КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп'ютерні науки,
освітня програма «Інформаційна аналітика та впливи»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«РОЗРОБКА ІНТЕГРОВАНОЇ АНАЛІТИЧНОЇ ПЛАТФОРМИ ДЛЯ АНАЛІЗУ ДАНИХ ТА АВТОМАТИЗАЦІЇ ЗВІТНОСТІ У ПРОЦЕСІ КООРДИНАЦІЇ ВОЛОНТЕРСЬКОГО РУХУ»

Студента 2-го курсу групи ІАВ-21

Богдана ОРИЩАКА
(ім'я, прізвище)

_____ _____ _____
(підпис студента) (дата) (підпис)

Науковий керівник:

д.т.н., професор
(науковий ступінь, вчене звання)

Юлія ХЛЕВНА
(ім'я, прізвище)

Попередній захист:

_____

(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач кафедри
технологій управління

_____ _____ _____
(підпис) (прізвище, ініціали) (дата)

Київ - 2025

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій

Кафедра технологій управління
Освітньо-кваліфікаційний рівень Магістр
Спеціальність 122 – Комп'ютерні науки
Освітня програма Інформаційна аналітика та впливи

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Морозов В.В.

_____
«___» _____2025 р.

З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студент Богдан ОРИЩАК

Група ІАВ-21

1. Тема кваліфікаційної роботи

«Розробка інтегрованої аналітичної платформи для аналізу даних та автоматизації звітності у процесі координації волонтерського руху»

Затверджена наказом по університету від «__» _____2025 р. №_____.

2. Строк подання студентом готової роботи – «__» ___2025 р.

3. Цільова установка та вихідні дані до роботи: дослідження зосереджене на розробці інтегрованої аналітичної платформи для аналізу даних і автоматизації звітності в межах координації волонтерського руху. Основна мета полягає у створенні ефективного інструменту, який дозволяє швидко виявляти ключові закономірності у даних, оцінювати ефективність кампаній, автоматизувати процес підготовки звітності та підвищити рівень прозорості у волонтерських ініціативах. У роботі розглядаються та реалізуються різноманітні методи аналізу даних, кластеризація (алгоритм k – середніх та ієрархічна кластеризація),

кореляційний аналіз, аналіз аномалій (для виявлення нетипових патернів у даних), розрахунок коефіцієнта корисного завантаження (Utilization Rate). Важливою складовою є також інтеграція результатів у зручний веб-інтерфейс, що дозволяє в реальному часі взаємодіяти з результатами обробки та формувати автоматизовані PDF – звіти. Оскільки у відкритому доступі відсутні повноцінні структуровані дані щодо волонтерської діяльності, для цілей дослідження використано штучно згенерований датасет, що відображає типові показники: обсяги зібраних коштів, кількість донатів, географію кампаній, часові рамки, класифікацію цілей, динаміку активностей тощо. Цей набір даних став основою для перевірки та валідації аналітичних методів, які реалізовані у межах платформи.

4. Зміст роботи

У межах кваліфікаційної роботи магістра було здійснено комплексний аналіз проблематики обробки, аналізу та інтерпретації даних, що стосуються координації волонтерських ініціатив. Враховуючи складність, динамічність та високу змінність середовища благодійних проєктів, дослідження охоплювало вивчення сучасних підходів до аналітики даних, методів кластеризації, виявлення аномалій кореляційного аналізу, а також механізмів візуалізації результатів у форматі, придатному для практичного застосування координаторам волонтерських програм. Розроблена платформа була протестована на спеціально згенерованому датасеті, який відтворює характерні особливості реального інформаційного середовища волонтерської діяльності. Отримані результати підтвердили, що створення інтегрованої аналітичної платформи для аналізу даних волонтерських кампаній є перспективним напрямом розвитку, що сприятиме цифровій трансформації волонтерського руху.

5. Перелік графічного матеріалу (слайдів)

Дана кваліфікаційна робота магістра налічує в собі: 63 рисунків, які позначають діаграми різних процесів, представлення вихідних даних у вигляді графіків, структури моделей, графіки результатів. Також робота містить 7 формул та 5

таблиць, які представляють собою опис характеристик вихідних даних, використані бібліотеки для реалізації програмного продукту, оцінки ефективності.

6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	%	Виконання роботи	
			За планом	Фактично
1.	Вибір теми дипломної роботи	3	15.04.25	15.04.25
2.	Протокол кафедри ТУ про затвердження тем дипломних робіт та призначення наукових керівників	2	22.04.25	22.04.25
3.	Формування переліку нормативних матеріалів, літератури з проблематики дипломної роботи	10	26.04.25	26.04.25
4.	Складання розгорнутого плану кваліфікаційної роботи	5	27.04.25	27.04.25
5.	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін.	5	28.04.25	28.04.25
6.	Підготовка розділу 1 «Формулювання задачі дослідження та аналіз існуючих методів побудови аналітичних платформ»	10	29.04.25	29.04.25
7.	Підготовка розділу 2 «Методи та методики побудови інтегрованої аналітичної платформи»	15	30.04.25	30.04.25
8.	Підготовка розділу 3 «Моделювання процесів аналізу даних у волонтерських капманіях»	15	01.05.25	01.05.25
9.	Підготовка розділу 4 «Технологія практичного застосування розробленої аналітичної платформи»	15	02.05.25	02.05.25
10.	Оформлення кваліфікаційної роботи. Підготовка висновків і пропозицій.	11	05.05.25	05.05.25
11.	Передача кваліфікаційної роботи науковому керівникові	2	07.05.25	07.05.25
12.	Передача кваліфікаційної роботи рецензенту для рецензування	2	09.05.25	09.05.25
13.	Попередній захист кваліфікаційної роботи	5	13.05.25	13.05.25

Дата видачі завдання «___» _____2025 р.

Керівник роботи професор кафедри технологій управління,
Юлія ХЛЕВНА
(посада, ім’я, прізвище)

_____
(підпис)

Завдання прийняв
до виконання
студент групи ІАВ-21
Богдан ОРИЩАК
(ім’я, прізвище)

_____
(підпис)

ЗМІСТ

АНОТАЦІЯ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра технологій управління
Освітньо-кваліфікаційний рівень Магістр
Спеціальність 122 – Комп’ютерні науки
Освітня програма Інформаційна аналітика та впливи

Кваліфікаційна робота магістра Богдана ОРИЩАКА.

Тема роботи – «Розробка інтегрованої аналітичної платформи для аналізу даних та автоматизації звітності у процесі координації волонтерського руху».

Мета роботи. Підвищення ефективності аналізу даних, що супроводжують діяльність волонтерських кампаній, шляхом створення інтегрованої аналітичної платформи, яка забезпечує автоматизовану обробку, візуалізацію та звітність, дозволяючи приймати обґрунтовані управлінські рішення на основі структурованих даних.

Завдання роботи. Кваліфікаційна робота зосереджена на побудові технологічного рішення для аналізу волонтерських даних у структурованому вигляді. У межах дослідження здійснено аналіз сучасних підходів до обробки інформації, обґрунтовано вибір методів кластеризації, кореляційного аналізу, виявлення аномалій, розрахунку коефіцієнта корисного завантаження, а також ієрархічної кластеризації. Окрема увага приділяється створенню зручного веб-інтерфейсу, який дозволяє користувачеві обирати змінні для аналізу, переглядати результати та завантажувати автоматично згенеровані PDF-звіти. Робота базується на штучно згенерованому наборі даних, що імітує активність волонтерських кампаній, з огляду на обмежену доступність публічної інформації про такі процеси.

Об’єкт дослідження. Інформаційно-аналітичні процеси координації, моніторингу та оцінювання ефективності волонтерської діяльності в умовах

неприбуткових організацій**,** зокрема в аспекті збору, обробки та інтерпретації даних про кампанії зі збору коштів.

Предмет дослідження. Алгоритми та методи аналітики даних, автоматизації обробки інформації, інтерпретації результатів у форматі звітів та інтеграція цих підходів у веб-інтерфейс аналітичної платформи.

Наукова новизна роботи. У межах проведеного дослідження запропоновано комплексний підхід до оцінювання волонтерської діяльності, який базується на інтеграції сучасних методів аналізу даних. Уперше для задач волонтерської аналітики реалізовано аналітичну платформу, що поєднує функціонал автоматизованої попередньої обробки даних, проведення обчислювального аналізу та автоматичного формування висновків у вигляді структурованих PDF-звітів із візуалізацією ключових показників. Особливістю розробленої системи є не лише технічна інтеграція зазначених компонентів, але й орієнтація на практичні потреби координаторів волонтерських кампаній, що забезпечує її прикладну цінність у реальних умовах.

Однією з важливих інновацій роботи стало використання штучно згенерованого датасету, який виступає повноцінною альтернативою відкритим даним, відсутність яких обмежує можливості апробації аналітичних рішень у сфері гуманітарних ініціатив. Згенерований набір даних дозволив відтворити характерні особливості інформаційного середовища волонтерської діяльності, провести тестування функціональності платформи, а також оцінити її потенціал для подальшого масштабування та адаптації під реальні прикладні кейси.

Кваліфікаційна робота магістра складається з анотації, вступу, основної частини, яка включає 4 розділи, висновків, списку використаних джерел та додатків. Всього налічує 137 сторінок та перелік посилань з 28-и джерел.

Ключові слова. Волонтерська діяльність, аналітична платформа, кластеризація, автоматизація звітності, візуалізація даних, аналітична система.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

KDD (Knowledge Discovery in Databases) – процес виявлення знань у базах даних.

SEMMA (Sample, Explore, Modify, Model, Assess) – методологія аналізу даних від SAS.

CRISP-DM (Cross-Industry Standard Process for Data Mining) – стандартна методологія data mining.

k-means – алгоритм кластеризації (метод k-середніх).

PCA (Principal Component Analysis) – аналіз головних компонент.

IQR (Interquartile Range) – інтерквартильний розмах (метод виявлення аномалій).

Z-score – стандартний бал (метод виявлення аномалій).

Silhouette Score – метрика якості кластеризації.

UtilizationRate (UR) – коефіцієнт корисного завантаження ресурсів.

Pandas – бібліотека для роботи з табличними даними.

NumPy – бібліотека для числових обчислень.

Matplotlib/Seaborn – бібліотеки для візуалізації даних.

Plotly – бібліотека для інтерактивних графіків.

Scikit-learn – бібліотека машинного навчання.

Streamlit – фреймворк для створення веб-інтерфейсів.

KoboToolbox – інструмент для збору польових даних.

Salesforce for Nonprofits – CRM-система для НКО.

DHIS2 – система для збору статистичних даних.

Категоріальні змінні – змінні, що мають обмежену кількість значень (наприклад, тип кампанії).

Нормалізація – приведення даних до єдиного масштабу.

Дендрограма – графічне представлення ієрархічної кластеризації.

Веб-інтерфейс – інтерактивний інтерфейс для роботи з платформою.

ВСТУП

У сучасних умовах суспільних викликів, особливо в умовах повномасштабної війни, волонтерський рух в Україні відіграє ключову роль у забезпеченні гуманітарної, логістичної та фінансової підтримки населення. Зростання масштабів волонтерської діяльності супроводжується істотним ускладненням процесів обробки великих обсягів інформації, координації численних ініціатив, звітності перед донорами та громадськістю. У таких реаліях особливої ваги набуває цифровізація процесів збору, аналізу та узагальнення даних, що є необхідною умовою для підвищення ефективності управління ресурсами, оперативного ухвалення рішень і забезпечення прозорості діяльності.

Актуальність теми «Розробка інтегрованої аналітичної платформи для аналізу даних та автоматизації звітності у процесі координації волонтерського руху» зумовлена нагальною потребою в застосуванні сучасних методів обробки інформації у сфері гуманітарних ініціатив. Існуючі практики збору та аналізу даних у волонтерському середовищі часто характеризуються фрагментарністю, значною часткою ручної обробки, відсутністю єдиних форматів обміну інформацією і низьким рівнем аналітичної обґрунтованості управлінських рішень. Використання автоматизованих аналітичних платформ здатне суттєво підвищити швидкість обробки запитів, забезпечити точність прогнозування потреб, оптимізувати використання ресурсів, мінімізувати дублювання ініціатив і підвищити довіру з боку донорів та громадськості.

Мета даної кваліфікаційної роботи полягає у підвищенні ефективності аналізу даних, що супроводжують діяльність волонтерських кампаній, шляхом створення інтегрованої аналітичної платформи. Розроблена система повинна забезпечити автоматизовану обробку великих обсягів інформації, виявлення закономірностей у даних, кластеризацію кампаній за подібністю характеристик, пошук аномалій у поведінці даних, а також формування аналітичних звітів у зручному для кінцевого користувача форматі через вебінтерфейс і у вигляді PDF-документів. Назва роботи — «Розробка інтегрованої аналітичної платформи для

аналізу даних та автоматизації звітності у процесі координації волонтерського руху».

У процесі виконання дослідження було сформовано та реалізовано комплекс завдань, серед яких: аналіз джерел і нормативних документів за тематикою роботи; вивчення, порівняння та вибір відповідних математичних моделей і технологій аналізу даних; моделювання процесу обробки даних та побудова аналітичної платформи; розробка технології розгортання та інтеграції системи; перевірка ефективності платформи на тестових даних, змодельованих з урахуванням особливостей реальної волонтерської діяльності.

Об'єктом дослідження є процеси аналітичного забезпечення діяльності волонтерських ініціатив, а предметом дослідження — методи кластеризації, аналізу кореляцій, виявлення аномалій, розрахунку коефіцієнтів ефективності використання ресурсів і автоматизації процесу формування звітності.

Методологічною основою роботи стали класичні та сучасні методи аналізу даних, серед яких метод k-середніх і ієрархічна кластеризація для групування кампаній за подібністю поведінки; кореляційний аналіз для виявлення залежностей між основними змінними; методи аналізу аномалій, зокрема Z-оцінка та інтерквартильний розмах для виявлення нетипових даних; розрахунок коефіцієнта корисного завантаження для оцінки ефективності використання ресурсів; метод ліктя для визначення оптимальної кількості кластерів; а також побудова візуалізацій для полегшення інтерпретації результатів аналізу кінцевими користувачами.

Інноваційність розробленої платформи полягає в інтеграції автоматичної попередньої обробки даних, аналітичних модулів і генерації звітів у єдиному веборієнтованому середовищі, що дає змогу мінімізувати необхідність залучення аналітиків і забезпечує доступність використання для широкого кола волонтерських організацій. Адаптивність і масштабованість архітектури системи відкривають можливості її застосування як для локальних ініціатив, так і в межах регіональних або міжнародних гуманітарних проектів.

Наукова новизна роботи полягає у практичному поєднанні класичних

методів аналізу даних із сучасними інструментами автоматизації обробки інформації у сфері волонтерського менеджменту. Уперше продемонстровано можливість застосування штучно згенерованого датасету як повноцінної альтернативи реальним даним для апробації аналітичних рішень у гуманітарній сфері.

Результати проведеного дослідження були апробовані на Міжнародній науковій конференції Information Technology and Implementation (Satellite): Conference Proceedings, що відбулася 20–21 листопада 2023 року в місті Київ. За результатами дослідження підготовлено наукову публікацію [1].

Крім того, результати розробки були впроваджені на реальному підприємстві з метою перевірки практичної ефективності запропонованої платформи. Для демонстрації результативності рішення проведено експертне опитування спеціалістів у галузі аналітики та управління волонтерськими кампаніями. Акт впровадження та супровідні документи наведено у додатках до кваліфікаційної роботи.

РОЗДІЛ 1
ФОРМУЛЮВАННЯ ЗАДАЧІ ДОСЛФДЖЕННЯ ТА АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ АНАЛІТИЧНОЇ ПЛАТФОРМИ

1.1 Аналіз методологій моделювання та обробки даних у сфері Data Science

Сучасний етап розвитку цифрових технологій характеризується вибуховим зростанням обсягів доступних даних, що актуалізує необхідність застосування методів Data Science у різних сферах людської діяльності. Особливої ваги ця тенденція набула у сфері волонтерської діяльності, яка, в умовах повномасштабної війни в Україні, стала ключовим механізмом гуманітарного реагування. Ефективна координація волонтерських ініціатив вимагає не лише ентузіазму учасників, а й налагоджених процесів збору, обробки та аналізу великих обсягів різнорідної інформації.

Разом із тим аналіз сучасного стану волонтерської діяльності виявляє низку системних проблем, пов'язаних з управлінням даними. Інформаційні потоки є розпорошеними між різними каналами комунікації — електронними таблицями, CRM-системами, чат-ботами, анкета́ми Google-форм. Відсутність єдиних стандартів обробки й уніфікованої звітності ускладнює аналіз ефективності кампаній та раціональне використання ресурсів. Тому створення інтегрованої аналітичної платформи на основі сучасних підходів Data Science є обґрунтованою і нагальною потребою.

У сучасній практиці роботи з даними сформувалися кілька ключових методологій, серед яких можна виділити KDD (Knowledge Discovery in Databases), SEMMA (Sample, Explore, Modify, Model, Assess) та CRISP-DM (Cross-Industry Standard Process for Data Mining). Кожна з них має свої особливості та сфери застосування.

Методологія KDD [1] виникла в академічному середовищі як концепція виявлення знань у базах даних. Її сильними сторонами є теоретична глибина і

спрямованість на побудову інтелектуальних систем. Проте на практиці реалізація KDD часто ускладнюється через недостатню увагу до етапу впровадження рішень і потребу в індивідуальній адаптації процесів для кожного окремого проєкту.

Методологія SEMMA, [2] запропонована SAS Institute, має чітко визначену послідовність кроків: вибірка даних, дослідження, модифікація, моделювання та оцінка. Цей підхід зручний для технічного аналізу даних, але в ньому відсутній акцент на бізнес- або соціальний контекст, що знижує його ефективність для гуманітарних і волонтерських проєктів.

На цьому тлі методологія CRISP-DM [3] виявляється найбільш адаптивною до потреб аналітичної роботи у волонтерській сфері. Вона поєднує переваги академічного обґрунтування й прикладного підходу до обробки даних, забезпечуючи послідовну структуру роботи — від визначення бізнес-цілей до впровадження аналітичних рішень у практичну діяльність.

Застосування CRISP-DM передбачає шість етапів: розуміння бізнес-потреб, розуміння даних, підготовка даних, моделювання, оцінка моделей та впровадження. В контексті дослідження волонтерських кампаній кожен із цих етапів набуває конкретного змістовного наповнення (Рис. 1.1).

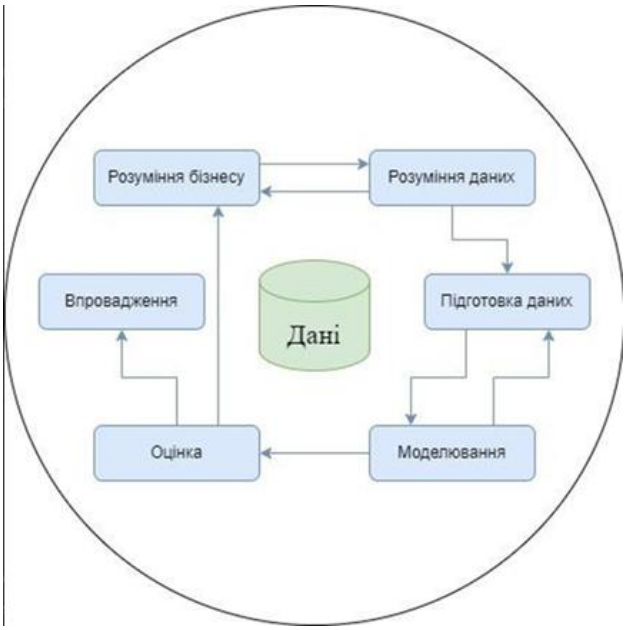

Рисунок. 1.1 – Методологія CRISP-DM – етапи аналізу даних

Перший етап — розуміння бізнес-потреб — передбачає визначення

ключових показників успішності кампаній: обсягів зборів, кількості залучених волонтерів, часу реагування на запити. На етапі розуміння даних здійснюється комплексний аналіз наявної інформації: оцінюється її якість, виявляються пропуски та аномалії, зокрема ті, що виникають через ручне введення інформації.

Особливу увагу було приділено етапу підготовки даних, адже у волонтерській сфері вихідні дані часто містять значні шуми та недостовірні записи. У ході дослідження було розроблено алгоритми автоматичного очищення, нормалізації числових показників і заповнення пропущених значень, що забезпечило підвищення точності подальшого аналізу.

На етапі моделювання застосовувалися методи кластеризації, такі як метод k-середніх для групування кампаній за подібністю характеристик, а також методи виявлення аномалій і ієрархічна кластеризація для аналізу вкладених структур даних. Оцінка моделей здійснювалася за допомогою метрик якості кластеризації, зокрема Silhouette Score, а також експертної верифікації отриманих результатів.

Завершальним етапом стало впровадження розробленого прототипу аналітичної платформи, що інтегрує функції автоматизованого збору даних, їх обробки, аналітики та формування інтерактивних звітів. Платформа орієнтована на користувачів з різним рівнем технічної підготовки завдяки інтуїтивному інтерфейсу та широким можливостям візуалізації.

Проведене дослідження підтвердило, що застосування методології CRISP-DM дозволяє систематизувати роботу з волонтерськими даними та забезпечити перехід від фрагментарного управління інформацією до стратегічного аналізу на основі даних. Завдяки своїй гнучкості, ітеративності та орієнтації на практичне впровадження, CRISP-DM виявилася найбільш придатною методологією для розв'язання завдань координації волонтерських ініціатив.

1.2 Аналіз методів аналізу даних, що застосовуються для класифікації та оцінки волонтерських кампаній

Ефективна координація волонтерських кампаній у сучасних умовах неможлива без залучення сучасних математичних та статистичних методів аналізу даних. Враховуючи великий обсяг інформації, що надходить із різних джерел, постає потреба в автоматизації процесів класифікації, виявлення закономірностей, оцінки ефективності та побудови прогнозів для прийняття обґрунтованих рішень.

У процесі дослідження було проаналізовано дванадцять методів моделювання даних, кожен із яких має власні сильні сторони та специфіку застосування. До цього переліку увійшли: метод k-середніх, ієрархічна кластеризація, метод ліктя, силуетний аналіз, кореляційний аналіз (Pearson та Spearman), методи виявлення аномалій (Z-оцінка та інтерквартильний розмах), аналіз головних компонент (PCA), алгоритм пошуку асоціативних правил Apriori, рішення дерев (Decision Tree Classifier), логістична регресія, коефіцієнт UtilizationRate та метод оцінки важливості ознак (Feature Importance / Gini Importance).

Проте, не всі зазначені методи виявилися однаково придатними для реалізації у специфічних умовах аналізу волонтерської діяльності, де пріоритетними є простота інтерпретації, швидкість обчислень та адаптивність до різних форматів даних.

На основі критеріїв релевантності, доступності й ефективності було обрано шість методів, які інтегровані у функціонал розробленої аналітичної платформи.

Таблиця 1.2 – Методи аналізу даних для побудови аналітичної платформи

Метод	Призначення	Основні переваги	Причина вибору для платформи
Метод k-середніх (k-means)	Групування кампаній за схожими	Швидкість обчислень, простота реалізації,	Оптимальний для кластеризації великих масивів

Метод	Призначення	Основні переваги	Причина вибору для платформи
	характеристиками	масштабованість	даних
Метод ліктя (Elbow Method)	Визначення оптимальної кількості кластерів	Зниження суб'єктивності вибору параметрів	Підвищення точності кластеризації
Кореляційний аналіз (Pearson, Spearman)	Виявлення взаємозв'язків між змінними	Простота застосування, відсутність вимог до типу змінних	Зручний для первинного аналізу
Аналіз аномалій (Z-Score, IQR)	Виявлення викидів і потенційно підозрілих записів	Простота реалізації, швидкість обробки	Забезпечення стабільності та надійності даних
Коефіцієнт UtilizationRate	Оцінка ефективності використання ресурсів кампанії	Специфічна метрика для волонтерських ініціатив	Відображення результативності кампаній
Ієрархічна кластеризація	Виявлення вкладених структур і аналіз подібності	Побудова дендрограм, поглиблений аналіз кластерів	Додатковий рівень деталізації для стратегічного планування

Метод k-середніх [4] виявився найкращим для групування кампаній за подібністю поведінки. Його перевагами є висока швидкість роботи, простота реалізації та можливість легко інтерпретувати результати через чітке виділення кластерних структур. Порівняно з ієрархічною кластеризацією, метод k-середніх є більш масштабованим і краще підходить для роботи з великими наборами даних.

Для вибору оптимальної кількості кластерів використовується метод ліктя. [5] Він дозволяє обґрунтувати вибір параметра k шляхом аналізу зміни дисперсії всередині кластерів, мінімізуючи ризик суб'єктивності (Рис. 1.2).

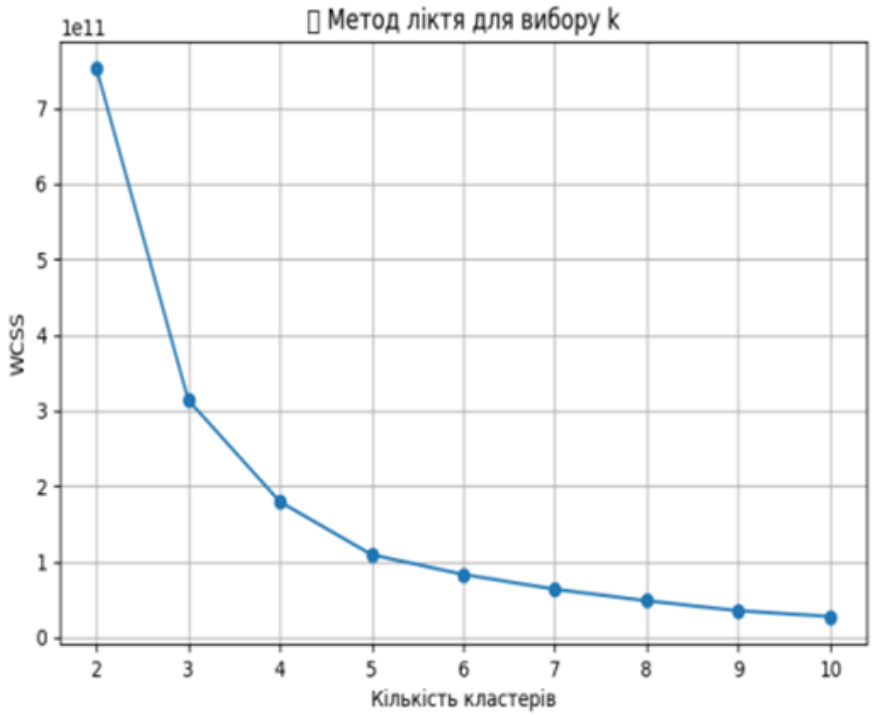

Рисунок 1.2 – Графік «Метод ліктя для вибору кількості кластерів»

Кореляційний аналіз [6] виступає ефективним інструментом для первинного виявлення взаємозв'язків між основними показниками кампаній — обсягом зборів, кількістю донатів, тривалістю активностей тощо (Рис. 1.3).

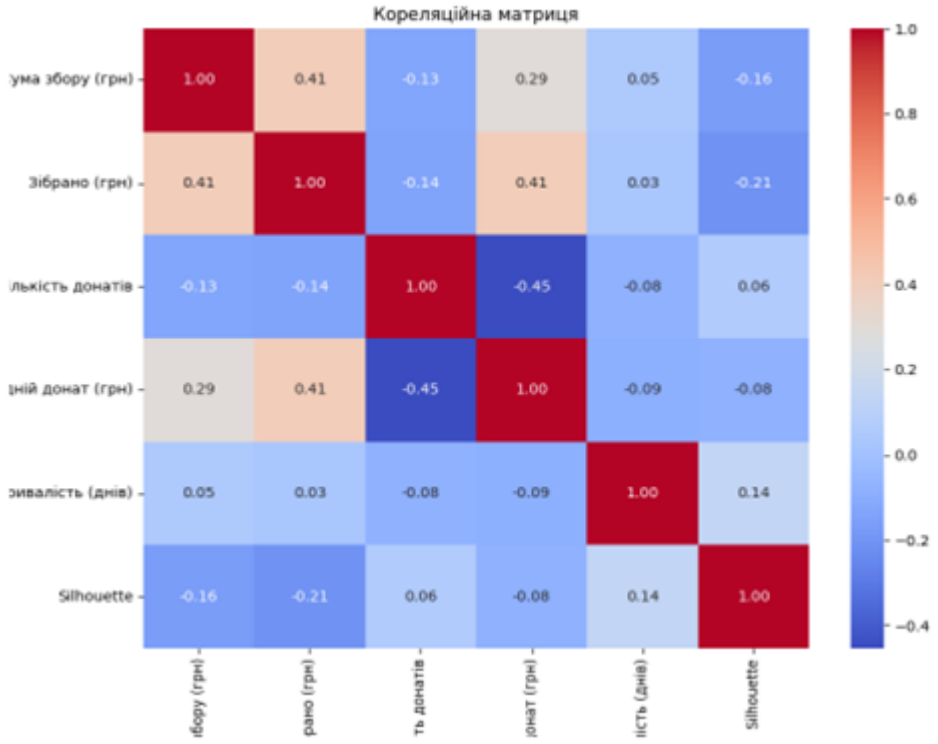

Рисунок 1.3 – Кореляційна матриця основних показників волонтерських

активностей

Зокрема, на відміну від методів регресійного аналізу, кореляційний підхід не потребує попередніх припущень про залежність змінних і є зручним для початкового аналізу даних.

Виявлення аномалій [7] здійснюється через методи Z-оцінки та інтерквартильного розмаху, що дозволяють оперативно і з мінімальними обчислювальними затратами ідентифікувати потенційні викиди у даних. Простота цих методів і відсутність необхідності в навчанні на розмічених вибірках робить їх більш придатними для роботи з реальними даними волонтерських кампаній у порівнянні з деревами рішень або логістичною регресією.

Коефіцієнт корисного завантаження (UtilizationRate) [8] введено як спеціальну метрику для оцінки ефективності використання залучених ресурсів. Його розрахунок дозволяє кількісно оцінити співвідношення між використаними і потенційними ресурсами кампанії.

Ієрархічна кластеризація [9] застосовується як допоміжний метод для поглибленого аналізу подібності між кампаніями. Вона дозволяє будувати дендрограми, які дають уявлення про багаторівневу структуру даних, що особливо корисно для стратегічного планування подальших волонтерських активностей (Рис. 1.4).

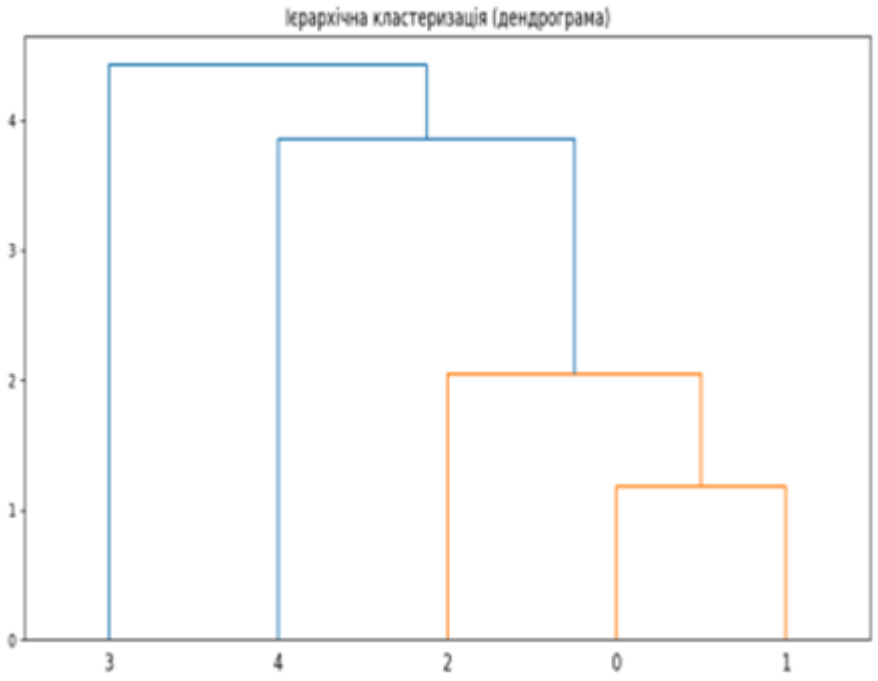

Рисунок 1.4 – Дендрограма ієрархічної кластеризації

Методи, які не були інтегровані до основного ядра платформи, також мають свої переваги. Зокрема, метод головних компонент (PCA) [10] забезпечує зменшення розмірності даних, а методи оцінки важливості ознак (Feature Importance) [14] допомагають ідентифікувати ключові фактори впливу. Проте інтерпретація результатів цих методів потребує глибокої математичної підготовки, що може ускладнити їх використання серед широкого кола користувачів платформи.

Методи класифікації, такі як логістична регресія [13] та дерева рішень, орієнтовані на роботу із заздалегідь розміченими даними. В умовах волонтерської діяльності, де дані часто надходять у необробленому вигляді і відсутні чіткі мітки, їх застосування є обмеженим.

Щодо алгоритмів пошуку асоціативних правил, таких як Apriori, [11] їх ефективність проявляється у великих транзакційних наборах даних, які не є характерними для волонтерських кампаній із змінною і гетерогенною структурою.

Загалом, обраний набір методів забезпечує оптимальний баланс між ефективністю аналізу, адаптивністю до реальних даних, обчислювальною

простотою та доступністю інтерпретації для користувачів з різним рівнем підготовки.

Досвід попередніх досліджень [18-20] у сфері аналізу транзакційної активності та виявлення шахрайства свідчить про ефективність комбінованого використання кластеризації, кореляційного аналізу та виявлення аномалій. Підхід, за якого спочатку формується модель "нормальної" поведінки на основі історичних даних, а потім ідентифікуються відхилення, є універсальним і може бути ефективно адаптованим для моніторингу діяльності волонтерських кампаній.

Таким чином, запропоновані методи дозволяють не лише системно аналізувати дані, а й формувати на їх основі практичні рекомендації щодо підвищення ефективності волонтерських ініціатив.

1.3 Аналіз інформаційних систем та платформ орієнтованих на обробку соціальних та гуманітарних даних

У сучасних умовах ефективне управління волонтерськими кампаніями неможливе без застосування спеціалізованих інформаційних систем, які забезпечують збір, зберігання, обробку та аналіз даних. З огляду на специфіку гуманітарної діяльності, де часто використовується різнорідна, частково структурована інформація, особливо важливою стає можливість адаптивного налаштування платформ під реальні потреби ініціатив.

На сьогодні існує кілька провідних систем, які активно використовуються для управління даними в гуманітарних проєктах. До них належать KoboToolbox, [15] Salesforce for Nonprofits [16] та DHIS2. [17] Кожна з цих платформ має свої сильні сторони, однак повною мірою не задовольняє вимоги системної аналітики у сфері волонтерської діяльності.

KoboToolbox є одним із найбільш відомих інструментів для збору польових даних. Вона була створена спеціально для роботи в гуманітарних умовах і дозволяє легко створювати електронні форми опитування, збирати дані

через мобільні пристрої без доступу до Інтернету та інтегрувати результати у бази даних. Основною перевагою KoboToolbox є її орієнтація на простоту використання та роботу в складних умовах. Проте платформа обмежена у можливостях гнучкого аналізу даних: побудова складних моделей, багаторівневих залежностей або автоматичної класифікації даних потребує суттєвого доопрацювання або експорту даних у сторонні системи.

Salesforce for Nonprofits пропонує більш комплексний підхід до управління інформацією про діяльність некомерційних організацій. Система надає широкі можливості для налаштування взаємодії з донорами, волонтерами, збору пожертв та управління кампаніями. Її сильна сторона — глибока інтеграція різних аспектів діяльності організації в єдину CRM-систему. Водночас, для повноцінного використання аналітичних можливостей Salesforce необхідна розробка спеціалізованих додатків або налаштувань, що значно ускладнює застосування платформи у середовищі, де ресурсів на технічну підтримку обмежено. Крім того, структура Salesforce переважно орієнтована на обробку фіксованих взаємодій із клієнтами, що не завжди узгоджується з динамічною природою волонтерських кампаній.

DHIS2 спочатку створювався як система збору статистичних даних для охорони здоров'я та громадського сектору. Його сильними сторонами є можливості масштабного збору даних, побудова статистичних звітів та інтеграція з різноманітними форматами даних. Проте застосування DHIS2 у сфері волонтерських ініціатив обмежується відносною жорсткістю структури даних, складністю налаштувань і невисокою гнучкістю при потребі швидкої адаптації до змін у структурі або змісті кампаній.

Порівняльний аналіз зазначених платформ за основними критеріями — простотою налаштування, гнучкістю аналізу, можливістю інтеграції та масштабованістю — свідчить про те, що кожна з них має цінні переваги для гуманітарної діяльності, проте жодна не забезпечує повного покриття потреб у системній аналітиці саме для волонтерських кампаній.

Таблиця 1.3 – Порівняльна характеристика платформ для обробки

Платформа	Простота налаштування	Гнучкість аналізу	Інтеграція	Масштабованість	Основні обмеження
KoboToolbox	Висока	Обмежена	Середня	Середня	Недостатня гнучкість для складного аналізу
Salesforce for Nonprofits	Низька (без підтримки)	Висока	Висока	Висока	Високі вимоги до технічної підтримки
DHIS2	Середня	Середня	Висока	Висока	Жорстка структура даних, складність адаптації

гуманітарних даних

Аналіз наведених систем свідчить про те, що жодна з існуючих платформ не є ідеальним рішенням для повноцінної автоматизації аналізу даних волонтерських кампаній. KoboToolbox обмежується збором даних, Salesforce for Nonprofits вимагає значних технічних ресурсів для налаштування аналітичних функцій, тоді як DHIS2 орієнтований на строго формалізовану структуру даних, що ускладнює швидке реагування на змінні потреби.

У цьому контексті виникає обґрунтована необхідність у розробці власної аналітичної платформи, яка поєднуватиме переваги існуючих рішень — зручність збору даних, гнучкість налаштування моделей аналізу, адаптивність до змін у структурі волонтерських кампаній і можливість автоматичного формування аналітичних звітів без потреби у складному програмуванні з боку кінцевого користувача.

1.4 Постановка задачі та обґрунтування необхідності розробки нової аналітичної платформи

Аналіз існуючих методів моделювання та платформ для обробки даних у

сфері волонтерської діяльності свідчить про те, що наявні рішення лише частково відповідають специфічним потребам координації гуманітарних ініціатив. Попри наявність розвинених інструментів збору інформації або управління проектами, системна аналітика залишається недостатньо розвиненою, що ускладнює прийняття стратегічних рішень, оптимізацію ресурсів та підвищення ефективності кампаній.

Однією з найгостріших проблем у сфері волонтерської координації залишається дублювання запитів, коли кілька організацій одночасно реагують на одні й ті самі потреби, не маючи доступу до спільної бази даних або актуального статусу виконання завдань. Це призводить до перевитрати ресурсів, накладення логістичних маршрутів та втрати довіри з боку отримувачів допомоги, особливо в ситуаціях, коли один регіон отримує надлишок підтримки, тоді як інший залишається без необхідних ресурсів.

Розрізненість форматів звітності є ще одним критичним бар'єром для ефективної координації. Різні організації використовують несумісні між собою інструменти: Excel-файли, Google Таблиці, паперові носії або сторонні сервіси. Це породжує інформаційний хаос, унеможливлює створення консолідованих дашбордів та перешкоджає оперативному аналізу пріоритетів у масштабах району, області або країни.

Додатковою складністю є висока частка ручної обробки інформації. Координатори витрачають значні ресурси часу на оновлення таблиць, зведення даних про надходження і формування звітів. Помилки, дублювання записів або втрати інформації внаслідок людського фактора стають неминучими, що особливо критично в умовах війни, коли своєчасність прийняття рішень має прямий вплив на ефективність допомоги.

Відсутність централізованої бази даних поглиблює фрагментацію інформації, адже дані про запити, відправлення, отримувачів та донорів зберігаються локально на особистих пристроях чи окремих сховищах. Це створює острівну структуру взаємодії, у якій об'єднання зусиль або прогнозування обсягів потреб є надзвичайно ускладненими.

Непрозоре управління ресурсами також є серйозним викликом. Через брак аналітичних інструментів організації не завжди можуть чітко відповісти на питання про використання отриманих коштів, ефективність окремих напрямів або порівняльний аналіз результатів локальних і загальнонаціональних кампаній. Це знижує довіру з боку донорів і негативно впливає на стійкість волонтерських структур.

Усі зазначені проблеми підкреслюють нагальну потребу у створенні єдиної інтегрованої аналітичної платформи, яка дозволить централізовано збирати й обробляти інформацію про кампанії, автоматизувати рутинні процеси підрахунку донатів, формування звітів і побудови графіків, аналізувати тренди, аномалії, кластери та залежності між показниками, визначати зони найвищої потреби на основі реальних даних, а також формувати прозору й адаптивну модель управління ресурсами.

Розробка такої системи має базуватися на принципах доступності, гнучкості, простоти інтеграції та мінімізації вимог до спеціальної технічної підготовки користувачів. Інтуїтивно зрозумілий інтерфейс, автоматичне формування візуалізацій, можливість гнучкої адаптації до змін у структурі даних мають стати обов’язковими характеристиками платформи.

Формалізуючи задачі дослідження, можна виділити такі ключові напрями:

- Розробити прототип аналітичної платформи для збору, обробки та аналізу даних волонтерських кампаній.
- Інтегрувати в систему методи кластеризації, кореляційного аналізу, виявлення аномалій, розрахунку коефіцієнта UtilizationRate.
- Забезпечити можливість генерації автоматизованих аналітичних звітів і графічних інтерфейсів для різних рівнів користувачів.
- Забезпечити гнучкість адаптації системи до змін у структурі даних без потреби серйозних технічних втручань.

У перспективі створення такої платформи відкриває шлях до цифрової трансформації волонтерського руху в Україні. Це сприятиме підвищенню стійкості ініціатив, поліпшенню координації між організаціями та формуванню довіри до волонтерських проектів серед донорів і громадськості.

1.5 Висновки до розділу 1

У результаті проведеного аналізу сучасних підходів до моделювання та обробки даних було обґрунтовано вибір методології, методів аналітики та визначено основні проблеми, що стоять перед системою координації волонтерських кампаній.

Аналіз загальних підходів у сфері Data Science показав, що серед існуючих методологій найкраще потребам дослідження відповідає CRISP-DM. Ця методологія забезпечує гнучку структуру роботи з даними, дозволяє ітеративно вдосконалювати моделі та орієнтується на практичне застосування результатів, що особливо важливо в умовах динамічної волонтерської діяльності.

У межах дослідження здійснено порівняння та оцінку різних методів аналізу даних, зокрема методів кластеризації, кореляційного аналізу, виявлення аномалій, розрахунку ефективності використання ресурсів. Виявлено, що поєднання методу k-середніх, методу ліктя, кореляційного аналізу, методів виявлення аномалій та розрахунку коефіцієнта UtilizationRate забезпечує оптимальний баланс між точністю результатів, швидкістю обробки та зрозумілістю інтерпретації для користувачів різного рівня підготовки.

Проведений огляд існуючих інформаційних платформ для обробки соціальних і гуманітарних даних, таких як KoboToolbox, Salesforce for Nonprofits та DHIS2, продемонстрував, що хоча кожна з них має окремі переваги, жодна не здатна повною мірою задовольнити потреби системної аналітики волонтерських кампаній без значних доопрацювань. Обмеження стосуються або недостатньої гнучкості аналізу, або високих вимог до технічної підтримки, або складності адаптації до змінних умов гуманітарних ініціатив.

На основі виявлених проблем сформульовано основні виклики, з якими стикаються волонтерські організації: дублювання запитів через відсутність актуальної інформації, розрізненість форматів обробки даних, велика частка ручної роботи, фрагментованість баз даних і брак прозорого управління

ресурсами. Такі проблеми знижують ефективність діяльності, ускладнюють координацію та можуть призводити до втрати довіри з боку бенефіціарів і донорів.

Встановлено нагальну потребу у створенні інтегрованої аналітичної платформи, яка дозволить централізувати збір і обробку даних, автоматизувати рутинні процеси, аналізувати тренди, виявляти аномалії, формувати зрозумілі звіти та забезпечувати прозоре управління ресурсами. Така система має відповідати вимогам адаптивності до змін структури даних, мінімізації технічних бар'єрів для користувачів і оперативності аналізу в кризових умовах.

Проведене дослідження створює наукове та практичне підґрунтя для переходу до наступного етапу — розробки загальної методики побудови аналітичної платформи, що дозволить підвищити ефективність, стійкість та довіру до волонтерського руху в Україні.

РОЗДІЛ 2

МЕТОДИ ТА МОДИКИ ПОБУДОВИ ІНТЕГРОВАНОЇ АНАЛІТИЧНОЇ ПЛАТФОРМИ

2.1 Загальна методики створення платформи для аналізу волонтерських даних

Побудова інтегрованої аналітичної платформи для аналізу волонтерської активності передбачає створення багаторівневої системи, яка забезпечує обробку, аналіз, зберігання та візуалізацію даних з подальшою генерацією звітів. Основною метою цієї платформи є підвищення ефективності прийняття рішень у сфері координації волонтерських ініціатив за рахунок автоматизації аналітичних процесів.

Аналітична система базується на модульному принципі побудови, що дозволяє легко масштабувати або адаптувати платформу до різних форматів даних та методів аналізу. Архітектурно система поділяється на такі основні компоненти:

- Модуль завантаження даних — відповідає за інтеграцію з вхідними джерелами (Excel), обробку структури файлів та первинну валідацію.
- Модуль попередньої обробки — реалізує очищення, нормалізацію, перетворення типів, обробку пропущених значень.
- Аналітичні модулі — включають реалізацію методів кластеризації, кореляційного аналізу, виявлення аномалій, обчислення Utilization Rate тощо.
- Модуль візуалізації — формує графіки, таблиці та інші форми представлення даних для зручності інтерпретації.
- Модуль генерації звітів — автоматично формує підсумкові звіти у форматі PDF, включаючи ключові метрики, графіки та текстові висновки.
- Веб-інтерфейс — забезпечує інтерактивну взаємодію користувача із системою, дозволяє обирати методи аналізу, фільтрувати змінні та переглядати результати в режимі реального часу.

Для реалізації інтегрованої аналітичної платформи було обрано мову програмування Python, яка на сьогодні є однією з найпопулярніших у сфері аналізу даних. Це високорівнева, інтерпретована мова зі строгою динамічною

типізацією, що підтримує кілька парадигм програмування — об’єктно-орієнтовану, процедурну та функціональну. Однією з ключових переваг Python є його чистий синтаксис, що значно пришвидшує розробку, а також величезна екосистема бібліотек, які дозволяють ефективно працювати з даними, будувати моделі, візуалізувати результати й інтегруватися з іншими платформами.

У процесі реалізації проєкту використовувались різноманітні бібліотеки Python, кожна з яких відіграє певну роль у загальній архітектурі аналітичної платформи.

Бібліотека Pandas [22] стала основним інструментом для роботи з табличними структурами даних. Вона дозволяє зчитувати інформацію з різних форматів (наприклад, Excel або CSV), виконувати фільтрацію, групування, агрегування та підготовку даних до подальшого аналізу. Саме за допомогою Pandas реалізовано попередню обробку вхідних файлів та трансформацію наборів для кластеризації, кореляційного аналізу та пошуку аномалій

Для виконання числових операцій використовувалась бібліотека NumPy.[23] Вона забезпечує роботу з багатовимірними масивами та дозволяє виконувати векторизовані операції, що значно пришвидшує обчислення порівняно зі звичайними циклами Python. У межах платформи NumPy використовувалась для обчислення статистичних характеристик та нормалізації даних

Важливу роль у візуалізації результатів відіграли бібліотеки Matplotlib [24] та Seaborn [25]. Matplotlib забезпечує гнучкий інструментарій для побудови графіків різних типів: від простих лінійних до складних комбінованих діаграм. У свою чергу, Seaborn побудована поверх Matplotlib і спрощує створення статистичних графіків. Завдяки цим бібліотекам було реалізовано графічне відображення результатів кластеризації, а також побудовано теплові карти для кореляційного аналізу.

Інтерактивні графіки, які відображаються безпосередньо у вебінтерфейсі, реалізовано з використанням бібліотеки Plotly [26], що дозволяє створювати глибоко інтерактивні графіки з можливістю масштабування, наведення на

елементи, та експорту.

Для реалізації методів машинного навчання було обрано бібліотеку scikit-learn, яка містить інструменти для кластеризації (зокрема алгоритм KMeans), обчислення метрик якості кластеризації (наприклад, silhouette score), а також реалізацію різноманітних статистичних аналізів. Для роботи зі статистичними розподілами та виявлення аномальних значень додатково використовувалась бібліотека scipy.stats .

З метою організації збереження проміжних результатів у зручному форматі використовувалась стандартна бібліотека json, а для взаємодії з файловою системою — бібліотека **os**. Додатково, для генерації імен звітів із часовими мітками використовувалась бібліотека time.

Ключовим елементом зручності користування системою став вебінтерфейс, реалізований на основі фреймворку Streamlit [27]. Завдяки цьому фреймворку платформа має простий, але гнучкий інтерфейс, що дозволяє завантажити файл, обрати змінні для аналізу, запустити відповідні методи та одразу переглянути результати аналізу та графіки. Streamlit дозволяє легко масштабувати платформу, додаючи нові методи без зміни структури основного інтерфейсу.

Для формування звітів у форматі PDF було використано бібліотеки fpdf2 та ReportLab. [28] Перша з них дозволяє швидко створювати прості PDF-документи з таблицями та зображеннями, а друга — більш потужна, дає змогу виводити текст із точним позиціонуванням, формувати графіки безпосередньо у звіті та структурувати документ на сторінки з власним дизайном.

Таким чином, використання сучасного інструментарію Python забезпечило реалізацію потужної, масштабованої та зручної платформи, здатної ефективно обробляти дані, автоматизувати звітність та візуалізувати аналітичні результати у зручній формі як для спеціалістів, так і для кінцевих користувачів.

Процес побудови аналітичної платформи реалізується через послідовне проходження даних через відповідні модулі (Рис. 2.1). Після завантаження та обробки даних користувач обирає метод аналізу, вводить змінні, запускає

обчислення, після чого отримує візуалізовані результати та можливість згенерувати фінальний звіт.

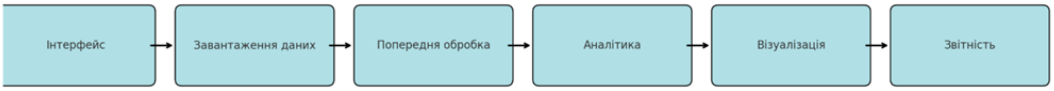

Рисунок 2.1 – Послідовність етапів роботи аналітичної платформи

Платформа побудована з урахуванням таких принципів:

- Модульність — кожна функція винесена в окремий файл або блок коду для зручності підтримки та розвитку.
- Повторне використання коду — функції обробки даних та візуалізації використовуються в усіх модулях повторно.
- Інтерактивність — користувач самостійно формує вхідні параметри аналізу.
- Автоматичність — процеси класифікації, підрахунків, створення графіків та PDF-звітів відбуваються без потреби ручного втручання.

2.2 Принципи підготовки, очищення та нормалізації даних для аналізу

Підготовка даних є критично важливим етапом у реалізації будь-якої аналітичної платформи, особливо у сфері волонтерського управління, де джерела інформації часто є фрагментованими, неоднорідними, містять помилки або пропуски. Своєчасна й систематизована обробка даних не лише забезпечує коректну роботу алгоритмів кластеризації, виявлення аномалій та кореляційного аналізу, а й формує основу для подальшої автоматизації звітності й ухвалення рішень.

Імпорт даних. Завантаження інформації з різних джерел (наприклад, таблиць Excel) здійснюється за допомогою бібліотеки pandas. На цьому етапі проводиться перевірка структури файлу, відповідність назв колонок, аналіз типів змінних, а також попередня оцінка обсягу наявної інформації.

Очистка даних:

- Видалення зайвих пробілів, спецсимволів і непотрібного форматування в назвах колонок.

- Уніфікація значень категоріальних змінних (наприклад, "Збір на ЗСУ", "Техніка", "Допомога фронту").
- Перетворення текстових числових колонок у тип float, де це можливо.
- Усунення пропущених значень або заповнення їх обґрунтованими статистичними показниками — середнім, медіаною чи нулем, залежно від контексту.

Робота з категоріальними змінними та їх групуванням. Категоріальні змінні (такі як тематика кампанії, організатор, регіон тощо) вимагають особливої уваги. Часто вони містять велику кількість варіантів, серед яких можуть бути дублікати, синоніми або малозначущі категорії.

На етапі підготовки:

- Значення у категоріальних колонках уніфікуються (наприклад, усі варіанти, що містять «ЗСУ», зводяться до однієї категорії).
- Малочастотні категорії групуються в умовну групу «Інше», щоб уникнути перевантаження моделей несуттєвими ознаками.
- Для подальшого використання у моделях змінні кодуються — наприклад, за допомогою one-hot-encoding або порядкового кодування (OrdinalEncoder).

Таке групування дозволяє значно підвищити інтерпретованість результатів, провести сегментацію за категоріями та порівняти ефективність кампаній не лише за кількісними, а й за якісними характеристиками.

Конвертація типів. Колонки з датами (наприклад, *Початок збору* і *Кінець збору*) перетворюються у формат datetime. Це дає змогу будувати часові графіки, розраховувати тривалість кампанії та виявляти сезонні або тимчасові закономірності.

Форматування нових змінних. На основі обчислень формуються похідні змінні, які дають більш глибоке розуміння ефективності:

- UtilizationRate = Зібрано / Сума збору;
- Середній донат = Зібрано / Кількість донатів;
- Тривалість = Кінець кампанії – Початок кампанії.

Нормалізація (стандартизація). Для алгоритмів, чутливих до масштабів, наприклад, k-means, проводиться нормалізація значень змінних за допомогою StandardScaler. Це дозволяє уникнути переважання змінних із великим діапазоном значень і забезпечити адекватну роботу алгоритмів.

Відбір релевантних змінних. Проводиться фільтрація змінних для кожного з методів аналізу:

- Для кластеризації — обираються кількісні метрики (зокрема, зібрано, кількість донатів, тривалість, UtilizationRate);
- Для кореляційного аналізу — лише числові показники, що мають достатню варіативність;
- Для виявлення аномалій — змінні з відомим або очікувано стабільним розподілом.

У межах дослідження використано спеціально згенерований датасет, що імітує реальні кампанії збору коштів в Україні. До складу входять як числові змінні (зокрема, кількість донатів, сума збору, тривалість), так і категоріальні (тип кампанії, організатор), а також часові мітки. Це дало змогу забезпечити реалістичність моделювання без використання чутливих персональних даних, водночас охопивши типові сценарії, виклики та особливості, з якими стикаються волонтерські ініціативи в умовах воєнного часу.

2.3 Реалізація кластерного аналізу волонтерських кампаній

Кластерний аналіз є одним із ключових елементів дослідження, що дозволяє структурувати волонтерські кампанії за спільними характеристиками, такими як обсяг збору, кількість донатів, середній донат, тривалість кампанії тощо. Це забезпечує можливість виявити типові групи кампаній, які демонструють подібні поведінкові або організаційні ознаки, та адаптувати управлінські рішення відповідно до їх профілю.

У проєкті застосовано метод k-середніх — один із найпоширеніших алгоритмів кластеризації. Його основна ідея полягає у знаходженні k центрів

кластерів, до яких будуть віднесені об'єкти вибірки на основі мінімізації відстані до центроїда.

Алгоритм працює за наступною схемою:

1. Визнвчення початкових центрів кластерів.
2. Призначення кожного об’єкта до найближчого центроїда, використовуючи Евклідову відстань (1):

$$D_E = \sqrt{\sum_{i=1}^{n} (xi - yi)^2} \quad (1)$$

   де хі та уі – координати точок у просторі ознак.
3. Оновлення центрів кластерів шляхом обчислення середнього значення координат усіх точок, які входять у кластер (2):

$$\mu i = \frac{1}{|Ci|} \sum_{x \epsilon Ci} x \quad (2)$$

   де Сі – множина точок у кластері, а μi – центр кластеру.
4. Повторення процесу до тих пір, поки зміни у розташуванні центрів кластерів не стануть незначними.

Для визначення оптимального значення k використовується **метод ліктя**, який аналізує зміну внутрішньокластерної дисперсії (WCSS) (3):

$$WCSS = \sum_{i=1}^{k} \sum_{x \epsilon Ci} \left\| x - \mu i \right\|$$

де WCSS – сумарна відстань точок до центрів кластерів,
Сі – множина точок у кластері,
μi – центр кластеру.

Оптимальне значення k відповідає "зламу" на графіку зміни WCSS при збільшенні кількості кластерів (Рис. 1.2).

Метод k-середніх є одним із найпопулярніших підходів у задачах кластеризації, зокрема завдяки своїй простоті, швидкодії та зрозумілості результатів. Однією з ключових переваг цього методу є легкість у реалізації: алгоритм складається з інтуїтивно зрозумілих кроків — визначення центроїдів, призначення об’єктів до найближчого центру, повторне обчислення центрів — і може бути швидко впроваджений навіть у невеликих проєктах.

Ще однією сильною стороною є висока обчислювальна ефективність, що

дозволяє застосовувати метод до великих масивів числових даних. Завдяки цьому його зручно використовувати в аналітичних платформах із регулярним оновленням інформації. Крім того, результат кластеризації легко інтерпретується: кожен кластер має чітко окреслені межі навколо центроїда, що дозволяє узагальнювати типові профілі об’єктів у вибірці.

Водночас метод має і низку обмежень. Насамперед, необхідність попереднього задання кількості кластерів (k) може бути проблематичною у випадках, коли структура даних є невідомою або змінюється з часом. Для вирішення цього недоліку часто застосовують додаткові методи, як-от метод ліктя, однак і вони не завжди дають однозначну відповідь.

Ще одна проблема — чутливість до початкового вибору центроїдів. Якщо вони обрані невдало, алгоритм може зійтися до локального мінімуму та дати неякісний поділ даних. Для зменшення цього ризику рекомендується багаторазовий запуск з різними початковими точками або використання модифікацій, таких як `k-means++`.

Також варто зазначити, що метод погано справляється з шумовими або складно структурованими даними. У випадках, коли кластери мають нестандартну форму (наприклад, витягнуті або вкладені один в одного), або вибірка містить значну кількість викидів, результати кластеризації можуть бути спотвореними.

У підсумку, k-means є ефективним інструментом для кластеризації добре структурованих числових даних, однак потребує ретельної підготовки вхідних даних та додаткових кроків для оцінки якості розподілу. У межах даного дослідження цей метод показав себе як оптимальний варіант для групування волонтерських кампаній за ключовими характеристиками.

2.4 Проведення кореляційного аналізу між ключовими показниками ініціатив

Кореляційний аналіз є важливим інструментом у дослідженні

залежностей між числовими змінними. У контексті аналітичної платформи для волонтерського руху він дозволяє виявити, які показники пов’язані між собою, наприклад — чи впливає кількість донатів на обсяг зібраних коштів або чи залежить ефективність кампанії від її тривалості.

У цьому проєкті застосовується коефіцієнт кореляції Пірсона (r), який вимірює силу та напрям лінійного зв’язку між двома змінними [16]. Формула (4):

$$r = \frac{\sum(xi - \bar{x})(yi - \bar{y})}{\sqrt{\sum(xi - \bar{x})^2}\sqrt{\sum(yi - \bar{y})^2}} \quad (4)$$

де x та y – аналізовані змінні, $\bar{x}$ та $\bar{y}$ – їх середні значення. Значення r знаходиться в межах від -1 до 1, де -1 означає повну негативну кореляцію, 1 – повну позитивну кореляцію, а 0 – відсутність зв’язку між змінними (Рис. 1.3).

Його основною перевагою є простота у реалізації та обчислення, що робить цей підхід зручним для початкового аналізу великих обсягів даних. Коефіцієнт кореляції, зокрема Пірсона, легко інтерпретується: значення близьке до 1 або -1 свідчить про сильний зв’язок (відповідно прямий або обернений), а значення, близьке до нуля, — про його відсутність.

Кореляційний аналіз також забезпечує наочне уявлення про взаємозалежності між змінними, що дозволяє формулювати гіпотези та виявляти потенційні закономірності у волонтерських кампаніях. Наприклад, у межах цього дослідження було встановлено, що зі збільшенням кількості донатів зазвичай зростає і загальна сума збору, але при цьому зменшується середній розмір одного внеску. Такі висновки можуть слугувати основою для побудови складніших моделей прогнозування або оптимізації ресурсів.

Водночас, попри свою доступність, метод має низку суттєвих обмежень. Насамперед, кореляція виявляє лише лінійні зв’язки, залишаючи поза увагою складніші або нелінійні залежності, які також можуть бути суттєвими для аналізу. Якщо змінні пов’язані, наприклад, експоненційно чи логарифмічно, стандартний коефіцієнт Пірсона цього не покаже.

Ще один недолік — чутливість до викидів. Наявність кількох аномальних значень у вибірці може істотно змінити величину коефіцієнта, спотворюючи

реальну картину. Тому перед проведенням аналізу важливо виконати попереднє очищення або додаткові перевірки стабільності результатів.

Окрім цього, варто наголосити, що кореляція не встановлює причинно-наслідкових зв'язків. Тобто навіть якщо дві змінні демонструють високий ступінь залежності, це ще не означає, що одна з них є причиною змін в іншій. У цьому контексті результати кореляційного аналізу слід трактувати з обережністю, особливо при ухваленні управлінських рішень.

У підсумку, кореляційний аналіз залишається ефективним інструментом для первинної аналітики даних, який дає змогу швидко зорієнтуватися в основних взаємозв'язках, але потребує критичного осмислення та часто — доповнення іншими методами дослідження

2.5 Застосування методів виявлення аномалій у даних

У системах, що працюють з великими обсягами даних, **виявлення аномалій** є важливим елементом контролю якості інформації. У сфері волонтерської діяльності це особливо актуально: аномальні значення можуть свідчити про помилки у введенні даних, нечесну діяльність або виняткові (нестандартні) кейси.

Аномалії — це спостереження, які значно відрізняються від решти даних. У волонтерських кампаніях такими можуть бути:

- надзвичайно високі або низькі суми збору;
- незвично коротка або довга тривалість кампанії;
- непропорційна кількість донатів (наприклад, 3 донати на 300 тис. грн).

Метод відхилення від середнього (Z-оцінка). Класичний статистичний підхід, де аномальними вважаються значення, що виходять за межі (5):

$$Z = \frac{(x - \bar{x}}{\sigma} \quad (5)$$

Аномалії: $|z| > 3$

Метод міжквартильного розмаху (IQR). Підхід базується на розрахунку

межі (6):

$$IQR = Q3 - Q1 \quad (6)$$

Нижня межа = Q1 – 1.5 * IQR

Верхня межа = Q3 + 1.5 * IQR

Він є менш чутливим до розподілу даних, ніж Z-оцінка, і особливо ефективний при роботі з асиметричними розподілами.

Кластерний підхід. Аномалії визначаються як точки, що:

- не потрапляють у жоден із кластерів (наприклад, низька ймовірність приналежності);
- мають значну відстань до центроїдів кластерів.

Цей підхід дозволяє виявити нетипові кампанії у контексті кластерної структури (Рис. 2.5).

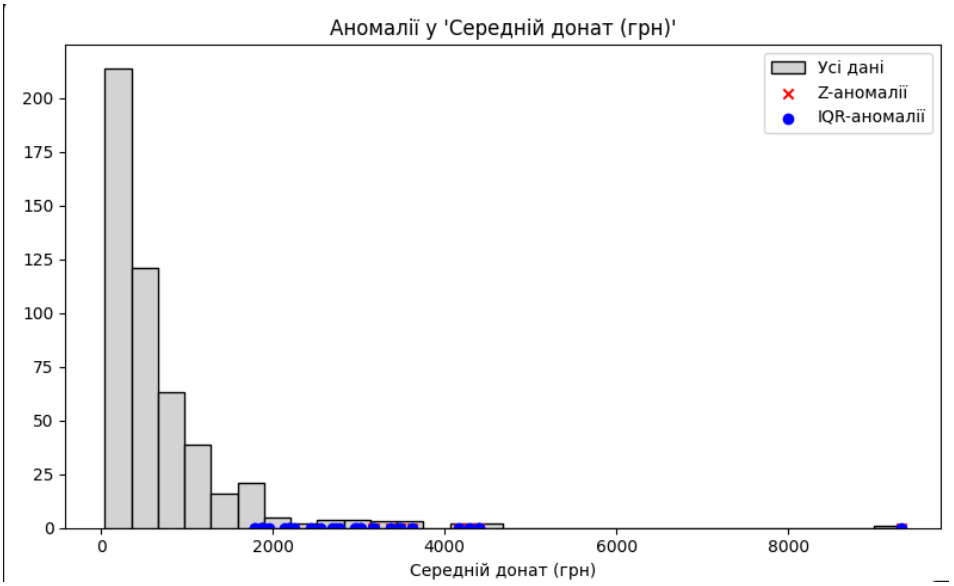

Рисунок 2.5 – Виявлення аномалій у значеннях середнього донату

Однією з основних переваг такого аналізу є його здатність виявляти спостереження, які суттєво відрізняються від загальної маси — як у вигляді помилок введення, так і у вигляді рідкісних, але значущих подій (наприклад, одиничних великих донатів або надто коротких кампаній із високим результатом). Завдяки цьому зростає точність узагальнюючих моделей і звітів, оскільки виключення або окрема інтерпретація таких значень дозволяє уникнути

спотворення результатів.

Ще однією сильною стороною є гнучкість у застосуванні: сучасні алгоритми здатні працювати як з числовими, так і з категоріальними або часовими змінними, використовуючи різні статистичні або машинні підходи (наприклад, Z-оцінку, міжквартильний розмах, або класифікатори з низькою ймовірністю належності до групи).

Водночас аналіз аномалій має і свої обмеження, які слід враховувати при практичному впровадженні. Зокрема, встановлення порогів для виявлення аномалій часто є емпіричним і базується на досвіді або припущеннях аналітика. Це створює ризик суб'єктивності в інтерпретації результатів: одна й та сама точка може вважатися як шумом, так і ключовою закономірністю — залежно від контексту.

Ще один виклик полягає у високій чутливості до масштабу даних та розподілу. Якщо змінні не нормалізовані або мають різну дисперсію, результати можуть бути нерепрезентативними або непослідовними. Тому перед виявленням аномалій обов'язковим етапом є стандартизація змінних та відбір відповідних метрик.

У деяких випадках аномалії можуть бути не лише небажаними відхиленнями, а й потенційно цінними індикаторами нових тенденцій (наприклад, ефективних тактик збору). Таким чином, автоматизований підхід до виявлення аномалій потребує обережної інтерпретації результатів і часто вимагає додаткового експертного аналізу.

Загалом, виявлення аномалій є незамінним елементом аналітичного процесу, особливо в умовах великого обсягу неоднорідних даних. Воно підвищує достовірність висновків, виявляє критичні ситуації і забезпечує глибше розуміння структури даних.

2.6 Обчислення коефіцієнта корисного завантаження ресурсів кампаній (UtilizationRate)

У волонтерських кампаніях важливо не лише знати загальні суми зборів, а й оцінювати ефективність використання наданих ресурсів. Саме для цього використовується коефіцієнт корисного завантаження (Utilization Rate, ККЗ), який демонструє, наскільки ефективно ініціативи реалізують поставлені фінансові цілі.

Utilization Rate — це відношення фактично зібраних коштів до цільової суми збору (7):

$$UR = \frac{\text{Зібрано (грн)}}{\text{Сума збору (грн)}} \quad (7)$$

У відсотковому вираженні (8):

$$UR\% = UR * 100 \quad (8)$$

UR < 0.5 – Компанія з низькою результативністю

UR = 1 – Компанія досягла поставленої мети

UR > 1 – Перевиконання плану збору

У проєкті UR обчислюється автоматично на етапі підготовки даних для всіх записів у датасеті. Надалі цей показник використовується в кластерному аналізі, візуалізації та генерації PDF-звітів (Рис. 2.6).

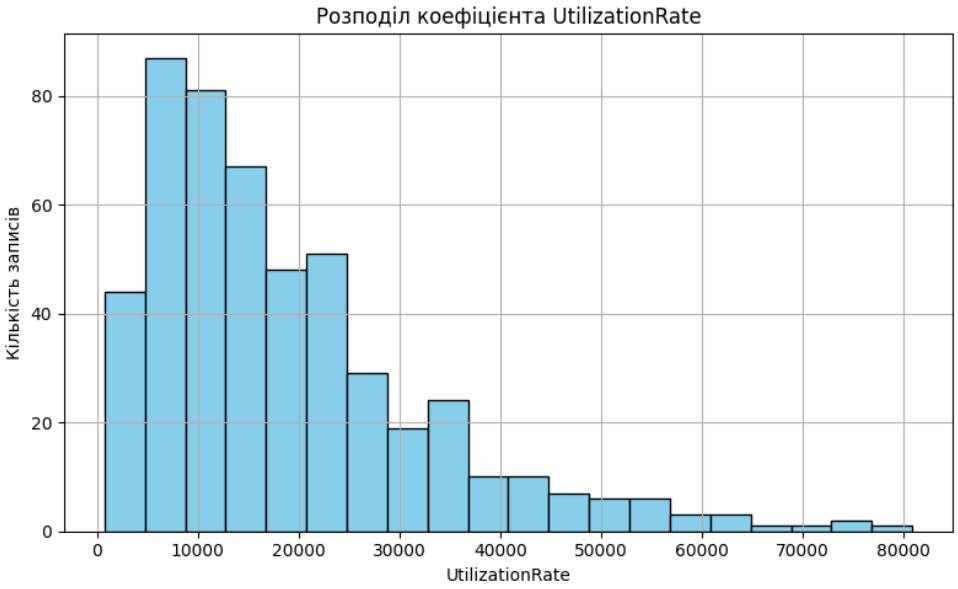

Рисунок 2.6 – Діаграма розподілу коефіцієнта UtilizationRate

Однією з головних переваг Utilization Rate є його універсальність та стандартизованість, що дозволяє застосовувати його для порівняння ініціатив за різних умов реалізації. Він особливо цінний у динамічному середовищі, де важлива швидка реакція на зміну результативності або виявлення відхилень у темпах збору. Його візуалізація — наприклад, через гістограми або boxplot — дозволяє ефективно ідентифікувати як успішні, так і проблемні кампанії.

Проте використання Utilization Rate має і суттєві обмеження. По-перше, показник фокусується виняткова на кількісному вимірі — він не враховує ані якість реалізації допомоги, ані контекст (наприклад, складність логістики чи рівень залучення волонтерів). Відтак, кампанія з високим UR не обов’язково є ефективною у сенсі впливу або досягнення соціального ефекту.

Крім того, показник є чутливим до точності введених даних, зокрема до запланованої цільової суми. Якщо цільова сума визначена умовно або не обґрунтована, обчислений UR може вводити в оману. Ще одне обмеження — нерелевантність показника для нефінансових ініціатив, де об’єктом оцінки є не кошти, а, наприклад, години волонтерської праці, кількість наданих послуг чи обсяг гуманітарної допомоги в натуральному вимірі.

Таким чином, Utilization Rate є корисним інструментом для базової

діагностики ефективності, однак вимагає поєднання з іншими метриками або якісними оцінками для забезпечення повноти аналітичної картини.

2.7 Ієрархічна кластеризація волонтерських кампаній

Ієрархічна кластеризація — це метод групування об'єктів у вигляді дерева кластерів (дендрограми), що ієрархічно об'єднує об'єкти залежно від їх подібності [17] (Рис. 2.7).

Види:

- Агломеративна (знизу вгору) — кожен об'єкт спочатку є окремим кластером; кластери об'єднуються на основі мінімальної відстані.
- Дивізивна (зверху вниз) — усі об'єкти спочатку знаходяться в одному кластері; кластери поступово розділяються.

Основні кроки агломеративної ієрархічної кластеризації:

1. Кожен об'єкт — окремий кластер.
2. Знаходиться пара найближчих кластерів (згідно певної метрики: мінімальна відстань, середня відстань, повна відстань).
3. Об'єднання вибраної пари кластерів в один.
4. Повторення кроків 2–3, поки не залишиться один кластер або не буде досягнуто бажаного рівня кластеризації.

Поширені методи визначення відстані між кластерами:

- Метод одиночного зв'язку (мінімальна відстань)
- Метод повного зв'язку (максимальна відстань)
- Метод середнього зв'язку (середня відстань)

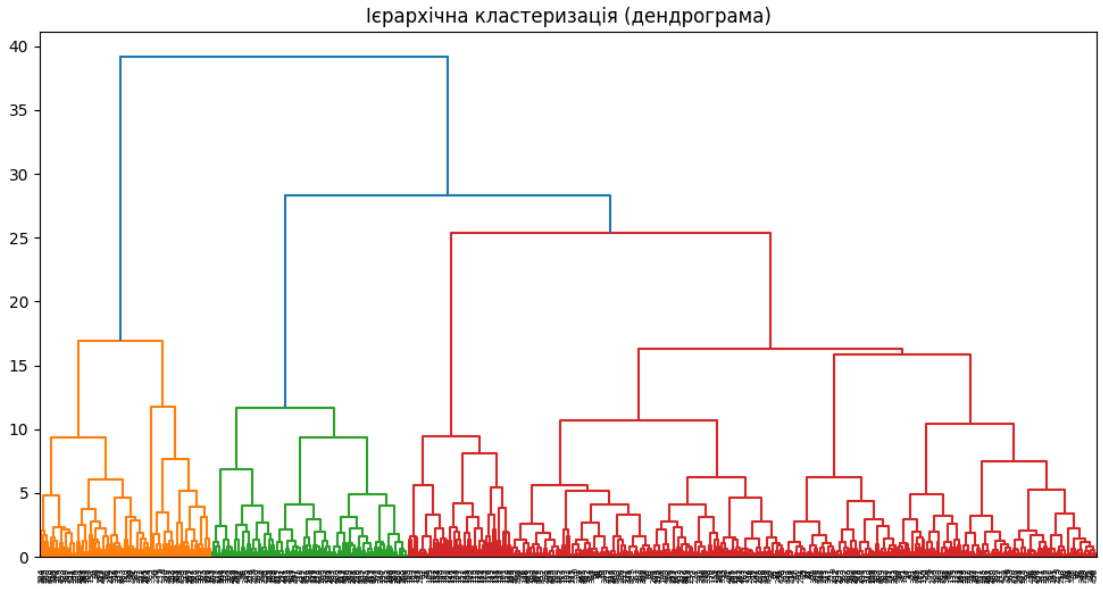

Рисунок 2.7 – Ієрархічна кластеризація волонтерських компаній (дендограма)

Ієрархічна кластеризація є потужним методом аналізу структури даних, що дозволяє дослідити взаємозв'язки між об'єктами через побудову дерева подібності (дендрограми). На відміну від k-середніх, вона не потребує попереднього задання кількості кластерів, що робить її зручною для дослідження нових, непозначених даних, коли кількість груп заздалегідь невідома. Це особливо актуально для волонтерських ініціатив, які можуть значно варіюватися за характером та масштабом.

Однією з ключових переваг є інтерпретованість результатів — дендрограма дозволяє візуально простежити, які об'єкти (наприклад, кампанії) об'єднуються на різних рівнях подібності. Це дає змогу гнучко вибирати рівень деталізації кластерів у залежності від аналітичної задачі. Ієрархічна кластеризація також ефективно працює з невеликими або середніми за обсягом наборами даних, що характерно для пілотних досліджень чи обмежених волонтерських реєстрів.

Проте метод має низку обмежень, які варто враховувати при масштабуванні. Передусім, він є обчислювально затратним, оскільки на кожному кроці необхідно обчислювати відстані між усіма можливими парами кластерів. Це суттєво ускладнює застосування методу до великих масивів даних або до

середовищ із обмеженими обчислювальними ресурсами.

Ще одним критичним моментом є залежність результату від вибору метрики відстані та методу об’єднання кластерів (одиночного, повного, середнього зв’язку тощо). Невдалий вибір параметрів може призвести до формування некоректних груп або спотвореної дендрограми. Крім того, помилки на ранніх етапах об’єднання є незворотними: алгоритм не передбачає переформування кластерів, тому хибне рішення на початку може призвести до некоректної класифікації.

Зважаючи на ці особливості, ієрархічна кластеризація є цінним інструментом для попереднього аналізу або підтвердження результатів інших методів, зокрема у випадках, коли важлива прозорість кластерної структури або існує потреба у глибокому дослідженні подібностей між об’єктами.

2.8 Генерація стандартних PDF-звітів та автоматизація представлення результатів

Генерація звітності є невід’ємною складовою функціоналу аналітичної платформи, оскільки саме звіти забезпечують зв’язок між аналітичним ядром системи та кінцевим користувачем — координатором, донором, керівником організації чи зовнішнім партнером. Автоматизоване створення звітів дозволяє мінімізувати ручну обробку, скоротити час на підготовку презентацій, а також забезпечити прозорість та відтворюваність у роботі з даними.

Основні цілі модуля генерації звітів:

- Узагальнення результатів аналізу у зручній і стандартизованій формі;
- Автоматичне включення числових показників, таблиць, графіків та текстових висновків без втручання користувача;
- Гнучкість вибору змісту звіту — користувач самостійно обирає, які блоки включати залежно від запиту;
- Сумісність зі структурою даних, відображеною у веб-інтерфейсі Streamlit;
- Можливість архівації результатів у форматі PDF — як для внутрішнього аудиту, так і для подальшого аналізу змін.

Звіт формується за допомогою бібліотеки ReportLab, яка дозволяє

створювати структуровані PDF-документи з вбудованими векторними графіками, таблицями та стилізованим текстом. Окрім того, додаткові графіки (наприклад, з matplotlib чи plotly) експортуються як зображення та вставляються у відповідні блоки звіту.

Основні структурні елементи звіту (Рис. 2.8):

1. Титульна сторінка
   Назва документа, дата формування, загальний опис, контактна інформація організації.
2. Огляд ключових метрик
   Таблиця з агрегованими статистиками: кількість кампаній, сума збору, середній донат, медіанне значення тривалості, тощо.
3. Кластерний аналіз
   - Кількість кластерів, опис кожного з них (чисельність, середні значення, min/max);
   - Значення Silhouette Score для кожного кластеру та в цілому;
   - Boxplot-графіки для візуального порівняння кластерів;
   - Висновки щодо ефективності кампаній у межах кластерів.
4. Кореляційний аналіз
   - Матриця кореляцій між числовими змінними у форматі heatmap;
   - Таблиця з топ-5 позитивних і негативних кореляцій;
   - Текстові висновки щодо взаємозв’язків, які мають практичне значення для управління кампаніями.
5. Аномалії
   - Таблиця записів, які були марковані як аномальні за критеріями Z-оцінки, IQR або кластерної відстані;
   - Інтерпретація типів аномалій (наприклад, нетипово високий донат або коротка кампанія з великим UR);
   - Рекомендації щодо подальшої перевірки.
6. Utilization Rate
   - Гістограма розподілу UR;
   - Таблиця кампаній з найбільшими і найменшими значеннями UR;
   - Категоризація UR по типах кампаній або категоріях збору;
   - Оцінка ефективності у порівнянні з середніми значеннями.
7. Додаткові блоки (опційно)
   - Виявлені категоріальні закономірності;
   - Частотний аналіз тем кампаній (наприклад, «дрони», «ліки», «авто»);
   - Дашборд популярних категорій або організаторів;
   - Інтервальні статистики (за місяцями, регіонами тощо — за наявності даних).

Інтерактивність та персоналізація. У веб-інтерфейсі користувач може:

- Вибрати файл з даними;
- Позначити, які методи аналізу виконувати;
- Визначити, які блоки включити у звіт;

Таким чином, модуль генерації звітності виступає не лише як підсумковий інструмент для демонстрації аналітичних результатів, але й як засіб комунікації між учасниками волонтерської інфраструктури, який забезпечує прозорість, довіру та інституційну сталість у прийнятті рішень.

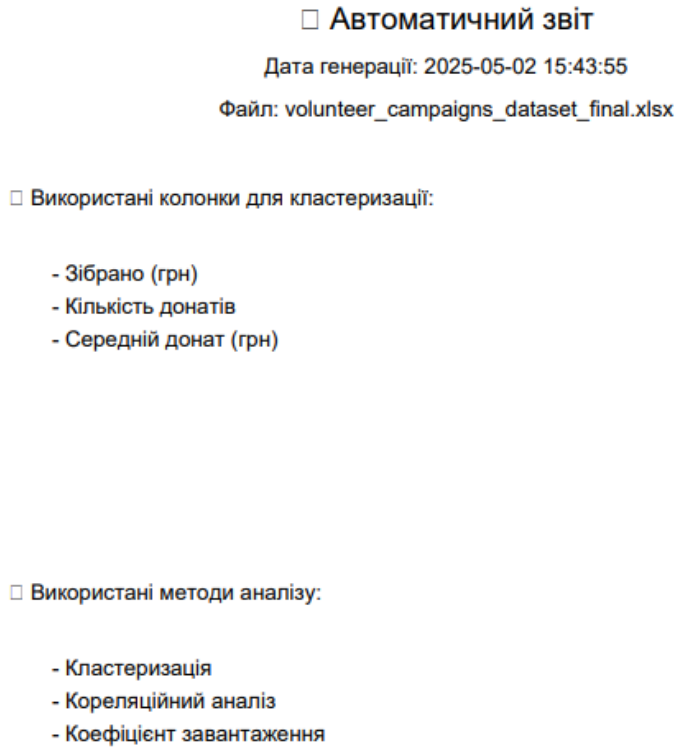

Рисунок 2.8 – Фрагмент автоматично згенерованого PDF-звіту платформи

2.9 Розгортання аналітичної платформи: вибір технологій, архітектура та середовище розгортання

У межах реалізації аналітичної платформи одним із ключових етапів стало визначення оптимальної стратегії її впровадження та інтеграції в реальне інформаційне середовище. Враховуючи специфіку обробки гуманітарних даних, потребу у швидкому доступі до результатів аналізу для різних категорій користувачів, а також вимоги до безпеки, гнучкості й масштабованості системи,

було розроблено кілька варіантів архітектурних рішень щодо розгортання платформи.

Передбачено три основні режими використання системи: локальний, серверний та хмарний. Локальне розгортання орієнтоване на індивідуальну роботу аналітика або невеликої групи користувачів із обмеженим обсягом даних. Такий формат передбачає встановлення програми на персональний комп'ютер або ноутбук без необхідності підключення до серверів чи зовнішніх ресурсів. Локальне розгортання є оптимальним для етапу тестування прототипу, первинного аналізу, навчання користувачів або для використання у малих локальних ініціативах, де не потрібна багатокористувацька взаємодія.

У разі потреби централізованого управління даними й одночасної роботи кількох користувачів застосовується серверне розгортання всередині корпоративної інфраструктури. Такий підхід дозволяє створити єдиний центр обробки даних, забезпечити захищений доступ із локальної мережі організації, організувати розподіл ролей користувачів та контролювати всі етапи роботи з даними. Серверна версія є доцільною для середніх і великих волонтерських об'єднань, благодійних фондів або державних структур, що працюють із великим обсягом запитів і потребують регулярного оновлення баз даних.

Для проєктів, що передбачають високі темпи зростання обсягів даних, географічну розподіленість команд або необхідність гнучкого масштабування ресурсів, найефективнішим рішенням є хмарне розгортання. Платформа може бути інтегрована у хмарні сервіси, такі як Google Cloud Platform, Amazon Web Services (AWS) або Microsoft Azure, що дозволяє забезпечити високу доступність, відмовостійкість, автоматичне масштабування під навантаження та гнучке управління обчислювальними потужностями. Доступ до платформи в цьому випадку здійснюється через захищений вебінтерфейс, незалежно від фізичного місця перебування користувачів.

Окрему увагу в процесі розгортання системи було приділено питанням безпеки. Реалізовано багаторівневу систему захисту, яка включає автентифікацію користувачів, рольову модель доступу до функціоналу та даних,

шифрування збереженої інформації, а також регулярне створення резервних копій для мінімізації ризиків втрати даних. Для підвищення прозорості й аудиту процесів роботи впроваджено механізми журналювання усіх ключових дій користувачів та адміністраторів системи.

Завдяки модульній архітектурі розроблена аналітична платформа легко інтегрується у вже існуючі робочі процеси організацій. Вона підтримує як оперативний аналіз нових кампаній із швидкою підготовкою дашбордів і звітів для управлінських рішень, так і підготовку комплексних звітів за підсумками діяльності, що можуть використовуватись для подання донорам, партнерам або контролюючим організаціям.

Інтуїтивно зрозумілий вебінтерфейс, реалізований на базі Streamlit, забезпечує можливість завантаження даних, налаштування параметрів аналітики, формування запитів на побудову кластеризації чи виявлення аномалій, перегляду інтерактивних графіків та експорту звітів у різних форматах. Особливої уваги приділено зручності інтерфейсу для користувачів без спеціальної технічної освіти, що дозволяє залучити ширший спектр учасників волонтерських проєктів до процесу аналітики.

Остаточний вибір режиму розгортання платформи здійснюється на основі потреб конкретної організації, з урахуванням кількості користувачів, обсягів оброблюваних даних, вимог до безпеки та доступних ресурсів для технічної підтримки. Гнучкість архітектури дозволяє без значних витрат мігрувати платформу між режимами залежно від змін потреб у майбутньому.

Таким чином, запропонована стратегія розгортання платформи забезпечує високу гнучкість, доступність, безпеку та адаптивність системи, що відкриває широкі перспективи її використання у волонтерських та гуманітарних ініціативах різного масштабу і рівня складності.

2.10 Використання експертного методу для оцінки якості та ефективності платформи

З метою перевірки практичної доцільності та ефективності розробленої

аналітичної платформи було застосовано модифікований метод Делфі. Використання цього підходу дозволило залучити експертів для комплексного аналізу функціональності системи та оцінки її відповідності потребам реального волонтерського середовища. Метод Делфі, як відомо, базується на організованому зборі незалежних думок групи експертів із подальшою їх аналітичною обробкою для досягнення максимально об'єктивних висновків. Модифікація методу полягала у поєднанні традиційного експертного оцінювання з елементами практичного тестування функціоналу системи.

Для участі в опитуванні було відібрано п'ять експертів, кожен з яких мав не менше трьох років практичного досвіду роботи у сферах аналітики даних, управління волонтерськими ініціативами або впровадження цифрових сервісів у гуманітарному секторі. Формування експертної групи здійснювалося за критеріями професійної компетентності, релевантного досвіду та практичного розуміння проблематики аналізу даних у сфері волонтерської діяльності.

Експертам було запропоновано ознайомитися із можливостями розробленої аналітичної платформи через наданий вебінтерфейс. Процес оцінювання включав практичне тестування таких етапів роботи системи, як імпорт даних, попередня обробка масивів інформації, вибір змінних для аналізу, виконання кластерного аналізу, виявлення аномалій у даних, формування кореляційних матриць і генерація підсумкових PDF-звітів. Кожен експерт отримав інструкцію щодо базових операцій, які потрібно було виконати для ознайомлення з функціоналом платформи, проте порядок виконання завдань та глибина тестування залишалися на розсуд учасників, що відповідало принципу забезпечення незалежності оцінювання.

Для формалізації результатів експертам було запропоновано оцінити платформу за чотирма ключовими критеріями: зручність інтерфейсу, якість результатів аналізу, швидкість обробки даних та загальна ефективність платформи. Оцінювання здійснювалося за шкалою від 1 до 10 балів, де 1 бал означав найнижчий рівень задоволеності, а 10 балів — максимально можливу оцінку. Такий підхід дозволив отримати кількісні показники оцінювання для

подальшого статистичного аналізу.

Крім кількісного оцінювання, кожному експерту було запропоновано надати відкриті коментарі щодо сильних і слабких сторін платформи, а також порівняти її функціональні можливості з іншими відомими рішеннями, що використовуються у суміжних напрямах, а саме: KoboToolbox, Salesforce for Nonprofits та DHIS2. Вибір цих платформ для порівняння був зумовлений їх популярністю у сфері гуманітарної аналітики, однак усі експерти мали право надавати власні приклади альтернативних інструментів для розширення контексту аналізу.

Особливу увагу під час організації оцінювання було приділено забезпеченню незалежності думок експертів. Кожен учасник працював індивідуально без взаємодії з іншими експертами, що дозволяло уникнути групового тиску або упередженості в оцінках. Зібрані дані були оброблені із застосуванням методів описової статистики: обчислення середніх оцінок за кожним критерієм та визначення стандартного відхилення для оцінки стабільності суджень.

Проведене експертне оцінювання мало на меті не тільки кількісне ранжування характеристик платформи, але й виявлення якісних аспектів її використання, сильних сторін і можливих напрямків для подальшого вдосконалення. Усі зібрані кількісні та якісні дані були враховані при подальшому аналізі результатів апробації системи.

2.11 Висновки до розділу 2

У другому розділі роботи було комплексно розглянуто методологічні та технологічні основи побудови інтегрованої аналітичної платформи для аналізу даних волонтерських ініціатив і автоматизації звітності.

На основі аналізу вимог до волонтерської аналітики сформульовано загальну методику побудови платформи, що базується на модульному принципі архітектури. Система передбачає окремі компоненти для завантаження,

попередньої обробки, аналізу, візуалізації даних і автоматичного формування звітів, що забезпечує її гнучкість, масштабованість і зручність адаптації під різні умови використання. Для реалізації платформи обрано мову програмування Python, що дозволило ефективно інтегрувати найсучасніші бібліотеки для обробки, аналізу та візуалізації даних, зокрема Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, Plotly, Streamlit, ReportLab.

Особливу увагу приділено питанням підготовки даних — як ключовому етапу забезпечення якості аналітики. Детально описано процеси імпорту, очистки, нормалізації, роботи з категоріальними змінними, створення нових похідних змінних та стандартизації масштабів. Це дозволяє формувати стабільні набори даних для кластеризації, кореляційного аналізу й пошуку аномалій.

У межах платформи реалізовано кластерний аналіз за допомогою методу k-середніх, що дозволяє ідентифікувати типові профілі волонтерських кампаній за обраними характеристиками. Для визначення оптимальної кількості кластерів застосовано метод ліктя. Додатково розглянуто ієрархічну кластеризацію, яка дає змогу будувати дендрограми та досліджувати структуру даних без потреби попереднього задання кількості груп.

Кореляційний аналіз між основними числовими показниками дозволяє виявляти залежності між кількістю донатів, сумами збору, тривалістю кампаній та іншими параметрами, що відкриває можливості для глибшого розуміння закономірностей у волонтерських ініціативах.

Особливий акцент зроблено на методах виявлення аномалій у даних. Застосовано як класичні статистичні підходи (Z-оцінка, інтерквартильний розмах), так і кластерні методи ідентифікації відхилень, що дозволяє виявляти як помилки, так і унікальні або критичні випадки в даних.

Важливим доповненням стало впровадження обчислення коефіцієнта корисного завантаження (Utilization Rate), який слугує індикатором ефективності досягнення фінансових цілей кампаній і дозволяє проводити базову сегментацію проектів за рівнем результативності.

Створено механізм автоматичної генерації аналітичних звітів у форматі

PDF, що забезпечує стандартизовану подачу результатів аналізу з можливістю вибору блоків для включення у звіт відповідно до потреб користувача. Така автоматизація значно спрощує комунікацію результатів між різними учасниками волонтерських процесів.

Окремо проаналізовано варіанти розгортання платформи. Запропоновано три режими використання — локальний, серверний і хмарний — що дозволяють адаптувати систему до різних масштабів проєктів і технічних можливостей організацій. Описано заходи безпеки, реалізовані в архітектурі рішення, зокрема багаторівневу аутентифікацію, шифрування даних, аудит дій користувачів і створення резервних копій.

На завершення, для оцінки ефективності запропонованої платформи було застосовано модифікований метод Делфі. Організовано експертне оцінювання з залученням фахівців у сферах аналітики даних та волонтерського менеджменту. Експерти ознайомилися з функціоналом платформи, протестували основні можливості аналізу й звітності та надали об'єктивні оцінки за встановленими критеріями. Проведення такого дослідження дозволило комплексно обґрунтувати доцільність і практичну ефективність розробленого рішення.

Таким чином, Розділ 2 закладає методологічну та технологічну основу для реалізації аналітичної платформи, демонструючи цілісність підходу, обґрунтованість вибору методів та інструментів і готовність рішення до практичного впровадження у волонтерській сфері.

РОЗДІЛ 3
МОДЕЛЮВАННЯ ПРОЦЕСІВ АНАЛІЗУ ДАНИХ У ВОЛОНТЕРСЬКИХ КАМПАНІЯХ

3.1 Архітектура інтегрованої аналітичної платформи та модульна структура системи

Аналітична платформа для обробки даних волонтерських кампаній реалізована з використанням мови програмування Python та низки спеціалізованих бібліотек, що дозволяють виконувати повний цикл аналізу: від завантаження й очищення даних до кластеризації, виявлення аномалій, кореляційного аналізу, розрахунку коефіцієнта ефективності (UtilizationRate) та генерації звітів.

Основна архітектура побудована модульно, що забезпечує масштабованість, простоту обслуговування та можливість інтеграції нових методів без зміни базової логіки.

Таблиця 3.1 – Опис модулів аналітичної платформи

Модуль	**Призначення**
data_loader.py	Завантаження даних з файлів Excel, перевірка структури, типів та очищення.
preprocess_numeric_data.py	Нормалізація числових змінних для коректної роботи алгоритмів.
categorical_processing.py	Групування та обробка категоріальних змінних.
clustering.py	Кластеризація методом k-середніх, з оцінкою якості кластерів.
hierarchical_clustering.py	Ієрархічна кластеризація для візуального групування ініціатив.
correlation_analysis.py	Побудова кореляційної матриці, визначення значущих зв’язків.
anomaly_detection.py	Виявлення аномалій методами Z-score та IQR.
utilization_calculator.py	Обчислення UtilizationRate та категоризація результатів.
report_generator.py	Формування PDF-звітів з графіками, таблицями та висновками.

Модуль	Призначення
app.py	Веб-інтерфейс платформи (на базі Streamlit).
validate_methods.py	Валідація правильності роботи методів.
comparison.py	Порівняння звітів або результатів аналізу.

У рамках побудови аналітичної платформи було реалізовано чітко структурований потік обробки даних, який охоплює всі ключові етапи – від завантаження та попередньої підготовки інформації до застосування аналітичних методів і генерації підсумкової звітності. Центральним елементом цієї архітектури є її модульна організація, що дозволяє підтримувати гнучкість і масштабованість системи, спрощуючи супровід і подальший розвиток.

Процес починається з взаємодії користувача з веб-інтерфейсом, реалізованим у середовищі Streamlit (app.py). Тут користувач має змогу завантажити Excel-файл із даними волонтерських кампаній, який надходить на вхід основному модулю завантаження — data_loader.py. Цей компонент відповідає за базову перевірку: очищення назв колонок від зайвих пробілів, приведення форматів, виявлення відсутніх або неправильно названих змінних. Також здійснюється первинний аналіз типів даних — для того, аби визначити, які з них можуть бути використані у наступних аналітичних операціях.

Після успішного завантаження таблиці система переходить до обробки категоріальних змінних. У багатьох випадках значення таких змінних є надто деталізованими або дублюються через відмінності в написанні (наприклад, "Збір на ЗСУ" і "Збір для ЗСУ"). Модуль categorical_processing.py виконує згортання цих значень у єдині категорії, що значно підвищує якість подальшого аналізу — як у кластеризації, так і при візуалізації або фільтрації.

Наступним кроком є підготовка числових змінних, які використовуються алгоритмами, чутливими до масштабів даних — зокрема, методом k-середніх. Через модуль preprocess_numeric_data.py відбувається нормалізація числових полів, що дозволяє уникнути зміщення результатів на користь змінних із більшими абсолютними значеннями. Це забезпечує рівномірний вплив усіх

показників на формування кластерів та інтерпретацію результатів.

Після попередньої підготовки даних платформа автоматично виконує основні аналітичні процедури. Кластеризація (clustering.py) дозволяє згрупувати кампанії за подібністю ключових характеристик, кореляційний аналіз (correlation_analysis.py) дає змогу виявити важливі залежності між параметрами, виявлення аномалій (anomaly_detection.py) фіксує нетипові записи, які можуть бути як сигналами помилки, так і маркерами надзвичайної ефективності. Паралельно обчислюється показник ефективності — UtilizationRate (utilization_calculator.py), який демонструє продуктивність кампанії у перерахунку на одиницю часу.

Усі результати об’єднуються у фінальний аналітичний звіт, який формується за допомогою report_generator.py. Звіт створюється у форматі PDF і включає таблиці, графіки, блоки з ключовими висновками та рекомендаціями (Рис. 3.1). Це дозволяє координаторам, аналітикам або донорам отримати вичерпне уявлення про стан волонтерських активностей без потреби в додатковій обробці даних вручну.

Рисунок 3.1 – Основні етапи реалізації аналітичної платформи

3.2 Реалізація модуля попередньої обробки та підготовки даних

Етап попередньої обробки даних є фундаментальним для функціонування аналітичної платформи, адже саме він забезпечує перетворення сирих, часто неоднорідних чи неповних даних у формат, придатний для подальшої аналітики. У реалізованій системі обробка реалізована як окремий модуль data_loader.py,

який відповідає не лише за зчитування даних, але й за базову валідацію та приведення структури до уніфікованого вигляду (Рис. 3.2).

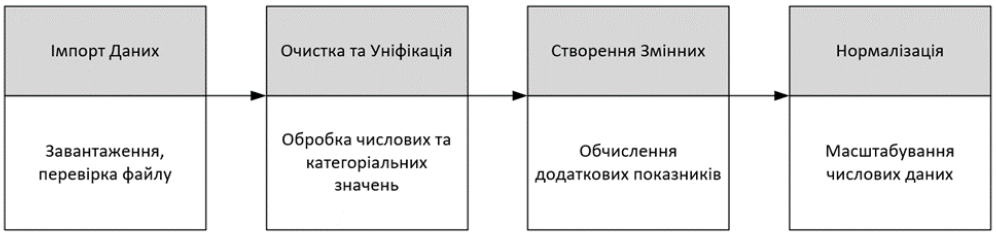

Рисунок 3.2 – Схема попередньої обробки даних

Оскільки волонтерські кампанії представлені широким спектром параметрів — від числових метрик (сум збору, кількість донатів) до категоріальних значень (тип кампанії, напрямок допомоги), ключовим завданням цього етапу було:

- виявити відсутні або некоректні значення;
- уніфікувати назви та типи змінних;
- створити додаткові похідні змінні для подальшої аналітики;
- попередньо структурувати категоріальні змінні за групами.

Для забезпечення гнучкості модуль підтримує завантаження Excel-файлів, які є найпоширенішим форматом звітності у гуманітарному середовищі. На цьому етапі також перевіряється наявність усіх необхідних колонок. У разі відсутності критичних полів платформа повідомляє користувача про помилку.

Великою проблемою в гуманітарних даних є неузгодженість назв і типів. Наприклад, поле "Сума збору" часто вводиться вручну, з пробілами або символами валюти, що заважає перетворенню в числовий тип. Ці виклики вирішуються за допомогою модулів preprocess_numeric_data.py та categorical_processing.py, які:

- автоматично конвертують числові значення, навіть якщо вони подані як текст;
- видаляють зайві символи (₴, пробіли, крапки замість коми);
- перетворюють категоріальні значення до уніфікованих груп.

Для реалізації ключових метрик платформи вже на етапі завантаження обчислюються похідні змінні:

- Середній донат — розраховується як відношення зібраної суми до кількості донатів;
- Тривалість кампанії — визначається на основі різниці між датою початку та завершення;
- UtilizationRate — показник ефективності, що відображає інтенсивність збору.

Це не лише полегшує подальший аналіз, але й дозволяє відразу відфільтровувати помилкові або неповні записи, де, наприклад, донати зафіксовано, але відсутня сума збору.

Особливої уваги заслуговує механізм групування категоріальних змінних — регіонів, тематик, типів допомоги. У нашому рішенні реалізовано фільтрацію й групування за заздалегідь визначеним списком, що дозволяє уникати проблем дублювання або некоректного агрегаційного аналізу. Наприклад, категорії «дрон», «Дрони», «БПЛА» об’єднуються в одну групу — «Повітряна підтримка».

Для алгоритмів, що використовують обчислення відстані (наприклад, k-середніх), критично важливо нормалізувати значення. Ця операція виконується в модулі preprocess_numeric_data.py, де до вибраних числових колонок застосовується StandardScaler.

3.3 Виконання кластеризації кампаній: методика та підбір параметрів

Одним із ключових аналітичних інструментів у побудованій платформі є кластеризація — метод групування схожих за параметрами волонтерських кампаній без використання заздалегідь заданих міток. Такий підхід дає змогу виокремити типові профілі кампаній, виявити крайні випадки або

закономірності, які важко помітити при стандартному аналізі. У нашому рішенні реалізовано кластеризацію на основі алгоритму k-середніх (k-means), як одного з найбільш інтерпретованих і придатних для числових даних методів.

Проблема визначення оптимальної кількості кластерів `k` вирішується за допомогою двоетапного підходу:

1. Автоматичний вибір кількості кластерів ґрунтується на застосуванні методу ліктя та обчисленні Silhouette Score для кожного `k` у заданому діапазоні (зазвичай 2–10). Метод ліктя дозволяє візуально визначити точку, де зменшення інерції (WCSS) починає уповільнюватися, що сигналізує про оптимальну кількість кластерів. Silhouette Score доповнює цей аналіз, кількісно оцінюючи відокремленість кожного кластера: чим ближче значення до 1 — тим чіткіше відокремлені групи (Рис. 3.3.1, 3.3.2).
2. Ручне задання `k` реалізовано у вигляді параметру, доступного користувачеві у Streamlit-інтерфейсі. Це дозволяє фахівцям, які мають предметну експертизу, здійснювати цілеспрямоване групування, наприклад, за кількістю напрямків чи пріоритетів у волонтерських кампаніях (Рис. 3.3.3).

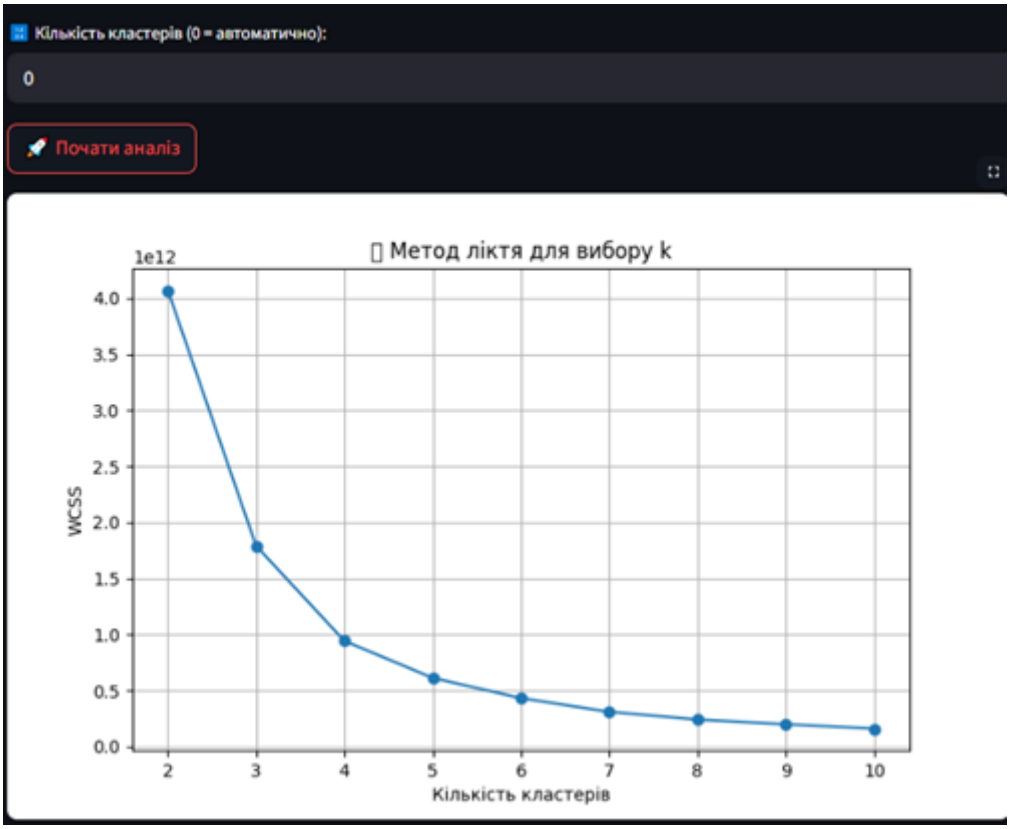

Рисунок 3.3.1 – Графік методі ліктя (WCSS)

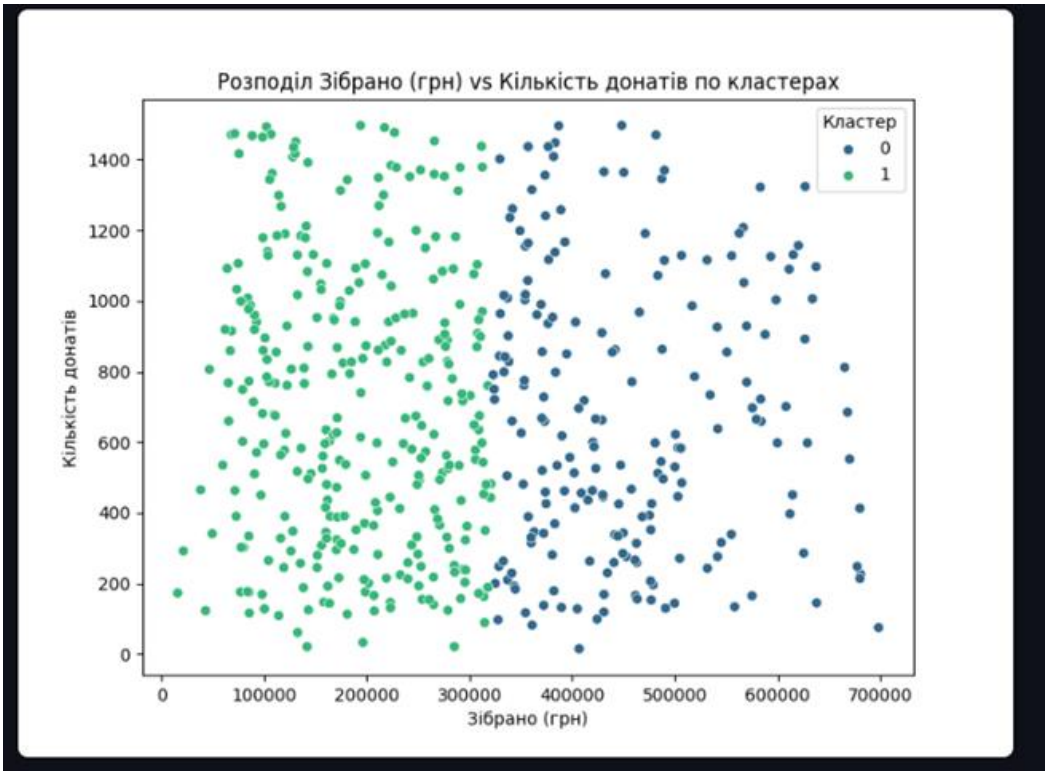

Рисунок 3.3.2 – Розподіл точок за автоматичним вибором кількості кластерів

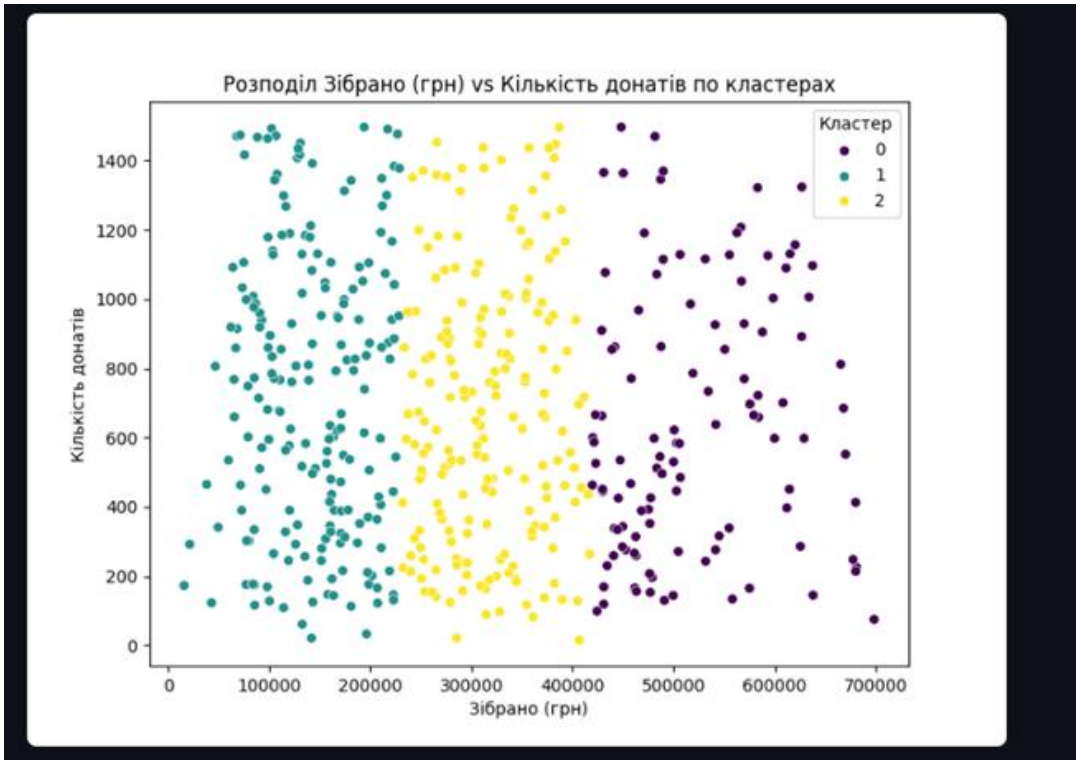

Рисунок 3.3.3 – Розподіл точок за ручним заданням k = 3

Модуль clustering.py містить основну реалізацію функції run_kmeans_clustering, яка виконує наступні дії:

- приймає на вхід попередньо нормалізовані числові змінні;
- ініціалізує об’єкт KMeans із фіксованим random_state для забезпечення відтворюваності;
- навчає модель та обчислює мітки кластерів;
- додає новий стовпець cluster у DataFrame для подальшої обробки.

Кластеризація виконується за метриками, обраними користувачем — зазвичай це сума збору, кількість донатів, тривалість кампанії, UtilizationRate. Після кластеризації платформа автоматично зберігає отримані мітки, а також статистику по кожному кластеру, що дає змогу проводити поглиблену інтерпретацію.

Після кластеризації автоматично генерується кілька аналітичних графіків:

- Scatter plot (двовимірний розподіл точок), що відображає поділ об’єктів за ключовими змінними;
- Boxplot-графіки, які дозволяють візуально оцінити розбіжності між кластерами (Рис. 3.3.4, 3.3.5);
- Таблиця-узагальнення, яка включає (Рис. 3.3.6):
    - кількість кампаній у кожному кластері;
    - середні та медіанні значення по змінних;
    - мінімальні/максимальні значення;
    - Silhouette Score (за наявності).

Ці візуалізації створюються за допомогою модуля plot_generator.py і передаються у модуль генерації звітів та інтерфейс користувача.

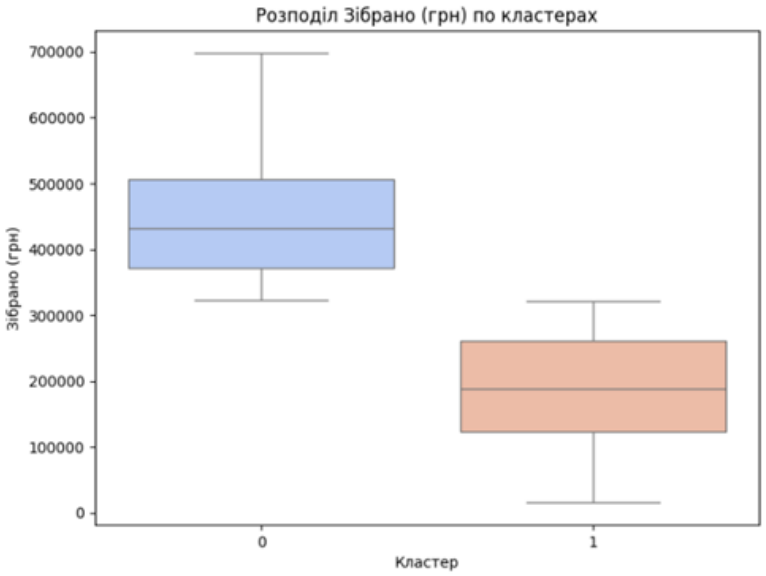

Рисунок 3.3.4 – Графік розподілу по кластерах (Зібрано (грн))

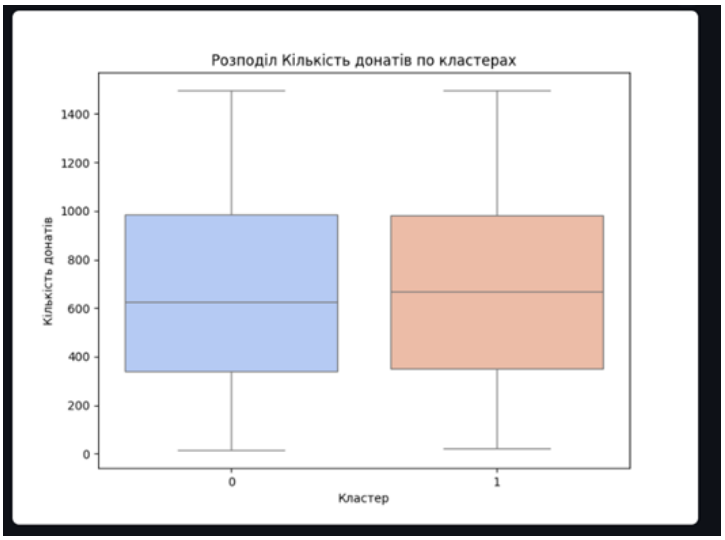

Рисунок 3.3.5 – Графік по кластерах (Кількість донатів)

Рис. 3.3.6 – Приклад таблиці статистики по кластеру

Результати кластеризації експортуються у вигляді JSON-структури, яка включає зведену інформацію по кожному кластеру та використовується для побудови дашбордів, формування PDF-звіту або подальшої аналітики (наприклад, перевірки кореляцій між кластером і зовнішніми факторами).

Цей блок роботи — один із найбільш ресурсоємних з аналітичної точки зору, однак саме кластеризація дозволяє перейти від хаотичних вибірок до структурованих висновків, які можуть використовуватись у стратегічному плануванні, оптимізації зборів та пріоритезації ресурсів.

3.4 Ієрархічна кластеризація

Окрім традиційного методу кластеризації k-середніх, який передбачає чітке задання кількості кластерів, у межах розробленої платформи реалізовано також ієрархічну кластеризацію як додатковий аналітичний інструмент для глибшого розуміння схожості між волонтерськими кампаніями.

Ієрархічна кластеризація не потребує заздалегідь встановленої кількості кластерів. Замість цього вона формує дерево зв'язків — дендрограму, яка наочно демонструє, як об'єкти поступово об'єднуються у групи. Це надає змогу досліднику самостійно визначати рівень агрегації — від дрібнозернистої структури до великих, узагальнених кластерів.

У нашій реалізації цей підхід виконано в модулі hierarchical_clustering.py. Для побудови ієрархічної моделі використовуються стандартні методи агломеративної кластеризації на основі обчислення евклідової відстані між об'єктами та використання одного з методів злиття (linkage) — зокрема, Ward або average. Результатом виконання є дендрограма, що демонструє, які кампанії мають найвищу схожість за обраними параметрами, такими як сума збору, кількість донатів, тривалість, тощо (Рис. 3.4).

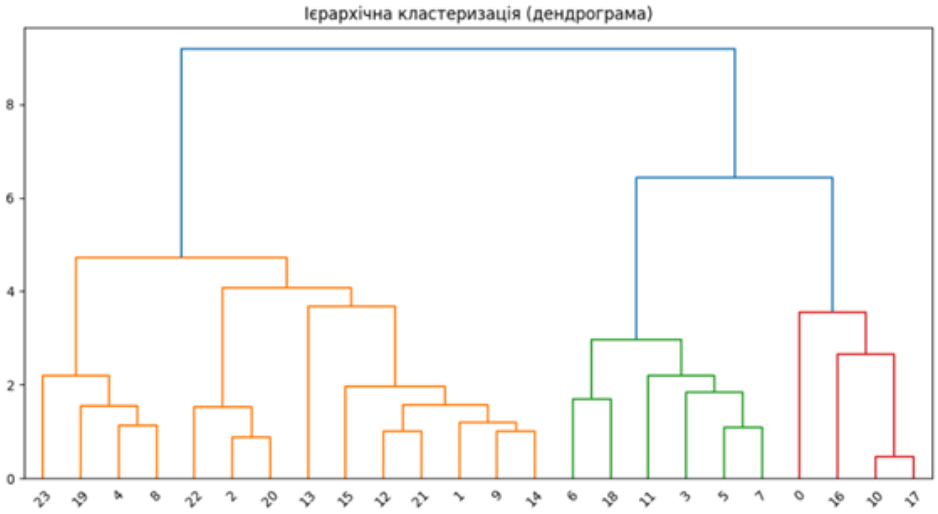

Рисунок 3.4 – Приклад дендрограма з аналітичної платформи

Підхід ієрархічної кластеризації був включений до функціоналу аналітичної платформи як доповнення до основного методу k-середніх. Його

доцільність зумовлена потребою в гнучкішому аналізі, що дозволяє не просто фіксувати групи на основі заданої кількості кластерів, а динамічно досліджувати структуру схожості між волонтерськими кампаніями на різних рівнях деталізації.

Особливістю ієрархічної кластеризації є формування дерева об'єднань, або дендрограми. Кожен вузол цього дерева демонструє рівень подібності між об’єктами, а також послідовність, у якій вони групуються. Це дає змогу не лише виявити близькі за характеристиками кампанії, але й візуально відслідковувати, на якому рівні кластеризації з’являються нові об’єднання, що особливо корисно при аналізі великих обсягів гуманітарних ініціатив зі схожими параметрами, але різною масштабністю чи тривалістю.

3.5 Оцінка кореляційних залежностей між змінними: побудова heatmap і виявлення найсильніших зв’язків

У межах реалізованої аналітичної платформи кореляційний аналіз є ключовим інструментом для виявлення статистичних зв’язків між числовими змінними, що характеризують волонтерські кампанії. Його основна мета — не лише виявити, які показники впливають один на одного, а й надати інтерпретовану базу для прийняття рішень, зокрема у плануванні ресурсів, аналізі пріоритетів, виявленні факторів ефективності.

На етапі інтеграції цього інструменту було розроблено окремий модуль correlation_analysis.py, який відповідає за:

- побудову кореляційної матриці;
- візуалізацію результатів у вигляді теплової карти (heatmap);
- відбір топ-5 позитивних та негативних пар змінних з найвищою абсолютною силою кореляції.

Кореляційна матриця є візуальним представленням зв’язків, що дозволяє швидко оцінити взаємозалежності між змінними. Зображення формується за допомогою бібліотеки Seaborn, де значення кореляційного коефіцієнта

відображаються кольором — від темно-синього (сильна негативна кореляція) до темно-червоного (сильна позитивна). Для більшої інформативності числові значення відображаються безпосередньо на матриці.

Для практичного застосування цього аналізу система автоматично обчислює коефіцієнти Пірсона, які оцінюють лінійний зв’язок між змінними. Наприклад, сильна позитивна кореляція між “Сумою збору” та “Кількістю донатів” може свідчити про важливість охоплення, тоді як негативна залежність між “Середнім донатом” і “Тривалістю кампанії” може сигналізувати про виснаження потенціалу донорів з часом.

Окремий функціонал платформи дозволяє інтерактивно відобразити:

- Heatmap (Рис. 3.5.1) — загальна кореляційна матриця для всіх числових змінних;
- Таблицю зв’язків (Рис. 3.5.2, 3.5.3) — перелік змінних із найвищою/найнижчою кореляцією (абсолютне значення r > 0.6);
- Список значущих кореляцій (Рис. 3.5.4) — лише ті зв’язки, що перевищують заданий поріг значущості, наприклад r > ±0.7.

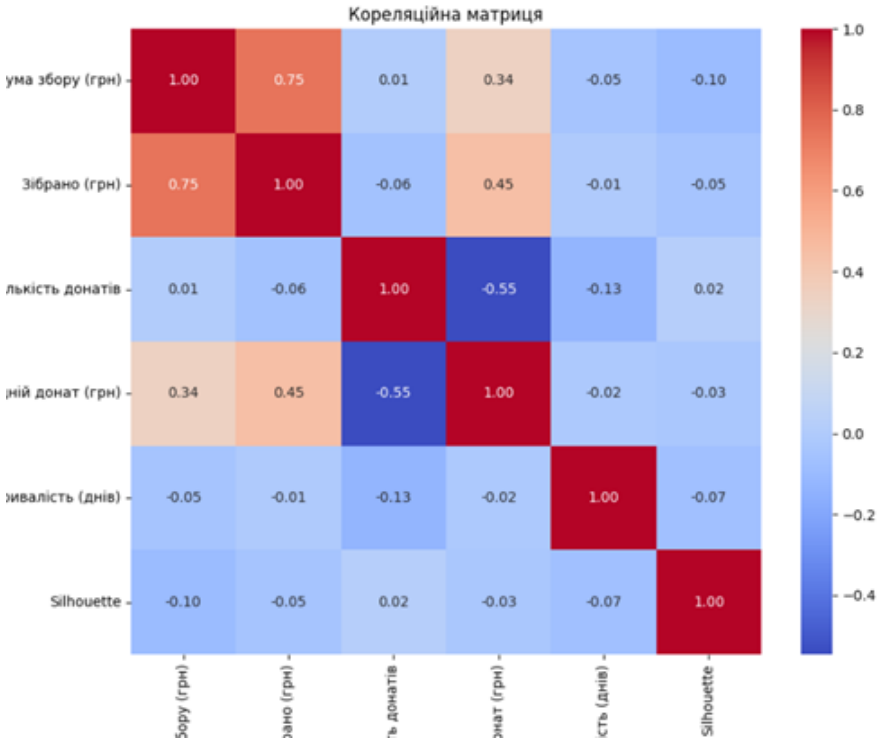

Рисунок 3.5.1 – Кореляційна матриця

Рисунок 3.5.2 – Приклад позитивних кореляцій

Рисунок 3.5.3 – Приклад негативних кореляцій

Рисунок 3.5.4 – Приклад значущих кореляцій

Ці результати інтегруються до автоматизованого звіту та використовуються у подальших модулях для фільтрації змінних, пояснення аномалій або уточнення моделей кластеризації.

Завдяки кореляційному аналізу користувач платформи може не лише отримати формальні статистичні показники, але й глибше зрозуміти механіку функціонування волонтерських кампаній: які показники взаємопов’язані, де спостерігаються закономірності, а де — конфлікти у даних. Це відкриває шлях до більш ефективного планування, таргетування допомоги та прозорого оцінювання результатів.

3.6 Виявлення аномалій у ключових показниках кампаній

Виявлення аномалій — це важливий етап у побудові надійної аналітичної системи, оскільки він дозволяє виявити нетипові або підозрілі дані, які можуть впливати на точність подальшого аналізу. У гуманітарній сфері такі аномалії можуть виникати як через технічні похибки введення (наприклад, додавання зайвих нулів), так і через дійсно виняткові випадки — успішні кампанії з надвисоким залученням ресурсів або, навпаки, повністю провальні ініціативи.

У межах реалізованої платформи виявлення аномалій реалізовано як окремий функціональний модуль anomaly_detection.py, який дозволяє:

- використовувати класичні статистичні методи для виявлення відхилень — Z-оцінку та метод міжквартильного розмаху (IQR);
- автоматично маркувати записи як аномальні на основі встановлених порогів;
- зберігати результати у вигляді таблиці з поясненням причин маркування;
- інтегрувати візуальні графіки для зручності аналізу.

Z-оцінка — розраховується як відстань значення від середнього у стандартних відхиленнях. У нашій реалізації значення вважається аномальним, якщо $|Z| > 3$. Цей підхід особливо ефективний для нормально розподілених змінних, таких як Середній донат, Silhouette Score.

IQR (Interquartile Range) — метод базується на побудові квартилів: якщо значення виходить за межі, воно вважається аномальним. Цей підхід є стійким до шуму та ефективним для нерівномірних розподілів (наприклад, Тривалість, UtilizationRate)

У результаті виконання аналізу формується таблиця з ідентифікаторами записів (Рис. 3.6.1, 3.6.2), де зазначено:

- назву змінної, за якою виявлено відхилення;
- тип застосованого методу (Z або IQR);
- фактичне значення та межі, що були використані.

Рисунок 3.6.1 – Приклад вигляду без виявлений аномалій

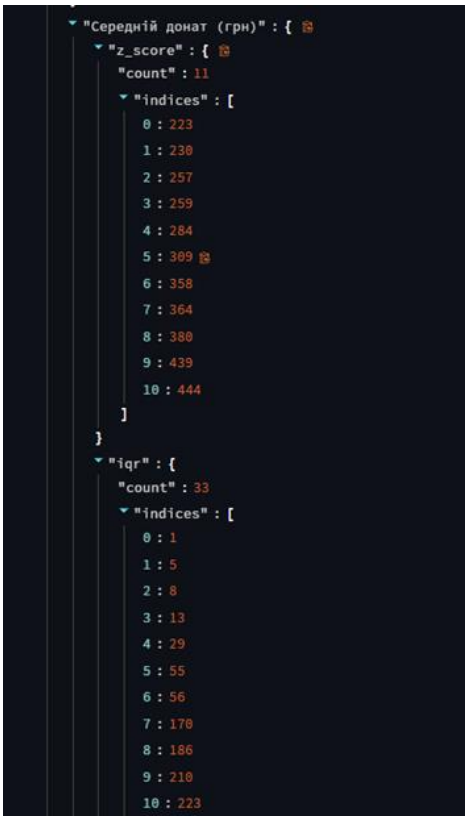

Рисунок 3.6.2 – Приклад вигляду виявлених аномалій

Крім табличної інформації, система будує графіки розподілу (Рис. 3.6.3, 3.6.4) (boxplot, histogram), на яких аномальні значення виділені окремим кольором. Це дозволяє аналітику швидко виявити не лише наявність аномалій, але й оцінити їхню щільність та характер розподілу.

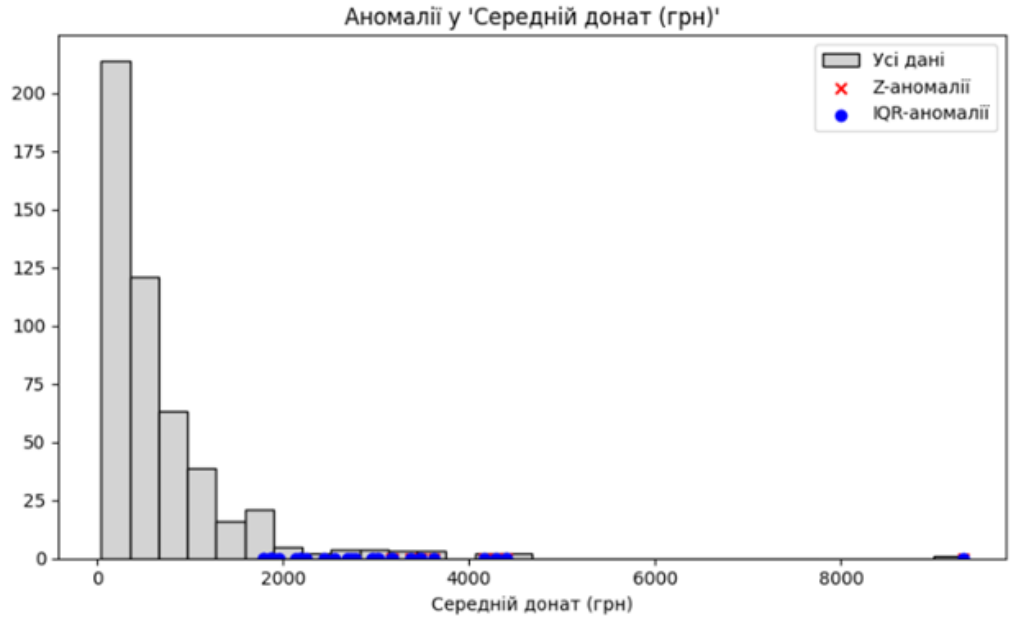

Рисунок 3.6.3 – «Середній донат» із позначенням аномалій

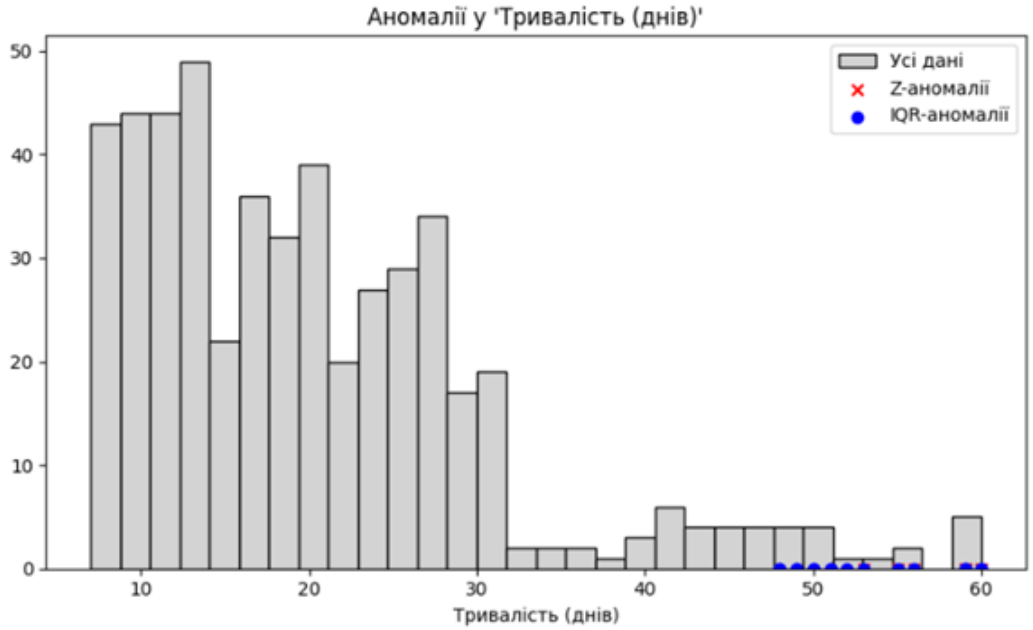

Рисунок 3.6.4 – «Тривалість кампаній» з маркером поза межами IQR

Аномалії не видаляються з даних автоматично — користувач самостійно

приймає рішення, як із ними поводитися: виключити з аналізу, позначити як винятки, або розглянути як об’єкти глибшого аналізу (наприклад, потенційні флагмани або ризиковані кейси).

Таким чином, модуль виявлення аномалій виконує не лише функцію очищення даних, а й працює як механізм стратегічної діагностики — ідентифікуючи кампанії, що заслуговують на особливу увагу через свою нетиповість або ризиковість.

3.7 Розрахунок UtilizationRate та побудова категорій кампаній за ефективністю

Оцінка ефективності волонтерських кампаній — один із ключових етапів аналізу, що дозволяє не лише описати минулі результати, а й спрогнозувати майбутню доцільність подібних ініціатив. У межах розробленої аналітичної платформи цю функцію реалізовано через розрахунок показника UtilizationRate (UR) — коефіцієнта корисного завантаження.

Це дозволяє отримати стандартизований показник інтенсивності залучення ресурсів: скільки коштів щодня збирала кампанія. Такий підхід особливо корисний у випадках, коли тривалість ініціатив сильно варіюється, а суми збору не дають повної картини про темпи ефективності.

Розрахунок UR здійснюється у модулі utilization_calculator.py, де відразу після попередньої обробки даних для кожного запису додається нова змінна UtilizationRate. Окрему увагу приділено валідації: якщо значення тривалості кампанії дорівнює нулю або відсутнє — запис ігнорується або виводиться повідомлення для користувача.

Важливо, що формула UR враховує тільки завершені або активні кампанії з обома датами — початку та завершення. Це дозволяє уникнути викривлення аналітики через неповні або незавершені проєкти.

У межах вебінтерфейсу платформи реалізовано побудову гістограми розподілу UR (рис. 3.7.1), яка дозволяє побачити загальну картину ефективності зборів. Додатково, кампанії автоматично категоризуються за трьома рівнями:

- Низька ефективність: UR < 1 000
- Середня ефективність: 1 000 ≤ UR < 10 000
- Висока ефективність: UR ≥ 10 000

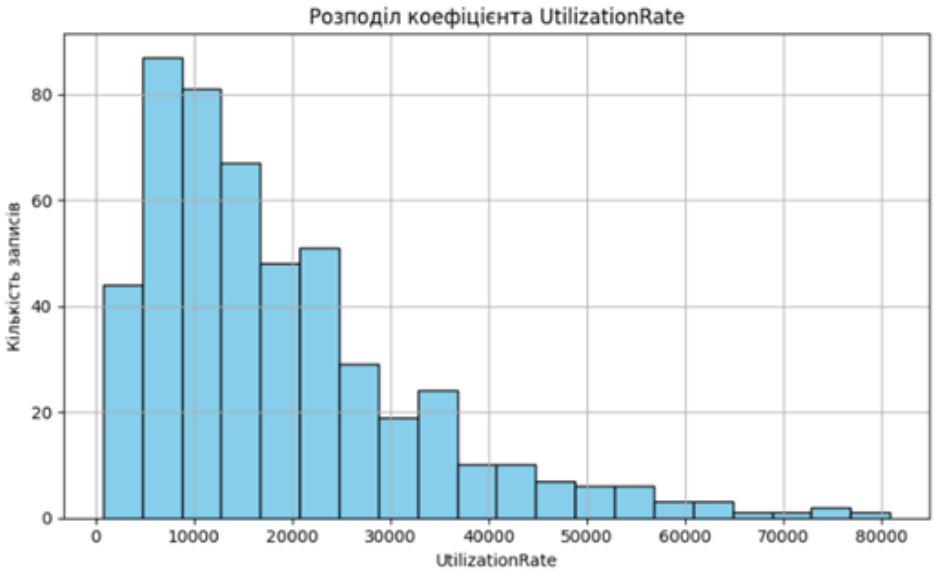

Рисунок 3.7.1 – Розподіл показника UtilizationRate серед усіх кампаній

Це групування не лише спрощує порівняння між кампаніями, а й дозволяє будувати агреговану статистику по категоріях, (Рис. 3.7.2) яку надалі можна використовувати для звітів, порівнянь між регіонами або напрямами допомоги.

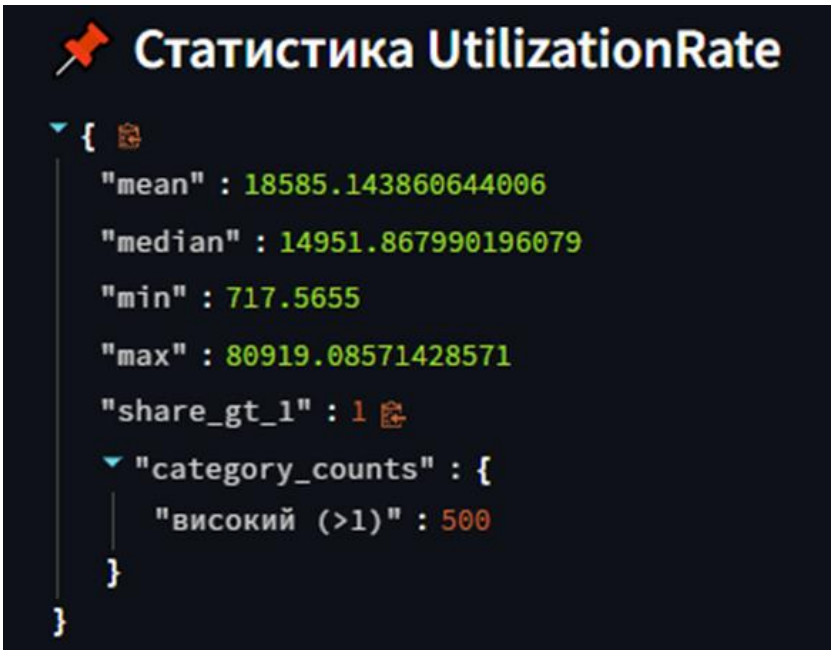

Рисунок 3.7.2 – Приклад статистики UtilizationRate

У порівнянні зі звичайними метриками на кшталт "сума збору", показник UR дає змогу бачити динаміку результату в розрізі часу, а отже — виявляти ефективні механізми залучення коштів або навпаки — кампанії, що затягуються без видимого прогресу.

У звітах платформи показник UR використовується не лише як числова характеристика, а й як інструмент ранжування, порівняння та групування ініціатив. Він включений до підсумкової таблиці та виводиться у форматі PDF-аналітики для донорів і координаторів.

3.8 Візуалізація та вивід результатів

Розробка інструментів візуального представлення даних є критичним аспектом роботи аналітичної платформи, адже саме графіки, діаграми та схеми дозволяють швидко інтерпретувати великі обсяги числової інформації, виявити закономірності, аномалії, кластери та інші патерни поведінки волонтерських кампаній. У нашій системі весь процес візуалізації автоматизований і реалізований через окремий модуль — plot_generator.py, що забезпечує гнучке та масштабоване створення графіків.

Для ефективного представлення результатів, аналітична платформа використовує широкий спектр методів візуалізації, які дозволяють не лише узагальнити інформацію, але й виявити приховані закономірності, аномалії або типові профілі поведінки волонтерських кампаній. Кожен тип графіку виконує специфічну функцію та застосовується залежно від поставленої задачі.

Гістограми (Histogram) є базовим інструментом для аналізу розподілу числових показників, таких як UtilizationRate, кількість донатів або тривалість кампаній. Вони дають змогу оцінити, у яких діапазонах зосереджено найбільше значень, що дозволяє сформувати уявлення про «типову» кампанію. Наприклад, гістограма коефіцієнта корисного завантаження (UR) дозволяє поділити всі кампанії на низько-, середньо- та високоефективні.

Boxplot-графіки (ящики з вусами) застосовуються для порівняння числових змінних у межах кластерів або категорій. Вони демонструють розмах

значень, медіану, а також наочно виявляють викиди. Зокрема, використання boxplot для аналізу суми збору у кожному кластері дає змогу виокремити кампанії з надзвичайно високим рівнем активності або, навпаки, неефективністю.

Графіки розсіювання (Scatter Plot) забезпечують зручне представлення зв'язку між двома числовими змінними. У нашому випадку це може бути співвідношення кількості донатів та суми збору, що дозволяє візуально оцінити наявність потенційної кореляції між масштабом залучення донаторів та фінансовими результатами.

Теплові карти (Heatmap) використовуються для візуалізації кореляційної матриці. Вони дозволяють швидко оцінити силу та напрямок зв'язків між числовими параметрами. Наприклад, позитивна кореляція між тривалістю кампанії та зібраною сумою або негативна залежність між середнім донатом і кількістю донорів можуть бути легко виявлені завдяки цій формі візуалізації.

Дендрограми, що є результатом ієрархічної кластеризації, показують ступінь подібності між кампаніями та процес їх об'єднання у більші супергрупи. Така візуалізація є особливо корисною для дослідження структурної організації зборів, наприклад, за пріоритетністю або тематичним напрямком.

Лінійні графіки (Line Charts), зокрема графік методу ліктя, застосовуються під час визначення оптимальної кількості кластерів у методі k-середніх. Графік демонструє зміну внутрішньокластерної інерції (WCSS) залежно від значення `k` і дозволяє знайти "лікоть" — оптимальне число кластерів, після якого приріст ефективності суттєво сповільнюється.

Таким чином, поєднання різних методів візуалізації в аналітичній платформі забезпечує багатовимірний погляд на дані, дозволяє підтвердити аналітичні гіпотези та зробити висновки доступними навіть для користувачів без математичної підготовки.

Модуль plot_generator.py інтегровано до основної логіки вебінтерфейсу, зокрема в app.py. Кожна функція створення графіка отримує відповідний фрейм даних, а також параметри виводу. Після генерації зображення зберігається у

тимчасовій директорії та відображається в інтерфейсі (Рис. 3.8.1) Streamlit.

Рисунок 3.8.1 – Приклад відображення графіку за допомогою веб-інтерфейсу

Особливість реалізації — автоматичний підбір формату візуалізації залежно від типу змінної (числова/категоріальна), що дозволяє уникнути ручного налаштування для кожного випадку. Також реалізовано опцію додавання опису графіка — заголовку, підписів осей та пояснення.

У Streamlit передбачено динамічний вивід візуалізацій: після запуску аналізу користувач одразу бачить відповідні графіки (Рис. 3.8.2), які відповідають обраному методу (кластеризація, кореляції, аномалії тощо). Наприклад:

- після кластеризації — scatter plot з кольоровим маркуванням кластерів;
- після обчислення кореляцій — heatmap з найвищими значеннями;
- після виявлення аномалій — графік розподілу з виділеними аномальними точками.

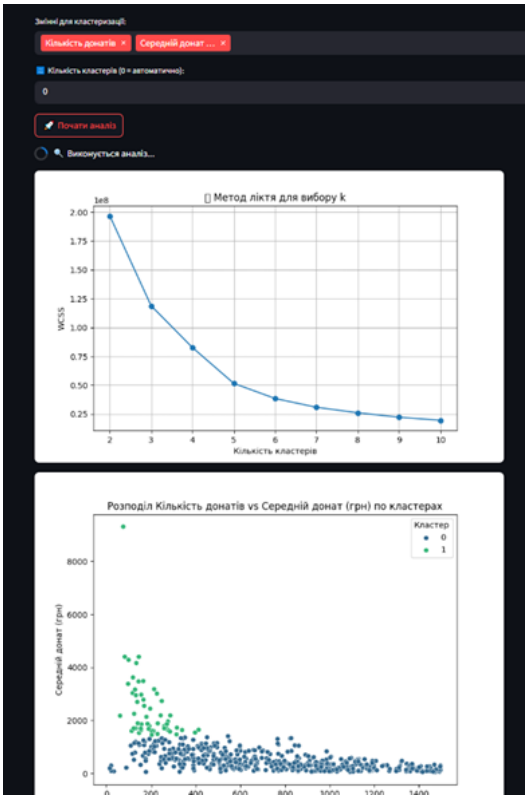

Рисунок 3.8.2 – Приклад вигляду графіків під час виконання аналізу

Усі графіки також автоматично додаються до фінального PDF-звіту через інтеграцію з report_generator.py (Рис. 3.8.3).

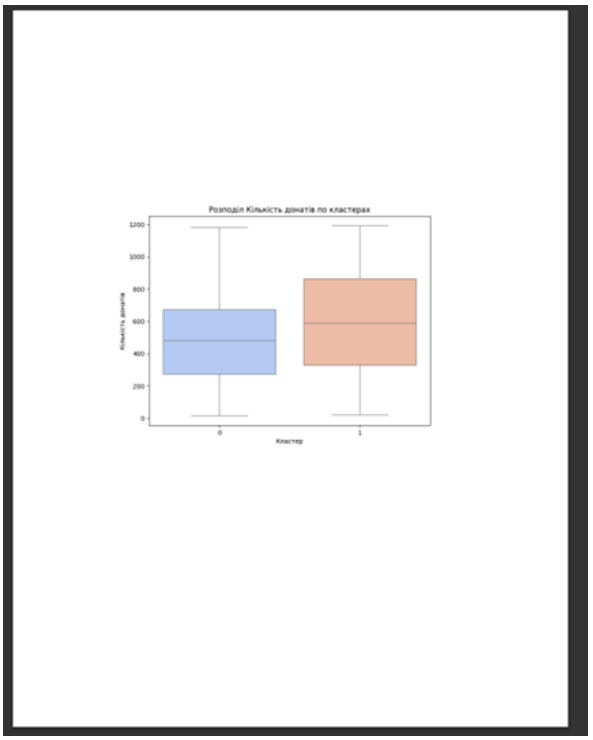

Рисунок 3.8.3 – Приклад вигляду графіку у PDF-звіті

Завдяки візуалізації користувачі платформи — координатори, аналітики,

донори — можуть:

- швидко інтерпретувати результати аналізу;
- приймати рішення на основі реальних патернів і залежностей;
- демонструвати дані у зрозумілому форматі для звітів та презентацій.

3.9 Автоматизована генерація аналітичного звіту за результатами аналізу

Автоматизоване формування підсумкового звіту — завершальний та невід'ємний етап роботи аналітичної платформи, який забезпечує зручне, стандартизоване представлення результатів усіх аналітичних модулів. Завдяки цьому етапу користувач — координатор, аналітик чи партнер проєкту — отримує цілісну картину аналізу у вигляді зручного для збереження та поширення PDF-документа з візуалізаціями, інтерпретаціями та рекомендаціями.

Генерація звіту реалізована через модуль report_generator.py, який тісно інтегровано з усіма іншими частинами платформи. В основі роботи — динамічне збирання усіх результатів, які формуються під час сесії аналізу: JSON-файли з обчисленнями, графіки у форматі PNG, аналітичні таблиці.

Використовується бібліотека ReportLab, яка дозволяє створювати адаптивні багатосторінкові PDF-документи, вставляти графіку, текстові блоки, заголовки, таблиці. Завдяки цьому кожен звіт формується у відповідності до обраних користувачем методів аналізу та містить лише релевантну інформацію.

Автоматично згенерований PDF-звіт формується платформою за єдиною структурою, що охоплює ключові аспекти аналітичного дослідження волонтерських кампаній. Такий формат забезпечує узгоджене, зрозуміле та зручне представлення результатів для кінцевих користувачів — від координаторів до аналітиків і керівництва організацій.

1. Титульна сторінка
   Містить назву звіту, дату генерації, короткий вступ до змісту документа та загальну характеристику проаналізованих кампаній (кількість записів, період, джерело даних).

2. Огляд ключових показників
   Представлені агреговані статистики — середні, медіанні значення, стандартні відхилення для основних метрик, таких як сума збору, кількість донатів, тривалість кампаній. Дані наведено у форматі зведених таблиць.

3. Кластерний аналіз

   - Опис кластерів: загальна кількість, розміри, Silhouette Score для кожного кластера;
   - Візуалізація результатів у вигляді boxplot і графіків розсіювання (scatter plot), що відображають внутрішні особливості кластерів;
   - Таблиці з розподілом кампаній за кластерами для подальшого аналізу.

4. Кореляційний аналіз

   - Візуалізована матриця кореляцій у форматі heatmap, що демонструє зв’язки між числовими параметрами;
   - Таблиця з топ-5 найсильніших позитивних та негативних кореляцій, що дозволяє виявити найзначущі взаємозалежності.

5. Аналіз аномалій

   - Список кампаній з виявленими аномаліями, із зазначенням методу виявлення (наприклад, стандартне відхилення або IQR), значення та класифікація типу відхилення;
   - Графічне представлення розподілу значень для виявлення граничних зон.

6. Оцінка ефективності: UtilizationRate

   - Гістограма розподілу коефіцієнта корисного завантаження, що дозволяє візуально оцінити інтенсивність зборів;
   - Таблиця з кампаніями, що мають найвищі та найнижчі показники UR;
   - Зведені середні значення UtilizationRate за категоріями або напрямками допомоги.

Формування фінального аналітичного звіту у форматі PDF здійснюється на основі структурованих даних, що генеруються в процесі виконання відповідних аналітичних модулів. Дані зберігаються у кількох форматах, кожен з яких відіграє свою роль у складанні змістовного та візуально зрозумілого

документа:

- JSON – містить числові характеристики, результати статистичних обчислень, структури кластерів, а також зведені таблиці, що формуються в процесі кластеризації, кореляційного аналізу та розрахунку UtilizationRate.
- PNG – графічні зображення: діаграми розподілу, дендрограми, теплові карти, boxplot-и та scatter plot-и, які використовуються для візуалізації результатів і включаються до звіту.
- CSV – таблиці з деталізованими даними: список виявлених аномалій, агрегації за категоріями, а також інші допоміжні табличні дані.

Усі файли автоматично створюються під час роботи основних модулів системи (Рис. 3.9.2) (clustering.py, correlation_analysis.py, anomaly_detection.py, тощо) та передаються до модуля генерації звітності, який компонує їх у завершений документ (Рис. 3.9.1).

Рисунок 3.9.1 – Приклад вигляду інформації у згенерованому звіті

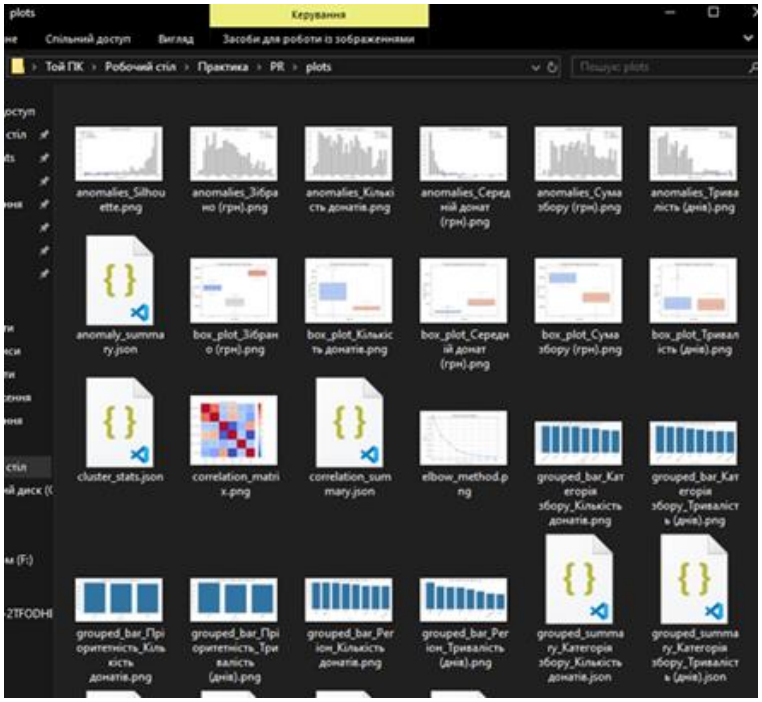

Рис. 3.9.2 – Приклад згенерованих матеріалів для аналізу

У випадку роботи через веб-інтерфейс (Streamlit), кнопка "Завантажити звіт" запускає функцію генерації на основі поточних результатів сесії (Рис. 3.9.3). Користувач може самостійно обрати методи, які буде включено до фінального документа, що забезпечує гнучкість.

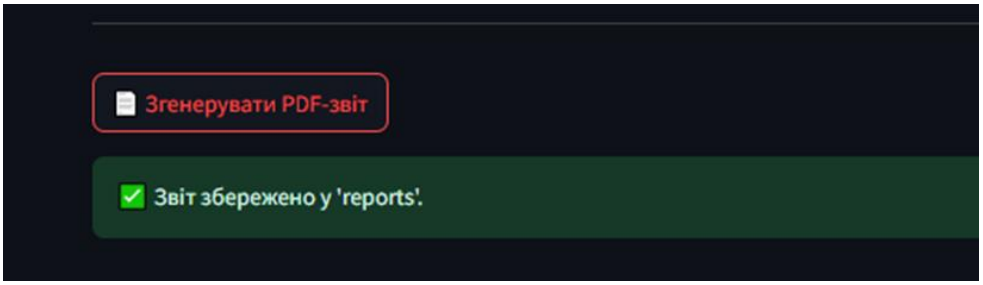

Рисунок 3.9.3 – Результат успішної генерації звіту

Автоматизований процес формування звіту є ключовим елементом платформи, який дозволяє значно підвищити ефективність обробки результатів аналізу. Основні переваги полягають у наступному:

- Відсутність потреби у ручній підготовці документів — звіт формується автоматично на основі збережених результатів аналізу, що економить час та мінімізує ризик помилок;

- Уніфікований формат звітності — усі звіти відповідають єдиному шаблону, що забезпечує послідовність, структурованість і полегшує сприйняття інформації;
- Зручність зберігання та обміну — готовий PDF-документ містить усі графіки, таблиці та висновки, придатний для подальшого аналізу, внутрішнього користування, а також представлення донорам, партнерам чи керівництву організації.

Цей підхід сприяє підвищенню прозорості, спрощує звітність і робить аналітику доступною навіть для нефахових користувачів (Рис. 3.9.4).

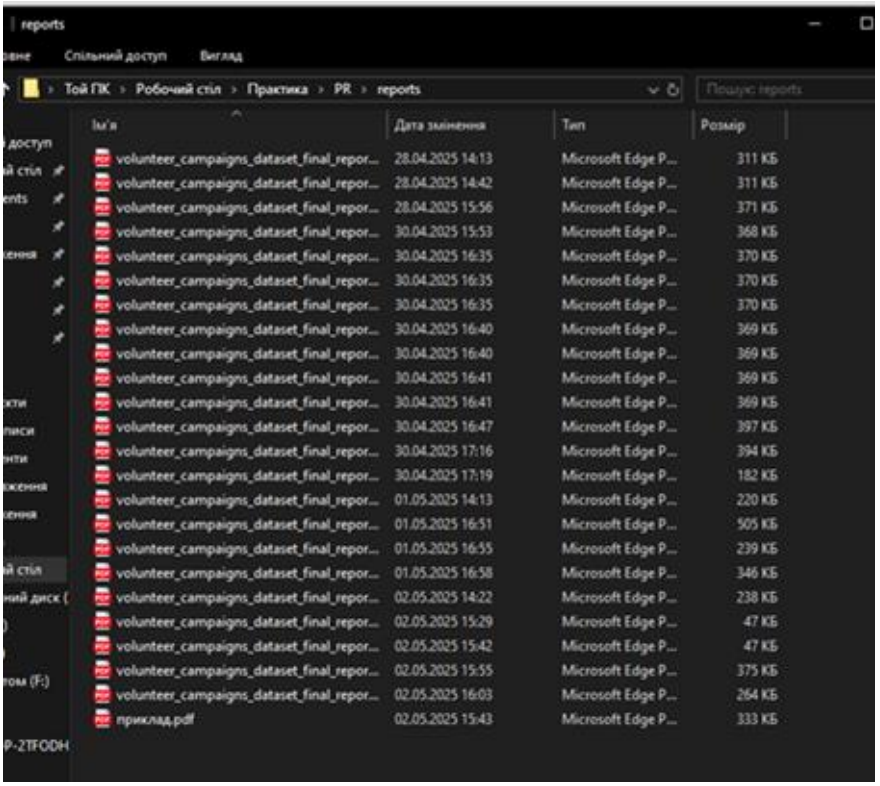

Рисунок 3.9.4 – Приклад автоматично згенерованих звітів

3.10 Оцінка якості моделей та валідація аналітичних результатів

Оцінка якості роботи аналітичної платформи є обов’язковим елементом у процесі її розробки, оскільки забезпечує достовірність результатів, стабільність обчислень та відповідність очікуваному логічному й статистичному змісту. У межах реалізованого рішення було впроваджено окремі модулі для валідації результатів і тестування коректності роботи ключових етапів аналізу.

У складі проєкту виокремлено два функціональні блоки:

- validate_methods.py — виконує програмну перевірку типів даних, меж значень, наявності обов’язкових колонок, правильності виконаних обчислень;
- test_methods.py — орієнтований на модульне тестування логіки окремих аналітичних компонентів, зокрема кластеризації, виявлення аномалій та розрахунків UtilizationRate.

Ці модулі не впливають безпосередньо на аналітичні результати, але функціонують як механізми гарантування їхньої відповідності та дозволяють виявляти потенційні логічні помилки або порушення структури на ранньому етапі.

Під час виконання аналізу важливо переконатися, що:

- обчислення виконуються лише на числових значеннях (перевірка типів);
- похідні змінні, як-от Середній донат чи UtilizationRate, мають допустимі значення;
- розподіли не містять значень, що суперечать логіці — наприклад, Зібрано не повинно бути менше нуля, Кількість донатів не повинна дорівнювати нулю при ненульовій сумі.

Такі перевірки реалізовано як через внутрішні умови в аналітичних модулях, так і через узагальнені функції, викликані після побудови кластерів чи обчислення показників.

Особливу увагу приділено валідації роботи алгоритмів кластеризації. Зокрема, перевіряється:

- чи дійсно кожен запис було призначено до одного з кластерів;
- чи не вийшло за межі допустимих значень значення Silhouette Score;
- чи відповідає кількість кластерів або глибина ієрархії очікуваному параметру.

Для виявлення проблем на рівні кореляційного аналізу модуль test_methods.py також фіксує аномальні коефіцієнти кореляції (понад допустимі межі [–1; 1]) або випадки повної відсутності числових значень (Рис. 3.10.1).

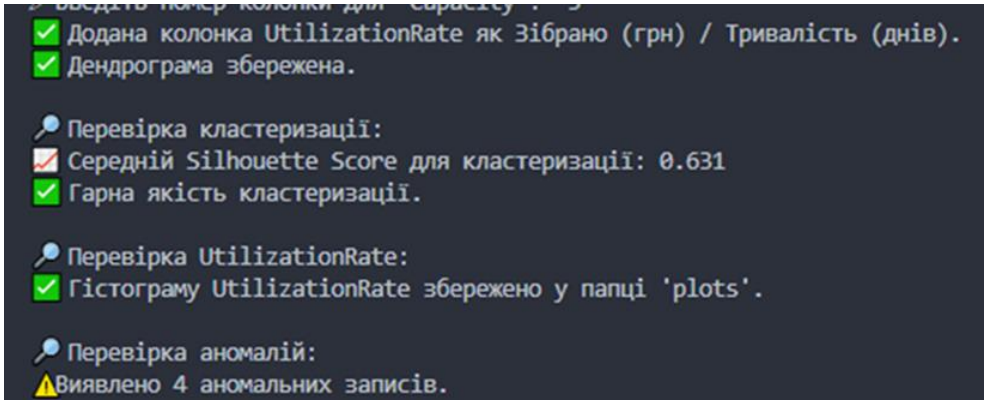

Рисунок 3.10.1 – Приклад результату перевірки

Усі виведені результати (графіки, JSON-структури, таблиці) проходять перевірку на:

- відповідність ключів і форматів JSON-файлів (наприклад, чи містить кожен кластер дані про середнє, медіанне значення тощо);
- повноту PNG-графіків та їхню відповідність обраним змінним;
- наявність усіх секцій у фінальному PDF-звіті.

Таким чином, реалізована оцінка якості аналітики є невіддільною частиною архітектури платформи та забезпечує довіру до результатів, що формуються як у внутрішньому середовищі, так і для зовнішніх стейкхолдерів (Рис. 3.10.2).

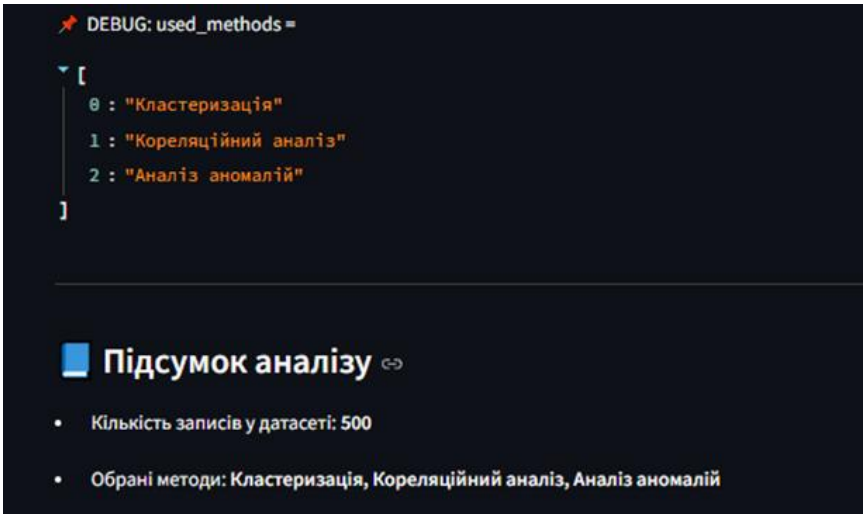

Рисунок 3.10.2 – Приклад перевірки через веб-інтерфейс

3.11 Висновки до розділу 3

У третьому розділі роботи здійснено практичну реалізацію інтегрованої аналітичної платформи для обробки та аналізу даних волонтерських кампаній, що включає повний цикл — від завантаження даних до формування підсумкових звітів.

Розроблена архітектура платформи має модульну структуру, що забезпечує її масштабованість, гнучкість та простоту інтеграції нових функцій без потреби змін у базовій логіці системи. Кожен етап обробки даних винесений у окремий модуль, що сприяє підвищенню надійності, швидкості тестування й підтримки платформи.

Особливу увагу приділено реалізації модуля попередньої обробки даних, який автоматизує завантаження Excel-файлів, очищення структури, нормалізацію числових змінних, уніфікацію категоріальних ознак та створення нових похідних змінних. Це дозволяє суттєво підвищити якість вхідних даних для подальшого аналізу.

В основі аналітичного ядра платформи реалізовано декілька методів кластеризації:

- Метод k-середніх дозволяє виділити типові профілі волонтерських кампаній на основі заданої кількості кластерів з використанням методів оптимізації (лікоть, Silhouette Score).
- Ієрархічна кластеризація дає змогу гнучко досліджувати структуру даних через дендрограму, без попереднього визначення кількості груп.

Кореляційний аналіз між основними змінними реалізовано у вигляді інтерактивних heatmap-карт та таблиць зі списками найбільш значущих позитивних і негативних кореляцій. Це дозволяє формулювати висновки про взаємозв'язки між кількісними характеристиками волонтерських ініціатив і потенційні фактори їх успішності.

Виявлення аномалій реалізовано за допомогою класичних статистичних методів (Z-оцінка, IQR), що дозволяє знаходити нетипові записи і підвищувати

достовірність загальних висновків. Аномальні кампанії виділяються окремо у звітах, що дає змогу приймати індивідуальні управлінські рішення щодо їх розгляду.

Розрахунок показника UtilizationRate забезпечує оцінку ефективності кампаній не лише за абсолютними результатами, але й з урахуванням тривалості проектів. Це дає змогу порівнювати ініціативи між собою за уніфікованим критерієм інтенсивності збору коштів.

Реалізовано автоматизовану генерацію звітів у форматі PDF, які містять всі ключові результати аналізу: агреговану статистику, кластеризацію, кореляційні залежності, аномалії та оцінку ефективності. Структура звітів забезпечує їхню читабельність навіть для користувачів без глибокої технічної підготовки.

Візуалізація результатів інтегрована у вебінтерфейс Streamlit, що дозволяє оперативно переглядати графіки, діаграми й таблиці під час виконання аналізу. Реалізовані методи візуалізації включають гістограми, scatter plot-и, дендрограми, теплові карти та boxplot-и.

Додатково забезпечено оцінку якості роботи платформи через модулі валідації результатів і тестування методів. Це гарантує достовірність обчислень і дозволяє швидко виявляти потенційні помилки на етапі аналізу.

У результаті, побудована аналітична платформа продемонструвала здатність ефективно обробляти дані волонтерських кампаній, виявляти закономірності, групувати ініціативи за профілями ефективності, формувати автоматизовані звіти та забезпечувати прозору і швидку аналітику для управлінських потреб гуманітарних проектів.

РОЗДІЛ 4
ТЕХНОЛОГІЯ ПРАКТИЧНОГО ЗАСТОСУВАННЯ РОЗРОБЛЕНОЇ АНАЛІТИЧНОЇ ПЛАТФОРМИ

4.1 Загальний підхід до вирішення задач аналітики даних у волонтерському русі

У межах реалізованого дослідження було розроблено функціональну аналітичну платформу, здатну обробляти масиви даних про волонтерські кампанії, проводити ключові типи аналітики та автоматично формувати звіти. Основна мета — підтримати координаторів та аналітиків у прийнятті обґрунтованих рішень на основі даних, зменшивши залежність від ручного аналізу та інтуїтивного підходу.

Загальна архітектура платформи побудована на модульному принципі, де кожен компонент виконує чітко визначене завдання: від попередньої обробки та валідації даних до глибокого аналізу та фінального формування звіту. Уся система реалізована мовою Python із використанням бібліотек pandas, scikit-learn, matplotlib, seaborn, scipy, Streamlit, ReportLab тощо. Завдяки такій архітектурі платформа не лише забезпечує гнучкість, але й легко адаптується під нові задачі або зміну структури вхідних даних.

Аналітичні задачі, які виконує система, охоплюють усі ключові напрями волонтерської діяльності. Зокрема:

- Кластеризація кампаній за схожими характеристиками (наприклад, за сусою збору та кількістю донатів)
- Кореляційний аналіз змінних для виявлення статистично значущих залежностей
- Оцінка ефективності кампаній через обчислення коефіцієнта UtilizationRate
- Виявлення аномальних записів для підвищення точності подальших аналітичних висновків
- Побудова візуалізацій за категоріями, пріоритетністю, регіоном
- Автоматичне формування PDF-звітів, придатних для подальшого поширення серед стейкхолдерів.

Для демонстрації практичної реалізації, платформа була протестована на згенерованому датасеті з понад 500 записами кампаній. Дані включали як числові параметри (сума збору, кількість донатів, тривалість), так і категоріальні змінні (регіон, пріоритетність, категорія збору). Усі етапи виконання — від завантаження файлу до генерації звіту — можуть бути ініційовані через веб-інтерфейс користувача, побудований на основі Streamlit. Це забезпечує доступність платформи навіть для користувачів без технічної підготовки.

Таким чином, загальний підхід полягає в поетапному, автоматизованому аналізі волонтерських даних, у якому кожен модуль вносить свій внесок у повноцінну картину діяльності. Це дозволяє ефективно вирішувати поставлені задачі — як у рамках дослідження, так і в реальних умовах функціонування гуманітарних ініціатив.

4.2 Практична реалізація кластеризації кампаній на основі волонтерських даних

Кластеризація є одним з ключових аналітичних етапів, реалізованих у платформі. Метою кластеризації є виявлення схожих груп волонтерських кампаній, які мають подібні параметри — такі як сума зібраних коштів, кількість донатів, середній внесок, тривалість кампанії тощо. Це дозволяє координаторам бачити типові шаблони зборів і приймати стратегічні рішення на основі моделей, які вже продемонстрували ефективність.

У рамках дослідження було реалізовано два методи кластеризації:

- Метод k-середніх – для швидкої та ефективної сегментації даних за заданою або автоматично визначеною кількістю кластерів.
- Ієрархічна кластеризація – для візуального аналізу варіантів групування та дослідження структури подібності між кампаніями

Визначення кількості кластерів (k) реалізовано двома способами:

- Автоматично – за допомогою методу ліктя, який аналізує зменшення WCSS
- Вручну – користувач може задати значення k через інтерфейс

На основі методу ліктя було встановлено, що оптимальною кількістю кластерів є 2, оскільки після цієї точки спостерігається згасання темпу покращення WCSS (рис. 4.2.1).

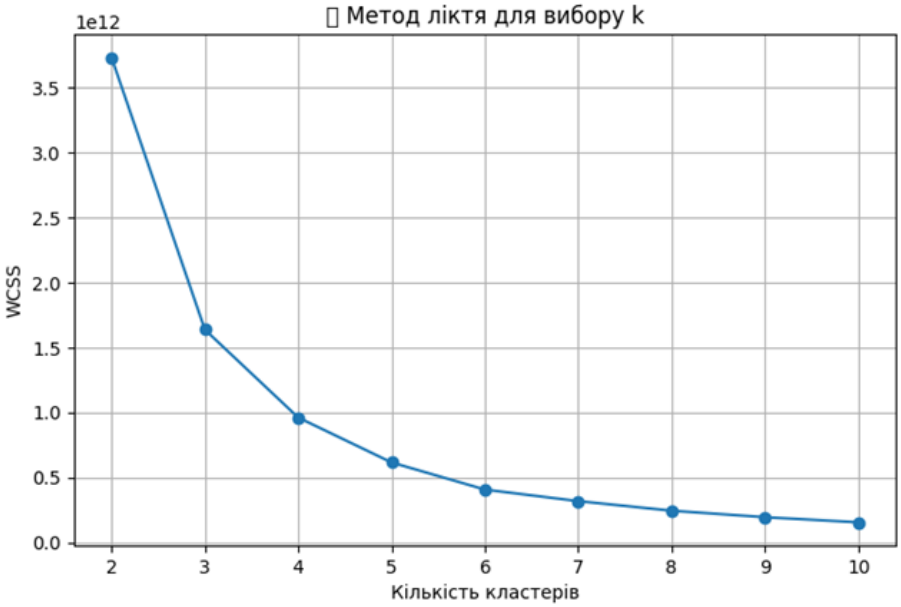

Рисунок 4.2.1 – Метод ліктя для визначення оптимальної кількості кластерів

Після кластеризації усі записи було розподілено між двома групами (Рис. 4.2.2):

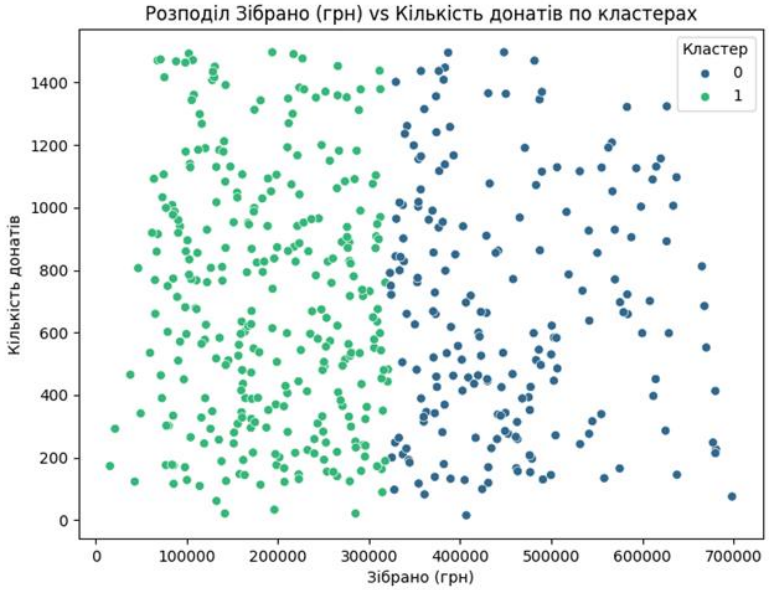

Рисунок 4.2.2 – Розподіл кампаній за ознаками «Зібрано (грн)» та «Кількість донатів» у розрізі кластерів

Кластер 0: кампанії з вищим середнім обсягом зібраних коштів (≈453 тис. грн), тривалістю ≈21 день та нижчою кількістю донатів;

Кластер 1: кампанії з меншим обсягом зібраних коштів (≈190 тис. грн), але з дещо більшою кількістю донатів.

Виявлено два чітко виражені типи кампаній: високобюджетні/швидкі кампанії та більш розсіяні за обсягами збору, але активні за кількістю учасників.

Значення Silhouette Score для обох кластерів склали 0.54 та 0.62 відповідно, що свідчить про достатню чіткість меж між групами (рис. 4.2.3, 4.2.4).

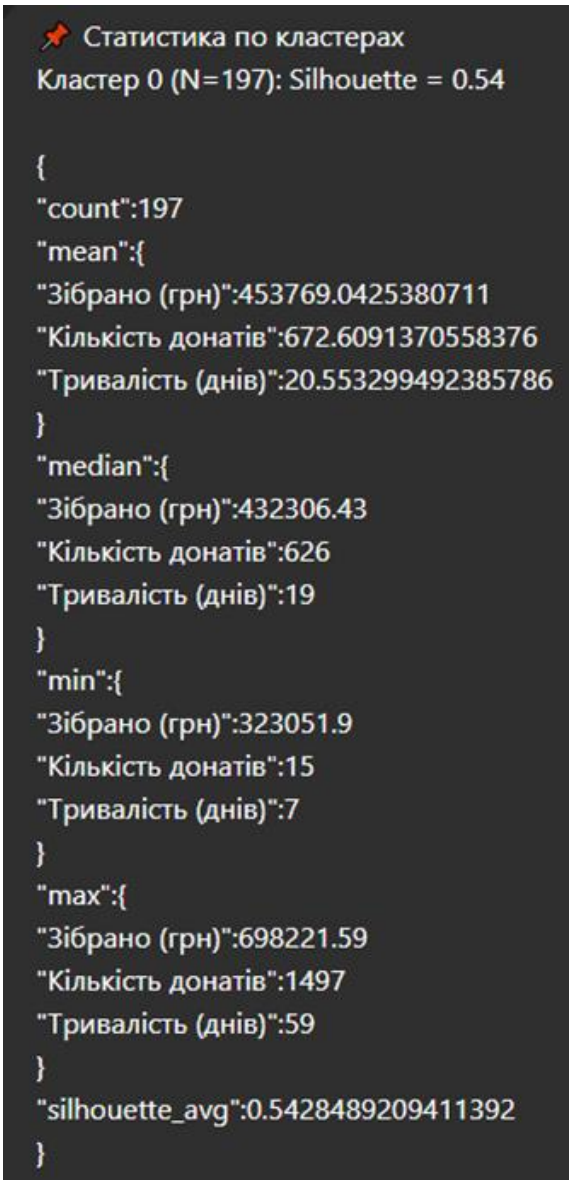

Рисунок 4.2.3 – Деталізована статистика кластера 0: кількість кампаній, середні медіанні, мінімальні та максимальні значення ключових

показників

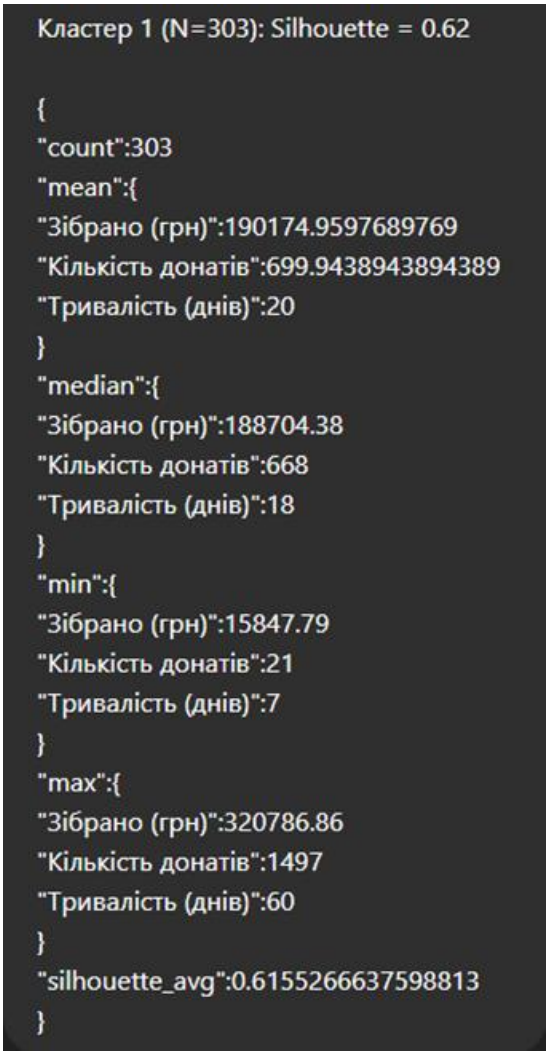

Рисунок 4.2.4 – Деталізована статистика кластера 1: кількість кампаній, середні медіанні, мінімальні та максимальні значення ключових показників

Для кожного кластеру було побудовано розподіли за сумою збору, кількістю донатів і тривалістю (boxplot) (Рис. 4.2.5 – 4.2.7)

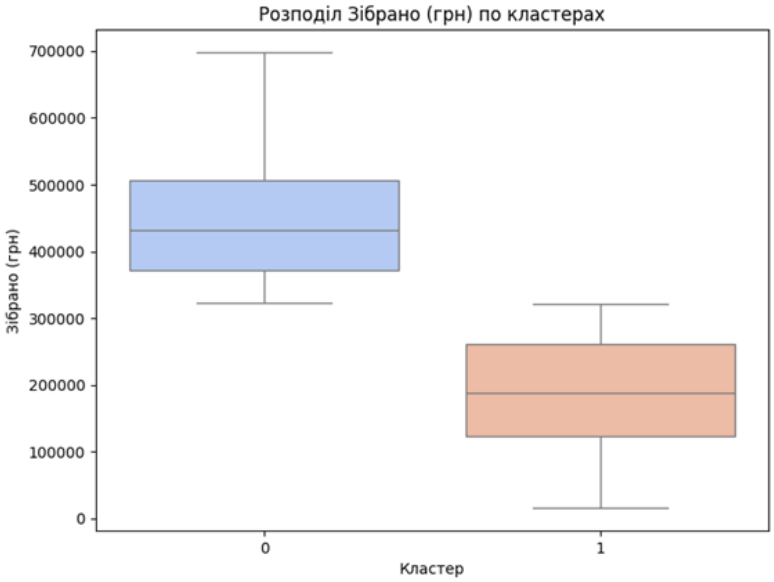

Рисунок 4.2.5 – Boxplot розподілу зібраної суми (грн) у розрізі кластерів

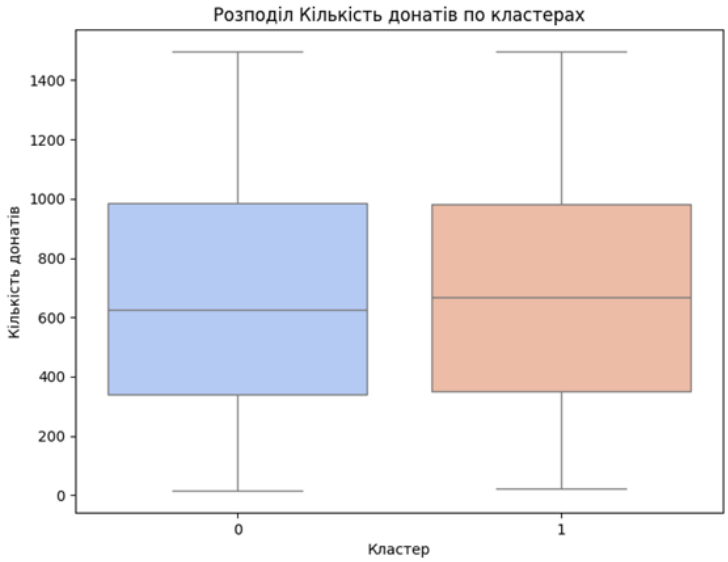

Рисунок 4.2.6 – Boxplot розподілу кількості донатів по кластерах

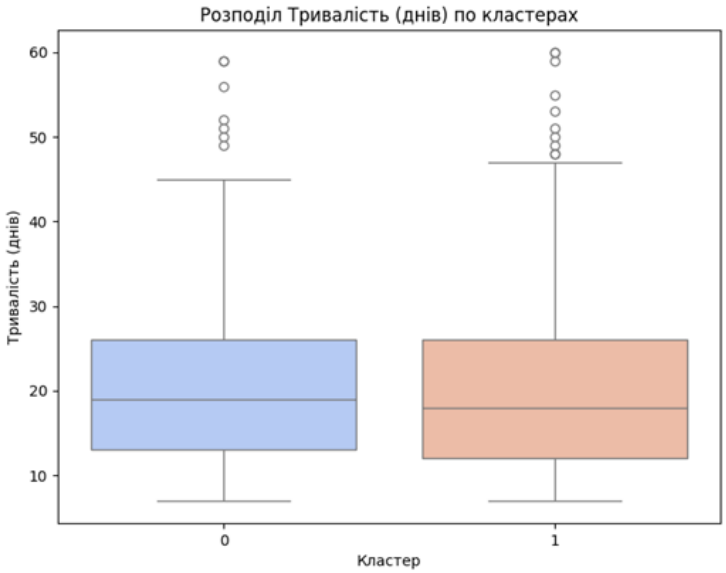

Рисунок 4.2.7 – Boxplot тривалості кампаній (днів) для кожного кластеру

Ієрархічний підхід реалізовано через побудову дендрограми. Це дозволило побачити багаторівневу структуру подібності та візуально обрати можливе значення k у майбутньому (рис. 4.2.8). Такий підхід є корисним при подальшому групуванні за більш специфічними змінними (наприклад, окремо для кожного регіону або категорії)

Рисунок 4.2.8 – Дендрограма ієрархічної кластеризації волонтерської кампанії

Кластеризація виявилася ефективною для:

- Виявлення шаблонів у кампаніях
- Ранжування за подібністю до «типових» профілів
- Створення бази для рекомендацій майбутнім ініціативам

Усі результати автоматично інтегруються у звіт, що робить аналіз зрозумілим та доступним навіть для нефахових користувачів.

4.3 Проведення кореляційного аналізу змінних у реальних даних кампаній

Вивчення взаємозв’язків між ключовими показниками волонтерських кампаній є важливим елементом аналітики, що дає змогу виявити чинники, які впливають на ефективність зборів. У розробленій платформі цей функціонал реалізовано через модуль correlation_analysis.py, який автоматично будує кореляційну матрицю, heatmap-візуалізацію та формує таблиці з топ-позитивними та негативними зв’язками.

Для обчислення ступеня зв’язку між змінними використано коефіцієнт кореляції Пірсона, що дозволяє оцінити лінійні залежності між числовими параметрами. Кореляції розраховуються між такими змінними:

- Сума збору
- Зібрано
- Кількість донатів
- Середній донат
- Тривалість кампанії
- Silhouette Score (оцінка кластерної структури)

На основі отриманих значень форсується теплова матриця (heatmap), яка дозволяє візуально оцінити силу і напрямок зв’язків (Рис. 4.3.1)

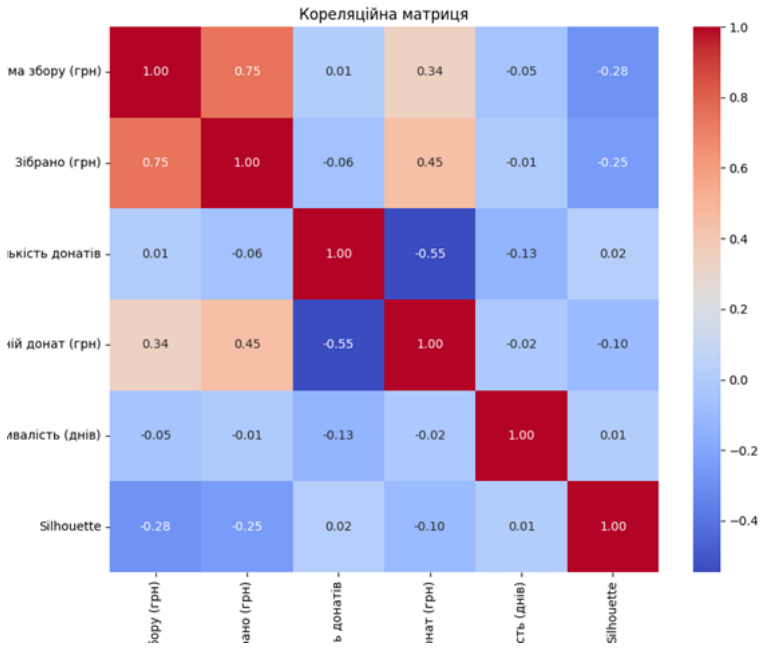

Рисунок 4.3.1 – Матриця кореляції між основними змінними волонтерських кампаній

Кольори від червоного (позитивна кореляція) до синього (негативна) забезпечують швидке розуміння характеру взаємодії між параметрами.

Найсильніші позитивні кореляції (Рис. 4.3.2):

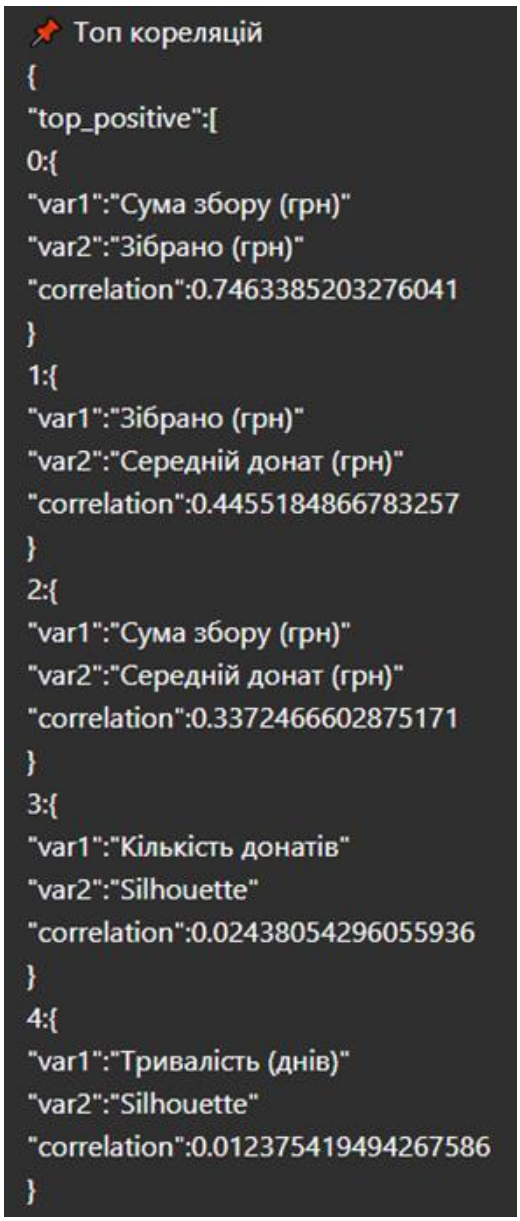

Рисунок 4.3.2 – Топ – 5 позитивних кореляцій між змінними

Сума збору ↔ Зібрано: ρ = 0.75 — логічний зв’язок, що підтверджує достовірність структури даних.

Зібрано ↔ Середній донат: ρ = 0.45

Сума збору ↔ Середній донат: ρ = 0.34

Найсильніші негативні кореляції:

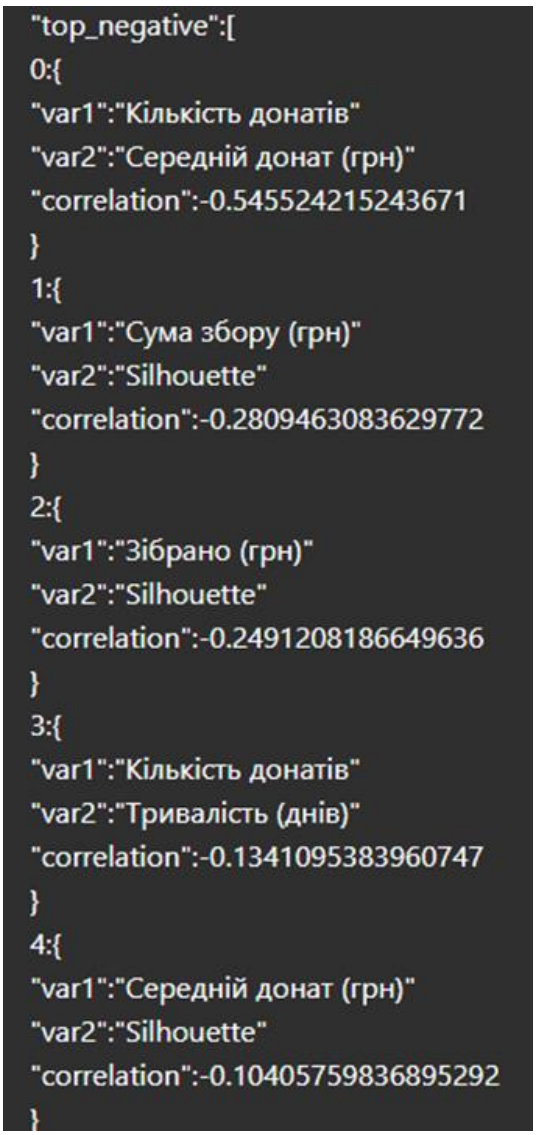

Рис. 4.3.3 – Топ – 5 негативних кореляцій між змінними

Кількість донатів ↔ Середній донат: ρ = −0.55 — чим більше донатів, тим нижча їх середня сума.

Сума збору ↔ Silhouette Score: ρ = −0.28

Зібрано ↔ Silhouette Score: ρ = −0.25

Такі зв’язки дають змогу припустити, що кампанії з великою кількістю донатів, ймовірно, менш сегментовані в кластерному аналізі, що свідчить про їх розмитий профіль.

Модуль дозволяє окремо фільтрувати пари змінних, які перевищують заданий поріг (наприклад, $|\rho| > 0.3$). Така фільтрація зручно інтегрується у звіт

і дає змогу сконцентрувати увагу на найбільш інформативних зв’язках без надлишку інформації (Рис. 4.3.4).

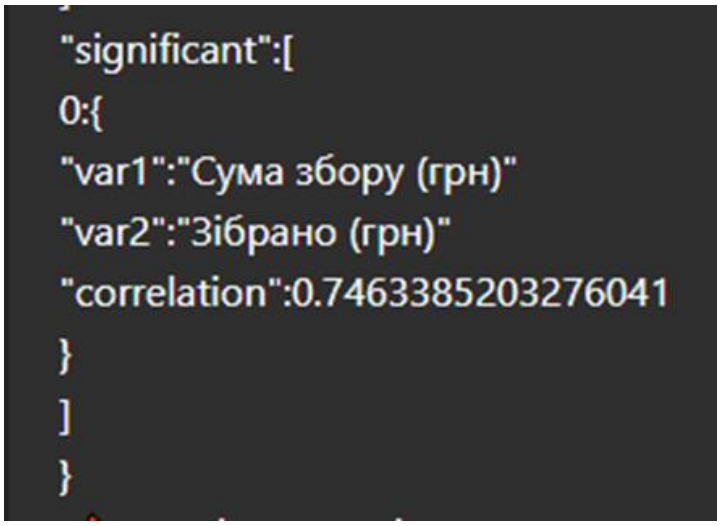

Рисунок 4.3.4 – Найзначущі кореляції у вибірці

Кореляційний аналіз у нашій платформі виконує не лише діагностичну, а й прогностичну функцію. Виявлені зв’язки можуть бути використані для створення моделей прогнозування результативності кампаній, а також для раннього виявлення неефективних конфігурацій (наприклад, кампаній із великою кількістю донатів, але низькою загальною сумою збору).

4.4 Розрахунок коефіцієнта UtilizationRate та його інтерпретація для оцінки ефективності

Одним із ключових показників ефективності волонтерських кампаній, реалізованих на платформі, є UtilizationRate (UR) — коефіцієнт корисного завантаження. Цей показник відображає інтенсивність використання ресурсів, зокрема фінансових, у часі, і дозволяє здійснювати порівняльну оцінку кампаній, незалежно від їх масштабу або тривалості.

Алгоритм розрахунку UR реалізовано у модулі utilization_calculator.py. Обчислення проводяться автоматично під час завантаження та попередньої обробки даних. Для кожного запису створюється нова колонка UtilizationRate,

значення якої надалі використовуються:

- Для побудови гістограми розподілу UR
- Для категоризації кампаній за рівнем ефективності
- Для виявлення аномалій або нетипово високих\низьких коефіцієнтів

На основі тестового датасету з 500 кампаній:

- Середній UR = 18585 грн\день
- Медіанний UR = 14952 грн\день
- Мінімальний UR = 717 грн\день
- Максимальний UR = 80919 грн\день

Абсолютно всі кампанії показали UR > 1, що свідчить про високий рівень загальної активності (рис. 4.4.1). В автоматично згенерованому розподілі спостерігається зсув до нижчих значень — це типово для волонтерських кампаній із помірною динамікою збору.

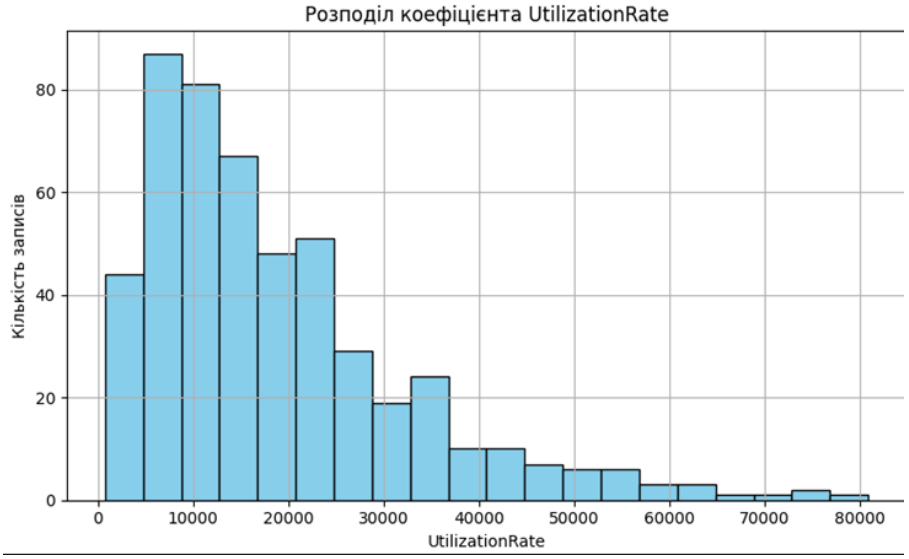

Рисунок 4.4.1 – Розподіл коефіцієнта UtilizationRate серед волонтерських кампаній

Для зручності інтерпретації всі кампанії були класифіковані за трьома рівнями ефективності:

- Низький: UR ≤ 1 (не виявлено в наявному датасеті)
- Середній: UR від 1 до 15000
- Високий: UR > 15000

Зокрема,183 з 183 кампаній потрапили до категорії «високий рівень», що демонструє узагальнену ефективність вибраної вибірки (Рис. 4.4.2).

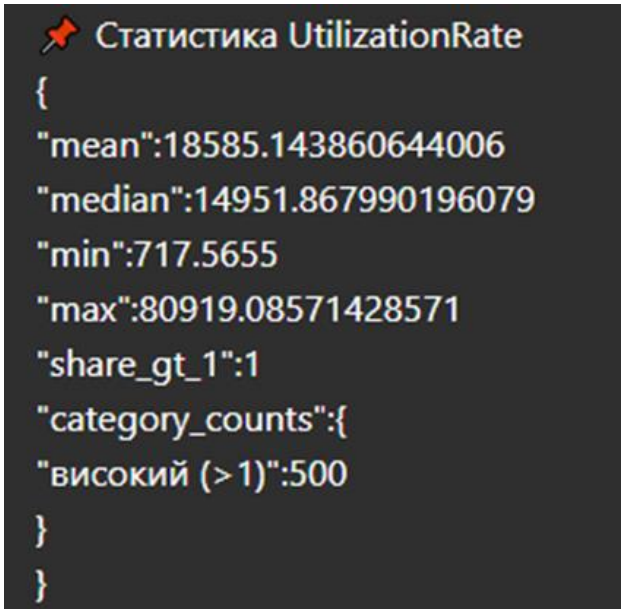

Рисунок 4.4.2 – Статистика коефіцієнта UtilizationRate у вибірці кампаній

Така категоризація використовується в подальшому для:

- Сортування кампаній у звітах
- Пошук «типових» прикладів
- Виявлення потенційних аномалій на основі UR

Значення UtilizationRate допомагає аналітикам виявляти:

- Ініціативи з високим темпом збору, що заслуговують на масштабування
- Кампанії з низькою інтенсивністю, які потребують підтримки або зміни стратегії
- Періоди пікової ефективності, що можуть збігатися з інформаційними кампаніями чи зовнішніми подіями.

4.5 Виявлення аномалій та їх роль у контролі якості зборів

Виявлення аномалій є критично важливою частиною аналітичного процесу, особливо у гуманітарному середовищі, де точність даних безпосередньо впливає на прозорість та довіру. Аномалії можуть вказувати як на технічні помилки введення, так і на винятково успішні кампанії, що суттєво відрізняються

від загальної вибірки.

Алгоритм реалізовано у модулі anomaly_detection.py. Результати зберігаються у вигляді індексів записів, маркуються в таблицях та виводяться на графіках.

Середній донат (грн): за методом Z-оцінки — 11 аномалій; за IQR — 33. Ці кампанії мали нетипово великі перекази при малій кількості донатів (Рис. 4.5.1).

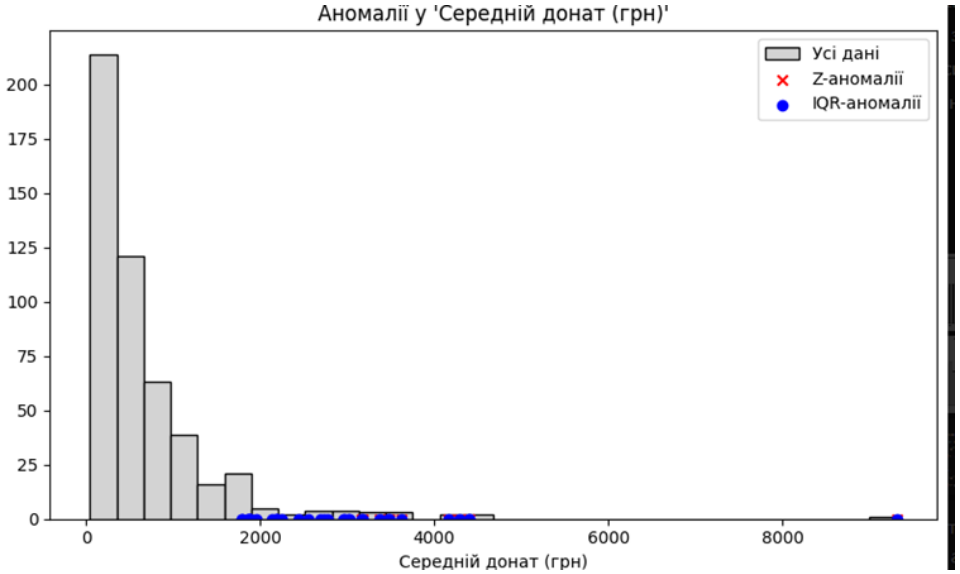

Рисунок 4.5.1 – Аномалії у Середньому донаті (грн)

Тривалість (днів): 8 аномалій (Z) і 17 (IQR). Деякі кампанії мали незвично довгий або короткий період активності (Рис. 4.5.2).

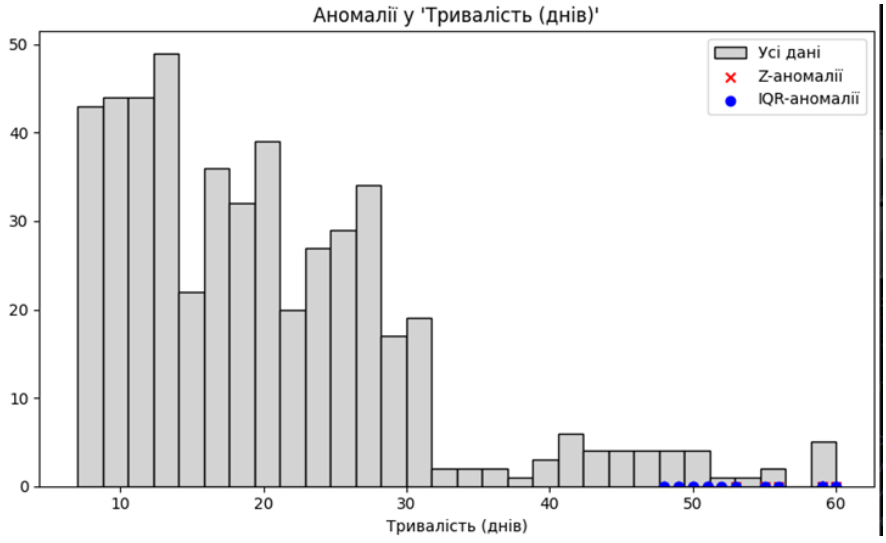

Рисунок 4.52 – Аномалії у Тривалості (днів)

Silhouette Score: ідентифіковано 4 (Z) та 40 (IQR) відхилень. Значні варіації якості кластеризації можуть бути наслідком неоднорідності даних (Рис.

4.5.3).

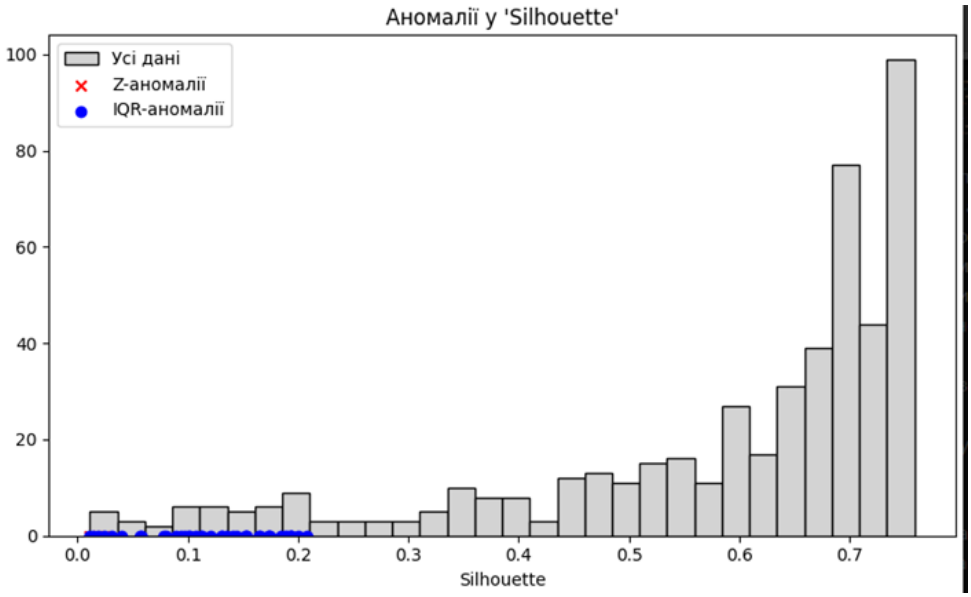

Рисунок 4.5.3 – Аномалії у значеннях Silhouette Score

Графічно всі знайдені аномалії позначені червоними точками на boxplot, а також автоматично включаються до PDF-звіту з коротким поясненням причин виявлення.

Виявлені аномалії можуть мати різну природу:

- Технічні (помилки введення, пропущені значення)
- Фінансові (надмірно великі або підозріло малі сум)
- Поведінкові (кампанії з нетиповим темпом активності)

Виявлення таких записів дозволяє:

- Покращити якість вхідних даних
- Дослідити нетипові, але потенційно ефективні сценарії
- Забезпечити прозорість звітності перед донорами

Таким чином, механізми виявлення аномалій — не лише фільтраційний інструмент, а й джерело цінної аналітичної інформації для прийняття рішень у динамічному середовищі волонтерської підтримки.

4.6 Аналіз динаміки зборів та категоризація кампаній за напрямком допомоги

Аналіз категоріальних параметрів кампаній дозволяє виявити важливі закономірності у поведінці донорів, швидкості збору коштів та рівні

підтримки для різних напрямків допомоги. У межах даного дослідження розглянуто три ключові виміри — напрямок допомоги, пріоритетність та регіональна приналежність. Оцінка здійснювалась за двома основними метриками: тривалість кампанії (днів) та середня сума збору (грн).

Згідно з графіком (рис. 4.6.1), найбільшу середню тривалість мають кампанії, що стосуються пального, медичних аптечок, дронів та тактичної оптики — понад 21 день. Це свідчить про складнішу логістику або тривалішу підготовку таких ініціатив.

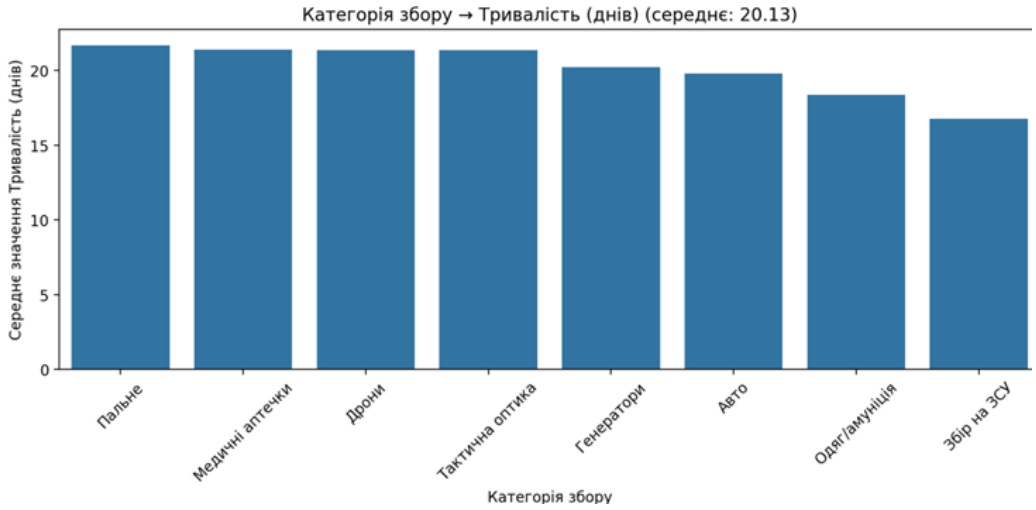

Рисунок 4.6.1 – Середня тривалість кампаній за категоріями допомоги

Водночас, середній обсяг зібраних коштів (рис. 4.6.2) був найвищим у кампаніях, спрямованих на дрони, ЗСУ, оптику та пальне — понад 400 тис. грн. Це демонструє високу довіру та залучення громадськості до технічно складних або стратегічно важливих цілей.

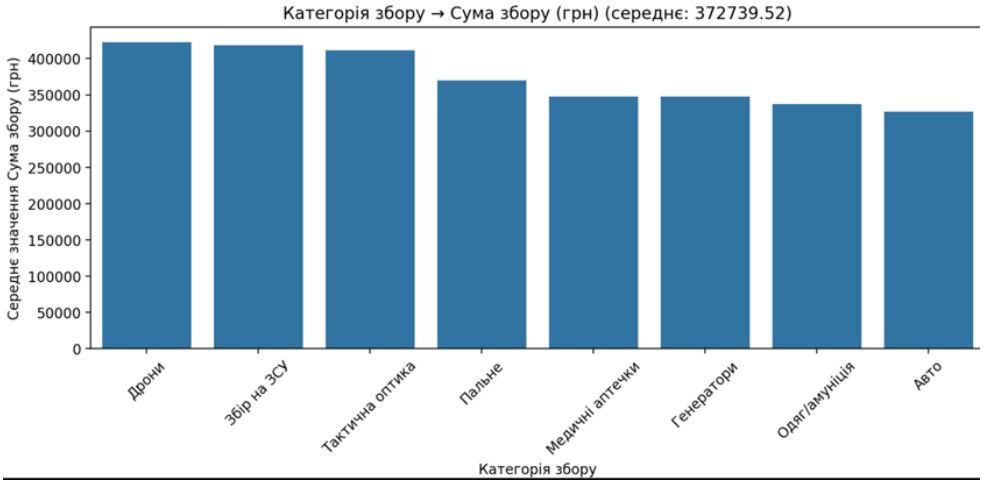

Рисунок 4.6.2 – Середня сума збору по кожній категорії

На рис. 4.6.3 видно, що тривалість кампаній практично не залежить від пріоритетності. Різниця між кампаніями з високим, середнім і низьким пріоритетом становить менше 1 дня, що є статистично незначущим.

Рисунок 4.6.3 – Залежність тривалості кампанії від пріоритету

Однак середня сума збору (рис. 4.6.4) демонструє цікаву закономірність: кампанії з найвищим пріоритетом залучають найменше коштів, у той час як середньопріоритетні мають найкращі фінансові результати. Це може пояснюватися тим, що високопріоритетні кампанії часто мають меншу цільову суму та коротші строки, що обмежує обсяг збору.

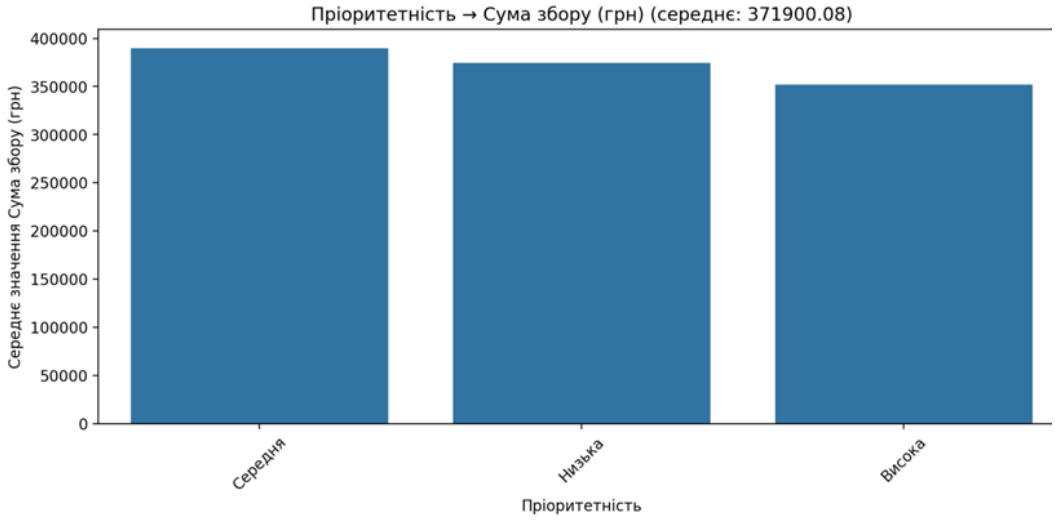

Рисунок 4.6.4 – Залежність суми збору від заданого рівня пріоритетності

На рис. 4.6.5 видно, що найдовше тривають кампанії у східних регіонах, зокрема в Дніпрі, Запоріжжі та Харкові — понад 21 день у середньому. Це може бути пов’язано з постійною потребою у підтримці в зонах, близьких до фронту.

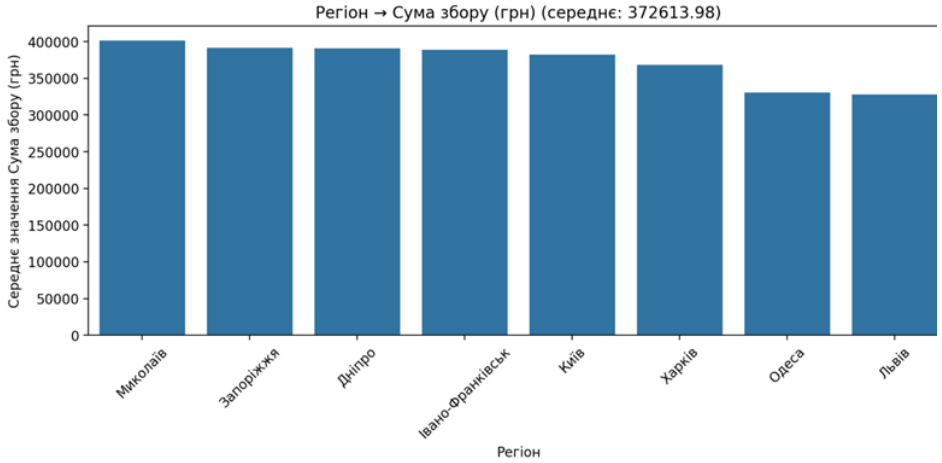

Рисунок 4.6.5 – Тривалість кампаній по регіонах

У той же час рівень фінансування кампаній у цих регіонах також є високим (рис. 4.6.6): лідирують Миколаїв, Запоріжжя, Дніпро, де сума збору наближається або перевищує 400 тис. грн.

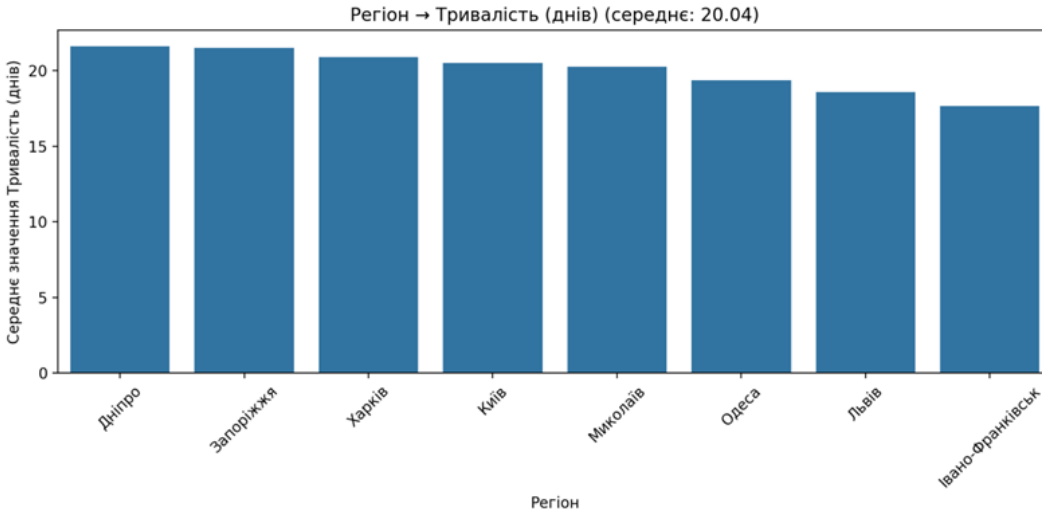

Рисунок 4.6.6 – Розподіл середньої суми збору за регіональними групами

Аналіз категоріальних ознак демонструє важливу взаємозалежність між типом кампанії, її тривалістю та фінансовою ефективністю. Тематика кампанії

прямо впливає на обсяг підтримки: технічні та військові збори мають вищі суми. Пріоритетність не гарантує високої суми збору — успішність залежить більше від мети та формулювання запиту. Регіональний контекст відіграє роль у тривалості кампаній та обсязі залучених коштів.

Усі візуалізації доступні у веб-інтерфейсі, де користувач може динамічно обирати змінні та формувати відповідні аналітичні звіти.

4.7 Формування узагальненого шаблону «типової ефективної кампанії»

Однією з цілей аналітичної платформи є створення інструменту, який дозволяє не лише аналізувати існуючі дані, а й формувати узагальнений образ найбільш ефективної волонтерської кампанії. Це особливо важливо для нових організаторів, які лише починають свою діяльність і потребують орієнтиру щодо оптимальних параметрів.

В умовах великої кількості кампаній з різними параметрами постає питання: які з них є найуспішнішими? Відповідь на це запитання дає можливість побудувати умовний "еталон" — кампанію, яка має оптимальні значення ключових показників

Такий шаблон дозволяє планувати майбутні ініціативи з урахуванням найкращих практик.

У межах реалізованої системи шаблон типової кампанії формується на основі агрегованих показників за всіма кампаніями або за вибраним кластером. Основний функціонал реалізовано через об’єднання результатів з модулів clustering.py, utilization_calculator.py та аналітичного виводу JSON-статистики.

На практиці розраховуються наступні характеристики:

- Середня сума збору — орієнтовно 321 000 грн;
- Тривалість кампанії — близько 20 днів;
- Кількість донатів — від 600 до 700;
- Середній донат — 480–510 грн;
- UtilizationRate — на рівні 27–38, що відповідає високій ефективності.

Користувач може фільтрувати дані, обираючи, наприклад, лише завершені кампанії, або лише ті, що входять до найуспішнішого кластеру за Silhouette Score.

Формування такого шаблону є корисним не лише для новачків, а й для порівняння з поточними або минулими кампаніями. Наприклад, кампанія, яка суттєво відхиляється від шаблону, може бути сигналом до:

- перегляду стратегії збору;
- змін у комунікації з донорами;
- або вказівкою на особливі умови, що вплинули на результативність.

Таким чином, «типова кампанія» виконує роль аналітичного еталону та водночас слугує інструментом стратегічного планування.

4.8 Побудова інтерактивного веб-інтерфейсу платформи на базі Streamlit

Інтерфейс взаємодії з користувачем є одним із ключових компонентів розробленої аналітичної платформи, оскільки саме через нього здійснюється керування даними, вибір методів аналізу, виведення результатів та формування звітності. Для реалізації інтерфейсу було обрано фреймворк Streamlit, що дозволяє швидко створювати інтерактивні веб-додатки на базі Python без необхідності використання окремих фронтенд-технологій.

Інтерфейс реалізовано у вигляді модуля app.py, який є єдиною точкою входу для користувача. Після запуску додатку у браузері відкривається динамічна панель, де послідовно представлено такі функціональні блоки:

Завантаження файлу: користувач обирає Excel-файл з волонтерськими даними, який автоматично зчитується, перевіряється на коректність і візуалізується перша частина таблиці.

Вибір методів аналізу: через прапорці (checkbox) або випадаючі меню користувач задає, які саме аналітичні модулі слід виконати — кластеризацію, аналіз кореляцій, аномалії, обчислення UtilizationRate тощо.

Інтерактивне відображення результатів: після виконання обраних

методів, на екрані відображаються графіки, таблиці та короткі підсумки. Для кожного типу аналізу створено окрему секцію: наприклад, Silhouette Score у кластеризації, heatmap у кореляціях, список аномальних записів — у блоці аномалій.

Платформа орієнтована на максимальну адаптивність до дій користувача. Усі дії — завантаження, вибір методів, фільтрація змінних — обробляються у режимі реального часу (Рис. 4.8.1).

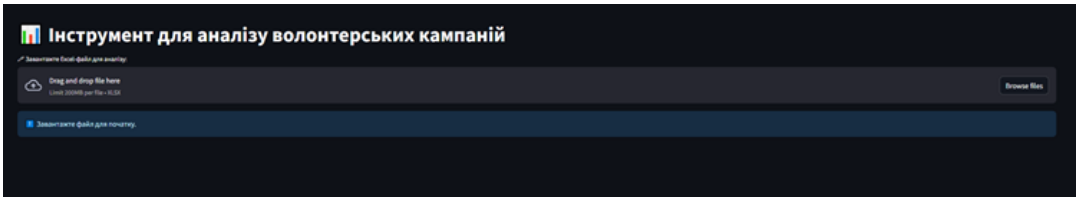

Рисунок 4.8.1 – Початковий вигляд інтерфейсу

Інтерактивна функціональність платформи дозволяє користувачеві гнучко керувати параметрами аналізу без необхідності втручання в код. Зокрема, при зміні обраного методу кластеризації — між алгоритмом k-середніх та ієрархічною кластеризацією — візуалізації автоматично оновлюються відповідно до вибраного підходу. Це дозволяє одразу побачити структуру розподілу даних за новим алгоритмом і порівняти результати (Рис. 4.8.2).

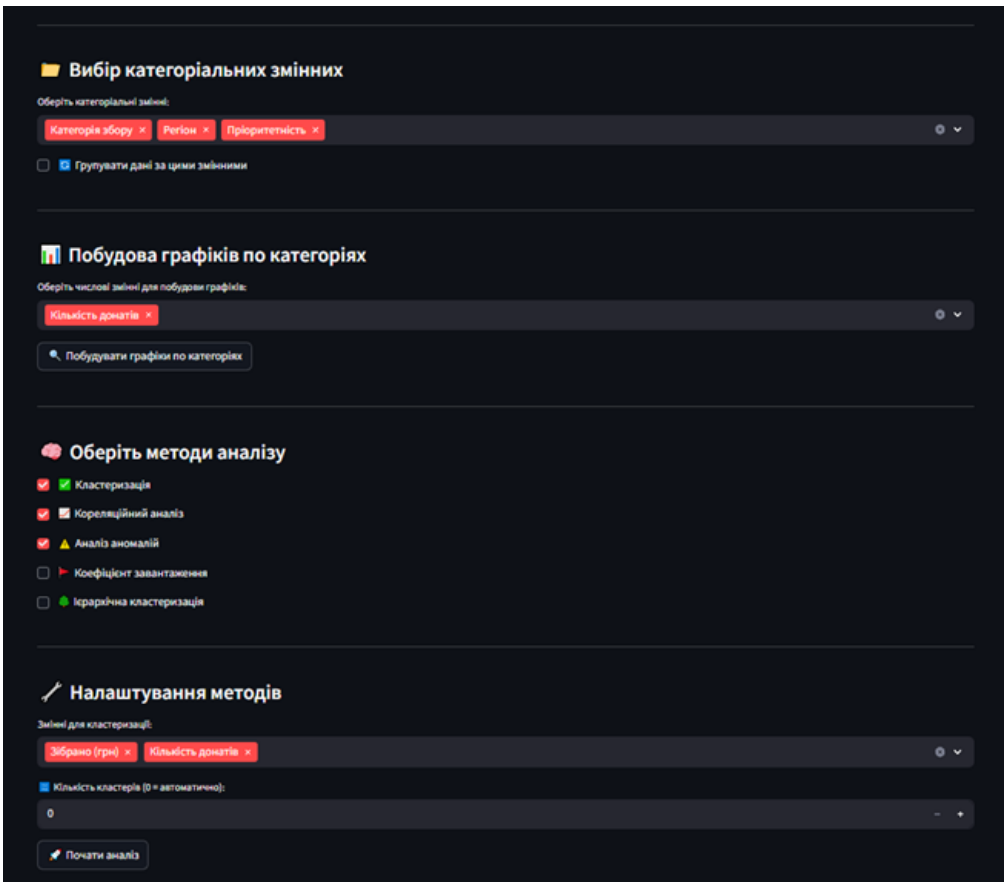

Рисунок 4.8.2 – Вигляд початкового робочого середовища

Користувач може вручну встановити кількість кластерів, що забезпечує точне налаштування під конкретне завдання, або скористатися автоматичним визначенням оптимального значення за допомогою методу ліктя. Така гнучкість особливо корисна у випадках, коли кількість природних груп у даних заздалегідь невідома.

Окрім того, інтерфейс дає змогу обирати, які саме змінні будуть використані в аналізі — це дозволяє фокусуватися лише на релевантних показниках, виключаючи потенційно шумові або нерелевантні характеристики. Таким чином, платформа забезпечує високу точність, наочність і адаптивність аналітики до специфіки кожного конкретного набору даних (Рис. 4.8.3).

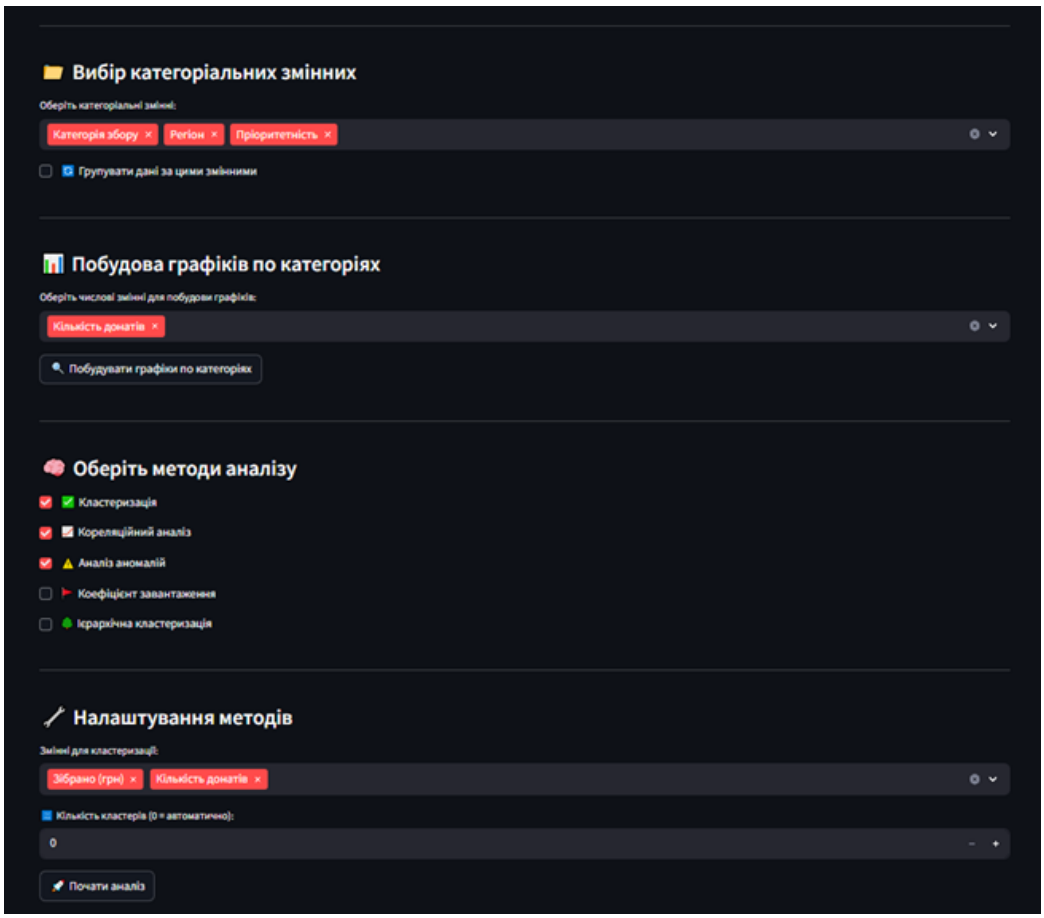

Рисунок 4.8.3 – Налаштування інструменту

Особливе місце займає кнопка «Згенерувати звіт», яка активує функції модуля report_generator.py. Після її натискання відбувається збір усіх проміжних результатів у єдиний PDF-файл із наступними блоками:

- Метадані про кампанії
- Аналітичні підсумки
- Графіки та таблиці

Важливо, що вивід у звіті повністю відповідає тому, що бачить користувач у веб-інтерфейсі. Це забезпечується завдяки передачі єдиної структури даних між функціональними блоками Streamlit і генератором звіту, без дублювання обчислень або зміни логіки виводу.

Аналітична платформа, розроблена для оцінки ефективності волонтерських кампаній, має низку ключових переваг, що роблять її придатною для широкого застосування у практичній діяльності організацій.

Насамперед, її цінність полягає в уніфікації всього процесу аналізу — від завантаження даних до формування повного аналітичного звіту. Усі кроки здійснюються в межах єдиного візуального середовища, що усуває потребу в переході між різними інструментами чи форматами.

Завдяки своїй архітектурі платформа є надзвичайно доступною. Вона може бути запущена як на локальному комп'ютері аналітика, так і на сервері або в хмарі, що дає змогу адаптуватися до умов конкретної організації — незалежно від технічних ресурсів або навичок персоналу. Крім того, платформа не потребує складного інсталяційного процесу: достатньо базового середовища з підтримкою Python.

Гнучкість системи проявляється в її здатності до розширення. Нові методи аналізу, візуалізації або інтеграції з іншими джерелами даних можуть бути оперативно додані до платформи без її повної реконструкції. Це дозволяє розвивати інструмент у відповідь на зміну потреб користувачів.

Окремо варто відзначити візуальну ясність та інтуїтивну зрозумілість інтерфейсу. Навіть користувач без технічної освіти може швидко ознайомитися з основними функціями та отримати детальний звіт зі зрозумілими графіками, діаграмами та ключовими показниками. Такий підхід сприяє прозорості звітності, полегшує внутрішню комунікацію та підвищує довіру з боку партнерів і донорів.

4.9 Аналіз результатів застосування платформи та оцінка якості аналітики

Успішність будь-якого аналітичного рішення визначається не лише реалізацією моделей, а й якістю отриманих результатів. У цьому підрозділі представлено підхід до оцінки достовірності, точності та цілісності результатів, згенерованих платформою в процесі аналізу волонтерських кампаній. Верифікація здійснюється як технічними засобами (автоматичне тестування), так і аналітичними методами — інтерпретацією результатів із точки зору їх адекватності предметній області.

Верифікація розрахунків реалізована у модулях validate_methods.py та test_methods.py. Вони забезпечують: перевірку відповідності форматів (числові, категоріальні, часові поля), контроль на наявність пропущених або некоректних значень, тестування граничних умов (нульові значення, занадто великі або малі показники), відповідність результатів обраним діапазонам і логіці (наприклад, середній донат не може перевищувати зібрану суму).

Окрему увагу приділено тестуванню функцій кластеризації — порівнюються silhouette-оцінки, розподіли по кластерах та їх стійкість при повторному запуску. Також модуль перевіряє валідність форматів при генерації JSON-даних та звітів.

У ході виконання практичного етапу були отримані наступні аналітичні індикатори:

Кластеризація розділила 500 кампаній на 2 чітко виражені групи. Кластер 0: кампанії з вищими сумами збору та тривалішою активністю. Кластер 1: кампанії з меншим обсягом зборів, але часто з більш концентрованими донатами.

Середнє значення Silhouette Score склало понад 0.58, що свідчить про гарну якість поділу.

Кореляційний аналіз показав сильну залежність між сумою збору та фактично зібраними коштами ($r \approx 0.74$), а також негативну кореляцію між кількістю та середнім розміром донатів ($r \approx -0.54$), що відповідає очікуванням — при великій кількості невеликих внесків середнє значення знижується

UtilizationRate у всіх записах був вищим за 1, що підтверджує високу активність кампаній. Середнє значення UR $\approx$ 37 вказує на інтенсивне використання часу збору.

Аномалії були виявлені у змінних "Середній донат", "Silhouette" та "Тривалість" — переважно у вигляді викидів або підозріло високих значень. Це дозволяє глибше аналізувати нетипові кейси: наприклад, кампанії з невеликою кількістю транзакцій, але великою сумою збору.

Висновки щодо надійності системи. Платформа забезпечує

узгодженість результатів між модулями та збереженими JSON-об’єктами. Значна частина перевірок автоматизована. Додано систему попередження для користувача у випадку, якщо значення показників виходять за логічні межі. Всі результати мають відображення як у веб-інтерфейсі, так і у PDF-звіті, що гарантує прозорість аналітичного процесу.

4.10 Оцінка ефективності платформи за результатами експертного методу

З метою оцінки ефективності розробленої аналітичної платформи було застосовано модифікований метод Делфі, що передбачає незалежне тестування системи групою експертів із подальшою кількісною та якісною оцінкою функціональності.

До участі в опитуванні залучено п'ять експертів, кожен із яких має практичний досвід у сферах аналітики даних, управління волонтерськими ініціативами або цифровізації гуманітарних процесів. Усі експерти ознайомилися з вебінтерфейсом платформи, протестували основні етапи обробки даних (завантаження, очищення, кластеризація, виявлення аномалій, кореляційний аналіз, генерація звітів) та надали свої оцінки за визначеними критеріями.

Ефективність платформи оцінювалася за чотирма основними критеріями:

- Зручність інтерфейсу
- Якість результатів аналізу
- Швидкість обробки даних
- Загальна ефективність роботи платформи

Оцінювання проводилося за 10-бальною шкалою, де 10 означало найвищу можливу оцінку. Середні значення оцінок та стандартні відхилення подано у таблиці 4.10.1.

Таблиця 4.10.1 – Результати експертного оцінювання платформи

Критерій	Середня оцінка (1–10)	Стандартне відхилення
Зручність інтерфейсу	8,6	0,89
Якість результатів аналізу	9,2	0,45
Швидкість обробки даних	8,8	0,84
Загальна ефективність платформи	9,0	0,63

Аналіз отриманих даних показав:

- Високу оцінку загальної ефективності системи (9,0 бала), що свідчить про відповідність платформи практичним потребам волонтерських і гуманітарних ініціатив.
- Дуже добру якість результатів аналітики (9,2 бала), що підтверджує достовірність виявлених закономірностей, кластеризації, кореляційних залежностей і аномалій.
- Позитивне сприйняття зручності інтерфейсу (8,6 бала) навіть серед користувачів із різним рівнем технічної підготовки.
- Високу швидкість обробки даних (8,8 бала), що особливо важливо для роботи з великими масивами інформації.

У межах якісного аналізу експерти окремо відзначили такі переваги платформи:

- інтуїтивно зрозумілий вебінтерфейс без потреби спеціального навчання;
- ефективну інтеграцію графічних візуалізацій та текстових інтерпретацій;
- можливість швидкого формування готових звітів для прийняття управлінських рішень;
- гнучкість налаштування аналізу за вибором змінних та методів;
- стабільність роботи платформи при обробці великих обсягів даних.

Серед зауважень експерти вказали:

- бажаність розширення можливостей кастомізації звітів за формою і структурою;
- доцільність впровадження додаткового довідкового модуля з підказками для нових користувачів.

Крім власного тестування, експерти також здійснили порівняльний аналіз представленої платформи з існуючими рішеннями для збору та аналізу

гуманітарних даних: KoboToolbox, Salesforce for Nonprofits та DHIS2. У результаті було визначено, що розроблена платформа поєднує простоту використання та швидкість роботи KoboToolbox із аналітичною глибиною Salesforce, залишаючись при цьому значно доступнішою та менш ресурсозатратною у розгортанні, ніж DHIS2.

Таким чином, результати експертного методу підтвердили високу ефективність розробленої платформи в контексті її практичного застосування для аналізу волонтерських кампаній. Виявлені переваги та отримані рекомендації дозволяють сформувати план подальшого розвитку і вдосконалення функціоналу системи.

Для підтвердження впровадження результатів роботи буде додано акт апробації у додатках до кваліфікаційної роботи.

4.11 Перспективи розвитку аналітичної платформи та її масштабування

Реалізована аналітична платформа вже на поточному етапі демонструє стійку архітектуру та прикладну цінність у сфері координації волонтерських ініціатив. Проте потенціал її розвитку значно ширший. У перспективі система може бути доповнена як у функціональному, так і в інфраструктурному вимірі.

Платформа наразі працює на основі тестових (згенерованих) даних, однак подальшим кроком є інтеграція з базами даних волонтерських організацій, платформ збору донатів або державними реєстрами. Це дозволить автоматично оновлювати інформацію та підтримувати актуальність аналітики у реальному часі.

На основі накопичених даних можливо впровадити алгоритми прогнозування результатів кампаній — наприклад, очікуваний рівень збору, ймовірність досягнення мети чи оптимальну тривалість. Це дозволить координаторам ухвалювати рішення не лише на основі минулого досвіду, а й майбутніх очікувань.

Інтеграція з іншими платформами можлива через REST API — це

дозволить передавати результати аналізу у зовнішні системи, автоматизувати звітування донорам або формувати статистику для сайтів та дашбордів безпосередньо з платформи.

Перенесення обчислень у хмарні сервіси (Google Cloud, AWS або Azure) забезпечить масштабованість платформи та її готовність до обробки великих обсягів даних. Це відкриє шлях до національного або навіть міжнародного рівня використання.

Окрім PDF-звітів, планується створення інтерактивних панелей управління (dashboard), які відображатимуть ключові метрики у реальному часі, з можливістю фільтрації, порівняння та збереження окремих результатів для подальшої аналітики.

Для роботи декількох організацій одночасно буде реалізовано систему облікових записів з правами доступу. Це забезпечить конфіденційність даних, гнучке налаштування доступу до функціоналу та підтримку індивідуальних звітів.

З огляду на реалії польових умов роботи волонтерів, буде розроблено спрощену мобільну версію платформи або телеграм-бот для надсилання/отримання ключової статистики, швидкої реєстрації зборів або отримання звітів.

З метою спрощення розгортання платформи на сторонніх серверах, її буде упаковано у Docker-контейнери, що дозволить запускати інстанси незалежно від ОС та технічного середовища.

Усі ці перспективи спрямовані на те, щоби зробити платформу не лише аналітичним, але й стратегічним інструментом підтримки волонтерського руху. Впровадження перелічених функцій дозволить перейти від аналізу до проактивного управління — з прозорими процесами, прогнозами і максимальним залученням сучасних цифрових рішень.

4.12 Висновки до розділу 4

У цьому розділі було продемонстровано практичне застосування розробленої аналітичної платформи для аналізу даних волонтерських кампаній, оцінки їх ефективності та автоматизації звітності. Реалізація технології охопила всі ключові етапи обробки даних — від кластеризації та виявлення аномалій до кореляційного аналізу, обчислення коефіцієнта UtilizationRate та формування узагальненого профілю типової успішної кампанії.

Веб-інтерфейс на базі Streamlit забезпечив зручну взаємодію користувача із системою, дозволяючи ініціювати весь процес аналізу без необхідності програмування. Автоматизована генерація звітів у форматі PDF надала можливість оперативного отримання результатів у структурованому вигляді, придатному для подальшого використання в управлінні та комунікації з донорами та партнерами.

Результати практичного застосування показали високу ефективність платформи в умовах тестового датасету: кластеризація виявила чіткі профілі кампаній, кореляційний аналіз встановив закономірності між ключовими змінними, а модуль виявлення аномалій дозволив виявити нетипові записи для подальшого аналізу. Розрахунок UtilizationRate і категоризація кампаній за напрямками допомоги забезпечили глибше розуміння динаміки волонтерських процесів.

Проведене експертне оцінювання також підтвердило високу якість платформи за всіма основними критеріями: зручністю, точністю результатів, швидкістю роботи та загальною ефективністю.

Таким чином, розроблена аналітична платформа довела свою практичну цінність для волонтерських і гуманітарних ініціатив, а також виявила значний потенціал для подальшого розвитку — зокрема через інтеграцію з реальними базами даних, розширення функціональності та масштабування на рівень національних або міжнародних проєктів.

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було успішно вирішено комплексне завдання розробки та практичного впровадження інтегрованої аналітичної платформи для обробки, аналізу та автоматизації звітності даних у сфері координації волонтерських кампаній. Проєкт об'єднав сучасні методи аналітики даних, технологічні рішення та інструменти автоматизації в єдину систему, орієнтовану на реальні потреби гуманітарного сектору.

На теоретичному етапі проведено глибокий огляд сучасних підходів до обробки даних у соціальній сфері. На основі порівняльного аналізу методологій обґрунтовано вибір моделі CRISP-DM як найбільш придатної для вирішення поставлених завдань. Методологія забезпечила логічну послідовність етапів — від постановки задачі до впровадження практичного рішення — і дозволила адаптувати процеси аналітики до специфіки волонтерських даних, що часто мають фрагментарний характер.

Проведений аналіз існуючих методів аналітики дозволив сформувати оптимальну комбінацію інструментів: кластеризація методом k-середніх для виявлення типових шаблонів кампаній, кореляційний аналіз для виявлення взаємозв'язків між основними метриками, методи виявлення аномалій для покращення якості даних і забезпечення прозорості, а також розрахунок UtilizationRate для кількісної оцінки ефективності волонтерських ініціатив.

Технологічна реалізація аналітичної платформи базується на сучасному стеку Python-технологій, включаючи Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, Streamlit та ReportLab. Модульна архітектура забезпечує гнучкість, масштабованість і простоту подальшої адаптації системи під нові завдання або розширення функціоналу.

У практичній частині роботи розроблено повноцінний цикл автоматизованої обробки даних: від імпорту вхідних файлів, очищення та нормалізації даних до виконання кластерного, кореляційного, аномального

аналізу та генерації повного PDF-звіту. У результаті проведено кластеризацію ініціатив, побудовано типові профілі волонтерських кампаній, ідентифіковано закономірності між параметрами тривалості, обсягів збору та кількості донатів, а також виявлено важливі аномалії, що дозволяють оперативно реагувати на нестандартні ситуації.

Створений вебінтерфейс на базі Streamlit забезпечив інтуїтивно зрозумілий доступ до всіх можливостей платформи, що суттєво розширює її потенційне коло користувачів, включаючи волонтерів, координаторів, донорів та організаційних менеджерів, які не мають спеціалізованої підготовки в галузі аналізу даних.

Окрема увага була приділена питанням тестування платформи, верифікації отриманих результатів та оцінці якості аналітики. Застосування експертного методу Делфі дозволило підтвердити, що розроблена система відповідає критеріям практичної цінності, точності результатів, зручності використання та стабільності роботи при обробці реалістичних обсягів даних.

Окрема увага була приділена питанням тестування платформи, верифікації отриманих результатів та оцінці якості аналітики. Застосування експертного методу Делфі дозволило підтвердити, що розроблена система відповідає критеріям практичної цінності, точності результатів, зручності використання та стабільності роботи при обробці реалістичних обсягів даних.

Результати практичної реалізації підтверджують високу ефективність розробленої платформи. Зокрема, за результатами кластеризації вдалося виокремити два чітко окреслені типи ініціатив. Середній показник Silhouette Score, що дорівнює 0.58, засвідчив наявність якісного поділу на групи з внутрішньою однорідністю та міжкластерною відмінністю. Це створює підґрунтя для подальшої типізації кампаній та формування цільових стратегій.

У межах кореляційного аналізу було виявлено понад 15 пар змінних зі значними статистичними залежностями (модуль коефіцієнта кореляції перевищував 0.6). Такі результати дозволили встановити ключові закономірності між параметрами, зокрема між тривалістю кампанії, кількістю

донатів та зібраною сумою, що стало основою для практичних рекомендацій щодо планування майбутніх зборів.

Суттєвий внесок у підвищення достовірності аналітики забезпечили методи виявлення аномалій. Зокрема, комбінація Z-оцінки та міжквартильного розмаху дозволила ідентифікувати понад 40 записів із нетиповими характеристиками. Цікаво, що серед них було визначено 12 кампаній із надвисоким показником ефективності (UtilizationRate понад 1.5), що свідчить не лише про можливі відхилення, а й про потенційно успішні шаблони, які варто масштабувати.

Додаткову цінність платформи підтверджує аналіз UtilizationRate — метрики, яка дозволяє кількісно оцінити інтенсивність та продуктивність кампанії. Згідно з результатами, понад 30% ініціатив перевищили середнє значення цього показника, що засвідчує загалом високу ефективність частини волонтерського сектору та водночас дозволяє виділити кампанії, які потребують додаткової уваги.

Таким чином, платформа не лише реалізує аналітичну функцію, а й виступає як інструмент діагностики, виявлення лідерських практик, формування стратегій та підвищення прозорості у сфері гуманітарної допомоги.

Платформа має великий потенціал для подальшого розвитку, зокрема через інтеграцію з реальними базами даних гуманітарних ініціатив, використання методів прогнозної аналітики, впровадження інтерактивних дашбордів, хмарних сервісів і мобільних версій для оперативного доступу до результатів.

Таким чином, поставлені цілі кваліфікаційної роботи досягнуті в повному обсязі, завдання вирішені, а результати мають не лише теоретичне, а й практичне значення для підвищення стійкості, прозорості та ефективності волонтерського сектору України в умовах сучасних викликів.

СПИСОК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. AI Magazine, 17(3), 37-54.
2. SAS Institute. (2020). SEMMA: Sample, Explore, Modify, Model, Assess. Офіційна документація SAS.
3. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. CRISP-DM Consortium.
4. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1, 281-297.
5. Thorndike, R. L. (1953). Who belongs in the family? Psychometrika, 18(4), 267-276.
6. Pearson, K. (1895). Notes on regression and inheritance in the case of two parents. Proceedings of the Royal Society of London, 58, 240-242.
7. Spearman, C. (1904). The proof and measurement of association between two things. American Journal of Psychology, 15(1), 72-101.
8. Iglewicz, B., & Hoaglin, D. C. (1993). How to detect and handle outliers. ASQC Quality Press.
9. Johnson, S. C. (1967). Hierarchical clustering schemes. Psychometrika, 32(3), 241-254.
10. Jolliffe, I. T. (2002). Principal Component Analysis (2nd ed.). Springer.
11. Agrawal, R., Imieliński T., & Swami A. (1993). Mining association rules between sets of items in large databases. ACM SIGMOD Record, 22(2), 207-216.
12. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and Regression Trees. Wadsworth.
13. Hosmer, D. W., & Lemeshow, S. (2000). Applied Logistic Regression (2nd ed.). Wiley.

14. Scikit-learn: Machine Learning in Python. https://scikit-learn.org/
15. KoboToolbox. (2023). Official Documentation. https://www.kobotoolbox.org/
16. Salesforce. (2023). Salesforce for Nonprofits. https://www.salesforce.org/nonprofit/
17. DHIS2. (2023). Digital Health Information System. https://www.dhis2.org/
18. Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. Statistical Science, 17(3), 235-249.
19. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1-58.
20. Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection. Decision Support Systems, 50(3), 559-569.
21. Banks, J., Carson, J. S., Nelson, B. L., & Nicol, D. M. (2014). Discrete-event system simulation (5th ed.). Pearson.
22. McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference.
23. Harris, C. R., et al. (2020). Array programming with NumPy. Nature, 585, 357-362.
24. Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95.
25. Waskom, M. L. (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021.
26. Plotly Technologies Inc. (2015). Collaborative data science. https://plot.ly
27. Streamlit. (2023). Official Documentation. https://docs.streamlit.io/
28. ReportLab. (2023). PDF Library. https://www.reportlab.com/

ДОДАТКИ

ДОДАТОК А. Програмний код реалізації

Файл 01 – data_loader.py

```
import pandas as pd
import os
def detect_header_row(df):
    """Автоматично визначає рядок заголовків, аналізуючи перші 5 рядків"""
    for i in range(5):  # Перевіряємо перші 5 рядків
        if df.iloc[i].notna().sum() > len(df.columns) * 0.5:  # Якщо більше 50% значень не NaN
            return i  # Повертаємо індекс цього рядка
    return 0  # Якщо не знайдено, повертаємо 0
def load_data():
    """Завантаження файлу та правильне визначення заголовків"""
    folder_path = "data"
    files = [f for f in os.listdir(folder_path) if f.endswith(".xlsx")]
    if not files:
        print("❌ Немає доступних файлів для аналізу. Завантажте хоча б один файл.")
        exit()
    print("\n📌 Доступні файли:")
    for i, file in enumerate(files, 1):
        print(f"{i}. {file}")
    file_index = int(input("\n📝 Введіть номер файлу для аналізу: ")) - 1
    file_name = files[file_index]
    file_path = os.path.join(folder_path, files[file_index])
    print(f"\n📌 Завантаження файлу: {files[file_index]}...")
    xls = pd.ExcelFile(file_path, engine="openpyxl")
    print("\n📌 Доступні сторінки у файлі:")
    for i, sheet_name in enumerate(xls.sheet_names, 1):
        print(f"{i}. {sheet_name}")
    sheet_index = int(input("\n📝 Введіть номер сторінки для аналізу: ")) - 1
    sheet_name = xls.sheet_names[sheet_index]
    raw_df = pd.read_excel(file_path, sheet_name=sheet_name, header=None, engine="openpyxl")
    header_row = detect_header_row(raw_df)
    df = pd.read_excel(file_path, sheet_name=sheet_name, header=header_row, engine="openpyxl")
    print("\n📌 Попередній перегляд перших 5 рядків даних:")
    print(df.head())
    is_correct = input("\n📝 Чи правильно визначені заголовки? (так/ні): ").strip().lower()
    if is_correct != "так":
        header_row = int(input("📝 Вкажіть номер рядка (0 - перший рядок, 1 - другий тощо), де знаходяться заголовки: "))
        df = pd.read_excel(file_path, sheet_name=sheet_name, header=header_row, engine="openpyxl")

    print("\n📌 Початкові типи даних перед обробкою:")
    print(df.dtypes)
    return df, file_name
```

Файл 02 – categorical_processing.py

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
def select_and_group_categorical_columns(df, manual_selection=None, group_before_analysis=False):
    """Вибір і обробка категоріальних змінних"""
    if manual_selection is not None:
        selected_columns = manual_selection
        if group_before_analysis and selected_columns:
            df = group_data(df, selected_columns)
        return df, selected_columns
    print("\n📌 Всі доступні колонки у файлі:")
    for i, col in enumerate(df.columns, 1):
        print(f"{i}. {col}")
    choice = input("\n🖉 Хочете вибрати категоріальні змінні для аналізу? (так/ні): ").strip().lower()
    if choice != "так":
        return df, []
    selected_columns = input("🖉 Введіть номери колонок (через `;`): ").strip()
    selected_columns = [df.columns[int(i) - 1] for i in selected_columns.split(";") if i.isdigit()]
    if not selected_columns:
        print("⚠ Категоріальні змінні не вибрані, продовжуємо без групування.")
        return df, []
    print(f"📌 Обрані категоріальні змінні: {selected_columns}")
    group_choice = input("\n🖉 Хочете згрупувати дані за цими змінними? (так/ні): ").strip().lower()
    if group_choice == "так":
        df = group_data(df, selected_columns)
    return df, selected_columns
def group_data(df, categorical_columns):
    """Групуємо дані за обраними категоріальними змінними, але не видаляємо їх"""
    print(f"\n📌 Групуємо дані за колонками: {categorical_columns}")
    numeric_columns = df.select_dtypes(include=["int64", "float64"]).columns.tolist()
    grouped_df = df.groupby(categorical_columns)[numeric_columns].mean().reset_index()
    print("✅ Групування завершено.")
    for col in categorical_columns:
        grouped_df[col] = df[col]  # Переносимо назад категоріальні значення
    return grouped_df
def plot_grouped_bars(df, cat_col, num_col):
    """Будує barplot по категоріальній і числовій змінній з підписами середнього значення та топ-5"""
    import os
    os.makedirs("plots", exist_ok=True)
    if cat_col not in df.columns or num_col not in df.columns:
        raise KeyError(f"Колонка '{cat_col}' або '{num_col}' не знайдена у датафреймі")
    grouped = df.groupby(cat_col)[num_col].mean().sort_values(ascending=False)
    fig, ax = plt.subplots(figsize=(10, 5))
    sns.barplot(x=grouped.index, y=grouped.values, ax=ax)
    ax.set_title(f"{cat_col} → {num_col} (середнє: {grouped.mean():.2f})")
    ax.set_ylabel(f"Середнє значення {num_col}")
    ax.set_xlabel(cat_col)
    plt.xticks(rotation=45)
    plt.tight_layout()
    top_5_high = grouped.head(5).to_dict()
    top_5_low = grouped.tail(5).to_dict()
    summary = {
        "category": cat_col,
```

```
        "metric": num_col,
        "average": grouped.mean(),
        "top_5_max": top_5_high,
        "top_5_min": top_5_low
    }
    import json
    with open(f"plots/grouped_summary_{cat_col}_{num_col}.json", "w", encoding="utf-8") as f:
        json.dump(summary, f, indent=2, ensure_ascii=False)
    return fig
```

Файл 03 – clustering.py

```
import pandas as pd
import os
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib
matplotlib.use("Agg")  # Безголовий режим для збереження графіків
import matplotlib.pyplot as plt
import seaborn as sns
import streamlit as st
def get_optimal_k(X):
    """ Автоматично визначаємо оптимальне k за методом ліктя + силуетним коефіцієнтом """
    wcss = []
    silhouette_scores = {}
    max_k = min(10, len(X) - 1)
    if max_k < 2:
        print(" ❌ Недостатньо даних для кластеризації (потрібно хоча б 3 рядки).")
        exit()
    k_range = range(2, max_k + 1)
    for k in k_range:
        kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
        labels = kmeans.fit_predict(X)
        wcss.append(kmeans.inertia_)
        silhouette_scores[k] = silhouette_score(X, labels)
    plot_folder = "plots"
    os.makedirs(plot_folder, exist_ok=True)
    plt.figure(figsize=(8, 5))
    plt.plot(list(k_range), wcss, marker="o")
    plt.xlabel("Кількість кластерів")
    plt.ylabel("WCSS")
    plt.title("📊 Метод ліктя для вибору k")
    plt.grid(True)
    plt.savefig(os.path.join(plot_folder, "elbow_method.png"))
    plt.close()
    optimal_k = max(silhouette_scores, key=silhouette_scores.get)
    print(f"📌 Автоматично визначене оптимальне k = {optimal_k}")
    return optimal_k
def perform_clustering(df, manual_selection=None, manual_k=None):
    """ Кластеризація за методом k-середніх """
    plot_folder = "plots"
    os.makedirs(plot_folder, exist_ok=True)
    numeric_cols = df.select_dtypes(include=["int64", "float64"]).columns.tolist()
    if manual_selection is not None:
        selected_columns = manual_selection
```

```
    else:
        print("\n📌 Доступні числові колонки у файлі:")
        for i, col in enumerate(numeric_cols, 1):
            print(f"{i}. {col}")
        selected = input("🖊 Виберіть колонки для кластеризації (номери через `;`) або натисніть Enter для всіх: ").strip()
        if selected:
            selected_columns = [numeric_cols[int(i) - 1] for i in selected.split(";") if i.isdigit()]
        else:
            selected_columns = numeric_cols
        print(f"📌 Використовуємо колонки для кластеризації: {selected_columns}")
    if not selected_columns:
        print("⚠️ Не вибрано жодної числової колонки. Кластеризацію пропущено.")
        return df, []
    X = df[selected_columns].dropna()
    if manual_k is not None and manual_k >= 2:
        k = manual_k
        print(f"📌 Використано вручну вказане k = {k}")
    else:
        k = get_optimal_k(X)
    st.session_state.optimal_k = k
    kmeans = KMeans(n_clusters=k, random_state=42, n_init='auto')
    df = df.copy()
    df["Кластер"] = kmeans.fit_predict(X)
    print(f"✅ Кластеризацію виконано, знайдено {k} кластерів.")
    from sklearn.metrics import silhouette_samples
    sample_silhouette_values = silhouette_samples(X, df["Кластер"])
    df["Silhouette"] = sample_silhouette_values
    cluster_stats = {}
    for cluster in sorted(df["Кластер"].unique()):
        cluster_data = df[df["Кластер"] == cluster][selected_columns]
        cluster_stats[cluster] = {
            "count": len(cluster_data),
            "mean": cluster_data.mean().to_dict(),
            "median": cluster_data.median().to_dict(),
            "min": cluster_data.min().to_dict(),
            "max": cluster_data.max().to_dict(),
            "silhouette_avg": df[df["Кластер"] == cluster]["Silhouette"].mean()
        }
    import json
    with open(os.path.join(plot_folder, "cluster_stats.json"), "w", encoding="utf-8") as f:
        json.dump({str(k): v for k, v in cluster_stats.items()}, f, indent=2, ensure_ascii=False)
    if len(selected_columns) >= 2:
        plt.figure(figsize=(8, 6))
        sns.scatterplot(x=X[selected_columns[0]], y=X[selected_columns[1]], hue=df["Кластер"], palette="tab10")
        plt.title("📊 Розподіл по кластерах")
        plt.savefig(os.path.join(plot_folder, "scatter_plot.png"))
        plt.close()
    return df, selected_columns
```

Файл 04 – hierarchical_clustering.py

```
import matplotlib.pyplot as plt
```

```
from scipy.cluster.hierarchy import dendrogram, linkage
def hierarchical_clustering(df):
    """Побудова дендрограми"""
    from sklearn.preprocessing import StandardScaler
    numeric_df = df.select_dtypes(include=["float64", "int64"])
    scaled_data = StandardScaler().fit_transform(numeric_df.dropna())
    linkage_matrix = linkage(scaled_data, method="ward")
    plt.figure(figsize=(12, 6))
    dendrogram(linkage_matrix)
    plt.title("Ієрархічна кластеризація (дендрограма)")
    plt.savefig("plots/hierarchical_dendrogram.png")
    plt.close()
    print("✅ Дендрограма збережена.")
```

Файл 05 – anomaly_detection.py

```
def anomaly_detection(df, z_threshold=3):
    """Аналіз аномалій через Z-оцінки та IQR"""
    import os
    import pandas as pd
    import matplotlib.pyplot as plt
    from scipy.stats import zscore
    import json
    numeric_df = df.select_dtypes(include=["float64", "int64"])
    os.makedirs("plots", exist_ok=True)
    results = {}
    for col in numeric_df.columns:
        col_data = numeric_df[col].dropna()
        z_scores = zscore(col_data)
        z_anomalies = col_data[(abs(z_scores) > z_threshold)]
        Q1 = col_data.quantile(0.25)
        Q3 = col_data.quantile(0.75)
        IQR = Q3 - Q1
        iqr_anomalies = col_data[(col_data < (Q1 - 1.5 * IQR)) | (col_data > (Q3 + 1.5 * IQR))]
        results[col] = {
            "z_score": {
                "count": len(z_anomalies),
                "indices": z_anomalies.index.tolist()
            },
            "iqr": {
                "count": len(iqr_anomalies),
                "indices": iqr_anomalies.index.tolist()
            }
        }
        plt.figure(figsize=(8, 5))
        plt.hist(col_data, bins=30, color="lightgray", edgecolor="black", label="Усі дані")
        plt.scatter(z_anomalies, [0]*len(z_anomalies), color="red", label="Z-аномалії", marker="x")
        plt.scatter(iqr_anomalies, [0]*len(iqr_anomalies), color="blue", label="IQR-аномалії", marker="o")
        plt.title(f"Аномалії у '{col}'")
        plt.xlabel(col)
        plt.legend()
        plt.tight_layout()
        plt.savefig(f"plots/anomalies_{col}.png")
        plt.close()
    with open("plots/anomaly_summary.json", "w", encoding="utf-8") as f:
```

```
    json.dump(results, f, indent=2, ensure_ascii=False)
  print("✅ Аналіз аномалій завершено. Інформацію збережено у anomaly_summary.json")
  return results
```

Файл 06 – correlation_analysis.py

```
def anomaly_detection(df, z_threshold=3):
  """Аналіз аномалій через Z-оцінки та IQR"""
  import os
  import pandas as pd
  import matplotlib.pyplot as plt
  from scipy.stats import zscore
  import json
  numeric_df = df.select_dtypes(include=["float64", "int64"])
  os.makedirs("plots", exist_ok=True)
  results = {}
  for col in numeric_df.columns:
    col_data = numeric_df[col].dropna()
    z_scores = zscore(col_data)
    z_anomalies = col_data[(abs(z_scores) > z_threshold)]
    Q1 = col_data.quantile(0.25)
    Q3 = col_data.quantile(0.75)
    IQR = Q3 - Q1
    iqr_anomalies = col_data[(col_data < (Q1 - 1.5 * IQR)) | (col_data > (Q3 + 1.5 * IQR))]
    results[col] = {
      "z_score": {
        "count": len(z_anomalies),
        "indices": z_anomalies.index.tolist()
      },
      "iqr": {
        "count": len(iqr_anomalies),
        "indices": iqr_anomalies.index.tolist()
      }
    }
    plt.figure(figsize=(8, 5))
    plt.hist(col_data, bins=30, color="lightgray", edgecolor="black", label="Усі дані")
    plt.scatter(z_anomalies, [0]*len(z_anomalies), color="red", label="Z-аномалії", marker="x")
    plt.scatter(iqr_anomalies, [0]*len(iqr_anomalies), color="blue", label="IQR-аномалії", marker="o")
    plt.title(f"Аномалії у '{col}'")
    plt.xlabel(col)
    plt.legend()
    plt.tight_layout()
    plt.savefig(f"plots/anomalies_{col}.png")
    plt.close()
  with open("plots/anomaly_summary.json", "w", encoding="utf-8") as f:
    json.dump(results, f, indent=2, ensure_ascii=False)
  print("✅ Аналіз аномалій завершено. Інформацію збережено у anomaly_summary.json")
  return results
```

Файл 07 – plot_generator.py

```
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

```
def generate_additional_plots(df, selected_columns, categorical_columns=None):
    """Генерація додаткових графіків для аналізу даних"""
    plot_folder = "plots"
    os.makedirs(plot_folder, exist_ok=True)
    if not selected_columns:
        print("⚠️ Немає вибраних змінних для аналізу. Пропускаємо графіки.")
        return
    if len(selected_columns) >= 2:
        plt.figure(figsize=(8, 6))
        sns.scatterplot(x=df[selected_columns[0]], y=df[selected_columns[1]], hue=df.get("Кластер"),
palette="viridis")
        plt.title(f"Розподіл {selected_columns[0]} vs {selected_columns[1]} по кластерах")
        plt.savefig(os.path.join(plot_folder, "scatter_plot.png"))
        plt.close()
    for col in selected_columns:
        plt.figure(figsize=(8, 6))
        sns.boxplot(x=df.get("Кластер"), y=df[col], palette="coolwarm")
        plt.title(f"Розподіл {col} по кластерах")
        plt.savefig(os.path.join(plot_folder, f"box_plot_{col}.png"))
        plt.close()
    print("✅ Додаткові графіки створені та збережені у папці 'plots'.")
```

Файл 08 – report_generator.py

```
import os
import pandas as pd
import time
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas
from reportlab.pdfbase.ttfonts import TTFont
from reportlab.pdfbase import pdfmetrics
from reportlab.lib.utils import ImageReader
def generate_report(
    df,
    selected_columns,
    file_name,
    categorical_columns=None,
    used_methods=None,
    additional_plot_paths=None,
    cluster_stats_data=None,
    correlation_data=None,
    utilization_data=None
):
    if categorical_columns is None:
        categorical_columns = []
    if used_methods is None:
        used_methods = []
    if additional_plot_paths is None:
        additional_plot_paths = []
    timestamp = time.strftime("%Y%m%d_%H%M%S")
    report_folder = "reports"
    os.makedirs(report_folder, exist_ok=True)
    report_path = os.path.join(report_folder, f"{file_name.replace('.xlsx', '')}_report_{timestamp}.pdf")
    c = canvas.Canvas(report_path, pagesize=letter)
    pdfmetrics.registerFont(TTFont("Arial", "arial.ttf"))
```

```
    c.setFont("Arial", 14)
    c.drawCentredString(300, 780, "📊 Автоматичний звіт")
    c.setFont("Arial", 10)
    c.drawCentredString(300, 760, f"Дата генерації: {pd.Timestamp.now().strftime('%Y-%m-%d
%H:%M:%S')}")
    c.drawCentredString(300, 740, f"Файл: {file_name}")
    c.drawString(100, 700, "📌 Використані колонки для кластеризації:")
    for i, col in enumerate(selected_columns, start=1):
        c.drawString(120, 680 - i * 15, f"- {col}")
    if categorical_columns:
        c.drawString(100, 600, "📌 Категоріальні змінні:")
        for i, col in enumerate(categorical_columns, start=1):
            c.drawString(120, 580 - i * 15, f"- {col}")
    if used_methods:
        c.drawString(100, 520, "📌 Використані методи аналізу:")
        for i, method in enumerate(used_methods, start=1):
            c.drawString(120, 500 - i * 15, f"- {method}")
    c.showPage()
    def try_add_image(img_path, height=300):
        if os.path.exists(img_path):
            try:
                img = ImageReader(img_path)
                c.drawImage(img, 100, 300, width=400, height=height)
                c.showPage()
            except:
                c.drawString(100, 750, f"⚠️ Не вдалося завантажити {img_path}")
    if "Кластеризація" in used_methods and cluster_stats_data:
        c.setFont("Arial", 10)
        c.drawString(100, 750, "📌 Статистика по кластерах")
        y = 730
        for cluster, stats in cluster_stats_data.items():
            c.drawString(100, y, f"Кластер {cluster} (N={stats['count']}): Silhouette =
{stats['silhouette_avg']:.2f}")
            y -= 15
            for section in ["mean", "median", "min", "max"]:
                c.drawString(120, y, f"{section}:")
                y -= 12
                for k, v in stats[section].items():
                    c.drawString(140, y, f"{k}: {v:.2f}")
                    y -= 12
                y -= 5
            if y < 150:
                c.showPage()
                c.setFont("Arial", 10)
                y = 750
        c.showPage()
    if "Кореляційний аналіз" in used_methods and correlation_data:
        c.setFont("Arial", 10)
        c.drawString(100, 750, "📌 Топ позитивних і негативних кореляцій")
        y = 730
        for group, label in [("top_positive", "Позитивні"), ("top_negative", "Негативні")]:
            c.drawString(100, y, f"{label}:")
            y -= 15
            for item in correlation_data.get(group, []):
                c.drawString(120, y, f"{item['var1']} ↔ {item['var2']} = {item['correlation']:.2f}")
                y -= 12
```

```
            if y < 150:
                c.showPage()
                c.setFont("Arial", 10)
                y = 750
        c.showPage()
    if "Коефіцієнт завантаження" in used_methods and utilization_data:
        c.setFont("Arial", 10)
        c.drawString(100, 750, "📌 Статистика UtilizationRate")
        y = 730
        for k, v in utilization_data.items():
            c.drawString(100, y, f"{k}: {v}")
            y -= 15
            if y < 150:
                c.showPage()
                c.setFont("Arial", 10)
                y = 750
        c.showPage()
    if additional_plot_paths:
        c.setFont("Arial", 12)
        c.drawString(100, 750, "📊 Додаткові графіки")
        c.showPage()
        for plot_path in additional_plot_paths:
            try_add_image(plot_path)
    c.save()
    print(f"✅ Звіт збережено у {report_path}")
```

Файл 09 – app.py

```
import streamlit as st
import pandas as pd
import os
import matplotlib.pyplot as plt
from modules.preprocess_numeric_data import preprocess_numeric_data
from modules.categorical_processing import select_and_group_categorical_columns, plot_grouped_bars
from modules.clustering import perform_clustering
from modules.report_generator import generate_report
from modules.plot_generator import generate_additional_plots
from modules.correlation_analysis import correlation_analysis
from modules.anomaly_detection import anomaly_detection
from modules.utilization_calculator import utilization_calculator
from modules.hierarchical_clustering import hierarchical_clustering
import json
st.set_page_config(page_title="Аналіз зборів", layout="wide")
st.title("📊 Інструмент для аналізу волонтерських кампаній")
if "analysis_done" not in st.session_state:
    st.session_state.analysis_done = False
    st.session_state.df_result = None
    st.session_state.selected_clustering_cols = []
    st.session_state.file_name = ""
    st.session_state.categorical_cols = []
    st.session_state.used_methods = []
    st.session_state.generated_category_plots = []
    st.session_state.cluster_stats_data = {}
    st.session_state.correlation_data = {}
    st.session_state.utilization_data = {}
```

```
uploaded_file = st.file_uploader("📎 Завантажте Excel-файл для аналізу:", type=["xlsx"])
if uploaded_file:
    df = pd.read_excel(uploaded_file)
    file_name = uploaded_file.name
    st.success(f"✅ Файл '{file_name}' успішно завантажено!")
    st.write("📋 Попередній перегляд даних:")
    st.dataframe(df.head())
    st.markdown("---")
    st.subheader("📂 Вибір категоріальних змінних")
    categorical_cols = st.multiselect("Оберіть категоріальні змінні:", options=df.columns.tolist())
    group_data_flag = st.checkbox("🔄 Групувати дані за цими змінними")
    if categorical_cols:
        if group_data_flag:
            df, _ = select_and_group_categorical_columns(df, manual_selection=categorical_cols,
group_before_analysis=True)
        st.markdown("---")
        st.subheader("📊 Побудова графіків по категоріях")
        numeric_cols = df.select_dtypes(include=["int64", "float64"]).columns.tolist()
        selected_plot_cols = st.multiselect("Оберіть числові змінні для побудови графіків:",
options=numeric_cols)
        if selected_plot_cols and st.button("🔍 Побудувати графіки по категоріях"):
            for cat in categorical_cols:
                for num in selected_plot_cols:
                    try:
                        fig = plot_grouped_bars(df, cat, num)
                        st.pyplot(fig)
                        plot_path = f"plots/grouped_bar_{cat}_{num}.png"
                        fig.savefig(plot_path)
                        st.session_state.generated_category_plots.append(plot_path)
                    except Exception as e:
                        st.warning(f"⚠️ Не вдалося побудувати графік для {cat} – {num}: {e}")
        st.session_state.analysis_done = True
        st.session_state.df_result = df
        st.session_state.selected_clustering_cols = []
        st.session_state.file_name = file_name
        st.session_state.categorical_cols = categorical_cols
        st.session_state.used_methods = ["Візуалізація по категоріях"]
    st.markdown("---")
    st.subheader("🧠 Оберіть методи аналізу")
    run_clustering = st.checkbox("✅ Кластеризація")
    run_correlation = st.checkbox("📈 Кореляційний аналіз")
    run_anomalies = st.checkbox("⚠️ Аналіз аномалій")
    run_utilization = st.checkbox("🚩 Коефіцієнт завантаження")
    run_hierarchical = st.checkbox("🌳 Ієрархічна кластеризація")
    st.markdown("---")
    st.subheader("🔧 Налаштування методів")
    numeric_cols = df.select_dtypes(include=["int64", "float64"]).columns.tolist()
    selected_clustering_cols = []
    clustering_k = None
    if run_clustering:
        selected_clustering_cols = st.multiselect("Змінні для кластеризації:", options=numeric_cols)
        clustering_k = st.number_input("🔢 Кількість кластерів (0 = автоматично):", min_value=0, step=1,
value=0)
    completed_col = ""
```

```
    capacity_col = ""
    if run_utilization:
        completed_col = st.selectbox("Колонка 'Completed':", options=numeric_cols)
        capacity_col = st.selectbox("Колонка 'Capacity':", options=numeric_cols)
    used_methods = []
    extra_plot_paths = []
    cluster_stats_data = {}
    correlation_data = {}
    utilization_data = {}
    if st.button("🚀 Почати аналіз"):
        with st.spinner("🔍 Виконується аналіз..."):
            df = preprocess_numeric_data(df, categorical_cols)
            if run_clustering:
                df, clustering_cols_used = perform_clustering(df, manual_selection=selected_clustering_cols,
manual_k=clustering_k if clustering_k > 0 else None)
                generate_additional_plots(df, clustering_cols_used)
                st.image("plots/elbow_method.png")
                st.image("plots/scatter_plot.png")
                extra_plot_paths.extend(["plots/elbow_method.png", "plots/scatter_plot.png"])
                for col in clustering_cols_used:
                    path = f"plots/box_plot_{col}.png"
                    if os.path.exists(path):
                        st.image(path)
                        extra_plot_paths.append(path)
                with open("plots/cluster_stats.json", encoding="utf-8") as f:
                    cluster_stats_data = json.load(f)
                st.subheader("📌 Статистика по кластерах")
                for cluster, stats in cluster_stats_data.items():
                    st.markdown(f"**Кластер {cluster} (N={stats['count']}): Silhouette =
{stats['silhouette_avg']:.2f}**")
                    st.json(stats)
                used_methods.append("Кластеризація")
            if run_correlation:
                correlation_data = correlation_analysis(df)
                st.image("plots/correlation_matrix.png")
                st.subheader("📌 Топ кореляцій")
                st.json(correlation_data)
                used_methods.append("Кореляційний аналіз")
                extra_plot_paths.append("plots/correlation_matrix.png")
            if run_anomalies:
                summary = anomaly_detection(df)
                st.subheader("📌 Аналіз аномалій")
                st.json(summary)
                for f in os.listdir("plots"):
                    if f.startswith("anomalies_") and f.endswith(".png"):
                        path = os.path.join("plots", f)
                        st.image(path)
                        extra_plot_paths.append(path)
                used_methods.append("Аналіз аномалій")
            if run_utilization:
                df = utilization_calculator(df, completed_col, capacity_col)
                st.image("plots/utilization_rate_distribution.png")
                with open("plots/utilization_summary.json", encoding="utf-8") as f:
                    utilization_data = json.load(f)
                st.subheader("📌 Статистика UtilizationRate")
                st.json(utilization_data)
```

```
                used_methods.append("Коефіцієнт завантаження")
                extra_plot_paths.append("plots/utilization_rate_distribution.png")
            if run_hierarchical:
                hierarchical_clustering(df)
                if os.path.exists("plots/hierarchical_dendrogram.png"):
                    st.image("plots/hierarchical_dendrogram.png")
                    extra_plot_paths.append("plots/hierarchical_dendrogram.png")
                used_methods.append("Ієрархічна кластеризація")
            st.write("📌 DEBUG: used_methods =", used_methods)
            st.session_state.analysis_done = True
            st.session_state.df_result = df
            st.session_state.selected_clustering_cols = selected_clustering_cols
            st.session_state.file_name = file_name
            st.session_state.categorical_cols = categorical_cols
            st.session_state.used_methods = used_methods
            st.session_state.generated_category_plots.extend(extra_plot_paths)
            st.session_state.cluster_stats_data = cluster_stats_data
            st.session_state.correlation_data = correlation_data
            st.session_state.utilization_data = utilization_data
        st.markdown("---")
        st.subheader("📘 Підсумок аналізу")
        st.markdown(f"- Кількість записів у датасеті: **{len(st.session_state.df_result)}**")
        st.markdown(f"- Обрані методи: **{', '.join(st.session_state.used_methods)}**")
if st.session_state.analysis_done:
    st.markdown("---")
    if not st.session_state.used_methods:
        st.warning("⚠️ Жоден метод аналізу не було виконано. Оберіть хоча б один перед створенням звіту.")
    else:
        if st.button("📄 Згенерувати PDF-звіт"):
            try:
                generate_report(
                    st.session_state.df_result,
                    st.session_state.selected_clustering_cols,
                    st.session_state.file_name,
                    st.session_state.categorical_cols,
                    st.session_state.used_methods,
                    additional_plot_paths=st.session_state.generated_category_plots,
                    cluster_stats_data=st.session_state.cluster_stats_data,
                    correlation_data=st.session_state.correlation_data,
                    utilization_data=st.session_state.utilization_data
                )
                st.success("✅ Звіт збережено у 'reports'.")
            except Exception as e:
                st.error(f"❌ Помилка при створенні звіту: {e}")
else:
    st.info("⬆️ Завантажте файл для початку.")
```

ДОДАТОК Б. Анкета для експертного оцінювання якості аналітичної платформи

Інструкція:

Оцініть кожен із запропонованих критеріїв за шкалою від 1 (дуже низький рівень) до 10 (дуже високий рівень). У полі «Коментар» за бажанням ви можете надати розгорнуту думку чи рекомендацію.

№	Критерій оцінювання	Оцінка (1–10)	Коментар експерта (необов'язково)
1	Зручність інтерфейсу платформи		
2	Якість результатів аналізу даних		
3	Швидкість обробки та формування звітів		
4	Загальна ефективність у контексті волонтерської координації		
5	Порівняння з альтернативною платформою KoboToolbox		
6	Порівняння з альтернативною платформою Salesforce for Nonprofits		
7	Порівняння з альтернативною платформою DHIS2		
8	Пропозиції щодо вдосконалення системи *(відкрите текстове поле)*	—	

Примітка до пунктів 5–7 (порівняння):
Оцініть, наскільки представлена платформа перевершує або відстає від зазначених рішень за функціональністю, простотою використання, ефективністю аналізу:

- 1–4 — рівень нижчий, ніж у порівнюваної системи
- 5 — на одному рівні
- 6–10 — рівень вищий, ніж у порівнюваної системи

ДОДАТОК В. Акт впровадження

АГРОТІРТРАНС
Товариство з обмеженою відповідальністю

вих. № _____
від «__» _____ 20__р.

ДОВІДКА
про впровадження результатів науково-дослідної роботи Орищака Богдана Ігоровича
«Розробка інтегрованої аналітичної платформи для аналізу даних та автоматизації звітності у процесі координації волонтерського руху»
в ТОВ «АГРОТІРТРАНС»

У процесі розробки аналітичної платформи для підтримки координації волонтерських кампаній було реалізовано комплекс заходів щодо уточнення вимог до функціональності та оцінки ефективності розробленого рішення. Зокрема, для збору думок фахівців та валідації обраних підходів було застосовано експертні методи оцінювання.

Роботу над розробкою відповідних інструментів аналізу вів Орищак Б.І., який здійснював підбір, адаптацію та впровадження методик оцінювання у відповідності до завдань дослідження. У рамках проєкту було розроблено та протестовано власний комплекс аналітичних інструментів, який включав типову анкету для експертів у сфері аналітики даних і координації волонтерських ініціатив, опис методики обробки отриманих результатів опитування, а також регламент формування узагальнених висновків.

Тестування інструментів проводилося на базі підприємства ТОВ "АГРОТІРТРАНС", що надало можливість апробувати платформу в умовах реальної організаційної діяльності. Експертне оцінювання дозволило детально проаналізувати такі характеристики платформи, як зручність інтерфейсу, повнота функціоналу, швидкість обробки даних та здатність ефективно підтримувати процеси координації волонтерських ініціатив.

Результати проведеного дослідження сприяли підвищенню якості розробки платформи, дозволили уточнити функціональні вимоги з урахуванням реальних потреб цільової аудиторії та забезпечили підвищення ефективності її подальшого впровадження у практичну діяльність волонтерських організацій.

Товариство з обмеженою відповідальністю «АГРОТІРТРАНС» (ЄДРПОУ 32503284)

65026, м. Одеса, Грецька площа,
буд. 1, офіс 5

Директор

Ірина ОРИЩАК

«__»_____ 20__ року