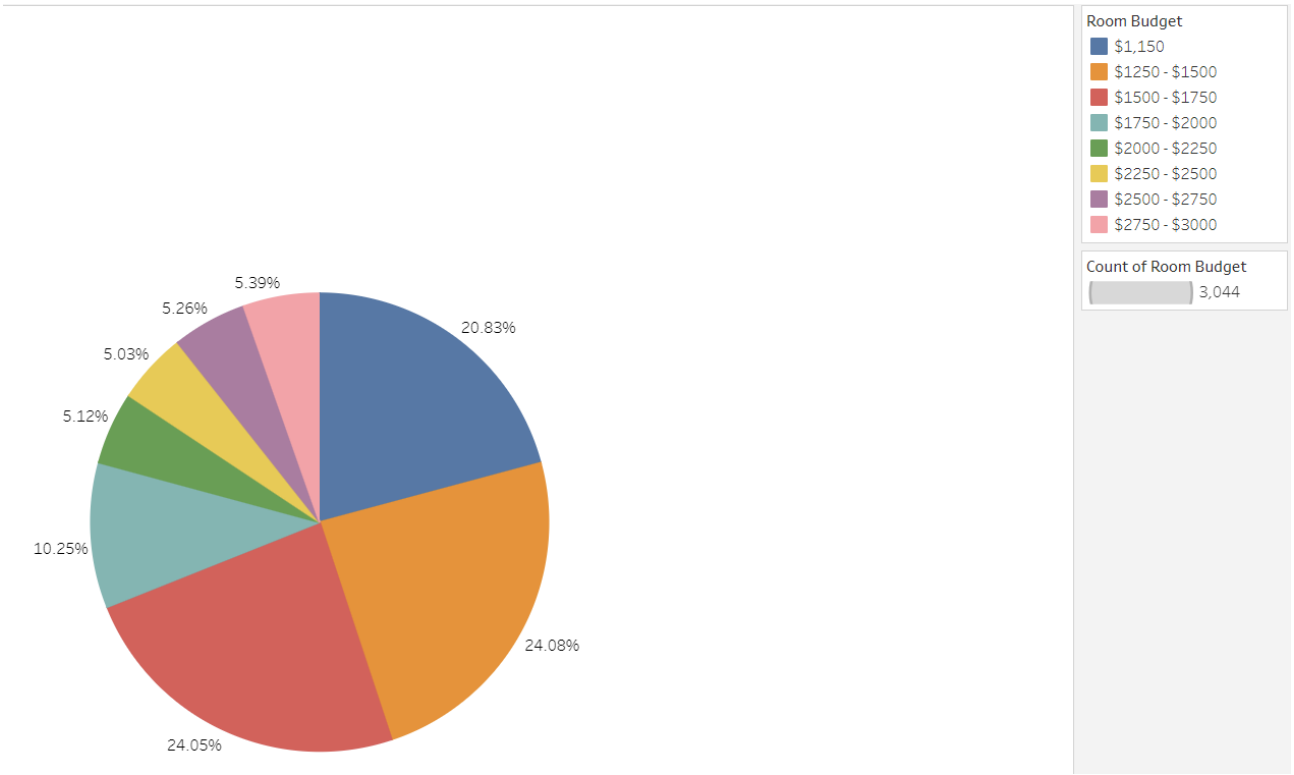

Рис. 1.5. Розподіл користувачів за бюджетом

Джерело: побудовано автором на основі [3]

Для того щоб відобразити розподіл користувачів за віком були створені кошики з кроком у два роки (див. рис. 1.6).

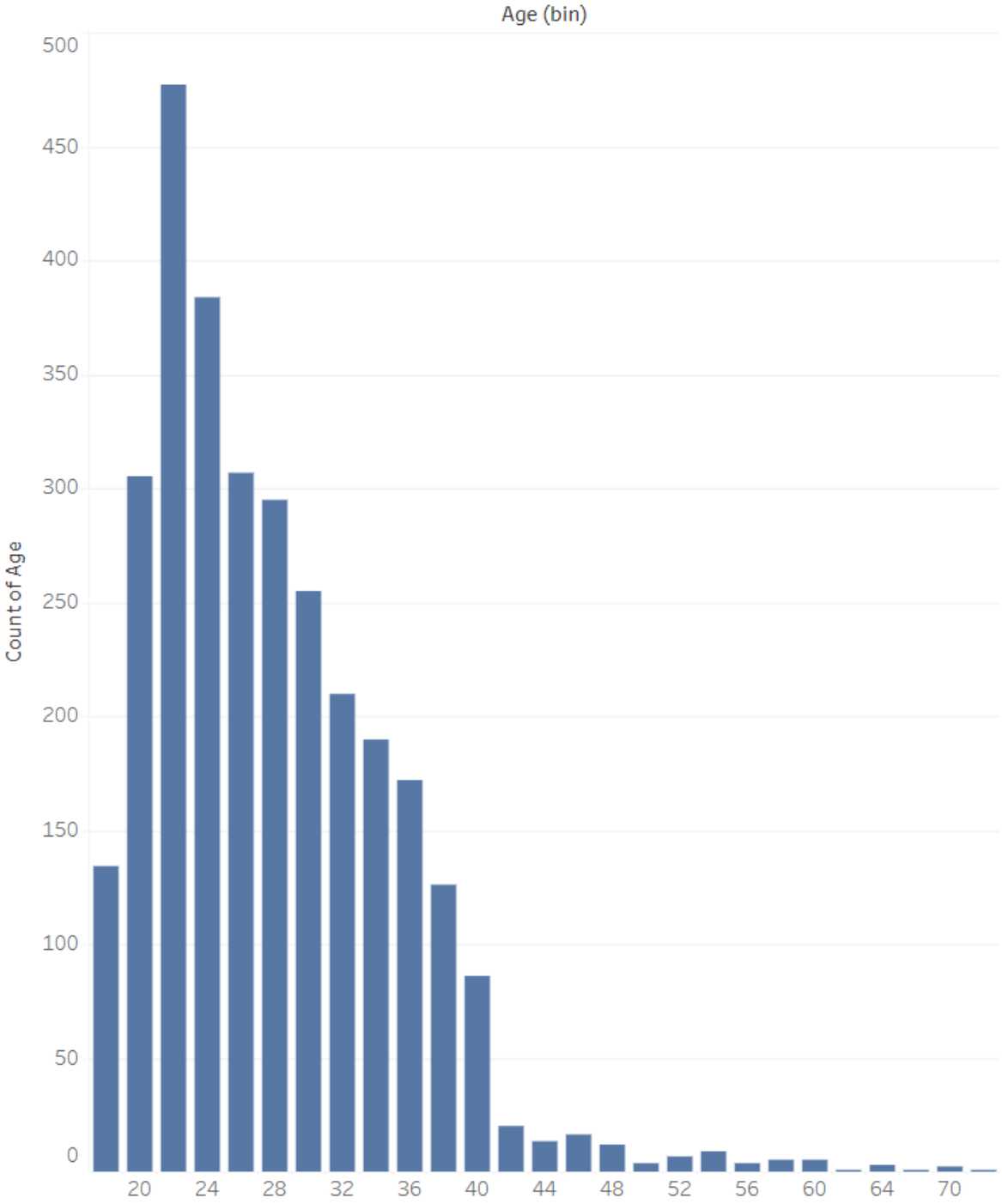

Рис. 1.6. Розподіл користувачів за віком

Джерело: побудовано автором на основі [3]

Наведені візуалізації, створені у Tableau, що включають розподіл користувачів за гендером, каналом трафіку, бюджетом та віком. Кожна з цих візуалізацій ілюструє важливі аспекти користувацьких даних, що допомагає краще зрозуміти аудиторію та приймати більш точні бізнес-рішення.

Наприклад, розподіл користувачів за каналом трафіку дозволяє визначити найефективніші канали залучення користувачів, що допомагає оптимізувати маркетинговий бюджет та зосередити зусилля на найпродуктивніших каналах. Аналіз розподілу користувачів за гендером допомагає краще зрозуміти демографічний склад аудиторії, що впливає на контент-стратегію та орієнтування рекламних кампаній на певні групи користувачів. Дані про розподіл за віком допомагають визначити вікові групи, на які слід спрямувати маркетингові зусилля, адаптуючи продукти чи послуги до потреб цих груп.

Розподіл користувачів за бюджетом дає можливість оцінити платоспроможність різних сегментів аудиторії, що сприяє кращому плануванню ціноутворення та розробці різних цінових пропозицій.

Використання Tableau дозволяє швидко перетворювати складні набори даних у зрозумілу і візуально привабливу форму, що робить його незамінним інструментом для сучасних аналітиків. Таким чином, було показано, що візуалізація даних за допомогою Tableau є ефективним засобом для проведення розвідувального аналізу для розуміння аудиторії.

РОЗДІЛ 2. КЛАСТЕРИЗАЦІЯ ЗА ДОПОМОГОЮ МОДЕЛІ K-MEANS

2.1 Математичний апарат моделі K-means

Математичний апарат моделі k-means включає в себе ряд концепцій та методик, які лежать в основі одного з найпопулярніших алгоритмів кластеризації в області машинного навчання та аналізу даних. K-means використовується для групування набору об'єктів у такий спосіб, що об'єкти в одному кластері є більш схожими один до одного, ніж до об'єктів у інших кластерах. Цей алгоритм знаходить широке застосування у багатьох областях, включаючи ринкову сегментацію, організацію документів, образотворче мистецтво та багато іншого.

K-means є методом векторної квантизації, що організовує набір даних у k кластерів, кожен з яких характеризується своїм центром або середнім (центроїдом), який зазвичай вираховується як середнє арифметичне всіх точок, що належать до цього кластеру.

Нехай задано множину спостережень $X=\{x_1,x_2,\ldots,x_n\}$, де кожне спостереження є d-вимірним реальним вектором. K-means класифікує ці спостереження на k кластерів $G=\{G_1,G_2,\ldots,G_k\}$, мінімізуючи внутрішньокластерну суму квадратів відстаней від точок до центроїдів своїх кластерів [16]:

$$S = \sum_{i=1}^{k} \sum_{x \in G_i} \| x - u_i \|^2 \qquad (2.1)$$

де μ_i є центроїдом кластера G_i.

Для запуску алгоритму k-means необхідно визначити кількість кластерів k, на які будуть розділені дані, та вибрати початкові центри цих кластерів, відомі як центроїди. Центроїди часто вибираються випадково з множини спостережень, хоча існують різні стратегії їх ініціалізації для покращення результатів алгоритму.

Основа роботи алгоритму полягає в ітеративному переобчисленні приналежності елементів до кластерів і переміщенні центроїдів. На кожному кроці кожен об'єкт датасету присвоюється до кластера, центроїд якого знаходиться найближче до нього. Після цього центроїди кожного кластера оновлюються так, що вони стають середнім значенням усіх об'єктів, приписаних до відповідного кластера.

Цей процес повторюється до тих пір, поки не буде досягнуто стабільності у положенні центроїдів, або інша умова зупинки не вступить в дію. Такі умови можуть включати максимальну кількість ітерацій чи мінімальну зміну в положенні центроїдів між ітераціями.

Однак, алгоритм має певні недоліки. Він чутливий до вибору початкових центроїдів, що може призвести до менш оптимальних рішень у формі локальних мінімумів. Також k-means припускає, що кластери мають сферичну форму і розподіл однакової величини, що не завжди є правдою для реальних даних.

Незважаючи на ці обмеження, k-means залишається популярним вибором через свою простоту, ефективність і широкий спектр застосування. Якість кластеризації, отриманої за допомогою k-means, може бути оцінена за допомогою різних метрик, які допоможуть зрозуміти, наскільки добре модель виконує свою роботу у групуванні даних.

Вибір кількості кластерів у моделі k-means є одним із ключових кроків у процесі кластеризації, який значно впливає на результати аналізу. Правильний вибір кількості кластерів (k) може допомогти досягти більш точної та інформативної сегментації даних.

Метод ліктя є одним із найпопулярніших способів визначення оптимальної кількості кластерів. Він полягає в побудові графіка, на якому по осі X відкладається кількість кластерів, а по осі Y — сума квадратів відстаней від кожної точки до центру найближчого кластера (це звано сумою внутрішньокластерних квадратів відстаней, англ. WCSS — within-cluster sum of squares). Після обчислення WCSS для різних значень k, графік зазвичай має форму, яка нагадує руку з зігнутим ліктем. Кількість кластерів, що відповідає точці "ліктя" (де графік починає виплощуватися), вважається оптимальною.

Силуетний аналіз вимірює якість кластеризації, визначаючи, наскільки добре кожен об'єкт був віднесений до свого кластера. Коефіцієнт силуету може варіюватися від -1 до +1, де високе значення вказує на те, що об'єкт добре вписується в свій кластер і погано сумісний з сусідніми кластерами. Перевага цього методу полягає в тому, що він дає як глобальне оцінювання всієї кластеризації, так і оцінку кожного окремого кластера.

Часто вибір кількості кластерів може базуватися на доменних знаннях або бізнес-вимогах. Наприклад, у маркетингових дослідженнях кількість кластерів може відповідати кількості цільових сегментів ринку, які компанія планує розрізнити.

Також можна використовувати статистичні інформаційні критерії, такі як критерій Акаіке (AIC) чи Байєсівський інформаційний критерій (BIC), для вибору кількості кластерів. Ці методи оцінюють моделі, беручи до уваги кількість параметрів моделі та доброту відповідності моделі даним.

Хоча k-means зазвичай не використовує перехресну валідацію в традиційному розумінні, можливий підхід полягає в розподілі даних на набори, незалежному застосуванні k-means до кожного набору та аналізі стабільності результатів кластеризації. Якщо певне значення k призводить до консистентно схожих кластерних структур у різних наборах, це може вказувати на його адекватність.

Вибір кількості кластерів не є точною наукою і часто потребує комбінації вищеописаних методів, а також тестування та налагодження залежно від контексту та доступних даних.

Алгоритм k-means має також декілька обмежень та складнощів, які варто враховувати при аналізі даних. Одна з основних проблем полягає у необхідності заздалегідь визначити кількість кластерів k, що може бути складно без детального розуміння структури даних. Це може потребувати додаткових обчислень і не завжди гарантує точність.

Також результати кластеризації можуть залежати від випадково вибраних початкових центрів кластерів, що може призвести до локальних оптимумів замість глобального. Часто рекомендують кількаразово запускати алгоритм з різними стартовими центрами, аби зменшити ризик такої залежності.

Алгоритм також чутливий до викидів у даних, які можуть значно впливати на визначення центрів кластерів, і в кінцевому результаті — на кластеризацію. Викиди можуть зміщувати центри кластерів, що може призвести до неправильного групування даних.

K-means найефективніший, коли кластери є гіперсферичними та мають приблизно однаковий розмір. Якщо кластери не відповідають цим характеристикам, як-от у випадках, коли кластери мають нерегулярні форми або розміри, алгоритм може мати труднощі з їх ідентифікацією.

Ще одна важлива проблема полягає у використанні євклідової відстані для вимірювання схожості між точками, що робить результати чутливими до масштабування особливостей. Це означає, що перед використанням k-means важливо стандартизувати дані, щоб усі особливості вносили однаковий вплив у визначення кластерів.

Нарешті, k-means може не бути ідеальним для дуже великих наборів даних через великі обчислювальні витрати, зумовлені необхідністю багаторазових обчислень відстаней між точками даних і центрами кластерів.

Ці проблеми стимулювали розробку численних варіацій алгоритму. Одна з таких варіацій — k-means++, яка покращує вибір початкових центрів. Замість випадкового вибору, перший центр вибирається випадково, а кожен наступний центр обирається з урахуванням розподілу відстаней до вже обраних центрів, що значно зменшує ймовірність поганого вибору та прискорює збіжність алгоритму.

Інша популярна альтернатива — Mini-Batch K-Means, який використовує менші підвибірки набору даних для кожної ітерації. Це дозволяє зменшити витрати пам'яті та час обчислень, що є особливо корисним при роботі з великими наборами даних.

K-medoids, відомий також як метод PAM (Partitioning Around Medoids), замінює усереднені центри кластерів на найбільш типові об'єкти, що робить метод менш чутливим до викидів та більш стійким у різноманітних застосуваннях.

Також існує ієрархічний k-means, який спочатку використовується для поділу великого кластера на менші, а потім ці менші кластери можуть бути оброблені за допомогою стандартного k-means, дозволяючи більш точно контролювати процес кластеризації та ієрархію кластерів.

На відміну від звичайного k-means, де кожен об'єкт належить лише одному кластеру, Fuzzy k-means вводить концепцію часткової приналежності, дозволяючи об'єктам належати до кількох кластерів одночасно з різними ступенями приналежності. Це робить алгоритм здатним ефективніше обробляти датасети, де межі між кластерами не чітко визначені.

Вибір підходящої варіації k-means та коректне визначення числа кластерів k є ключовими для успіху в реалізації проектів машинного навчання, де важлива кластеризація. Тому перед застосуванням цього алгоритму важливо звернути увагу на специфіку даних та задачі.

Загалом K-means залишається одним із найбільш популярних алгоритмів кластеризації завдяки своїй простоті, ефективності та широкому застосуванню в різних областях. Його математичний апарат є відносно простим для розуміння, але водночас достатньо потужним, щоб забезпечити реальні рішення для складних задач обробки та аналізу даних.

2.2 Підготовка даних для моделі K-means

Для побудови моделі k-means якісні дані потрібно перевести у кількісні. Для вирішення цієї задачі була застосована бібліотека Pandas для мови Python та файл KMeans_data.csv, створений у розділі 1.2. Колонки, які треба трансформатувати: “Room_Budget”, “Lifecycle Stage” та “Original Source”.

Завантажемо файл KMeans_data.csv за допомогою бібліотеки Pandas та перевіремо типи даних за допомогою команди df.dtypes(). Результат виведемо у таблицю 2.1.

Таблиця 2.1 Типи даних

Назва	Тип
Lifecycle Stage	object
Room_Budget	object
Original Source	object
age	float64
gender	int64
us_citizen	int64
room_type	int64

Джерело: розрахунки автора на основі [5]

Напишемо код для виконання самого кодування на прикладі колонки “Original Source”:

Кожне категоріальне значення було позначено числом від 1 до 8

Був створений словник, де кожному типу джерела присвоюється числове значення. Значення в стовпці “Original Source” датасету df замінюються відповідними числовими значеннями з цього словника, використовуючи метод map.

За таким самим принципом були перетворені значення для “Room_Budget” та “Lifecycle Stage”. У результаті усіх перетворень отримаємо наступну таблицю (див. табл. 2.2) та збережемо її у файл encoded_data.csv.

Таблиця 2.2 Огляд файлу encoded_data.csv

Lifecycle Stage	Room_Budget	Original Source	age	gender	us_citizen
3	2	1	26	0	0
4	1	4	27	0	0
3	8	3	35	0	0
4	2	7	20	0	0
1	1	7	21	1	1

Джерело: розрахунки автора на основі [5]

2.3 Побудова моделі K-means

Для побудови моделі k-means скористаємося бібліотекою sklearn та файлом encoded_data.csv, який був створений у попередньому розділі.

Були імпортовані потрібні бібліотеки та завантажений файл для роботи:

- import pandas as pd: Імпорт бібліотеки Pandas, яка використовується для обробки та аналізу даних. Вона надає зручні структури даних і функції для роботи з табличними даними.
- import numpy as np: Імпорт бібліотеки NumPy, важливий інструмент для числових обчислень у Python. Вона використовується для роботи з масивами, матрицями та широким спектром математичних функцій.
- from sklearn.cluster import KMeans: Імпорт алгоритму K-Means з бібліотеки scikit-learn, який використовується для кластеризації даних.
- from sklearn.metrics import silhouette_score: Імпорт функції для обчислення силуетного індексу, який може бути використаний для оцінки якості кластеризації, що виконується алгоритмом K-Means.
- from matplotlib import pyplot as plt: Імпорт pyplot з бібліотеки matplotlib, яка є стандартним інструментом для візуалізації даних у Python.

Для побудови моделі спочатку було визначено оптимальну кількість кластерів за допомогою методу ліктя та суми квадратів помилок (SSE).

Сума квадратів помилок, є широко використовуваною мірою в статистиці для оцінки різниці між прогнозованими значеннями та реальними даними. Цей метод зазвичай використовується при лінійній регресії та інших статистичних моделях для визначення "найкращої" лінії або кривої, що апроксимує набір даних.

Визначення SSE [17]:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2.2}$$

- *SSE* — сума квадратів помилок,
- y_i — реальне значення відповідно до і-го спостереження,
- $\hat{y}_i$ — прогнозоване значення за і-те спостереження,
- n — загальна кількість спостережень.

Суть цього показника полягає у вимірюванні загальної помилки моделі, сумуючи квадрати відхилень прогнозованих значень від фактичних. Квадратування допомагає врахувати як позитивні, так і негативні відхилення, забезпечуючи, що всі помилки вважаються "штрафами", та зосереджує увагу на більших помилках.

Мінімізація SSE є ключовим компонентом в методах найменших квадратів, де параметри моделі (наприклад, коефіцієнти у лінійній регресії) вибираються таким чином, щоб сума квадратів помилок була якомога меншою, що вказує на більш високу точність моделі щодо використаних даних.

Виконаємо обчислення суми квадратів помилок (SSE) для різних кількостей кластерів використовуючи алгоритм K-Means.

Створюється порожній список sse. Встановлюється діапазон значень для кількості кластерів від 1 до 9. Для кожного значення k з цього діапазону створюється об'єкт KMeans з відповідною кількістю кластерів, модель навчається на даних df_imputed, і значення інерції моделі додається до списку sse.

Побудуємо графік для визначення оптимальної кількості кластерів за допомогою бібліотеки matplotlib для візуалізації даних (див. рис. 2.1).

Мітка для осі X встановлюється як 'K'. Мітка для осі Y встановлюється як 'Sum of squared error'. Лінійний графік створюється для відображення значень кількості кластерів по осі X і відповідних значень суми квадратів похибок по осі Y.

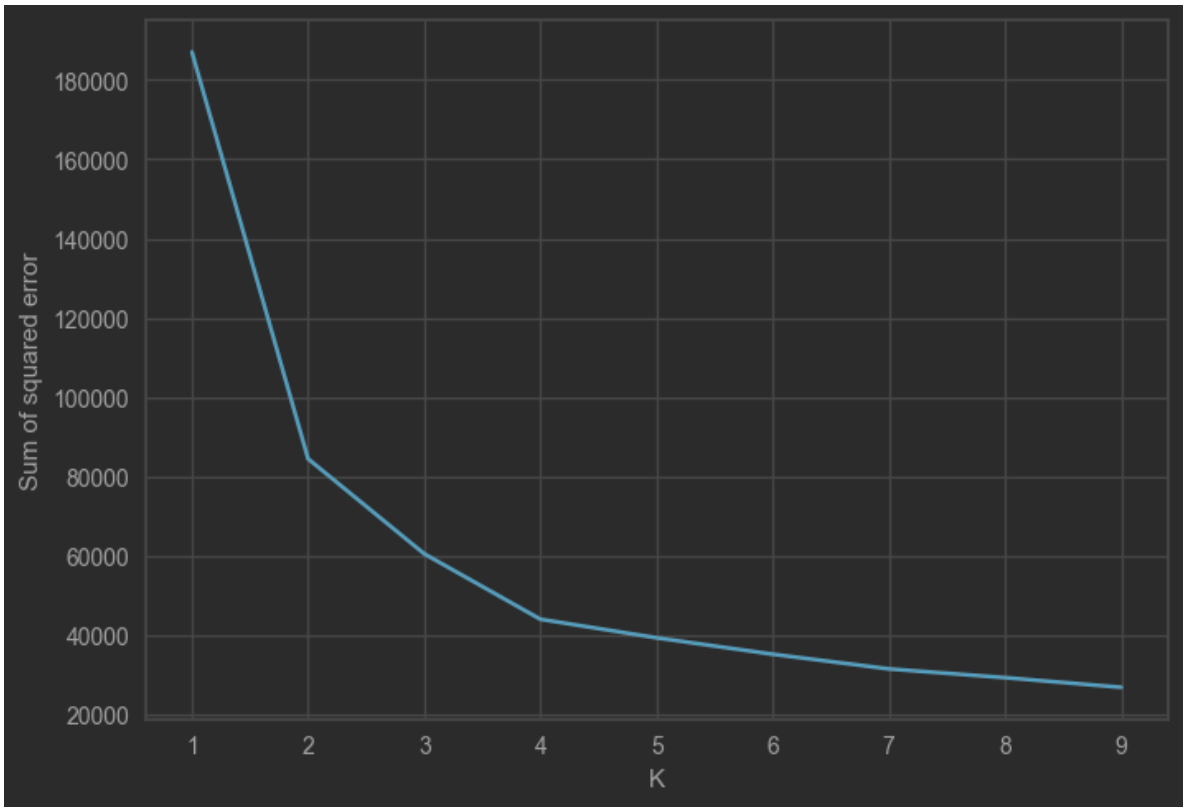

Рис. 2.1. Застосування методу ліктя для визначення оптимальної кількості кластерів

Джерело: побудовано автором на основі [6]

З рис. 2.1. чітко видно, що оптимальна кількість кластерів для побудови моделі це - 4.

Напишемо код для побудови моделі:

```
kmeans = KMeans(n_clusters=4)
clusters = kmeans.fit_predict(df_imputed)
df_imputed['Cluster'] = clusters
```

Рис. 2.2. Створення моделі KMeans

Джерело: створено автором на основі [7]

Цей код використовується для кластеризації даних за допомогою алгоритму K-means для розділення даних на передвизначену кількість кластерів:

1. kmeans = KMeans(n_clusters=4): Ініціалізація об'єкта KMeans із бібліотеки sklearn.cluster. Тут параметр n_clusters=4 означає, що алгоритм намагатиметься розділити дані на чотири кластери.
2. clusters = kmeans.fit_predict(df_imputed): Застосування методу fit_predict до набору даних df_imputed. Цей метод спочатку тренує модель K- means на наборі даних (визначаючи центри кластерів), а потім призначає кожному спостереженню у наборі даних мітку кластеру відповідно до найближчого центру кластеру. Результатом є масив clusters, де кожен елемент є міткою кластеру для відповідного спостереження в df_imputed.
3. df_imputed['Cluster'] = clusters: Додає новий стовпчик до DataFrame df_imputed під назвою 'Cluster', куди записується масив міток кластерів. Таким чином, кожному рядку в df_imputed присвоюється мітка відповідного кластеру, до якого він належить.

Відобразимо результати та порахуємо силуетну міру для оцінки моделі.

Спочатку дані у DataFrame групуються за стовпцем 'Cluster', що містить мітки кластерів. Після групування обчислюється медіана для кожного стовпця у кожній групі, створюючи новий DataFrame cluster_characteristics, який містить моди усіх значень, окрім віку, вік агрегується за середнім показником для кожного кластера.

Далі видаляється стовпець 'Cluster' з df_imputed, залишаючи лише фактичні дані для кластеризації. Функція silhouette_score вимірює, наскільки добре кожне спостереження відповідає своєму кластеру в порівнянні з іншими кластерами. Вона може приймати значення від -1 до 1, де вищі значення означають кращу відповідність. Значення silhouette_avg відображає середнє значення цього індексу по всім спостереженням. Силуетна міра має коефіцієнт: 0.3545.

Нарешті, показуються перші п'ять рядків з DataFrame cluster_characteristics (див. табл. 2.3), щоб бачити медіанні значення характеристик для перших кількох кластерів.

Таблиця 2.3. Створення кластерів

Cluster	Lifecycle Stage mode	Room_Budget mode	Original Source mode	age avg	gender mode	us_citizen mode
1	2	3	3	29	1	1
2	2	3	3	22	1	1
3	2	3	1	36	0	0
4	2	2	3	51	2	1

Джерело розрахунки автора на основі [5]

Переведемо кількісні дані назад у якісні для більш зручного аналізу (див. табл. 2.4).

Таблиця 2.4. Результати кластеризації

Cluster	Lifecycle Stage mode	Room_Budget mode	Original Source mode	age avg	gender mode	us_citizen mode
1	Customer	$1500 - $1750	Offline Sources	29	жінки	США
2	Customer	$1500 - $1750	Offline Sources	22	жінки	США
3	Customer	$1500 - $1750	Direct Traffic	36	невідомий	невідомий
4	Customer	$1250 - $1500	Offline Sources	51	чоловіки	США

Джерело: розрахунки автора на основі [5]

Силуетний коефіцієнт 0.3545 вказує на середній рівень кластеризації. Це може означати, що хоча кластери дещо чітко визначені, все ж таки існує певний ступінь перекриття між ними.

Кластер 1 і 2 мають жіночу більшість, використовують офлайн джерела, є громадянами США, і мають приблизно однаковий бюджет на житло. Відмінність в основному у віці (29 проти 22 років). Це вказує на поділ на основі вікової категорії у рамках подібної демографічної групи.

Кластер 3 Має найбільш різні характеристики. Цей кластер використовує прямий трафік і має середній вік 36 років з невідомим гендером та громадянством. Відмінність у джерелі трафіку та невідомих демографічних даних робить цей кластер унікальним порівняно з іншими.

Кластер 4 Складається переважно з чоловіків, старших за інші групи (середній вік 51 рік), та має трохи нижчий бюджет на житло. Це також єдина група, де домінує чоловіча стать.

Дані отримані у результаті виконаного аналізу можна застосувати для побудови більш ефективних рекламних кампаній у діджитал маркетингу, адже переважна більшість клієнтів прийшла з оффлайн ресурсів. Таким чином демографічні дані та дані про бюджет клієнтів можна використати для таргетування на ці групи людей та показувати їм пропозиції саме з такими цінами для підвищення конверсій та прибутку.

Загалом застосування даних кластеризації дозволяє не лише більш точно таргетувати різні сегменти аудиторії, але й забезпечує розуміння їх унікальних потреб та поведінки. Це сприяє більш ефективному використанню маркетингових ресурсів та підвищує ймовірність успішного залучення клієнтів. Персоналізація та точне таргетування є ключовими для досягнення кращих результатів у маркетингових кампаніях.

РОЗДІЛ 3. Прогнозування прибутку за допомогою моделі XGBoost

3.1 Математичний апарат моделі XGBoost

XGBoost (eXtreme Gradient Boosting) — це потужна бібліотека машинного навчання, яка використовує градієнтний бустинг для реалізації високопродуктивних моделей. Градієнтний бустинг є методом ансамблювання, який побудований на послідовному додаванні нових моделей для корекції помилок попередніх моделей. Цей підхід дозволяє створювати сильніші прогнозуючі моделі шляхом поступового покращення точності. В основі градієнтного бустингу лежить концепція використання слабких моделей, які зазвичай є простими деревами рішень, для поступового зменшення залишкових помилок, зроблених попередніми моделями.

XGboost використовує метод побудови ансамблю слабких моделей, кожна з яких намагається виправити помилки своїх попередників. Основна ідея XGBoost полягає в тому, що кожне нове дерево рішень додається до моделі таким чином, щоб зменшити помилки, які залишилися після попередніх дерев. Це досягається за допомогою оптимізації функції втрат і використання градієнтів для корекції прогнозів.

XGBoost також має кілька ключових особливостей, які роблять його особливо ефективним:

- Регуляризація: XGBoost включає регуляризаційні терміни, які допомагають уникнути перенавчання моделі.
- Обробка відсутніх значень: Модель може ефективно працювати з даними, що містять пропущені значення, шляхом визначення оптимальних шляхів обробки таких випадків.
- Паралельне обчислення: Завдяки підтримці паралельного обчислення, XGBoost може швидко обробляти великі обсяги даних.
- Крос-валідація: XGBoost надає вбудовані засоби для крос-валідації, що дозволяє більш точно оцінити ефективність моделі.

- Можливість налаштування параметрів: XGBoost має багато гіперпараметрів, які можуть бути налаштовані для досягнення оптимальної продуктивності моделі.

Завдяки своїм можливостям і гнучкості, XGBoost став одним із найпопулярніших інструментів для вирішення завдань машинного навчання, включаючи задачі класифікації, регресії та ранжування.

Функція витрат є одним із ключових елементів у будь-якій моделі машинного навчання, включаючи XGBoost. Вона визначає, наскільки добре або погано модель працює, порівнюючи прогнозовані значення з фактичними значеннями. Вибір відповідної функції витрат залежить від типу завдання — регресії чи класифікації.

У XGBoost функція витрат впливає на побудову дерев і оптимізацію параметрів. XGBoost намагається мінімізувати обрану функцію витрат, коригуючи ваги і структуру дерев на кожному етапі бустінгу. Функція витрат також враховується під час регуляризації для уникнення перенавчання моделі.

Основні функції витрат у XGBoost

Для регресійних задач:

- Mean Squared Error (MSE) [18]: Використовується для завдань регресії і обчислюється як середнє квадратичне відхилення між прогнозованими і фактичними значеннями.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.1}$$

- Mean Absolute Error (MAE) [19]: Використовується, коли важливо мінімізувати середнє абсолютне відхилення між прогнозами і фактичними значеннями.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{3.2}$$

Для класифікаційних задач:

- Logistic Loss (Log Loss) [20]: Використовується для задач бінарної класифікації і обчислює негативне логарифмічне правдоподібність передбачуваних класів.

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{3.3}$$

- Multi-class Log Loss [19]: Узагальнення логістичної втрати для задач багатокласової класифікації.

$$Multi-classLogLoss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{i=1}^{M} y_{i,j}\log(p_{i,j}) \tag{3.4}$$

Алгоритм додавання нових дерев у моделі XGBoost є одним з ключових компонентів цієї потужної техніки машинного навчання. Спочатку модель XGBoost створює прогнозні значення, використовуючи базове передбачення, наприклад, середнє значення цільової змінної для регресії або лог-значення ймовірностей для класифікації. На кожній ітерації XGBoost обчислює різницю між фактичними значеннями цільової змінної та поточними прогнозними значеннями. Ці різниці називаються залишками або псевдо-резидуалами.

Наступний крок - побудова нового дерева, яке буде навчатися на залишках. Кожне нове дерево намагається зменшити залишки, тобто покращити прогнози моделі. В XGBoost дерева будуються з використанням алгоритму CART (Classification and Regression Trees). На кожному кроці t, XGBoost додає нове дерево f_t до ансамблю, яке мінімізує залишкову помилку поточної моделі. Нова модель f_t шукається так, щоб мінімізувати наступну функцію [22]:

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{3.5}$$

де $\hat{y}_i^{(t-1)}$ — передбачення поточної моделі на кроці *t-1*, $\Omega(f_t)$ — регуляризація для запобігання перенавчанню.

XGBoost використовує залишки для обчислення ваг для кожного прикладу навчальної вибірки. Ваги вказують на важливість кожного прикладу при навчанні нового дерева.

Нові прогнози додаються до попередніх з врахуванням вагового коефіцієнта (learning rate). Це означає, що кожне нове дерево вносить свій внесок у остаточне передбачення, але зменшений на коефіцієнт навчання, що дозволяє контролювати ступінь внеску кожного нового дерева.

Процес повторюється багаторазово, додаючи нові дерева до моделі, поки не досягнуто заданої кількості дерев або інші критерії зупинки (наприклад, незначне покращення похибки на валідаційній вибірці).

Регуляризація є ключовим компонентом в XGBoost, що допомагає запобігати перенавчанню та покращувати загальну продуктивність моделі. XGBoost використовує кілька методів регуляризації, включаючи L1 та L2 регуляризації.

L1 регуляризація (або Lasso) додає до функції втрат суму абсолютних значень коефіцієнтів. Це призводить до зменшення кількості неважливих ознак шляхом їх обнулення, що спрощує модель та робить її більш інтерпретованою. Формула L1 регуляризації виглядає як [21]:

$$\lambda \sum_{j=1}^{p} \left| \beta_j \right| \tag{3.6}$$

де λ є коефіцієнтом регуляризації, а w_j – це коефіцієнти моделі.

L2 регуляризація (або Ridge) додає до функції втрат суму квадратів коефіцієнтів. Це допомагає зменшити вплив кожного окремого коефіцієнта, що зменшує складність моделі та ризик перенавчання. Формула L2 регуляризації виглядає як [21]:

$$\lambda \sum_{j=1}^{p} \left| \beta_j^2 \right| \tag{3.7}$$

XGBoost дозволяє налаштовувати параметри регуляризації за допомогою параметрів alpha та lambda. Alpha є параметром L1 регуляризації, збільшення значення якого посилює регуляризацію. Lambda є параметром L2 регуляризації, збільшення якого також посилює регуляризацію.

Функція втрат у XGBoost з урахуванням регуляризації виглядає наступним чином [8]:

$$L(\theta) = \sum_{i}^{n} l(y_{i,}\hat{y}_i) + \sum_{k=1}^{K} \omega(f_k) \tag{3.8}$$

Тут $\sum_{i}^{n} l(y_{i,}\hat{y}_i)$ є базовою функцією втрат, яка вимірює різницю між передбаченими значеннями $\hat{y}_i$ і справжніми значеннями y_i, а $\sum_{k=1}^{K} \omega(f_k)$ є регуляризаційним членом, що визначає складність моделі і включає як L1, так і L2 регуляризації.

Регуляризаційний член $\omega(f)$ виглядає як [8]:

$$\omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \tag{3.9}$$

де T є кількістю листків у дереві, ω – вага листка, а γ є параметром регуляризації, що контролює кількість листків.

Одним із ключових етапів навчання моделей XGBoost є обчислення залишків. Залишки відіграють важливу роль у процесі бустінгу, оскільки вони визначають, як кожне наступне дерево вносить корективи для зменшення похибки моделі. Залишки (residuals) — це різниця між реальними значеннями цільової змінної та прогнозованими значеннями, отриманими від моделі. Вони показують, наскільки точні прогнози моделі на поточному етапі навчання. Якщо залишки великі, це означає, що модель ще не достатньо добре описує дані, і потрібно додати більше дерев для покращення точності.

На кожному кроці XGBoost обчислює залишки (різницю між передбаченими та реальними значеннями) та використовує їх для навчання наступного дерева [23]:

$$r_i^{(t)} = -\frac{\partial L(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} \tag{3.10}$$

Залишки дозволяють моделі XGBoost поступово зменшувати похибки. Кожне наступне дерево вносить корективи, ґрунтуючись на помилках попередніх дерев, що забезпечує високу точність і стабільність моделі. Цей підхід робить XGBoost ефективним інструментом для роботи з великими і складними наборами даних.

Скорочення залишкової помилки в XGBoost досягається шляхом поступового додавання нових дерев до ансамблю. Кожне нове дерево намагається скорегувати помилки попередніх дерев, що дозволяє моделі покращувати свою точність на кожному етапі навчання. Важливим аспектом цього процесу є використання коефіцієнта навчання (learning rate), що позначається як η (ета).

Додавання нового дерева до ансамблю можна описати наступною формулою [23]:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i) \tag{3.11}$$

Де $\hat{y}_i^{(t)}$ — передбачення для *i*-го зразка на *t*-й ітерації.

$\hat{y}_i^{(t-1)}$ — передбачення для *i*-го зразка на попередній (t-1)-й ітерації.

η — коефіцієнт навчання (learning rate), гіперпараметр, який контролює вагу нових дерев.

$f_t(x_i)$ — нове дерево, яке намагається скорегувати помилки попередніх передбачень для *i*-го зразка.

Коефіцієнт навчання η визначає, наскільки сильно нове дерево коригує попередні передбачення. Високе значення η призведе до того, що нове дерево матиме більший вплив на остаточне передбачення, що може спричинити перенавчання. З іншого боку, низьке значення η зменшує внесок кожного нового дерева, роблячи процес навчання більш поступовим і стійким до перенавчання.

Загалом математичний апарат XGBoost поєднує в собі ефективні алгоритми градієнтного бустингу з регуляризацією для створення швидких і точних моделей. Завдяки цьому XGBoost став одним з найпопулярніших інструментів у сфері машинного навчання для вирішення різноманітних завдань.

3.2 Підготовка даних для моделі XGBoost

Для побудови моделі XGBoost напишемо SQL запит до бази даних компанії для ствоерення таблиці з прибутком від кожного клієнта та факторами, які можуть впилавти на прибуток.

SQL запит виглядає наступним чином:

```
SELECT s_bookings.id, DATE_FORMAT(s_bookings.arrive, '%Y-
%m') AS arrive_month,
       DATEDIFF(s_bookings.depart, s_bookings.arrive) AS
stay_length,
       ROUND(s_bookings.price_month,2) AS price_month,
       s_bookings.house_id, s_bookings.client_type_id,
       YEAR(CURDATE()) - YEAR(birthday) AS age, gender,
us_citizen,
       DATEDIFF(s_bookings.arrive, s_bookings.created) AS
days_to_arrive,
       manager_login,
       s_bookings.total_price AS revenue
FROM s_bookings
    JOIN s_bookings_users on s_bookings_users.booking_id
= s_bookings.id
    JOIN s_users on s_bookings_users.user_id = s_users.id

WHERE s_bookings.status = 3 AND s_bookings.total_price >
0
      AND s_bookings.created > '2020-01-01' AND
s_bookings.client_type_id IN (1,2)
      AND s_bookings.total_price > 800
      AND YEAR(CURDATE()) - YEAR(birthday) IS NOT NULL
      AND gender <> 0 AND us_citizen <> 0
      AND DATEDIFF(s_bookings.arrive, s_bookings.created)
> 0;
```

Рис. 3.1. SQL запит для відбору даних для побудови моделі

Джерело: створено автором на основі [2]

Цей запит вибирає інформацію про бронювання з таблиці s_bookings, а також відповідні дані про користувачів з таблиць s_bookings_users та s_users. Він форматує дату прибуття у вигляді рік-місяць, обчислює тривалість перебування в днях, округлює місячну ціну до двох знаків після коми, отримує ідентифікатори будинку та типу клієнта, обчислює вік клієнта на основі дати народження, враховує стать, громадянство США, кількість днів до прибуття з моменту створення бронювання, логін менеджера, та загальну ціну як дохід. Запит враховує лише ті бронювання, які мають статус 3 (підтвердження оплати), загальну ціну понад 800, створені після 1 січня 2020 року, з клієнтами типу 1 (клієнти, які прийшли через сайт компанії) або 2 (клієнти, які прийшли через сайт «AirBnb»), вік яких відомий, стать відмінну від 0, громадянство США, та кількість днів до прибуття більше 0.

Таблицю було збережено у файл XGBoost_Data.csv, який налічує 7300 записів. Огляд даного файлу наведений у табл. 3.1.

Таблиця 3.1. Огляд файлу XGBoost_Data.csv

id	arrive_month	stay_length	price_month	house_id	client_type_id	age	gender	us_citizen	days_to_arrive	manager_login	revenue
1834	2020-08	31	973.2	307	2	26	2	1	1	Julia	1038
2067	2020-09	71	1117.8	307	2	26	2	1	5	Nika	2682.05
2579	2020-12	31	1293.9	317	2	30	2	1	14	Katrin	1380
2723	2020-12	53	1026.9	307	2	22	2	1	1	maria	1848
2740	2020-12	31	1117.5	307	2	26	2	1	1	Nika	1192
2742	2020-12	31	818.7	307	2	27	2	1	2	Vlada	873
2804	2020-12	31	1015.5	307	2	54	2	1	11	Katrin	1083
2838	2020-12	32	843	307	2	55	1	1	2	nathalia	927
2855	2020-12	31	990	307	2	35	1	1	11	maria	1056
2918	2021-01	31	1293.9	317	2	30	2	1	13	maria	1380

Джерело: розрахунки автора на основі [2]

Наступним кроком цей файл був завантажений у середовище Python зі зміном типу даних на категоріальний у всіх колонках окрім “price_month”, “age” та “revenue” та видаленням колонки “id”. Запустимо код та перевіримо типи даних (див. табл. 3.2).

Таблиця 3.2. Типи даних

Name	Type
arrive_month	category
stay_length	category
price_month	float64
house_id	category
client_type_id	category
age	category
gender	category
us_citizen	category
days_to_arrive	int64
manager_login	category
revenue	float64

Джерело: розрахунки автора на основі [5]

Також напишемо SQL запит для відбору даних для моделі, яка буде прогнозувати прибуток. Потрібні лише дві колонки: дата у форматі рік-місяць та сума прибутку по кожному місяцю:

```
SELECT  DATE_FORMAT(s_bookings.created, '%Y-%m') AS sale_date,
        SUM(s_bookings.total_price) AS revenue
FROM s_bookings
WHERE s_bookings.status = 3 AND s_bookings.total_price > 0
  AND s_bookings.created > '2020-08-01' AND s_bookings.created
< '2024-01-01'
  AND s_bookings.client_type_id IN (1,2)
GROUP BY sale_date
ORDER BY sale_date;
```

Рис. 3.2. SQL запит для відбору даних для побудови моделі прогнозування

Джерело: створено автором на основі [2]

Таблицю було збережено у файл XGBoost_Months.csv, який налічує 41 запис (див. табл. 3.3).

Таблиця 3.3. Прибуток по місяцям

sale_date	revenue
2020-08	$12,008.00
2020-09	$6,065.05
2020-10	$4,280.08
2020-11	$35,665.18
2020-12	$82,571.72
2021-01	$88,846.13
2021-02	$65,820.96
2021-03	$95,433.29
2021-04	$145,255.00
2021-05	$327,526.00
2021-06	$317,453.08
2021-07	$1,830,683.52
2021-08	$2,597,863.20
2021-09	$1,818,353.71
2021-10	$1,350,571.48
2021-11	$1,621,500.51
2021-12	$1,423,478.91
2022-01	$1,482,506.98
2022-02	$1,524,783.60
2022-03	$2,464,995.31
2022-04	$3,287,061.50
2022-05	$2,377,899.16
2022-06	$2,386,790.00
2022-07	$2,768,025.33
2022-08	$3,580,252.42
2022-09	$1,885,227.10
2022-10	$1,599,808.00
2022-11	$1,457,573.56
2022-12	$1,340,870.34
2023-01	$2,484,517.00
2023-02	$1,522,194.00
2023-03	$2,314,011.24
2023-04	$2,075,919.54
2023-05	$2,757,738.53
2023-06	$2,152,951.42
2023-07	$2,636,540.13
2023-08	$2,747,768.31
2023-09	$1,172,701.35
2023-10	$1,105,773.60

Продовження табл. 3.3.

2023-11	$1,494,920.17
2023-12	$1,080,861.16

Джерело розрахунки автора на основі [2]

У середовище Python імпортуємо ствоерну таблицю та підготуємо її для побудови моделі. Колонка з датами продажу використовується для встановлення індексу, а самі значення індексу перетворюються у формат дат.

Наступним кроком є додавання часових ознак. Функція create_features створює нові ознаки у DataFrame на основі індексу, що містить часові ряди. Спочатку створюється копія вхідного DataFrame. Потім додаються три нові колонки:

- Перша нова колонка відображає місяці, які відповідають індексу.
- Друга нова колонка відображає квартали, які відповідають індексу.
- Третя нова колонка відображає роки, які відповідають індексу.

Після додавання нових колонок, функція повертає оновлений DataFrame (див. табл. 3.4). Ця функція застосовується до існуючого DataFrame, і результат зберігається в тій самій змінній.

Таблиця 3.4. Часові ознаки для побудови моделі

sale_date	revenue	month	quarter	year
8/1/2020	12008	8	3	2020
9/1/2020	6065.05	9	3	2020
10/1/2020	4280.08	10	4	2020
11/1/2020	35665.18	11	4	2020
12/1/2020	82571.72	12	4	2020
1/1/2021	88846.13	1	1	2021
2/1/2021	65820.96	2	1	2021
3/1/2021	95433.29	3	1	2021
4/1/2021	145255	4	2	2021
5/1/2021	327526	5	2	2021

Джерело: розрахунки автора на основі [5]

Далі дані розбиваються на тестову на тренувальну вибірки:

```
train, test = train_test_split(df, test_size=0.2)
train = create_features(train)
test = create_features(test)

FEATURES = ['month', 'quarter', 'year']
TARGET = 'revenue'

X_train = train[FEATURES]
y_train = train[TARGET]

X_test = test[FEATURES]
y_test = test[TARGET]
```

Рис. 3.3. Розбиття вибірки

Джерело: створено автором на основі [5]

Спочатку дані поділяються на навчальну і тестову вибірки, де тестова вибірка становить 20% від загального обсягу даних. Потім для обох вибірок створюються додаткові ознаки за допомогою функції create_features.

Після цього виділяються ознаки для навчання, зокрема місяць, квартал і рік, і цільова змінна, яка в цьому випадку - дохід. Ці ознаки виділяються окремо для навчальної та тестової вибірок, щоб підготувати дані для моделювання.

3.3 Побудова моделі XGBoost

Для побудови моделі були визначені X та y з табл. 3.1, де y – це прибуток, а X – всі фактори, які можуть на нього впливати. Спочатку з DataFrame під назвою df видаляє колонку 'revenue', залишаючи всі інші колонки, і зберігає ці дані у змінну X. Потім з того ж DataFrame витягує колонку 'revenue' і зберігає її у змінну y. Після цього виконується поділ даних на тренувальний та тестовий набори, де 80% даних використовуються для навчання моделі, а 20% даних - для тестування. Ці частини зберігаються у змінних X_train, X_test, y_train та y_test. Розподіл даних здійснюється випадковим чином, але з фіксованим початковим значенням для генератора випадкових чисел, щоб забезпечити однаковий результат при кожному запуску.

Побудуємо модель за допомогою наступного коду:

```
model = XGBRegressor(n_estimators=1000, max_depth=7,
                     eta=0.1, subsample=0.7,
                     colsample_bytree=0.8,
                     enable_categorical=True)
model.fit(X_train, y_train)
```

Рис. 3.4. Побудова моделі XGBoost

Джерело: створено автором на основі [8]

Спочатку створюється модель градієнтного бустінгу для задачі регресії за допомогою класу XGBRegressor з бібліотеки XGBoost. Параметри моделі включають кількість дерев (n_estimators=1000), максимальну глибину кожного дерева (max_depth=7), швидкість навчання (eta=0.1), частку випадкових зразків для кожного дерева (subsample=0.7), частку випадкових ознак для кожного дерева (colsample_bytree=0.8) та увімкнення підтримки категоріальних змінних (enable_categorical=True).

Після визначення моделі, вона навчається на тренувальних даних, де X_train містить ознаки, а y_train - цільові значення.

Тепер оцінемо якість моделі за допомогою середнього абсолютного відхилення.

Об'єкт для крос-валідації створюється для виконання 10 поділів даних з повторенням цього процесу 3 рази, щоб забезпечити надійність оцінок. Випадковий стан встановлюється для відтворюваності результатів.

Крос-валідація моделі виконується на навчальних даних з використанням від'ємної середньої абсолютної помилки як метрики.

Після обчислення бали перетворюються на абсолютні значення, а потім обчислюється і виводиться середнє значення середніх абсолютних помилок.

Отримані значення похибок конвертуються до абсолютних значень, щоб отримати позитивні значення середньої абсолютної похибки (MAE).

Середня абсолютна похибка становить: 1906.17, що є достатньо низьким значенням в контексті даної моделі і свідчить про високу точність.

Для кращого розуміння точності моделі визначимо її коефіцієнт детермінації R^2.

Модель, яка була навчена на тренувальних даних, використовується для прогнозування цільових значень на тестовому наборі даних X_test. Прогнозовані значення зберігаються у змінній y_pred.

Далі обчислюється коефіцієнт детермінації R^2 між фактичними значеннями y_test та прогнозованими значеннями y_pred за допомогою функції r2_score. Коефіцієнт R^2 показує, яку частку варіації у цільовій змінній пояснює модель. Значення варіюються від 0 до 1, де 1 означає ідеальну відповідність, а 0 означає, що модель не пояснює варіацію в даних.

Отримане значення коефіцієнта R^2 округлюється до двох десяткових знаків і множиться на 100 для перетворення в проценти. Результат виводиться як точність моделі.

Отже точність створеної моделі: 94.0%, модель пояснює 94% варіації в цільовій змінній, що є ознакою дуже високої точності моделі.

Наступним кроком був побудований графік (див. рис. 3.5) для визначення важливості ознак моделі.

Спочатку отримуються важливості ознак з бустера моделі за допомогою методу get_score(), який повертає їх у вигляді словника. Ключі та значення з цього словника зберігаються окремо.

Потім створюється датафрейм з важливостями ознак, який сортується за значеннями важливості у порядку зростання. Далі цей датафрейм використовується для побудови горизонтальної гістограми.

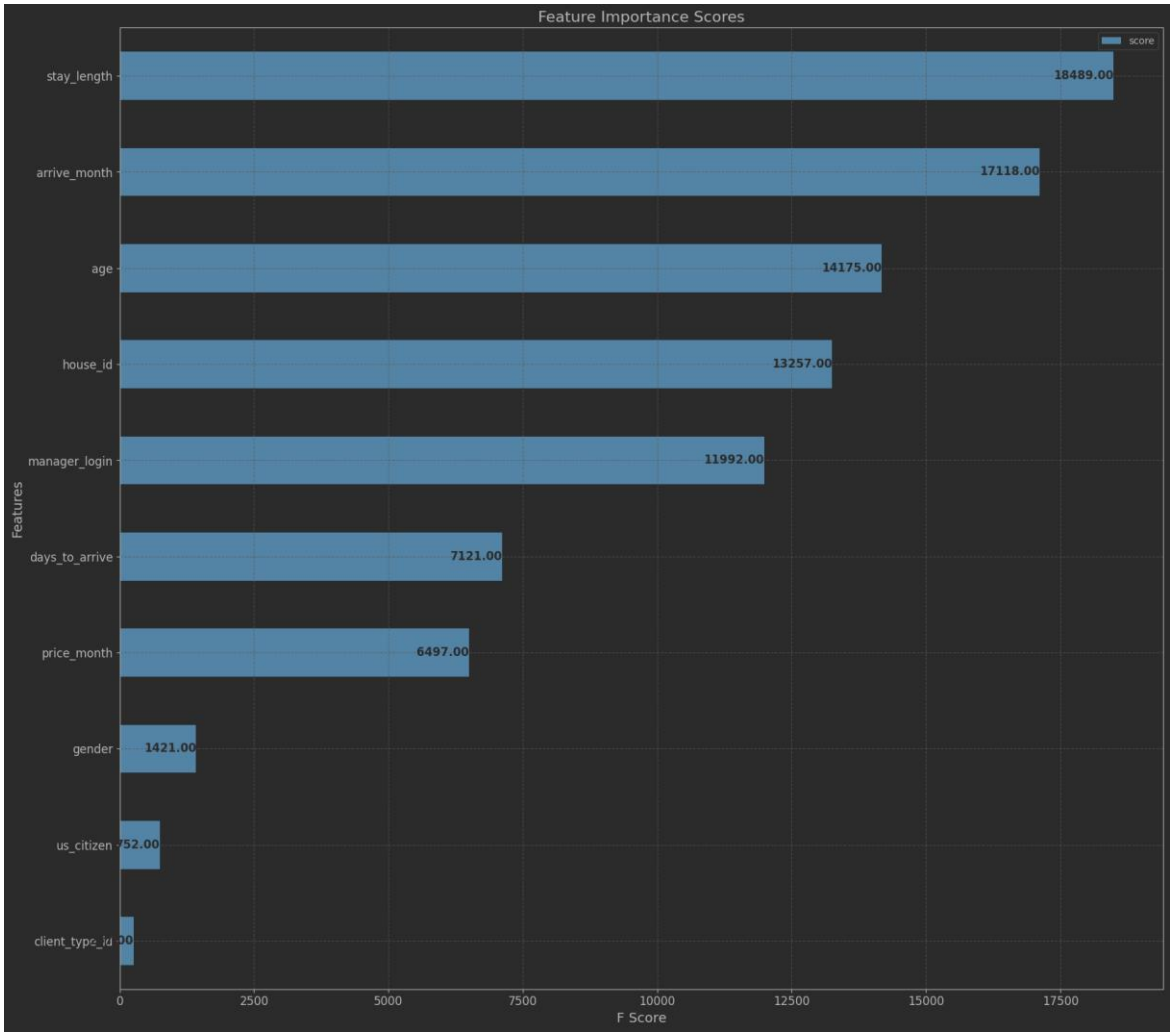

Рис. 3.5. Діаграма ранжування ознак моделі

Джерело: побудовано автором на основі [6]

Графік демонструє важливість різних ознак для прогнозування прибутку за допомогою моделі XGBoost, розташованих у порядку спадання їхнього впливу. Найвищий F-Score має тривалість перебування (stay_length), що робить її найбільш вагомою ознакою, яка впливає на прибуток. Це свідчить про те, що довші періоди перебування клієнтів суттєво збільшують прибуток.

Другою за важливістю ознакою є місяць прибуття (arrive_month), що також має високий F-Score. Це може бути пов’язано з сезонними коливаннями попиту та цін, які впливають на прибуток. Вік клієнтів (age) також є значущим фактором, вказуючи на те, що різні вікові групи мають різні фінансові можливості та поведінкові характеристики, що впливають на прибуток.

Ідентифікатор будинку (house_id) також має вагомий вплив, що може свідчити про різні рівні популярності, цін та якості обслуговування у різних будинках. Вхід менеджера (manager_login) відображає ефективність роботи менеджерів, що також впливає на прибутковість.

Кількість днів до прибуття (days_to_arrive) важлива для моделі, вказуючи на те, наскільки заздалегідь клієнти бронюють послуги, що дозволяє оптимізувати ціни та підвищити прибуток. Ціна за місяць (price_month) також впливає на модель, можливо через зміни в попиті та пропозиції.

Стать (gender) має певний вплив на прибуток, відображаючи поведінкові патерни клієнтів залежно від статі. Громадянство США (us_citizen) має менший вплив, можливо, вказуючи на відмінності у витратах між громадянами США та іноземцями. Ідентифікатор типу клієнта (client_type_id) має найменший вплив на прибуток, що може свідчити про його менш значущу роль у порівнянні з іншими ознаками.

Тепер побудуємо модель на основі даних з файлу XGBoost_Months.csv, підготованим у підрозділі 3.2, для прогнозування прибутку на 2024 рік:

```
reg = xgb.XGBRegressor(base_score=0.5, booster='gbtree',
                       n_estimators=10000,
                       early_stopping_rounds=50,
                       objective='reg:linear',
                       max_depth=5,
                       learning_rate=0.01)
reg.fit(X_train, y_train,
        eval_set=[(X_train, y_train), (X_test, y_test)],
        verbose=100)
```

Рис. 3.6 Модель для прогнозування

Джерело: створено автором на основі [8]

Цей код налаштовує та тренує модель регресії за допомогою XGBoost. Спочатку створюється модель XGBRegressor з певними параметрами: базова оцінка для передбачень встановлена на 0.5, використовується бустер на основі дерев рішень, максимальна кількість дерев обмежена 10 тисячами, рання зупинка навчання відбудеться після 50 ітерацій без покращення, мета моделі - лінійна регресія, максимальна глибина дерев встановлена на 5, а швидкість навчання на 0.01.

Після цього модель навчається на тренувальних даних (X_train, y_train). Під час навчання також використовуються тестові дані (X_test, y_test) для оцінки прогресу та забезпечення ранньої зупинки у випадку відсутності покращень протягом 50 ітерацій. В процесі навчання інформація про прогрес виводиться кожні 100 ітерацій.

Тепер оцінимо точність моделі за допомогою RMSE та R^2 .

Спочатку модель робить передбачення для тестових даних, і ці передбачення додаються до тестового датафрейму. Потім ці результати об'єднуються з основним датафреймом за індексами.

Після цього обчислюється корінь середньоквадратичної помилки (RMSE) між фактичними значеннями доходу та передбаченнями на тестовому наборі даних.

На завершення розраховується коефіцієнт детермінації (R^2) для тестових даних, який показує, наскільки добре модель пояснює варіацію в даних.

Результати тренування моделі:

- RMSE = 194295.53
- R^2 = 94.0%

Результати оцінки моделі свідчать про дуже високу точність, отже дану модель можна використовувати для прогнозування майбутніх прибутків.

Для прогнозування створимо таку ж саму таблицю як табл. 3.4. тільки на весь 2024 рік.

Згенеруємо послідовність дат для кожного місяця 2024 року у форматі 'YYYY-MM', створимо DataFrame з цими датами як індексом та перетворимо індекс на тип datetime для роботи з датами у Pandas, а потім додамо нові ознаки за допомогою функції create_features (див. табл. 3.5).

Таблиця 3.5. Часові ознаки для 2024 року

Date	month	quarter	year
1/1/2024	1	1	2024
2/1/2024	2	1	2024
3/1/2024	3	1	2024
4/1/2024	4	2	2024
5/1/2024	5	2	2024
6/1/2024	6	2	2024
7/1/2024	7	3	2024
8/1/2024	8	3	2024
9/1/2024	9	3	2024
10/1/2024	10	4	2024
11/1/2024	11	4	2024
12/1/2024	12	4	2024

Джерело: розрахунки автора на основі [5]

За допомогою вже натренованої моделі, спрогнозуємо прибуток на кожен місяць 2024 року, та виведемо результати у табл. 3.6.

Таблиця 3.6. Прогнозований прибуток

month	prediction
1	$1,914,142.00
2	$1,697,471.50
3	$2,177,188.00
4	$2,257,800.25
5	$2,440,427.00
6	$2,235,541.25
7	$2,412,015.50
8	$2,549,474.00
9	$1,223,428.38
10	$1,103,776.25
11	$1,258,189.88
12	$1,258,189.88
Всього	**$22,527,643.88**

Джерело: розрахунки автора на основі [5]

Отже з вірогідністю у 94% компанія заробить $22,527,643.88 у 2024 році.

Створимо також графік для прівняння прибутків компанії з 2021 по 2024 роки (див. рис. 3.7).

Рис. 2.7. Прибутки компанії за 2021-2024 роки

Джерело: побудовано автором на основі [10]

На графіку видно, що існує певна спадна тенденція у 2023-2024 роках. Це може бути спричинене кількома факторами. Зростання вартості утримання нерухомості, таких як витрати на обслуговування, ремонт і податки, може негативно впливати на фінансові результати. Також може знижуватися попит на оренду житла через економічну кризу, демографічні зміни або переваги орендарів на користь інших місць.

Крім того, зростаюча конкуренція з боку інших компаній та альтернативних варіантів житла, таких як “Airbnb”, може призводити до зниження цін і скорочення доходів. Регуляторні зміни, такі як нові закони та правила щодо оренди, також можуть збільшувати витрати або знижувати доходи. Загальний економічний стан, включаючи економічний спад, збільшення безробіття та зниження доходів населення, також може впливати на здатність людей платити високу орендну плату.

Зміна уподобань орендарів може призвести до зниження попиту на пропоновані компанією кімнати, якщо орендарі віддають перевагу іншим типам житла або іншим районам міста. Пандемія COVID-19 також могла вплинути на попит на оренду через зміну робочих умов та міграцію з великих міст до менш населених місць. Зважаючи на ці фактори, компанія можуть повинна переглянути цінову політику та процесс продажів а такж розробити стратегії для мінімізації ризиків і підтримки стабільного фінансового стану.

ВИСНОВКИ

У даній роботі було проведено всебічне дослідження з метою моделювання та аналізу даних користувачів веб-сайту компанії «Outpost Club» для підвищення ефективності бізнесу. Використання сучасних методів аналізу даних та моделювання дозволило досягти поставлених завдань, що включали аналіз поведінки користувачів, кластеризацію користувачів, а також прогнозування прибутку на наступний рік.

У першому розділі було здійснено ознайомлення зі середовищем HubSpot та базою даних компанії. Зібрані дані були оброблені та візуалізовані за допомогою Tableau, що дозволило виявити основні тенденції та закономірності у поведінці користувачів. Виявлено, що більшість користувачів звертаються до компанії через офлайн-ресурси та мають схожі демографічні характеристики. Такий розвідувальний аналіз даних є важливим кроком у підготовці до подальшого моделювання та кластеризації.

Другий розділ присвячено кластеризації користувачів за допомогою моделі K-means. Було детально розглянуто математичний апарат моделі та виконано підготовку даних для кластеризації. В результаті кластеризації було отримано чотири основні кластери користувачів, кожен з яких мав свої унікальні характеристики. Наприклад, один з кластерів складався переважно з молодих жінок, що використовують офлайн-ресурси для звернення до компанії. Отримані результати дозволяють більш точно таргетувати різні сегменти аудиторії та оптимізувати маркетингові стратегії компанії.

У третьому розділі було розглянуто прогнозування прибутку за допомогою моделі XGBoost. Описано математичний апарат моделі та проведено підготовку даних для побудови моделі.

Особливу увагу було приділено дослідженню ознак, які найбільше впливають на прогнозування прибутку. Аналіз показав, що такі фактори, як тривалість перебування клієнта (stay_length), тип клієнта (client_type_id), та кількість днів до прибуття з моменту створення бронювання (days_to_arrive) мають значний вплив на прибуток компанії. Зокрема, було виявлено, що клієнти, які здійснюють бронювання заздалегідь, а також клієнти, які залишаються на довший період, приносять більше прибутку.

З використанням історичних даних про прибуток було виконано прогнозування на наступний рік. Модель XGBoost показала високу точність прогнозування, що дозволяє компанії планувати свої фінансові ресурси та маркетингові активності більш ефективно.

Ефективність аналізу даних підтверджується використанням сучасних інструментів для аналізу та візуалізації даних, таких як HubSpot, Tableau, SQL та Python. Вони дозволяють отримати глибокі інсайти щодо поведінки користувачів та ефективності маркетингових стратегій.

Кластеризація користувачів за допомогою моделі K-means є ефективним інструментом для сегментації, що дозволяє більш точно орієнтувати маркетингові кампанії та підвищувати їх ефективність.

Прогнозування прибутку за допомогою моделі XGBoost демонструє високу точність у прогнозуванні фінансових показників, що є ключовим фактором для стратегічного планування та оптимізації бізнес-процесів.

Загалом, проведене дослідження демонструє, що застосування сучасних методів аналізу та моделювання даних є важливим інструментом для підвищення ефективності бізнесу та оптимізації маркетингових стратегій. Робота підтверджує, що глибокий аналіз користувацьких даних дозволяє краще розуміти потреби клієнтів та приймати більш обґрунтовані бізнес-рішення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Довідник з мови Python. URL: https://docs.python.org/uk/3/reference/index.html
2. Довідник з мови SQL. URL: https://w3schoolsua.github.io/sql/sql_ref_keywords.html#gsc.tab=0
3. Довідник з Tableu. URL: https://help.tableau.com/current/guides/get-started-tutorial/en-us/get-started-tutorial-next.htm
4. Документація з користування HubSpot. URL: https://knowledge.hubspot.com/
5. Документація з бібліотеки Pandas. URL: https://pandas.pydata.org/docs/
6. Документація з бібліотеки Matplotlib. URL: https://matplotlib.org/stable/users/index
7. Документація з бібліотеки scikit-learn. URL: https://scikit-learn.org/stable/
8. Документація з бібліотеки xgboost. URL: https://xgboost.readthedocs.io/en/stable/index.html
9. Документація з бібліотеки NumPy. URL: https://numpy.org/
10. Довідник по Excel. URL: https://support.microsoft.com/en-us
11. Офіційний сайт викладача кафедри економічної кібернетики – Ставицького А.В. URL: http://www.andriystav.cc.ua/.
12. Курс з дата аналітики URL: https://www.coursera.org/professional-certificates/google-data-analytics
13. Черняк О.І., Комашко О.В., Ставицький А.В., Баженова О.В. Економетрика: підручник. За ред. О.І.Черняка. –К.: Видавчничо-поліграфічний центр «Київський університет» , 2010. – 359 с.
14. Черняк О.І., Ставицький А.В., Чорноус Г.О. Системи обробки економічної інформації К: Знання, 2006. – 447 с.
15. Єріна А.М., Пальян З.О. Статистика: Підручник /А.М.Єріна, З.О.Пальян. – К.: КНЕУ, 2010. – 351с.
16. Блог «The Science of Machine Learning & AI». URL: https://www.ml-science.com/k-means-clustering

17. Офіційний сайт Девіда Зиганто. URL: https://dziganto.github.io/data%20science/linear%20regression/machine%20learning/python/Linear-Regression-101-Metrics/
18. Офіційний сайт «SUBOPTIMaL». URL: https://suboptimal.wiki/explanation/mse/
19. Стаття «MAPE vs MAE: Which Metric is Better?» на офіційному сайті «Medium». URL: https://medium.com/trusted-data-science-haleon/mape-vs-mae-which-metric-is-better-68dd559cbfb1
20. Платформа «Quora». URL: https://www.quora.com/What-is-log-loss-in-Kaggle-competitions
21. Офіційний сайт «builtin». URL: https://builtin.com/data-science/l2-regularization
22. Стаття «XGBoost Mathematics Explained» на офіційному сайті «Medium» URL: https://dimleve.medium.com/xgboost-mathematics-explained-58262530904a
23. Тенці Чжен, Карлос Гестрінг, 2016. – 13 с. «XGBoost: A Scalable Tree Boosting System». URL: https://arxiv.org/pdf/1603.02754
24. Офіційний сайт Філіпа Гетсмана. URL: https://philippeheitzmann.com/2022/01/implementing-kmeans-clustering/
25. Публікація «K-Means Clustering Approach for Intelligent Customer Segmentation Using Customer Purchase Behavior Data» на офіційному сайті «ResearchGate». URL: https://www.researchgate.net/publication/361268468_K-Means_Clustering_Approach_for_Intelligent_Customer_Segmentation_Using_Customer_Purchase_Behavior_Data
26. Публікація «Augmented Data and XGBoost Improvement for Sales Forecasting in the Large-Scale Retail Sector» на офіційному сайті «MDPI». URL: https://www.mdpi.com/2076-3417/11/17/7793

27. Стаття «How to Use XGBoost for Time-Series Forecasting?» на офіційному сайті «AnalyticsVidhya». URL: https://www.analyticsvidhya.com/blog/2024/01/xgboost-for-time-series-forecasting/

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

студента 4 курсу спеціальності 051 «Економіка», ОПП «Економічна кібернетика»

Руденка Андрія Андрійовича

1. Тема роботи: «Моделювання та аналіз даних користувачів веб-сайту на прикладі компанії «Outpost Club»»
2. Термін завершення роботи: 02.06.2024
3. Попередній захист роботи: 03.06.2024
4. Об’єкт дослідження: поведінка користувачів веб-сайту компанії «Outpost Club».
5. Предмет дослідження: методи та алгоритми моделювання і аналізу даних, для розуміння поведінки користувачів та оптимізації взаємодії з ними.
6. Мета дослідження: розробка та застосування комплексного підходу до моделювання та аналізу даних користувачів веб-сайту компанії «Outpost Club» для оптимізації їх взаємодії з сайтом, підвищення конверсії та, як наслідок, збільшення ефективності бізнесу за допомогою інструментів дата аналітики.
7. Завдання дослідження:
   7.1. Аналіз поведінки користувачів.
   7.2. Кластеризація користувачів.
   7.3. Дослідження факторів впилву на прибуток.
   7.3. Прогнозування прибутку на наступний рік.

Науковий керівник: кандидат економічних наук, доцент, Шпирко Віктор Васильович. Підпис: _____

Студент: _____

(підпис)

Затверджено на засіданні кафедри економічної кібернетики

протокол № 4 від 22.11.2023 р.

Календарний план виконання кваліфікаційної роботи бакалавра

№	Етапи роботи	Терміни виконання	Відмітка керівника про виконання
1	Вибір теми кваліфікаційної роботи бакалавра	15.10.2023	
2	Розробка та затвердження завдання кваліфікаційної роботи бакалавра	10.11.2023	
3	Збір інформації, її аналіз, обробка, консультації з науковим керівником	10.11.2023-18.03.2024	
4	Підготовка роботи відповідно до вимог оформлення	18.03.2024-03.06.2024	
5	Подання роботи до попереднього захисту	03.06.2024	
6	Перевірка на плагіат	06.05.2024	
7	Отримання відгуку наукового керівника		
8	Отримання рецензії на кваліфікаційну роботу бакалавра		
9	Подача остаточного варіанту роботи		
10	Захист роботи на засіданні ЕК		

Науковий керівник: Шпирко Віктор Васильович

Студент: Руденко Андрій Андрійович