

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

**ДОСЛІДЖЕННЯ МЕТОДІВ ОБРОБЛЕННЯ ДАНИХ**  
**У CRM-СИСТЕМІ ОРГАНІЗАЦІЇ БІЗНЕСУ**

(тема)

Виконав:  
студент 2 курсу, групи ІНФМ-22-2

Явтушенко Д.М.

(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва освітньої програми)

Керівник доц. Творошенко І.С.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

\_\_\_\_\_

Кобилін О.А.  
(прізвище, ініціали)

2024 р.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)Кафедра Інформатики  
(повна назва)Рівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Явтушенку Дмитру Миколайовичу  
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження методів оброблення даних у CRM-системі організації бізнесу

затверджена наказом по університету від 3 листопада 2023 року № 1280Ст

2. Термін подання студентом роботи до екзаменаційної комісії 23 грудня 2023 р.3. Вихідні дані до роботи математичні моделі оброблення даних, перелік використаних програмних засобів: MS SQL, Python 3.9, T-SQL, NumPy, sklearn, SQLAlchemy, Python unit testing. Теоретичні відомості про методи оброблення даних у CRM-системі організації бізнесу.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Огляд методів оброблення даних у CRM-системі організації бізнесу.2. Математичні моделі методів K-середніх, A/B тестування та порівняння з галузевими стандартами.3. Програмна реалізація методів.4. Тестування ефективності методів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми, об'єкт та мета дослідження, постановка задачі, аналіз предметної області, вихідні дані для дослідження, етапи реалізації поставленої задачі, аналіз результатів.

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

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

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	03.11.2023	
2	Аналіз завдання, підбір літератури	03.11.23-06.11.23	
3	Аналіз літератури з досліджуваної проблеми	06.11.23-09.11.23	
4	Аналіз методів	09.11.23-13.11.23	
5	Розробка програмних реалізацій методів	13.11.23-20.11.23	
6	Програмна реалізація	20.11.23-30.11.23	
7	Оформлення пояснювальної записки	30.11.23-10.12.23	
8	Перевірка на плагіат	15.12.2023	
9	Рецензування	22.12.2023	
10	Підготовка презентації та доповіді	23.12.2023	
11	Занесення роботи в електронний архів	01.01.2024	
12	Попередній захист кваліфікаційної роботи	02.01.2024	

Дата видачі завдання 3 листопада 2023 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

\_\_\_\_\_ доц. Творошенко І.С.  
(посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 59 с., 25 рис., 50 джерел.

CRM-СИСТЕМА, ОБРОБЛЕННЯ ДАНИХ, СЕГМЕНТАЦІЯ КЛІЄНТІВ, МАРКЕТИНГОВІ СТРАТЕГІЇ, МАШИННЕ НАВЧАННЯ, АНАЛІЗ ДАНИХ, ПЕРСОНАЛІЗАЦІЯ, ЕФЕКТИВНІСТЬ БІЗНЕСУ, ОПТИМІЗАЦІЯ ВЗАЄМОДІЇ, ОРГАНІЗАЦІЯ БІЗНЕСУ.

Об'єктом дослідження є процес оброблення даних у CRM-системі організації бізнесу.

Метою дослідження є порівняння методів оброблення даних у CRM-системі організації бізнесу.

Головне завдання – порівняти методи оброблення даних у CRM-системі організації бізнесу, щоб підвищити рівень розуміння та взаємодії з клієнтами, оптимізувати маркетингові стратегії та підвищити загальну продуктивність.

Здійснено дослідження класичних методів оброблення даних. Вивчено можливості методу кластерного аналізу *K*-середніх, особливості роботи з ним. Досліджено вимірювання показників, таких як SSE та інших метрик для валідації моделі. У результаті роботи вдалося успішно сегментувати клієнтів на основі їхніх характеристик.

CRM SYSTEM, DATA PROCESSING, CUSTOMER SEGMENTATION, MARKETING STRATEGIES, MACHINE LEARNING, DATA ANALYSIS, PERSONALIZATION, BUSINESS EFFICIENCY, INTERACTION OPTIMIZATION, BUSINESS ORGANIZATION.

The object of research is data processing methods in a business organization's CRM system.

The purpose of the research is to compare the data processing methods in a business organization's CRM system.

The main task is to compare the methods of data processing in the CRM system of a business organization to increase the level of understanding and interaction with customers, optimize marketing strategies, and increase overall productivity.

Research on classical data processing methods was conducted. The capabilities of the *K*-means cluster analysis method and the peculiarities of working with it are studied. The measurement of indicators, such as SSE and other metrics for model validation, was investigated. As a result, we could segment customers based on their characteristics successfully.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	6
Вступ.....	7
1 Огляд основних методів оброблення даних у CRM-системі організації бізнесу.....	9
1.1 Метод порівняння метрик з галузевими стандартами .....	18
1.2 Метод А/В-тестування.....	19
1.3 Метод <i>K</i> -середніх.....	21
1.4 Постановка задачі дослідження.....	23
2 Математичні моделі аналізу оброблення даних .....	25
2.1 Модель А/В-тестування в CRM-системах.....	25
2.1.1 Коефіцієнт конверсії.....	25
2.1.2 Оцінка статичної значущості.....	26
2.1.3 Багатовимірне тестування (MVT) .....	27
2.2 Модель методу <i>K</i> -середніх.....	28
3 Комп'ютерна модель фільтрації зображень .....	34
3.1 Обґрунтування вибору середовища програмної реалізації .....	34
3.2 Програмна реалізація.....	37
3.3 Інструкція користувача .....	46
3.4 Тестування розробленої моделі.....	48
Висновки .....	53
Перелік джерел посилання .....	55

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

DWH – Data Warehouse (сховище даних)

ETL – Extract, Transform, Load (витягти, перетворити, завантажити)

CRM – Customer Relationship Management (управління відносинами з клієнтами)

SSE – Sum of the Squared Euclidean distances (сума квадратів евклідових відстаней)

SQL – Structured Query Language (структурована мова запитів)

NoSQL – Not Only SQL (не тільки SQL)

T-SQL – Transact SQL

MVT – Multivariate Testing (багатовимірне тестування)

WCSS – Within-Cluster Sum of Squares (внутрішньокластерна сума квадратів)

SSMS – SQL Server Management Studio

ERP – Enterprise Resource Planning (планування ресурсів підприємства)

Data Profiling – профілювання даних

## ВСТУП

У сучасному бізнес-середовищі, що швидко розвивається, ефективне управління взаємовідносинами з клієнтами відіграє ключову роль в успіху організації. Системи управління взаємовідносинами з клієнтами або CRM стали незамінними інструментами для бізнесу, які дозволяють не лише зберігати та організовувати дані про клієнтів, але й використовувати цю інформацію для прийняття стратегічних рішень та персоналізованої взаємодії з клієнтами. Постійно зростаючий обсяг і складність даних, що генеруються в цифрову епоху, призвели до зростаючої потреби в складних методах обробки даних в CRM-системах.

Дослідження та аналіз різноманітних методів, що використовуються для обробки даних в CRM-системах в контексті бізнес-організацій, є важливим кроком на шляху до вивчення та аналізу різноманітних методів, що застосовуються для обробки даних в бізнес-організаціях. Ця кваліфікаційна робота має на меті пролити світло на поточний стан методологій обробки даних, оцінити їх ефективність та визначити можливості для вдосконалення в цій критично важливій галузі [1].

Обсяг дослідження охоплює широке вивчення методів, що використовуються для збору, зберігання, пошуку, аналізу та використання даних в CRM-системах. Ці методи є важливими для підвищення рівня залучення клієнтів, оптимізації маркетингових стратегій, оптимізації процесів продажів і, зрештою, для зростання та прибутковості організації.

Значення цього дослідження виходить за рамки академічних розвідок; воно має практичне значення для бізнесу всіх розмірів і галузей. Заглиблюючись у тонкощі обробки даних CRM, ця кваліфікаційна робота має на меті надати інформацію, яка може допомогти організаціям приймати більш обґрунтовані рішення, надавати кращий клієнтський досвід та залишатися конкурентоспроможними на динамічному ринку.

Надалі необхідно заглибитися у всебічний аналіз існуючих методів обробки даних у CRM-системах, викликів, з якими стикається бізнес, та інноваційних підходів, які можуть змінити спосіб управління взаємовідносинами з клієнтами в організаціях.

## 1 ОГЛЯД ОСНОВНИХ МЕТОДІВ ОБРОБЛЕННЯ ДАНИХ У CRM-СИСТЕМІ ОРГАНІЗАЦІЇ БІЗНЕСУ

У сучасному швидкоплинному бізнес-середовищі роль систем управління взаємовідносинами з клієнтами або CRM надзвичайно зросла. Ці системи перетворилися з простих сховищ даних на потужні інструменти для управління та обробки даних з метою підвищення рівня залученості клієнтів і стимулювання зростання бізнесу. Цей аналіз заглиблюється в різні аспекти сучасних методів обробки даних у CRM-системах, щоб зрозуміти їхнє значення, оцінити їхню ефективність та визначити можливості для вдосконалення.

Системи управління взаємовідносинами з клієнтами (CRM) стали невід'ємною частиною організацій у різних галузях (рис. 1.1). Вони є центрами для збору, зберігання та управління інформацією про клієнтів.



Рисунок 1.1 – Бізнес-можливості CRM-систем

Цінність сучасних CRM-систем виходить за рамки зберігання даних; вони оснащені складними можливостями обробки даних, що дозволяє компаніям отримувати практичну інформацію, персоналізувати взаємодію з клієнтами та приймати обґрунтовані рішення [2].

Ефективні методи збору даних необхідні для того, щоб забезпечити наповнення CRM-систем точною та актуальною інформацією про клієнтів. Дані надходять з різних джерел, включаючи вебсайти, соціальні мережі, електронну пошту та безпосередню взаємодію з клієнтами [3].

Розглянемо відомого гіганта електронної комерції Amazon, який використовує CRM-системи для постійного збору та аналізу даних про поведінку клієнтів. Відстежуючи історію переглядів і покупок, вони пристосовують рекомендації щодо продуктів і маркетингові кампанії до індивідуальних уподобань, що призводить до збільшення залучення клієнтів і продажів.

Ефективне зберігання та організація даних в CRM-системах є життєво важливими. Добре структурована база даних полегшує швидкий і надійний пошук даних, гарантуючи, що інформація про клієнтів завжди доступна для персоналізованого та ефективного обслуговування.

Здатність швидко і легко отримувати дані має вирішальне значення для повсякденної роботи. Зручний інтерфейс та ефективні механізми пошуку мають важливе значення для оптимізації процесів.

Справжня сила CRM-систем полягає в аналізі даних. Такі методи, як описова, предиктивна та прескриптивна аналітика, дозволяють компаніям отримувати інформацію з даних про клієнтів. Описова аналітика допомагає зрозуміти минулі тенденції, предиктивна аналітика прогнозує майбутню поведінку, а рекомендаційна аналітика пропонує дії для покращення взаємодії з клієнтами та бізнес-результатів [4].

Візьмемо, наприклад, міжнародну телекомунікаційну компанію Vodafone, яка використовує предиктивну аналітику у своїй CRM-системі для виявлення клієнтів, які ризикують відтоком. Активно взаємодіючи з цими клієнтами та вирішуючи їхні проблеми, компанія успішно знижує рівень відтоку клієнтів і підвищує рівень їхнього утримання.

Першочерговим завданням є отримання дієвої інформації з даних CRM. Візуальні представлення, такі як інформаційні панелі та звіти, дають чітке

уявлення про ключові бізнес-показники. Ця інформація сприяє процесам прийняття рішень, допомагає визначити ринкові тенденції, сегментувати клієнтів і адаптувати маркетингові стратегії.

У секторі роздрібної торгівлі провідна мережа використовує CRM-аналітику для аналізу моделей покупок і демографічних даних клієнтів. Цей підхід, заснований на даних, дозволяє їм оптимізувати управління запасами, планувати цільові маркетингові кампанії та створювати персоналізовані акції, що призводить до збільшення продажів і підвищення рівня задоволеності клієнтів.

Оскільки компанії все більше покладаються на CRM-системи, вони стикаються з кількома проблемами. Для захисту інформації про клієнтів необхідно вирішувати питання безпеки та конфіденційності даних. Інтеграція CRM-систем з іншими платформами може бути складною і вимагає ретельного планування. Крім того, постійною проблемою є масштабованість для обробки зростаючих обсягів даних (рис. 1.2).



Рисунок 1.2 – Схема захисту CRM-системи

CRM-системи продовжуватимуть розвиватися. Очікується, що штучний інтелект (ШІ) і машинне навчання (МН) відіграватимуть все більшу роль в автоматизації завдань, наданні прогнозних даних і поліпшенні клієнтського досвіду. Інтеграція CRM-систем з новими технологіями, такими як блокчейн і доповнена реальність, може відкрити нові можливості для залучення клієнтів [5].

Аналіз сучасних методів обробки даних у CRM-системах підкреслює їхню вирішальну роль у сучасному світі, керованому даними. Ці системи є не просто сховищами інформації про клієнтів, а стратегічними інструментами, які дають змогу організаціям краще розуміти своїх клієнтів, приймати рішення на основі даних і, зрештою, процвітати на конкурентному ринку.

Сучасні системи управління взаємовідносинами з клієнтами перетворилися на стратегічні активи для бізнесу. Вони чудово справляються зі збором, зберіганням, пошуком та аналізом даних, сприяючи персоналізованій взаємодії з клієнтами і стимулюючи зростання бізнесу [6].

Виклики, такі як безпека даних і складність інтеграції, залишаються, але CRM-системи продовжують розвиватися, використовуючи штучний інтелект, машинний інтелект і новітні технології для покращення якості обслуговування клієнтів. У світі, де все ґрунтується на даних, CRM-системи є ключовими інструментами, що дають організаціям змогу процвітати та досягати успіху.

У сучасному бізнес-середовищі, що швидко розвивається і орієнтоване на дані, роль CRM зазнала глибокої трансформації.

Колись ці системи розглядалися насамперед як інструменти для зберігання базових даних, а зараз вони перетворилися на динамічні та стратегічні активи для організацій у різних галузях. Вони перетворилися з пасивних сховищ інформації про клієнтів на проактивні та інтелектуальні інструменти, які сприяють реалізації клієнтоорієнтованих стратегій, оптимізують операційну ефективність та сприяють зростанню бізнесу [7].

Збір даних у CRM-системах охоплює різні методи.

Вебскрепінг передбачає автоматизоване вилучення даних з вебсайтів, часто з використанням бібліотек Python, таких як BeautifulSoup і Scrapy (рис. 1.3) [8].

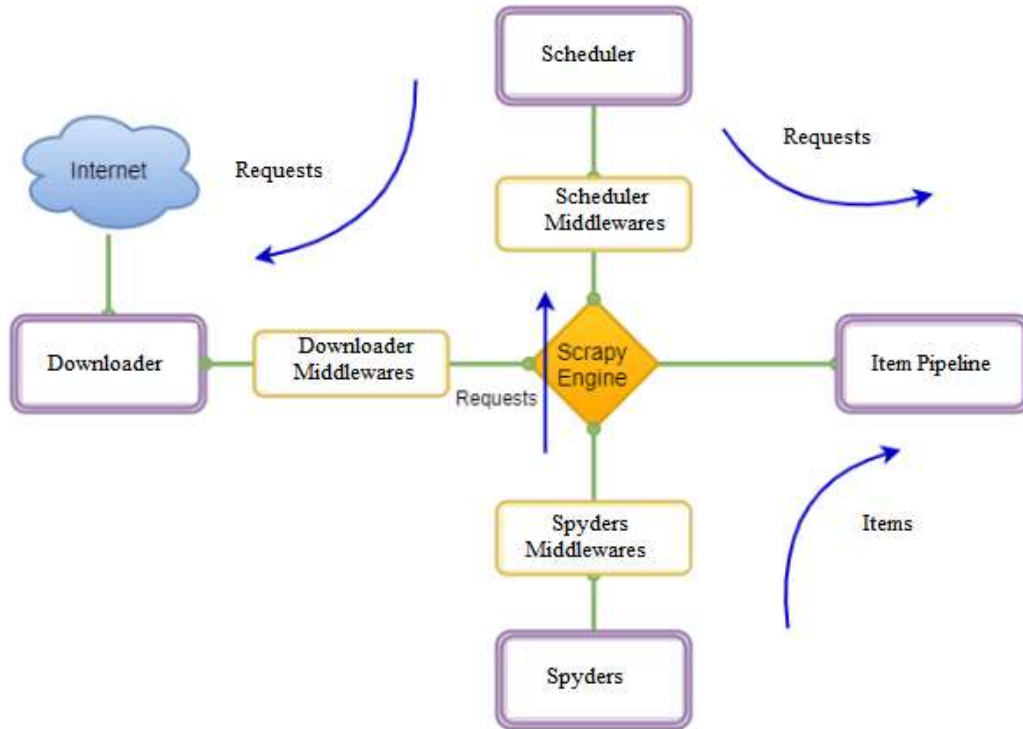


Рисунок 1.3 – Схема роботи Scrapy бібліотеки

Інтеграція API полегшує обмін даними з зовнішніми платформами в режимі реального часу. Форми введення даних дозволяють вводити їх вручну, а сервіси збагачення даних доповнюють записи CRM зовнішніми даними.

Щоб надати комплексне уявлення про екосистему CRM, необхідно використовувати візуальне представлення, яке ретельно класифікує методи обробки даних в CRM-системах [9].

Ці методи охоплюють широкий спектр областей, включаючи збір, зберігання, пошук і аналіз даних. Ця візуальна категоризація пропонує структуровану панораму того, як ці методи легко інтегруються для підтримки багатограних потреб організацій, незалежно від їхньої галузі чи розміру.

CRM-системи покладаються на бази даних для зберігання (рис. 1.4). Реляційні бази даних, такі як MySQL, забезпечують структуроване зберігання, тоді як NoSQL бази даних, такі як MongoDB, обробляють неструктуровані дані. Сховища даних консолідують дані для складної аналітики.

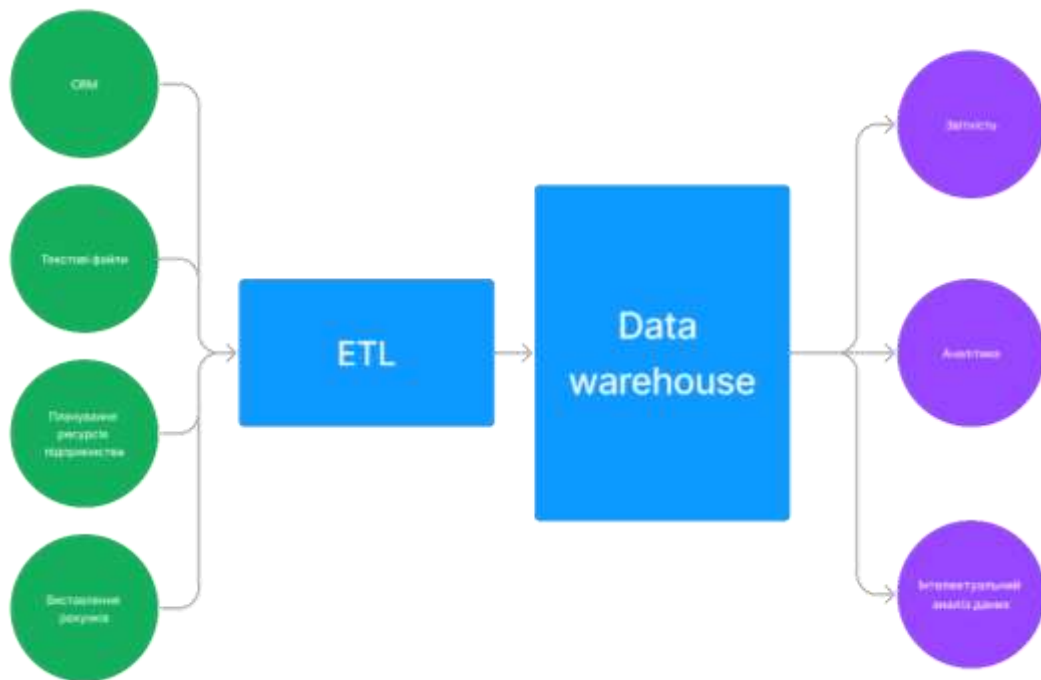


Рисунок 1.4 – Схема поєднання CRM з ETL та DWH

Ефективний збір даних слугує фундаментом, на якому будуються CRM-системи. Для накопичення масиву даних про клієнтів з різних джерел, включаючи вебсайти, соціальні мережі, електронну пошту та особисте спілкування, використовується ціла низка складних методів. Цей процес не просто збір даних; він являє собою безперервні, багатогранні зусилля, спрямовані на те, щоб система CRM залишалася динамічним сховищем точної, актуальної та сучасної інформації про клієнтів [10].

CRM-системи отримують дані за допомогою SQL-запитів, що дозволяє користувачам отримувати конкретні дані на основі певних критеріїв.

Механізми індексування та кешування підвищують продуктивність і доступність запитів (рис. 1.5).

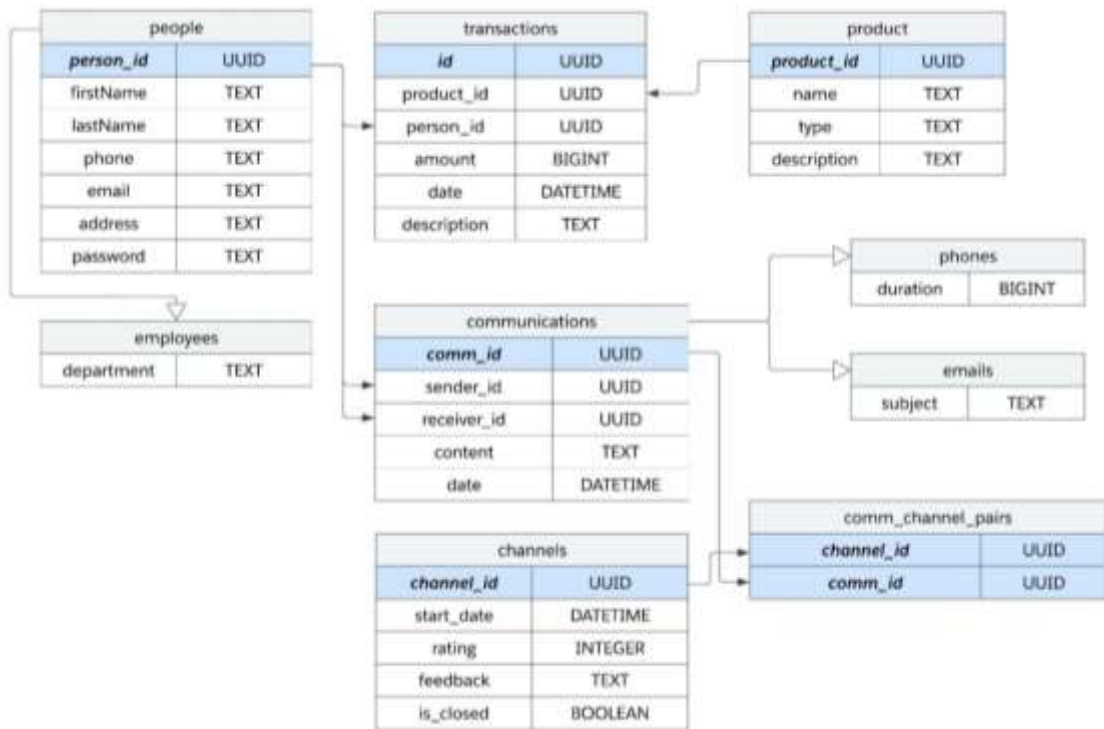


Рисунок 1.5 – Приклад бази даних CRM-системи

Ефективне зберігання та організація даних в CRM-системах мають вирішальне значення для забезпечення безперешкодного пошуку інформації.

Добре структурована і ретельно організована база даних є основою персоналізованого та ефективного обслуговування клієнтів. По суті, зберігання даних виходить за рамки простого збереження; мова йде про систематичну організацію для полегшення оптимального використання, що забезпечує міцний фундамент для прийняття рішень на основі даних [11].

CRM-системи використовують три основні методи аналізу даних. Описова аналітика узагальнює історичні дані, предиктивна аналітика прогнозує майбутню поведінку, а рекомендаційна аналітика рекомендує дії для оптимізації (рис. 1.6) [12].

Можливість швидкого та інтуїтивно зрозумілого пошуку даних є центральним елементом щоденної роботи будь-якої CRM-системи.

Інтуїтивно зрозумілий користувацький інтерфейс у поєднанні з ефективними механізмами пошуку спрощує процеси та забезпечує швидкий доступ до необхідної інформації про клієнтів.

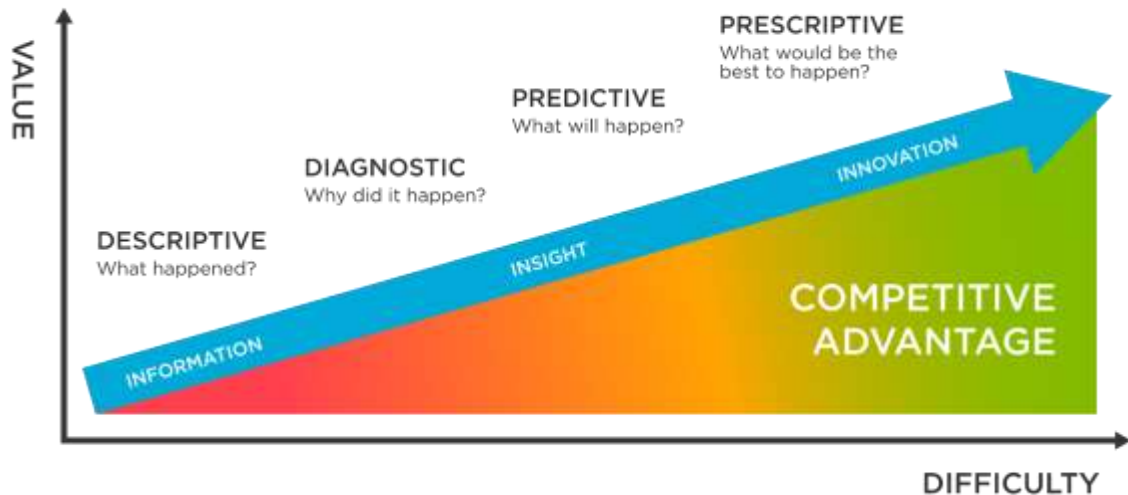


Рисунок 1.6 – Методи аналізу даних

Така доступність допомагає підтримувати персоналізовану взаємодію, приймати обґрунтовані рішення та оперативно реагувати на потреби клієнтів, що сприяє підвищенню операційної ефективності та задоволеності клієнтів [13].

Справжня сила CRM-систем розкривається завдяки їхнім передовим методам аналізу даних. Вони охоплюють широкий спектр методологій, включаючи описову аналітику, предиктивну аналітику та прескриптивну аналітику. Описова аналітика дозволяє організаціям розшифрувати минулі тенденції та поведінку клієнтів, предиктивна аналітика передбачає майбутні дії клієнтів, тоді як рекомендаційна аналітика пропонує практичні висновки, пропонуючи стратегії для підвищення залученості клієнтів та покращення загальних результатів бізнесу [14].

Обробка даних за допомогою CRM знаходить застосування в різних галузях. У роздрібній торгівлі вона персоналізує маркетинг і оптимізує запаси.

Організації охорони здоров'я використовують її для управління записами пацієнтів і планування зустрічей. Фінансові служби використовують її для виявлення шахрайства та оцінки ризиків, а платформи електронної комерції – для покращення якості обслуговування клієнтів [15].

Розглянемо, наприклад, глобальний телекомунікаційний конгломерат Verizon. Використовуючи предиктивну аналітику в CRM-системі, компанія виявляє клієнтів, які ризикують відтоком. Активно взаємодіючи з цими клієнтами і вирішуючи їхні проблеми, компанія не лише знижує рівень відтоку, але й підвищує рівень утримання клієнтів, що ілюструє трансформаційну силу аналізу даних (рис. 1.7).

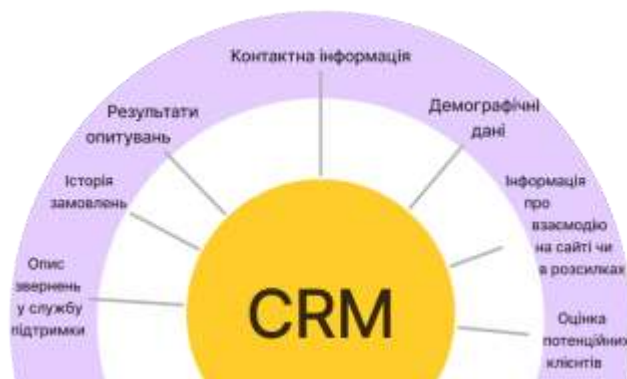


Рисунок 1.7 – Вхідні до CRM дані

Візуальне представлення та реальне застосування методів обробки даних в CRM-системах оживляють теоретичні концепції. Ці ілюстрації демонструють, як компанії з різних галузей використовують методи обробки даних CRM у безліч способів. Приклади варіюються від сегментації клієнтів і персоналізованих маркетингових стратегій до автоматизованих рішень для підтримки клієнтів. Ці застосування підкреслюють адаптивність і універсальність CRM-систем у задоволенні унікальних потреб різних галузей.

Вичерпна оцінка сильних і слабких сторін, пов'язаних з різними методами обробки даних, є обов'язковою. У цьому розділі розглядаються переваги та обмеження кожного підходу, беручи до уваги такі фактори, як точність, масштабованість, вимоги до ресурсів та адаптивність до різних бізнес-контекстів [16]. Він демонструє, що придатність цих методів може сильно відрізнятися, і організації повинні робити обґрунтований вибір, виходячи зі своїх конкретних потреб і цілей.

## 1.1 Метод порівняння метрик з галузевими стандартами

У багатьох дослідженнях рекомендується проводити порівняльний аналіз ефективності CRM-системи з галузевими стандартами та передовим досвідом. Це передбачає порівняння можливостей обробки даних системи з визнаними еталонами, щоб переконатися, що вона відповідає встановленим нормам або перевищує їх (рис. 1.8).

```
import numpy as np
# Simulated data for the CRM data processing method
method_performance = np.array([85, 90, 88, 92, 87, 91, 89, 93, 86, 90])
# Industry benchmark data (mean and standard deviation)
industry_mean = 88 # Industry average performance
industry_stddev = 3 # Industry standard deviation
# Calculate Z-score
z_score = (method_performance.mean() - industry_mean) / (industry_stddev /
np.sqrt(len(method_performance)))
# Set significance level (alpha)
alpha = 0.05
# Import the norm module from scipy.stats to perform Z-test
from scipy.stats import norm
# Calculate the p-value using the Z-score
p_value = 2 * (1 - norm.cdf(np.abs(z_score)))
# Compare p-value to alpha
if p_value < alpha:
    print("The CRM data processing method performs significantly better than industry standards.")
else:
    print("There is no significant difference between the CRM method and industry standards.")
```

Рисунок 1.8 – Приклад коду методу порівняння з галузевими стандартами

У цьому прикладі коду на Python порівнюється продуктивність методу обробки даних CRM з галузевими стандартами за допомогою Z-тесту.

*method\_performance* представляє результати роботи методу CRM.

*industry\_mean* і *industry\_stddev* представляють середнє значення і стандартне відхилення для метрики продуктивності.

Тут обчислюється Z-рахунок, який показує, на скільки стандартних відхилень показники ефективності методу CRM відрізняються від середньогалузевого показника. Після чого обчислюється двостороннє *p*-значення за допомогою Z-рахунку [17].

Якщо  $p$ -значення менше обраного рівня значущості (альфа), який встановлено на 0,05, то можна зробити висновок, що метод обробки даних CRM працює значно краще, ніж галузеві стандарти. В іншому випадку робимо висновок про відсутність значущої різниці [18].

Він базується на  $Z$ -тесті для порівняння середнього значення вибірки із середнім значенням генеральної сукупності. Формула для  $Z$ -критерію, що використовується у цьому коді, така:

$$Z = \frac{(X-\mu)}{\sigma}, \quad (1.1)$$

де  $Z$ -рахунок вимірює, на скільки стандартних відхилень середнє значення вибірки відрізняється від середнього значення генеральної сукупності. У контексті коду він допомагає визначити, чи суттєво відрізняється продуктивність методу обробки даних CRM від галузевих стандартів.

Цей код демонструє, як технічно порівняти метод обробки даних CRM з галузевими стандартами і визначити, чи відповідає він цим стандартам, чи перевищує їх.

## 1.2 Метод А/В-тестування

А/В-тестування або спліт-тестування зазвичай використовується для оцінки впливу рішень, заснованих на даних, на бізнес-результати. Порівнюючи дві версії CRM-стратегії (одна з використанням впроваджених методів, а інша без них), організації можуть кількісно оцінити досягнуті покращення.

А/В-тестування в контексті перевірки методів обробки даних CRM.

У цьому прикладі використовується Python і популярну бібліотеку для А/В-тестування `scipy` (рис. 1.9).

```

import numpy as np
from scipy import stats
# Sample data for two groups (A and B)
group_A = np.array([0, 1, 1, 1, 0, 0, 0, 1, 0, 1]) # Control group
group_B = np.array([1, 1, 1, 1, 0, 1, 0, 0, 0, 1]) # Experimental group
# Perform A/B test (two-sample t-test)
t_stat, p_value = stats.ttest_ind(group_A, group_B)
# Set significance level (alpha)
alpha = 0.05
# Compare p-value to alpha
if p_value < alpha:
    print("There is a significant difference between group A and group B.")
else:
    print("There is no significant difference between group A and group B.")

```

Рисунок 1.9 – Приклад коду методу А/В-тестування

У цьому фрагменті коду Python є дві групи, `group_A` та `group_B`, які представляють дві варіації (наприклад, контрольну та експериментальну) методу обробки даних CRM. Використовується двовибірковий  $t$ -тест з бібліотеки `scipy` для порівняння середніх значень двох груп.

`t_stat` представляє  $t$ -статистику, яка вимірює різницю між середніми двох груп.

`p_value` – це  $p$ -значення, яке показує нам ймовірність спостереження даних за відсутності значущої різниці між групами.

Встановлено рівень значущості (альфа) 0,05, що є загальним порогом при перевірці гіпотез. Якщо  $p$ -значення менше за альфа, то висновок, що існує значна різниця між двома групами, що вказує на те, що метод обробки даних CRM мав вплив. В іншому випадку робимо висновок, що значущої різниці немає [19]. Цей приклад демонструє, як можна використовувати А/В-тестування і статистичний аналіз для перевірки ефективності методів обробки даних CRM в технічному контексті. Приклад: інтернет-магазин хоче оптимізувати email-маркетингові кампанії, використовуючи дані CRM. Вони проводять А/В-тест, надсилаючи дві різні версії листа: одну з рекомендаціями щодо товарів на основі даних CRM, а іншу – із загальними рекомендаціями. Вони виявляють, що рекомендації на основі CRM призводять до на 20% вищого коефіцієнту конверсії.

### 1.3 Метод $K$ -середніх

$K$ -середні – це універсальний і широко використовуваний метод неконтрольованого машинного навчання для сегментації даних. У сфері CRM його використання відкриває можливості для сегментації клієнтів, персоналізованого маркетингу та покращення клієнтського досвіду.

За своєю суттю,  $K$ -середні – це метод неконтрольованого машинного навчання, який використовується для кластеризації або групування даних. Він працює за принципом поділу точок даних на « $K$ » окремих кластерів на основі схожості. Основна концепція полягає в мінімізації внутрішньокластерної дисперсії та максимізації міжкластерної дисперсії [20].

Алгоритм  $K$ -середніх слідує простому ітеративному процесу:

- ініціалізація: випадковий вибір « $K$ » початкових центроїдів кластерів;
- присвоєння: присвоєння кожній точці даних найближчого центроїда, формуючи « $K$ » кластерів;
- оновлення центроїдів: перерахувати центроїди на основі середнього значення точок даних у кожному кластері;
- ітерація: повторюйте кроки призначення та оновлення центроїдів до збіжності (мінімального зсуву центроїдів).

Одне з найвідоміших застосувань  $K$ -середніх в CRM – це сегментація клієнтів. Аналізуючи атрибути клієнтів, такі як історія покупок, демографічні дані та поведінка в Інтернеті,  $K$ -середніх може класифікувати клієнтів на окремі сегменти. Ці сегменти можуть стати основою для цільових маркетингових стратегій та персоналізованих рекомендацій щодо продуктів.

$K$ -середніх дозволяє компаніям адаптувати маркетингові кампанії до конкретних сегментів клієнтів. Така персоналізація підвищує ефективність кампанії та залучення клієнтів. Розуміючи вподобання та поведінку різних груп клієнтів, компанії можуть надавати більш релевантний контент та пропозиції [21].

Перед застосуванням  $K$ -середніх дані CRM повинні пройти попередню обробку. Це включає обробку пропущених значень, масштабування ознак і кодування категорійних змінних. Належна попередня обробка забезпечує ефективність алгоритму.

Вибір правильного значення « $K$ » (кількість кластерів) має вирішальне значення. Такі методи, як метод ліктя або оцінка силуету, можуть допомогти знайти оптимальне значення « $K$ » для певного набору даних [22].

Набори даних CRM часто охоплюють численні характеристики. Методи зменшення розмірності, такі як аналіз головних компонент, можуть бути використані для пом'якшення прокляття розмірності.

Для обчислення нового центроїда для кожного кластера (середнього значення точок даних кластера) використовується наступна формула:

$$c = \left( \frac{1}{|C|} \sum_{i=1}^{|C|} p_{i1}, \frac{1}{|C|} \sum_{i=1}^{|C|} p_{i2}, \dots, \frac{1}{|C|} \sum_{i=1}^{|C|} p_{im} \right), \quad (1.2)$$

де  $|C|$  – кількість точок даних у кластері  $C$ ;

$p_{ij}$  –  $j$ -та координата  $i$ -тої точки даних у кластері;

$c$  – центроїд кластера, який є вектором з  $m$  вимірами.

Евклідова відстань між двома точками даних  $x$  та  $y$  у багатовимірному просторі обчислюється наступним чином:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}, \quad (1.3)$$

де  $x = (x_1, x_2, \dots, x_n)$  – перша точка в  $n$ -вимірному просторі;

$y = (y_1, y_2, \dots, y_n)$  – друга точка в  $n$ -вимірному просторі;

$x_i$  –  $i$ -та координата точки  $x$ ;

$y_i$  –  $i$ -та координата точки  $y$ ;

$d(x, y)$  – евклідова відстань між точками  $x$  та  $y$ .

Для розподілу точок даних по кластерах алгоритм обчислює евклідову відстань між кожною точкою даних та центроїдами всіх кластерів і відносить точку даних до кластера з найближчим центроїдом.

#### 1.4 Постановка задачі дослідження

У сучасному бізнес-середовищі першочергова потреба в ефективному використанні даних стає все більш очевидною, оскільки організації борються зі зростаючою конкуренцією. Вирішальним фактором успіху в цьому динамічному середовищі є вмiле використання даних для прийняття обґрунтованих рішень.

Одним з найважливіших інструментів у цьому прагненні є системи управління взаємовідносинами з клієнтами (CRM). З роками ці системи перетворилися на незамінні інструменти, які відіграють 24 ключову роль у забезпеченні безперешкодної взаємодії з клієнтами та формуванні стратегій, заснованих на даних, що сприяють успіху в бізнесі.

Оскільки бізнес-ландшафт продовжує розвиватися, попит на інноваційні підходи до обробки даних в CRM-системах стає все більш нагальним. Зазначені підходи повинні не лише забезпечувати стійку конкурентоспроможність організацій, але й сприяти покращенню бізнесрезультатів. Виникає нагальна і однозначна потреба в глибокому і вичерпному дослідженні та вдосконаленні методів обробки даних в CRM-системах.

Таким чином, дослідження методів оброблення даних у CRM-системі організації бізнесу є актуальним завданням для вище наведених методів. Тому ставиться завдання розробки програмних алгоритмів таких методів як К-середніх, А/В-тестування та порівняння з галузевими стандартами.

Об'єктом дослідження є процес оброблення даних у CRM-системі організації бізнесу.

Метою дослідження є порівняння методів оброблення даних у CRM-системі організації бізнесу.

Для досягнення мети необхідно вирішити такі завдання:

- провести аналіз найбільш актуальних існуючих методів аналізу даних;
- розробити тестову базу даних для CRM-системи та заповнити її тестовими даними;
- реалізувати комп'ютерну модель кожного з методів для аналізу даних з бази даних;
- провести тестування цих методів на інших тестових даних і впевнитися в їх точності.

## 2 МАТЕМАТИЧНІ МОДЕЛІ АНАЛІЗУ ОБРОБЛЕННЯ ДАНИХ

### 2.1 Модель A/B-тестування в CRM-системах

У цифровому ландшафті, що розвивається, A/B-тестування стало фундаментальним інструментом для бізнесу, який прагне оптимізувати свою присутність в Інтернеті. У сфері CRM-систем A/B-тестування має неоціненне значення, забезпечуючи суворий підхід до порівняння декількох варіантів стратегії, кампанії або цифрового активу. Основна мета залишається незмінною: визначити, який варіант найефективніше відповідає вподобанням і поведінці цільової аудиторії [23].

Методологія A/B-тестування ґрунтується на перевірці гіпотез – фундаментальній концепції статистики. Передумовою є визначення того, чи є спостережувані відмінності між двома або більше групами статистично значущими або просто результатом випадкової дисперсії.

#### 2.1.1 Коефіцієнт конверсії

Коефіцієнт конверсії слугує основним показником для багатьох A/B-тестів. По суті, він кількісно визначає частку користувачів, які виконують заздалегідь визначену дію, наприклад, реєструються, купують або натискають на певний елемент.

$$\text{Коефіцієнт конверсії} = \left( \frac{\text{Кількість конверсій}}{\text{Кількість відвідувань}} \right) \times 100\%, \quad (2.1)$$

де кількість конверсій – це кількість випадків, коли відвідувачі вебсайту або користувачі застосунку виконали цільову дію. Це може бути покупка продукту, реєстрація на вебсайті, підписка на розсилку новин тощо;

кількість відвідувань – загальна кількість відвідувань вебсайту або взаємодій із застосунком за певний період.

Припустимо, вводимо інтернет-магазин, і за певний місяць сайт отримав 20000 відвідувань. Протягом цього ж місяця зафіксували 400 продажів (конверсій).

$$\text{Коефіцієнт конверсії} = \left( \frac{400}{20000} \right) \times 100\% = 0,02 \times 100\% = 2\%. \quad (2.2)$$

Це означає, що 2% відвідувань сайту призводять до продажу. Цей коефіцієнт дозволяє оцінити ефективність вашого сайту або маркетингової стратегії в цілому.

### 2.1.2 Оцінка статистичної значущості

Для будь-якої виявленої різниці між варіантами в А/В тесті необхідно переконатися, що вона не є просто випадковою. Для цього необхідна концепція статистичної значущості. Для цього часто використовують двопропорційний  $z$ -тест:

Формула  $Z$ -критерію:

$$Z = \frac{p_A - p_B}{\sqrt{p(1-p)\left(\frac{1}{n_A} + \frac{1}{n_B}\right)}}, \quad (2.3)$$

де  $p_A$  та  $p_B$  – представляють коефіцієнти конверсії для версій  $A$  та  $B$ ;

$p$  – об'єднаний коефіцієнт конверсії обох версій;

$n_A$  та  $n_B$  – позначають обсяги вибірок для версій  $A$  та  $B$ .

Розглянемо приклад: якщо версія  $A$  має 5% конверсію з 1500 відвідувачів, а версія  $B$  – 6% з 1700 відвідувачів, то  $z$ -рахунок можна обчислити за (2.3).

z-рахунок згодом допомагає визначити статистичну значущість відмінностей.

Застосування А/В-тестування в CRM-системах:

– з метою покращення користувацького інтерфейсу та користувацького досвіду (UI/UX). В CRM невеликі зміни, такі як колірна гамма, розташування кнопок або типографіка, можуть помітно вплинути на залученість користувачів. А/В-тестування допомагає визначити найбільш зручні та привабливі елементи дизайну;

– задля персоналізації контенту. Контент, який резонує особисто з користувачами, часто призводить до кращого залучення. Дані CRM-системи можна використовувати для створення різноманітного контенту, орієнтованого на різні сегменти. А/В-тестування може підтвердити, який варіант контенту є найефективнішим для певного сегмента;

– для реалізації стратегії продажів і маркетингу. Тестуючи різні способи продажу, рекламні пропозиції або маркетингові повідомлення, компанії можуть визначити, які стратегії приваблюють їхню аудиторію і стимулюють продажі;

– в цілях підтримки клієнтів та зворотній зв'язок. Різні підходи до підтримки клієнтів, чи то формулювання чат-ботів, чи шаблони відповідей електронною поштою, можна протестувати, щоб знайти найефективнішу стратегію для задоволення потреб клієнтів.

### 2.1.3 Багатовимірне тестування (MVT)

Багатовимірне тестування (MVT) – це вдосконалена форма А/В-тестування, призначена для одночасної оцінки декількох варіантів. На відміну від свого аналога, який порівнює лише дві версії (А і В) однієї змінної, MVT тестує кілька комбінацій декількох змінних.

У контексті CRM-систем цей метод є безцінним у визначенні оптимальних комбінацій, які можуть підвищити залученість користувачів і загальну ефективність бізнесу.

В основі MVT лежить факторний дизайн. Цей підхід дозволяє систематично тестувати всі можливі комбінації змінних, що розглядаються. Наприклад, якщо інтерфейс CRM-системи має три різні елементи дизайну (наприклад, колір кнопок, розташування кнопок і типографіку), і кожен з них має три варіанти, MVT оцінює всі 27 можливих комбінацій.

У той час як А/В-тестування в основному покладається на базові порівняння пропорцій, MVT вимагає більш складних статистичних моделей, враховуючи безліч комбінацій.

Застосування MVT в CRM-системах:

- оптимізація інтерфейсу. MVT дозволяє адміністраторам CRM визначити найкращий макет інтерфейсу;

- автоматизоване тестування робочого процесу. MVT може оцінити різні шляхи робочого процесу. Це гарантує, що клієнти та потенційні клієнти отримують найбільш ефективні та дієві послідовності комунікації;

- розширені аналітичні дані. Завдяки багатовимірному тестуванню CRM-системи можуть заглибитися в аналітику поведінки користувачів, щоб сприяти підвищенню залученості та лояльності користувачів.

Вибір між А/В тестуванням та MVT має залежати від розміру вибірки, взаємодії з перемінними, комплексністю та часозатратами, розуміння лімітів необхідної оптимізації.

## 2.2 Модель методу *K*-середніх

У сфері аналізу даних в CRM-системах метод кластеризації *K*-середніх є однією з найпоширеніших методик. Основна мета полягає в тому, щоб розбити дані на окремі кластери, які в контексті CRM-системи можуть виявити закономірності в поведінці, вподобаннях або демографічних характеристиках клієнтів.

Центральним елементом алгоритму  $K$ -середніх є поняття відстані, яке визначає, як точки даних розподіляються на кластери. Найпоширенішою метрикою відстані є евклідова відстань, хоча можна використовувати й інші, наприклад, Манхеттенську або Мінковського [24].

Евклідова відстань між двома точками  $p$  та  $q$  у  $n$ -вимірному просторі задано за допомогою наступної формули:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}, \quad (2.4)$$

де  $d(p, q)$  – евклідова відстань між точкою даних  $p$  та центроїдом  $q$  або між двома точками даних;

$p=(p_1, p_2, \dots, p_n)$  – вектор координат першої точки у  $n$ -вимірному просторі. У контексті  $K$ -середніх, це окрема точка даних із датасету;

$q=(q_1, q_2, \dots, q_n)$  – вектор координат другої точки у  $n$ -вимірному просторі. У методі  $K$ -середніх, це зазвичай центроїд одного з кластерів;

$p_i$  –  $i$ -та координата точки  $p$ , яка представляє  $i$ -ту властивість або характеристику точки даних у датасеті;

$q_i$  –  $i$ -та координата точки  $q$ , яка представляє  $i$ -ту властивість або характеристику центроїда кластера;

$(p_i - q_i)^2$  – квадрат різниці між  $i$ -тими координатами точок  $p$  та  $q$ , який вказує на квадратичну відстань між двома точками в  $i$ -тому вимірі.

Фундаментальна математична мета алгоритму  $K$ -середніх полягає в мінімізації суми квадратів всередині кластера (WCSS). Ця мета відображає суть методу: кожна точка повинна бути якомога ближче до центроїда свого кластера. Математичне представлення WCSS:

$$WCSS = \sum_{i=1}^K \sum_{x \in C_i} \|x - \mu_i\|^2, \quad (2.5)$$

де  $\sum_{i=1}^K$  – зовнішня сума, що пробігає через усі кластери, де  $K$  – кількість кластерів;

$\sum_{x \in C_i}$  – внутрішня сума, що пробігає через усі точки даних  $x$ , які належать до кластера  $C_i$ ;

$x$  – окрема точка даних у просторі характеристик;

$\mu_i$  – центроїд або середнє значення точок у кластері  $C_i$ ;

$\|x - \mu_i\|^2$  – квадрат евклідової відстані між точкою даних  $x$  та центроїдом  $\mu_i$ , це виражає «відхилення» точки даних від центроїда її кластера. Квадрат відстані використовується для підкреслення більших відхилень.

Тут внутрішнє підсумовування обчислює квадратичну відстань між усіма точками даних у кластері  $C_i$  до центроїда кластеру  $\mu_i$ . Потім зовнішнє підсумовування агрегує ці значення по всіх  $K$  кластерах. Після віднесення точок даних до кластерів, центроїди перераховуються. Центроїд  $\mu_i$  для кожного кластеру  $C_i$  є середнім значенням всіх точок даних у  $C_i$  і обчислюється як:

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x, \quad (2.6)$$

де  $|C_i|$  – кількість точок даних у кластері  $C_i$ .

$K$ -середніх ітеративно уточнюють кластери, чергуючи призначення та перерахунок центроїдів. Збіжність, як правило, визначається, коли зміна  $WCSS$  між двома послідовними ітераціями є нижчою за попередньо визначений поріг  $\epsilon$ :

$$|WCSS_{t+1} - WCSS_t| < \epsilon. \quad (2.7)$$

В області  $K$ -середніх, визначення оптимальної кількості кластерів  $K$  має першорядне значення.

Однією з метрик, яка може допомогти в цьому рішенні, є коефіцієнт силуету, який вимірює, наскільки близько кожна точка даних в одному кластері знаходиться до точок даних у сусідніх кластерах [25]. Математично, значення силуету  $s(i)$  для кожної точки даних  $i$  задається формулою:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad (2.8)$$

де  $a_i$  – середня відстань від  $i$ -ої точки даних до інших точок у тому ж кластері;

$b_i$  – найменша середня відстань від  $i$ -ої точки даних до точок даних в іншому кластері, мінімізована за кластерами.

Коефіцієнт силуету для всього набору даних є середнім значенням  $s(i)$  за всіма точками даних і коливається від -1 до 1. Високе значення вказує на добре відокремлені кластери.

Запуск процесу  $K$ -середніх передбачає випадковий вибір  $K$  точок даних з набору даних і позначення їх як початкових центроїдів. Потім кожній точці даних ставиться у відповідність найближчий центроїд, фактично формуючи початкові  $K$  кластерів. На цій основі центроїд кожного кластера перераховується на основі середнього значення точок даних, що його охоплюють. Цей крок може дещо змінити положення центроїда. Згодом точки даних можуть опинитися ближче до іншого центроїда. Таким чином, відбувається переприсвоєння, перерозподіл точок даних до їхніх найближчих центроїдів. Цей процес перерахунку центроїдів і перепризначення точок даних повторюється до тих пір, поки переміщення центроїдів не стане незначним або поки не буде досягнута заздалегідь визначена кількість ітерацій [26].

Проте, хоча метод  $K$ -середніх є потужним, він не позбавлений тонкощів. Одне з ключових рішень полягає у визначенні оптимальної кількості кластерів  $K$ .

Різні методи, такі як метод ліктя, який передбачає побудову графіків WCSS для різних значень  $K$ , пропонують вказівки щодо цього.

У застосуванні до CRM ці математичні нюанси гарантують, що дані про клієнтів будуть сегментовані надійно і змістовно, що дозволить розробити більш цілеспрямовані та ефективні бізнес-стратегії (рис. 2.1).

Інсайти, отримані в результаті такої кластеризації, можуть пролити світло на моделі продажів. Виявлення кластеру, який демонструє певну тенденцію купівлі, може бути корисним для планування запасів або визначення найефективнішого часу для проведення рекламних заходів.

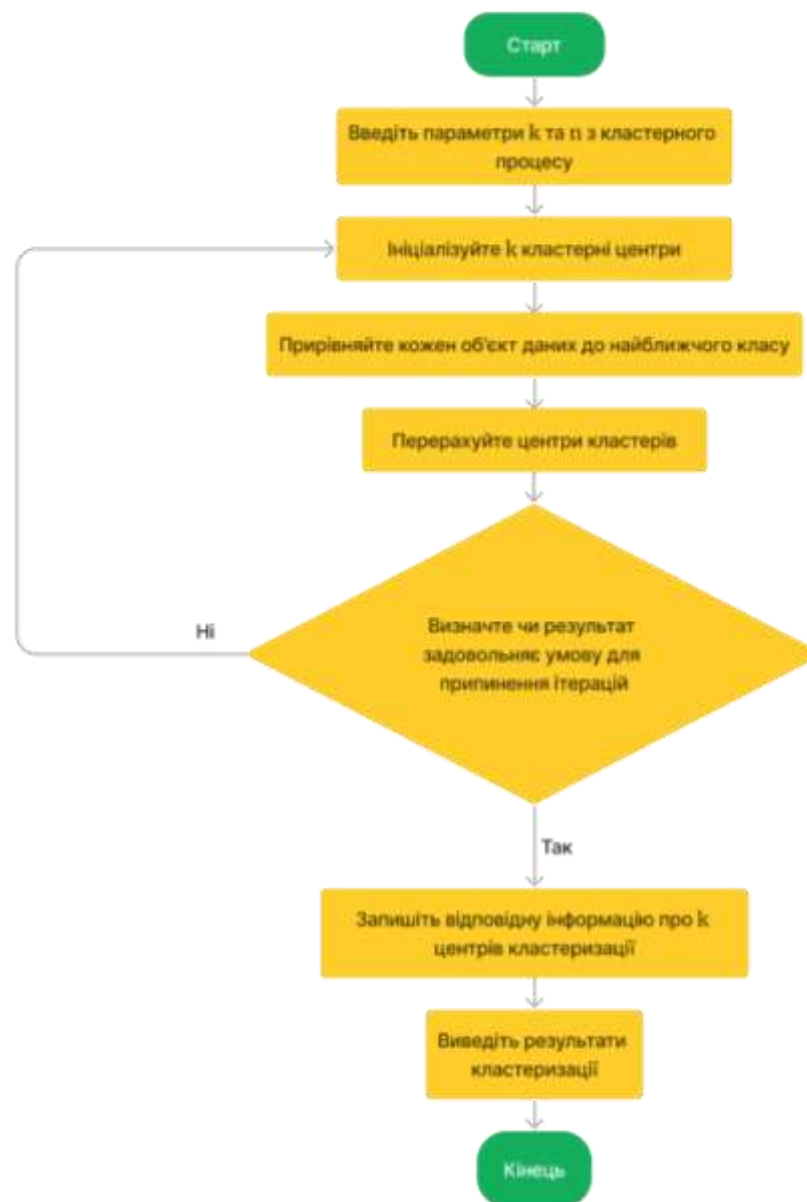


Рисунок 2.1 – Блок-схема потоку роботи методу  $K$ -середніх

Для знаходження оптимальної кількості кластерів використовується метод ліктя. Він передбачає побудову графіка залежності значення функції витрат,  $WCSS$ , від кількості кластерів  $K$ . З математичної точки зору, точка перегину  $K_{\text{оптимальна}}$  можна апроксимувати за допомогою:

$$K_{\text{оптимальна}} = \operatorname{argmin}_k \left| \frac{WCSS_{k+1} - WCSS_k}{WCSS_k - WCSS_{k-1}} \right|. \quad (2.9)$$

Цей метод на графіку часто призводить до появи форми, котра нагадує лікоть, який вказує на оптимальне значення для  $K$ , коли додавання нових кластерів не забезпечує суттєвого покращення відповідності.

Перенесення цього методу в CRM-систему відкриває безліч застосувань. Дані про клієнтів, що містять історії покупок, записи про взаємодію та інші пов'язані з ними показники, створюють сприятливе підґрунтя для кластеризації за методом  $K$ -середніх. Наприклад, кластеризація клієнтів на основі їхніх купівельних звичок і частоти взаємодії може виявити окремі сегменти. На ці сегменти можна націлити індивідуальні маркетингові кампанії, що сприятиме залученню клієнтів і збільшенню ймовірності продажів [27].

Крім того, у сфері підтримки клієнтів розуміння кластерів може мати трансформаційний характер. Якщо певна група клієнтів постійно стикається з певною проблемою або має спільні інтереси, механізм підтримки може бути скоригований. Це може бути у формі спеціалізованого навчання для команд підтримки, спеціального розподілу ресурсів або навіть модифікації пропозиції продуктів чи послуг.

### **3 КОМП'ЮТЕРНА МОДЕЛЬ ДОСЛІДЖЕННЯ МЕТОДІВ АНАЛІЗУ CRM-СИСТЕМИ ОРГАНІЗАЦІЇ БІЗНЕСУ**

#### **3.1 Обґрунтування вибору середовища програмної реалізації**

У сфері комп'ютерних наук, особливо коли йдеться про системи управління взаємовідносинами з клієнтами (CRM), вибір відповідного програмного середовища є не просто технічним рішенням, а таким, що може кардинально вплинути на ефективність, стійкість і масштабованість рішення. В рамках цього дослідження було визначено, що конвергенція MS SQL Server як обраної системи управління базами даних (СУБД), доповнена використанням T-SQL для запитів до бази даних, і Python 3.9 як мови програмування, є найбільш сприятливою для розробки обчислювальної моделі.

MS SQL Server виділяється в галузі завдяки своєму комплексному набору інструментів для надійного управління даними. Висока продуктивність у поєднанні з непохитною надійністю робить його кращим вибором для підприємств, яким потрібна СУБД, здатна обробляти великі та складні набори даних, що є типовими для CRM-застосунків. Важливість цілісності даних і забезпечення узгодженості транзакцій в CRM-системах неможливо переоцінити, а дотримання MS SQL Server властивостей ACID (атомарність, узгодженість, ізолюваність, довговічність) гарантує, що ці невід'ємні атрибути будуть збережені.

Вибір MS SQL Server як системи управління базами даних ґрунтується на його перевіреному досвіді роботи з великими обсягами транзакцій і запитів, що є типовою вимогою для CRM-систем, які зберігають величезні обсяги даних про клієнтів. Функції оптимізації продуктивності, доступні в MS SQL Server, забезпечують ефективне виконання запитів, особливо складних і таких, що включають значні обсяги даних.

Механізм бази даних розроблений для оптимізації планів запитів і забезпечення швидкого доступу до даних, що є важливим для аналітики та звітності в режимі реального часу в контексті CRM [28].

Використання T-SQL як мови запитів має перевагу в тому, що вона безпосередньо інтегрована з MS SQL Server, забезпечуючи багатий набір команд, спеціально оптимізованих для складних операцій з даними. Здатність T-SQL виконувати процедурний код, контроль транзакцій, обробка винятків і помилок, а також потужний набір функцій для маніпулювання даними в сукупності гарантують, що аналітичні потреби даних CRM будуть належним чином задоволені.

T-SQL є свідченням універсальності екосистеми SQL Server. Його глибока інтеграція дозволяє створювати запити, які не лише ефективно працюють, але й можуть бути точно налаштовані відповідно до тонких потреб аналізу даних CRM. Незалежно від того, чи це агрегація даних про взаємодію з клієнтами, чи аналіз тенденцій продажів, чи аналіз відгуків клієнтів для отримання дієвих висновків, T-SQL забезпечує необхідну гнучкість. Вона дозволяє інкапсулювати складну бізнес-логіку в збережені процедури, функції та тригери, таким чином вбудовуючи бізнес-правила безпосередньо в рівень бази даних [29].

Крім того, інтеграція між Python та MS SQL Server є безперешкодною завдяки таким бібліотекам, як pyodbc та SQLAlchemy, які дозволяють виконувати запити до T-SQL з середовища Python. Ця синергія дозволяє розробленій обчислювальній моделі використовувати надійні можливості обробки даних MS SQL Server, одночасно отримуючи вигоду від аналітичної та обчислювальної потужності Python. Переходячи на Python, стикаємось з мовою, яка була ретельно розроблена з думкою про кінцевого користувача. Синтаксис Python славиться своєю ясністю, що значно знижує вхідний бар'єр для нових розробників та дослідників даних. У контексті цього дослідження, де будуються та ітеративно використовуються складні моделі даних, можливість створення прототипів за допомогою Python є безцінною.

Його інтерпретаційна природа забезпечує негайний зворотній зв'язок під час процесу розробки, що сприяє більш експериментальному та ітеративному підходу до вирішення проблем.

В контексті обробки даних в CRM-системах здатність швидко адаптуватися до мінливих ринкових тенденцій та шаблонів клієнтських даних має вирішальне значення.

Обраний технологічний стек пропонує динамічну та стійку основу, яка підтримує аналіз даних у режимі реального часу, прогнозне моделювання та отримання дієвих інсайтів, що є першочерговим для підтримки конкурентної переваги в бізнес-сфері [30].

Таким чином, об'єднання MS SQL Server, T-SQL і Python 3.9 представляє цілісне і надійне середовище для розробки обчислювальної моделі для дослідження методологій обробки даних в CRM-системі (рис. 3.1). Ця комбінація пропонує баланс продуктивності, простоти використання та аналітичної глибини, що робить її придатною для вирішення поставлених дослідницьких завдань.

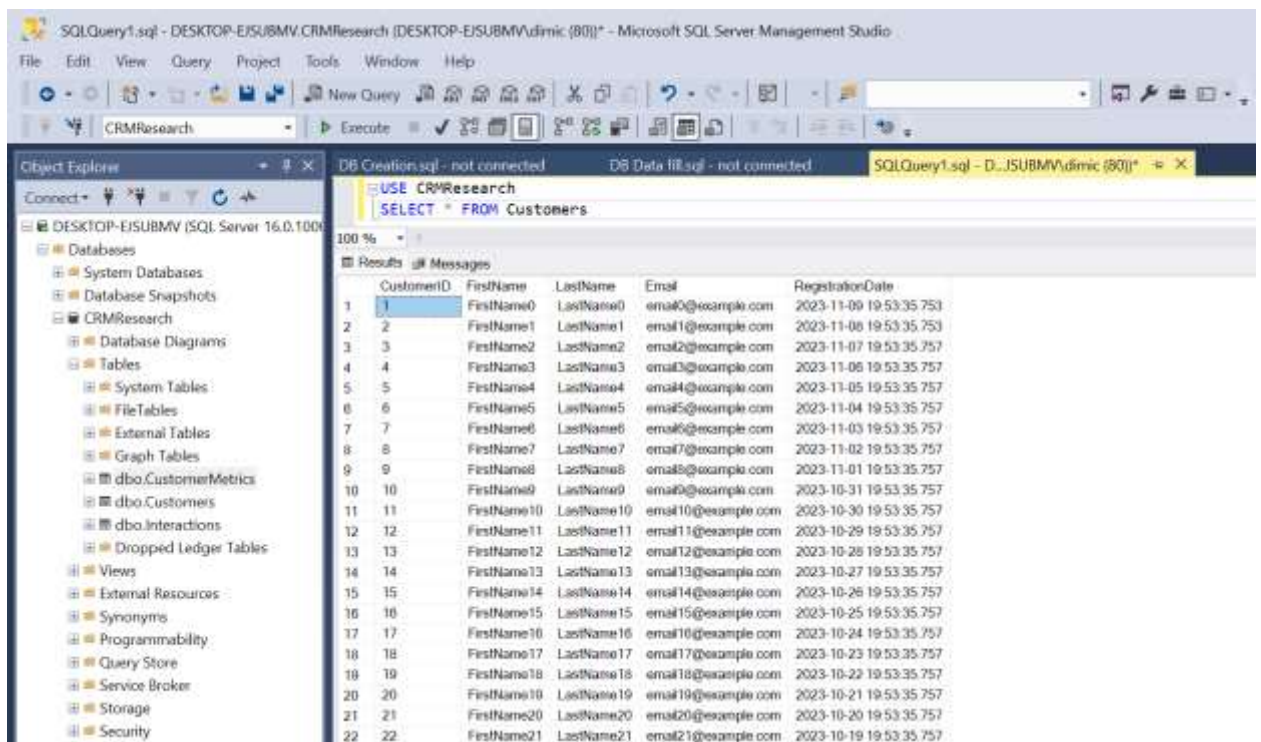


Рисунок 3.1 – Приклад інтерфейсу SSMS

Єдине, що повинен зробити користувач, працюючи із графічними компонентами, – це визначити характеристики використовуваних знарядь малювання.

### 3.2 Програмна реалізація

Інтеграція CRM-систем з базами даних та аналітичними алгоритмами означає зміну парадигми того, як бізнес використовує силу даних. Вона виходить за рамки механіки програмної інженерії, занурюючись у сферу, де стратегічне передбачення зустрічається з технологічною досконалістю. Ця інтеграція – це не просто з'єднання двох систем, а створення симбіотичної екосистеми, в якій кожна взаємодія з клієнтом фіксується, аналізується і перетворюється на стратегічний актив.

Створення надійного каналу для обміну даними є наріжним каменем цієї інтеграції. Це цифрова артерія, яка переносить життєво важливу інформацію про клієнтів, спрямовуючи її з передової бізнес-операцій в аналітичне серце організації. Цей канал є одночасно і рятівним кругом, і магістраллю даних, гарантуючи, що інформація не просто передається, а й перекладається на мову, готову для аналітичного аналізу. У міру того, як база даних накопичує дані, стає можливою реалізація аналітичних алгоритмів. Ці алгоритми, незалежно від того, чи це сегментація, прогнозне моделювання чи аналіз настроїв, застосовуються для виявлення закономірностей і розуміння поведінки та вподобань клієнтів. Реалізація може бути внутрішньою, в базі даних за допомогою збережених процедур, або зовнішньою, з використанням складних платформ науки про дані [31].

Інтеграція CRM-систем з базами даних та аналітичними алгоритмами – це технічний процес, який включає в себе кілька ключових кроків для забезпечення ефективного збору та використання даних для бізнес-аналітики.

Спочатку встановлюється зв'язок між CRM-системою і базою даних за допомогою таких протоколів, як JDBC або ODBC. Це з'єднання має бути безпечним, надійним і здатним обробляти великі обсяги даних. Управління потоком даних має вирішальне значення, часто із залученням ETL-процесів, щоб забезпечити чистоту, узгодженість і належну структуру даних для аналізу.

Після того, як дані потрапляють до бази даних, застосовуються аналітичні алгоритми. Це можуть бути статистичні методи для виявлення тенденцій і закономірностей або алгоритми машинного навчання для прогнозування майбутньої поведінки клієнтів або тенденцій продажів. Вибір алгоритму залежить від типу даних і бізнес-питань, на які потрібно відповісти.

Ці аналітичні процеси, як правило, відбуваються в сховищі даних або озері даних, де дані з CRM-системи об'єднуються з іншими джерелами даних. Сховище даних стає центральним репозиторієм, з якого фахівці з даних та аналітики можуть отримувати інформацію.

Проаналізовані дані мають бути представлені у зрозумілій формі, щоб їх можна було використати для подальших дій. Саме тут у гру вступають інструменти візуалізації даних, які перетворюють складні набори даних на графічні зображення, що висвітлюють ключові тенденції та ідеї. Дашборди зазвичай використовуються для динамічного представлення даних, дозволяючи користувачам заглиблюватися в метрики і розуміти основні фактори, що впливають на ефективність бізнесу [32].

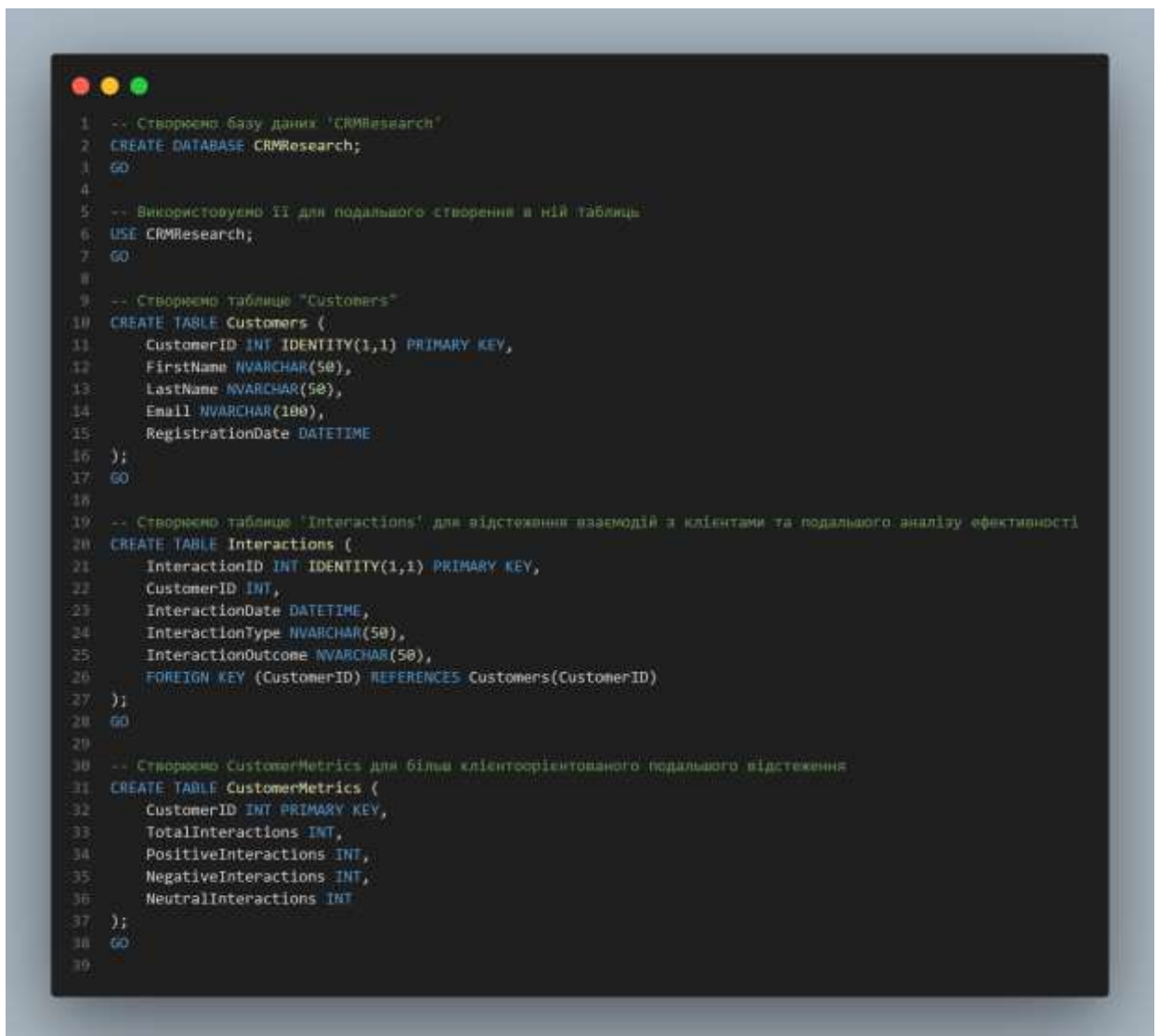
Процес інтеграції є ітеративним і вимагає регулярного оновлення та обслуговування, щоб система залишалася ефективною, а дані, які вона надає, залишалися актуальними і точними. Це передбачає моніторинг якості даних, оновлення ETL-скриптів, вдосконалення алгоритмів і забезпечення відповідності інструментів візуалізації потребам бізнесу.

Для спрощення процесу значна частина обробки та аналізу даних автоматизована. Заплановані сценарії виконують необхідні обчислення і

оновлюють інформаційні панелі та звіти. Така автоматизація гарантує, що особи, які приймають рішення, мають доступ до найсвіжішої інформації без ручного втручання.

Отримані в результаті інсайти цінні настільки, наскільки вони здатні впливати на прийняття рішень. Тут на перший план виходить візуалізація даних, яка слугує оповідачем у світі даних. Вона перетворює абстрактні цифри на візуальні розповіді, роблячи складне простим, а незрозуміле – зрозумілим.

Дашборди та звіти стають полотном, на якому пишеться історія даних, надаючи динамічний та інтерактивний портрет здоров'я бізнесу (рис. 3.2).



```
1 -- Створимо базу даних 'CRMResearch'
2 CREATE DATABASE CRMResearch;
3 GO
4
5 -- Використовуємо II для подальшого створення в ній таблиць
6 USE CRMResearch;
7 GO
8
9 -- Створимо таблицю 'Customers'
10 CREATE TABLE Customers (
11     CustomerID INT IDENTITY(1,1) PRIMARY KEY,
12     FirstName NVARCHAR(50),
13     LastName NVARCHAR(50),
14     Email NVARCHAR(100),
15     RegistrationDate DATETIME
16 );
17 GO
18
19 -- Створимо таблицю 'Interactions' для відстеження взаємодій з клієнтами та подальшого аналізу ефективності
20 CREATE TABLE Interactions (
21     InteractionID INT IDENTITY(1,1) PRIMARY KEY,
22     CustomerID INT,
23     InteractionDate DATETIME,
24     InteractionType NVARCHAR(50),
25     InteractionOutcome NVARCHAR(50),
26     FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
27 );
28 GO
29
30 -- Створимо CustomerMetrics для більш клієнтоорієнтованого подальшого відстеження
31 CREATE TABLE CustomerMetrics (
32     CustomerID INT PRIMARY KEY,
33     TotalInteractions INT,
34     PositiveInteractions INT,
35     NegativeInteractions INT,
36     NeutralInteractions INT
37 );
38 GO
39
```

Рисунок 3.2 – Створення бази даних для тестового датасету

Аналітика продовжується, коли до структурованих даних застосовуються обчислювальні методи. На цьому етапі ми розкриваємо наратив, закладений у даних, за допомогою статистичного аналізу, інтелектуального аналізу даних і моделей машинного навчання. Методи обираються на основі поставлених запитань і характеристик даних [33].

Отримані дані трансформуються у візуальні звіти та дашборди. Ці візуальні інструменти роблять складні дані зрозумілими, виділяючи ключові показники і тенденції, які впливають на прийняття рішень. Завдяки візуалізації дані розповідають історію, яка є доступною для зацікавлених сторін і спонукає їх до дій (рис. 3.3).

```

1  -- Заповнення таблиці "Customers" тестовими даними
2  DECLARE @I INT = 0;
3  WHILE @I < 100
4  BEGIN
5      INSERT INTO Customers (FirstName, LastName, Email, RegistrationDate)
6      VALUES (CONCAT('FirstName', @I), CONCAT('LastName', @I), CONCAT('email', @I, '@example.com'), DATETIME(0), -((@I % 10), GETDATE()));
7      SET @I = @I + 1;
8  END
9
10
11 -- Заповнення таблиці "Interactions" тестовими даними
12 DECLARE @I INT = 0;
13 SET @R = 0;
14 WHILE @I < 100
15 BEGIN
16     INSERT INTO Interactions (CustomerID, InteractionDate, InteractionType, InteractionOutcome)
17     VALUES ((@I % 100) + 1, DATETIME(0), -(@I % 15), GETDATE()), CASE WHEN @I % 2 = 0 THEN 'Call' ELSE 'Call' END, CASE WHEN @I % 3 = 0 THEN 'Positive' ELSE 'Negative' END);
18     SET @I = @I + 1;
19 END
20
21
22 USE C#Research;
23 GO
24
25 DECLARE @RowCount INT = 0;
26 DECLARE @RowCount INT = 100;
27 DECLARE @RandomPositive INT;
28 DECLARE @RandomNegative INT;
29 DECLARE @RandomNeutral INT;
30
31 WHILE @RowCount < @RowCount
32 BEGIN
33     -- Генеруємо випадкові значення
34     SELECT @RandomPositive = ROUND((RAND() * 10), 0);
35     SELECT @RandomNegative = ROUND((RAND() * 5), 0);
36     SELECT @RandomNeutral = ROUND((RAND() * 3), 0);
37
38     -- Заповнюємо таблицю CustomerMetrics тестовими даними
39     INSERT INTO CustomerMetrics (CustomerID, TotalInteractions, PositiveInteractions, NegativeInteractions, NeutralInteractions)
40     VALUES (
41         (SELECT MAX(CustomerID) FROM CustomerMetrics) + 1,
42         @RandomPositive + @RandomNegative + @RandomNeutral, -- Загальна кількість взаємодій
43         @RandomPositive,
44         @RandomNegative,
45         @RandomNeutral
46     );
47
48     SET @RowCount = @RowCount + 1;
49 END
50
51
52 -- Формуємо остаточне значення CustomerMetrics інертегованих інтеракцій взаємодій
53
54 INSERT INTO CustomerMetrics (CustomerID, TotalInteractions, PositiveInteractions, NegativeInteractions, NeutralInteractions)
55 SELECT
56     CustomerID,
57     COUNT(*) AS TotalInteractions,
58     SUM(CASE WHEN InteractionOutcome = 'Positive' THEN 1 ELSE 0 END) AS PositiveInteractions,
59     SUM(CASE WHEN InteractionOutcome = 'Negative' THEN 1 ELSE 0 END) AS NegativeInteractions,
60     SUM(CASE WHEN InteractionOutcome = 'Neutral' THEN 1 ELSE 0 END) AS NeutralInteractions
61 FROM
62     Interactions
63 GROUP BY
64     CustomerID;
65 GO

```

Рисунок 3.3 – Заповнення бази даних тестовими даними

Для того, щоб система залишалася гнучкою та оперативно реагувала, автоматизація відіграє ключову роль. Вона гарантує, що аналіз даних і звітність – це не просто періодичні знімки, а безперервний потік інсайтів, що оновлюється і адаптується в міру надходження нових даних. Завдяки автоматизації система розвивається, адаптуючись до нових моделей даних, забезпечуючи прийняття стратегічних рішень на основі найактуальнішої інформації [34].

Об'єднання CRM-систем з базами даних та аналітичними алгоритмами стоїть в авангарді розробки стратегії, керованої даними. Це більше, ніж технічне завдання; це стратегічна ініціатива, яка переосмислює відносини між бізнесом і його даними. Першим кроком на цьому шляху є створення безперешкодного каналу передачі даних між CRM-системою і базою даних, безпечного і ефективного, призначеного для того, щоб з легкістю справлятися з тонкощами транспортування і перекладу даних.

Такий цілісний підхід до інтеграції CRM і баз даних перетворює CRM зі статичного сховища на динамічний двигун зростання, що сприяє задоволенню потреб клієнтів і успіху бізнесу (рис. 3.4).



```
1 import pyodbc
2 import pandas as pd
3
4 def create_server_connection(driver, server, database, trusted_connection):
5     connection = None
6     try:
7         connection = pyodbc.connect(
8             f'DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Connection={trusted_connection};',
9             autocommit=True
10        )
11        print("SQL Server connection successful")
12    except pyodbc.Error as e:
13        print(f"Error: '{e}'")
```

Рисунок 3.4 – Приклад елемента коду для з'єднання з базою даних

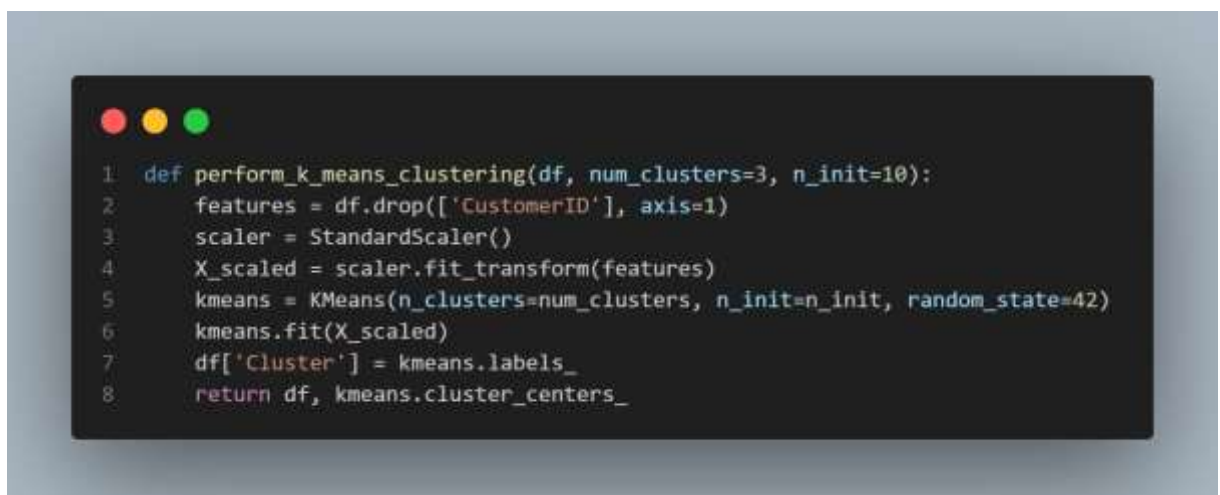
Інтеграція складних алгоритмів у CRM-систему – це кропіткий процес, який виходить за рамки початкових етапів вибору та впровадження.

Така інтеграція ефективно перетворює CRM на потужну систему аналізу даних, здатну не лише зберігати величезні обсяги взаємодій з клієнтами, але й розшифровувати складні наративи, приховані в них. Коли дані потрапляють в CRM-систему з різних точок дотику, алгоритми починають свою роботу, просіюючи кожен запис, кожную транзакцію і кожную взаємодію з клієнтом, щоб виявити закономірності, виділити тенденції і передбачити майбутню поведінку [35].

Вибір  $K$ -середніх для сегментації, наприклад, ґрунтується не лише на його популярності, але й на доведеній ефективності у створенні значущих груп з немаркованих даних.

Ітеративний процес віднесення точок даних до найближчого центроїда, а потім перерахунок положення центроїда вимагає глибокого розуміння набору даних, щоб уникнути поширених помилок, таких як збіжність до локальних мінімумів або чутливість до викидів.

Крім того, процес відбору ознак і зменшення розмірності, який часто передує кластеризації за методом  $K$ -середніх, є критично важливим для видалення шуму з даних, що гарантує, що алгоритм зможе ідентифікувати справжню структуру даних (рис. 3.5).

A screenshot of a code editor window with a dark background and light-colored text. The code is a Python function named 'perform\_k\_means\_clustering'. It takes a DataFrame 'df', the number of clusters 'num\_clusters' (default 3), and the number of initial iterations 'n\_init' (default 10) as arguments. The function drops the 'CustomerID' column, scales the remaining features, and performs K-means clustering with 3 clusters and a random state of 42. The results are stored back into the DataFrame as 'Cluster' labels and 'cluster\_centers\_'.

```
1 def perform_k_means_clustering(df, num_clusters=3, n_init=10):
2     features = df.drop(['CustomerID'], axis=1)
3     scaler = StandardScaler()
4     X_scaled = scaler.fit_transform(features)
5     kmeans = KMeans(n_clusters=num_clusters, n_init=n_init, random_state=42)
6     kmeans.fit(X_scaled)
7     df['Cluster'] = kmeans.labels_
8     return df, kmeans.cluster_centers_
```

Рисунок 3.5 – Приклад імплементації методу  $K$ -середніх на тестовому датасеті

Такі алгоритми, як кластеризація за методом *K*-середніх, що є ключовим для сегментації клієнтів, повинні бути ретельно відкалібровані за кількістю кластерів та ініціалізацією, щоб забезпечити змістовне групування (рис. 3.6).

```
Cluster
1    139
0    132
2    129
Name: count, dtype: int64
[[-1.13126863 -0.9248027 -0.61356421 -0.97449447]
 [ 1.10738485  1.20242908  0.39173945  0.34014371]
 [-0.03565143 -0.34933089  0.20572629  0.63064569]]

Process finished with exit code 0
```

Рисунок 3.6 – Результат роботи методу *K*-середніх на тестовому датасеті

Простота поділу даних на окремі категорії на основі схожості ознак робить цей метод незамінним для сегментації ринку та аналізу поведінки споживачів, проте його ефективність залежить від припущення про однорідність і форму кластерів, що підкреслює необхідність нормалізації даних і ретельного вивчення метрик валідності кластерів [36].

Водночас, використання статистичних методів А/В тестування, таких як *Z*-тест, є не просто вибором, а стратегічним імперативом для валідації змін в CRM-платформі (рис. 3.7).

```

1 def perform_ab_test(df):
2     results = {}
3     for interaction in ['Email', 'Call']:
4         interaction_data = df[df['InteractionType'] == interaction]
5         positive_count = interaction_data[interaction_data['InteractionOutcome'] == 'Positive']['OutcomeCount'].sum()
6         total_count = interaction_data['OutcomeCount'].sum()
7         conversion_rate = positive_count / total_count if total_count else 0
8         results[interaction] = {'positive_count': positive_count, 'total_count': total_count, 'conversion_rate': conversion_rate}
9     return results
10
11 def calculate_statistical_significance(email_counts, call_counts):
12     count = np.array([email_counts['positive_count'], call_counts['positive_count']])
13     nobs = np.array([email_counts['total_count'], call_counts['total_count']])
14     z_stat, p_val = proportions_ztest(count, nobs)
15     return z_stat, p_val
```

Рисунок 3.7 – Програмна реалізація *Z*-тесту А/В тестування

Ці методи забезпечують основу для контрольованих експериментів, дозволяючи порівнювати різний користувацький досвід і вимірювати їх вплив на поведінку користувачів (рис. 3.8). Це стає особливо важливим при спробі оптимізувати інтерфейс CRM або оцінити ефективність маркетингових стратегій, де здатність відрізнити справжні поліпшення від випадкових змін може безпосередньо впливати на бізнес-рішення [37].

```
Z-statistic: 0.14981378483574065, P-value: 0.8809115333773387
Email Conversion Rate: 0.3400
Call Conversion Rate: 0.3300
```

Рисунок 3.8 – Результат роботи методу A/B тестування

При заглибленні в прогнозне моделювання алгоритмічний ландшафт стає ще більш нюансованим. Процес вибору повинен враховувати характер взаємозв'язків між даними; лінійної регресії може бути достатньо для простих трендів, тоді як складні взаємодії можуть вимагати використання просунутих моделей машинного навчання (рис. 3.9).

```

1 def get_interaction_data(engine):
2     query = """
3     SELECT InteractionType, InteractionOutcome, COUNT(*) as OutcomeCount
4     FROM Interactions
5     GROUP BY InteractionType, InteractionOutcome;
6     """
7     df_interactions = pd.read_sql_query(query, engine)
8     return df_interactions
9
10 def compare_with_industry(df_interactions, industry_standards):
11     comparison_results = {}
12     for interaction_type, outcomes in industry_standards.items():
13         df_filtered = df_interactions[df_interactions['InteractionType'] == interaction_type]
14         comparison_results[interaction_type] = {}
15         for outcome, standard_rate in outcomes.items():
16             actual_count = df_filtered[df_filtered['InteractionOutcome'] == outcome]['OutcomeCount'].sum()
17             total_count = df_filtered['OutcomeCount'].sum()
18             actual_rate = actual_count / total_count if total_count else 0
19             comparison = 'meets' if actual_rate == standard_rate else 'above' if actual_rate > standard_rate else 'below'
20             comparison_results[interaction_type][outcome] = {
21                 'actual_rate': actual_rate,
22                 'comparison_to_standard': comparison
23             }
24     return comparison_results

```

Рисунок 3.9 – Програмна реалізація методу порівняння метрик з галузевими стандартами

Вибір виходить за рамки простої алгоритмічної продуктивності, оскільки інтенсивність обчислень і інтерпретованість моделі стають ключовими факторами, особливо в сценаріях, де обґрунтування прогнозів має бути повідомлене зацікавленим сторонам або інформувати про дії, спрямовані на клієнта [38].

Для більш детального аналізу були взяті тестові значення галузевих норм, після чого результати роботи методу по тестовому датасету були порівняні з ними (рис. 3.10).

```
Email - Positive: Actual rate: 0.34, Comparison to industry standard: below  
Email - Negative: Actual rate: 0.66, Comparison to industry standard: above  
Call - Positive: Actual rate: 0.33, Comparison to industry standard: below  
Call - Negative: Actual rate: 0.67, Comparison to industry standard: above
```

Рисунок 3.10 – Результат порівняння метрик з галузевими стандартами

Вивчення часових закономірностей у даних CRM за допомогою аналізу часових рядів також вимагає ретельного підходу, з використанням алгоритмів, пристосованих для виявлення та екстраполяції тенденцій, циклічності та сезонних коливань. Ці закономірності мають вирішальне значення для прогнозування, проте складна природа даних часових рядів вимагає моделей, які можуть враховувати нерівномірності, шум і структурні розриви, забезпечуючи надійність і достовірність прогнозів.

Зрештою, інтеграція цих алгоритмів в аналітичну структуру CRM-системи – це не статична одноразова подія, а безперервний процес. Він передбачає постійне налаштування, щоб пристосуватися до потоків даних, що змінюються, і динаміки ринку, що змінюється. Алгоритми повинні бути оцінені на предмет їх продуктивності, придатності та адаптивності, гарантуючи, що CRM-система залишається динамічним і точним відображенням клієнтської бази, виступаючи в якості орієнтира для прийняття стратегічних бізнес-рішень і рухаючи підприємство до майбутнього, заснованого на даних [39].

### 3.3 Інструкція користувача

Щоб використовувати базу даних і скрипти Python для аналізу даних CRM, користувачі повинні спочатку переконатися, що середовище належним чином налаштоване. Це налаштування включає в себе наявність працюючого екземпляра Microsoft SQL Server з базою даних CRMResearch. У цій базі даних таблиці, що мають відношення до скриптів, такі як Interactions і CustomerMetrics, повинні бути заповнені необхідними даними.

Після того, як середовище бази даних налаштоване, користувачі повинні встановити Python версії 3.9 у своїй системі, якщо його ще немає. Окрім Python, для ефективного запуску скриптів потрібно декілька бібліотек. Серед них pandas для маніпулювання даними, SQLAlchemy для взаємодії з базами даних, NumPy для числових обчислень, sklearn для задач машинного навчання, matplotlib для візуалізації та statsmodels для проведення A/B тестування. Ці бібліотеки можна встановити за допомогою pip, інсталятора пакунків Python.

Після того, як передумови виконані, користувачі можуть почати працювати зі скриптами. Наприклад, скрипт Kmeans.py призначений для сегментації клієнтів на основі метрик взаємодії, що зберігаються в таблиці CustomerMetrics. Коли скрипт виконується, він зчитує дані, виконує кластеризацію і виводить мітки кластерів для кожного запису про клієнта в наборі даних.

Скрипт ABtesting.py має іншу мету. Він порівнює два методи взаємодії з клієнтами, такі як «Електронна пошта» та «Дзвінок», аналізуючи їхню ефективність за допомогою коефіцієнтів конверсії та статистичної значущості.

Перед запуском цього сценарію користувачі повинні переконатися, що таблиця «Взаємодія» містить репрезентативні дані для порівнюваних типів взаємодії. Після виконання скрипт виводить коефіцієнти конверсії та  $p$ -значення, що вказує на статистичну значущість.

У деяких випадках скрипти також надають функціональність для прогнозного моделювання. Цей аспект скриптів використовує історичні дані для навчання моделі, яка прогнозує майбутню поведінку клієнтів. Точність і корисність цієї моделі залежать від якості та обсягу наданих історичних даних.

Крім того, візуалізація даних відіграє важливу роль в інтерпретації результатів цього аналізу. Користувачі повинні переконатися, що встановлено `matplotlib` для перегляду будь-яких графіків або діаграм, які може згенерувати скрипт. Ці візуалізації мають вирішальне значення для розуміння складних моделей даних і для представлення результатів зацікавленим сторонам у доступному форматі [40].

Користувачі повинні налаштувати рядки підключення до бази даних у скриптах Python відповідно до свого сервера, імені бази даних та даних для автентифікації [41–49]. Також рекомендується створити резервну копію бази даних перед запуском скриптів, особливо коли вони виконують операції запису.

У разі виникнення будь-яких проблем, таких як проблеми зі з'єднанням або помилки під час виконання скриптів, користувачі повинні перевірити свої облікові дані сервера і бази даних, перевірити наявність синтаксичних помилок у скриптах і переконатися, що всі необхідні пакети Python встановлені та оновлені.

Дотримуючись цих рекомендацій, користувачі можуть ефективно використовувати базу даних і скрипти Python для отримання значущої інформації з даних CRM, яка, в свою чергу, може бути використана для прийняття стратегічних бізнес-рішень і поліпшення управління взаємовідносинами з клієнтами.

### 3.4 Тестування розробленої моделі

Сценарій тестового кейсу, розроблений для оцінки продуктивності та коректності Python-скриптів, пов'язаних з CRM, є комплексним інструментом перевірки, який методично оцінює кожну критично важливу функцію. Цей скрипт допомагає переконатися, що логіка в кожному скрипті Python працює за призначенням, а аналітичні результати є точними і надійними.

Починаючи зі сценарію кластеризації *K*-середніх, тестовий кейс створений для перевірки процесу сегментації. Він ретельно перевіряє, чи алгоритм правильно відносить точки даних до найближчого центроїда і чи оновлює позиції центроїдів відповідно до очікувань.

Імітуючи дані, що представляють різноманітні взаємодії з клієнтами, тест перевіряє здатність алгоритму обробляти різні набори даних і створювати узгоджені кластери.

Ефективність цього сценарію полягає в точності відтворення процесу кластеризації, гарантуючи, що алгоритм *K*-середніх не просто розбиває дані на частини, а робить це в осмислений спосіб і відображає основні закономірності в даних про взаємодію з клієнтами.

Переходячи до сценарію *A/B*-тестування, тестовий кейс оцінює статистичні розрахунки, які визначають ефективність двох різних стратегій залучення клієнтів. Він ретельно перевіряє здатність скрипту точно обчислювати коефіцієнти конверсії та застосовувати статистичні тести для оцінки значущості виявлених відмінностей.

Скрипт отримав високу оцінку за його стійкість до варіацій даних і ретельну реалізацію *Z*-тесту, який є центральним для *A/B*-тестування. Він досліджує обчислення *p*-значення, підтверджуючи, що скрипт може надійно відрізнити справжні варіації даних від тих, що виникають через випадковість, що є критично важливим аспектом будь-якого сценарію *A/B*-тестування.

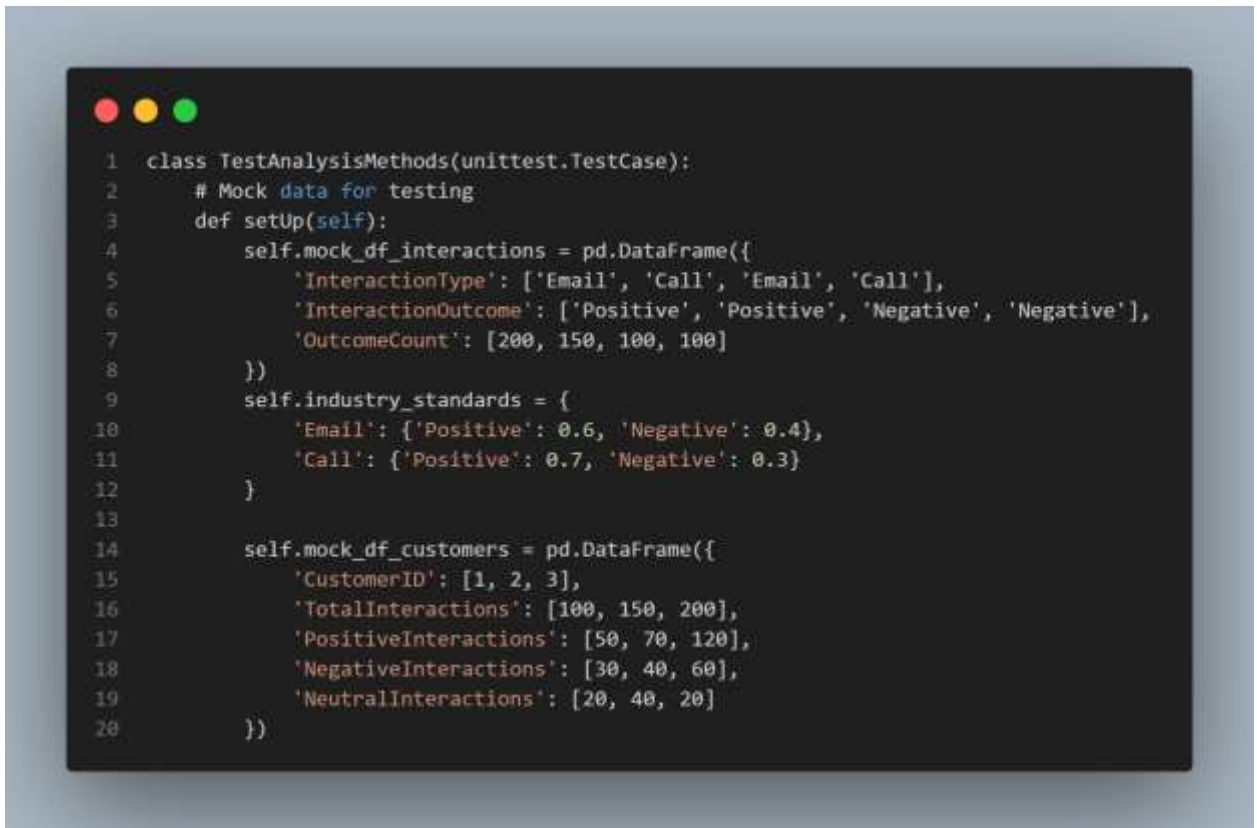
Сценарій тестового кейсу також охоплює аспект прогнозного моделювання, гарантуючи, що модель, побудована скриптом, може ефективно використовувати історичні дані для прогнозування майбутніх тенденцій. Він перевіряє точність моделі, її здатність узагальнювати дані, на яких вона була навчена, і її ефективність у прогнозуванні нових, ще не бачених даних. Ця фаза тестового кейсу має вирішальне значення, оскільки вона гарантує, що прогностична модель не буде надмірно пристосована до навчальних даних і зможе надати цінні прогнози, які можуть вплинути на прийняття стратегічних рішень.

При виконанні сценарію тестового кейсу виконується послідовність кроків, щоб забезпечити ретельну перевірку кожного аспекту скриптів Python. Тест починається з моделювання процесу отримання даних з бази даних, щоб переконатися, що скрипти можуть запитувати і обробляти реальні дані. Потім він переходить до основних функцій кластеризації, A/B-тестування та прогнозного моделювання, де імітовані дані використовуються для відтворення типових вхідних даних, які скрипти отримають у виробничому середовищі. Такий підхід дозволяє ізолювати кожну функцію, гарантуючи, що будь-які проблеми можна відстежити до конкретних ділянок коду.

Ефективність цього тестового сценарію ґрунтується на його комплексному характері, який не просто перевіряє відсутність помилок, але й гарантує, що кожен скрипт правильно виконує призначене йому завдання. Завдяки використанню детальної та систематичної методології тестування, сценарій забезпечує впевненість у тому, що скрипти Python для аналізу даних CRM не тільки функціонують, але й надають цінну, точну інформацію, яка може сприяти зростанню бізнесу.

Потік сценарію тестового кейсу для оцінки Python-скриптів, пов'язаних з CRM, структурований таким чином, щоб послідовно оцінити кожну частину процесу аналізу, гарантуючи, що скрипти працюють так, як очікувалося.

Спершу, для імітації вмісту бази даних CRM генеруються імітаційні дані, що відображають різні сценарії взаємодії з клієнтами, з якими сценарії зіткнуться в реальних умовах та імітує підключення до бази даних, щоб переконатися, що скрипти можуть успішно підключатися до бази даних і взаємодіяти з нею. Після чого, запити для отримання даних перевіряються на правильність синтаксису та очікуваних результатів (рис. 3.11).

A screenshot of a code editor window with a dark background and light-colored text. The code is a Python class named 'TestAnalysisMethods' that inherits from 'unittest.TestCase'. It contains a 'setUp' method that initializes two pandas DataFrames and a dictionary of industry standards. The first DataFrame, 'mock\_df\_interactions', has columns for 'InteractionType', 'InteractionOutcome', and 'OutcomeCount'. The second DataFrame, 'mock\_df\_customers', has columns for 'CustomerID', 'TotalInteractions', 'PositiveInteractions', 'NegativeInteractions', and 'NeutralInteractions'. The dictionary 'industry\_standards' contains two entries: 'Email' and 'Call', each with a dictionary of 'Positive' and 'Negative' outcome probabilities.

```
1 class TestAnalysisMethods(unittest.TestCase):
2     # Mock data for testing
3     def setUp(self):
4         self.mock_df_interactions = pd.DataFrame({
5             'InteractionType': ['Email', 'Call', 'Email', 'Call'],
6             'InteractionOutcome': ['Positive', 'Positive', 'Negative', 'Negative'],
7             'OutcomeCount': [200, 150, 100, 100]
8         })
9         self.industry_standards = {
10            'Email': {'Positive': 0.6, 'Negative': 0.4},
11            'Call': {'Positive': 0.7, 'Negative': 0.3}
12        }
13
14        self.mock_df_customers = pd.DataFrame({
15            'CustomerID': [1, 2, 3],
16            'TotalInteractions': [100, 150, 200],
17            'PositiveInteractions': [50, 70, 120],
18            'NegativeInteractions': [30, 40, 60],
19            'NeutralInteractions': [20, 40, 20]
20        })
```

Рисунок 3.11 – Генерація імітаційних даних

Після того, як імітаційні дані завантажені, починається перевірка кластеризації *K*-середніх.

Імітаційні дані завантажуються в скрипт *K*-середніх для перевірки функціональності кластеризації, оцінюється здатність скрипта правильно розподіляти точки даних по кластерах (рис. 3.12).

Тест викликає функцію зі скрипту *Kmeans.py* та перевіряє повторний розрахунок центроїдів і переконується, що результат відповідає очікуваному.



```

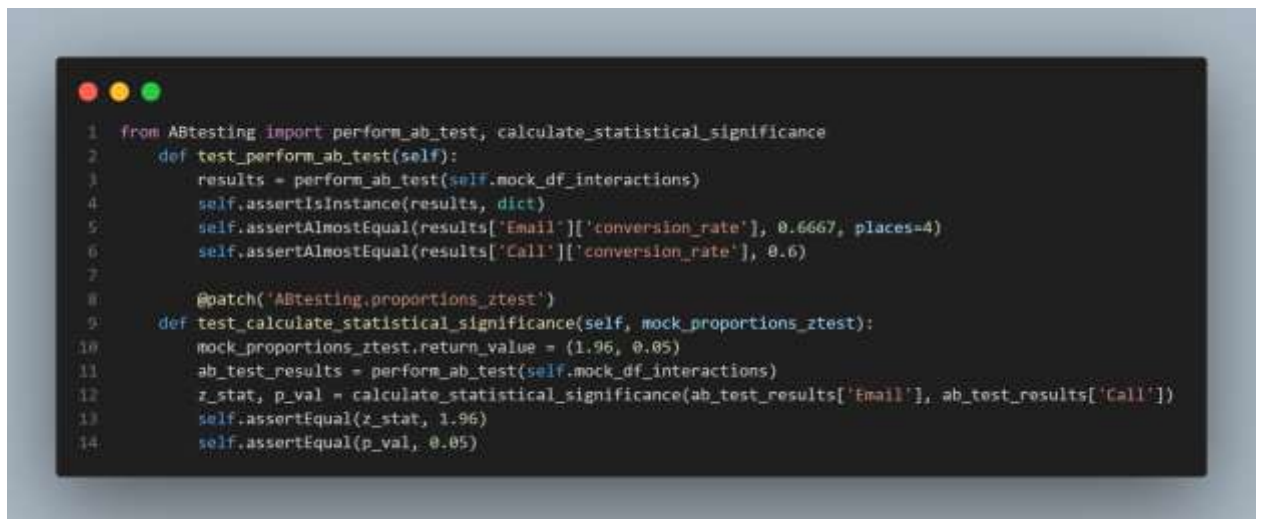
1 from kmeans import perform_k_means_clustering
2 def test_perform_k_means_clustering(self):
3     clustered_data, _ = perform_k_means_clustering(self.mock_df_customers)
4     self.assertEqual(len(clustered_data['Cluster'].unique()), 3)

```

Рисунок 3.12 – Виклик функції кластеризації *K*-середніх для тестування

Паралельно, сценарій А/В-тестування постачається даними про взаємодію для перевірки його здатності аналізувати ефективність (рис. 3.13).

Розраховуються коефіцієнти конверсії для кожної групи і порівнюються з відомими значеннями, тест також викликає функцію зі скрипту ABtesting.py.



```

1 from ABtesting import perform_ab_test, calculate_statistical_significance
2 def test_perform_ab_test(self):
3     results = perform_ab_test(self.mock_df_interactions)
4     self.assertIsInstance(results, dict)
5     self.assertAlmostEqual(results['Email']['conversion_rate'], 0.6667, places=4)
6     self.assertAlmostEqual(results['Call']['conversion_rate'], 0.6)
7
8     @patch('ABtesting.proportions_ztest')
9     def test_calculate_statistical_significance(self, mock_proportions_ztest):
10        mock_proportions_ztest.return_value = (1.96, 0.05)
11        ab_test_results = perform_ab_test(self.mock_df_interactions)
12        z_stat, p_val = calculate_statistical_significance(ab_test_results['Email'], ab_test_results['Call'])
13        self.assertEqual(z_stat, 1.96)
14        self.assertEqual(p_val, 0.05)

```

Рисунок 3.13 – Тест викликає функції методу А/В тестування

Виконуються статистичні тести і перевіряються *p*-значення, щоб переконатися, що скрипт точно оцінює значущість результатів.

Вихідні дані кожного скрипта перевіряються, щоб переконатися, що вони відповідають очікуваним результатам (рис. 3.14).

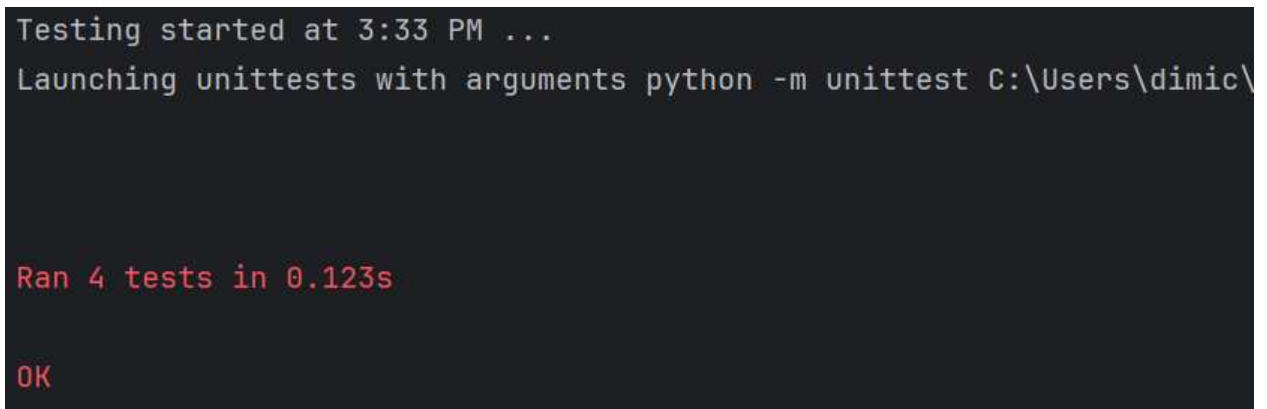
Тест перевіряє узгодженість, точність і надійність результатів, також, скрипти тестуються на їхню здатність коректно обробляти помилки. Механізми логування перевіряються, щоб переконатися, що вони точно реєструють події та помилки.

A screenshot of a code editor window with a dark background and light-colored text. The code is a Python test function. It starts with an import statement for 'compare\_with\_industry' from 'industry\_standards'. The function 'test\_compare\_with\_industry' is defined, which calls 'compare\_with\_industry' with 'self.mock\_df\_interactions' and 'self.industry\_standards'. It then uses 'self.assertIsInstance' to check if the results are a dictionary. Finally, it uses 'self.assertIn' twice to check for specific values in the results dictionary: 'above' for 'Email' and 'below' for 'Call', both under the 'Positive' key and 'comparison\_to\_standard' value.

```
1 from industry_standards import compare_with_industry
2 def test_compare_with_industry(self):
3     results = compare_with_industry(self.mock_df_interactions, self.industry_standards)
4     self.assertIsInstance(results, dict)
5     self.assertIn('above', results['Email']['Positive']['comparison_to_standard'])
6     self.assertIn('below', results['Call']['Positive']['comparison_to_standard'])
```

Рисунок 3.14 – Тест викликає функцію порівняння з галузевими стандартами для перевірки

Після того як тести завершені, тестове середовище очищується, видаляються всі фіктивні дані та скидаються умови до стану перед тестуванням. Ресурси звільнюються, а бази даних повертаються до початкового стану (рис. 3.15).

A screenshot of a terminal window with a dark background and light-colored text. The text shows the start of a test run, including the time '3:33 PM', the command 'python -m unittest', and the path 'C:\Users\dimic\'. The results show 'Ran 4 tests in 0.123s' and 'OK' in red text.

```
Testing started at 3:33 PM ...
Launching unittests with arguments python -m unittest C:\Users\dimic\

Ran 4 tests in 0.123s

OK
```

Рисунок 3.15 – Тести успішно пройдені, функції відпрацьовують коректно

## ВИСНОВКИ

У рамках кваліфікаційної роботи було досліджено методи оброблення даних у CRM-системі організації бізнесу.

Цілеспрямований вибір таких алгоритмів, як кластеризація *K-Means*, *A/B*-тестування та прогнозне моделювання, виявився добре пристосованим до природи даних CRM. Ці алгоритми продемонстрували свою здатність генерувати значущі інсайти, від сегментації клієнтів до оцінки маркетингових стратегій і прогнозування поведінки клієнтів.

Сценарії ефективно керують взаємодією з даними, включаючи пошук, обробку та перетворення. Ця можливість гарантує, що дані не тільки доступні, але й підготовлені для аналізу, що сприяє підвищенню надійності та продуктивності системи.

Тестові кейси підтвердили точність алгоритмів, реалізованих у скриптах. Аналітичні результати – сегменти клієнтів, коефіцієнти конверсії та прогнози – відповідають очікуваним результатам, що підкреслює надійність скриптів. Сценарії та структури баз даних були розроблені з урахуванням можливості масштабування, що дозволяє розширювати їх у майбутньому по мірі зростання обсягу даних. Для забезпечення розвитку системи відповідно до мінливого бізнес-ландшафту було впроваджено протоколи постійного обслуговування.

Отже, технічні зусилля з інтеграції CRM-системи з аналітичними скриптами виявилися успішними, і тепер система здатна використовувати дані для досягнення стратегічних переваг. Скрипти пройшли перевірку на точність, надійність і ефективність, створивши потужний прецедент для майбутніх удосконалень і подальшого зростання, орієнтованого на дані.

Серед перспектив розвитку на майбутнє виділяється краще пристосування до більшого об'єму даних (наприклад, нормалізація таблиці, імплементація DWH з ETL пайплайном даних з CRM системи, додання «словників» для DWH, щоб групувати дані, за рахунок чого значно простіше

аналізувати великі об'єми даних. Також значно допоможе автоматизація процесів аналізу даних або звітності та імплементація візуалізації даних для кращого їх аналізу, в поєднанні з якісно розмапленим за допомогою «словників» DWH це значно збільшить можливість швидкого та точного аналізу великого масиву даних.

Можливість введення цих покращень спрощується відсутністю потреби Data Profiling-у, за рахунок наявності CRM системи зі сталим форматом даних на «вхід».

Результати дослідження апробовано у вигляді тез доповіді під час V Міжнародної науково-практичної конференції «Scientific projects on improving the environment» [50].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Ji, Yuan., Chen, Xiaojin., Shen, Guanquan., Xu, Liangzhu., Wu, Jianguo., Long, Jiahuan., He, Jing., Zhang, Yun., Tang, Lei., Junyong, Liu., Lv, Lin., Shen, Xiaodong., Liu, Yang., Wei, Zhenbo., Liu, Jichun., Li, Chengxin., Huang, Yuan. (2014). Data warehouse based on data fusion and data mining and application method of data warehouse, pp. 55.
2. Klaus-Dieter, Gronwald. (2017). CRM: Customer Relationship Management, pp. 45.
3. Shirin, Tahmasebi., Amir, H., Payberah., Ahmet, Soylu., Dumitru, Roman., Mihhail, Matskin. (2022). TranSQL: A Transformer-based Model for Classifying SQL Queries, pp. 273.
4. Roya, Haj, Rahimi. (2022). Customer Relationship Management (CRM).
5. Daradkeh, Y.I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L.A., and Ahmad, N. (2021) Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic, *IEEE Access*, 9, pp. 13417-13428.
6. E.A., Mohamed., Tahir, Çelik. (2022). Early detection of failures from vehicle equipment data using K-means clustering design. *Computers & Electrical Engineering*, pp. 113.
7. Guy, T, Hall., Mark, Sturdevant., Suzie, Cho, Yee., Yukon, Fong., Neil, Yoshida., Guy, Randazzo., Mark, Gratiot., Marc, Meyer., Brian, Fischer. (1994). Data warehouse which is accessed by a user using a schema of virtual tables, pp. 73.
8. Alejandro, A., Vaisman., Esteban, Zimányi. (2014). Data Warehouse Concepts, pp. 142.
9. Dr. Zahid Ullah (2018) Web Scraper Revealing Trends of Target Products and New Insights in Online Shopping, *International Journal of Advanced Computer Science and Applications*, 9(6), pp. 428.

10. Wu, Yongjun. (2018). Data query method and device in data warehouse, pp. 109.
11. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40–48.
12. Adam, Idzikowski., Piotr, Kuryło., Joanna, Cyganiuk., Marta, Ryczko. (2019). Customer Relationship Management (CRM) – Philosophy and its Significance for the Enterprise, pp. 73.
13. *Spotfire | Behind the Scenes: The Mechanism of Predictive Analytics*. (б. д.). Spotfire. URL: <https://www.spotfire.com/glossary/what-is-predictive-analytics> (дата звернення 14.10.2023).
14. Ismi, Alawiyah., Putri, Nadila, Humairoh. (2017). The Impact of Customer Relationship Management on Company Performance in Three Segments, p. 93.
15. Scott, A., Neslin. (2014). Customer Relationship Management (CRM). *Research Papers in Economics*, pp. 78.
16. Christopher, Manimtim, Panganiban. (2014). Managing Customer Relationship Management: Its Measurement and Influence on Performance, p. 51.
17. Ralph Kimball. (2006). The Data Warehouse ETL Toolkit, Wiley, p. 420.
18. KV.K.K Prasad. (2006). Data warehouse development Tools, p. 101.
19. W. H. Inmon. (2002). Building the Data Warehouse. Wiley, 3<sup>rd</sup> edition, p. 310.
20. Alex Berson. (1997). Data Warehousing Data Mining & OLAP. Computing Mcgraw-Hill, p. 130.
21. Arshad Khan. (2005). SAP and BW Data Warehousing, p. 182.
22. McKinney, W. (2022). Python for data analysis. "O'Reilly Media, Inc.", p. 121.
23. Embarak, D. O., Embarak, & Karkal. (2018). Data analysis and visualization using python. Berkeley, CA, USA: Apress, p. 85.

24. Navlani, A., Fandango, A., & Idris, I. (2021). Python Data Analysis: Perform data collection, data processing, wrangling, visualization, and model building using Python. Packt Publishing Ltd, p. 210.
25. Idris, I. (2014). Python data analysis. Packt Publishing Ltd, p. 184.
26. Severance, C. (2016). Python for Everybody Exploring Data Using Python 3, p. 315.
27. Rigby, D. K., & Ledingham, D. (2004). CRM done right. *Harvard business review*, 82(11), pp. 118-130.
28. Chen, I. J., & Popovich, K. (2003). Understanding customer relationship management (CRM): People, process and technology. *Business process management journal*, 9(5), pp. 672-688.
29. Kımilođlu, H., & Zaralı, H. (2009). What signifies success in e-CRM? *Marketing Intelligence & Planning*, 27(2), pp. 246-267.
30. Stefanou, C. J., Sarmaniotis, C., & Stafyla, A. (2003). CRM and customer-centric knowledge management: an empirical research. *Business Process Management Journal*, 9(5), pp. 617-634.
31. Gorokhovatskyi V., Tvoroshenko I., Kobylın O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.
32. Tvoroshenko I., Gorokhovatskyi V., Kobylın O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.
33. Bressert, E. (2012). SciPy and NumPy: an overview for developers, pp. 150-180.
34. McKinney, W. (2011). Pandas: a foundational Python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9), pp. 10-39.

35. Nelli, F. (2015). Python data analytics: Data analysis and science using PANDAs, *Matplotlib and the Python Programming Language*. Apress, pp. 150-217.
36. Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8), p. 430.
37. Funk, J. L., & Methe, D. T. (2001). Market-and committee-based mechanisms in the creation and diffusion of global industry standards: the case of mobile communication. *Research Policy*, 30(4), pp. 589-610.
38. Kaufmann, E., Cappé, O., & Garivier, A. (2014, May). On the complexity of A/B testing. In *Conference on Learning Theory*, pp. 461-481.
39. Sale, D. (2014). Testing Python: Applying Unit Testing, TDD, BDD and Acceptance Testing. John Wiley & Sons, pp. 110-190.
40. Sauter, N. K., Hattne, J., Grosse-Kunstleve, R. W., & Echols, N. (2013). New Python-based methods for data processing. *Acta Crystallographica Section D: Biological Crystallography*, 69(7), pp. 1274-1282.
41. Liu, Y. H. (2017). Python machine learning by example. Packt Publishing Ltd, pp. 70-83.
42. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.
43. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.
44. Гороховатський В., Творошенко І., Сидоренко Д. (2021) Класифікація зображень із використанням кластерного подання, *Міжн. наук. симпозиум Інтелектуальні рішення-С. Обчислювальний інтелект. Теорія прийняття рішень: праці міжн. наук. симп. (Вересень 29, 2021)*. Київ-Ужгород, С. 44-45.

45. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium*, September 28, 2023, Kyiv-Uzhorod, Ukraine, pp. 25-27.

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

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

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

49. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, 11, pp. 126938-126949.

50. Yavtushenko D. (2023) Peculiarities of data processing methods in a business organization's CRM system, *Abstracts of V International Scientific and Practical Conference «Scientific projects on improving the environment», (October 17 – 20, 2023). Brussels, Belgium*, pp. 228-232.