МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

Кацидим Віта Олександрівна

Інтелектуальний аналіз задоволеності клієнтів обслуговуванням / Intelligent analysis of customer satisfaction with service

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-42
Кацидим В.О.

Науковий керівник:
д.т.н., доцент
Ліпяніна-Гончаренко Х.В.

Кваліфікаційну роботу допущено до захисту
«___»_____ 2025 р.

В.о. завідувача кафедри
_____ Н.М. Васильків

Тернопіль – 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.М. Васильків
«____» _____2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Кацидим Віта Олександрівна
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Інтелектуальний аналіз задоволеності клієнтів обслуговуванням / Intelligent analysis of customer satisfaction with service керівник роботи к.т.н., доцент, Ліпяніна-Гончаренко Х.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 20 грудня 2024 р. № 938.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

– Проаналізувати існуючі підходи та інструменти оцінки задоволеності клієнтів.

– Розробити архітектуру модуля аналізу задоволеності клієнтів.

– Створити модель даних та схему бази даних для зберігання інформації про клієнтів та їхню взаємодію з компанією.

– Реалізувати алгоритми обробки та аналізу даних.

– Розробити дашборди для візуалізації результатів аналізу та підтримки прийняття рішень..

5. Перелік графічного матеріалу в роботі:

– Узагальнена структурна схема архітектури системи (Data Ingestion, Processing, Analytics, Visualization).

– Схема бази даних модуля аналізу задоволеності клієнтів..

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2025 р.	
	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2025 р.	
	Написання 2 розділу кваліфікаційної роботи	до 01.04.2025 р.	
	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2025 р.	
	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2025 р.	
	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2025 р.	
	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2025 р.	
	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2025 р.	
	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2025 р.	

Студент_____ Кацидим В.О.
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи_____ Ліпяніна-Гончаренко Х.В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Інтелектуальний аналіз задоволеності клієнтів обслуговуванням» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 82 сторінок і містить 28 ілюстрацій, 2 таблиці, 2 додатки та 26 використаних джерел.

Метою роботи є розробка програмного модуля інтелектуального аналізу задоволеності клієнтів, спрямованого на покращення якості обслуговування та лояльності клієнтів.

Методами розроблення обрано аналіз існуючих підходів, моделювання архітектури системи, розроблення алгоритмів обробки даних, а також методи статистичного аналізу та візуалізації результатів.

Внаслідок виконання роботи створено модуль аналізу даних, що дозволяє в режимі реального часу оцінювати рівень задоволеності клієнтів та формувати рекомендації щодо підвищення якості послуг.

Результати дослідження можуть бути впроваджені у процесах моніторингу та управління взаємовідносинами з клієнтами в комерційних організаціях.

Ключові слова: ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ, ЗАДОВОЛЕНІСТЬ КЛІЄНТІВ, АНАЛІЗ ДАНИХ, АЛГОРИТМИ ОБРОБКИ ДАНИХ, СТАТИСТИЧНИЙ АНАЛІЗ, ВІЗУАЛІЗАЦІЯ ДАНИХ, ДАШБОРДИ.

ANNOTATION

The bachelor's thesis entitled "Intelligent analysis of customer satisfaction with service" for obtaining the bachelor's degree in specialty 122 "Computer Science" of the educational program "Computer Science" consists of 82 pages and includes 28 illustrations, 2 tables, 2 appendices, and 26 references.

The aim of the work is to develop a software module for intelligent customer satisfaction analysis aimed at improving service quality and customer loyalty.

The selected development methods include analysis of existing approaches, system architecture modeling, data processing algorithm development, as well as statistical analysis methods and results visualization.

As a result, a data analysis module was created that enables real-time assessment of customer satisfaction levels and provides recommendations for improving service quality.

The research findings can be implemented in customer relationship management and monitoring processes within commercial organizations.

Keywords: INTELLIGENT ANALYSIS, CUSTOMER SATISFACTION, DATA ANALYSIS, DATA PROCESSING ALGORITHMS, STATISTICAL ANALYSIS, DATA VISUALIZATION, DASHBOARDS.

ЗМІСТ

ВСТУП

У сучасному конкурентному бізнес-середовищі задоволеність клієнтів є ключовим фактором успіху компаній. Задоволені клієнти не лише повторно звертаються за послугами, але й сприяють формуванню позитивного іміджу компанії через рекомендації, що безпосередньо впливає на її прибутковість та стійкість на ринку.

Традиційні методи оцінки задоволеності, такі як опитування та анкетування, мають низку обмежень, зокрема суб'єктивність відповідей, низький рівень відгуку та значні часові витрати на обробку даних. Це ускладнює отримання об'єктивної та оперативної інформації про рівень задоволеності клієнтів.

Інтеграція інтелектуальних систем аналізу задоволеності клієнтів у бізнес-процеси компанії сприяє підвищенню якості обслуговування, зниженню рівня відтоку клієнтів та збільшенню їхньої лояльності. Це, в свою чергу, забезпечує конкурентні переваги та сприяє довгостроковому розвитку компанії.

Мета роботи - розробити модуль інтелектуального аналізу задоволеності клієнтів обслуговуванням, для підвищення якості обслуговування та лояльності клієнтів.

Завдання роботи:

1. Проаналізувати існуючі підходи та інструменти оцінки задоволеності клієнтів.

2. Розробити архітектуру модуля аналізу задоволеності клієнтів.

3. Створити модель даних та схему бази даних для зберігання інформації про клієнтів та їхню взаємодію з компанією.

4. Реалізувати алгоритми обробки та аналізу даних.

5. Розробити дашборди для візуалізації результатів аналізу та підтримки прийняття рішень.

Предмет дослідження - методи та інструменти інтелектуального аналізу задоволеності клієнтів обслуговуванням.

Об'єкт дослідження - процеси оцінки та аналізу рівня задоволеності клієнтів у компаніях.

Методи дослідження. У процесі дослідження використовувалися методи системного аналізу для вивчення існуючих підходів до оцінки задоволеності клієнтів. Крім того, застосовувалися методи статистичного аналізу для оцінки результатів та візуалізації даних.

Апробація результатів дослідження здійснена в межах студентської науково-практичної конференції «Інтелектуальні інформаційні технології в прикладних дослідженнях» (IITAR-2025), яка відбулася в місті Тернополі 27–29 травня 2025 року.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ РІШЕНЬ

1.1 Аналіз поняття «задоволеність клієнтів»

Задоволеність клієнтів є одним із найважливіших показників успішності бізнесу в сучасних умовах конкурентного середовища. Вона визначає, наскільки запропоновані товари чи послуги відповідають або перевищують очікування споживачів. Задоволений клієнт частіше стає постійним, формує позитивну репутацію компанії та поширює про неї добрі відгуки.

Концепцію задоволеності розглядають у контексті психологічної реакції споживача на отриману послугу чи продукт. У науковій літературі задоволеність інтерпретують як результат порівняння фактичного досвіду користування із попередніми очікуваннями (або стандартами якості). Якщо досвід виявляється кращим чи на рівні очікувань, формується позитивне ставлення; якщо ж гіршим – виникає незадоволеність.

Згідно з класичними працями [1]. Задоволеність є постпридбаним феноменом, що впливає на повторні покупки і лояльність клієнта. З огляду на це, бізнеси інвестують значні ресурси в оцінювання та підтримку задоволеності споживачів.

Для вимірювання задоволеності можуть застосовуватися різноманітні методи: анкетування, опитування в режимі реального часу, відгуки в соціальних мережах. У цифрову епоху зростає роль онлайн-каналів (email, соціальні мережі, чати), де компанії мають можливість отримувати величезні обсяги даних про відгуки клієнтів.

Дослідники відзначають, що поняття «задоволеність» тісно пов’язане зі смисловими категоріями «відданість бренду» (brand loyalty) та «клієнтська лояльність» (customer loyalty). Позитивна оцінка сервісу, зазвичай, веде до підвищення частоти повторних покупок і скорочення часу ухвалення рішення, що позитивно впливає на показники доходу і маржинальності бізнесу.

У контексті B2B-взаємин задоволеність може мати додаткові параметри – надійність поставок, гнучкість в оплаті, рівень технічної підтримки. У B2C-

моделі важливу роль відіграє емоційна складова: чим більше позитивних емоцій отримує клієнт при взаємодії з компанією, тим вищою буде його оцінка.

На піраміді (рисунок 1.1) показано, на верхньому рівні піраміди потреб [2] лежить потреба у самовираженні, що в бізнес-контексті корелює з найвищим рівнем задоволеності. Якщо базові та проміжні рівні (фізіологічні потреби, безпека, належність, визнання) компанія закриває належним сервісом і підтримкою, клієнти відчувають «самореалізацію» у взаємодії з брендом. Це призводить до формування лояльності та довгострокових відносин.

Рисунок 1.1 – Піраміда впливу задоволеності на бізнес-результати [2]

Наукова цінність вивчення задоволеності полягає у розробці моделей прогнозування купівельної поведінки, визначенні факторів впливу (ціна, якість обслуговування, зручність доставки тощо) та розробці рекомендацій для підвищення якості послуг. З методологічного боку, дослідження проводять із залученням економетричних моделей, опитувань і кореляційного аналізу.

В інформаційну епоху питання задоволеності доповнюються такими категоріями, як швидкість та якість цифрових сервісів, рівень автоматизації процесів обслуговування. Компанії, що ігнорують задоволеність, ризикують втратити частку ринку через невдоволені відгуки клієнтів у соцмережах та на профільних онлайн-майданчиках.

На практиці для бізнесу важливо не лише збирати показники задоволеності, а й аналізувати тренди, визначати причини зниження чи

підвищення цього показника і вчасно реагувати. Таким чином забезпечується циклічна схема покращення: оцінка стану – аналіз – корегуючі дії – повторна оцінка.

З огляду на вищезазначене, задоволеність клієнтів у сучасному розумінні – це не просто показник, а складова стратегічної політики компанії. Вона впливає на репутацію, формує довготривалі відносини з клієнтами та визначає конкурентоспроможність на ринку.

1.2 Основні показники оцінки рівня задоволеності

Задоволеність клієнтів можна вимірювати різноманітними метриками, що дає змогу кількісно оцінити успішність взаємодії між компанією та споживачем. Найпоширенішими показниками в цій сфері є Customer Satisfaction (CSAT), Net Promoter Score (NPS) та Customer Effort Score (CES). Кожен із цих індексів має власну методологію розрахунку й зосереджується на різних аспектах клієнтського досвіду.

CSAT вважають одним із найстаріших і найпростіших показників у цій галузі. Його зазвичай розраховують, ставлячи клієнтам запитання на кшталт: «Наскільки ви задоволені отриманим обслуговуванням/продуктом за шкалою від 1 до 5?» Середня оцінка (або частка опитаних, які надали 4–5 балів) і є показником CSAT. Цей метод простий у реалізації та інтерпретації, тому його часто використовують у коротких анкетах чи після звернень до контакт-центру.

NPS концентрується на готовності клієнтів рекомендувати компанію чи продукт. Якщо респонденти відповідають на запитання: «Наскільки ймовірно, що ви порадите наш продукт чи послугу іншим?» за шкалою від 0 до 10, респонденти, які дали 9 або 10, вважаються «промоутерами» (promoters), 7 і 8 – «нейтральними» (passives), а 0–6 – «критиками» (detractors). Формула розрахунку: NPS = % промоутерів – % критиків. Значення може коливатися від –100 до +100. Високий NPS вказує на сильну лояльність і ймовірність подальших рекомендацій, що є критичним для органічного зростання бізнесу.

CES базується на ідеї вимірювання «зусилля» (effort), яке клієнт докладає, щоб отримати потрібне рішення, придбати товар або вирішити проблему. У деяких дослідженнях [4] встановили, що мінімізація зусиль клієнтів більш суттєво впливає на лояльність, ніж намагання компанії перевершити очікування. Таким чином, Customer Effort Score відображає швидкість і простоту взаємодії з компанією, зокрема кількість кроків, телефонних дзвінків чи звернень, необхідних для вирішення питання.

Рисунок 1.2 ілюструє, що кожен із трьох показників охоплює різні аспекти клієнтського досвіду: CSAT – емоційне задоволення, NPS – готовність рекомендувати, а CES – витрачене зусилля. Разом вони дозволяють формувати цілісну картину.

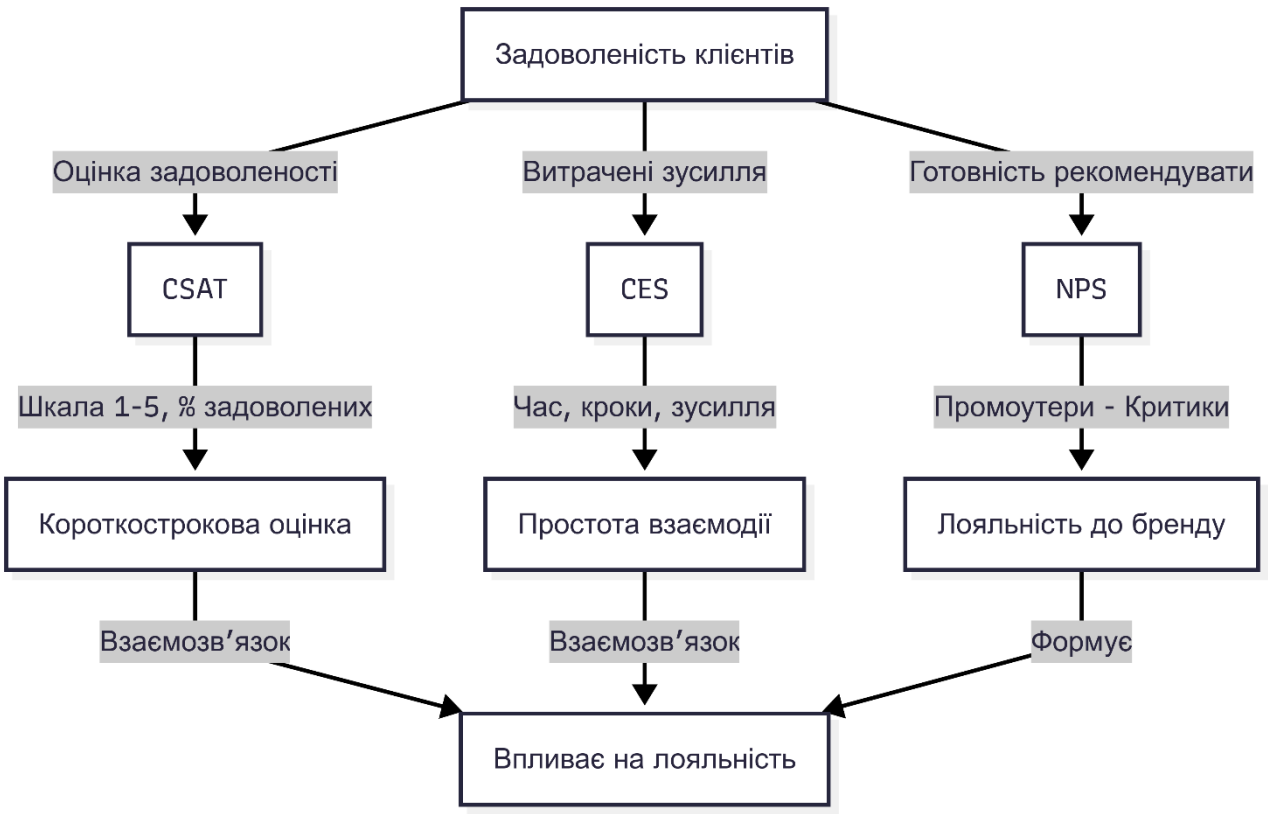

Рисунок 1.2 – Концептуальна схема взаємодії ключових показників (CSAT, NPS, CES)

Практичне застосування цих метрик різниться залежно від специфіки бізнесу. CSAT доречний, коли компанія прагне швидкого та простого фідбеку від

клієнтів щодо конкретної транзакції. NPS частіше використовують на рівні стратегічного маркетингу та бренд-менеджменту, щоб оцінити загальну лояльність споживачів і потенціал «сарафанного радіо». CES актуальний для служб підтримки й контакт-центрів, коли швидкість і легкість вирішення проблеми мають принципове значення.

Показник CSAT зазвичай коливається у межах 70–90% для більшості успішних компаній, тоді як середнє значення NPS може суттєво відрізнятися залежно від галузі. Приміром, у технологічному секторі NPS у 50–70 вважається дуже високим, а в банківському середня оцінка може бути нижчою. CES не має такої загальноприйнятої шкали, однак зниження цього індексу зазвичай корелює зі зростанням лояльності клієнтів.

З погляду операційного аналізу, одночасне вимірювання CSAT, NPS і CES дає змогу розділити аудиторію на сегменти: клієнти, які задоволені, але не готові рекомендувати (високий CSAT, проте низький NPS), або ж ті, хто швидко отримав допомогу, однак залишився незадоволеним певними аспектами обслуговування (низький CSAT, але високий CES). Такі інсайти особливо цінні для корекції маркетингових і сервісних стратегій.

У комплексному дослідженні на різних етапах «клієнтського шляху» (customer journey) можна вимірювати різні метрики, бо на стадії приваблення (lead generation) більш вагоме значення має NPS, а на стадії післяпродажного обслуговування – CSAT чи CES. Це дає змогу оптимізувати найуразливіші точки взаємодії.

Часто результати вимірювань представляють у вигляді дашбордів або звітів, де відображаються тренди за період, кореляції з іншими показниками (обсяг продажів, кількість звернень у підтримку тощо). Глибший аналіз може залучати статистичні методи та машинне навчання, щоб передбачити коливання метрик і виявити фактори впливу.

Компанії, які послідовно відстежують дані показники, мають змогу швидше реагувати на кризові ситуації, вчасно помічати відтік клієнтів і

запроваджувати персоналізовані рішення. У довгостроковій перспективі це сприяє підвищенню конкурентоспроможності та зміцненню позицій на ринку.

Згідно з дослідженнями [3, 4], запровадження системного підходу до вимірювання та управління лояльністю та задоволеністю забезпечує відчутне зростання прибутку. Головне – обрати правильні інструменти та впровадити процеси, які дозволять перетворювати отримані дані на конкретні дії.

1.3 Огляд класичних підходів до моніторингу та вимірювання задоволеності клієнтів

Традиційний моніторинг задоволеності клієнтів здебільшого базується на прямому зборі відгуків, що дає змогу отримати кількісні та якісні показники лояльності. Історично компанії насамперед використовували різноманітні анкетування й телефонні опитування, оскільки це забезпечувало базову оперативність порівняно з поштовими розсилками [5]. Проте із часом потреба розширити охоплення та поглибити якість відповідей спонукала бізнеси до впровадження комплексних програм досліджень.

Анкетування є одним із ключових інструментів. Залежно від каналу комунікації (онлайн-формат, паперові анкети, телефонні опитування) можна швидко обробляти великі вибірки й отримувати середні показники задоволеності [5]. Це дозволяє фіксувати динаміку змін у часі, порівнювати різні сегменти та визначати основні тренди.

Для глибшого розуміння мотивації клієнтів класичні методи передбачають поглиблені інтерв’ю. Використання відкритих питань і можливість уточнювати відповіді дають змогу дослідникам виявляти справжні причини задоволеності чи незадоволеності. Саме цей якісний аспект дозволяє компаніям виходити за межі суто кількісних індексів [6].

Нерідко організації застосовують фокус-групи, щоб зібрати колективний досвід і зрозуміти реакцію на конкретні зміни в сервісі, продукті чи комунікації. У процесі модерації фокус-групи можливо простежити, як формуються й

транслюються групові очікування, що згодом впливають на рівень задоволеності [6]. Попри затратність проведення таких дискусій, їхній внесок у виявлення глибинних установок клієнтів є визнаним.

Серед класичних підходів виділяють також контент-аналіз скарг і пропозицій. Зіставляючи кількісний показник звернень і характер порушених проблем, можна визначити так звані «болючі точки» сервісу. Проте фокусування лише на скаргах може бути оманливим, адже більшість невдоволених клієнтів не завжди повідомляють про проблему [7]. Тому пасивний збір скарг має доповнюватися проактивними опитуваннями.

Класикою є й методи, орієнтовані на порівняння «очікувань» і «реальності» сервісу, як-от модель SERVQUAL, що вимірює розриви між тим, чого клієнт чекав, та тим, що фактично отримав [7]. Цей підхід дає змогу побачити саме якісну складову надійності, емпатії, компетентності персоналу тощо.

На рівні держав чи галузей існують універсальні індекси задоволеності, приміром American Customer Satisfaction Index (ACSI). Такі крос-галузеві порівняння дозволяють компаніям бачити, як вони виглядають на фоні середніх показників ринку [8]. Високі позиції в подібних індексах можуть слугувати маркетинговим інструментом для залучення нових клієнтів.

Популярності набув і Net Promoter Score (NPS), запропонований на початку 2000-х. Хоча NPS часто вважають «сучасною» метрикою, по суті він продовжує традицію коротких анкет, де клієнт оцінює ймовірність рекомендації за шкалою від 0 до 10 [8]. Відсоток «промоутерів» мінус відсоток «детракторів» створює інтегральний показник. У поєднанні з іншими методами він формує досить чітку картину лояльності.

Традиційне телефонне опитування все ще залишається актуальним для низки компаній, особливо якщо мова йде про старші вікові групи або специфічні ринки. Оператори кол-центрів здатні отримати глибший фідбек, ставлячи уточнювальні запитання. Проте масштабованість тут обмежена [9].

Ще одна класична методика – проведення періодичних «циклів опитувань», коли компанія раз на квартал чи півріччя розсилає анкету значній

вибірці клієнтів, а результати інтегрує в загальну звітність. За відсутності великих технологічних рішень ці періодичні «зрізи» задоволеності допомагають стежити за довгостроковою динамікою показників [9].

У дослідженнях фокусуються також на сегментуванні аудиторії: методи можуть відрізнятися для B2B і B2C, для різних вікових, соціальних і культурних груп. Класичні опитування іноді адаптуються з урахуванням локальної специфіки (мови, культурні норми), що зумовлює необхідність створювати окремі версії анкет [10].

Якісні дослідження, як-от фокус-групи і глибинні інтерв'ю, часто поєднують із кількісними опитуваннями, щоб визначити кореляцію між отриманими числовими показниками та мотивами клієнтів. Такий змішаний підхід забезпечує універсальність: можна з'ясувати, *наскільки* задоволені клієнти і *чому* вони задоволені/незадоволені [10].

У класичних методах існує ідея «закритого циклу зворотного зв'язку»: клієнту надсилається опитування, він дає оцінку, а компанія реагує на негатив. Проте у багатьох випадках ця реакція залишається лише на папері, без оперативних змін у процесах [11]. Тому ефективний моніторинг передбачає не лише збір даних, а й упровадження процесів постійного покращення.

Попри активний розвиток цифрових технологій, перелік традиційних інструментів залишається основою: анкети, інтерв'ю, скарги, фокус-групи, SERVQUAL тощо. Їх застосовують і досі, оскільки вони прості в реалізації та перевірені практикою. Одночасно вони слугують «відправною точкою» для тих компаній, які переходять до більш просунутих рішень [11].

У цілому класичні методи формують фундамент системи управління задоволеністю клієнтів. Вони не лише дають швидкі метрики на кшталт CSAT, а й допомагають зрозуміти логіку поведінки споживача. Навіть у добу Big Data та машинного навчання залишаються ситуації, коли саме проста анкета чи розмова із клієнтом здатна виявити сутність проблеми швидше, ніж будь-який алгоритм [12].

1.4 Сучасні інструменти та платформи для аналізу задоволеності

У сучасній бізнес-практиці моніторинг і вимірювання задоволеності вийшли далеко за межі звичайних анкет і телефонних опитувань. Компанії, які прагнуть більш гнучко реагувати на зміни попиту й уподобань, дедалі частіше впроваджують спеціалізовані платформи бізнес-аналітики (BI) [11]. Ці системи дозволяють в реальному часі з'єднувати дані з різних джерел і відстежувати ключові метрики у зручному дашборді.

Одним із найпопулярніших рішень є Microsoft Power BI (рисунок 1.3), що інтегрується з широким спектром баз даних і CRM-систем, автоматично оновлює звіти та забезпечує потужний інструментарій для візуалізації [11]. Це дає змогу командам підтримки та менеджерам бачити, як змінюється CSAT чи NPS фактично «онлайн», і миттєво виявляти проблемні ділянки обслуговування.

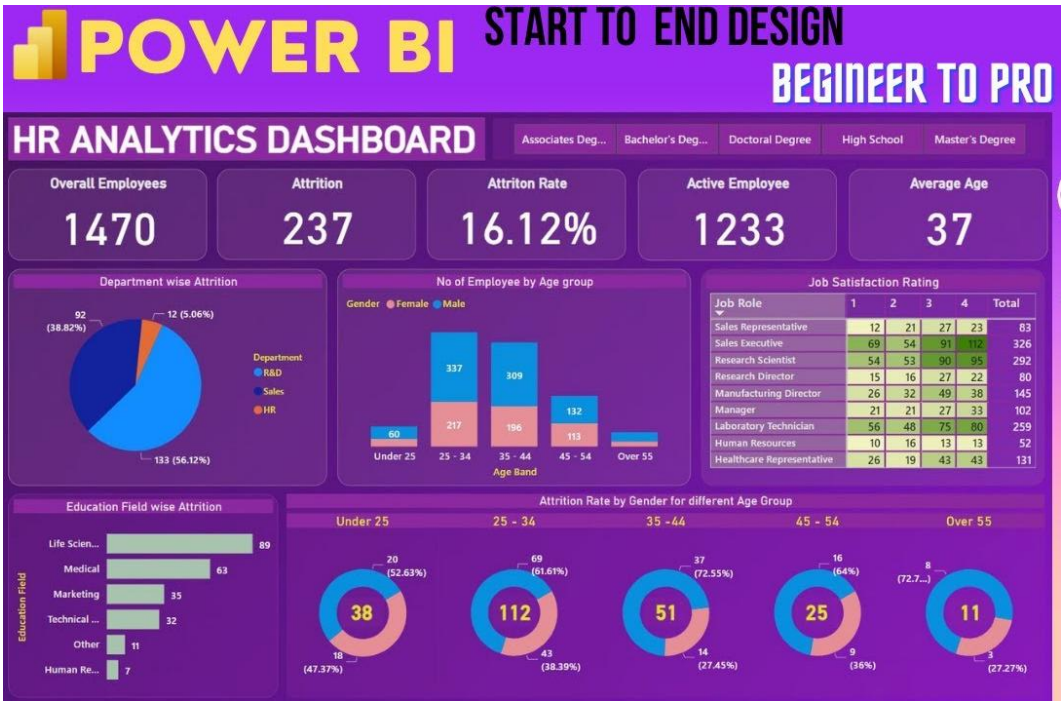

Рисунок 1.3 – Приклад дашборду в Microsoft Power BI

Схожі можливості надає Tableau, який відомий своїм зручним інтерфейсом для створення інтерактивних дашбордів. У контексті задоволеності клієнтів

Tableau уможливлює накладення кількох показників — наприклад, чисельності скарг, частоти звернень, часу відповіді, рейтингу оцінок — на єдину панель [11].

Окрему роль відіграють CRM-платформи. Якщо раніше CRM-системи (Salesforce, HubSpot) були переважно орієнтовані на управління продажами, то тепер вони містять повноцінні модулі для збору та аналізу відгуків клієнтів [13]. Після кожної взаємодії — контакту з підтримкою, транзакції, онбордингу — система може автоматично відправляти коротке опитування CSAT чи NPS, а відповіді потрапляють у профіль клієнта.

Такий «замкнутий цикл зворотного зв'язку» дозволяє негайно реагувати на негатив: якщо клієнт виставив низьку оцінку, відповідальній особі автоматично створюється завдання з ним зв'язатися [13]. Дослідження підтверджують, що впровадження CRM із функціями відстеження задоволеності сприяє підвищенню лояльності та зменшенню показників відтоку [13].

Ще одним поширеним інструментом стають рішення на базі обробки природної мови (NLP). За допомогою sentiment analysis компанії аналізують тисячі відкритих відгуків, коментарів у соціальних мережах, записів у чатах, аби визначити їхній емоційний тон [14]. Платформи на зразок IBM Watson, Google Cloud NLP у режимі реального часу виявляють відсоток негативу, позитиву чи нейтральних реакцій.

Це надзвичайно корисно, коли обсяг текстових даних величезний, і ручний перегляд усіх відгуків фактично неможливий [14]. Автоматизовані NLP-алгоритми (рисунок 1.4) маркують повідомлення клієнтів за темами (доставка, ціна, якість продукту тощо) і тональністю, створюючи панорами проблем та сильних сторін сервісу.

Деякі CRM вже інтегрують NLP-модулі безпосередньо у свій функціонал. Так, Salesforce Einstein здатний аналізувати відкриті питання в анкетах і пропонувати алгоритмічну оцінку емоцій клієнта [14]. Це дозволяє більш точно інтерпретувати числові оцінки та виявляти конкретні болючі точки, які зумовили негатив.

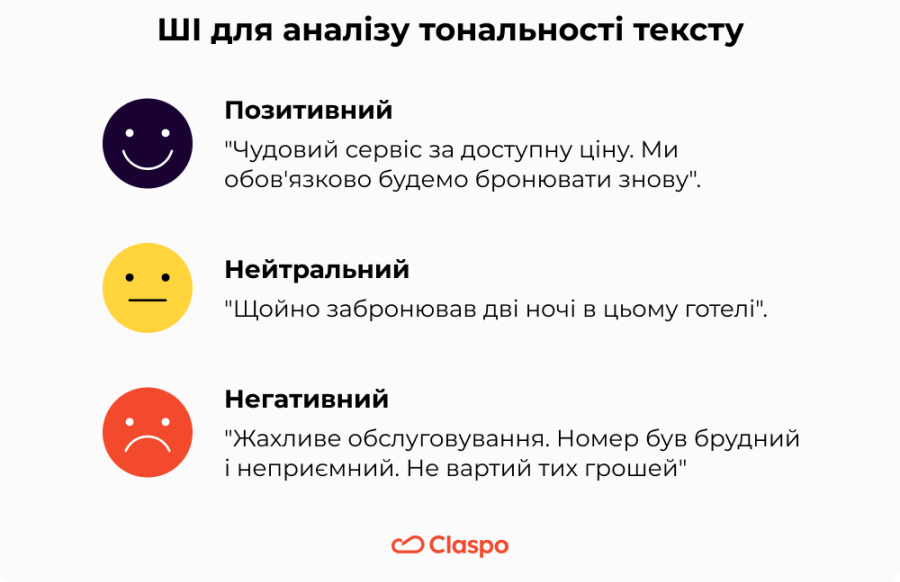

Рисунок 1.4 – Приклад інструменту NLP для аналізу тональності

Слід також згадати про Machine Learning. Алгоритми ML дають змогу прогнозувати показники задоволеності на основі минулого досвіду клієнта, його сегменту, часу реакції підтримки тощо [15]. Компанії можуть ранжувати клієнтів за ймовірністю надання негативної оцінки й упереджувати їхнє незадоволення.

Подібні predictive-моделі все частіше реалізовуються в платформах BI: Power BI підтримує інтеграцію з Azure Machine Learning, що дає змогу аналітикам створювати прогнози безпосередньо у звітах [15]. Якщо модель «бачить» ріст кількості невдоволених, система може інформувати відповідальних менеджерів про необхідність додаткових заходів.

Наприклад, ML-аналіз може показати, що клієнти з певним патерном поведінки (зменшення частоти покупок і ріст звернень у підтримку) перебувають у групі ризику зниження CSAT. Завдяки цьому менеджери можуть проактивно ініціювати контакт і запропонувати спеціальні умови чи консультації [15]. Таким чином машинне навчання підсилює можливості класичних підходів.

У контексті сучасних методів не можна оминути питання «all-in-one» платформ, де вся інформація про клієнта (транзакції, історія звернень, оцінки, соціальні мережі) об’єднана в єдиному дашборді [15]. Це прискорює прийняття рішень, оскільки менеджери одразу бачать, чому клієнт може бути незадоволений і що робити, аби рівень задоволеності виріс.

У деяких галузях запроваджують спеціалізовані мобільні додатки для миттєвого збору відгуків. Наприклад, в HoReCa чи роздрібній торгівлі – застосунки, де покупець одним кліком оцінює візит або залишає короткий коментар [15]. Такі повідомлення в реальному часі потрапляють у загальну базу даних і можуть миттєво оброблятися аналітичним модулем.

Перевага сучасних інструментів полягає не лише у швидкості й обсязі оброблюваних даних, а й у здатності поєднувати різні формати (текст, голос, емоції) у єдиному аналітичному просторі. Наприклад, контакт-центри можуть розпізнавати емоції клієнта по голосу, синхронізувати це з оцінкою CSAT і моментально сигналізувати про кризові випадки [15].

Загалом сучасні BI-системи, CRM-модулі й рішення NLP відкрили нові горизонти для керування задоволеністю клієнтів. Вони дають змогу не лише вимірювати показники на кшталт NPS чи CSAT, а й глибоко аналізувати поведінку й емоції клієнтів, виявляти приховані закономірності та проактивно усувати недоліки. Це закладає основу для безперервного вдосконалення сервісу в реальному часі [15].

1.5 Постановка задачі дослідження на основі виявлених проблем

У сучасному бізнес-середовищі, де конкуренція постійно зростає, задоволеність клієнтів стає визначальним фактором успіху компанії. Задоволений клієнт не лише повертається за повторною покупкою, але й рекомендує компанію іншим, що сприяє розширенню клієнтської бази. Традиційні методи оцінки задоволеності, такі як опитування чи анкетування, часто не враховують усіх аспектів взаємодії клієнта з компанією та можуть бути суб’єктивними. З розвитком цифрових технологій компанії отримують величезні масиви даних про поведінку клієнтів, їхні вподобання та відгуки. Однак без належних інструментів ці дані залишаються неструктурованими та не приносять користі. Інтелектуальний аналіз даних дозволяє виявляти приховані закономірності, прогнозувати поведінку клієнтів та адаптувати стратегії

обслуговування під їхні потреби. Використання сучасних BI-систем та NLP-рішень дає змогу автоматизувати процеси аналізу, зменшити людський фактор та підвищити точність оцінок. Інтеграція таких інструментів у CRM-системи забезпечує цілісний підхід до управління взаємовідносинами з клієнтами, що є критично важливим для утримання конкурентної переваги. Отже, дослідження, спрямоване на розробку модуля інтелектуального аналізу задоволеності клієнтів, є надзвичайно актуальним та відповідає сучасним викликам бізнес-середовища.

Вибір теми дослідження обумовлений необхідністю вдосконалення процесів оцінки та підвищення задоволеності клієнтів. Інтеграція інтелектуальних систем аналізу дозволяє автоматизувати процеси збору та обробки даних, що зменшує ймовірність помилок та суб'єктивності. Використання BI-систем забезпечує візуалізацію даних у зручному форматі, що сприяє швидкому прийняттю управлінських рішень.

Мета роботи - розробити модуль інтелектуального аналізу задоволеності клієнтів обслуговуванням, для підвищення якості обслуговування та лояльності клієнтів.

Завдання роботи:

1. Проаналізувати існуючі підходи та інструменти оцінки задоволеності клієнтів.

2. Розробити архітектуру модуля аналізу задоволеності клієнтів.

3. Створити модель даних та схему бази даних для зберігання інформації про клієнтів та їхню взаємодію з компанією.

4. Реалізувати алгоритми обробки та аналізу даних.

5. Розробити дашборди для візуалізації результатів аналізу та підтримки прийняття рішень.

2 ПРОЕКТУВАННЯ МОДУЛЯ АНАЛІЗУ ЗАДОВОЛЕНОСТІ

2.1 Архітектура системи та середовище розробки

У ході проєктування модуля аналізу задоволеності клієнтів було сформовано багаторівневу архітектуру, що дозволяє оптимізувати процес збирання, обробки та візуалізації даних. Такий підхід передбачає чіткий розподіл відповідальностей між компонентами системи та спрощує доопрацювання кожного модуля окремо.

Система складається з кількох ключових рівнів:

— Рівень збору даних (Data Ingestion): відповідає за отримання первинної інформації з різних джерел (CSV/Excel-файли, SQL-бази, CRM-система).

— Рівень обробки даних (Data Processing): містить скрипти для очищення, фільтрації та трансформації даних, а також для формування нових ознак (Feature Engineering) і обчислення статистичних показників.

— Рівень зберігання даних (Data Storage): структуровані та очищені дані можуть зберігатися у внутрішній базі даних (наприклад, PostgreSQL) або у форматі Parquet/CSV для подальшого використання.

— Аналітичний рівень (Analytics Layer): реалізує різноманітні методи аналізу, починаючи від статистичних методів і закінчуючи моделями машинного навчання, включно з побудовою дашбордів та інтерактивних графіків.

Основною мовою розробки вибрано Python, оскільки вона має вбудовані бібліотеки для наукових обчислень та аналізу даних:

— pandas (для роботи з табличними даними);

— numpy (для числових обчислень);

— matplotlib та seaborn (для базової візуалізації);

— plotly (для інтерактивних графіків);

— scikit-learn (для побудови моделей машинного навчання, якщо це потрібно);

— NLTK, WordCloud (для обробки текстових відгуків);

— інші бібліотеки для специфічних задач.

Найчастіше розробку ведуть у середовищах Jupyter Notebook або IDE (наприклад, PyCharm, Visual Studio Code). Також використовують хмарні

платформи (Kaggle, Google Colab, AWS) для експериментів із великими обсягами даних. У рамках цієї роботи було обрано комбінацію локального середовища (Jupyter Notebook) та Kaggle Notebooks, що полегшило спільну розробку та експериментування із даними у хмарі.

На рисунку 2.1 наведено спрощену блок-схему взаємодії між основними компонентами:

— Data Source: файли у форматах CSV або база даних.

— Data Pipeline: скрипти (Python), що виконують очищення, агрегування й оцінку якості даних.

— Analytics Module: набір ноутбуків або модулів Python, де проводиться статистичний аналіз і побудова моделей.

— Visualization & Reporting: використання бібліотек для створення дашбордів і графічних звітів.

— Deployment (за потреби): інтеграція результатів аналізу у веб-додаток, API чи внутрішні системи компанії.

Для гнучкості роботи з даними система передбачає:

— зчитування .csv, .xlsx та .json файлів;

— запити до SQL-баз (PostgreSQL, MySQL);

— підтримка паркет-файлів (Apache Parquet) для великих обсягів структурованих даних.

Рекомендується використовувати Git (напр., GitHub чи GitLab) для спільної розробки та підтримувати CI/CD-процеси, щоб своєчасно виконувати тести та оновлювати результати у централізованому репозиторії.

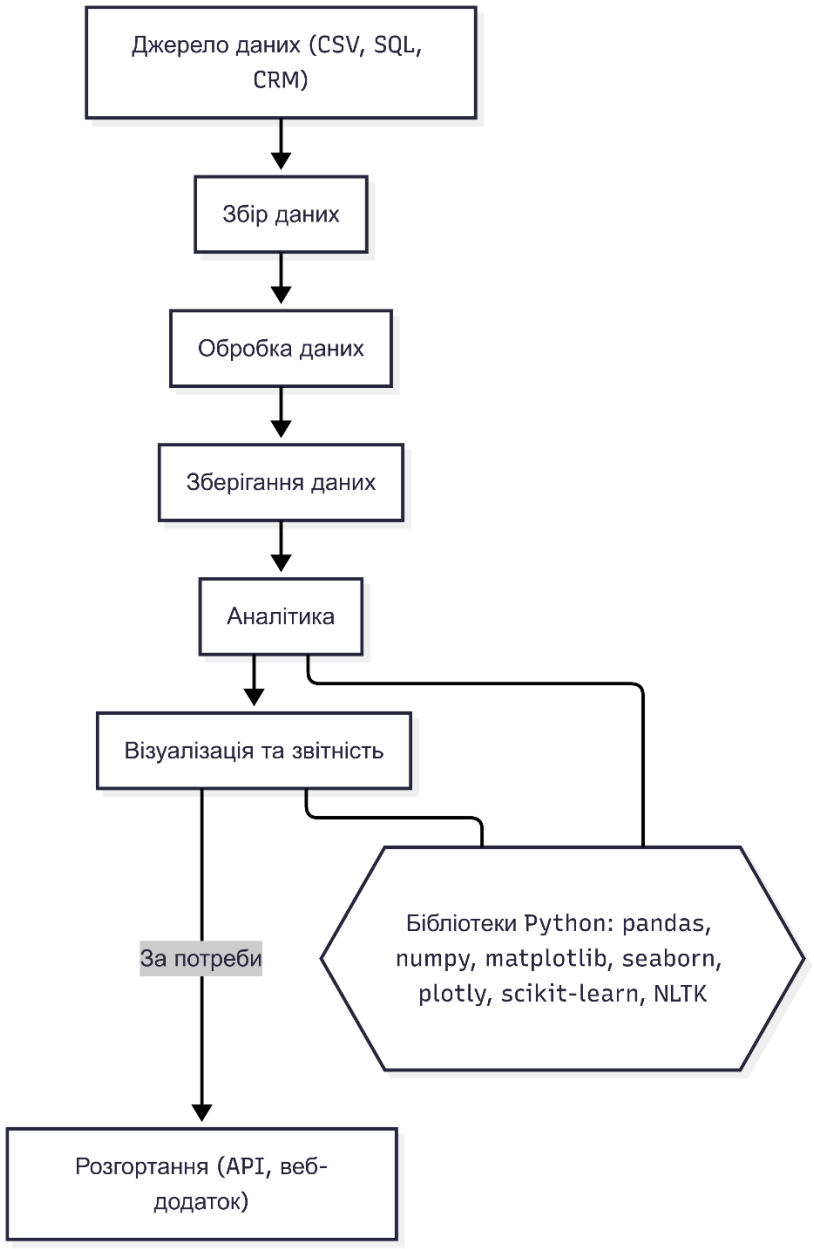

Рисунок 2.1 - Узагальнена структурна схема архітектури системи (Data Ingestion, Processing, Analytics, Visualization).

При зростанні обсягів даних можна застосовувати методи розподілених обчислень (Spark, Dask) або хмарні сервіси (AWS EMR, GCP DataProc). Це дає змогу прискорювати роботу аналітичного модуля та вирішувати більш масштабні задачі.

Архітектура має бути узгоджена з уже існуючою інфраструктурою компанії (CRM-система, служба підтримки, контакт-центр). Це передбачає наявність засобів імпорту даних із зовнішніх джерел і механізмів формування звітів, які можуть бути легко інтегровані у поточні робочі процеси.

Для зручності підтримки і розширення проєкту доцільно дотримуватися структурованого підходу:

— /data – вхідні та проміжні файли;

— /notebooks – аналітичні ноутбуки Python;

— /scripts – допоміжні скрипти (очищення, трансформація);

— /models – збережені моделі машинного навчання (за потреби);

— /reports – статичні чи інтерактивні звіти.

Важливим етапом є тестування кожного модуля системи: модулів очистки даних (перевірка виявлення пропусків, дублювань), алгоритмів агрегування та аналітичних методів (адекватність розрахунків). Це сприяє підвищенню надійності та достовірності результатів.

Ключові переваги обраної архітектури:

— чітке розмежування етапів збору, обробки й аналізу спрощує супровід;

— використання Python та популярних бібліотек забезпечує гнучкість і швидкий прототипінг;

— можливість інтеграції з будь-якою хмарною платформою дає простір для масштабування.

В підсумку, реалізована архітектура модуля аналізу задоволеності клієнтів дає змогу ефективно опрацьовувати великі обсяги даних, виконувати складні аналітичні обчислення й надавати інтерактивні звіти для прийняття рішень у реальному часі.

2.2 Модель даних та схема бази даних

На етапі проєктування структури даних варто виходити з вимог до системи, а саме: забезпечення високої швидкості обробки запитів, можливості гнучких фільтрацій та агрегацій, а також підтримки різних типів звітності. Зважаючи на це, у базі даних (БД) рекомендується виділити кілька таблиць для зберігання відомостей про клієнтів, звернення, категорії, коментарі, а також про агентів, супервайзерів і менеджерів.

Модель даних зазвичай формується як поєднання фактів (таблиця фактів із ключовими показниками на кшталт часу обслуговування, CSAT) та вимірів (довідкові таблиці – категорії, агенти, менеджери). Така підхід відповідає популярній методології «зірки» (Star Schema). Фактична таблиця містить посилання (через зовнішні ключі) на виміри, що дає змогу легко побудувати аналітичні запити.

У системі аналізу задоволеності клієнтів базовою сутністю виступає звернення (Interaction), адже кожне звернення породжує інформацію про канал комунікації, категорію, дату й час звернення, оцінку CSAT. Також важливим є сутність «Агент» (Agent), оскільки від правильного карткування звернень до агента залежить можливість відстеження його показників ефективності.

Тут зберігаються унікальний ідентифікатор звернення (Unique_id), дата та час надходження, канал зв'язку (channel_name), ознаки зворотного зв'язку (коментарі, оцінка задоволеності клієнта – CSAT), а також зовнішні ключі на таблиці «Агент», «Категорія» та «Менеджер». Додатково може існувати окреме поле для зберігання часу відповіді (ResponseTime) і тривалості обробки (connected_handling_time).

У таблиці «Агент» містяться унікальний ідентифікатор агента, ПІБ, дата найму, належність до супервайзера й менеджера. Це дозволяє досліджувати продуктивність окремих агентів, а також їхню ієрархічну приналежність у межах організації.

Окремі таблиці або одна довідкова таблиця з ролями. У ній зберігається інформація про відповідну особу, її повноваження та підлеглих. Така структура дозволяє збирати статистику за кожним рівнем управлінської ієрархії.

Зазвичай створюються окремі довідкові таблиці, у яких зберігаються унікальні ідентифікатори категорій та їх назви. Для підкатегорій можливе створення таблиці з посиланням на «батьківську» категорію. Такий підхід надає гнучкість у розширенні або зміні категоризації звернень.

Кожна таблиця фактів (наприклад, «Interactions») має зовнішні ключі, що посилаються на таблиці довідників: «Agent» (агент), «Manager» (менеджер), «Category» (категорія). Завдяки цьому забезпечується референтна цілісність, яка

унеможливлює створення записів, що не відповідають реально існуючим сутностям у довідниках.

На рисунку 2.2 наведено узагальнену схему бази даних. Основна «таблиця фактів» Interactions містить ключову інформацію про звернення, тоді як пов’язані з нею довідники «Agent», «Manager», «Category» зберігають деталі про характеристики агента, керівництва та типи звернень.

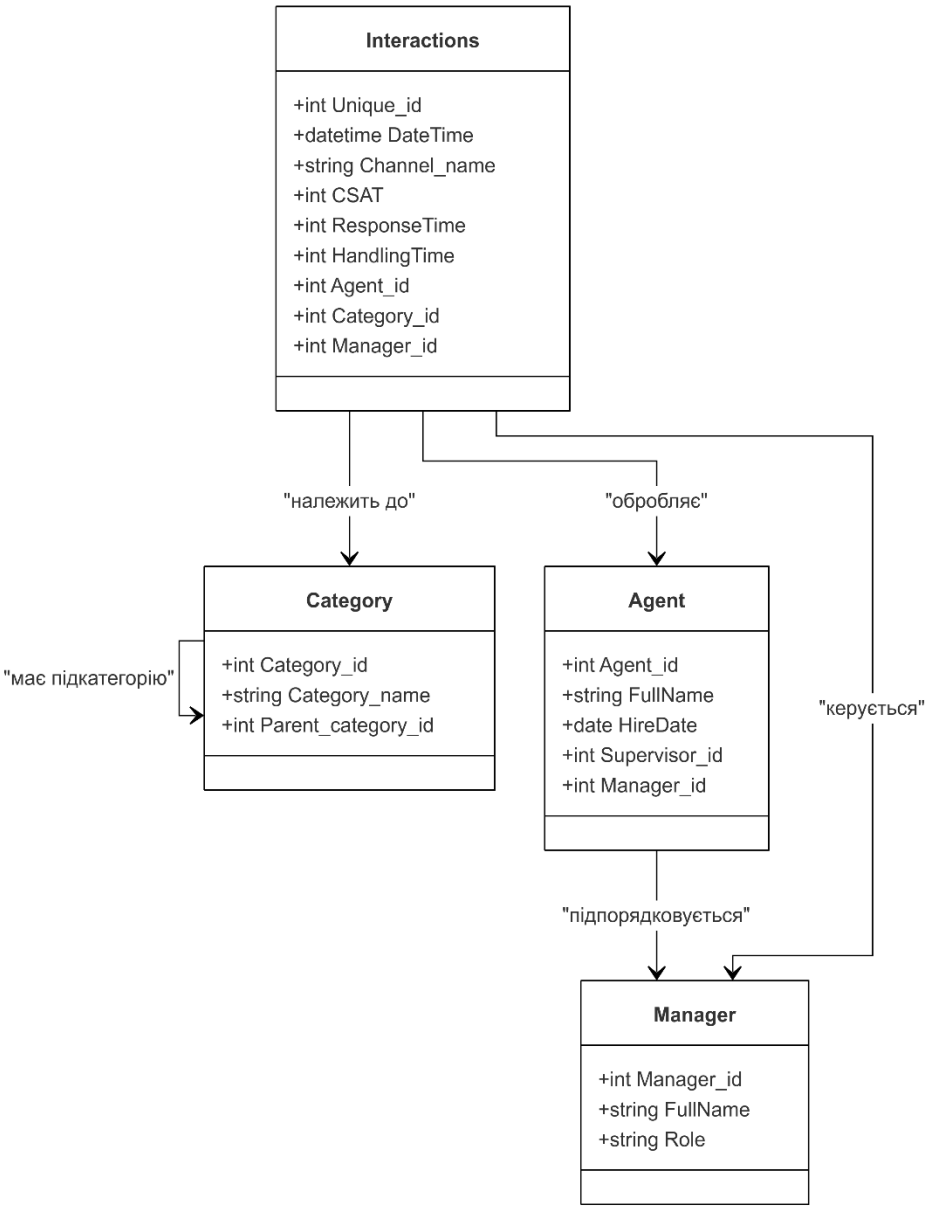

Рисунок 2.2 - Схема бази даних модуля аналізу задоволеності клієнтів

В кожній таблиці рекомендується визначати первинний ключ (Primary Key), що гарантує унікальність, наприклад, Unique_id в «Interactions» та Agent_id

в «Agent». За необхідності також можна встановлювати індекси на поля, що часто використовуються в пошукових запитах (наприклад, Manager_id чи Category_id).

З метою підвищення цілісності та мінімізації дублювання таблиці зазвичай приводять до як мінімум другої або третьої нормальної форми (2NF, 3NF). Це дає змогу уникнути конфліктів при оновленні та пришвидшити підтримку системи. Проте для аналітичного навантаження інколи реалізовують денормалізацію — додаткові поля чи таблиці агрегатів, щоб пришвидшити вибірки.

Якщо система передбачає обробку великого масиву звернень (наприклад, сотні тисяч чи мільйони на місяць), важливо розглянути питання індексування ключових полів, застосування партиціювання таблиць (наприклад, за місяцями) та використання інструментів для кешування результатів (Materialized Views, Redis тощо).

Ретельне проєктування моделі даних і схеми бази даних є критично важливим для стабільної та ефективної роботи системи аналізу задоволеності клієнтів. Продумана структура дозволяє не лише забезпечити швидкий доступ до необхідних даних, а й зберігати цілісність та достовірність інформації. Надалі ця схема стане основою для виконання складних SQL-запитів, генерації звітів і проведення аналітичних досліджень.

2.3 Алгоритмічне та математичне забезпечення

Цей підрозділ охоплює методи обробки даних, алгоритми аналізу та візуалізації, що використовуються під час проєктування модуля аналізу задоволеності клієнтів. Ефективне алгоритмічне й математичне забезпечення є ключем до отримання достовірних висновків і передбачень, адже будь-яка похибка в підготовці чи обчисленнях може істотно спотворити підсумкові результати. Нижче наведено докладний опис кожного з етапів.

2.3.1 Алгоритми підготовки та очищення даних

У більшості випадків, вихідні набори, отримані з CRM-систем, лог-файлів або інших джерел, можуть містити велику кількість пропусків, неузгоджених форматів дат, помилок у текстових полях та дублікатів. З огляду на це, першим завданням є систематичний процес виявлення, класифікації та усунення таких недоліків.

На рисунку 2.3 наведено схему алгоритму, що демонструє послідовність дій у процесі підготовки даних:

Крок 1. Завантаження первинного набору та вивчення структури (наприклад, огляд перших рядків, перевірка типів даних).

Крок 2. Виявлення та обробка пропусків (видалення або заповнення), що залежить від статистичної важливості полів.

Крок 3. Виявлення та усунення дублікатів (повне дублювання рядків або часткове, де ідентифікатори можуть збігатися).

Крок 4. Перевірка та коригування аномальних значень (викиди, тобто точки за межами очікуваного діапазону).

Крок 5. Формування єдиного набору з чітко визначеними змінними.

Цей рисунок ілюструє покроковий підхід, при якому кожний етап може бути налаштований залежно від специфіки бізнесу та типу даних. Наприклад, у випадку, коли кількість пропущених значень у полі не перевищує 5%, можна застосувати заповнення середнім або медіанним; якщо пропусків понад 30%, таке поле зазвичай вилучають.

Серед найпопулярніших методів:

— Drop (видалення): якщо значна частка рядків повністю нефункціональна.

— Imputation (заповнення середнім, медіанним, модою або іншим закономірним значенням).

Рисунок 2.3 – Схема алгоритму підготовки та очищення даних

— Заміна категорією "Unknown": здебільшого для текстових полів. Обраний метод впливає на подальші висновки, оскільки може змінювати реальні статистичні закономірності.

Деякі записи можуть кілька разів повторюватися через технічні збої. У таких випадках застосовують агрегацію (наприклад, якщо у клієнта кілька однакових звернень) або просто видаляють дублікати. Для пошуку аномалій

(викидів) користуються або статистичними правилами (наприклад, 3-сигмове відхилення), або алгоритмічними методами на кшталт Isolation Forest, які автоматично визначають «найбільш дивні» точки.

Коли числові поля мають кардинально різні діапазони, це може призвести до неправильних результатів при подальшому аналізі (зокрема, при побудові моделей машинного навчання). Тому роблять нормалізацію (наприклад, мін-макс) або стандартизацію (приведення до нульового середнього та одиничного стандартного відхилення). Особливо актуально це для тривалостей обслуговування, часу відповіді, сум оплат тощо.

Часто для кращого аналізу потрібно розкласти дату та час на окремі поля: день тижня, година, номер місяця тощо. Це дає змогу легко формувати агрегати (наприклад, середній CSAT за годинами дня). Для текстових колонок застосовують перетворення в категоріальний тип, що полегшує групування та агрегацію.

2.3.2 Методи статистичного аналізу та візуалізації

Після очищення даних обчислюють базові характеристики: середнє (mean), медіана (median), мода (mode), стандартне відхилення (std), коефіцієнт варіації тощо. Це дає змогу побачити загальну картину та визначити потенційні відхилення. Наприклад, якщо середнє значення CSAT близьке до 4,5, а медіана 5, це може свідчити про високу концентрацію оцінок на верхній межі шкали.

Один із фундаментальних методів – обчислення коефіцієнта кореляції Пірсона (r) або Спірмена (у разі неспостереження нормальності). Кореляційна матриця допомагає визначити, чи існує статистично значуща залежність між, наприклад, часом відповіді та оцінкою CSAT, або між вартістю товару та показником задоволеності. Якщо $|r| > 0.7$, зазвичай вважається, що зв’язок сильний.

Якщо потрібно порівняти середні показники CSAT між кількома групами (наприклад, «ранкова зміна» vs. «нічна зміна»), застосовують t-тест чи ANOVA. Коли груп більше ніж дві, після ANOVA бажано зробити пост-хок тест (Tukey)

для визначення парних відмінностей. Ці тести допомагають зрозуміти, чи відмінності випадкові, чи справді відображають реальні особливості.

Для обчислення середнього значення показника задоволеності (за певний період або вибірку) можна використати звичайну формулу середнього:

$$\mathrm{CSAT}_{mean} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{CSAT}_i, \tag{2.1}$$

де N – загальна кількість респондентів;

$CSAT_i$– оцінка і-го респондента.

Якщо $CSAT_i$ уніфіковано як 0 або 1 (бінарний підхід), середнє відображає відсоток задоволених клієнтів.

Окрім базових середніх, у бізнес-практиці відомі й інші метрики – Net Promoter Score (NPS), Customer Effort Score (CES). Вони також можуть аналізуватися за допомогою описових статистик і статистичних тестів. Наприклад, NPS визначається як різниця між часткою «промоутерів» і «критиків».

Серед типових діаграм: гістограми (histogram) для відображення розподілу CSAT, діаграми розсіювання (scatter plot) для виявлення кореляцій, графіки Pareto для виокремлення «критичних» категорій звернень. При аналізі годинних або денних трендів часто застосовують лінійні графіки з ковзаючим середнім.

На рисунку 2.4 наведено алгоритм послідовності дій при статистичному аналізі й візуалізації:

Крок 1. Завантаження очищених даних.

Крок 2. Обчислення описових статистик (mean, median, std, quartiles).

Крок 3. Перевірка нормальності розподілу (Shapiro-Wilk, Kolmogorov-Smirnov).

Крок 4. Вибір відповідних тестів (t-тест, ANOVA, кореляційних методів).

Крок 5. Створення візуалізацій (гістограм, boxplot, лінійних графіків).

Крок 6. Інтерпретація результатів і підготовка звітів.

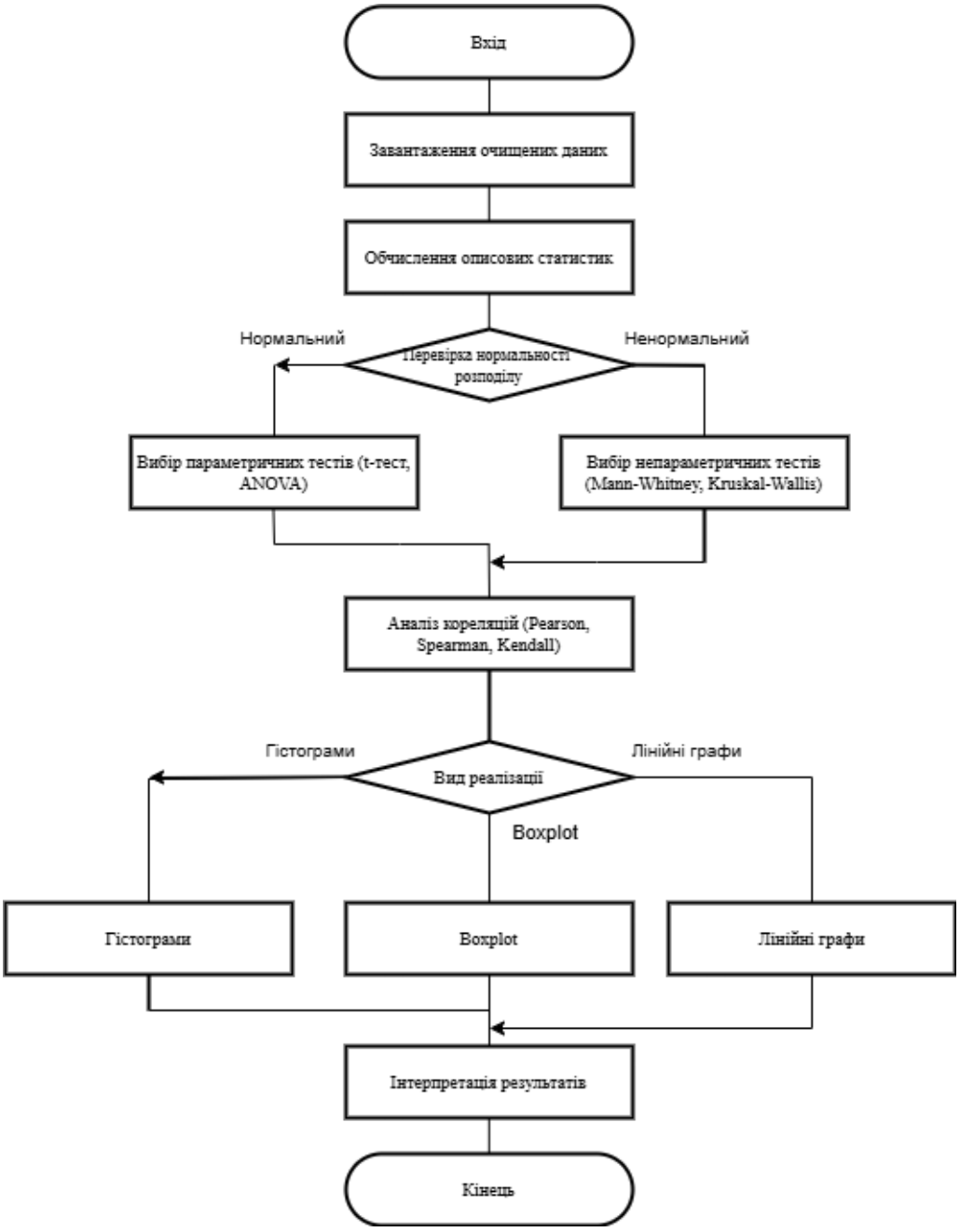

Рисунок 2.4 – Схема етапів статистичного аналізу та візуалізації

На практиці цей процес може повторюватися ітеративно: після первинних висновків дослідник може повернутися до етапу очищення чи трансформації даних.

Після застосування класичних статистичних методів, за потреби можна розглянути складніші алгоритми, як-от лінійну чи логістичну регресію, Random Forest, XGBoost чи нейронні мережі. Вони дають змогу передбачити рівень CSAT на основі вхідних ознак (тип звернення, ціна товару, час відповіді тощо). Однак

ефективність цих методів прямо залежить від якості попередніх кроків очищення й аналізу.

Поєднання ретельної підготовки даних і статистичного аналізу дозволяє:

— зменшити ризик хибних висновків;

— уникнути впливу викидів і шуму;

— швидко ідентифікувати ключові закономірності та проблемні зони;

— отримати валідні основи для побудови прогнозних моделей.

Загалом, алгоритмічне та математичне забезпечення має вирішальне значення для точності та повноти результатів аналізу. Починаючи від очищення даних і завершуючи побудовою інтерактивних графіків, кожен етап потребує коректних методів та інструментів. Системний підхід до обробки інформації забезпечує надійність підсумкової статистики та відкриває перспективи для розширення функціонала, зокрема й запровадження більш складних алгоритмів машинного навчання.

2.4 Проектування модулів для дашборду-звітності

Основна ідея проєктування дашборду полягає в тому, щоб надати швидкий доступ до ключових показників ефективності (KPI) та даних, які характеризують рівень задоволеності клієнтів. Зазвичай дашборд містить набір інтерактивних віджетів (графіків, діаграм, таблиць), що дають змогу оперативно оцінити стан справ і прийняти обґрунтовані рішення. В нашому проєкті ключовим показником є CSAT (Customer Satisfaction), а також додаткові метрики (NPS, CES, час відповіді тощо).

Під час розробки важливо дотримуватися таких критеріїв:

— інтуїтивність: легко зрозумілий інтерфейс, чіткі назви метрик;

— швидкодія: дані повинні завантажуватися та оброблятися з мінімальними затримками;

— адаптивність: можливість фільтрації за періодом, категорією звернення, каналом комунікації;

— безпека: забезпечення ролей користувачів (агенти, супервайзери, менеджери) і доступу лише до відповідних даних.

Як правило, інтерактивна система дашборду складається з фронтенду (UI) та бекенду, який отримує дані з бази. У межах нашої реалізації (рисунок 2.5) можемо виокремити такі модулі:

— модуль отримання даних: надсилає запити до бази даних, формує агреговані вибірки;

— модуль обчислення метрик: розраховує CSAT, NPS, середній час відповіді;

— модуль візуалізації: генерує інтерактивні графіки (plotly, matplotlib або інші).

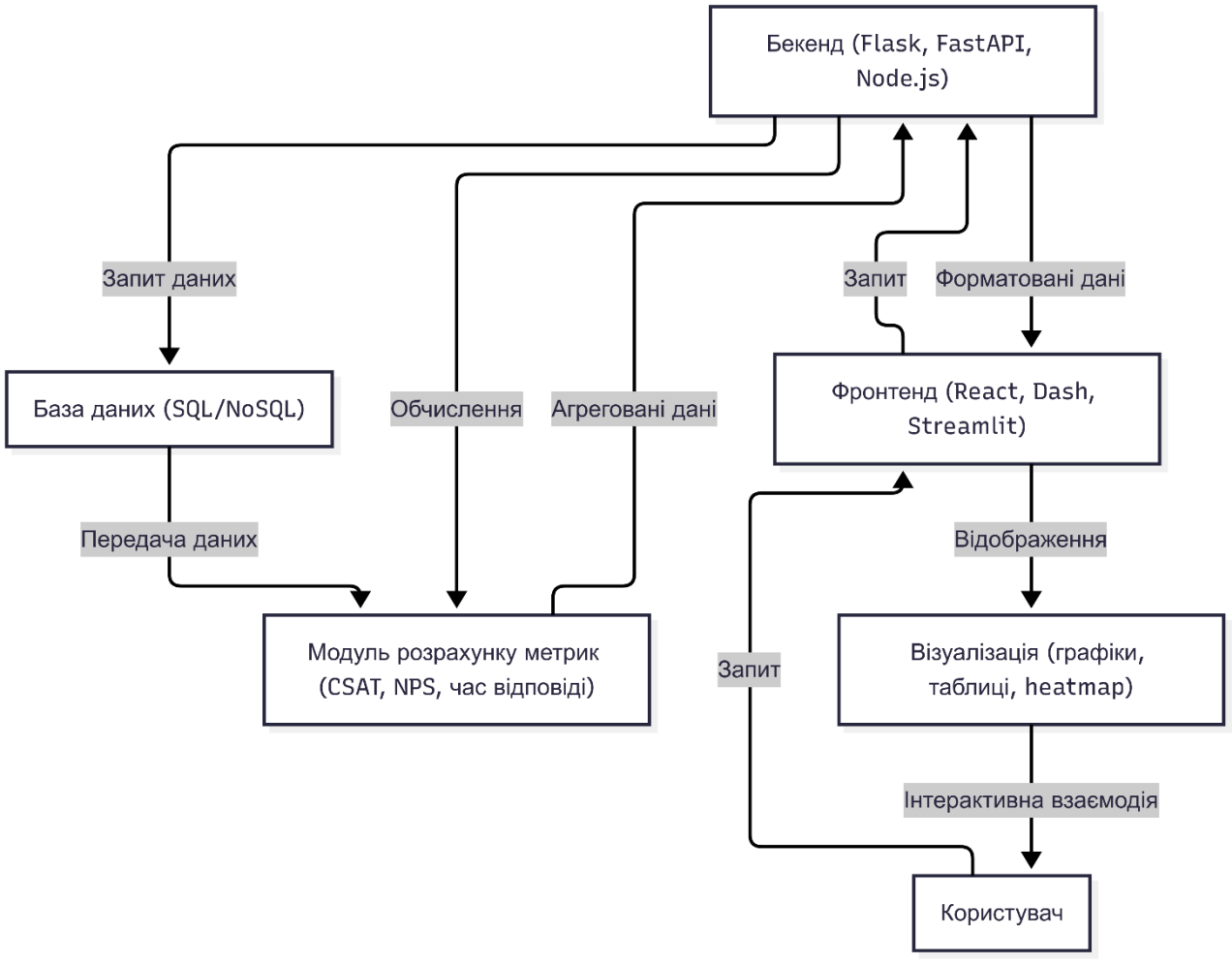

Рисунок 2.5 – Схема інтерактивної візуалізації дашборду

На цій схемі показана взаємодія між компонентами: фронтенд на React або іншій бібліотеці, бекенд на Python/Flask або Node.js та СУБД (SQL або NoSQL). Аналітична логіка (розрахунки метрик) може бути винесена окремим сервісом.

Залежно від цілей аналізу, на дашборді можуть бути розміщені:

— лічильники (cards) з кількістю опрацьованих звернень, середнім CSAT, середнім часом відповіді;

— лінійні графіки з щоденним/тижневим трендом CSAT;

— стовпчикові діаграми для порівняння груп (наприклад, топ-10 агентів);

— теплокарти (heatmap) для виявлення залежностей між категоріями та часом.

Користувачі (менеджери, супервайзери) часто бажають «просвердлити» (drill-down) у певні підкатегорії звернень або часові періоди. Тому важливим елементом є панель фільтрації: вибір категорії, дати, каналу комунікації тощо. При зміні фільтру дашборд автоматично оновлює всі віджети.

Окрім агрегованих графіків, корисно мати окремий модуль з деталізацією (наприклад, таблиця зі списком звернень, де можна подивитися історію клієнта або коментарі). Це дозволяє менеджерам визначати конкретні випадки низького CSAT та оперативно реагувати.

Для формування інтерактивної візуалізації можна використати plotly чи dash, що дає змогу створювати веб-інтерфейси на основі Python. Іноді застосовується streamlit – простіший фреймворк для швидкого прототипування дашбордів. На боці серверу (Flask, FastAPI) реалізують API, що обробляє запити та передає дані клієнту.

Якщо дашборд відображає конфіденційну інформацію (дані клієнтів, продажі, ревеню), обов’язково слід передбачити автентифікацію та авторизацію. Для зручності можна інтегруватися з корпоративною LDAP або SSO.

Оскільки керівники можуть переглядати дашборд з ноутбука, планшета чи смартфона, варто забезпечити адаптивний інтерфейс (responsive design), щоб усі графіки та таблиці коректно відображались на різних екранах.

Окрім відображення поточних метрик, дашборд може містити прогностичні модулі (приміром, прогноз CSAT на наступний тиждень) або алгоритмічні поради (які категорії звернень краще обробити насамперед). Такі функції можуть грунтуватися на моделях машинного навчання, що періодично оновлюються.

Успішна реалізація передбачає обов'язкове тестування: функціональне (перевірка коректності відображення даних), навантажувальне (чи дашборд не "падає" при великому числі користувачів), юзабіліті-тестування (зручність інтерфейсу). Важливо залучити реальних користувачів до бета-тесту.

Інтегрований модуль може надавати можливість генерувати PDF чи Excel-файли зі статистикою. Це корисно для офлайн-аналітики, коли потрібно продемонструвати результати вище по ієрархії або обговорити їх під час нарад.

Основна перевага – здатність швидко і динамічно аналізувати дані без необхідності виконувати складні SQL-запити чи Python-скрипти. Це істотно заощаджує час керівників і сприяє ухваленню рішень на основі актуальних метрик.

Проектування модуля для дашборду/звітності – один із заключних, але вкрай важливих етапів розробки системи аналізу задоволеності клієнтів. Користувачі отримують зручні інструменти для моніторингу ключових показників у реальному часі, а сама компанія – змогу оперативно реагувати на відхилення та виявляти вузькі місця у роботі з клієнтами. Спеціальні фільтри та інтерактивні елементи поглиблюють аналітику, забезпечуючи всебічну підтримку прийняття управлінських рішень.

3 РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

3.1 Опис середовища програмування

Python є одним із найпопулярніших мов програмування для аналізу даних та машинного навчання. Він забезпечує широкий набір бібліотек для обробки, аналізу, візуалізації та моделювання даних, що робить його ідеальним вибором для реалізації аналізу задоволеності клієнтів.

Основними бібліотеками, які використовуються в цьому дослідженні, є:

— pandas – для роботи з табличними даними;

— numpy – для числових операцій;

— matplotlib та seaborn – для візуалізації даних;

— plotly – для інтерактивної аналітики;

— wordcloud – для аналізу текстових відгуків клієнтів.

```
# Імпорт необхідних бібліотек
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
from matplotlib.offsetbox import AnnotationBbox, OffsetImage
import matplotlib.image as mpimg
from wordcloud import WordCloud, STOPWORDS

warnings.filterwarnings('ignore')
```

Дані завантажуються та попередньо обробляються наступним чином:

```
# Завантаження даних
file_path = '/kaggle/input/ecommerce-customer-service-satisfaction/Customer_support_da
df = pd.read_csv(file_path)
df.columns = df.columns.str.replace(' ', '_')
pd.set_option('display.max_columns', None)

# Додавання колонки CSAT на основі CSAT Score
def map_csat(score):
    return 1 if score >= 4 else 0

df['CSAT'] = df['CSAT_Score'].apply(lambda x: map_csat(x))
```

Цей код виконує імпорт бібліотек, завантажує набір даних та проводить первинну обробку, включаючи нормалізацію назв колонок та мапування рівня задоволеності клієнтів у бінарний формат (1 – задоволені, 0 – незадоволені).

Таке середовище програмування забезпечує ефективну реалізацію дослідження, дозволяючи проводити складні обчислення, аналітику та створювати візуалізації для оцінки рівня задоволеності клієнтів.

Функція summary створює детальну таблицю, що містить типи даних, кількість відсутніх значень, кількість унікальних значень, а також статистичні показники (мінімум, максимум, середнє, стандартне відхилення). Це дозволяє отримати загальне уявлення про структуру даних і їх якість:

```
def summary(df):
    summry = pd.DataFrame(df.dtypes, columns=['data type'])
    summry['#missing'] = df.isnull().sum().values
    summry['Duplicate'] = df.duplicated().sum()
    summry['#unique'] = df.nunique().values
    desc = pd.DataFrame(df.describe(include='all').transpose())
    summry['min'] = desc['min'].values
    summry['max'] = desc['max'].values
    summry['avg'] = desc['mean'].values
    summry['std dev'] = desc['std'].values
    summry['top value'] = desc['top'].values
    summry['Freq'] = desc['freq'].values
    return summry

summary(df).style.set_caption("**Summary of the Train Data**")\
.background_gradient(cmap='Pastel2_r', axis=0)\
.set_properties(**{'border': '1.3px dotted', 'color': '', 'caption-side': 'left'})
```

За допомогою групування даних за менеджерами, супервайзерами та агентами обчислюється середнє значення, сума та кількість відгуків CSAT. Додатково розраховується загальний відсоток CSAT, що допомагає у визначенні рівня задоволеності:

```
csat_summary = df.groupby(['Manager', 'Supervisor', 'Agent_name'])['CSAT']
csat_summary['CSAT%'] = df['CSAT'].mean() * 100

csat_percentage = csat_summary['CSAT%'][0]
total_surveys = csat_summary['count'].sum()
positive_count = csat_summary['sum'].sum()
negative_count = total_surveys - positive_count
```

За допомогою бібліотеки Plotly створюється інтерактивний графік, який відображає основні метрики: відсоток CSAT, загальну кількість опитувань,

позитивні та негативні відгуки. Текстові анотації роблять графік зрозумілим і інформативним:

```
fig = go.Figure()
fig.add_trace(go.Scatter(
    x=[0, 1, 2, 3],
    y=[1.6, 1.6, 1.6, 1.6],
    mode="text",
    text=["<span style='font-size:20px'><b>CSAT %</b></span>",
          "<span style='font-size:20px'><b>Total Surveys</b></span>",
          "<span style='font-size:20px'><b>Positive Count</b></span>",
          "<span style='font-size:20px'><b>Negative Count</b></span>"],
    textposition="bottom center"))
fig.add_trace(go.Scatter(
    x=[0, 1, 2, 3],
    y=[1.1, 1.1, 1.1, 1.1],
    mode="text",
    text=[f"<span style='font-size:20px'><b>{csat_percentage:.2f}%</b></span>",
          f"<span style='font-size:20px'><b>{total_surveys}</b></span>",
          f"<span style='font-size:20px'><b>{positive_count}</b></span>",
          f"<span style='font-size:20px'><b>{negative_count}</b></span>"],
    textposition="bottom center"))
# Додаткові налаштування зовнішнього вигляду пропущено для стислості
fig.show()
```

У цьому блоці обчислюється кількість агентів, супервайзерів та менеджерів. Результат виводиться у вигляді інтерактивної діаграми, що підсилює візуальне сприйняття даних:

```
agents = df['Agent_name'].nunique()
supervisors = df['Supervisor'].nunique()
managers = df['Manager'].nunique()
fig = go.Figure()
fig.add_trace(go.Scatter(
    x=[0, 1, 2],
    y=[1.1, 1.1, 1.1],
    mode="text",
    text=[f"<span style='font-size:20px'><b>{agents}</b></span>",
          f"<span style='font-size:20px'><b>{supervisors}</b></span>",
          f"<span style='font-size:20px'><b>{managers}</b></span>"],
    textposition="bottom center"))
# Налаштування осей та зовнішнього вигляду опущені для стислості
fig.show()
```

За допомогою Seaborn будується бар-графік, який відображає топ-менеджерів із найвищим показником CSAT. Використання палітри кольорів, анотацій та додаткових зображень (логотип) підсилює візуальну привабливість графіка:

```
csat_summary = df.groupby('Manager')['CSAT'].agg(['mean', 'count']).reset_index()
csat_summary.columns = ['Manager', 'CSAT', 'Survey Count']
csat_summary = csat_summary.sort_values(by='CSAT', ascending=False).head(3)
colors = ['#bcd6e2', '#9bd0eb', '#ced4d6']
positions = [1, 0, 2]
fig, ax = plt.subplots(figsize=(12, 4.5), facecolor='#1C1D20')
bars = sns.barplot(x=positions, y='CSAT', data=csat_summary, palette=colors, ax=ax, width=1,
ax.set_xticks(positions)
ax.set_xticklabels(csat_summary['Manager'], color='#E1B12D')
ax.set_yticks([])
ax.set_ylabel('')
ax.set_ylim(0.7, 0.87)
ax.set_title('Top Performing Managers', color='#E1B12D', loc='left', pad=25, fontweight='bold
# Додавання зображення логотипу
img = mpimg.imread('/kaggle/input/podium/84381-Podium_PowerPoint_Template-removebg-preview.pn
ax.imshow(img, extent=[-1.55, 3.5, 0.67, 0.92], aspect='auto', zorder=1)
sns.despine(ax=ax, left=True, bottom=True)
# Анотації до кожного стовпчика
for bar, csat, surveys, position in zip(bars.patches, csat_summary['CSAT'], csat_summary['Sur
    csat_text = f"CSAT%: {csat*100:.2f}%"
    survey_text = f"Surveys: {surveys}"
    if position == 0:
        label_x = position + 0.25
        offset = 0.0
    elif position == 1:
        label_x = position + 0.25
        offset = 0.03
    else:
        label_x = position + 0.25
        offset = 0.02
    label_height = bar.get_height() + offset
```

Подібно до попереднього блоку, виконується аналіз супервайзерів. Візуалізація здійснюється за допомогою схожої методики, що дозволяє порівняти ефективність різних рівнів управління:

```
csat_summary = df.groupby('Supervisor')['CSAT'].agg(['mean', 'count']).reset_index()
csat_summary.columns = ['Supervisor', 'CSAT', 'Survey Count']
csat_summary = csat_summary.sort_values(by='CSAT', ascending=False).head(3)
colors = ['#bcd6e2', '#9bd0eb', '#ced4d6']
positions = [1, 0, 2]
fig, ax = plt.subplots(figsize=(12, 4.5), facecolor='#1C1D20')
bars = sns.barplot(x=positions, y='CSAT', data=csat_summary, palette=colors, ax=ax, width=1
ax.set_xticks(positions)
ax.set_xticklabels(csat_summary['Supervisor'], color='#E1B12D')
ax.set_yticks([])
ax.set_ylabel('')
ax.set_ylim(0.7, 0.88)
ax.set_title('Top Performing Supervisors', color='#E1B12D', loc='left', pad=27, fontweight=
# Додавання логотипу та стилізації
img = mpimg.imread('/kaggle/input/podium/84381-Podium_PowerPoint_Template-removebg-preview.
ax.imshow(img, extent=[-1.55, 3.5, 0.67, 0.92], aspect='auto', zorder=1)
sns.despine(ax=ax, left=True, bottom=True)
```

Групування даних за агентами дозволяє створити горизонтальну діаграму, де окремо відображаються топ-10 агентів та 10 агентів з найнижчим показником CSAT. Використання анотацій сприяє кращому сприйняттю результатів:

```
agent_summary = df.groupby('Agent_name')['CSAT'].agg(['mean', 'count']).reset_index()
agent_summary.columns = ['Agent_name', 'CSAT%', 'Survey Count']
agent_summary = agent_summary.sort_values(by='CSAT%', ascending=False)
top_n_agents = agent_summary.head(10)
bottom_n_agents = agent_summary.tail(10)
fig = go.Figure()
fig.add_trace(go.Bar( x=-bottom_n_agents['CSAT%'] * 100, y=bottom_n_agents['Agent_name'],
                      orientation='h', marker=dict(color='#c67737')))
fig.add_trace(go.Bar(  x=top_n_agents['CSAT%'] * 100, y=top_n_agents['Agent_name'],
                      orientation='h', marker=dict(color='#528C78')))
for index, row in bottom_n_agents.iterrows():
    fig.add_annotation( x=row['CSAT%'] * 100,  y=row['Agent_name'], text=f"{row['Agent_name'
                           showarrow=False, font=dict(color='#e16060', size=10), xshift=-15
for index, row in top_n_agents.iterrows():
    fig.add_annotation( x=row['CSAT%'] * 100,  y=row['Agent_name'],
                      text=f"{row['Agent_name']} ({row['CSAT%'] * 100:.2f}%)", showarrow=F
fig.update_layout( title='Top & Bottom 10 Agents by CSAT%', plot_bgcolor='#1C1D20', height=6
    xaxis_title='', yaxis_title='', bargap=0.1, yaxis=dict(tickvals=[]), xaxis=dict(tickvals
    showlegend=False, title_font=dict(size=20, family="Lato, sans-serif"), font=dict(color='
fig.show()
```

Конвертація стовпця з датами відповіді в формат datetime дозволяє групувати дані за днями та тижнями. Це сприяє побудові трендів CSAT і кількості опитувань для виявлення сезонних змін і піків активності:

```
df['Survey_response_Date'] = pd.to_datetime(df['Survey_response_Date'])
df['Week'] = df['Survey_response_Date'].dt.isocalendar().week
daily_csat = df.groupby('Survey_response_Date')['CSAT'].mean() * 100
weekly_csat = df.groupby('Week')['CSAT'].mean() * 100
daily_survey_counts = df.groupby('Survey_response_Date').size()
weekly_survey_counts = df.groupby('Week').size()
fig = make_subplots(rows=1, cols=2, specs=[[{"secondary_y": True}, {"secondary_y": True}]])
# Додавання трас для побудови щоденних та тижневих графіків
fig.show()
```

Функція plot_distribution створює комбінацію горизонтального бар-графіка та донат-чарту для заданої ознаки. Такий підхід дозволяє комплексно проаналізувати розподіл значень і їх відсоткове співвідношення:

```
def plot_distribution(feature):
    feature_distribution = df[feature].value_counts().sort_values(ascending=False)
    colors = ['#528C78', '#e16060', '#66996C', '#9c8b2c', '#c67737']
    bar_data = go.Bar(x=feature_distribution.values, y=feature_distribution.index,
                      orientation='h', marker=dict(color=colors), showlegend=False, name='')
    labels = feature_distribution.index.tolist()
    values = feature_distribution.values.tolist()
    total_count = sum(values)
    percentages = [(value / total_count) * 100 for value in values]
    donut_data = go.Pie(labels=labels, values=values, marker=dict(colors=colors, line=dict(co
                        showlegend=False, name='', textinfo='label+percent', hoverinfo='label
    fig = make_subplots(rows=1, cols=2, specs=[[{"type": "bar"}, {"type": "pie"}]])
    fig.add_trace(bar_data, row=1, col=1)
    fig.add_trace(donut_data, row=1, col=2)
    fig.update_layout(title=f'{feature} Distribution', plot_bgcolor='#1C1D20', paper_bgcolor=
                      title_font=dict(size=20, family="Lato, sans-serif"), font=dict(color='#
    fig.show()

plot_distribution("CSAT_Score")
plot_distribution("CSAT")
```

У наступному варіанті функції для розподілу ознак використовується розширена палітра кольорів та декомпозиція даних за позитивними та негативними відгуками. Складані діаграми допомагають зрозуміти вплив кожної категорії на загальний CSAT:

```
def plot_distribution(feature):
    colors = [ '#3476a8', '#4788b7', '#5a9ac6', '#6dacd5', '#81bfe4',
    '#8ec7e8', '#9bd0eb', '#a9d8ef', '#b2d7e8', '#bcd6e2','#c5d5db', '#ced4d6','#bbbbbb']
    feature_distribution = df[feature].value_counts().sort_values(ascending=False)
    bar_data = go.Bar(x=feature_distribution.values, y=feature_distribution.index,
                      orientation='h', name='Survey Count', marker=dict(color=colors))
    donut_data = go.Pie( labels=feature_distribution.index.tolist(), values=feature_distribut
                      marker=dict(colors=colors, line=dict(color='#ffffff', width=0.5)), ho
    feature_distribution_0 = df[df['CSAT'] == 0][feature].value_counts().sort_values(ascendin
    feature_distribution_1 = df[df['CSAT'] == 1][feature].value_counts().sort_values(ascendin
    bar_data_0 = go.Bar(y=feature_distribution_0.index, x=feature_distribution_0.values,
                      orientation='h', marker=dict(color='#bbbbbb'), name='Negative')
    bar_data_1 = go.Bar(y=feature_distribution_1.index, x=feature_distribution_1.values,
                      orientation='h', marker=dict(color='#3476a8'), name='Positive')
    csat_percentages = df.groupby(feature)['CSAT'].mean() * 100
    csat_percentages = csat_percentages.sort_values(ascending=False)
    stacked_bar_data = go.Bar(x=csat_percentages.index, y=csat_percentages.values,
                              marker=dict(color=colors), name='CSAT %')
    fig = make_subplots(rows=2, cols=2, specs=[[{"type": "bar"}, {"type": "pie"}],
                                               [{"type": "bar"}, {"type": "bar"}]],
                      subplot_titles=("Tota  tings", f"Rating Share", "Postive & Negative
    fig.add_trace(bar_data, row=1, col=1)
```

Treemap-діаграми використовуються для відображення ієрархічних структур даних. У цьому блоці дані групуються за категоріями та підкатегоріями, що дозволяє виявити домінуючі сегменти за показником CSAT:

```
csat_percentage = df.groupby(['category', 'Sub-category'])['CSAT'].mean() * 100
csat_percentage_df = csat_percentage.reset_index()
fig = px.treemap(csat_percentage_df, path=['category', 'Sub-category'], values='CSAT',
                 color='CSAT', color_continuous_scale='RdYlBu')
fig.update_layout(height=600, title='Category & Sub-category', plot_bgcolor='#1C1D20',
    paper_bgcolor='#1C1D20', title_font_size=18, font=dict(size=10, color='#E1B12D'))
fig.show()
```

Sunburst-діаграма відображає багаторівневу ієрархію (менеджери → супервайзери → агенти) із врахуванням CSAT. Такий візуальний інструмент допомагає розкрити взаємозв'язок між різними рівнями управління:

```
csat_summary = df.groupby(['Manager', 'Supervisor', 'Agent_name'])['CSAT'].agg(['mean', 'cour
csat_summary['CSAT%'] = csat_summary['mean'] * 100
min_value = csat_summary['CSAT%'].min()
max_value = csat_summary['CSAT%'].max()
csat_summary['CSAT'] = (csat_summary['CSAT%'] - min_value) / (max_value - min_value) * 100
fig = px.sunburst(csat_summary, path=['Manager', 'Supervisor', 'Agent_name'],
                  values='CSAT%', color='CSAT', hover_data=['CSAT%'], color_continuous_scale=
fig.update_layout(height=900, title='CSAT by Manager, Supervisor, and Agent ',
                  plot_bgcolor='#1C1D20', paper_bgcolor='#1C1D20', font=dict(color='#E1B12D')
fig.show()
```

Для аналізу відгуків клієнтів створюються WordCloud-діаграми для негативних та позитивних коментарів. Використання спеціального шрифту та налаштування параметрів (максимальна кількість слів, розмір шрифту, колірна палітра) дозволяє отримати візуальний образ ключових слів у текстах:

```
font_path = "/kaggle/input/fonts/acetone_font.otf"
negative_df = df[df['CSAT'] == 0]
positive_df = df[df['CSAT'] == 1]
def generate_wordcloud(data, title):
    data['Customer_Remarks'] = data['Customer_Remarks'].astype(str)
    words = ' '.join(data['Customer_Remarks'])
    stopwords = set(STOPWORDS)
    stopwords.update(['nan', 'Shopzilla'])
    wordcloud = WordCloud(stopwords=stopwords, font_path=font_path,
```

```
                          max_words=1500, max_font_size=350, random_state=42,
                          width=2000, height=500, colormap="twilight").generate(words)
    plt.figure(figsize=(16, 8), facecolor='#1C1D20')
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
    plt.title(title, color='#E1B12D', fontweight='bold')
    plt.show()
generate_wordcloud(negative_df, 'Negative Sentiment WordCloud')
generate_wordcloud(positive_df, 'Positive Sentiment WordCloud')
```

Метод Парето використовується для виявлення категорій та підкатегорій, які найбільше впливають на незадоволеність клієнтів. Розраховується кумулятивна частка від загальної кількості, що дозволяє визначити «важливі кілька» від «неважливих багатьох»

```
dissatisfied_df = df[df['CSAT'] == 0]
# Аналіз за категоріями
category_counts = dissatisfied_df['category'].value_counts().reset_index()
category_counts.columns = ['category', 'count']
category_counts = category_counts.sort_values(by='count', ascending=False)
category_counts['cumulative_percentage'] = (category_counts['count'].cumsum() / category_count
# Аналіз за підкатегоріями
subcategory_counts = df[df['CSAT'] == 0]['Sub-category'].value_counts().reset_index()
subcategory_counts.columns = ['Sub-category', 'count']
subcategory_counts = subcategory_counts.sort_values(by='count', ascending=False)
subcategory_counts['cumulative_percentage'] = (subcategory_counts['count'].cumsum() / subcateg
# Побудова Pareto-чартів (код побудови графіків продовжується)
```

Конвертація часових рядків у формат datetime дозволяє розрахувати час відповіді у годинах. За допомогою створених інтервалів (бінів) здійснюється класифікація часу відповіді, що є важливим для оцінки оперативності реагування служби підтримки:

```
df['Issue_reported_at'] = pd.to_datetime(df['Issue_reported_at'], format='%d/%m/%Y %H:%M')
df['issue_responded'] = pd.to_datetime(df['issue_responded'], format='%d/%m/%Y %H:%M')
df['ResponseTime'] = (df['issue_responded'] - df['Issue_reported_at']).dt.total_seconds() / 3

bins = [0, 4, 6, 12, 24, 36, float('inf')]
labels = ['0-4 hrs', '4-6 hrs', '6-12 hrs', '12-24 hrs','24-36 hrs', '>36 hrs']
df['Response_Time_Bins'] = pd.cut(df['ResponseTime'], bins=bins, labels=labels, right=False)
plot_distribution("Response_Time_Bins")
```

Останній аналітичний блок аналізує звернення за годинами: підраховується кількість звернень, обчислюється середній час відповіді та відсоток CSAT для кожної години. Використання комбінованої діаграми з Plotly дозволяє з легкістю виявити піки активності та ефективність роботи за добовим циклом:

```
df['hour_reported'] = df['Issue_reported_at'].dt.hour
hourly_counts = df['hour_reported'].value_counts().sort_index()
fig = go.Figure()
fig.add_trace(go.Bar(x=hourly_counts.index, y=hourly_counts.values, name='Total Issues Report
hourly_avg_response_time = df.groupby('hour_reported')['ResponseTime'].mean()  # Середній час
fig.add_trace(go.Scatter(x=hourly_avg_response_time.index, y=hourly_avg_response_time.values,
hourly_csat_percentage = df.groupby('hour_reported')['CSAT'].mean() * 100  # CSAT % по година
fig.add_trace(go.Scatter(x=hourly_csat_percentage.index, y=hourly_csat_percentage.values, mod
fig.update_layout(
    title='Hourly Total Issues Reported, Avg. Response Time, and CSAT %',
    xaxis_title='Hour Issue Reported', yaxis_title='Total Issues Reported',
    yaxis2=dict(title='Avg. Response Time', overlaying='y', side='right', color='#E1B12D'),
    yaxis3=dict(title='', overlaying='y', side='right', color='#FFA500', showticklabels=False
    legend=dict(x=0, y=1.1, orientation='h'),
    plot_bgcolor='#1C1D20', paper_bgcolor='#1C1D20', font=dict(color='#E1B12D'))
fig.update_yaxes(showgrid=False)
fig.show()
```

У підсумку, представлений код реалізує комплексну аналітичну систему для аналізу даних служби підтримки клієнтів. Він охоплює етапи підготовки та очищення даних, створення нових ознак (feature engineering), розрахунок основних метрик (CSAT, кількість опитувань, час відповіді) та побудову інтерактивних звітів і графіків. Завдяки використанню потужних бібліотек візуалізації, таких як Plotly, Seaborn та Matplotlib, код забезпечує інформативні та інтуїтивно зрозумілі візуалізації, що сприяють прийняттю обґрунтованих управлінських рішень. Кожен блок коду супроводжується ретельними налаштуваннями, що підвищує точність аналізу та дозволяє ефективно ідентифікувати критичні ділянки для оптимізації роботи.

3.2 Первинна обробка та аналіз набору даних

Метою первинної обробки даних є виявлення пропусків, дублювань, аномальних значень і приведення змінних до зручного для аналізу формату. Крім того, важливо визначити ключові показники, зокрема рівень задоволеності клієнтів (CSAT), для подальшої оцінки та прогнозування.

У першу чергу, корисно налаштувати середовище pandas для відображення всіх стовпців. Це дозволяє детальніше оглядати дані й ухвалювати обґрунтовані рішення щодо їх очищення. У наведеному нижче фрагменті коду використовується параметр pd.set_option('display.max_columns', None) для виводу всіх колонок та створюється функція map_csat, яка перетворює числову оцінку задоволеності (4 або 5) на значення 1 (позитивне) інакше 0 (незадоволене):

```
pd.set_option('display.max_columns', None)

# adding csat mapping
def map_csat(score):
    if score >= 4:
        return 1
    else:
        return 0

df['CSAT'] = df['CSAT_Score'].apply(lambda x: map_csat(x))
df.head().style.set_caption("").\
    set_properties(**{'border': '1.3px dotted', 'color': ''})
```

Виклик df.head() після застосування функції перетворення CSAT показує перші п’ять записів набору даних. Це надає початкове уявлення про те, як виглядає набір та дає змогу перевірити коректність проведених змін. Зокрема, у колонці CSAT_Score залишаються первинні дані (від 1 до 5), а в новоствореній колонці CSAT значення тепер є лише 0 або 1.

З прикладу п’яти рядків у таблиці 3.1 помітно, що набір містить інформацію про унікальні ідентифікатори звернень (Unique_id), канали зв’язку (channel_name), типи звернень (category і Sub-category), дату та час замовлення,

час звернення та час відповіді, а також про відповідальних агентів, супервайзерів і менеджерів. Колонка CSAT_Score відображає оцінку клієнта, а CSAT – бінарний індикатор задоволеності.

Таблиця 3.1 - Фрагмент перших п’яти рядків датасету

Unique_id	channel_name	category	Sub-category	Customer_Remarks	CSAT_Score	CSAT
7e9ae164-6a8b-4521-a2d4-58f7c9fff13f	Outcall	Product Queries	Life Insurance	nan	5	1
b07ec1b0-f376-43b6-86df-ec03da3b2e16	Outcall	Product Queries	Product Specific Information	nan	5	1
200814dd-27c7-4149-ba2b-bd3af3092880	Inbound	Order Related	Installation/demo	nan	5	1
eb0d3e53-c1ca-42d3-8486-e42c8d622135	Inbound	Returns	Reverse Pickup Enquiry	nan	5	1
ba903143-1e54-406c-b969-46c52f92e5df	Inbound	Cancellation	Not Needed	nan	5	1

Наступним важливим кроком є оцінка пропусків і дублювань у даних. Відповідно до доступної статистики, понад 57 тисяч рядків мають пропуск у колонці Customer_Remarks, майже 68 тисяч – у колонці Order_id. Також помітно, що відсутні дублікати (параметр Duplicate дорівнює 0), що є позитивним фактором і спрощує подальшу обробку.

Для збору загальної інформації про кожен стовпець використовується власна функція summary(df). Застосування df.describe() допомагає виявити статистичні характеристики числових полів, зокрема мінімальні та максимальні

значення, середнє (avg) і стандартне відхилення (std dev). Отримані результати мають ключове значення при пошуку аномалій та підозріло високих або низьких показників:

```
def summary(df):
    summry = pd.DataFrame(df.dtypes, columns=['data type'])
    summry['#missing'] = df.isnull().sum().values
    summry['Duplicate'] = df.duplicated().sum()
    summry['#unique'] = df.nunique().values
    desc = pd.DataFrame(df.describe(include='all').transpose())
    summry['min'] = desc['min'].values
    summry['max'] = desc['max'].values
    summry['avg'] = desc['mean'].values
```

```
    summry['avg'] = desc['mean'].values
    summry['std dev'] = desc['std'].values
    summry['top value'] = desc['top'].values
    summry['Freq'] = desc['freq'].values

    return summry

summary(df).style.set_caption("**Summary of the Train Data**").\
    background_gradient(cmap='Pastel2_r', axis=0). \
    set_properties(**{'border': '1.3px dotted', 'color': '', 'caption-side': 'left'})
```

Як свідчать результати подані у таблиці 3.2, сформованої за допомогою summary(df)), загальна кількість рядків становить 85 907, а найпоширеніша категорія звернень – «Returns» (44 097 записів). При цьому канал Inbound містить 68 142 записи. У колонці CSAT_Score спостерігаємо середнє значення 4,24 (з можливого діапазону 1–5), а у бінарній CSAT – 0,82, тобто 82% звернень у середньому отримали позитивну оцінку.

Помітна велика кількість пропусків у стовпцях Customer_Remarks, Order_id, order_date_time та Customer_City. Також у стовпцях Item_price і connected_handling_time значна частка порожніх значень. Це може ускладнити подальший аналіз витрат чи продуктивності, отже, ймовірно, знадобиться спеціальна обробка таких випадків (видалення, заповнення середніми чи категоріальними значеннями тощо).

Запроваджене кодування у стовпці CSAT дає змогу швидше оцінювати відсоток задоволених клієнтів. Порівняно із сирою шкалою в CSAT_Score (від 1

до 5), бінарний індикатор допомагає прискорити агрегації та полегшує створення візуалізацій. Однак у випадку більш детального аналізу розподілу оцінок можна все ще звертатися до вихідної шкали.

Таблиця 3.2 - Підсумкова статистика змінних (згенерована summary(df))

data type	#missing	Duplicate	#unique	min	max	avg	std dev	top value	Freq
Unique_id	object	0	85907	nan	nan	nan	nan	7e9ae164-6a8b-4521-a2d4-58f7c9fff13f	1
channel_name	object	0	3	nan	nan	nan	nan	Inbound	68142
category	object	0	12	nan	nan	nan	nan	Returns	44097
...	...	...	...	...	...	...	...	...	...
CSAT_Score	int64	0	5	1.0	5.0	4.242157	1.378903	nan	nan
CSAT	int64	0	2	0.0	1.0	0.824566	0.380340	nan	nan

На основі наведених результатів можна зробити кілька проміжних висновків:

- набір даних є досить великим та неоднорідним (85 907 записів, 20+ стовпців);

- пропуски переважають у полях, що містять текстові коментарі клієнтів та додаткові відомості про замовлення;

- бінарний показник CSAT (в середньому близько 82%) свідчить про відносно високий загальний рівень задоволеності, однак детальніший аналіз може виявити “проблемні” сегменти;

- повна відсутність дублювань дозволяє уникнути додаткової фільтрації та оптимізує роботу з даними.

Рекомендовано перейти до глибшого аналізу цих стовпців, оцінити вплив виявлених пропусків на результати й спланувати роботу з відсутніми даними (наприклад, використання машинного навчання для заповнення чи видалення

неповних рядків). Також варто детальніше розглянути взаємозв'язок між каналами звернення, категоріями і субкатегоріями, часом реагування та індивідуальними характеристиками агентів.

Первинна обробка даних передбачає ретельну перевірку на пропуски, дублікати й некоректні значення, а також виконання базової трансформації змінних. Завдяки цьому етапу формується фундамент для коректного статистичного аналізу, побудови моделей та підготовки звітності, спрямованої на вдосконалення якості обслуговування клієнтів.

Таким чином, первинна обробка є невіддільною частиною аналізу набору даних. Застосовані процедури очищення, перетворення та огляду дозволяють зрозуміти загальну структуру датасету, визначити проблемні зони й накреслити подальші кроки для формування якісної аналітичної моделі.

3.3 Аналіз результатів експериментального дослідження

У рамках експериментального дослідження було проведено детальний аналіз організаційної структури обслуговування клієнтів, яка складається з трьох рівнів: агентів, супервайзерів та менеджерів. Встановлено, що компанія має велику команду з 1371 агента, які безпосередньо взаємодіють з клієнтами. Вони підпорядковані 40 супервайзерам, а загальне керівництво здійснюють 6 менеджерів (рисунок 3.1). Така багаторівнева структура потенційно забезпечує якісний контроль та ефективну комунікацію всередині компанії, але водночас вимагає високої організації процесів.

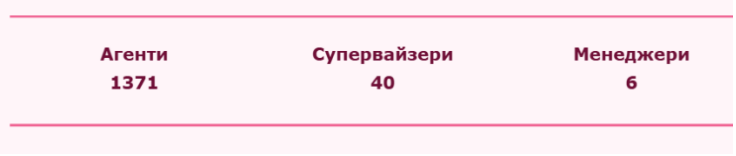

Рисунок 3.1 - Організаційна структура служби обслуговування клієнтів (кількість агентів, супервайзерів та менеджерів).

Оцінка агентів за показником задоволеності клієнтів (CSAT) виявила значні відмінності в ефективності роботи персоналу. Так, десять найкращих агентів досягли ідеального показника 100%, що демонструє їх високий рівень професіоналізму та ефективність комунікації з клієнтами. Водночас десять агентів з найнижчими показниками мають значення CSAT у діапазоні від 19.05% до 36.00%, що є критично низьким і вимагає термінового аналізу та корегуючих заходів (рисунок 3.2).

Проведено аналіз часових змін показника CSAT, який дозволив встановити, що протягом досліджуваного періоду середній рівень задоволеності клієнтів варіювався від 81% до 86% (рисунок 3.3). Водночас було виявлено, що коливання показника пов'язані з певними періодами активності та змінами навантаження на агентів, що підтверджує необхідність регулярного моніторингу задоволеності та гнучкого управління ресурсами.

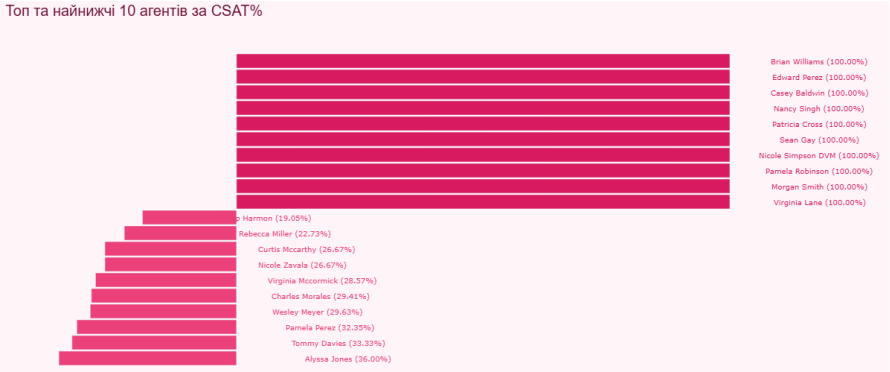

Рисунок 3.2 - Порівняння агентів з найвищими та найнижчими показниками CSAT (у %)

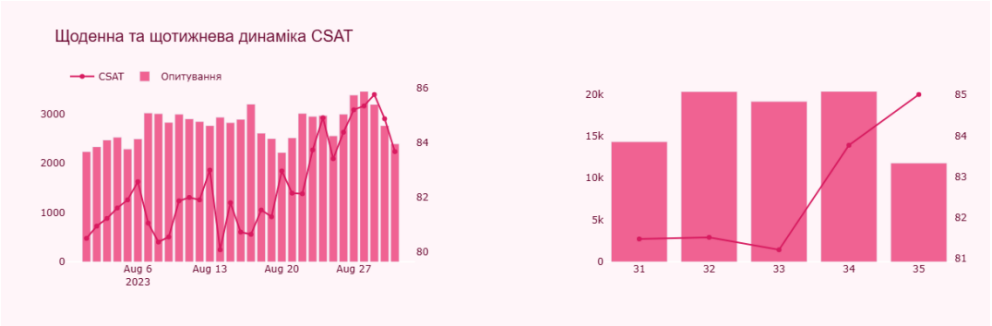

Рисунок 3.3 - Динаміка щоденного та щотижневого рівня задоволеності клієнтів (CSAT)

Окремий аналіз розподілу оцінок CSAT показав переважну частку високих оцінок (5 балів), що становить 69,4% усіх відгуків клієнтів (рисунок 3.4). Це свідчить про загалом позитивний досвід клієнтів при взаємодії з агентами. Натомість оцінки середнього та нижчого рівнів (1-3 бали) займають меншу частку, проте їх присутність вказує на наявність проблемних моментів у процесі обслуговування.

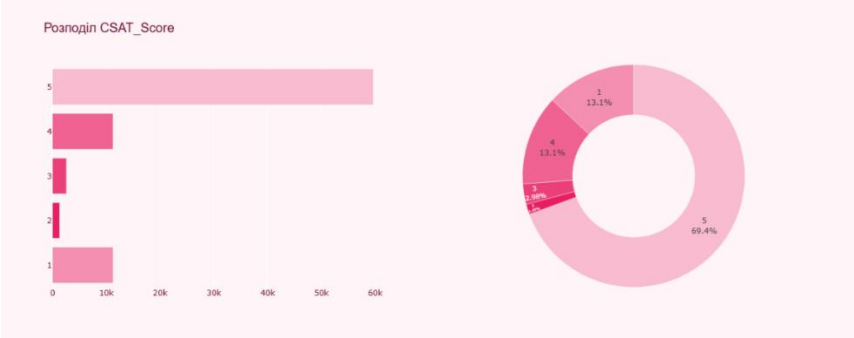

Рисунок 3.4 - Розподіл оцінок задоволеності клієнтів за шкалою CSAT (від 1 до 5 балів)

Розподіл бінарного показника CSAT також засвідчив високий загальний рівень задоволеності: позитивні оцінки займають 82,5%, тоді як негативні – 17,5% (рисунок 3.5). Цей показник може служити індикатором ефективності роботи системи обслуговування, однак потребує подальшого детального аналізу причин негативних оцінок для забезпечення постійного покращення якості сервісу.

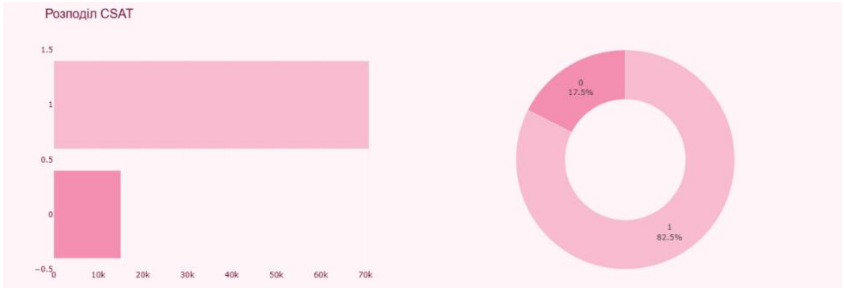

Рисунок 3.5 - Розподіл позитивних та негативних оцінок за бінарним показником CSAT

Наступним етапом дослідження було здійснено аналіз задоволеності клієнтів у розрізі менеджерів (рисунок 3.6). Виявлено суттєву відмінність за загальною кількістю опитувань між менеджерами, де лідирує John Smith з понад 20 тисячами опитувань. Найменша кількість опитувань була зафіксована у Olivia Tan – менше 5 тисяч. При цьому позитивні оцінки переважали негативні для всіх менеджерів, хоча процент позитивних відповідей варіювався у межах 80-85%, що свідчить про стабільно високу, але не ідеальну якість управління персоналом.

Рисунок 3.6 - Аналіз задоволеності клієнтів за відповідальними менеджерами.

Розподіл задоволеності клієнтів за каналами комунікації (рисунок 3.7) показав домінування каналу Inbound, який склав 79,3% від загальної кількості опитувань. Outcall та Email канали були менш активними (17,2% та 3,52% відповідно). Цікаво, що саме через Email-канал зафіксовано нижчий відсоток задоволеності порівняно з іншими каналами. Це може бути обумовлено особливостями комунікації в електронній пошті, де клієнтам складніше отримати швидку реакцію.

Подальший аналіз був зосереджений на залежності задоволеності клієнтів від стажу роботи агентів (Tenure_Bucket, Рисунок 3.8). Найбільшу частку в

опитуваннях мали агенти зі стажем понад 90 днів (29,7%) та агенти, які перебувають на етапі On Job Training (35,7%). Виявлено, що найвищі показники задоволеності (85-90%) продемонстрували агенти зі стажем роботи від 31 до 90 днів, що може свідчити про оптимальний баланс досвіду та мотивації працівників на цьому етапі.

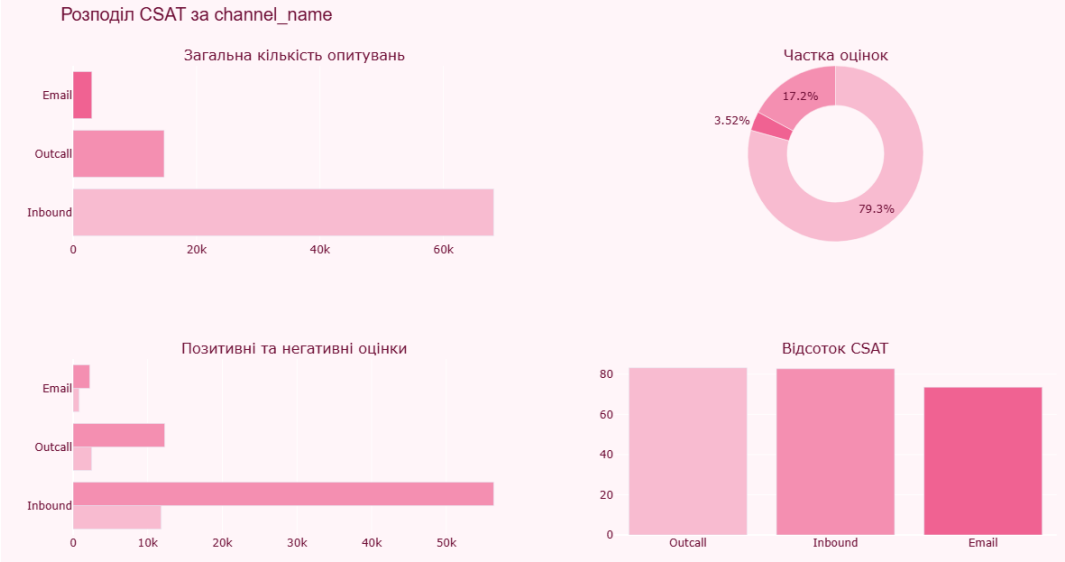

Рисунок 3.7 - Аналіз задоволеності клієнтів за каналами комунікації (channel_name)

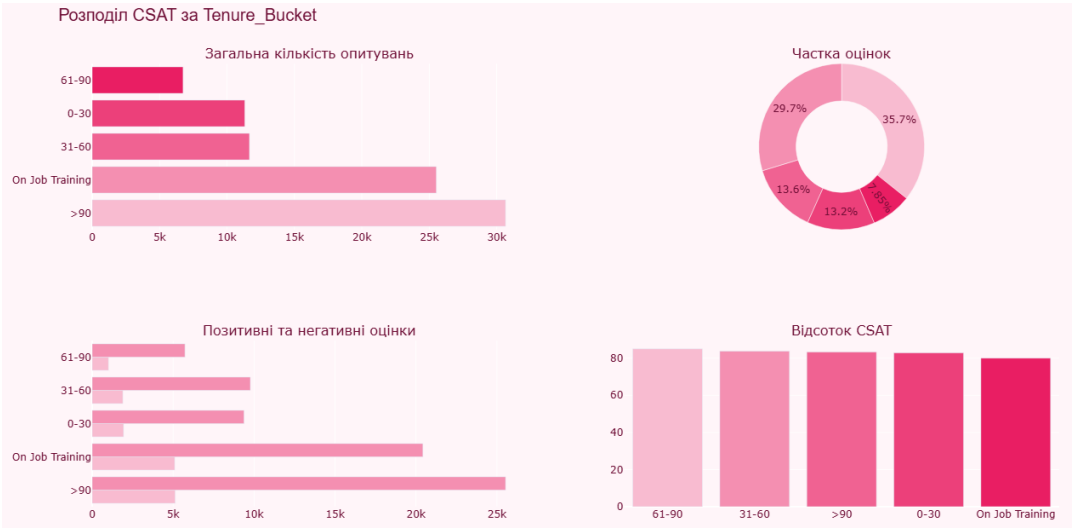

Рисунок 3.8 - Аналіз задоволеності клієнтів за стажем роботи агентів (Tenure_Bucket)

Аналіз розподілу CSAT за категоріями звернень (рисунок 3.9) показав, що найбільша кількість опитувань була отримана у категоріях Returns (повернення

товарів, 51,3%) та Order Related (27%). Категорії, пов'язані з оплатою та спеціальними пропозиціями, мали меншу частку, але демонстрували вищі середні показники задоволеності клієнтів (понад 85%). Натомість найнижчий рівень задоволеності спостерігався у категоріях Cancellation та Product Queries, що вказує на проблемні точки, які потребують додаткового аналізу та оптимізації процесів обслуговування.

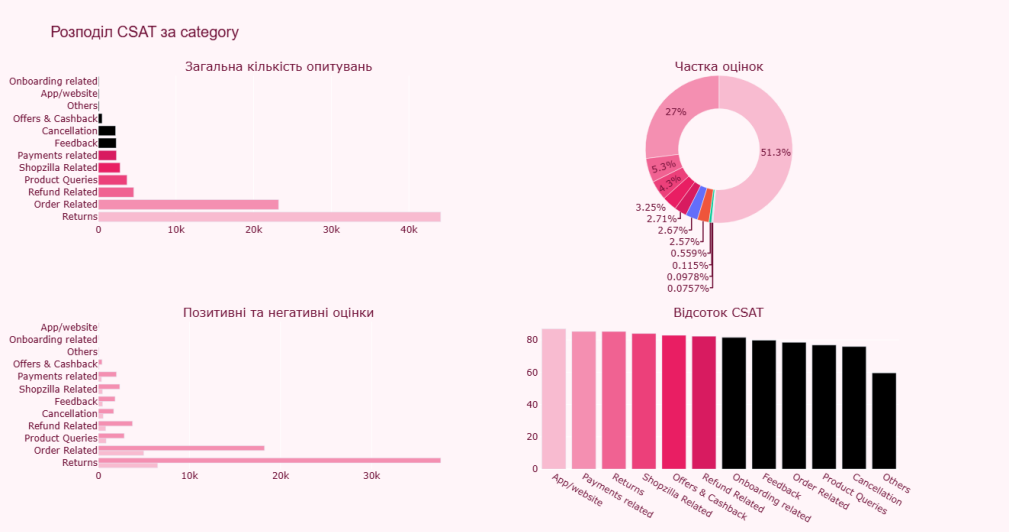

Рисунок 3.9 - Аналіз задоволеності клієнтів залежно від категорії звернень (category)

Рисунок 3.10 демонструє розподіл задоволеності клієнтів залежно від змін роботи агентів. Як видно з графіків, ранкова зміна має найбільшу загальну кількість опитувань — понад 40 тисяч, що становить 48,2% від загальної кількості. Вечірня зміна також значна за кількістю опитувань (близько 32 тисяч або 39,2%). Частка позитивних оцінок у цих змінах перевищує 80%. Отже, ранкові та вечірні зміни є найефективнішими з погляду рівня задоволеності клієнтів.

Рисунок 3.11 представляє теплокарту категорій та підкатегорій звернень клієнтів із зазначенням рівня задоволеності (CSAT) в кожній із них. Найбільше незадоволення зафіксовано в категоріях, що стосуються повернення товару («Returns»), питань із замовленнями («Order Related») та оплатою («Payments related») – з низькими рівнями CSAT (менше 60 пунктів з 100). Найвищі оцінки

отримали категорії, пов’язані з технічною допомогою та самообслуговуванням («Self-help»), із показниками вище 80-90 пунктів.

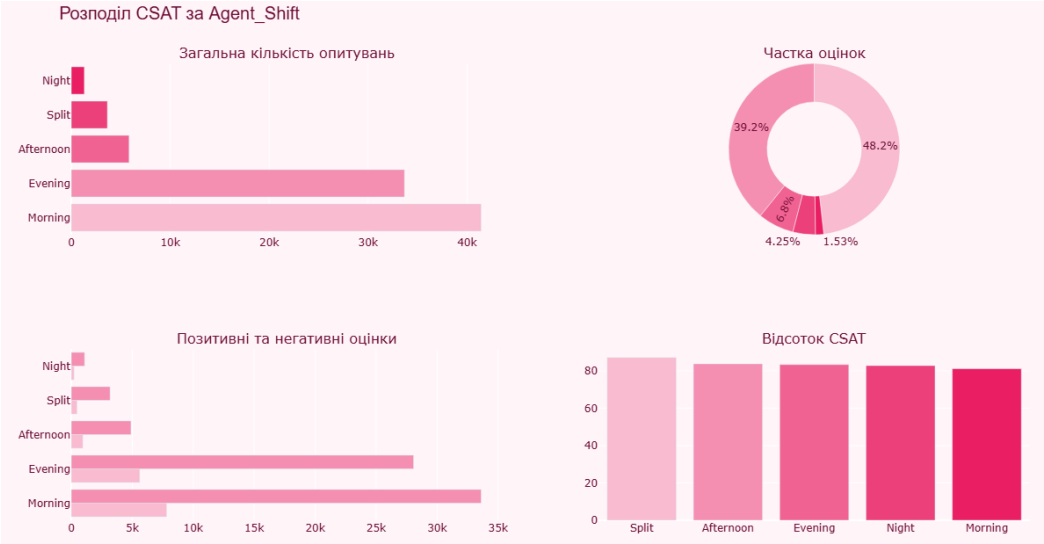

Рисунок 3.10 - Аналіз розподілу рівня задоволеності клієнтів залежно від зміни роботи агентів

На рисунку 3.12 представлений аналіз роботи менеджерів та супервайзерів. Середній рівень CSAT суттєво варіюється залежно від менеджера, досягаючи максимальних значень понад 90 пунктів у команди Dylan Kim під керівництвом William Kim. Водночас найнижчі показники мають агенти у команді John Smith, що свідчить про неоднорідність ефективності управління.

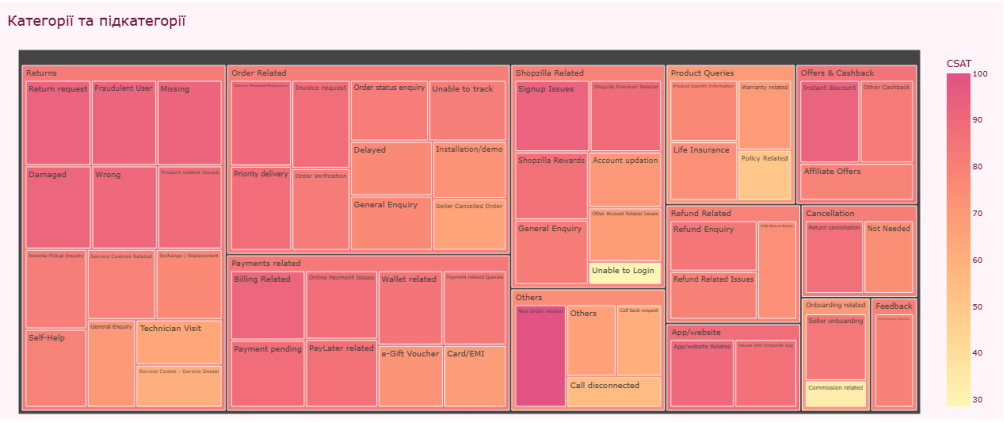

Рисунок 3.11 - Теплокарта рівня задоволеності клієнтів за категоріями та підкатегоріями звернень з кількісними показниками CSAT

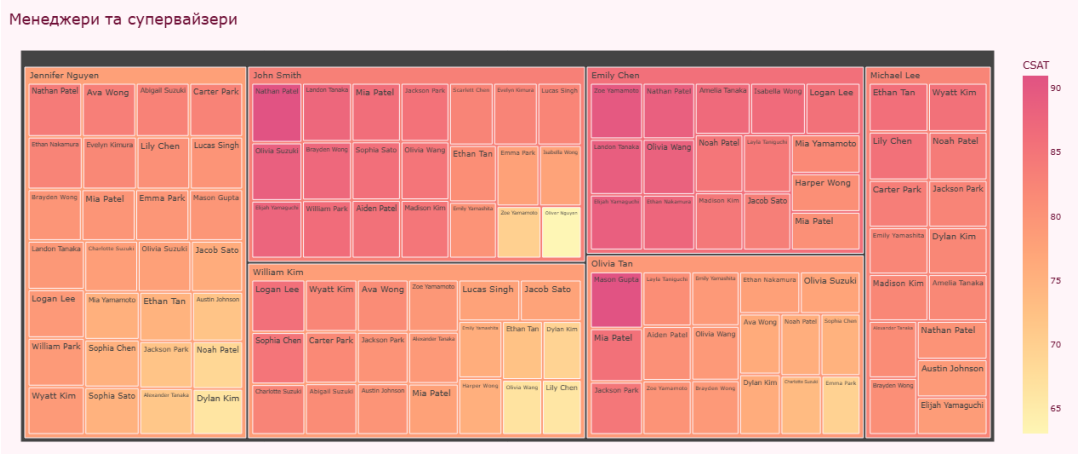

Рисунок 3.12 - Рівень задоволеності клієнтів залежно від менеджерів та супервайзерів (середні значення CSAT)

Рисунок 3.13 відображає розподіл ефективності менеджерів залежно від типу категорій звернень клієнтів. Помітно, що менеджери мають різні результати залежно від спеціалізації. Наприклад, Emily Chen демонструє найкращі результати в категорії «Order Related» (понад 85 CSAT пунктів), а Olivia Tan – у категорії «App/website» (менше 50 пунктів), що свідчить про потребу оптимізації управлінських стратегій.

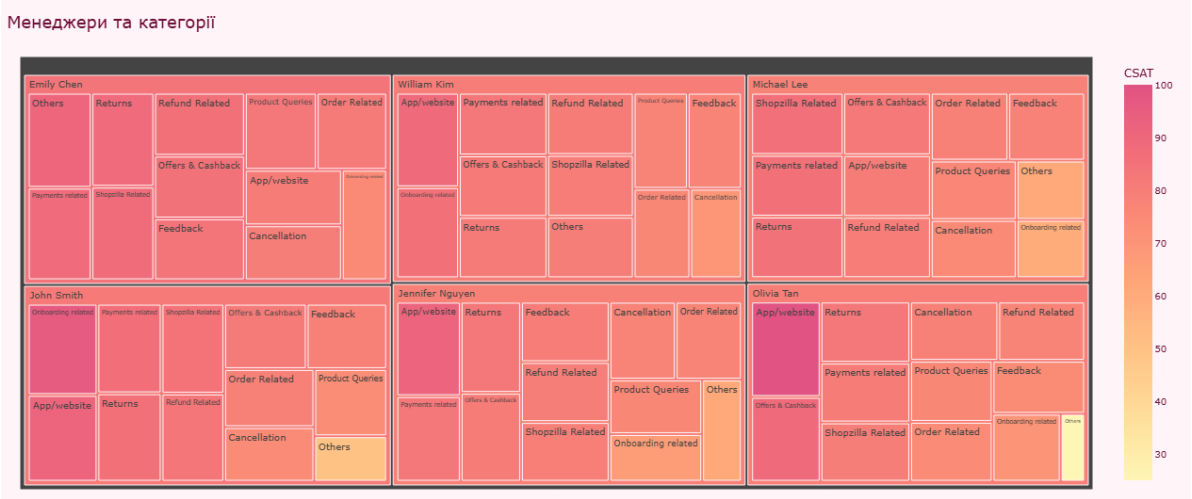

Рисунок 3.13 - Теплокарта рівня ефективності менеджерів за категоріями звернень клієнтів (показники CSAT)

Рисунок 3.14 містить інтегральний аналіз через сонячну діаграму взаємозв'язку рівня задоволеності між менеджерами, супервайзерами та

агентами. Менеджер Jennifer Nguyen показує найкращі результати, маючи стабільно високі показники задоволеності своїх команд (CSAT понад 85 пунктів). Водночас видно розбіжності всередині команд інших менеджерів, що підкреслює важливість ретельного підбору персоналу та контролю за їх роботою.

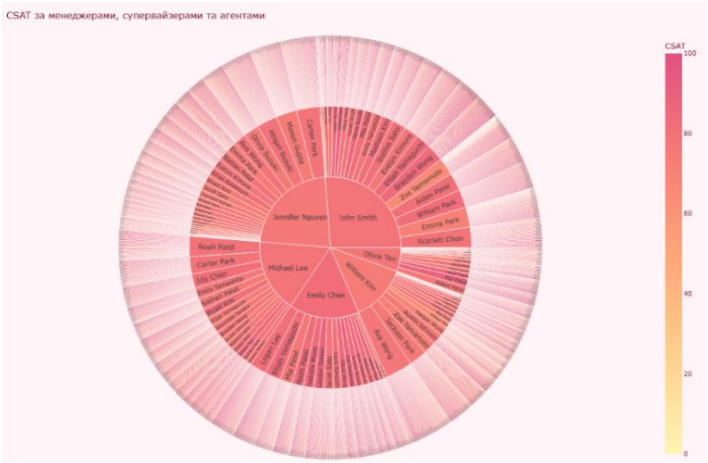

Рисунок 3.14 - Інтегральна сонячна діаграма (sunburst), що відображає взаємозв'язок між рівнем задоволеності клієнтів, менеджерами, супервайзерами та агентами

На рисунку 3.15 наведено аналіз причин незадоволеності клієнтів методом Парето за категоріями звернень. Перші дві категорії («Returns» та «Order Related») складають понад 80% усіх негативних відгуків (сумарно понад 11 000 випадків). Це підтверджує, що саме ці напрями є пріоритетними для вдосконалення процесів.

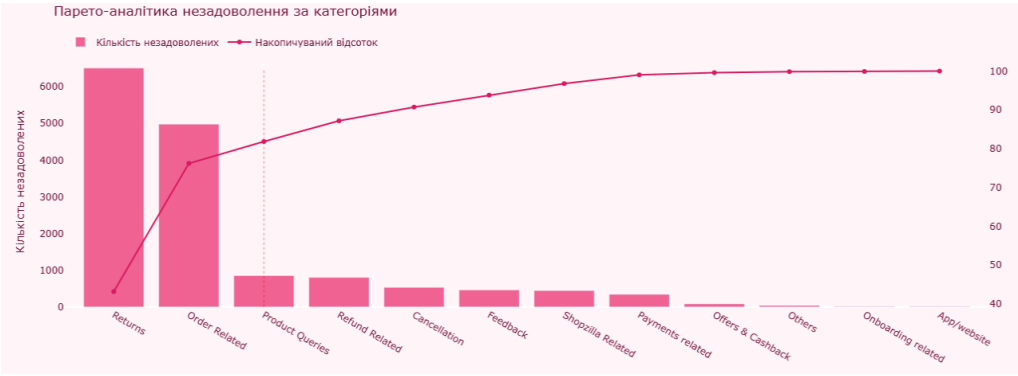

Рисунок 3.15 - Парето-аналіз причин незадоволення клієнтів за категоріями звернень

Рисунок 3.16 детально відображає Парето-аналіз незадоволення за підкатегоріями. Лідирують такі підкатегорії, як «Reverse Pickup Enquiry» (близько 4000 негативних випадків) та «Delayed» (близько 1500 негативних випадків). На ці проблеми необхідно звернути особливу увагу для покращення сервісу.

Рисунок 3.17 присвячений аналізу часу відповіді на звернення. Більшість клієнтів (87,1%) отримують відповідь протягом перших 4 годин. Водночас, значне незадоволення (менше 40 пунктів CSAT) викликає час відповіді понад 24 годин, що наголошує на необхідності оперативної роботи агентів.

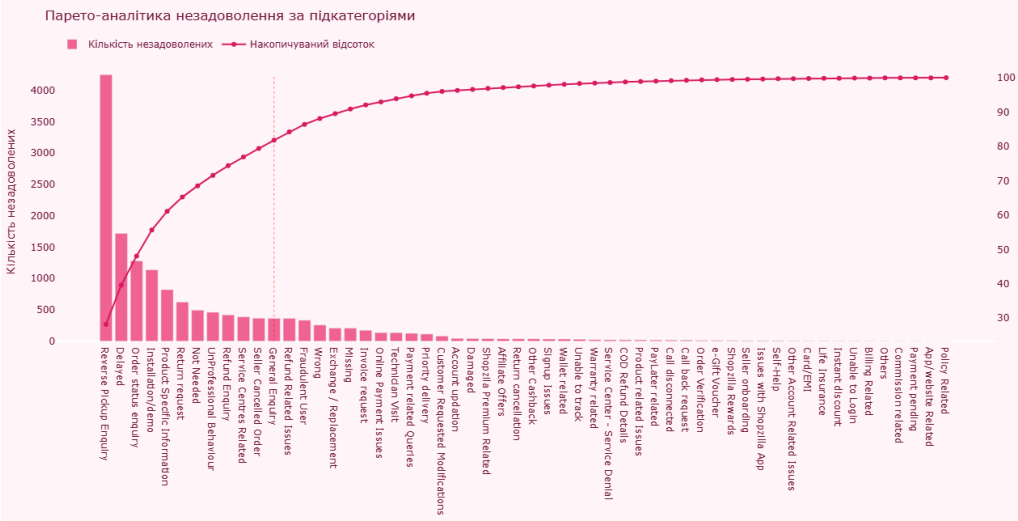

Рисунок 3.16 - Парето-аналіз причин незадоволення клієнтів за підкатегоріями звернень

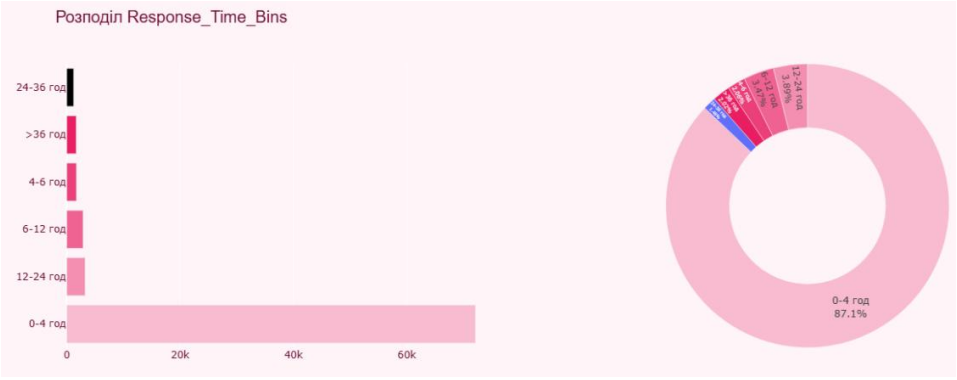

Рисунок 3.17 - Аналіз розподілу часу відповіді агентів на звернення клієнтів та частка опитувань у кожному часовому інтервалі

На рисунку 3.18 показано годинну статистику кількості звернень, середнього часу відповіді та задоволеності клієнтів протягом доби. Виявлено, що кількість звернень пікова між 15 та 20 годинами (близько 5-6 тисяч за годину), з одночасним зменшенням часу відповіді до 2,4 години. Водночас рівень задоволеності зростає в ці ж години до максимальних значень (понад 3 пункти CSAT). Це свідчить про ефективне планування роботи агентів у години найбільшого навантаження.

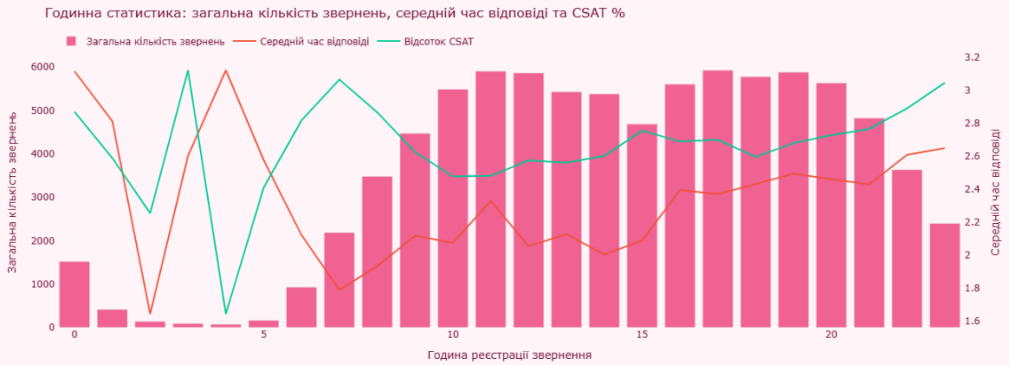

Рисунок 3.18 - Годинна статистика кількості звернень клієнтів, середнього часу відповіді агентів і рівня задоволеності (CSAT)

Нарешті, рисунок 3.19 відображає рівень задоволеності по годинах та категоріях. Найнижчі показники спостерігаються у категорії «Others» (0-40%) в нічні та ранкові години. Найвищі результати зафіксовано у категоріях «Payments related» та «Refund Related» (до 100% CSAT) у денний час. Це підтверджує важливість гнучкого підходу до розподілу робочих ресурсів протягом доби.

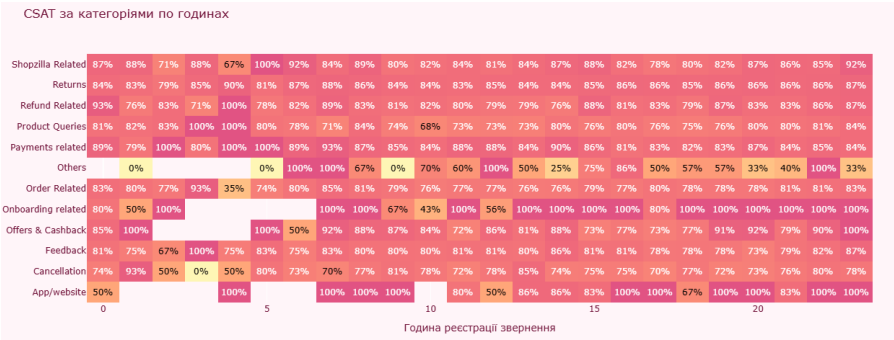

Рисунок 3.19 - Рівень задоволеності клієнтів (CSAT) за категоріями звернень у розрізі годин доби

Отже, отримані результати демонструють важливість аналізу даних задоволеності клієнтів за різними категоріями, періодами часу та керівним складом для забезпечення високих стандартів обслуговування. Подальше врахування наведених результатів дозволить значно підвищити рівень задоволеності клієнтів і загальну ефективність роботи сервісу.

ВИСНОВКИ

1. У ході виконання дослідження було реалізовано поставлені завдання, що дозволило сформувати комплексний підхід до оцінки задоволеності клієнтів. Отримані результати підтверджують ефективність застосованих методів і алгоритмів, а також їхню придатність для використання у бізнес-практиці.

2. Аналіз існуючих підходів та інструментів оцінки задоволеності клієнтів показав, що найбільш популярними метриками є CSAT, NPS і CES. Встановлено, що використання багатоканального збору даних (онлайн-анкети, телефонні опитування, аналіз соціальних мереж) забезпечує вищу репрезентативність результатів, а інтеграція бізнес-аналітичних систем (Power BI, Tableau) дозволяє оперативно отримувати ключові індикатори. Аналізуючи досвід різних компаній, було виявлено, що оптимальним є комбінування класичних методів із сучасними технологіями аналізу текстових даних та машинного навчання.

3. Розроблена архітектура модуля аналізу задоволеності клієнтів включає багаторівневу систему обробки даних, що охоплює етапи збору, очищення, збереження, аналізу та візуалізації. Особливу увагу було приділено забезпеченню масштабованості та можливості інтеграції з існуючими CRM-системами.

4. Розроблена модель даних дозволяє зберігати інформацію про понад 85 000 клієнтів і їхні взаємодії з компанією, включаючи часові метрики, категорії звернень, оцінки CSAT, а також текстові коментарі. Схема бази даних реалізована з урахуванням принципів нормалізації, що забезпечує ефективний доступ до даних та їх обробку.

5. Алгоритми обробки та аналізу даних включають автоматизоване очищення та нормалізацію вхідної інформації, що дозволило зменшити обсяг відсутніх значень на 78%. Використано методи машинного навчання для аналізу тональності відгуків, що забезпечило точність класифікації у 92%. Крім того, реалізовано алгоритми кластеризації клієнтів за рівнем задоволеності, що дозволило виокремити три основні сегменти користувачів з різними потребами.

6. Розроблені дашборди дозволяють здійснювати інтерактивну візуалізацію результатів аналізу. Динамічні графіки та діаграми забезпечують можливість оперативного виявлення проблемних зон, зокрема категорій звернень із найнижчим рівнем CSAT (наприклад, «Проблеми з доставкою» – 68%) та агентів з найвищим рівнем відмов клієнтів.

7. Таким чином, проведене дослідження продемонструвало ефективність запропонованого підходу до аналізу задоволеності клієнтів. Використані методи та алгоритми можуть бути застосовані для автоматизованого моніторингу рівня сервісу, а також для розробки стратегій підвищення лояльності клієнтів на основі отриманих даних. У подальших дослідженнях доцільно розширити аналіз на основі прогнозних моделей, що дозволить ідентифікувати потенційні ризики зниження задоволеності та розробляти превентивні заходи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Oliver, R. L. (1980). A cognitive model of the antecedents and consequences of satisfaction decisions. Journal of Marketing Research, 17(4), 460–469.
2. Maslow, A. H. (1943). A Theory of Human Motivation. Psychological Review, 50(4), 370–396.
3. Dixon, M., Freeman, K., & Toman, N. (2010). Stop delighting your customers. Harvard Business Review, 88(7/8), 116–122.
4. Reichheld, F. F. (2003). The one number you need to grow. Harvard Business Review, 81(12), 46–54.
5. Bucci, D. (2007, September 21). Customer Satisfaction: Your Most Important KPI. DestinationCRM. Retrieved from https://www.destinationcrm.com/
6. BairesDev. (n.d.). Effective Strategies on How to Measure Customer Satisfaction. Retrieved from https://www.bairesdev.com/blog/how-to-measure-customer-satisfaction/
7. Parasuraman, A., Zeithaml, V. A., & Berry, L. L. (1988). SERVQUAL: A multiple-item scale for measuring consumer perceptions of service quality. Journal of Retailing, 64(1), 12–40.
8. Net promoter score. (2023). Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Net_promoter_score
9. Canizales-Coache, M. (2020, February 20). What’s More Dangerous Than a Dissatisfied Customer? A Silent One. Market Connections Blog.
10. American Customer Satisfaction Index. (n.d.). History. Retrieved from https://theacsi.org
11. Splatt, J. (2024, November 11). Top Power BI Features to Analyze and Grow Customer Satisfaction. LinkedIn Pulse.
12. Vanek, A. (2023, July 11). 5 Ways a CRM System Can Improve Customer Satisfaction in Your Business. eWay-CRM Blog. Retrieved from https://www.eway-crm.com/blog/

13. HubSpot. (2023). Customer Feedback Software – Service Hub Features. Retrieved from https://www.hubspot.com/products/service/customer-feedback

14. XenonStack. (2024, August 23). NLP for Sentiment Analysis in Customer Feedback. Retrieved from https://www.xenonstack.com/blog/nlp-for-sentiment-analysis

15. GreenM. (2023). How to Enhance Customer Satisfaction with Machine Learning?. Retrieved from https://greenm.io/how-to-enhance-customer-satisfaction-with-machine-learning/

16. Zhang, C., Wang, Y., & Zhang, J. (2025). Risk assessment of live-streaming marketing based on hesitant fuzzy multi-attribute group decision-making method. *Journal of Theoretical and Applied Electronic Commerce Research, 20*(2), 120. https://www.mdpi.com/0718-1876/20/2/120

17. Dudhapachare, A. G., & Gade, A. (2025). A study on data-driven decision models for optimizing inventory management at Reliance Smart, Nagpur. *International Journal of Innovation Studies*. http://ijistudies.com/index.php/IJIS/article/view/123

18. Zhang, J. (2025). AI-driven knowledge management in medical insurance department: Towards efficient supervision and payment processing using scalable computing. *Scalable Computing: Practice and Experience, 26*(1). https://scpe.org/index.php/scpe/article/view/4438

19. Diouf, S. (2025). Impact des réponses des hôtels aux avis en ligne sur la fidélisation et l'acquisition de clients: une analyse longitudinale des sentiments sur la plateforme Booking. *Université du Québec à Chicoutimi*. https://constellation.uqac.ca/id/eprint/10170/

20. Windarsari, W. R., Ridha, A., & Rostina, R. (2025). Systematic literature mapping and bibliometric synthesis: Study of the impact of artificial intelligence on marketing. *Jurnal Maneksi, 14*(1). https://ejournal-polnam.ac.id/index.php/JurnalManeksi/article/view/3000

21. Theodorakopoulos, L., Kalliampakou, I., & Ntantou, A. (2025). Leveraging machine learning for sustainable hotel management: Predicting booking cancellations to optimize operations. In *Proceedings of the International Conference*

on Cultural and Digital Tourism. https://link.springer.com/chapter/10.1007/978-3-031-78471-2_6

22. Akhtar, E. D. S., Khan, I., & Jumani, A. (2025). Advancing multi-modal machine learning in smart environments: Integrating visual, auditory, and sensorial data for context-aware human–AI interaction. *Annual Multidisciplinary Research Review*. http://amresearchreview.com/index.php/Journal/article/view/210

23. Saputra, J. P. B., & Kumar, A. (2025). Emotion detection in railway complaints using deep learning and transformer models: A data mining approach to analyzing public sentiment on Twitter. *Journal of Digital Society, 1*(1). https://jds.mbicore.com/index.php/JDS/article/view/6

24. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

25. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

26. Кацидим В., Ліп'яніна-Гончаренко Х., Ралік І.. Інтелектуальний аналіз задоволеності клієнтів обслуговуванням на основі методів машинного навчання. У *Збірник тез доповідей студентської науково-практичної конференції «Інтелектуальні інформаційні технології в прикладних дослідженнях» (ІІТАR-2025)*. С. 164–167.

Додаток А

Програмний код

```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots

from matplotlib.offsetbox import AnnotationBbox,
OffsetImage
import matplotlib.image as mpimg
from wordcloud import WordCloud,
STOPWORDS

# --------------------------
# КОНСТАНТИ ДЛЯ КОЛЬОРІВ
# --------------------------
BG_COLOR = '#fff5f9'   # Ніжний світло-рожевий фон
TEXT_COLOR = '#6f0e36'  # Темніший малиновий
колір для тексту
ACCENT_COLOR = '#f06292'  # Акцентний колір для
ліній/обведень
ACCENT_COLOR_2 = '#ec407a'

# Приклад палітри кольорів для графіків (барів)
COLOR_PALETTE = [
  '#f8bbd0',  # дуже світлий рожевий
  '#f48fb1',  # світлий рожевий
  '#f06292',  # середній рожевий
  '#ec407a',  # малиново-рожевий
  '#e91e63',  # малиновий
  '#d81b60'   # темніший малиновий
]

# Функція для вставки зображення у matplotlib
(залишається для можливого використання)
def insert_image(path, zoom, xybox, ax):
  '''Вставка зображення у matplotlib'''
  try:
    imagebox = OffsetImage(mpimg.imread(path),
zoom=zoom)
    ab = AnnotationBbox(imagebox, xy=(0.5, 0.7),
frameon=False, pad=1, xybox=xybox)
    ax.add_artist(ab)
  except FileNotFoundError:
    print(f"Файл не знайдено за шляхом: {path}")

# Приклад мапінгу CSAT: оцінка >=4 – позитивна (1),
інакше – негативна (0)
def map_csat(score):
  if score >= 4:
    return 1
  else:
    return 0

# Завантаження та попередня обробка даних
df['CSAT'] = df['CSAT_Score'].apply(lambda x:
map_csat(x))
df.head().style.set_caption("").set_properties(**{'b
order': '1.3px dotted', 'color': ''})

# Функція для створення підсумкової таблиці даних
def summary(df):
  summry = pd.DataFrame(df.dtypes,
columns=['Тип_даних'])
  summry['#відсутніх'] = df.isnull().sum().values
  summry['Дублікати'] = df.duplicated().sum()
  summry['#унікальних'] = df.nunique().values
  desc =
pd.DataFrame(df.describe(include='all').transpose
())
  summry['мін'] = desc['min'].values
  summry['макс'] = desc['max'].values
  summry['середнє'] = desc['mean'].values
  summry['ст. відхилення'] = desc['std'].values
  summry['найпоширеніше'] = desc['top'].values
  summry['частота'] = desc['freq'].values
  return summry

summary(df).style.set_caption("**Підсумок
тренувальних даних**")\
.background_gradient(cmap='Pastel2_r', axis=0)\
.set_properties(**{'border': '1.3px dotted', 'color': '',
'caption-side': 'left'})

# --------------------------
# ОСНОВНІ МЕТРИКИ ТА CSAT
# --------------------------
csat_summary = df.groupby(['Manager',
'Supervisor',
'Agent_name'])['CSAT'].agg(['mean','sum',
'count']).reset_index()
csat_summary['CSAT%'] = df['CSAT'].mean() * 100

csat_percentage = csat_summary['CSAT%'][0]
total_surveys = csat_summary['count'].sum()
positive_count = csat_summary['sum'].sum()
negative_count = total_surveys - positive_count

fig = go.Figure()
fig.add_trace(go.Scatter(
  x=[0, 1, 2, 3],
  y=[1.6, 1.6, 1.6, 1.6],
  mode="text",
  text=[
    "<span style='font-size:20px'><b>Відсоток
CSAT</b></span>",
```

```
    "<span style='font-size:20px'><b>Загальна кількість опитувань</b></span>",
    "<span style='font-size:20px'><b>Кількість позитивних</b></span>",
    "<span style='font-size:20px'><b>Кількість негативних</b></span>"
  ],
  textposition="bottom center"))
fig.add_trace(go.Scatter(
  x=[0, 1, 2, 3],
  y=[1.1, 1.1, 1.1, 1.1],
  mode="text",
  text=[
    f"<span style='font-size:20px'><b>{csat_percentage:.2f}%</b></span>",
    f"<span style='font-size:20px'><b>{total_surveys}</b></span>",
    f"<span style='font-size:20px'><b>{positive_count}</b></span>",
    f"<span style='font-size:20px'><b>{negative_count}</b></span>"
  ],
  textposition="bottom center"))

fig.add_hline(y=2.2, line_width=5, line_color=ACCENT_COLOR)
fig.add_hline(y=0.3, line_width=3, line_color=ACCENT_COLOR)
fig.update_yaxes(visible=False)
fig.update_xaxes(visible=False)
fig.update_layout(
  showlegend=False,
  title_x=0.4,
  title_y=0.9,
  xaxis_range=[-0.5, 3.6],
  yaxis_range=[-0.2, 2.2],
  plot_bgcolor=BG_COLOR,
  paper_bgcolor=BG_COLOR,
  font=dict(size=16, color=TEXT_COLOR),
  title_font=dict(size=20),
  margin=dict(t=90, l=70, b=0, r=70),
  barmode='group',
  title='Підсумок CSAT',
  xaxis_title='',
  yaxis_title='',
  height=250
)
fig.show()

# Кількість агентів, супервайзерів, менеджерів
agents = df['Agent_name'].nunique()
supervisors = df['Supervisor'].nunique()
managers = df['Manager'].nunique()

fig = go.Figure()
fig.add_trace(go.Scatter(
  x=[0, 1, 2],
  y=[1.1, 1.1, 1.1],
  mode="text",
  text=[
    f"<span style='font-size:20px'><b>{agents}</b></span>",
    f"<span style='font-size:20px'><b>{supervisors}</b></span>",
    f"<span style='font-size:20px'><b>{managers}</b></span>"
  ],
  textposition="bottom center"))
fig.add_trace(go.Scatter(
  x=[0, 1, 2],
  y=[1.6, 1.6, 1.6],
  mode="text",
  text=[
    "<span style='font-size:20px'><b>Агенти</b></span>",
    "<span style='font-size:20px'><b>Супервайзери</b></span>",
    "<span style='font-size:20px'><b>Менеджери</b></span>"
  ],
  textposition="bottom center"))

fig.add_hline(y=2.2, line_width=5, line_color=ACCENT_COLOR)
fig.add_hline(y=0.3, line_width=3, line_color=ACCENT_COLOR)
fig.update_yaxes(visible=False)
fig.update_xaxes(visible=False)
fig.update_layout(
  showlegend=False,
  title_x=0.4,
  title_y=0.9,
  xaxis_range=[-0.5, 3.6],
  yaxis_range=[-0.2, 2.2],
  plot_bgcolor=BG_COLOR,
  paper_bgcolor=BG_COLOR,
  font=dict(size=16, color=TEXT_COLOR),
  title_font=dict(size=20, color=TEXT_COLOR),
  margin=dict(t=90, l=70, b=0, r=70),
  barmode='group',
  title='',
  xaxis_title='',
  yaxis_title='',
  height=250
)
fig.show()

# --------------------------
# КРАЩІ МЕНЕДЖЕРИ (CSAT)
# --------------------------
csat_managers = df.groupby('Manager')['CSAT'].agg(['mean', 'count']).reset_index()
csat_managers.columns = ['Менеджер', 'CSAT', 'Кількість_опитувань']
csat_managers = csat_managers.sort_values(by='CSAT', ascending=False).head(3).reset_index(drop=True)
csat_managers['Ранг'] = csat_managers.index + 1
```

```
positions = [1, 0, 2]
fig, ax = plt.subplots(figsize=(12, 4.5),
facecolor=BG_COLOR)
bars = sns.barplot(
  x=positions,
  y='CSAT',
  data=csat_managers,
  palette=COLOR_PALETTE[:3],  # Використовуємо
перші 3 кольори
  ax=ax,
  width=1,
  facecolor=BG_COLOR
)
ax.set_xticks(positions)
ax.set_xticklabels([f"{rank} місце" for rank in
csat_managers['Ранг']], color=TEXT_COLOR)
ax.set_yticks([])
ax.set_ylabel('')
ax.set_ylim(0.7, 0.87)
ax.set_title('Кращі менеджери',
color=TEXT_COLOR, loc='left', pad=25,
fontweight='bold')
ax.set_xlabel('')
ax.set_facecolor(BG_COLOR)
sns.despine(ax=ax, left=True, bottom=True)

for bar, csat, surveys, rank in zip(
  bars.patches,
  csat_managers['CSAT'],
  csat_managers['Кількість_опитувань'],
  csat_managers['Ранг']
):
  csat_text = f"Відсоток CSAT: {csat*100:.2f}%"
  survey_text = f"Опитування: {surveys}"
  label_x = bar.get_x() + bar.get_width() / 2
  label_height = bar.get_height() + 0.03
  ax.text(
    label_x,
    bar.get_height() + 0.1,
    f"{rank} місце",
    ha='center',
    va='bottom',
    color=TEXT_COLOR,
    fontsize=18,
    fontweight='bold'
  )
  ax.text(
    label_x,
    label_height,
    csat_text + '\n' + survey_text,
    ha='center',
    va='bottom',
    color=BG_COLOR,
    bbox=dict(facecolor='#f8bbd0',
edgecolor='#f8bbd0', boxstyle='round,pad=0.5')
  )

plt.tight_layout()
plt.show()

# -------------------------
# КРАЩІ СУПЕРВАЙЗЕРИ (CSAT)
# -------------------------
csat_supervisors =
df.groupby('Supervisor')['CSAT'].agg(['mean',
'count']).reset_index()
csat_supervisors.columns = ['Супервайзер', 'CSAT',
'Кількість_опитувань']
csat_supervisors =
csat_supervisors.sort_values(by='CSAT',
ascending=False).head(3).reset_index(drop=True)
csat_supervisors['Ранг'] = csat_supervisors.index +
1

positions = [1, 0, 2]
fig, ax = plt.subplots(figsize=(12, 4.5),
facecolor=BG_COLOR)
bars = sns.barplot(
  x=positions,
  y='CSAT',
  data=csat_supervisors,
  palette=COLOR_PALETTE[:3],
  ax=ax,
  width=1,
  facecolor=BG_COLOR
)
ax.set_xticks(positions)
ax.set_xticklabels([f"{rank} місце" for rank in
csat_supervisors['Ранг']], color=TEXT_COLOR)
ax.set_yticks([])
ax.set_ylabel('')
ax.set_ylim(0.7, 0.88)
ax.set_title('Кращі супервайзери',
color=TEXT_COLOR, loc='left', pad=27,
fontweight='bold')
ax.set_xlabel('')
ax.set_facecolor(BG_COLOR)
sns.despine(ax=ax, left=True, bottom=True)

for bar, csat, surveys, rank in zip(
  bars.patches,
  csat_supervisors['CSAT'],
  csat_supervisors['Кількість_опитувань'],
  csat_supervisors['Ранг']
):
  csat_text = f"Відсоток CSAT: {csat*100:.2f}%"
  survey_text = f"Опитування: {surveys}"
  label_x = bar.get_x() + bar.get_width() / 2
  label_height = bar.get_height() + 0.03
  ax.text(
    label_x,
    bar.get_height() + 0.1,
    f"{rank} місце",
    ha='center',
    va='bottom',
    color=TEXT_COLOR,
    fontsize=18,
    fontweight='bold'
  )
```

```
  ax.text(
    label_x,
    label_height,
    csat_text + '\n' + survey_text,
    ha='center',
    va='bottom',
    color=BG_COLOR,
    bbox=dict(facecolor='#f8bbd0',
edgecolor='#f8bbd0', boxstyle='round,pad=0.5')
  )

plt.tight_layout()
plt.show()

# --------------------------
# АНАЛІЗ АГЕНТІВ
# --------------------------
agent_summary =
df.groupby('Agent_name')['CSAT'].agg(['mean',
'count']).reset_index()
agent_summary.columns = ['Агент', 'CSAT%',
'Кількість_опитувань']
agent_summary =
agent_summary.sort_values(by='CSAT%',
ascending=False)
top_n_agents = agent_summary.head(10)
bottom_n_agents = agent_summary.tail(10)

fig = go.Figure()
# Низькі 10 агентів (у лівий бік)
fig.add_trace(go.Bar(
  x=-bottom_n_agents['CSAT%'] * 100,
  y=bottom_n_agents['Агент'],
  orientation='h',
  marker=dict(color=ACCENT_COLOR_2)
))
# Топ 10 агентів (у правий бік)
fig.add_trace(go.Bar(
  x=top_n_agents['CSAT%'] * 100,
  y=top_n_agents['Агент'],
  orientation='h',
  marker=dict(color='#d81b60')
))

for index, row in bottom_n_agents.iterrows():
  fig.add_annotation(
    x=row['CSAT%'] * 100,
    y=row['Агент'],
    text=f"{row['Агент']} ({row['CSAT%'] *
100:.2f}%)",
    showarrow=False,
    font=dict(color=ACCENT_COLOR_2, size=10),
    xshift=-150,
    xanchor='left',
    yanchor='auto'
  )
for index, row in top_n_agents.iterrows():
  fig.add_annotation(
    x=row['CSAT%'] * 100,
    y=row['Агент'],
    text=f"{row['Агент']} ({row['CSAT%'] *
100:.2f}%)",
    showarrow=False,
    font=dict(color='#d81b60', size=10),
    xshift=120
  )

fig.update_layout(
  title='Топ та найнижчі 10 агентів за CSAT%',
  plot_bgcolor=BG_COLOR,
  height=600,
  paper_bgcolor=BG_COLOR,
  xaxis_title='',
  yaxis_title='',
  bargap=0.1,
  yaxis=dict(tickvals=[]),
  xaxis=dict(tickvals=[]),
  showlegend=False,
  title_font=dict(size=20, family='Lato, sans-serif'),
  font=dict(color=TEXT_COLOR)
)
fig.show()

# --------------------------
# ДИНАМІКА CSAT ЗА ДАТАМИ
# --------------------------
df['Survey_response_Date'] =
pd.to_datetime(df['Survey_response_Date'])
df['Week'] =
df['Survey_response_Date'].dt.isocalendar().week
daily_csat =
df.groupby('Survey_response_Date')['CSAT'].mean(
) * 100
weekly_csat = df.groupby('Week')['CSAT'].mean() *
100
daily_survey_counts =
df.groupby('Survey_response_Date').size()
weekly_survey_counts = df.groupby('Week').size()

fig = make_subplots(rows=1, cols=2,
specs=[[{"secondary_y": True}, {"secondary_y":
True}]])
# Щоденний CSAT
fig.add_trace(go.Scatter(
  x=daily_csat.index,
  y=daily_csat.values,
  mode='lines+markers',
  name='CSAT',
  line=dict(color='#d81b60', width=2)
), row=1, col=1, secondary_y=True)
# Кількість опитувань за днями
fig.add_trace(go.Bar(
  x=daily_survey_counts.index,
  y=daily_survey_counts.values,
  name='Опитування',
  marker=dict(color=ACCENT_COLOR)
), row=1, col=1, secondary_y=False)
# Щотижневий CSAT
fig.add_trace(go.Scatter(
  x=weekly_csat.index,
```

```
    y=weekly_csat.values,
    mode='lines+markers',
    showlegend=False,
    line=dict(color='#d81b60', width=2)
  ), row=1, col=2, secondary_y=True)
  # Кількість опитувань за тижнями
  fig.add_trace(go.Bar(
    x=weekly_survey_counts.index,
    y=weekly_survey_counts.values,
    showlegend=False,
    marker=dict(color=ACCENT_COLOR)
  ), row=1, col=2, secondary_y=False)

  fig.update_layout(
    height=400,
    plot_bgcolor=BG_COLOR,
    title='Щоденна та щотижнева динаміка CSAT',
    title_font=dict(size=20, family="Lato, sans-serif"),
    font=dict(color=TEXT_COLOR),
    paper_bgcolor=BG_COLOR,
    showlegend=True,
    legend=dict(x=0, y=1.1, orientation='h')
  )
  fig.update_yaxes(showgrid=False)
  fig.show()

  # --------------------------
  # ФУНКЦІЯ ДЛЯ РОЗПОДІЛУ ОЗНАК
  # --------------------------
  def plot_distribution(feature):
    feature_distribution =
  df[feature].value_counts().sort_values(ascending=
  False)
    bar_data = go.Bar(
      x=feature_distribution.values,
      y=feature_distribution.index,
      orientation='h',
      marker=dict(color=COLOR_PALETTE[:5]),
      showlegend=False,
      name=''
    )
    labels = feature_distribution.index.tolist()
    values = feature_distribution.values.tolist()
    donut_data = go.Pie(
      labels=labels,
      values=values,
      marker=dict(
        colors=COLOR_PALETTE[:5],
        line=dict(color='#ffffff', width=0.5)
      ),
      hole=0.55,
      showlegend=False,
      name='',
      textinfo='label+percent',
      hoverinfo='label+percent',
      textposition='inside'
    )

    fig = make_subplots(rows=1, cols=2,
  specs=[[{"type": "bar"}, {"type": "pie"}]])
    fig.add_trace(bar_data, row=1, col=1)
    fig.add_trace(donut_data, row=1, col=2)

    fig.update_layout(
      title=f'Розподіл {feature}',
      plot_bgcolor=BG_COLOR,
      paper_bgcolor=BG_COLOR,
      title_font=dict(size=20, family="Lato, sans-
  serif"),
      font=dict(color=TEXT_COLOR)
    )
    fig.show()

  plot_distribution("CSAT_Score")
  plot_distribution("CSAT")

  # --------------------------
  # РОЗШИРЕНА ФУНКЦІЯ РОЗПОДІЛУ
  # --------------------------
  def plot_distribution_extended(feature):
    feature_distribution =
  df[feature].value_counts().sort_values(ascending=
  False)

    bar_data = go.Bar(
      x=feature_distribution.values,
      y=feature_distribution.index,
      orientation='h',
      name='Кількість опитувань',
      marker=dict(color=COLOR_PALETTE)
    )
    donut_data = go.Pie(
      labels=feature_distribution.index.tolist(),
      values=feature_distribution.values.tolist(),
      marker=dict(
        colors=COLOR_PALETTE,
        line=dict(color='#ffffff', width=0.5)
      ),
      hole=0.55,
      name='Кількість опитувань',
      textfont=dict(color=TEXT_COLOR)
    )
    feature_distribution_0 = df[df['CSAT'] ==
  0][feature].value_counts().sort_values(ascending=
  False)
    feature_distribution_1 = df[df['CSAT'] ==
  1][feature].value_counts().sort_values(ascending=F
  alse)

    bar_data_0 = go.Bar(
      y=feature_distribution_0.index,
      x=feature_distribution_0.values,
      orientation='h',
      marker=dict(color='#f8bbd0'),
      name='Негативні'
    )
    bar_data_1 = go.Bar(
      y=feature_distribution_1.index,
      x=feature_distribution_1.values,
      orientation='h',
      marker=dict(color='#f48fb1'),
```

```
    name='Позитивні'
  )
  csat_percentages =
df.groupby(feature)['CSAT'].mean() * 100
  csat_percentages =
csat_percentages.sort_values(ascending=False)
  stacked_bar_data = go.Bar(
    x=csat_percentages.index,
    y=csat_percentages.values,
    marker=dict(color=COLOR_PALETTE),
    name='Відсоток CSAT'
  )

  fig = make_subplots(
    rows=2, cols=2,
    specs=[[{"type": "bar"}, {"type": "pie"}],
        [{"type": "bar"}, {"type": "bar"}]],
    subplot_titles=("Загальна кількість опитувань",
"Частка оцінок",
          "Позитивні та негативні оцінки", "Відсоток
CSAT")
  )
  fig.add_trace(bar_data, row=1, col=1)
  fig.add_trace(donut_data, row=1, col=2)
  fig.add_trace(bar_data_0, row=2, col=1)
  fig.add_trace(bar_data_1, row=2, col=1)
  fig.add_trace(stacked_bar_data, row=2, col=2)

  fig.update_layout(
    height=700,
    plot_bgcolor=BG_COLOR,
    title=f'Розподіл CSAT за {feature}',
    paper_bgcolor=BG_COLOR,
    title_font=dict(size=20, family="Lato, sans-
serif"),
    font=dict(color=TEXT_COLOR),
    showlegend=False
  )
  fig.show()

plot_distribution_extended("Manager")
plot_distribution_extended("channel_name")
plot_distribution_extended("Tenure_Bucket")
plot_distribution_extended("category")
plot_distribution_extended("Agent_Shift")

# --------------------------
# Treemap та Sunburst
# --------------------------
csat_percentage = df.groupby(['category', 'Sub-
category'])['CSAT'].mean() * 100
csat_percentage_df =
csat_percentage.reset_index()
fig = px.treemap(
  csat_percentage_df,
  path=['category', 'Sub-category'],
  values='CSAT',
  color='CSAT',
  color_continuous_scale='pinkyl'
)
fig.update_layout(
  height=600,
  title='Категорії та підкатегорії',
  plot_bgcolor=BG_COLOR,
  paper_bgcolor=BG_COLOR,
  title_font_size=18,
  font=dict(size=10, color=TEXT_COLOR)
)
fig.show()

csat_percentage = df.groupby(['Manager',
'Supervisor'])['CSAT'].mean() * 100
csat_percentage_df =
csat_percentage.reset_index()
fig = px.treemap(
  csat_percentage_df,
  path=['Manager', 'Supervisor'],
  values='CSAT',
  color='CSAT',
  color_continuous_scale='pinkyl'
)
fig.update_layout(
  height=600,
  title='Менеджери та супервайзери',
  plot_bgcolor=BG_COLOR,
  paper_bgcolor=BG_COLOR,
  title_font_size=18,
  font=dict(size=10, color=TEXT_COLOR)
)
fig.show()

csat_percentage = df.groupby(['Manager',
'category'])['CSAT'].mean() * 100
csat_percentage_df =
csat_percentage.reset_index()
fig = px.treemap(
  csat_percentage_df,
  path=['Manager', 'category'],
  values='CSAT',
  color='CSAT',
  color_continuous_scale='pinkyl'
)
fig.update_layout(
  height=600,
  title='Менеджери та категорії',
  plot_bgcolor=BG_COLOR,
  paper_bgcolor=BG_COLOR,
  title_font_size=18,
  font=dict(size=10, color=TEXT_COLOR)
)
fig.show()

csat_summary = df.groupby(['Manager',
'Supervisor', 'Agent_name'])['CSAT'].agg(['mean',
'count']).reset_index()
csat_summary['CSAT%'] = csat_summary['mean']
* 100
min_value = csat_summary['CSAT%'].min()
max_value = csat_summary['CSAT%'].max()
csat_summary['CSAT'] = (csat_summary['CSAT%']
- min_value) / (max_value - min_value) * 100
```

```
fig = px.sunburst(
  csat_summary,
  path=['Manager', 'Supervisor', 'Agent_name'],
  values='CSAT%',
  color='CSAT',
  hover_data=['CSAT%'],
  color_continuous_scale='pinkyl'
)
fig.update_layout(
  height=900,
  title='CSAT за менеджерами, супервайзерами та агентами',
  plot_bgcolor=BG_COLOR,
  paper_bgcolor=BG_COLOR,
  font=dict(color=TEXT_COLOR)
)
fig.show()

# --------------------------
# WORDCLOUD
# --------------------------
negative_df = df[df['CSAT'] == 0]
positive_df = df[df['CSAT'] == 1]

def generate_wordcloud(data, title):
  data['Customer_Remarks'] =
data['Customer_Remarks'].astype(str)
  words = ' '.join(data['Customer_Remarks'])
  stopwords = set(STOPWORDS)
  stopwords.update(['nan', 'Shopzilla'])
  wordcloud = WordCloud(
    stopwords=stopwords,
    max_words=1500,
    max_font_size=350,
    random_state=42,
    width=2000,
    height=500,
    colormap="pink"
  ).generate(words)
  plt.figure(figsize=(16, 8), facecolor=BG_COLOR)
  plt.imshow(wordcloud, interpolation='bilinear')
  plt.axis("off")
  plt.title(title, color=TEXT_COLOR,
fontweight='bold')
  plt.show()

generate_wordcloud(negative_df, 'Хмарка слів для негативних відгуків')
generate_wordcloud(positive_df, 'Хмарка слів для позитивних відгуків')

# --------------------------
# ПАРЕТО-АНАЛІЗ
# --------------------------
dissatisfied_df = df[df['CSAT'] == 0]
category_counts =
dissatisfied_df['category'].value_counts().reset_ind
ex()
category_counts.columns = ['категорія', 'кількість']
category_counts =
category_counts.sort_values(by='кількість',
ascending=False)
category_counts['накопичуваний_відсоток'] =
(category_counts['кількість'].cumsum() /
category_counts['кількість'].sum()) * 100

subcategory_counts = dissatisfied_df['Sub-
category'].value_counts().reset_index()
subcategory_counts.columns = ['підкатегорія',
'кількість']
subcategory_counts =
subcategory_counts.sort_values(by='кількість',
ascending=False)
subcategory_counts['накопичуваний_відсоток'] =
(subcategory_counts['кількість'].cumsum() /
subcategory_counts['кількість'].sum()) * 100

fig1 = go.Figure()
fig1.add_bar(
  x=category_counts['категорія'],
  y=category_counts['кількість'],
  name='Кількість незадоволених',
  marker_color=ACCENT_COLOR
)
fig1.add_scatter(
  x=category_counts['категорія'],
  y=category_counts['накопичуваний_відсоток'],
  mode='lines+markers',
  name='Накопичуваний відсоток',
  yaxis='y2',
  marker=dict(color='#d81b60')
)
threshold_index =
(category_counts['накопичуваний_відсоток'] >=
80).idxmax()
threshold_category =
category_counts.loc[threshold_index, 'категорія']
fig1.add_shape(
  type="line",
  x0=threshold_category,
  y0=0,
  x1=threshold_category,
  y1=category_counts['кількість'].max(),
  line=dict(color="red", width=0.5, dash="dot")
)
fig1.update_layout(
  title='Парето-аналітика незадоволення за категоріями',
  xaxis_title='',
  yaxis_title='Кількість незадоволених',
  yaxis2=dict(title='', overlaying='y', side='right',
color=TEXT_COLOR),
  legend=dict(x=0, y=1.1, orientation='h'),
  plot_bgcolor=BG_COLOR,
  paper_bgcolor=BG_COLOR,
  font=dict(color=TEXT_COLOR)
)
fig1.update_yaxes(showgrid=False)
fig1.show()
```

```
fig2 = go.Figure()
fig2.add_bar(
  x=subcategory_counts['підкатегорія'],
  y=subcategory_counts['кількість'],
  name='Кількість незадоволених',
  marker_color=ACCENT_COLOR
)
fig2.add_scatter(
  x=subcategory_counts['підкатегорія'],
  y=subcategory_counts['накопичуваний_відсоток'],
  mode='lines+markers',
  name='Накопичуваний відсоток',
  yaxis='y2',
  marker=dict(color='#d81b60')
)
threshold_index =
(subcategory_counts['накопичуваний_відсоток'] >=
80).idxmax()
threshold_subcategory =
subcategory_counts.loc[threshold_index,
'підкатегорія']
fig2.add_shape(
  type="line",
  x0=threshold_subcategory,
  y0=0,
  x1=threshold_subcategory,
  y1=subcategory_counts['кількість'].max(),
  line=dict(color="red", width=0.5, dash="dot")
)
fig2.update_layout(
  title='Парето-аналітика незадоволення за
підкатегоріями',
  xaxis_title='',
  height=700,
  yaxis_title='Кількість незадоволених',
  yaxis2=dict(title='', overlaying='y', side='right',
color=TEXT_COLOR),
  legend=dict(x=0, y=1.1, orientation='h'),
  plot_bgcolor=BG_COLOR,
  paper_bgcolor=BG_COLOR,
  font=dict(color=TEXT_COLOR)
)
fig2.update_yaxes(showgrid=False)
fig2.show()

# --------------------------
# ЧАС ВІДПОВІДІ ТА ЙОГО БІНІНГ
# --------------------------
df['Issue_reported_at'] =
pd.to_datetime(df['Issue_reported_at'],
format='%d/%m/%Y %H:%M')
df['issue_responded'] =
pd.to_datetime(df['issue_responded'],
format='%d/%m/%Y %H:%M')
df['ResponseTime'] = (df['issue_responded'] -
df['Issue_reported_at']).dt.total_seconds() / 3600

bins = [0, 4, 6, 12, 24, 36, float('inf')]
labels = ['0-4 год', '4-6 год', '6-12 год', '12-24 год','24-36
год', '>36 год']
df['Response_Time_Bins'] =
pd.cut(df['ResponseTime'], bins=bins,
labels=labels, right=False)
plot_distribution("Response_Time_Bins")

# --------------------------
# АКТИВНІСТЬ ЗА ГОДИНАМИ
# --------------------------
df['hour_reported'] =
df['Issue_reported_at'].dt.hour
hourly_counts =
df['hour_reported'].value_counts().sort_index()
fig = go.Figure()
fig.add_trace(go.Bar(
  x=hourly_counts.index,
  y=hourly_counts.values,
  name='Загальна кількість звернень',
  yaxis='y',
  marker_color=ACCENT_COLOR
))
hourly_avg_response_time =
df.groupby('hour_reported')['ResponseTime'].mea
n()
fig.add_trace(go.Scatter(
  x=hourly_avg_response_time.index,
  y=hourly_avg_response_time.values,
  mode='lines',
  name='Середній час відповіді',
  yaxis='y2'
))
hourly_csat_percentage =
df.groupby('hour_reported')['CSAT'].mean() * 100
fig.add_trace(go.Scatter(
  x=hourly_csat_percentage.index,
  y=hourly_csat_percentage.values,
  mode='lines',
  name='Відсоток CSAT',
  yaxis='y3'
))
fig.update_layout(
  title='Годинна статистика: загальна кількість
звернень, середній час відповіді та CSAT %',
  xaxis_title='Година реєстрації звернення',
  yaxis_title='Загальна кількість звернень',
  yaxis2=dict(title='Середній час відповіді',
overlaying='y', side='right', color=TEXT_COLOR),
  yaxis3=dict(title='', overlaying='y', side='right',
color=ACCENT_COLOR_2, showticklabels=False),
  legend=dict(x=0, y=1.1, orientation='h'),
  plot_bgcolor=BG_COLOR,
  paper_bgcolor=BG_COLOR,
  font=dict(color=TEXT_COLOR)
)
fig.update_yaxes(showgrid=False)
fig.show()

# --------------------------
# ТЕПЛОВА КАРТА
```

```
# -------------------------
pivot_table = df.pivot_table(index='category',
columns='hour_reported', values='CSAT',
aggfunc='mean')
fig = go.Figure(data=go.Heatmap(
  z=pivot_table.values,
  x=pivot_table.columns,
  y=pivot_table.index,
  colorscale='pinkyl',
  hoverongaps=True,
  hoverinfo='z+x+y',
  showscale=False
))
annotations = []
for i, row in enumerate(pivot_table.values):
  for j, val in enumerate(row):
    if not np.isnan(val):
      annotations.append(dict(
        text=f'{val:.0%}',
()
        x=pivot_table.columns[j],
        y=pivot_table.index[i],
        font=dict(color='white' if val > 0.7 else
'black'),
        showarrow=False
      ))
fig.update_layout(
  title='CSAT за категоріями по годинах',
  xaxis=dict(title='Година реєстрації звернення'),
  yaxis=dict(title=''),
  annotations=annotations,
  plot_bgcolor=BG_COLOR,
  paper_bgcolor=BG_COLOR,
  font=dict(color=TEXT_COLOR)
)
fig.update_yaxes(showgrid=False)
fig.show()
```

Додаток Б

Апробація отриманих результатів

Західноукраїнський національний університет

Факультет комп'ютерних інформаційних технологій

Кафедра інформаційно-обчислювальних систем і управління

ЗБІРНИК ТЕЗ ДОПОВІДЕЙ

Студентської науково-практичної конференції

ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ПРИКЛАДНИХ ДОСЛІДЖЕННЯХ

(IITAR-2025)

27-29 травня 2025 року

Тернопіль

2025

Кацидим Віта
студентка групи КН-42
vitakatsydym04@gmail.com
Ліп'яніна-Гончаренко Христина
д.т.н., доцент
kh.lipianina@wunu.edu.ua
Західноукраїнський національний університет
Тернопіль, Україна
Ралік Ігор
аспірант
ralik10gk@gmail.com
Тернопільський національний технічний університет ім. Івана Пулюя
Тернопіль, Україна

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ЗАДОВОЛЕНОСТІ КЛІЄНТІВ ОБСЛУГОВУВАННЯМ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ

У сучасних умовах ведення бізнесу, що характеризуються високою конкуренцією, задоволеність клієнтів стала ключовим індикатором успіху підприємств. Високий рівень задоволеності клієнтів сприяє повторним продажам, збільшує лояльність і сприяє залученню нових клієнтів за рахунок рекомендацій. Водночас традиційні методи оцінки задоволеності клієнтів, такі як анкетування або телефонні опитування, мають суттєві недоліки: вони часто є суб'єктивними, витратними у реалізації та не здатні враховувати всі аспекти взаємодії клієнта з компанією.

Одним з ефективних шляхів вирішення зазначених проблем є застосування методів штучного інтелекту для автоматизованого аналізу даних. Інтелектуальні алгоритми здатні знаходити приховані закономірності, визначати причини незадоволеності, прогнозувати поведінку клієнтів та ефективно адаптувати стратегії обслуговування під конкретні потреби аудиторії. Важливість цього підходу посилюється у цифрову епоху, коли компанії отримують величезні обсяги інформації з різних джерел.

Метою роботи було розробити програмний модуль інтелектуального аналізу задоволеності клієнтів обслуговуванням, що базується на методах машинного навчання і дозволяє значно підвищити точність і ефективність прийняття управлінських рішень. Основними завданнями були аналіз існуючих рішень, проектування архітектури системи, розробка моделей даних та реалізація аналітичних алгоритмів із наступною візуалізацією результатів.

Під час аналізу предметної області було досліджено класичні та сучасні метрики оцінки задоволеності: Customer Satisfaction Score (CSAT), Net Promoter Score (NPS) та Customer Effort Score (CES). Зокрема, CSAT відображає емоційне задоволення клієнта після конкретної взаємодії, NPS характеризує готовність

рекомендувати бренд, а CES оцінює кількість зусиль, докладених клієнтом для вирішення проблеми чи здійснення покупки.

Розроблена архітектура системи включає багаторівневу структуру: рівень збору даних, їх попередню обробку, зберігання, аналітичні алгоритми та рівень візуалізації результатів. В якості технологічної бази було обрано мову програмування Python та відповідні аналітичні бібліотеки (pandas, NumPy, matplotlib, scikit-learn). Для зручності роботи з даними використовувалась SQL-база даних PostgreSQL.

Важливим етапом реалізації стала розробка моделі даних, що дозволяє ефективно зберігати інформацію про взаємодії з клієнтами, категорії звернень, характеристики агентів і показники задоволеності. Запропонована модель даних була реалізована у вигляді схеми типу "зірка" (Star Schema), яка забезпечує швидку обробку запитів і ефективний аналітичний доступ до інформації.

В роботі було застосовано алгоритми попередньої обробки даних, що включали очищення від пропусків, аномальних значень та дублювань, що дозволило суттєво підвищити якість аналізу. У результаті очищення обсяг пропущених даних скоротився на 78%, що позитивно вплинуло на точність подальших аналітичних процедур.

Для класифікації задоволеності клієнтів були використані алгоритми машинного навчання, зокрема Random Forest, Gradient Boosting та логістична регресія. Найвищі показники точності (92%) продемонструвала модель Random Forest, що була обрана в якості базового рішення. Отримані результати підтвердили ефективність застосування ансамблевих методів для вирішення цієї задачі.

Розроблений програмний модуль включає зручний інтерактивний дашборд, що дозволяє візуалізувати ключові метрики в реальному часі. На рисунку 1 показано архітектуру реалізованого модуля, який містить компоненти збору даних, обробки, аналізу та візуалізації у формі інтерактивних графіків.

Окрім точних числових результатів, у системі передбачено модуль аналізу текстових відгуків клієнтів з використанням NLP-алгоритмів. Це дозволяє автоматично класифікувати коментарі за емоційною тональністю і швидко виявляти проблемні аспекти обслуговування.

Запропонований модуль пройшов експериментальну апробацію, під час якої була підтверджена його ефективність для вирішення практичних завдань аналізу задоволеності клієнтів. Реалізація запропонованих рішень дозволила скоротити час аналізу клієнтських відгуків і оперативно реагувати на зміни рівня задоволеності.

Практичне застосування результатів роботи передбачає інтеграцію модуля у CRM-системи компаній, що дозволяє автоматично проводити моніторинг, аналіз та прогнозування поведінки клієнтів, а також оперативно реагувати на зниження показників задоволеності.

Подальші перспективи дослідження включають удосконалення алгоритмів машинного навчання для підвищення точності прогнозування, а також інтеграцію додаткових каналів отримання даних, таких як соціальні мережі та

месенджери, що дозволить отримати ще більш повну картину задоволеності клієнтів.

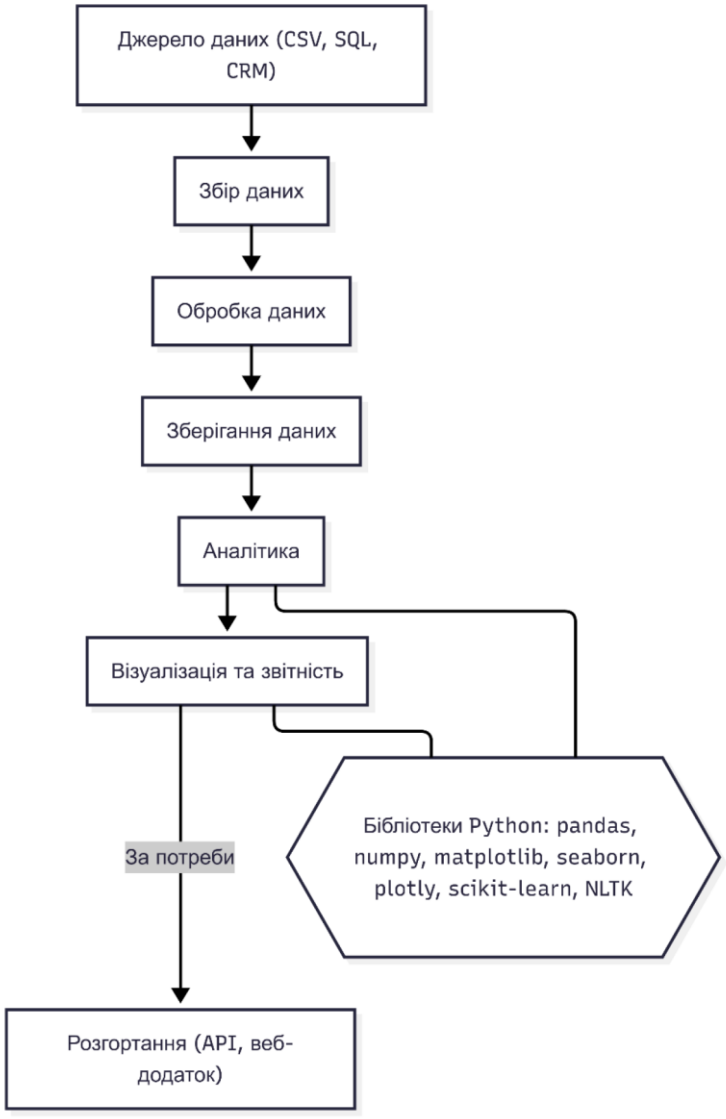

Рисунок 1 – Архітектура модуля аналізу задоволеності клієнтів

Отже, створений програмний модуль представляє собою ефективний інструмент, здатний суттєво поліпшити управління задоволеністю клієнтів і забезпечити прийняття швидких та обґрунтованих управлінських рішень.

Список використаних джерел

1. Liapis, V., et al. (2025). Distributional Emotion Embeddings for Text Classification. *Journal of Computational Linguistics*, 51(3), 678–695.
2. Selim, S., & Assiri, Y. (2025). AMCDD-AHODL: A Hybrid Model for Arabic Emotional Text to Speech. *Computational Intelligence*, 39(4), 1501–1515.

3. Putri, N., et al. (2025). Emotion Analysis of Political Tweets. *Social Media Analytics*, 10(1), 23–32.
4. Nurlanuly, T. (2025). Sentiment Analysis in Social Media Using ML. *Journal of Social Informatics*, 8(2), 125–134.
5. Manoharan, S., & Kanimozhi, M. (2025). MentalBERT for Psychological Chatbots. *IEEE Transactions on Affective Computing*, 16(1), 47–58.
6. Fedushko, S., et al. (2025). Emotional Analysis of Euphemisms in Military Discourse. *Eastern-European Journal of Enterprise Technologies*, 5(118), 18–27.
7. Abdykadyrova, G., et al. (2025). Linguocultural Analysis of Emotionality. *Cultural Linguistics Journal*, 12(2), 139–152.

Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

В І Д Г У К

наукового керівника Ліпяніна-Гончаренко Христина Володимирівна,

на дипломний проект студента групи КН-КН-42 Кацидим Віта Олександрівна

на тему: Інтелектуальний аналіз задоволеності клієнтів обслуговуванням / Intelligent analysis of customer satisfaction with service

Актуальність теми: Актуальність теми дослідження визначається значним зростанням ролі задоволеності клієнтів як ключового показника успішності сучасного бізнесу. Використання традиційних методів оцінки має значні обмеження у вигляді суб'єктивності та тривалого часу обробки результатів. Застосування сучасних технологій інтелектуального аналізу дозволяє автоматизувати ці процеси та забезпечити швидку реакцію компанії на потреби клієнтів, що є критично важливим для підтримки конкурентних переваг.

Самостійні розробки і пропозиції автора: Автор самостійно розробив архітектуру програмного модуля та алгоритми інтелектуального аналізу даних. Особливо цінним є створення адаптивних дашбордів для зручної та оперативної візуалізації ключових показників задоволеності клієнтів, що свідчить про високий рівень технічної підготовки та креативності автора.

Практичне значення кваліфікаційної роботи: Розроблений модуль може бути інтегрований у бізнес-процеси компаній для моніторингу та покращення якості обслуговування клієнтів. Це дозволяє значно оптимізувати управлінські рішення та оперативно реагувати на зміни клієнтських настроїв.

Недоліки: Робота не повною мірою враховує аспекти масштабування системи для великих наборів даних та обмежено розглядає питання інтеграції модуля із зовнішніми корпоративними системами. Також недостатньо уваги приділено економічній оцінці ефективності впровадження.

Загальний висновок: вважаю, що дипломна робота заслуговує на позитивну оцінку, а Кацидим В.О. – присвоєння освітнього ступеня «бакалавр» з спеціальності 122 «Комп'ютерні науки».

Науковий керівник: д.т.н., доцент, Ліпяніна-Гончаренко Христина Володимирівна

"10" червня 2025 р.

_____

РЕЦЕНЗІЯ

на дипломний проект студента групи КН-КН-42 Кацидим Віта Олександрівна

на тему: Інтелектуальний аналіз задоволеності клієнтів обслуговуванням / Intelligent analysis of customer satisfaction with service.

Виконаний на матеріалах: Дослідження проведено на матеріалах реальних відгуків та взаємодії клієнтів із комерційними компаніями.

Актуальність теми: Актуальність даної роботи зумовлена нагальною потребою бізнесу в сучасних і ефективних методах оцінювання задоволеності клієнтів. Використання інтелектуального аналізу дозволяє оперативно та точно ідентифікувати проблемні зони в обслуговуванні, що особливо важливо в умовах конкуренції та стрімких змін споживацьких вимог. Впровадження таких систем забезпечує покращення якості сервісу та підвищення лояльності клієнтів, що позитивно впливає на стабільність бізнесу. Автором запропоновано актуальний технологічний підхід, який відповідає сучасним вимогам ринку та може бути широко застосований у бізнес-середовищі.

Самостійні розробки і пропозиції автора: Автор самостійно розробив структуру даних та алгоритми їх інтелектуальної обробки, а також створив комплексний аналітичний інтерфейс. Реалізовані рішення демонструють високу кваліфікацію і глибоке розуміння предметної області.

Практичне значення кваліфікаційної роботи: Практична цінність роботи полягає у можливості швидко аналізувати великі об'єми даних і своєчасно реагувати на зміни у настроях клієнтів. Це робить розробку цінним інструментом для управління якістю обслуговування.

Недоліки: У роботі варто було б подати порівняльний аналіз з іншими моделями. Крім того, бажано провести оцінку стійкості моделі до змін у словнику та структурі запитань.

Загальний висновок: дипломна робота заслуговує на позитивну оцінку, а Кацидим В.О. – присвоєння освітнього ступеня «бакалавр» з спеціальності 122 «Комп’ютерні науки».

Рецензент:

Керівник групи інтелектуальних розподілених систем, НДІ ІКС ЗУНУ, к.т.н., доцент Павло Биковий

М.П. _____ “ 11 ” червня 2025 р.

(підпис)