Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

(назва освітнього ступеня)

на тему: «Розробка програмного забезпечення для аналізу та виявлення закономірностей у даних»

Виконала: студентка 4 курсу, групи СП-41
спеціальності 121 «Інженерія програмного забезпечення»
(шифр і назва спеціальності)

_____ Карпюк К. В.
(підпис) (прізвище та ініціали)

Керівник _____ Петрик М. Р.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ Стоянов Ю. М.
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Петрик М. Р
(підпис) (прізвище та ініціали)

Рецензент _____ _____
(підпис) (прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних технологій і програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Петрик М. Р.
(підпис) (прізвище та ініціали)
« » 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 121 «Інженерія програмного забезпечення»
(шифр і назва спеціальності)

студентці Карпюк Катерині Василівні
(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка програмного забезепечення для аналізу та виявлення закономірностей у даних»

Керівник роботи Петрик М. Р., д. ф.-м. н., професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 06 » квітня 2026 року №
2. Термін подання студентом завершеної роботи 22.06.2026
3. Вихідні дані до роботи наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. 1 Аналіз предметної області та постановка задачі
2. Проєктування архітектури системи
3. Реалізація програмного забезпечення
4. Тестування, впровадження та експлуатація
5. Безпека життєдіяльності, основи охорони праці.
Висновки
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Тема роботи. 2. Актуальність, мета, задачі дослідження
3. Існуючі технології реалізації подібних систем.
4. Функціональні та нефункціональні вимоги .5. Загальна архітектура системи.
6. Варіанти використання. 7. Компоненти програми для налаштування параметрів системи.
8. Програмні засоби та технології. 9. Інтерфейси реалізації застосунку.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці			

7. Дата видачі завдання 6 квітня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	*Розробка технічного завдання*	*6.04 – 12.04*	Виконано
2.	*Робота над першим розділом «Аналіз предметної області та постановка задачі»*	*13.04 – 26.04*	Виконано
3.	*Робота над другим розділом «Проєктування архітектури системи»*	*27.04 – 03.05*	Виконано
4.	*Робота над третім розділом «Реалізація програмного забезпечення»*	*04.05 – 10.05*	Виконано
5.	*Робота над четвертим розділом «Тестування, впровадження та експлуатація програмного забезпечення»*	*11.05 – 17.05*	Виконано
6.	*Робота над п'ятим розділом «Безпека життєдіяльності, основи охорони праці»*	*18.05 – 24.05*	Виконано
7.	*Оформлення пояснювальної записки і графічного матеріалу*	*25.05 – 7.06*	Виконано
8.	*Перевірка на академічний плагіат, перевірка керівником та консультантами*	*8.06 – 14.06*	
9.	*Попередній захист кваліфікаційної роботи бакалавра*	*15.06 – 21.06*	
10.	*Захист кваліфікаційної роботи бакалавра*		

Студентка _____ Карпюк К. В.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Петрик М. Р.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Розробка програмного забезпечення для аналізу та виявлення закономірностей у даних // Кваліфікаційна робота освітнього рівня «Бакалавр» // Карпюк Катерина Василівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СП-41 // Тернопіль, 2026 // С. 68, рис. – 19, табл. – 1, додат. – 2, бібліогр. – 24.

Ключові слова: аналіз даних, табличні дані, CSV, вебзастосунок, машинне навчання, закономірності.

Кваліфікаційна робота присвячена розробці навчального вебсередовища для аналізу табличних даних та виявлення закономірностей у них. У роботі реалізовано вебзастосунок, який дозволяє завантажувати CSV-файли, переглядати профіль набору даних, аналізувати якість даних, виконувати статистичний і кореляційний аналіз, будувати графіки, виявляти викиди, формувати рейтинг закономірностей та запускати базові моделі машинного навчання.

У першому розділі розглянуто предметну область аналізу даних, існуючі програмні рішення та сформовано вимоги до розроблюваної системи. У другому розділі спроєктовано архітектуру вебзастосунку, структуру серверної та клієнтської частин, базу даних, API та UML-діаграми. У третьому розділі описано реалізацію програмного забезпечення, зокрема серверної частини, модулів аналізу даних, машинного навчання та клієнтського інтерфейсу. У четвертому розділі наведено результати тестування системи, перевірено коректність роботи основних функцій і виконано оцінювання швидкодії на вибірках різного обсягу. У п'ятому розділі розглянуто питання безпеки життєдіяльності та основ охорони праці.

Об'єкт дослідження: процес навчального аналізу табличних даних.

Предмет дослідження: програмні засоби реалізації вебсередовища для профілювання, статистичного аналізу, виявлення закономірностей і застосування базових моделей машинного навчання до табличних даних.

ABSTRACT

Development of software for data analysis and pattern detection. Bachelor qualification thesis // Qualification work of the educational level Bachhelor // Kateryna Karpiuk // Ternopil Ivan Puliui National Technical Univercity, Computer and Information Systems and Software Engineering Faculty, Software Engineering Department, group SP-41 // Ternopil, 2026 // P. – 68, fig. – 19, tabl. – 1, annexes. – 2, references – 24.

Keywords*:* data analysis, tabular data, CSV, web application, machine learning, pattern detection.

The qualification work is devoted to the development of an educational web environment for analyzing tabular data and detecting patterns in them. The developed web application allows users to upload CSV files, view a dataset profile, analyze data quality, perform statistical and correlation analysis, build charts, detect outliers, generate a ranking of patterns, and run basic machine learning models.

The first chapter describes the subject area of data analysis, existing software solutions, and the requirements for the developed system. The second chapter presents the design of the web application architecture, the structure of the server and client parts, the database, API, and UML diagrams. The third chapter describes the implementation of the software, including the server part, data analysis modules, machine learning modules, and the client interface. The fourth chapter presents the testing results, verifies the correctness of the main functions, and evaluates the system performance on datasets of different sizes. The fifth chapter considers life safety and occupational safety issues.

Object of the research: the process of educational analysis of tabular data.

Subject of the study: software tools for implementing a web environment for profiling, statistical analysis, pattern detection, and application of basic machine learning models to tabular data.

ЗМІСТ

ВСТУП

Актуальність теми кваліфікаційної роботи зумовлена тим, що аналіз даних є важливою складовою підготовки фахівців у сфері інформаційних технологій, програмної інженерії та суміжних галузей. На практиці користувач часто працює з табличними даними у форматі CSV, однак для їх дослідження необхідно розуміти послідовність основних етапів: перегляд структури набору даних, перевірку якості, обчислення статистичних характеристик, побудову графіків, аналіз зв’язків між ознаками та використання базових моделей машинного навчання.

Існує багато програмних засобів для аналізу даних, зокрема професійні BI-системи, середовища програмування та спеціалізовані інструменти машинного навчання. Проте значна частина таких рішень орієнтована на практичне використання або професійних аналітиків. Для навчальних цілей важливо не лише отримати результат, а й бачити послідовність дій, пояснення до показників і попередження щодо можливих проблем у даних. Тому доцільним є створення програмного забезпечення, який виконує роль навчального середовища для вивчення базових підходів до аналізу табличних даних.

Метою кваліфікаційної роботи є розробка навчального програмного забезпечення для аналізу табличних даних та виявлення закономірностей у них.

Для досягнення поставленої мети необхідно виконати такі задачі:

– проаналізувати предметну область аналізу даних та існуючі програмні рішення;

– сформувати функціональні та нефункціональні вимоги до навчального програмного забезпечення;

– спроєктувати архітектуру системи, структуру бази даних, API та клієнтський інтерфейс;

– реалізувати серверну частину для завантаження, обробки та аналізу CSV файлів;

– реалізувати модулі профілювання, аналізу якості, описової статистики, кореляційного аналізу, регресії, виявлення закономірностей і машинного навчання;

– розробити вебінтерфейс для послідовного виконання етапів аналізу та відображення результатів у вигляді таблиць, графіків і пояснень;

– протестувати роботу системи на реальних і тестових наборах даних.

Об’єктом дослідження є процес навчального аналізу табличних даних.

Предметом дослідження є програмні засоби реалізації програмного забезпечення для профілювання, статистичного аналізу, виявлення закономірностей і застосування базових моделей машинного навчання до табличних даних.

У роботі використано методи описової статистики, кореляційного аналізу, лінійної регресії, класифікації та візуалізації результатів. Для реалізації програмного забезпечення застосовано Python, Flask, pandas, scikit-learn, SQLite, HTML, CSS, JavaScript і Chart.js.

Практичне значення одержаних результатів полягає у створенні програмного забезпечення, яке може використовуватися як навчальне середовище для ознайомлення з основними етапами аналізу CSV-файлів. Система дозволяє завантажити набір даних, переглянути його профіль, оцінити якість, виконати статистичний і кореляційний аналіз, побудувати прості моделі машинного навчання та отримати пояснення до результатів без написання програмного коду.

Особливість розробленої системи полягає у навчальній орієнтації програмного забезпечення. У межах одного інтерфейсу користувач може послідовно пройти основні етапи аналізу табличних даних: від завантаження CSV-файлу та перегляду структури набору до аналізу якості, статистичних показників, кореляцій, закономірностей і базових моделей машинного навчання. Додатково система формує попередження та короткі пояснення до результатів, що допомагає краще зрозуміти логіку виконуваних аналітичних дій.

Апробація результатів роботи здійснювалася шляхом підготовки та публікації тез наукової конференції за тематикою кваліфікаційної роботи. Матеріали публікації наведено в додатку А.

Кваліфікаційна робота складається зі вступу, п’яти розділів, висновків, списку використаних джерел і додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

Сучасні інформаційні системи накопичують значні обсяги даних, які потребують не лише зберігання, а й змістовної інтерпретації. Для цього використовуються статистичні методи, засоби візуалізації, алгоритми машинного навчання та програмні інструменти, що дають змогу перетворювати таблиці з даними на зрозумілі висновки. У цьому розділі розглянуто предметну область аналізу даних, існуючі програмні рішення, вимоги до розроблюваної системи та постановку задачі.

1.1 Аналіз предметної області

Аналіз даних є процесом дослідження, очищення, перетворення та інтерпретації інформації з метою отримання знань і підтримки прийняття рішень [1]. У практичних задачах користувач часто має не готову модель, а звичайний табличний файл, у якому потрібно зрозуміти структуру даних, знайти проблеми якості, оцінити зв’язки між показниками та визначити, чи можна використати дані для прогнозування.

Однією з найвідоміших методологій організації процесу аналізу даних є CRISP-DM (Cross Industry Standard Process for Data Mining) (див. рисунок 1.1), яка описує повний життєвий цикл проєкту інтелектуального аналізу даних [2]. Методологія складається з шести взаємопов’язаних етапів: розуміння предметної області, розуміння даних, підготовки даних, моделювання, оцінювання результатів та впровадження. Особливістю CRISP-DM є циклічний характер роботи, який дозволяє повертатися до попередніх етапів у разі виявлення нових закономірностей, проблем якості даних або необхідності уточнення поставлених цілей.

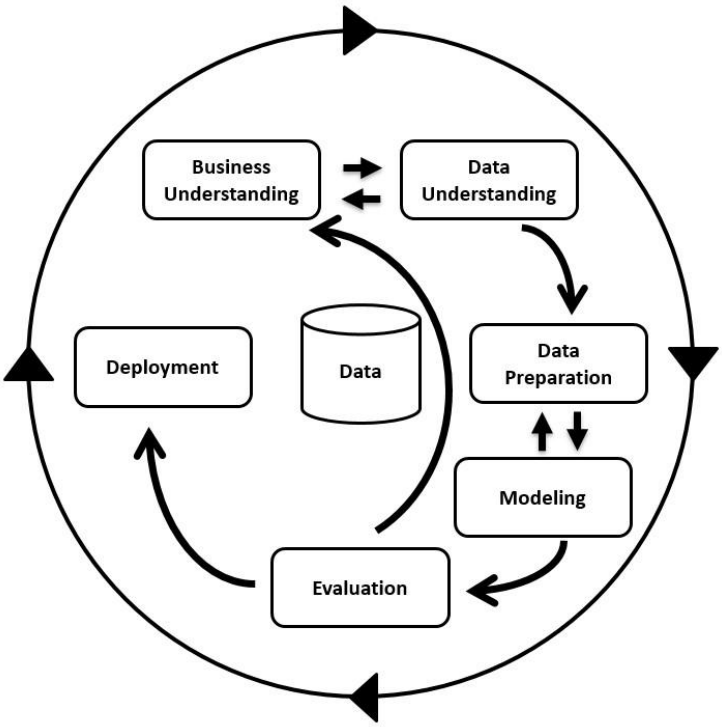


Рисунок 1.1 – Послідовність аналізу даних за логікою CRISP-DM

Оскільки розроблюване програмне забезпечення буде орієнтоване на підтримку процесу аналізу табличних даних, підходи CRISP-DM можуть бути використані як методологічна основа під час проєктування функціональності системи. Зокрема, майбутнє програмне забезпечення повинно надавати послідовний перехід від завантаження та дослідження набору даних до аналізу, побудови моделей і формування підсумкових висновків.

Важливим етапом аналізу є профілювання набору даних. Воно дозволяє швидко визначити кількість записів і колонок, типи ознак, наявність пропусків, дублікатів, константних значень та потенційних цільових колонок. Такий підхід зменшує ризик некоректного використання даних і допомагає користувачеві приймати обґрунтовані рішення щодо подальших дій.

Статистичний аналіз використовується для кількісного опису даних. До базових показників належать середнє значення, мінімум, максимум, медіана, квартилі та стандартне відхилення [3]. Ці характеристики дозволяють оцінити розподіл значень, знайти незвичні спостереження та сформувати первинне уявлення про дані.

Кореляційний аналіз застосовується для визначення сили та напряму зв’язку між числовими ознаками. Додатний коефіцієнт свідчить про прямий зв’язок, коли зі зростанням однієї ознаки зростає інша, а від’ємний – про обернений зв’язок, коли збільшення однієї ознаки пов’язане зі зменшенням іншої [4]. У системі підтримуються коефіцієнти Пірсона та Спірмена, що дозволяє аналізувати як лінійні, так і монотонні залежності.

Регресійний аналіз використовується для дослідження залежності числової цільової змінної від однієї або кількох ознак. Найпростішою є лінійна регресія, яка підбирає пряму, що найкраще описує розташування точок на графіку. Якість такої моделі оцінюється, зокрема, за коефіцієнтом детермінації R^2, який показує, яку частку варіації цільового показника пояснює побудована модель.

Машинне навчання дає змогу будувати моделі, які навчаються на прикладах і потім застосовуються до нових даних [5]. У системі планується використати базові алгоритми класифікації та регресії: логістичну регресію, лінійну регресію та Random Forest. Такий набір методів є достатнім для демонстрації основних підходів до прогнозування та порівняння результатів моделей.

1.2 Аналіз існуючих програмних рішень для аналізу даних

На ринку існує велика кількість інструментів для аналізу даних, серед яких Power BI, Tableau, Jupyter Notebook та Orange Data Mining. Ці рішення є ефективними та широко використовуються в різних сферах, однак вони мають різну цільову аудиторію, рівень складності та підхід до організації роботи користувача.

Power BI та Tableau орієнтовані насамперед на побудову інтерактивних звітів і бізнес-аналітичних панелей. Такі системи широко використовуються для візуального аналізу, підтримки прийняття рішень і представлення результатів у вигляді інформаційних панелей [6]. Водночас для користувача-початківця важливо не лише побачити графік, а й зрозуміти, що саме потрібно перевірити в наборі

даних, які проблеми якості можуть впливати на результат та як інтерпретувати отримані показники.

Jupyter Notebook є гнучким середовищем для аналізу даних, однак його використання передбачає написання програмного коду та базове володіння інструментами Python для обробки даних [7]. Це робить його ефективним інструментом для фахівців, проте менш зручним для користувачів, які хочуть виконати базовий аналіз CSV-файлу без програмування.

Orange Data Mining спрощує роботу завдяки візуальному конструюванню процесів, однак логіка аналізу все одно потребує розуміння послідовності кроків і налаштування окремих блоків [8].

Отже, існуючі програмні рішення ефективно вирішують широкий спектр задач аналізу даних та машинного навчання, однак переважно орієнтовані або на професійних аналітиків, або на користувачів, які вже мають значні базові знання з аналізу даних. Проведений аналіз показав доцільність створення програмного забезпечення, яке поєднує основні етапи первинного дослідження набору даних у межах єдиного інтерфейсу та супроводжує користувача поясненнями й рекомендаціями. Передбачається, що така система дозволить спростити початковий етап роботи з табличними даними, допоможе виявляти проблеми якості наборів даних, аналізувати статистичні залежності та отримувати базові результати моделювання без необхідності використання декількох окремих програмних продуктів або написання програмного коду.

1.3 Формування функціональних та нефункціональних вимог

На основі аналізу предметної області та існуючих рішень було сформовано функціональні та нефункціональні вимоги до програмної системи. Вимоги визначають, які можливості має надавати застосунок користувачеві та яким якісним характеристикам він повинен відповідати [9].

До функціональних вимог розроблюваної системи належать:

– реєстрація користувача та вхід до системи;

– завантаження набору даних у форматі CSV;

– автоматичне формування профілю набору даних (визначення типів ознак, пропусків, дублікатів і потенційних цільових колонок);

– формування попереджень щодо якості даних;

– підтримка базового очищення даних і створення класової колонки за правилами користувача;

– обчислення описової статистики;

– побудова гістограм, кореляцій, матриці кореляцій і лінійної регресії;

– виявлення викидів за Z-score;

– формування рейтингу закономірностей між цільовою ознакою та іншими числовими показниками;

– запуск моделей машинного навчання для класифікації та регресії;

– відображення метрик якості моделей, матриці помилок і важливості ознак;

– збереження історії дій користувача та останніх результатів аналізу.

До нефункціональних вимог належать:

– доступ до системи через сучасний веббраузер;

– зрозумілий інтерфейс для користувачів без досвіду програмування;

– обробка даних без встановлення додаткового клієнтського програмного забезпечення;

– розмежування клієнтської та серверної частин через API;

– захист доступу до основних сторінок за допомогою сесійного токена;

– можливість подальшого розширення переліку аналітичних методів;

– коректне відображення результатів у вигляді таблиць, графіків і пояснювальних повідомлень;

– збереження службових даних без накопичення повних завантажених датасетів у базі даних.

1.4 Постановка задачі та критерії оцінювання результатів

На основі проведеного аналізу предметної області, сучасних підходів до аналізу даних та існуючих програмних рішень було визначено основні завдання, які повинна вирішувати розроблювана система. Метою роботи є створення програмного забезпечення для аналізу табличних даних, яке забезпечуватиме користувачеві можливість завантаження наборів даних, дослідження їх структури, виявлення проблем якості, пошуку статистичних залежностей та використання базових методів машинного навчання через єдиний вебінтерфейс.

Основною задачею системи є спрощення початкового етапу роботи з даними для користувачів, які не мають глибоких знань у сфері програмування або аналізу даних. Розроблюване програмне забезпечення повинно забезпечувати послідовний процес дослідження набору даних: від його завантаження та профілювання до виконання статистичного аналізу, побудови моделей та формування підсумкових висновків. Особливу увагу планується приділити автоматичному виявленню потенційних проблем якості даних та наданню пояснювальної інформації щодо отриманих результатів.

Однією з важливих вимог до системи є забезпечення доступності основних інструментів аналізу через веббраузер без необхідності встановлення додаткового програмного забезпечення або написання програмного коду. Користувач повинен мати можливість виконувати основні етапи аналізу даних у межах єдиного середовища та отримувати результати у зрозумілому вигляді у формі таблиць, графіків і текстових пояснень.

У межах програмного забезпечення передбачається реалізація інструментів описової статистики, аналізу якості даних, кореляційного та регресійного аналізу, а також базових алгоритмів машинного навчання. Використання лінійної регресії, логістичної регресії та алгоритму Random Forest повинно забезпечити можливість демонстрації різних підходів до аналізу та прогнозування даних, а також порівняння отриманих результатів у межах одного програмного середовища.

Для оцінювання результатів роботи системи використовуватимуться критерії, що характеризують як якість програмного забезпечення, так і коректність виконання аналітичних операцій. До таких критеріїв належать правильність обробки вхідних даних, достовірність статистичних розрахунків, коректність результатів кореляційного та регресійного аналізу, стабільність роботи програмного забезпечення та зручність взаємодії користувача із системою [10].

Ефективність реалізованих моделей машинного навчання оцінюватиметься за допомогою стандартних метрик якості класифікації та прогнозування. Водночас основною метою використання моделей у роботі є не досягнення максимальної точності, а забезпечення можливості виконання базового моделювання та інтерпретації отриманих результатів у межах єдиного процесу аналізу даних.

Таким чином, постановка задачі визначає основні напрями розроблення програмного забезпечення для аналізу даних та встановлює критерії оцінювання його функціональності. Сформульовані вимоги є основою для подальшого проєктування архітектури системи, реалізації програмних компонентів та перевірки працездатності розробленого програмного забезпечення.

2 ПРОЄКТУВАННЯ АРХІТЕКТУРИ СИСТЕМИ

На основі сформованих функціональних та нефункціональних вимог виконується проєктування програмної системи. На цьому етапі визначаються архітектурні рішення, структура основних компонентів, засоби зберігання даних та програмні технології, необхідні для реалізації поставлених завдань. Від правильності вибору архітектури та технологічного стеку значною мірою залежить продуктивність, масштабованість, зручність супроводу та подальший розвиток програмного продукту. У цьому розділі буде розглянуто основні проєктні рішення, прийняті під час розроблення програмного забезпечення для аналізу даних.

2.1 Вибір архітектурних рішень

Архітектура програмної системи визначає принципи взаємодії між її компонентами та впливає на продуктивність, масштабованість, підтримуваність і складність подальшого розвитку програмного забезпечення. Для програмного забезпечення аналізу даних архітектурне рішення повинно забезпечувати ефективну роботу з наборами даних, підтримку аналітичних модулів, зручну взаємодію користувача із системою та можливість подальшого розширення функціональності без суттєвої зміни структури програмного забезпечення.

Під час проєктування було розглянуто декілька поширених архітектурних підходів, які можуть використовуватися для розробки вебзастосунків. Одним із найпоширеніших є багатошарова архітектура (Layered Architecture), яка передбачає поділ системи на окремі рівні представлення, бізнес-логіки та доступу до даних [11]. Основною перевагою такого підходу є чітке розмежування відповідальностей між компонентами, що спрощує підтримку та модернізацію програмного забезпечення. Водночас для невеликих вебзастосунків надмірна кількість проміжних рівнів може ускладнювати реалізацію та збільшувати складність взаємодії між окремими модулями.

Іншим поширеним підходом є мікросервісна архітектура (Microservices Architecture), у якій функціональність системи реалізується у вигляді набору незалежних сервісів [12]. Кожен сервіс відповідає за окрему предметну область та може розгортатися незалежно від інших компонентів. Такий підхід забезпечує високу масштабованість і гнучкість розвитку програмного забезпечення, однак потребує складнішої інфраструктури, додаткових механізмів обміну даними між сервісами та засобів моніторингу їхньої роботи. Для системи аналізу даних, що реалізується в межах одного вебзастосунку, використання мікросервісної архітектури є надмірним та призводить до невиправданого ускладнення реалізації.

Також було розглянуто клієнт-серверну архітектуру (Client–Server Architecture), у якій клієнтська та серверна частини виконують різні функції та взаємодіють між собою через мережеві запити [13]. У такому підході клієнт відповідає за взаємодію з користувачем і відображення результатів, тоді як сервер виконує обробку даних, реалізує бізнес-логіку та забезпечує доступ до сховищ даних. Саме цей підхід найкраще відповідає вимогам вебзастосунку для аналізу даних, оскільки дозволяє централізувати виконання ресурсоємних аналітичних операцій на сервері та мінімізувати навантаження на клієнтську частину.

З урахуванням функціональних та нефункціональних вимог до системи було обрано клієнт-серверну архітектуру (див. рисунок 2.1). Основними аргументами такого вибору стали простота реалізації, зручність розгортання, достатня продуктивність для роботи з великими наборами даних та можливість подальшого розширення функціональності без суттєвої перебудови програмного забезпечення.

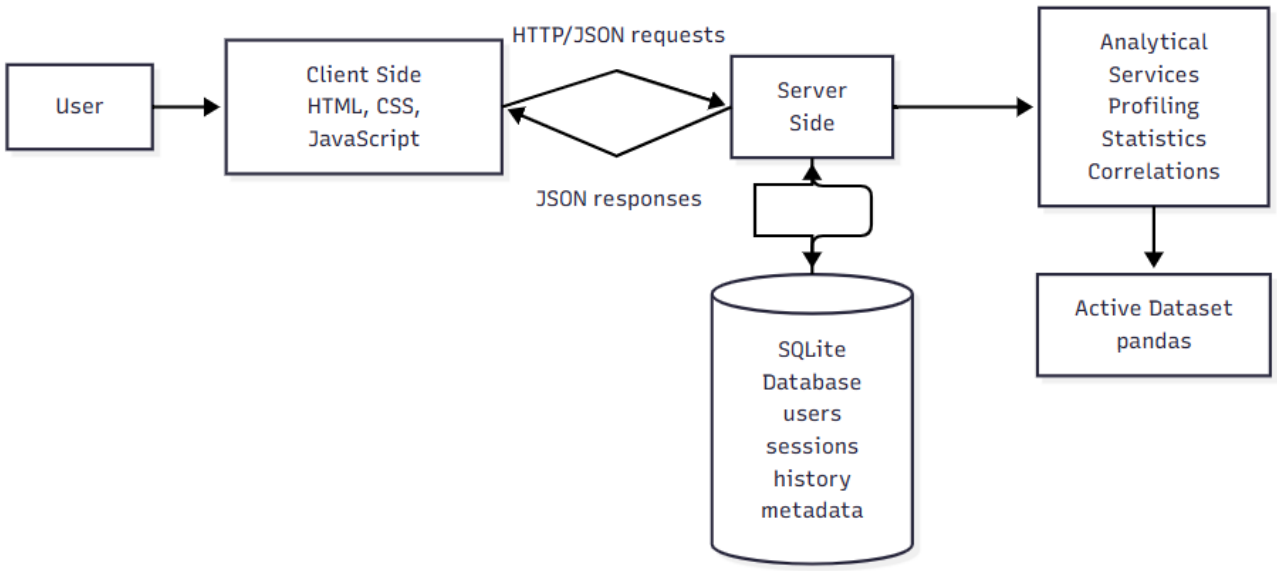


Рисунок 2.1 – Загальна архітектура вебзастосунку аналізу даних

Особливістю проєктованої системи є те, що серверна частина реалізується у вигляді єдиного застосунку з логічним поділом на окремі функціональні сервіси. Передбачається виділення компонентів для роботи з наборами даних, виконання статистичного аналізу, машинного навчання, керування користувачами та взаємодії з базою даних. Такий підхід дозволяє зберегти простоту архітектури та водночас забезпечити достатній рівень модульності програмного забезпечення.

Під час проєктування також було враховано необхідність розділення процесів аналізу даних та відображення результатів. Передбачається, що всі обчислення виконуватимуться на серверній стороні, а клієнтська частина відповідатиме лише за відображення отриманих результатів у вигляді таблиць, графіків та текстових пояснень. Це дозволить зменшити навантаження на пристрій користувача та забезпечити стабільну роботу системи незалежно від складності виконуваних аналітичних операцій.

Таким чином, клієнт-серверна архітектура найбільш повно відповідає вимогам розроблюваного вебзастосунку для аналізу даних. Вона забезпечує баланс між простотою реалізації, продуктивністю, підтримуваністю та можливістю подальшого розвитку системи.

2.2 Проєктування компонентів серверної частини та API

Серверна частина є центральним елементом розроблюваної системи, оскільки саме вона забезпечує виконання всіх операцій, пов'язаних з обробкою даних, взаємодією з користувачами та формуванням результатів аналізу. Під час проєктування особлива увага приділялася розподілу функціональності між окремими компонентами з метою підвищення зручності супроводу програмного коду та можливості подальшого розширення системи.

Основною вимогою до серверної частини є логічне розділення функціональності між окремими компонентами системи. Для досягнення цієї мети доцільно використати сервісний підхід, за якого кожен компонент буде відповідати за власну область відповідальності. Така організація дозволить уникнути надмірної концентрації логіки в одному модулі та спростить подальший супровід програмного забезпечення.

Одним із ключових компонентів системи пропонується модуль роботи з наборами даних. Його призначенням буде завантаження CSV-файлів, перевірка їхньої структури, формування профілю набору даних та підготовка інформації, необхідної для подальшого аналізу. Після завантаження набору даних система повинна формувати його профіль, який міститиме відомості про структуру таблиці, типи ознак, пропущені значення, дублікати та інші характеристики, необхідні для подальшого аналізу.

Для виконання статистичного аналізу передбачається окремий сервіс. До його функцій належатиме обчислення описових статистичних характеристик, аналіз якості даних, виявлення пропусків, дублікатів та потенційних аномальних значень. Результати роботи цього компонента повинні використовуватися для формування аналітичної інформації про набір даних та допомагати користувачеві оцінювати його якість перед подальшим аналізом.

Для реалізації пошуку закономірностей та побудови моделей машинного навчання передбачено окремий аналітичний модуль. Його завданням є виконання кореляційного аналізу, визначення взаємозв'язків між ознаками, побудова

регресійних моделей та класифікаторів. Додатковою особливістю є автоматичне формування інтерпретацій отриманих результатів, що дозволяє користувачеві швидше оцінити практичне значення виявлених залежностей.

Важливим компонентом серверної частини повинен стати модуль керування користувачами. Його призначенням буде забезпечення реєстрації нових користувачів, перевірки облікових даних під час входу до системи та підтримки активних користувацьких сесій. Наявність такого компонента дозволить персоналізувати роботу із системою та зберігати історію виконаних користувачем аналізів.

Окремий компонент доцільно виділити для взаємодії з базою даних. Його основним призначенням буде збереження та отримання інформації про користувачів, історію виконаних аналізів та інших службових даних.

Для організації взаємодії між клієнтською та серверною частинами планується використання API. Завдяки цьому всі запити від клієнтського інтерфейсу будуть надходити до сервера через стандартизований механізм обміну даними.

Схему взаємодії клієнтської частини із сервером наведено на рисунку 2.2.

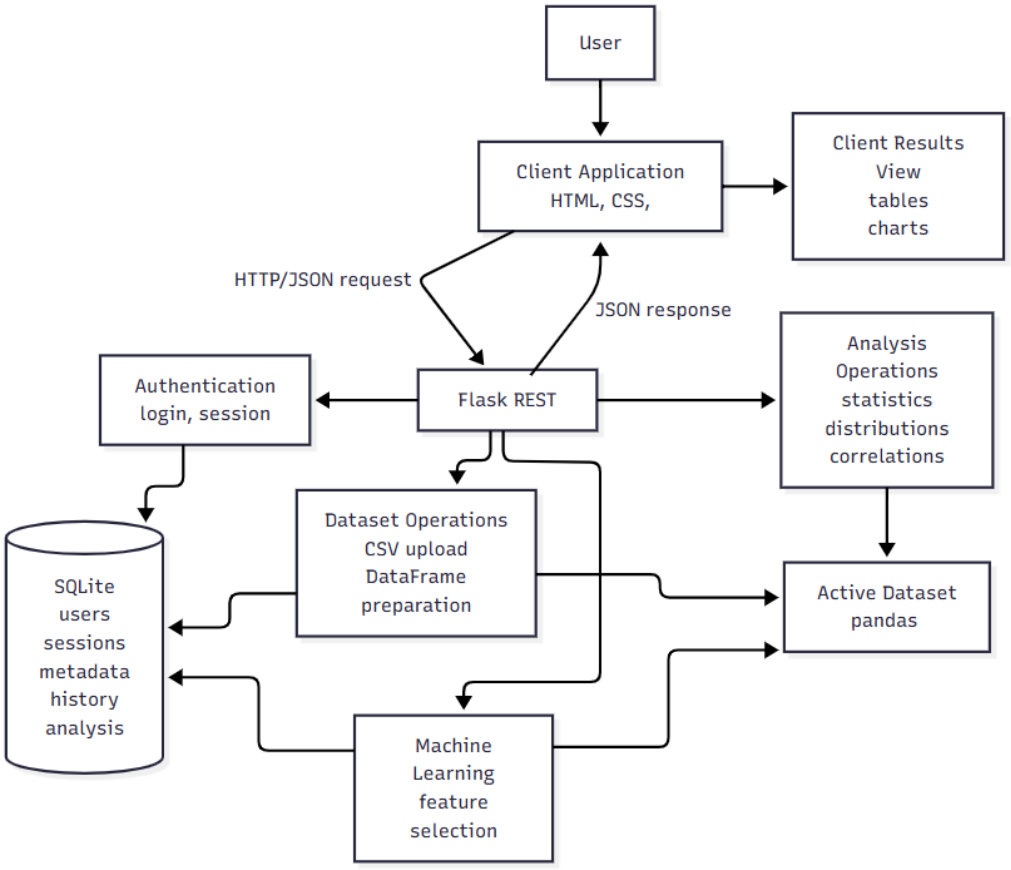


Рисунок 2.2 – Схема взаємодії клієнтської частини із сервером через API

Під час проєктування API передбачається групування запитів відповідно до їхнього призначення. Окремі маршрути планується використовувати для роботи з користувачами, завантаження даних, запуску аналітичних операцій та отримання результатів досліджень. Така організація дозволяє підтримувати зрозумілу структуру взаємодії між компонентами системи та спрощує подальший розвиток програмного продукту.

Важливою особливістю запропонованого рішення є те, що результати роботи аналітичних модулів повинні формуватися у вигляді структурованих даних, придатних для подальшого відображення у вебінтерфейсі. Передбачається, що клієнтська частина буде відповідати переважно за відображення результатів і побудову візуалізацій, тоді як основні аналітичні обчислення виконуватимуться на сервері.

Таким чином, спроєктована серверна частина поєднує набір спеціалізованих сервісів, кожен з яких відповідає за окрему функціональну область системи.

Використання API забезпечує стандартизовану взаємодію між компонентами та створює основу для ефективної роботи вебзастосунку аналізу даних.

2.3 Вибір бази даних та проєктування структури

Під час проєктування вебзастосунку було визначено, що база даних у системі повинна використовуватися не для зберігання повних завантажених наборів даних, а для збереження службової інформації. До такої інформації належать облікові записи користувачів, історія виконаних аналізів, метадані завантажених файлів і дані про активні сесії.

Для реалізації цього завдання було обрано SQLite. Це рішення є доцільним для розроблюваної системи, оскільки не потребує окремого серверного середовища, легко інтегрується із застосунками Python і забезпечує достатню продуктивність для зберігання обмеженого обсягу службових даних. Використання MySQL або PostgreSQL на цьому етапі було б надлишковим, оскільки система не орієнтована на одночасну роботу великої кількості користувачів і не передбачає довготривалого зберігання великих наборів даних у базі.

Окремо було визначено підхід до обробки CSV-файлів. Завантажені користувачем набори даних не імпортуються до реляційної бази даних, оскільки їх структура може суттєво відрізнятися. Такий підхід узгоджується з практикою проєктування систем, у яких вибір способу зберігання залежить від структури даних, характеру навантаження та вимог до обробки інформації [14].

Структура бази даних включає сутності користувача, історії аналізів, метаданих наборів даних і користувацьких сесій. Центральною сутністю є користувач, з яким пов’язані записи про виконані операції та завантажені набори даних. Загальну структуру бази даних наведено на рисунку 2.3.

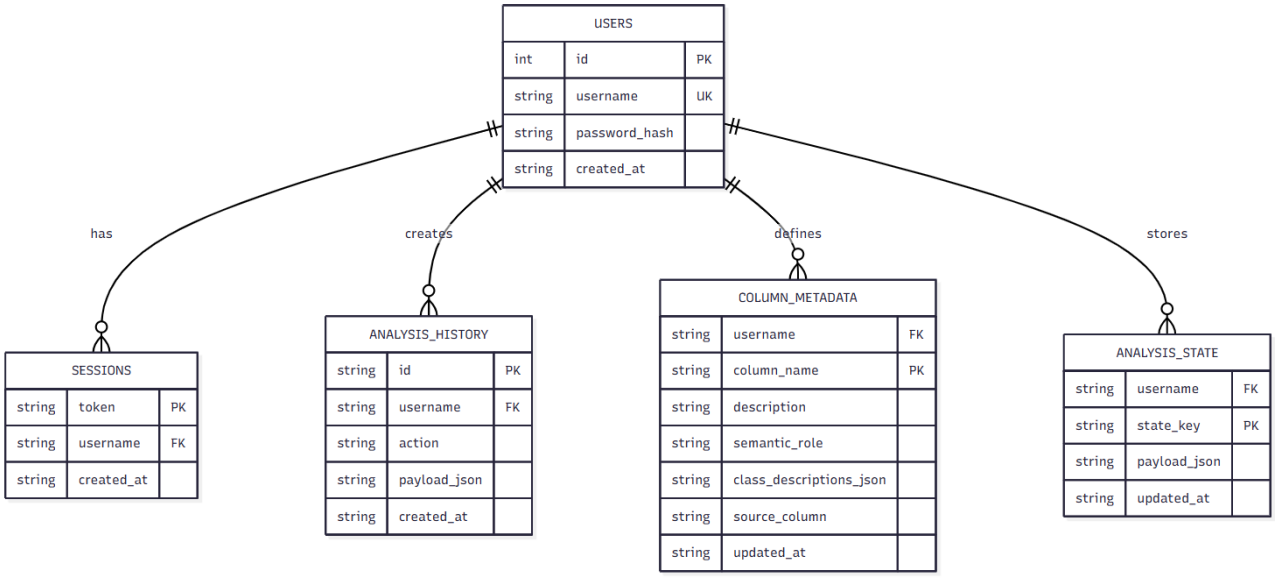


Рисунок 2.3 – ERD-діаграма структури бази даних вебзастосунку

2.4 Проєктування клієнтського інтерфейсу і структури дашбордів

Клієнтський інтерфейс проєктувався як засіб послідовної взаємодії користувача з основними функціями вебзастосунку. Його структура повинна забезпечувати простий перехід від завантаження набору даних до перегляду профілю, аналізу якості, статистичних результатів, кореляцій, закономірностей і моделей машинного навчання.

Оскільки система орієнтована на користувачів без необхідності написання програмного коду, інтерфейс має бути зрозумілим і не перевантаженим. Для цього функції розміщуються за етапами роботи з даними, а результати подаються у вигляді таблиць, графіків, інформаційних блоків і коротких пояснень.

Структура клієнтської частини включає сторінки реєстрації та входу, головну сторінку, сторінку завантаження CSV-файлу, блоки перегляду результатів аналізу та сторінку історії виконаних дій.

На відміну від класичних BI-систем, у яких основна увага приділяється одному узагальненому дашборду, у розроблюваному вебзастосунку результати організовано відповідно до логіки аналізу даних. Це дозволяє користувачеві

поступово переходити від загальної інформації про набір даних до детального аналізу окремих показників і моделей.

Таким чином, клієнтський інтерфейс спроєктовано з урахуванням послідовності виконання аналітичних операцій, зручності навігації та зрозумілого подання результатів користувачеві.

2.5 Проєктування UML-діаграм

Для формалізації структури та логіки роботи системи було використано UML-діаграми. Вони дозволяють показати функціональні можливості вебзастосунку, послідовність виконання основного сценарію роботи та взаємодію між програмними компонентами. У межах проєктування було підготовлено діаграму варіантів використання, діаграму діяльності та діаграму послідовності.

Діаграма варіантів використання відображає основні сценарії взаємодії користувача із системою. Користувач може зареєструватися, увійти до системи, завантажити набір даних, переглянути профіль, виконати аналіз якості, статистичний і кореляційний аналіз, виявити закономірності, побудувати лінійну регресію, запустити машинне навчання та переглянути історію аналізів. Діаграму варіантів використання наведено на рисунку 2.4.

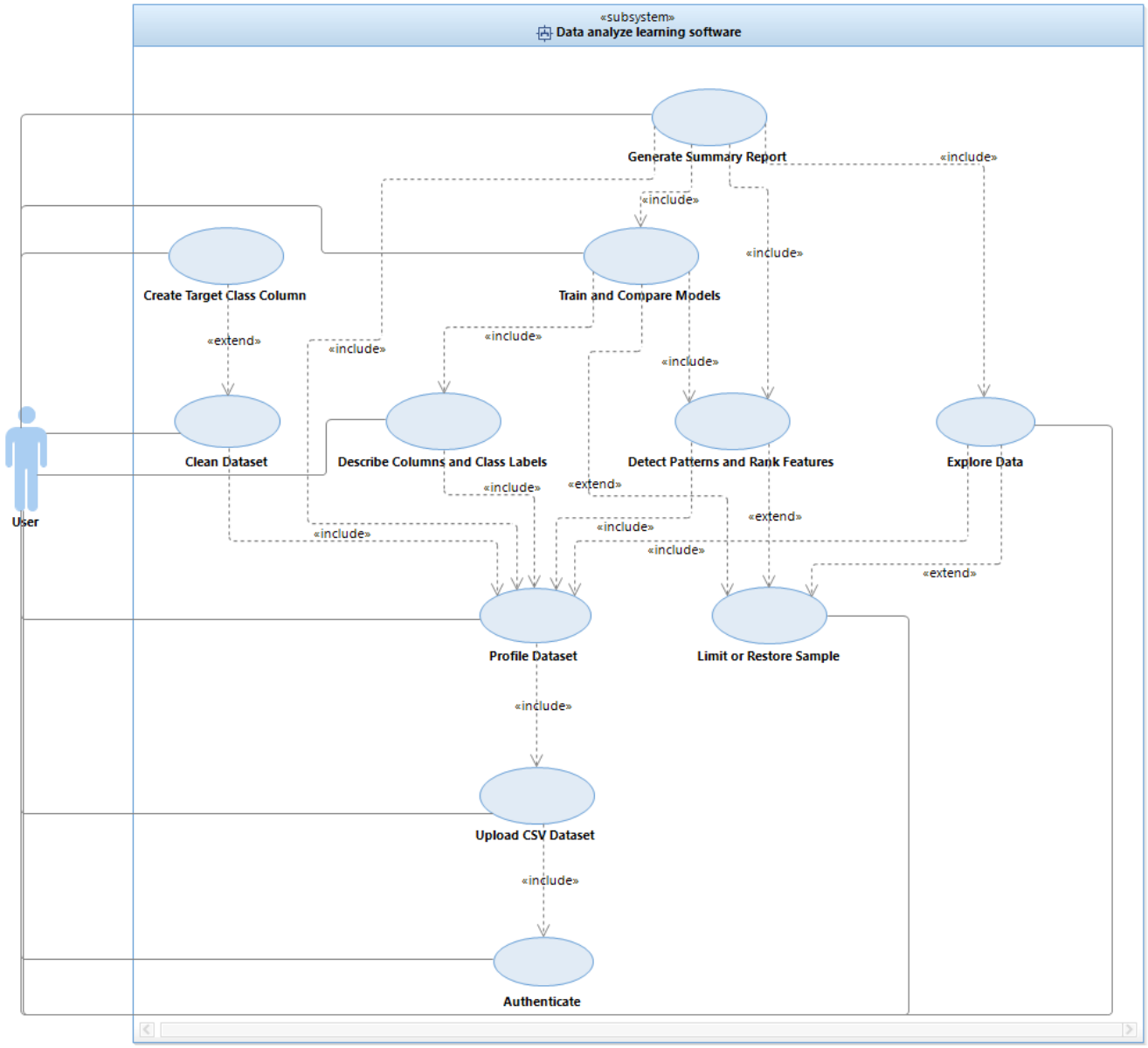


Рисунок 2.4 – Діаграма варіантів використання вебзастосунку аналізу даних

Діаграма діяльності описує основний процес роботи з набором даних. Спочатку користувач завантажує CSV-файл, після чого система перевіряє його структуру. Якщо файл некоректний, користувач отримує повідомлення про помилку. Якщо файл успішно оброблено, система формує профіль набору даних і надає доступ до подальших аналітичних операцій. Діаграму діяльності наведено на рисунку 2.5.

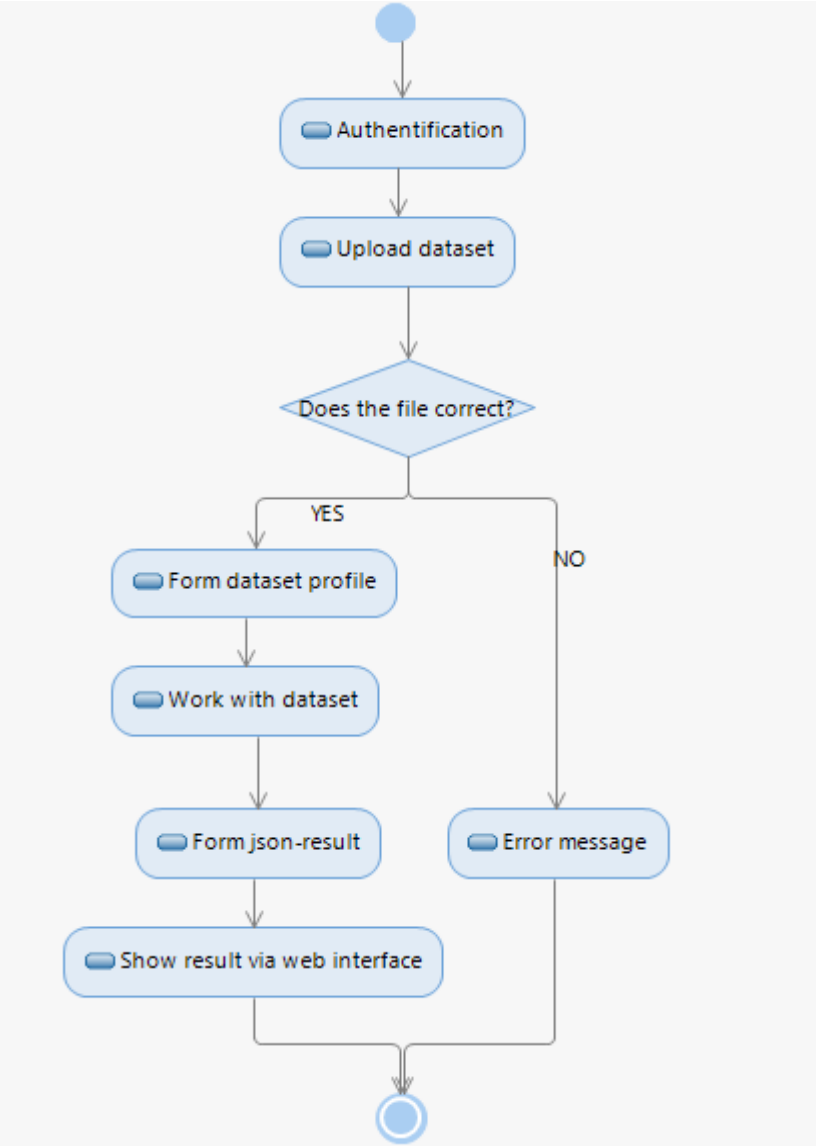


Рисунок 2.5 – Діаграма діяльності процесу аналізу даних

Діаграма послідовності показує порядок взаємодії користувача з основними частинами програмної системи під час виконання ключового сценарію роботи. На діаграмі відображено обмін повідомленнями між користувачем, клієнтським вебінтерфейсом, серверним API, сервісами аналізу даних, фоновим обробником моделювання та сховищем даних. Така діаграма дає змогу простежити, як після завантаження CSV-файлу система виконує профілювання, підготовку набору даних, запуск аналізу або моделювання, перевірку прогресу обчислень і відображення отриманих результатів. Діаграму послідовності наведено на рисунку 2.6.

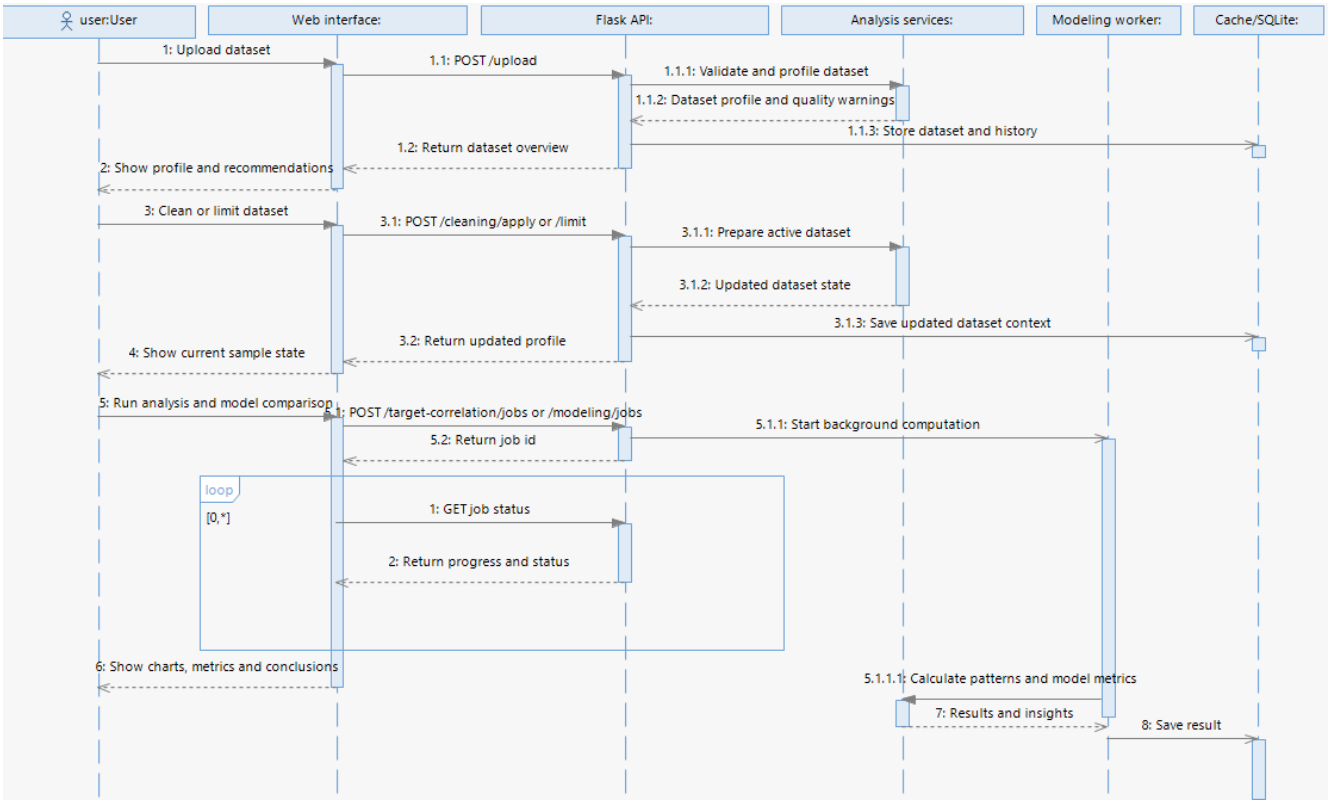


Рисунок 2.6 – Діаграма послідовності вебзастосунку аналізу даних

Таким чином, UML-діаграми доповнюють архітектурний опис системи та дозволяють представити її функціональність, логіку роботи й компонентну структуру у наочній формі.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У цьому розділі буде розглянуто практичну реалізацію вебзастосунку для аналізу даних відповідно до сформованих вимог і прийнятих проєктних рішень. Описано реалізацію архітектури та серверної частини системи, підсистеми профілювання й аналізу даних, модулів виявлення закономірностей і машинного навчання, а також клієнтської частини та засобів візуалізації результатів.

3.1 Реалізація архітектури та серверної частини системи

Серверна частина вебзастосунку реалізована як центральний рівень системи, який забезпечує приймання запитів від клієнтського інтерфейсу, обробку завантажених наборів даних, запуск аналітичних операцій та повернення результатів у структурованому вигляді. Для реалізації серверної логіки використано Flask, що дало змогу створити набір API-маршрутів для взаємодії між вебінтерфейсом, аналітичними модулями та базою даних [15].

У результаті реалізації серверна частина виконує роль координатора між усіма основними підсистемами вебзастосунку. Після отримання запиту від клієнта сервер визначає тип операції, перевіряє вхідні дані, передає виконання відповідному сервісу та формує відповідь для подальшого відображення у браузері. Такий підхід дозволив відокремити логіку обробки даних від клієнтської частини та забезпечити стабільну взаємодію між модулями системи.

Структура серверної частини реалізована з урахуванням функціонального поділу програмного коду. Окремі модулі відповідають за маршрути API, роботу з наборами даних, статистичний аналіз, машинне навчання, автентифікацію користувачів, збереження історії та взаємодію з базою даних. Завдяки такій організації вдалося уникнути надмірної концентрації логіки в одному файлі та спростити подальшу підтримку коду. Загальну структуру файлів реалізованої серверної частини наведено на рисунку 3.1.

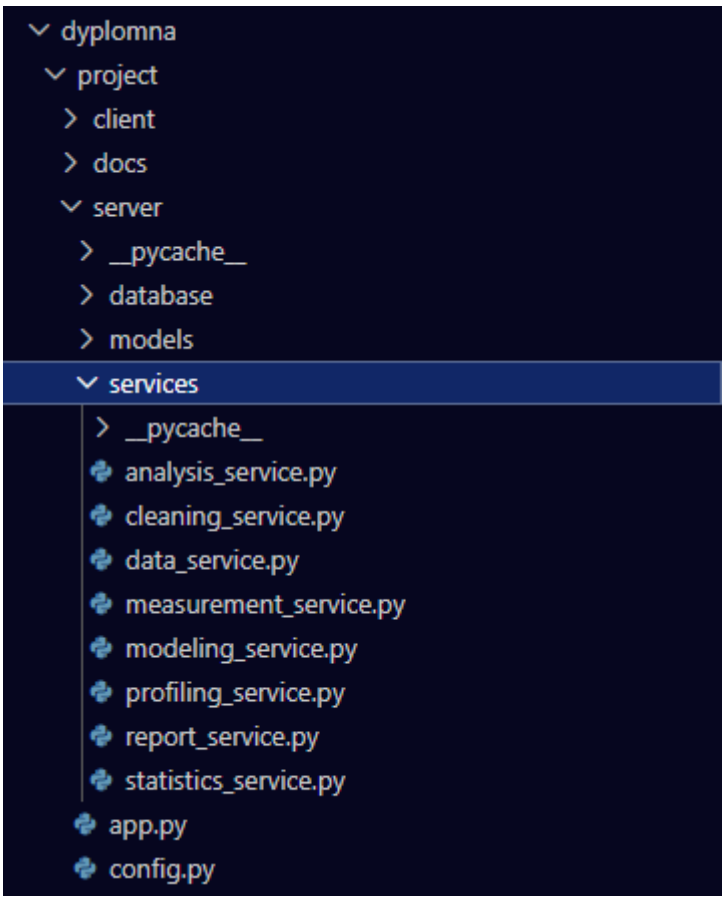


Рисунок 3.1 – Структура файлів серверної частини вебзастосунку

Основним результатом реалізації серверної частини стала можливість виконання повного циклу роботи з набором даних після його завантаження. Користувач передає CSV-файл через вебінтерфейс, після чого сервер зчитує його за допомогою бібліотеки pandas та формує активний DataFrame. DataFrame є зручною табличною структурою для обробки, фільтрації та статистичного аналізу даних у Python [16]. Саме ця структура використовується під час подальшого профілювання, аналізу якості, статистичних обчислень, кореляційного аналізу, виявлення закономірностей і запуску моделей машинного навчання.

Активний набір даних тимчасово зберігається в оперативній пам’яті сервера. Це дозволяє повторно використовувати вже завантажений DataFrame для різних аналітичних операцій без необхідності повторного зчитування файлу. У результаті користувач може послідовно виконувати кілька видів аналізу над одним набором даних, а сервер швидко передає ці дані до потрібних сервісів. При цьому сам CSV-

файл не перетворюється на окрему таблицю бази даних, оскільки структура завантажуваних наборів може суттєво відрізнятися.

API серверної частини реалізовано у вигляді окремих маршрутів, кожен з яких відповідає за певну функцію системи. Зокрема, передбачено маршрути для завантаження файлу, отримання профілю набору даних, аналізу якості, розрахунку статистичних показників, побудови кореляцій, запуску регресійного аналізу, виконання машинного навчання та перегляду історії дій користувача. У результаті клієнтська частина може звертатися до потрібної функції через окремий запит і отримувати відповідь у форматі, придатному для подальшого відображення.

Для обміну даними між сервером і клієнтською частиною використовуються структуровані відповіді, які містять числові результати, списки значень, параметри аналізу, службові повідомлення та текстові пояснення. Сервер не формує готові зображення графіків, а повертає дані, необхідні для їх побудови на клієнтській стороні. Завдяки цьому обчислювальна логіка залишається на сервері, а візуалізація результатів виконується у браузері засобами клієнтської частини.

Під час реалізації було також забезпечено коректну взаємодію між клієнтською та серверною частинами через HTTP-запити. Для цього налаштовано обробку запитів до API, передавання параметрів аналізу та повернення відповідей у форматі JSON. Це дало змогу організувати роботу вебінтерфейсу без перезавантаження всієї сторінки під час виконання окремих операцій аналізу.

Окрему роль у реалізації серверної частини відіграє взаємодія з базою даних SQLite. База даних використовується не для зберігання повних наборів даних, а для службової інформації, необхідної для роботи системи. Зокрема, у ній зберігаються облікові записи користувачів, інформація про виконані дії, параметри аналізів та історія роботи із системою. Такий підхід дозволив зберегти компактну структуру бази даних і водночас забезпечити персоналізацію роботи користувача.

У межах серверної частини реалізовано базові механізми реєстрації, входу до системи та контролю доступу до захищених функцій. Ці механізми не є основною функціональною цінністю вебзастосунку, однак вони необхідні для збереження історії аналізів і розмежування доступу між користувачами. Для підвищення

безпеки паролі зберігаються не у відкритому вигляді, а у вигляді хешованих значень, а доступ до основних сторінок і API-операцій надається лише після успішної автентифікації.

Практичним результатом реалізації серверної частини стало створення єдиного програмного ядра, яке забезпечує приймання даних, їх обробку, запуск аналітичних сервісів та підготовку результатів для відображення. Серверна частина не лише виконує окремі обчислення, а й поєднує їх у послідовний процес роботи з даними: від завантаження CSV-файлу до отримання профілю, результатів аналізу, моделей машинного навчання та історії виконаних операцій.

Таким чином, реалізована серверна частина забезпечує функціональну основу вебзастосунку для аналізу даних. Використання Flask API, pandas DataFrame, SQLite та модульної організації коду дозволило створити систему, у якій сервер відповідає за основні обчислення, керування даними, контроль доступу та підготовку результатів для клієнтської частини.

3.2 Реалізація підсистеми профілювання та аналізу даних

Підсистема профілювання та аналізу даних є однією з основних функціональних частин розробленого вебзастосунку. Її призначення полягає у первинному дослідженні завантаженого набору даних, визначенні його структури, оцінюванні якості та виконанні базових статистичних операцій. Реалізація цієї підсистеми дозволила автоматизувати початковий етап роботи з табличними даними та надати користувачеві зрозумілу інформацію про стан набору ще до запуску складніших аналітичних або навчальних алгоритмів.

Після завантаження CSV-файлу серверна частина зчитує його в об’єкт DataFrame бібліотеки pandas. Надалі всі операції профілювання та аналізу виконуються саме над цією структурою даних. Такий підхід дозволяє ефективно працювати з рядками та стовпцями таблиці, визначати типи ознак, обчислювати статистичні показники та виконувати фільтрацію даних без необхідності попереднього перетворення набору в окремі таблиці бази даних.

Першим результатом роботи підсистеми є формування профілю набору даних. У ньому відображається загальна інформація про завантажену таблицю: кількість рядків, кількість стовпців, назви ознак, типи даних, кількість числових і нечислових колонок, а також загальні характеристики структури набору. Завдяки цьому користувач одразу отримує уявлення про те, з якими даними працює система та які види аналізу можуть бути застосовані до конкретного набору.

Окрему увагу під час реалізації приділено аналізу якості даних. Підсистема автоматично перевіряє наявність пропущених значень, дубльованих записів, порожніх або майже порожніх колонок, а також інших ознак, які можуть ускладнювати подальший аналіз. Для кожної виявленої проблеми система формує не лише числову характеристику, а й коротке повідомлення для користувача. Це дозволяє не просто побачити кількість проблемних значень, а й зрозуміти, чому вони можуть впливати на результати дослідження. Результат аналізу якості даних наведено на рисунку 3.2.

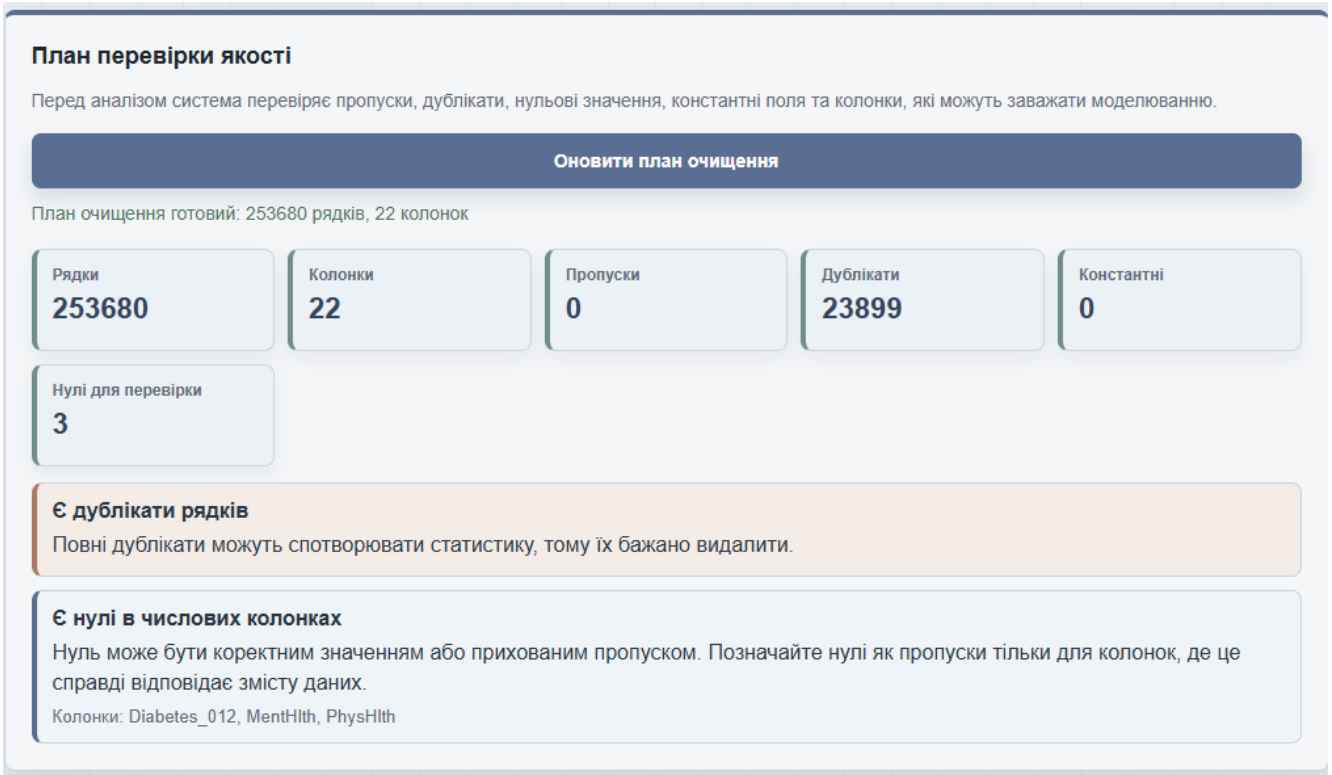
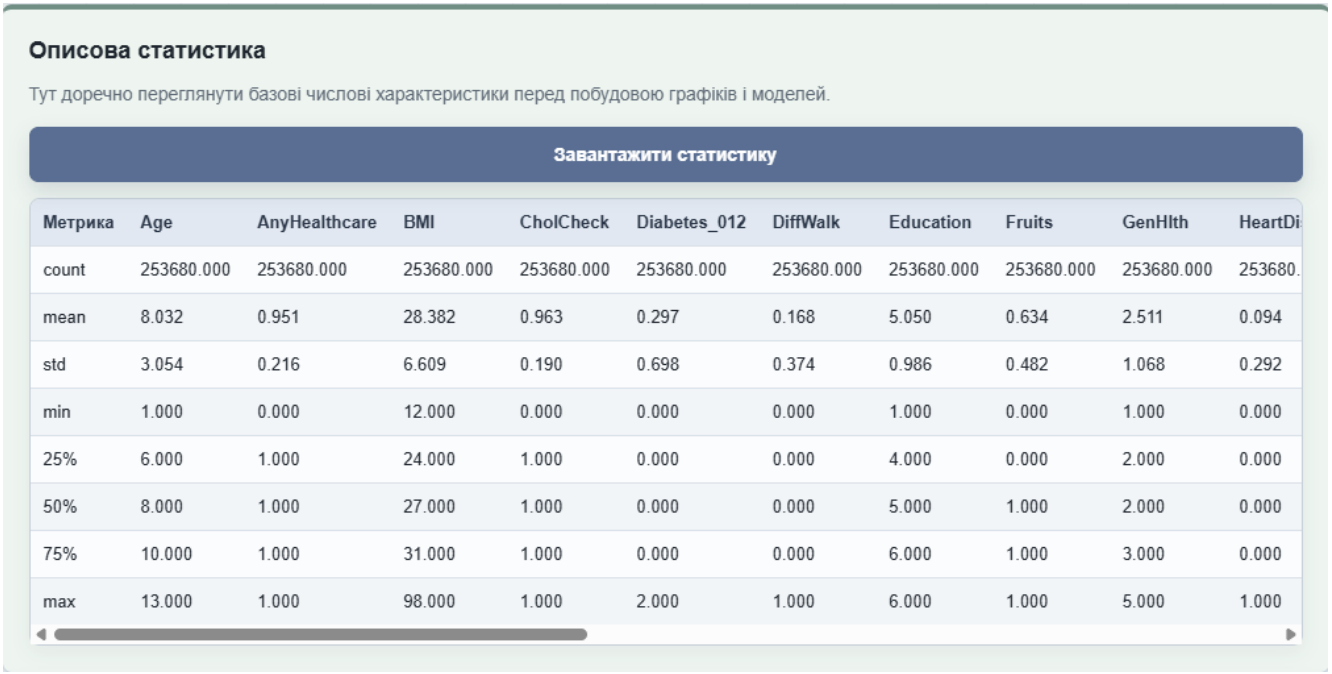


Рисунок 3.2 – Результат оцінювання якості набору даних

Реалізований механізм аналізу якості є важливим для подальшої роботи системи, оскільки неякісні або неповні дані можуть призводити до неправильних висновків. Наприклад, велика кількість пропущених значень може знижувати достовірність статистичних показників, а дублікати можуть спотворювати загальну структуру вибірки. Тому система надає користувачеві попередження ще на початковому етапі, до виконання кореляційного аналізу або запуску моделей машинного навчання.

Для числових ознак реалізовано обчислення основних статистичних характеристик. До них належать кількість значень, середнє арифметичне, стандартне відхилення, мінімальне та максимальне значення, а також квартилі. Ці показники дозволяють оцінити загальний розподіл даних, виявити можливі відхилення та попередньо зрозуміти, які ознаки можуть мати найбільше значення для подальшого дослідження. Результати статистичного аналізу відображаються у вигляді таблиці, що спрощує порівняння показників між різними колонками. Приклад відображення статистичних характеристик наведено на рисунку 3.3.



Описова статистика

Тут доречно переглянути базові числові характеристики перед побудовою графіків і моделей.

Завантажити статистику

Метрика	Age	AnyHealthcare	BMI	CholCheck	Diabetes_012	DiffWalk	Education	Fruits	GenHlth	HeartDi
count	253680.000	253680.000	253680.000	253680.000	253680.000	253680.000	253680.000	253680.000	253680.000	253680.
mean	8.032	0.951	28.382	0.963	0.297	0.168	5.050	0.634	2.511	0.094
std	3.054	0.216	6.609	0.190	0.698	0.374	0.986	0.482	1.068	0.292
min	1.000	0.000	12.000	0.000	0.000	0.000	1.000	0.000	1.000	0.000
25%	6.000	1.000	24.000	1.000	0.000	0.000	4.000	0.000	2.000	0.000
50%	8.000	1.000	27.000	1.000	0.000	0.000	5.000	1.000	2.000	0.000
75%	10.000	1.000	31.000	1.000	0.000	0.000	6.000	1.000	3.000	0.000
max	13.000	1.000	98.000	1.000	2.000	1.000	6.000	1.000	5.000	1.000

Рисунок 3.3 – Результат розрахунку статистичних характеристик

Крім табличного подання, для окремих числових ознак реалізовано можливість графічного аналізу розподілу значень. Після вибору потрібної колонки користувач може переглянути її розподіл у вигляді гістограми. Це дає змогу швидко оцінити характер даних, визначити наявність асиметрії, скупчення значень або потенційних аномалій. На відміну від звичайної таблиці, графічне представлення дозволяє швидше побачити загальну форму розподілу та зробити попередні висновки щодо особливостей ознаки.

У межах підсистеми аналізу також реалізовано виявлення аномальних значень. Для цього використовується метод Z-score, який дозволяє визначати значення, що суттєво відрізняються від середнього рівня за певною числовою ознакою. Якщо значення має надто велике відхилення від середнього, система позначає його як потенційний викид. Такий підхід не означає, що значення є помилковим, однак воно потребує додаткової уваги під час інтерпретації результатів. Приклад результатів виявлення викидів наведено на рисунку 3.4.

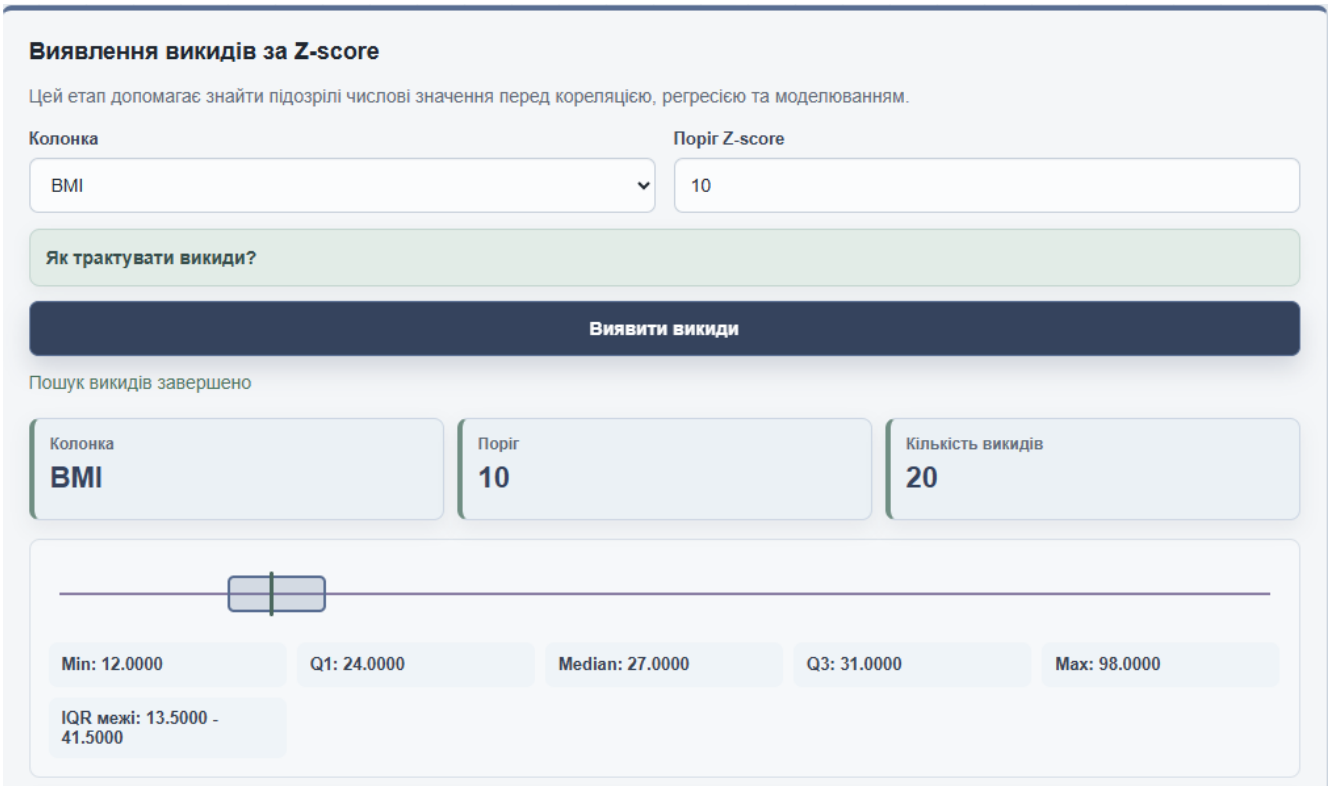


Рисунок 3.4 – Результат виявлення аномальних значень у наборі даних

Важливою частиною реалізованого аналізу є кореляційне дослідження числових ознак. Система обчислює коефіцієнти кореляції між вибраними показниками або між усіма доступними числовими колонками набору даних. Отримані результати дозволяють оцінити силу та напрям взаємозв'язку між ознаками. Додатне значення коефіцієнта свідчить про прямий зв'язок, від'ємне – про обернений, а значення, близьке до нуля, вказує на слабку лінійну залежність. Приклад результатів кореляційного аналізу наведено на рисунку 3.5.

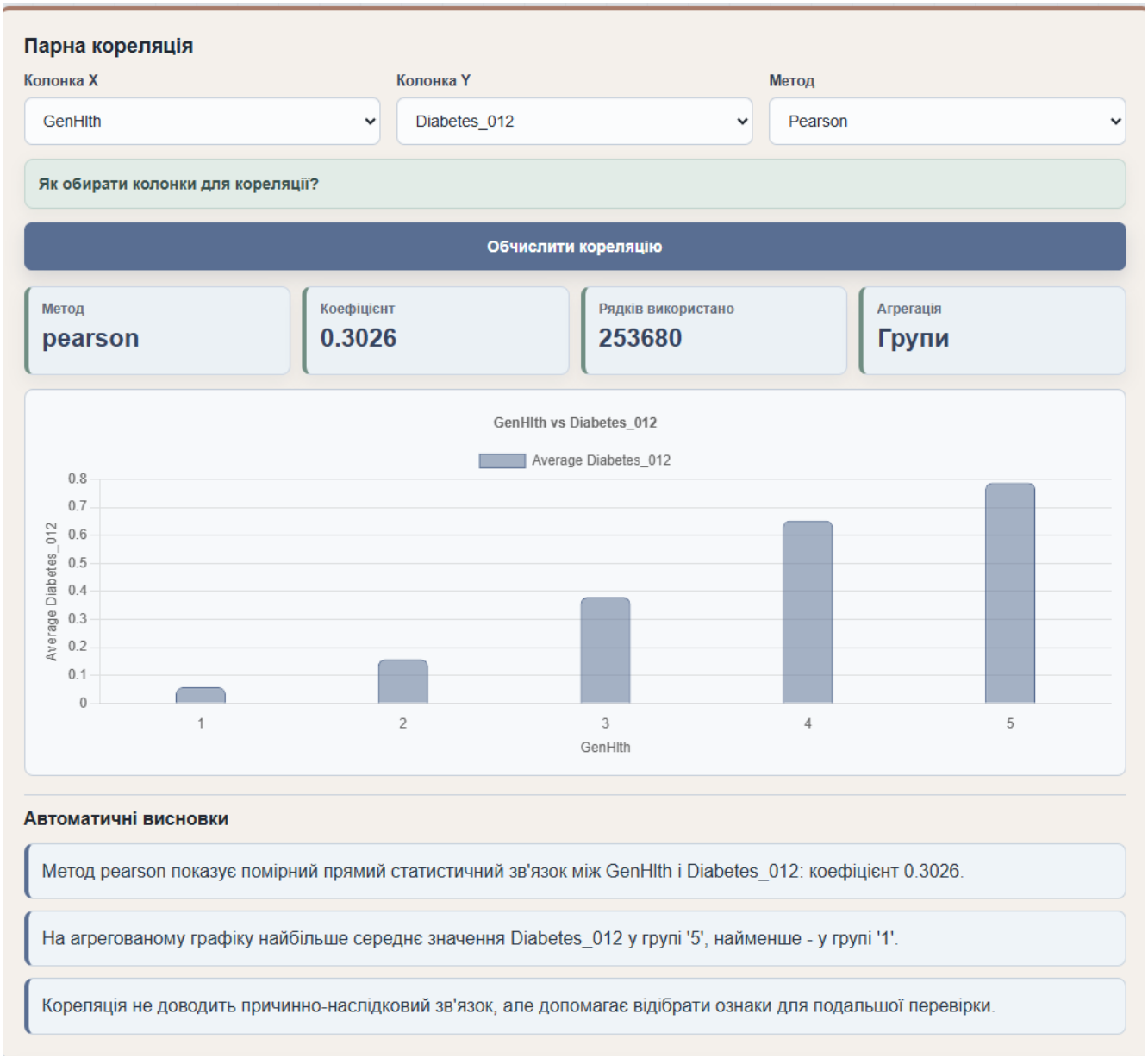


Рисунок 3.5 – Результат кореляційного аналізу числових ознак

Результати кореляційного аналізу використовуються не лише для відображення числових коефіцієнтів, а й як основа для подальшого виявлення закономірностей між ознаками. На цьому етапі підсистема аналізу даних готує необхідні розрахунки, які надалі можуть бути використані в модулі пошуку патернів та інтерпретації взаємозв’язків. Такий поділ дозволяє відокремити базові статистичні обчислення від складнішої логіки формування аналітичних висновків.

У результаті реалізації підсистеми профілювання та аналізу даних користувач отримує послідовний набір інструментів для первинного дослідження завантаженого файлу. Система автоматично формує профіль даних, оцінює їх якість, розраховує статистичні характеристики, дає змогу дослідити розподіли значень, виявити потенційні викиди та оцінити взаємозв’язки між числовими ознаками. Усі ці результати подаються у зрозумілій формі та створюють основу для подальшого пошуку закономірностей і застосування алгоритмів машинного навчання.

Таким чином, реалізована підсистема профілювання та аналізу даних забезпечує базовий аналітичний рівень вебзастосунку. Вона дозволяє перейти від простого завантаження CSV-файлу до змістовного розуміння структури, якості та основних характеристик набору даних. Це робить подальші етапи роботи із системою більш обґрунтованими та знижує ризик неправильного використання аналітичних результатів.

3.3 Реалізація модулів виявлення закономірностей та машинного навчання

Модулі виявлення закономірностей та машинного навчання реалізовано як окрему функціональну частину вебзастосунку, яка працює з активним набором даних після його завантаження та первинного аналізу. Їх основним призначенням є пошук взаємозв’язків між ознаками, побудова простих аналітичних моделей та формування результатів, які користувач може інтерпретувати без написання програмного коду. На відміну від підсистеми профілювання, яка описує загальний

стан набору даних, ці модулі спрямовані на виявлення залежностей і оцінювання можливості використання даних для прогнозування.

Першим елементом реалізованої функціональності є модуль пошуку закономірностей між цільовою ознакою та іншими числовими показниками набору даних. Користувач обирає цільову колонку, після чого система аналізує її взаємозв’язки з іншими доступними числовими ознаками. Для кожної пари показників обчислюється сила зв’язку, після чого результати впорядковуються за рівнем значущості. Завдяки цьому користувач отримує не випадковий набір статистичних коефіцієнтів, а структурований список найпомітніших залежностей у даних.

Результатом роботи модуля є рейтинг виявлених закономірностей (див. рисунок 3.6). У ньому відображаються ознаки, які мають найсильніший зв’язок із вибраним цільовим показником, а також коротке пояснення характеру цієї залежності. Такий підхід дозволяє швидко визначити, які змінні потенційно мають найбільший вплив на досліджуваний показник, і використати цю інформацію під час подальшого аналізу або побудови моделей.

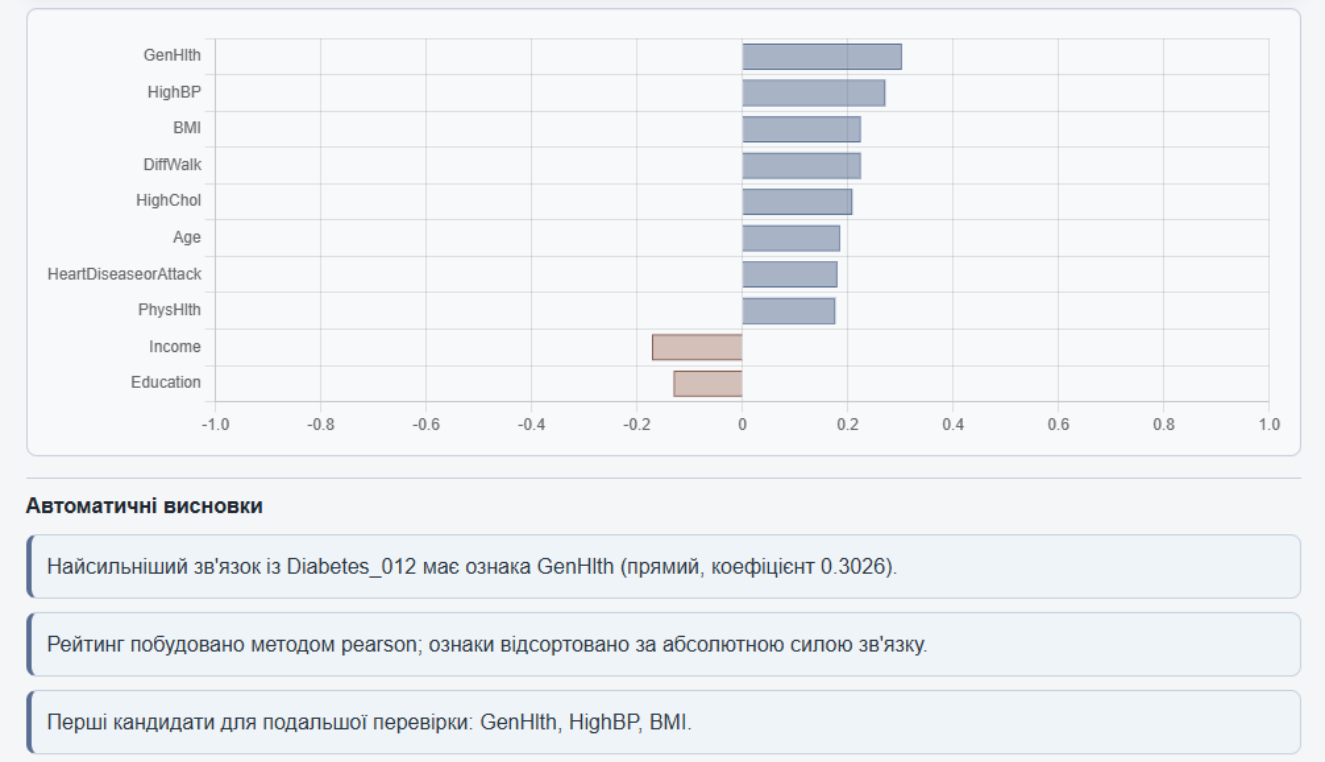


Рисунок 3.6 – Результат виявлення закономірностей між ознаками набору даних

Окрему увагу під час реалізації приділено формуванню текстових пояснень. Система не обмежується виведенням числових значень, а додатково подає інтерпретацію результатів у зрозумілій формі. Наприклад, якщо між двома числовими ознаками виявлено прямий зв’язок, користувач отримує пояснення, що збільшення одного показника може супроводжуватися збільшенням іншого. Якщо зв’язок є оберненим, система вказує на протилежний характер зміни показників. Це робить результати аналізу більш доступними для користувачів, які не мають глибокої підготовки у статистиці.

Для дослідження залежності між двома числовими показниками реалізовано модуль лінійної регресії. Він дозволяє користувачеві обрати незалежну та залежну змінні, після чого система будує регресійну модель і формує результати її роботи. У межах цього модуля визначається загальний напрям залежності між ознаками, параметри побудованої моделі та показники, які характеризують якість наближення. Результати подаються у вигляді числових значень і графічного представлення, що дозволяє оцінити, наскільки добре лінійна модель описує зв’язок між вибраними показниками. Приклад результатів побудови лінійної регресії наведено на рисунку 3.7.

Рисунок 3.7 – Результат побудови моделі лінійної регресії

Модуль лінійної регресії має допоміжне аналітичне значення, оскільки дає змогу не лише побачити наявність зв’язку, а й оцінити його форму. Якщо точки на графіку розташовані близько до регресійної прямої, це свідчить про сильніший лінійний характер залежності. Якщо ж значення суттєво розкидані, користувач може зробити висновок, що лінійна модель не повністю пояснює зміну залежної ознаки. Такий результат є корисним для попереднього дослідження даних і вибору подальших методів аналізу.

Для задач класифікації у вебзастосунку реалізовано модуль машинного навчання. Він працює з активним DataFrame та обраною користувачем цільовою ознакою. Попередні результати профілювання й аналізу допомагають користувачеві краще зрозуміти структуру даних, однак самі моделі навчаються безпосередньо на поточному наборі даних і вибраній цільовій колонці. Такий підхід забезпечує логічний зв’язок між етапами аналізу, але не створює залежності моделей від раніше сформованих графіків або текстових висновків.

У межах реалізованого модуля машинного навчання використовуються алгоритми Logistic Regression і Random Forest, які є поширеними базовими підходами для задач класифікації табличних даних [17]. Їх використання дозволяє порівняти простішу лінійну модель із ансамблевим методом, що працює на основі набору дерев рішень.

Логістична регресія використовується як базова модель класифікації, що дозволяє оцінити залежність між ознаками та ймовірністю належності об’єкта до певного класу.

Алгоритм Random Forest базується на побудові багатьох дерев рішень і поєднанні їх результатів, що дозволяє зменшити залежність моделі від окремого дерева та підвищити стійкість прогнозування. У системі цей алгоритм також використовується для формування важливості ознак.

Перед запуском моделей система виконує підготовку даних, необхідну для навчання. Зокрема, визначаються ознаки, які можуть бути використані як вхідні параметри, обробляються доступні числові значення та формується цільова змінна. Після цього набір даних розподіляється на навчальну та тестову частини. Навчальна частина використовується для побудови моделі, а тестова – для перевірки її роботи на даних, які не використовувалися під час навчання. Це дозволяє оцінити не лише здатність моделі запам’ятовувати дані, а й її придатність до узагальнення.

Після завершення навчання система формує набір метрик якості моделі. До них належать точність класифікації, збалансована точність, macro F1 та weighted F1. Використання кількох метрик є важливим, оскільки звичайна точність не завжди достатньо об’єктивно відображає якість моделі, особливо якщо класи в наборі даних представлені нерівномірно. Збалансована точність і F1-метрики дозволяють краще оцінити якість класифікації для різних класів та уникнути надто спрощеної інтерпретації результатів. Приклад відображення метрик якості моделей наведено на рисунку 3.8.

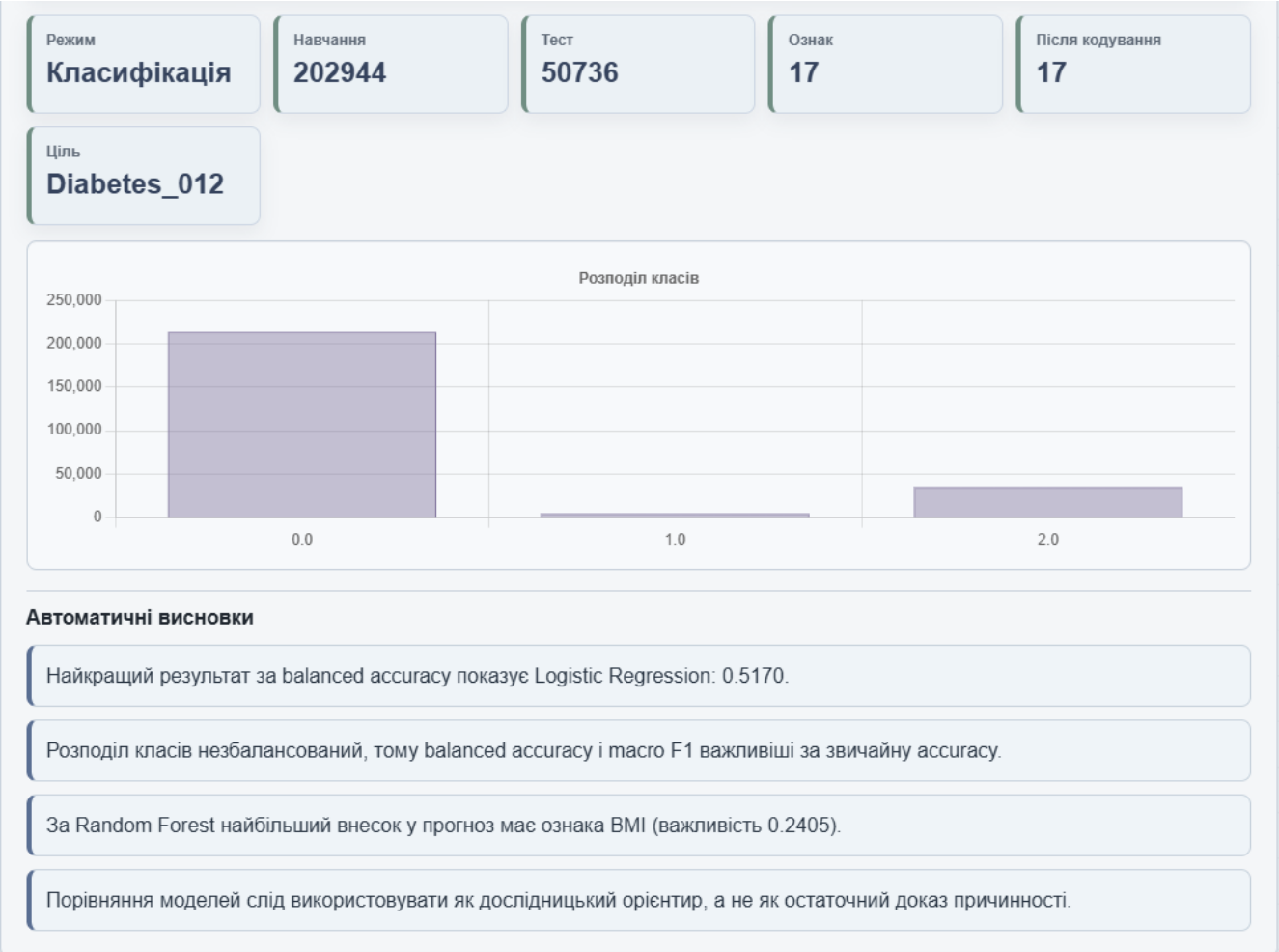
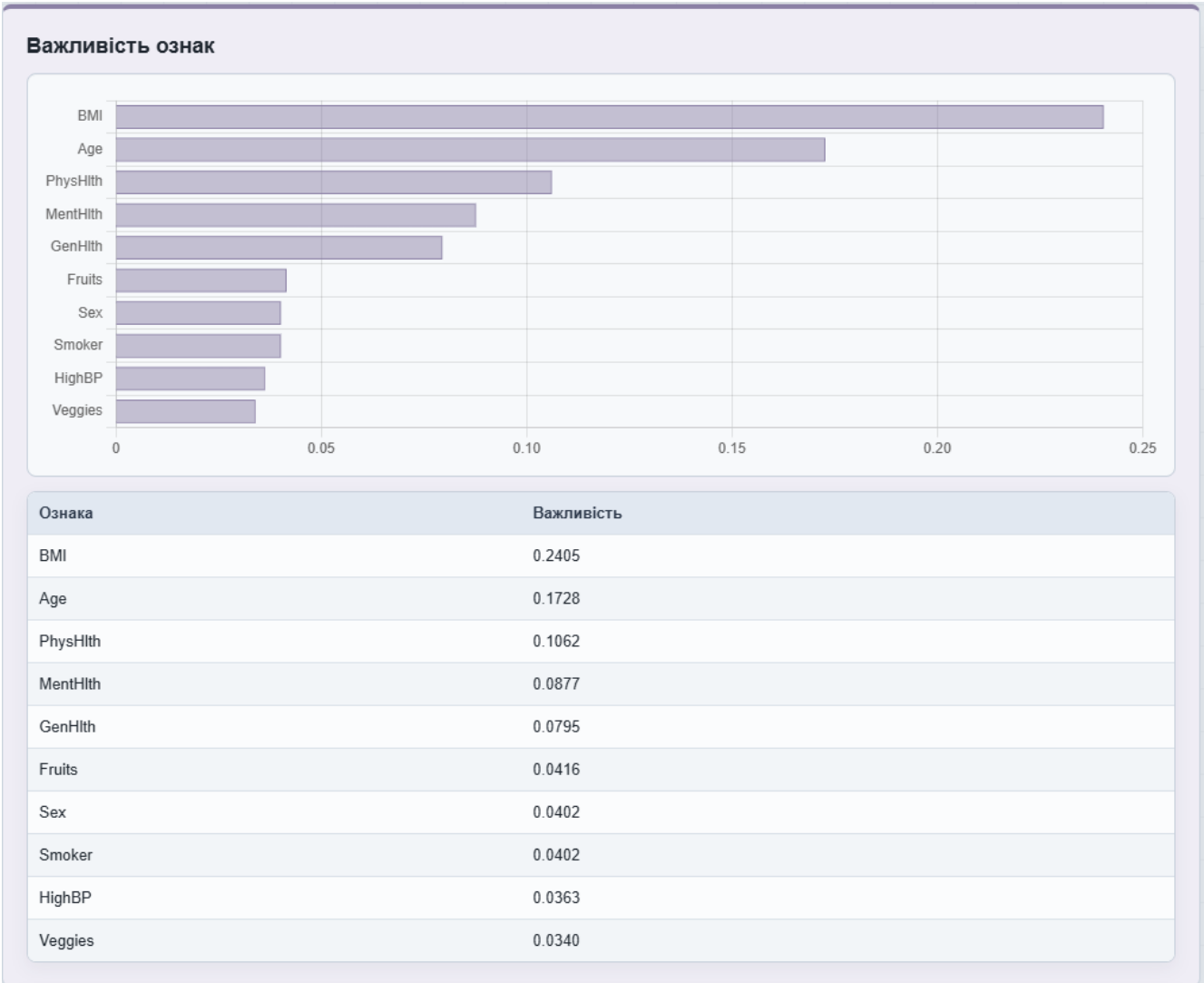


Рисунок 3.8 – Результати оцінювання моделей машинного навчання

Для детальнішого аналізу результатів класифікації реалізовано відображення матриці помилок. Вона дозволяє побачити, скільки об’єктів кожного класу було класифіковано правильно, а скільки – помилково віднесено до інших класів. Такий інструмент є корисним у випадках, коли загальна точність моделі виглядає прийнятною, але окремі класи розпізнаються значно гірше за інші. Завдяки цьому користувач може краще зрозуміти характер помилок моделі та обережніше інтерпретувати отримані результати.

Для моделі Random Forest додатково передбачено відображення важливості ознак. Цей результат показує, які вхідні змінні найбільше вплинули на формування прогнозу моделі. Такий підхід корисний не лише для оцінювання якості класифікації, а й для пояснення результатів моделювання. Користувач може побачити, які характеристики набору даних є найбільш значущими для побудованої

моделі, і використати цю інформацію під час подальшого аналізу. Приклад представлення важливості ознак наведено на рисунку 3.9.



Ознака	Важливість
BMI	0.2405
Age	0.1728
PhysHlth	0.1062
MentHlth	0.0877
GenHlth	0.0795
Fruits	0.0416
Sex	0.0402
Smoker	0.0402
HighBP	0.0363
Veggies	0.0340

Рисунок 3.9 – Відображення важливості ознак у моделі Random Forest

Реалізовані модулі машинного навчання не орієнтовані на досягнення максимально можливої точності для конкретного набору даних. Їх основне призначення полягає у наданні користувачеві інструменту для базового моделювання, порівняння різних алгоритмів та отримання зрозумілих результатів у межах єдиного процесу аналізу. Саме тому особлива увага приділяється не лише розрахунку метрик, а й їх поданню у формі, зручній для подальшої інтерпретації.

Практичним результатом реалізації модулів виявлення закономірностей та машинного навчання стала можливість перейти від первинного опису набору даних

до більш глибокого дослідження його структури. Користувач може визначити найпомітніші зв’язки між ознаками, побудувати просту регресійну модель, запустити класифікаційні алгоритми, переглянути метрики якості та отримати пояснення отриманих результатів. Усі ці дії виконуються в межах одного вебзастосунку без необхідності самостійного написання програмного коду.

Таким чином, реалізовані модулі виявлення закономірностей та машинного навчання розширюють можливості системи від базового аналізу даних до побудови моделей і оцінювання їх результатів. Поєднання рейтингу закономірностей, регресійного аналізу, класифікаційних моделей, метрик якості, матриці помилок і важливості ознак забезпечує користувачеві комплексний інструмент для дослідження табличних наборів даних.

3.4 Реалізація клієнтської частини та засобів візуалізації

Клієнтська частина вебзастосунку реалізована як інтерфейс для взаємодії користувача з основними функціями системи. Вона забезпечує завантаження наборів даних, запуск аналітичних операцій, перегляд результатів і роботу з історією виконаних дій. Для реалізації інтерфейсу використано HTML, CSS та JavaScript.

HTML відповідає за структуру сторінок, CSS – за їх візуальне оформлення, а JavaScript – за обробку дій користувача та надсилання запитів до серверної частини. Взаємодія між клієнтом і сервером здійснюється через HTTP-запити до API. Сервер повертає структуровані дані, а клієнтська частина відображає їх у вигляді таблиць, інформаційних блоків і графіків.

Практичним результатом реалізації клієнтської частини стало створення сторінок реєстрації та входу, головної сторінки, сторінки завантаження CSV-файлу, блоків профілювання, статистичного аналізу, кореляцій, закономірностей, машинного навчання та сторінки історії.

Після завантаження набору даних користувач отримує доступ до результатів профілювання та подальших аналітичних операцій. Результати відображаються без

необхідності перезавантаження сторінки, що забезпечує зручнішу роботу з системою.

Для візуалізації результатів використано Chart.js. Серверна частина не формує готові зображення графіків, а повертає дані у структурованому вигляді. Побудова гістограм, графіків кореляцій, результатів регресії, матриці помилок і важливості ознак виконується на клієнтській стороні.

Окремо реалізовано сторінку історії, де користувач може переглянути виконані раніше операції. Це дозволяє відстежувати послідовність роботи з наборами даних і повторно орієнтуватися в проведених аналізах. Сторінку історії виконаних аналізів наведено на рисунку 3.10.

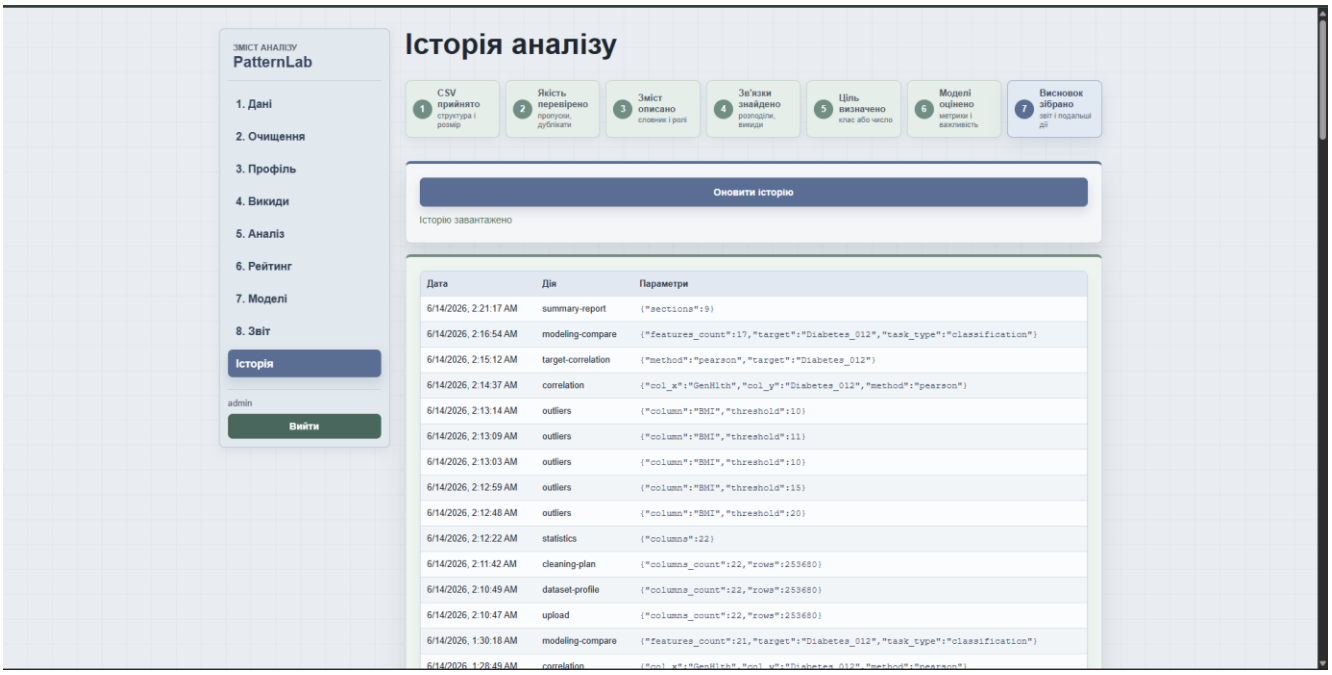


Рисунок 3.10 – Сторінка історії виконаних аналізів

Таким чином, клієнтська частина забезпечує доступ до всіх основних можливостей системи, а засоби візуалізації роблять результати аналізу зрозумілими для користувача без необхідності написання програмного коду.

4 ТЕСТУВАННЯ, ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ

У цьому розділі буде розглянуто перевірку працездатності розробленого вебзастосунку для аналізу даних, описано джерела експериментальних наборів даних, методику тестування основних функцій системи, результати профілювання та аналізу якості даних, а також оцінено роботу системи на вибірках різного обсягу.

4.1 Методика тестування та джерела експериментальних даних

Для перевірки працездатності розробленого вебзастосунку було використано реальні табличні набори даних, отримані з платформи Kaggle. Kaggle є відкритою платформою для пошуку, публікації та використання наборів даних, які застосовуються у задачах аналізу даних, машинного навчання та дослідницьких експериментів. Використання таких даних дозволяє перевірити роботу системи не на штучно створених прикладах, а на наборах, які мають реальну структуру, різні типи ознак, пропущені значення, дублікати та інші особливості, характерні для практичних задач аналізу даних.

Для експериментальної перевірки було використано декілька наборів даних, оскільки функціональність системи охоплює як первинний аналіз, так і побудову моделей машинного навчання. Основним набором для перевірки роботи системи на великому обсязі даних було обрано USA Real Estate Dataset, розміщений на платформі Kaggle [18]. Цей набір містить інформацію про об’єкти нерухомості у США та включає такі ознаки, як ціна, кількість кімнат, кількість ванних кімнат, площа земельної ділянки, площа будинку, місцезнаходження та інші характеристики. Наявність числових і категоріальних ознак робить цей набір зручним для перевірки профілювання, аналізу якості даних, статистичних розрахунків, кореляційного аналізу, побудови регресійних залежностей та оцінювання швидкодії системи на вибірках різного обсягу.

Додатково для перевірки задач класифікації було використано Diabetes Health Indicators Dataset, який також доступний на платформі Kaggle [19]. Цей набір даних

містить медико-соціальні показники, пов’язані зі станом здоров’я респондентів, та цільову ознаку, що дозволяє перевірити роботу класифікаційних моделей. Його використання є доцільним для тестування логістичної регресії, алгоритму Random Forest, розрахунку метрик якості класифікації, побудови матриці помилок та аналізу важливості ознак. Таким чином, використання двох різних наборів даних дозволило перевірити систему в різних сценаріях: аналіз числових показників, роботу з великим обсягом записів та виконання задач класифікації.

Тестування системи проводилося за двома основними підходами: ручним функціональним тестуванням і автоматизованою перевіркою окремих операцій. Ручне функціональне тестування виконувалося методом «чорної скриньки», тобто перевірка здійснювалася з позиції користувача без аналізу внутрішнього програмного коду під час виконання сценарію. У межах такого тестування перевірялися реєстрація та вхід до системи, завантаження CSV-файлів, відображення профілю набору даних, аналіз якості, перегляд статистичних характеристик, побудова графіків, виконання кореляційного аналізу, запуск моделей машинного навчання та перегляд історії виконаних дій.

Окремо перевірялися помилкові та граничні сценарії роботи системи. До таких сценаріїв належали спроба завантаження файлу неправильного формату, робота з набором даних без числових колонок, вибір некоректної або відсутньої цільової ознаки, повторне виконання аналізу для одного набору даних, а також перевірка доступу до функцій системи без авторизації. Такий підхід дозволив оцінити не лише правильність роботи основних функцій, а й здатність системи реагувати на некоректні дії користувача.

Автоматизована перевірка застосовувалася для повторюваних операцій, результати яких можна порівняти з очікуваними значеннями. Зокрема, перевірялися серверні запити до API, коректність обробки CSV-файлів, формування профілю набору даних, розрахунок статистичних характеристик, визначення пропущених значень, побудова кореляцій та запуск моделей машинного навчання. Для цього виконувалися однакові тестові сценарії на різних

вибірках даних, що дозволило перевірити стабільність роботи системи та порівняти час виконання основних операцій.

Під час тестування також оцінювалася швидкодія системи. Для цього з великого набору даних формувалися вибірки різного обсягу: малий, середній і великий набір записів. Для кожної вибірки перевірявся час завантаження, профілювання, статистичного аналізу, кореляційного аналізу та виконання складніших операцій. Такий підхід дозволив визначити, як обсяг даних впливає на швидкість роботи вебзастосунку та які операції створюють найбільше навантаження на серверну частину.

Таким чином, обрана методика тестування дозволила комплексно перевірити розроблене програмне забезпечення. Використання реальних наборів даних із Kaggle забезпечило наближеність експерименту до практичних умов, ручне функціональне тестування дало змогу оцінити роботу системи з позиції користувача, а автоматизована перевірка повторюваних операцій дозволила перевірити стабільність і коректність основних аналітичних функцій.

4.2 Первинне профілювання та аналіз якості даних у системі

Перевірка підсистеми профілювання та аналізу якості даних виконувалася на реальному наборі USA Real Estate Dataset і спеціально підготовлених тестових CSV-файлах із помилками. Метою перевірки було встановити, чи система правильно зчитує структуру файлу, визначає типи ознак, знаходить пропуски, дублікати, константні колонки та формує попередження для користувача.

Під час роботи з реальним набором даних система коректно визначила кількість рядків і колонок, типи ознак, загальну кількість пропущених значень та потенційні цільові колонки. Також було сформовано попередження щодо якості даних, зокрема про наявність пропусків і можливий дисбаланс класів.

Для перевірки стійкості системи було використано проблемний тестовий набір даних із навмисно доданими пропусками, дублікатами, константними колонками, квазипорожніми полями та закодованими категоріальними значеннями.

Система виявила основні проблеми якості та сформувала план очищення, у якому було зазначено кількість пропущених значень, повних дублікатів, константни0445 колонок і полів, що потребують додаткової перевірки (див. рисунок 4.1).

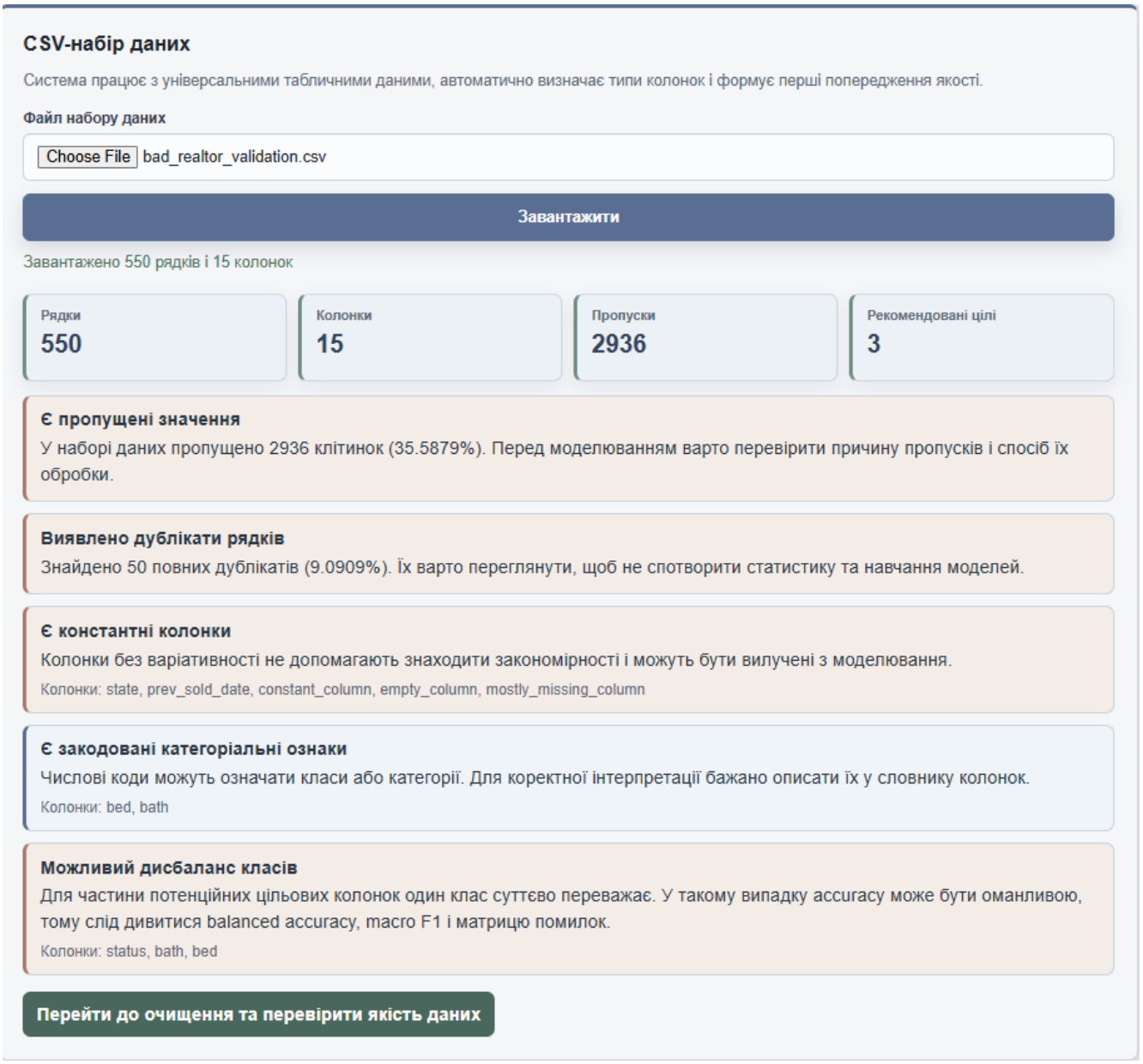


Рисунок 4.1 – Перевірка валідації даних

Додатково перевірялися некоректні або обмежені за структурою файли. Для набору без числових колонок система не виконувала числовий аналіз і не формувала помилкових статистичних висновків. Для файлу з неправильним розділювачем система зчитала дані як одну колонку та не визначила потенційну

цільову ознаку, що підтвердило коректну реакцію на некоректну структуру CSV файлу.

Таким чином, перевірка показала, що підсистема профілювання та аналізу якості даних коректно працює як із реальними наборами даних, так і з проблемними тестовими файлами. Система не лише зчитує CSV-файл, а й надає користувачеві інформацію про можливі обмеження подальшого аналізу.

4.3 Порівняння роботи системи на малому, середньому та великому обсязі вибірки

Для оцінювання швидкодії вебзастосунку було порівняно час виконання основних операцій на вибірках різного обсягу: 1000, 50000 і 100000 рядків. Як експериментальний набір даних використано Diabetes Health Indicators Dataset. Вимірювання виконувалися програмно на серверній стороні за службовими логами системи. Під час підготовки результатів інструменти штучного інтелекту використовувалися лише як допоміжний засіб для опрацювання логів і формування узагальнених значень.

Результати показали, що базові аналітичні операції масштабуються достатньо добре. Профілювання виконувалося приблизно за 0.034 с на малій вибірці, 0.155 с на середній і 0.183 с на великій. Статистичний аналіз також залишався швидким: 0.034 с, 0.072 с і 0.126 с відповідно. Це свідчить про те, що операції, пов’язані з узагальненими характеристиками набору даних, не створюють значного навантаження навіть при збільшенні кількості рядків.

Більш помітне зростання часу спостерігалося для плану очищення та аналізу якості даних. На 1000 рядків ця операція виконувалася за 0.096 с, на 50000 рядків – за 1.036 с, а на 100000 рядків – за 1.799 с. Це пояснюється тим, що система перевіряє пропуски, дублікати, константні колонки, нульові значення та інші потенційні проблеми в межах усього набору даних.

Кореляційний аналіз і рейтинг закономірностей також виконувалися швидко. Парна кореляція між двома ознаками навіть на 100000 рядків тривала лише 0.029 с,

а рейтинг закономірностей – 0.165 с. Це підтверджує, що такі операції можуть ефективно використовуватися для попереднього аналізу середніх за обсягом наборів даних.

Найбільш ресурсомісткою операцією стало машинне навчання. На 1000 рядків воно виконувалося приблизно за 1.187 с, на 50000 рядків – за 10.950 с, а на 100000 рядків – за 29.006 с. Таке зростання є очікуваним, оскільки машинне навчання включає підготовку даних, поділ вибірки, навчання моделей Logistic Regression і Random Forest, формування прогнозів та обчислення метрик.

Порівняння часу виконання основних операцій наведено у таблиці 1.

Таблиця 1 – Порівняння часу виконання

Операція	**1000 рядків**	**50000 рядків**	**100000 рядків**
Профілювання набору даних	0.034 с	0.155 с	0.183 с
План очищення / аналіз якості	0.096 с	1.036 с	1.799 с
Статистичний аналіз	0.034 с	0.072 с	0.126 с
Кореляція між двома ознаками	0.008 с	0.013 с	0.029 с
Кореляційна матриця	0.014 с	0.076 с	0.078 с
Рейтинг закономірностей	0.054 с	0.131 с	0.165 с
Машинне навчання	1.187 с	10.950 с	29.006 с

Отже, система демонструє прийнятну швидкодію для базових операцій аналізу навіть на вибірці у 100000 рядків. Найбільше навантаження створюють детальний аналіз якості та машинне навчання, тому саме ці модулі є основними кандидатами для подальшої оптимізації.

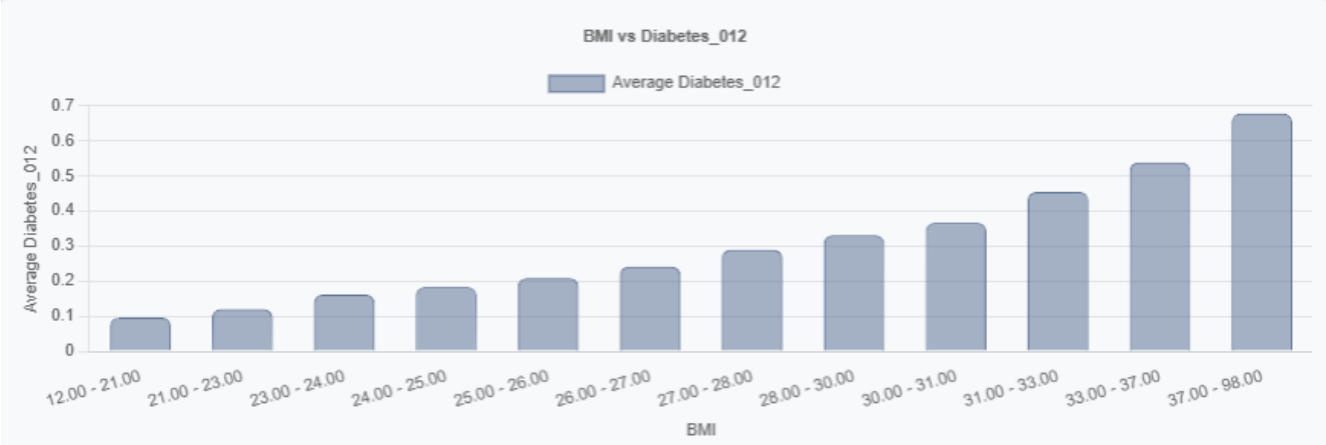
4.4 Перевірка роботи модулів виявлення закономірностей і машинного навчання

Для перевірки роботи модулів виявлення закономірностей і машинного навчання було використано Diabetes Health Indicators Dataset із цільовою колонкою

Diabetes_012. Основна увага приділялася не повторному опису структури набору даних, а перевірці узгодженості результатів між різними модулями системи та відповідності автоматичних висновків отриманим числовим значенням.

Правильність роботи оцінювалася за внутрішньою узгодженістю результатів. Значення парної кореляції повинні відповідати кореляційній матриці та рейтингу закономірностей, параметри регресії мають узгоджуватися з графічним представленням тренду, а висновки щодо моделей машинного навчання повинні відповідати отриманим метрикам.

Під час перевірки система визначила слабкий прямий зв'язок між BMI і Diabetes_012 з коефіцієнтом 0.2244, а також помірний прямий зв'язок між GenHlth і Diabetes_012 з коефіцієнтом 0.3026. Ці результати були узгоджені з кореляційною матрицею, у якій відповідні значення становили 0.224 і 0.302 (див. рисунок 4.2).



	Diabetes_012	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruit
Diabetes_012	1.000	0.272	0.209	0.068	0.224	0.063	0.107	0.180	-0.122	-0.04
HighBP	0.272	1.000	0.298	0.099	0.214	0.097	0.130	0.209	-0.125	-0.04
HighChol	0.209	0.298	1.000	0.086	0.107	0.091	0.093	0.181	-0.078	-0.04
CholCheck	0.068	0.099	0.086	1.000	0.035	-0.010	0.024	0.044	0.004	0.02
BMI	0.224	0.214	0.107	0.035	1.000	0.014	0.020	0.053	-0.147	-0.08

Рисунок 4.2 - Порівняння результатів

Рейтинг закономірностей також підтвердив узгодженість результатів: найсильніший зв'язок із Diabetes_012 мала ознака GenHlth, далі були HighBP, BMI,

DiffWalk, HighChol та Age. Отже, система правильно ранжує ознаки за силою зв’язку з цільовою колонкою.

Для перевірки регресійного аналізу було побудовано лінійну модель між HighBP і Diabetes_012. Система визначила нахил 0.3831, вільний член 0.1326 і R2 = 0.0738. Низьке значення R2 було коректно інтерпретовано як слабку пояснювальну здатність моделі, тому система не перебільшила значущість отриманого результату.

У модулі машинного навчання система порівняла Logistic Regression і Random Forest. Для Logistic Regression balanced accuracy становила 0.5243, а для Random Forest – 0.3768. Хоча Random Forest мав вищу загальну accuracy, система правильно визначила Logistic Regression як кращу модель за balanced accuracy, що є важливим для незбалансованого набору даних.

Матриця помилок підтвердила, що найбільша кількість правильних прогнозів припадає на домінуючий клас 0, тоді як класи 1 і 2 класифікуються гірше. Це пояснює, чому для такого набору даних недостатньо оцінювати модель лише за accuracy. Для Random Forest також було відображено важливість ознак, де найбільший внесок мали BMI, Age, Income, PhysHlth і GenHlth.

Таким чином, перевірка показала, що модулі виявлення закономірностей і машинного навчання працюють узгоджено. Система коректно передає параметри між інтерфейсом і серверною частиною, узгоджено відображає результати в різних модулях та формує висновки, які відповідають числовим показникам.

4.5 Впровадження експлуатація та напрями подальшого розвитку системи

На даному етапі розробки програмне забезпечення може використовуватися як локальне програмне рішення для аналізу табличних наборів даних. Для його запуску необхідно мати встановлене середовище Python, потрібні бібліотеки та сучасний веббраузер. Серверна частина запускається локально, після чого користувач отримує доступ до інтерфейсу системи через браузер. Такий спосіб

експлуатації є достатнім для демонстрації працездатності системи, проведення тестування та використання застосунку в навчальних або дослідницьких цілях.

У процесі експлуатації користувач може завантажити CSV-файл, переглянути профіль набору даних, оцінити його якість, виконати статистичний і кореляційний аналіз, виявити закономірності, запустити моделі машинного навчання та переглянути історію виконаних дій. Основні обчислення виконуються на серверній стороні, а результати відображаються у вебінтерфейсі у вигляді таблиць, графіків і текстових пояснень. Це дозволяє працювати із системою без необхідності написання програмного коду.

Для збереження та передачі результатів розроблення фінальну версію програмного коду було розміщено у віддаленому репозиторії (див. додаток Б). У цьому випадку репозиторій використовувався насамперед як засіб збереження завершеної версії проєкту та надання доступу до нього, а не як основний інструмент поетапного контролю версій під час розроблення. Такий підхід дозволяє зберегти структуру програмного проєкту, спростити його повторне завантаження та подальше розгортання в іншому середовищі.

Проведене тестування показало, що система коректно виконує основні функції та може працювати з наборами даних різного обсягу. Водночас під час експлуатації слід враховувати певні обмеження. Зокрема, поточна версія системи орієнтована переважно на CSV-файли, використовує тимчасове зберігання активного набору даних в оперативній пам’яті та не призначена для одночасної роботи великої кількості користувачів. Крім того, для дуже великих наборів даних найбільш ресурсомісткими залишаються операції машинного навчання та детального аналізу якості.

Подальший розвиток системи доцільно здійснювати насамперед у напрямі розширення її функціональних можливостей як навчального вебсередовища. Зокрема, варто додати підтримку Excel-файлів, можливість збереження повних результатів аналізу та експорт сформованих звітів у PDF або DOCX. Це дозволить користувачеві не лише переглядати результати у вебінтерфейсі, а й зберігати їх для подальшого використання в навчальних роботах, звітах або дослідницьких

матеріалах. Також перспективним є розширення кількості пояснювальних повідомлень, які супроводжують аналітичні результати. Наприклад, система могла б детальніше пояснювати значення коефіцієнта кореляції, метрик машинного навчання, результатів регресії або попереджень щодо якості даних. Це посилило б навчальну цінність застосунку та зробило його більш корисним для користувачів, які лише починають вивчати аналіз даних.

Окремим напрямом подальшого розвитку є підвищення продуктивності та готовності системи до складніших умов експлуатації. Поточна версія використовує тимчасове зберігання активного набору даних в оперативній пам’яті, що є прийнятним для локального використання та навчальних задач, але може створювати обмеження під час роботи з дуже великими файлами або кількома користувачами одночасно. У майбутньому доцільно реалізувати ефективніший механізм обробки великих наборів даних, додати обмеження на розмір файлів, оптимізувати виконання ресурсомістких операцій машинного навчання та передбачити можливість розгортання системи на віддаленому сервері. У разі переходу до багатокористувацького використання також доцільним буде замінити SQLite на серверну систему керування базами даних, наприклад PostgreSQL, що забезпечить кращу підтримку одночасних запитів і масштабування системи.

Таким чином, розроблене програмне забезпечення готове до локального використання та демонструє працездатність основних функцій аналізу даних. Реалізована система може бути використана як основа для подальшого розвитку, розширення функціональності та впровадження у більш складне середовище експлуатації.

5 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

У цьому розділі розглянуто питання безпеки життєдіяльності та основ охорони праці, пов’язані із забезпеченням безпечних умов діяльності людини. Особливу увагу приділено застосуванню ризик-орієнтованого підходу для аналізу можливих надзвичайних ситуацій, а також заходам психофізіологічного розвантаження працівників. Розгляд цих питань є важливим, оскільки безпечні умови праці, своєчасне виявлення небезпек і підтримання працездатності працівників сприяють зниженню ризиків, запобіганню професійному виснаженню та підвищенню ефективності трудової діяльності.

5.1 Застосування ризик-орієнтованого підходу для побудови імовірнісних структурно-логічних моделей виникнення та розвитку надзвичайних ситуацій

Одним із сучасних напрямів забезпечення безпеки життєдіяльності є застосування ризик-орієнтованого підходу, який передбачає виявлення небезпек, оцінювання ризиків та розроблення заходів щодо їх мінімізації. Такий підхід широко використовується у сфері цивільного захисту, охорони праці та техногенної безпеки, оскільки дозволяє прогнозувати виникнення небезпечних подій ще до моменту їх реалізації та своєчасно вживати необхідних профілактичних заходів [20].

Відповідно до сучасних підходів ризик розглядається як поєднання ймовірності виникнення небезпечної події та тяжкості можливих наслідків її реалізації. Абсолютна безпека практично недосяжна, тому головною метою управління ризиками є зниження їх рівня до прийнятних значень. При цьому особлива увага приділяється небезпекам, які можуть спричинити людські жертви, значні матеріальні збитки або негативний вплив на навколишнє середовище (за змістом [21].

Ризик-орієнтований підхід базується на послідовному виконанні декількох етапів. На першому етапі здійснюється ідентифікація небезпек. Виявляються

потенційні джерела виникнення аварій, катастроф, пожеж, вибухів, техногенних аварій або інших надзвичайних ситуацій. Другим етапом є аналіз причин виникнення небезпечних подій та оцінювання їх імовірності. Після цього визначаються можливі наслідки реалізації ризиків і розробляються заходи щодо їх попередження або мінімізації. Завершальним етапом є постійний моніторинг небезпек та контроль ефективності впроваджених заходів безпеки [20].

Для аналізу небезпек та прогнозування розвитку надзвичайних ситуацій широко використовуються структурно-логічні моделі. Такі моделі являють собою графічне або математичне відображення причинно-наслідкових зв'язків між окремими подіями. Вони дозволяють досліджувати механізм виникнення небезпечної ситуації, визначати критичні фактори та оцінювати вплив окремих елементів системи на загальний рівень ризику [21].

Одним із найбільш поширених методів є побудова дерева відмов. Даний метод ґрунтується на визначенні головної небажаної події та послідовному встановленні причин, які можуть призвести до її виникнення. Структура дерева відмов формується за допомогою логічних зв'язків «І» та «АБО», що дозволяє враховувати різні комбінації подій. Перевагою такого підходу є можливість кількісного оцінювання ризику та визначення найбільш небезпечних факторів системи [21].

Іншим поширеним методом є дерево подій. На відміну від дерева відмов, воно будується від початкової події до можливих наслідків її розвитку. Такий підхід дозволяє аналізувати різні сценарії розвитку надзвичайної ситуації та оцінювати ефективність захисних заходів. Наприклад, після виникнення пожежі подальший розвиток подій залежатиме від спрацювання систем пожежної сигналізації, ефективності евакуації людей, наявності засобів пожежогасіння та своєчасного реагування відповідних служб [20].

Особливого значення ризик-орієнтований підхід набуває в умовах зростання техногенного навантаження на навколишнє середовище. Сучасні підприємства використовують складні технологічні процеси, значну кількість обладнання та небезпечних речовин. За таких умов навіть незначна помилка або відмова окремого

елемента системи може призвести до масштабних наслідків. Саме тому особливу увагу приділяють виявленню так званих критичних подій, реалізація яких здатна спричинити розвиток аварії або катастрофи [20].

Під час побудови імовірнісних структурно-логічних моделей використовуються статистичні дані щодо частоти виникнення небезпечних подій, надійності обладнання, кількості аварій та інших показників. На основі цих даних визначаються ймовірності реалізації окремих подій, після чого розраховується загальна ймовірність виникнення надзвичайної ситуації. Це дозволяє перейти від якісної оцінки небезпеки до кількісного аналізу ризику [21].

Значну роль у виникненні надзвичайних ситуацій відіграє людський фактор. Порушення правил безпеки, недостатня підготовка персоналу, помилки під час виконання робіт або недотримання технологічних вимог часто стають причинами аварійних ситуацій. Тому під час побудови структурно-логічних моделей необхідно враховувати не лише технічні, а й організаційні та психофізіологічні чинники [20].

Для зменшення ризику виникнення надзвичайних ситуацій застосовується комплекс організаційних та технічних заходів. До організаційних заходів належать проведення інструктажів і навчання персоналу, розроблення планів реагування на надзвичайні ситуації, контроль дотримання вимог безпеки та проведення періодичних перевірок. Технічні заходи включають використання сучасних систем контролю, автоматичного захисту, пожежної сигналізації, аварійного відключення обладнання та інших засобів безпеки [21].

Важливим принципом ризик-орієнтованого підходу є безперервність процесу управління ризиками. Оскільки умови функціонування об'єктів постійно змінюються, необхідно регулярно переглядати результати оцінювання ризиків та актуалізувати структурно-логічні моделі. Це дозволяє своєчасно виявляти нові небезпеки та підтримувати належний рівень безпеки [21].

Отже, ризик-орієнтований підхід є ефективним інструментом прогнозування та попередження надзвичайних ситуацій. Використання імовірнісних структурно-логічних моделей дає змогу встановити причинно-наслідкові зв'язки між

небезпечними подіями, оцінити рівень ризику та визначити найбільш ефективні заходи щодо його зниження. Застосування таких методів сприяє підвищенню рівня безпеки людей, захисту матеріальних цінностей і забезпеченню сталого функціонування об'єктів господарської діяльності.

5.2 Психофізіологічне розвантаження для працівників

Ефективність трудової діяльності людини значною мірою залежить від її психофізіологічного стану. У процесі виконання професійних обов'язків працівник зазнає впливу різноманітних фізичних, психічних, емоційних та інформаційних навантажень, які можуть призводити до розвитку втоми, зниження працездатності та погіршення стану здоров'я. Одним із важливих напрямів охорони праці є забезпечення умов для психофізіологічного розвантаження працівників, що дозволяє підтримувати високий рівень працездатності, зменшувати ризик професійних захворювань та сприяти збереженню здоров'я персоналу [22].

Психофізіологічне розвантаження являє собою комплекс організаційних, санітарно-гігієнічних, психологічних та фізіологічних заходів, спрямованих на відновлення функціонального стану організму людини після впливу виробничих навантажень. Основною метою таких заходів є профілактика перевтоми, зниження нервово-емоційного напруження та забезпечення сприятливих умов праці [22].

У сучасних умовах розвитку інформаційного суспільства значна частина працівників виконує роботу, пов'язану з інтенсивною розумовою діяльністю. На відміну від фізичної праці, яка супроводжується значними м'язовими навантаженнями, розумова праця характеризується високою концентрацією уваги, необхідністю швидкого аналізу інформації, прийняття рішень та постійним психоемоційним напруженням. При цьому зовнішні ознаки втоми можуть бути менш помітними, однак функціональні зміни в організмі накопичуються поступово та негативно впливають на загальний стан працівника [23].

Однією з основних проблем сучасної трудової діяльності є розвиток виробничої втоми. Втома являє собою тимчасове зниження працездатності, яке

виникає внаслідок тривалого або інтенсивного виконання роботи. Вона супроводжується погіршенням уваги, зниженням швидкості реакції, збільшенням кількості помилок та погіршенням координації дій. За відсутності достатнього відпочинку втома може переходити у перевтому, яка характеризується більш глибокими порушеннями функціонального стану організму та потребує тривалого відновлення [24].

Особливу небезпеку становить хронічна втома, яка виникає внаслідок систематичного перевантаження працівника без належного відпочинку. Вона може супроводжуватися порушеннями сну, зниженням працездатності, підвищеною дратівливістю, погіршенням пам'яті та концентрації уваги. Хронічна втома також негативно впливає на серцево-судинну, нервову та імунну системи організму, що підвищує ризик виникнення різних захворювань [23].

Важливим чинником, який впливає на психофізіологічний стан працівників, є нервово-емоційне напруження. Воно виникає під час виконання відповідальної роботи, необхідності прийняття швидких рішень, дефіциту часу, високої інтенсивності праці або дії стресових факторів. Тривале перебування в умовах емоційного напруження може призводити до розвитку професійного стресу [24].

Професійний стрес є реакцією організму на несприятливі виробничі фактори та надмірні навантаження. Він проявляється у вигляді підвищеної тривожності, дратівливості, погіршення самопочуття та зниження ефективності роботи. За тривалого впливу стресових факторів можуть виникати психосоматичні порушення, захворювання серцево-судинної системи, нервові розлади та інші негативні наслідки [22].

Одним із сучасних проявів негативного впливу психоемоційних навантажень є синдром професійного вигорання. Цей стан характеризується емоційним виснаженням, втратою мотивації до роботи, зниженням професійної ефективності та байдужістю до результатів власної діяльності. Найчастіше професійне вигорання спостерігається серед працівників, діяльність яких пов'язана з високою відповідальністю, інтенсивною розумовою працею та постійною взаємодією з людьми [22].

Для попередження розвитку втоми, стресу та професійного вигорання важливе значення має раціональна організація режиму праці та відпочинку. Чергування періодів роботи й відпочинку дозволяє підтримувати оптимальний рівень працездатності протягом робочого дня та забезпечує своєчасне відновлення функціональних можливостей організму. Раціональний режим праці повинен враховувати характер роботи, її інтенсивність, рівень відповідальності та індивідуальні особливості працівників [23].

Одним із найефективніших засобів психофізіологічного розвантаження є виробнича гімнастика. Вона являє собою комплекс спеціально підібраних фізичних вправ, які виконуються під час робочих перерв. Основними завданнями виробничої гімнастики є покращення кровообігу, зменшення м'язового напруження, профілактика застійних явищ та відновлення працездатності працівників. Регулярне виконання фізичних вправ сприяє підвищенню загального тонусу організму та зменшенню негативного впливу малорухомого способу життя [23].

Ефективним способом відновлення працездатності є активний відпочинок. На відміну від пасивного відпочинку, який передбачає припинення трудової діяльності без зміни характеру навантаження, активний відпочинок пов'язаний зі зміною виду діяльності. Наприклад, після тривалої розумової праці корисними є фізичні вправи, прогулянки на свіжому повітрі або інші види рухової активності. Такий підхід сприяє швидшому відновленню працездатності та покращує функціональний стан організму [22].

Важливу роль у психофізіологічному розвантаженні відіграє психологічний клімат у колективі. Сприятлива атмосфера, взаємоповага між працівниками, справедливий розподіл обов'язків та підтримка з боку керівництва позитивно впливають на емоційний стан персоналу. Навпаки, конфліктні ситуації, надмірний контроль або невизначеність щодо службових обов'язків можуть виступати додатковими джерелами стресу [24].

Одним із сучасних методів психофізіологічного розвантаження є використання кімнат психологічного розвантаження. Такі приміщення обладнуються з урахуванням вимог ергономіки та психологічного комфорту. У них

можуть використовуватися зручні меблі, аудіосистеми для відтворення релаксаційної музики, засоби ароматерапії, декоративне оформлення та інші елементи, що сприяють зниженню нервово-емоційного напруження [24].

Для зменшення негативного впливу виробничих навантажень також застосовуються різноманітні методи психологічної саморегуляції. До них належать дихальні вправи, методи релаксації, аутогенне тренування, медитація та інші техніки керування психоемоційним станом. Використання таких методів дозволяє знизити рівень тривожності, покращити концентрацію уваги та підвищити стійкість до стресових ситуацій [23].

Не менш важливе значення мають санітарно-гігієнічні умови праці. Невідповідність параметрів мікроклімату, недостатнє освітлення, підвищений рівень шуму або вібрації негативно впливають на фізіологічний стан працівників та сприяють швидшому розвитку втоми. Тому під час організації робочих місць необхідно забезпечувати дотримання нормативних вимог щодо мікроклімату, освітлення, шуму та інших факторів виробничого середовища [22].

Важливим напрямом забезпечення психофізіологічного благополуччя працівників є впровадження принципів ергономіки. Раціональна організація робочого місця дозволяє зменшити фізичне навантаження на організм, підвищити комфортність праці та запобігти розвитку професійних захворювань. Ергономічні вимоги стосуються конструкції меблів, розташування обладнання, організації робочої пози та забезпечення оптимальних умов виконання трудових операцій [23].

Таким чином, психофізіологічне розвантаження є важливою складовою системи охорони праці. Застосування комплексу організаційних, психологічних, санітарно-гігієнічних та фізіологічних заходів дозволяє підтримувати високий рівень працездатності працівників, попереджати розвиток втоми та професійного стресу, зберігати здоров'я персоналу та підвищувати ефективність трудової діяльності.

ВИСНОВКИ

У кваліфікаційній роботі розв’язано науково-технічну задачу розробки навчального програмного забезпечення для аналізу табличних даних та виявлення закономірностей у них. Розроблена система призначена для послідовного ознайомлення користувача з основними етапами роботи з CSV-файлами: завантаженням набору даних, переглядом його структури, аналізом якості, обчисленням статистичних характеристик, побудовою графіків, дослідженням зв’язків між ознаками та запуском базових моделей машинного навчання.

У процесі виконання роботи було проаналізовано предметну область аналізу даних, розглянуто існуючі програмні рішення та визначено, що для навчальних цілей важливим є не лише отримання результату, а й послідовне подання етапів аналізу з поясненнями. На основі цього було сформовано функціональні та нефункціональні вимоги до вебзастосунку, спроєктовано його архітектуру, структуру серверної та клієнтської частин, базу даних і основні сценарії роботи користувача.

Серверну частину системи реалізовано на основі Flask API. Для обробки даних використано Python, pandas і scikit-learn, а для зберігання службової інформації – SQLite. Завантажений CSV-файл обробляється як активний DataFrame в оперативній пам’яті сервера, що дозволяє виконувати різні аналітичні операції над одним набором даних без повторного завантаження. Клієнтську частину реалізовано з використанням HTML, CSS, JavaScript і Chart.js, що забезпечує відображення результатів у вигляді таблиць, графіків і коротких пояснень.

У межах розробленого вебзастосунку реалізовано реєстрацію та вхід користувача, завантаження CSV-файлів, профілювання набору даних, аналіз якості, побудову плану очищення, обчислення описової статистики, побудову гістограм, кореляційний аналіз, кореляційну матрицю, лінійну регресію, виявлення викидів за Z-score, формування рейтингу закономірностей, запуск моделей машинного навчання та перегляд історії дій користувача.

Проведене тестування підтвердило працездатність основних функцій системи. Для перевірки використовувалися реальні та тестові набори даних, зокрема Diabetes Health Indicators Dataset і набір даних про нерухомість. Система коректно визначала структуру даних, формувала попередження про якість, будувала статистичні показники, кореляції, графіки та результати машинного навчання. Достовірність результатів підтверджено їх внутрішньою узгодженістю між різними модулями системи.

Під час оцінювання швидкодії було встановлено, що система може працювати з вибірками різного обсягу. Для 1000, 50000 і 100000 рядків профілювання виконувалося за 0,034 с, 0,155 с і 0,183 с відповідно, статистичний аналіз – за 0,034 с, 0,072 с і 0,126 с, а кореляційна матриця – за 0,014 с, 0,076 с і 0,078 с. Найбільш ресурсомісткою операцією було машинне навчання, час виконання якого становив 1,187 с, 10,950 с і 29,006 с відповідно. Отримані показники свідчать, що система є придатною для навчального використання та первинного аналізу малих і середніх наборів даних.

Практичне значення роботи полягає у створенні вебзастосунку, який може використовуватися як навчальне середовище для ознайомлення з основами аналізу табличних даних без написання програмного коду. Система може бути корисною під час виконання лабораторних робіт, навчальних проєктів або попереднього дослідження CSV-файлів перед використанням складніших інструментів.

Подальший розвиток системи може передбачати підтримку Excel-файлів, збереження повних звітів аналізу, експорт результатів у PDF або DOCX, додавання нових моделей машинного навчання, оптимізацію роботи з великими наборами даних і розгортання вебзастосунку на віддаленому сервері.

Отже, у результаті виконання кваліфікаційної роботи було створено навчальне вебсередовище для аналізу табличних даних, яке реалізує базовий цикл роботи з CSV-файлом і може використовуватися для вивчення основ аналізу даних та виявлення закономірностей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 4th ed. Cambridge : Morgan Kaufmann, 2022. 752 p. URL: https://datamineaz.org/textbooks/hanDataMiningConceptual.pdf. (date of access: 14.06.2026).

2. Chapman P., Clinton J., Kerber R., Khabaza T., Reinartz T., Shearer C., Wirth R. CRISP-DM 1.0: Step-by-step data mining guide. SPSS Inc., 2000. 76 p. URL: https://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/14.2/es/CRISP-DM.pdf. (date of access: 14.06.2026).

3. Bruce P., Bruce A., Gedeck P. Practical Statistics for Data Scientists: 50+ Essential Concepts Using R and Python. 2nd ed. Sebastopol : O'Reilly Media, 2020. 368 p. URL: https://datapot.vn/wp-content/uploads/2023/12/datapot.vn-Practical-Statistics-for-Data-Scientists.pdf?srsltid=AfmBOop0Aw4NogXNmGpNO8dG6EsGdynuyQ99sbq5ldXWUPBjjXe7hkzu. (date of access: 14.06.2026).

4. James G., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning: with Applications in R. 2nd ed. New York : Springer, 2021. 607 p. URL: https://www.casact.org/sites/default/files/2022-12/James-G.-et-al.-2nd-edition-Springer-2021.pdf. (date of access: 14.06.2026).

5. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 3rd ed. Sebastopol : O'Reilly Media, 2022. 864 p. URL: https://www.rasa-ai.com/wp-content/uploads/2022/02/Aur%C3%A9lien-G%C 3%A9ron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow_-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-O%E2%80%99Reilly-Media-2019.pdf. (date of access: 14.06.2026).

6. Power BI vs Tableau – Pros and Cons – The Data School. The Data School powered by The Information Lab. URL: https://www.thedataschool.co.uk/a/carlo-sanzeri/power-bi-vs-tableau-pros-and-cons/ (date of access: 14.06.2026).

7. GeeksforGeeks. Data Analysis and Visualization with Jupyter Notebook - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/data-analysis/data-analysis-and-visualization-with-jupyter-notebook/ (date of access: 14.06.2026).

8. Data Mining. Orange Data Mining – Data Mining. URL: https://oldorange.biolab.si/ (date of access: 14.06.2026).

9. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill Education, 2019. 705 p. URL: https://www.mlsu.ac.in/econtents/16_EBOOK-7th_ed_software_engineering_a_practitioners_approach_by_roger_s._pressman_.pdf. (date of access: 14.06.2026).

10. Bass L., Clements P., Kazman R. Software Architecture in Practice. 4th ed. Boston : Addison-Wesley, 2021. 464 p. URL https://www.scribd.com/document/771670822/Software-Architecture-in-Practice-4th -Edition. (date of access: 14.06.2026).

11. David R. Cheriton School of Computer Science | Cheriton School of Computer Science | University of Waterloo. URL: https://cs.uwaterloo.ca/~m2nagapp/courses/CS446/1195/Arch_Design_Activity/Layered.pdf (дата звернення: 14.06.2026).

12. Microservices Architecture | Atlassian. Collaboration software for software, IT and business teams | Atlassian. URL: https://www.atlassian.com/microservices/microservices-architecture (date of access: 14.06.2026).

13. Choosing the Right Software Architecture: A Guide to Selecting the Best Fit for Your Project. Medium. URL: https://ravindusandaruwandh. medium.com/choosing-the-right-software-architecture-a-guide-to-selecting-the-best-fit-for-your-project-ecdda7812fb1. (date of access: 14.06.2026).

14. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol : O'Reilly Media, 2017. 616 p. URL: https://0-lucas.github.io/digital-garden/99.-Books/Martin-Kleppmann-Designing-Data-Intensive-Applications_-O%E2%80%99Reilly-Media-(2017).pdf. (date of access: 14.06.2026).

15. GeeksforGeeks. Flask Tutorial – GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/python/flask-tutorial/ (date of access: 14.06.2026).

16. Müller A. C., Guido S. Introduction to Machine Learning with Python: A Guide for Data Scientists. Sebastopol : O'Reilly Media, 2017. 392 p. URL: https://www.nrigroupindia.com/e-book/Introduction%20to%20Machine%20Learning%20with%20Python%20(%20PDFDrive.com%20)-min.pdf. (date of access: 14.06.2026).

17. GeeksforGeeks. Logistic Regression Vs Random Forest Classifier – GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/machine-learning/logistic-regression-vs-random-forest-classifier/ (date of access: 14.06.2026).

18. Shree. USA Real Estate Dataset. Kaggle, 2023. URL: https://www.kaggle.com/datasets/ahmedshahriarsakib/usa-real-estate-dataset. (date of access: 14.06.2026).

19. UCI Machine Learning Repository. CDC Diabetes Health Indicators Dataset. Irvine : University of California, School of Information and Computer Sciences, 2023. URL: https://archive.ics.uci.edu/dataset/891/cdc+diabetes+health+indicators. (date of access: 14.06.2026).

20. Атаманчук П. С. Безпека життєдіяльності : навч. посіб. Київ : Центр учбової літератури, 2020. 276 с. URL: https://studfile.net/preview/10132083/. (дата звернення: 14.06.2026)

21. Запорожець О. І. Безпека життєдіяльності : підручник. 2-ге вид. Київ : Центр учбової літератури, 2020. 448 с. URL: https://opcb.kpi.ua/wp-content/uploads/2014/09/%D0%91%D0%96%D0%94_%D0%9D%D0%B0%D0%B2%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%BF%D0%BE%D1%81%D1%96%D0%B1%D0%BD%D0%B8%D0%BA.pdf. (дата звернення: 14.06.2026)

22. Андрейчук Н. І. Охорона праці : навч. посіб. / Н. І. Андрейчук, Ю. В. Кіт, С. В. Шибанов, О. В. Шерстньова. Львів : Видавництво Львівська політехніка, 2021. 276 с.

23. Бедрій Я. І. Основи охорони праці : навчальний посібник для студентів вищих навчальних закладів. Вид. 4-те, перероб. і допов. Тернопіль : Навчальна книга – Богдан, 2014. 240 с. URL: http://www.kgemt.org.ua/pdf /about_colege/library_fund/%D0%91%D0%B5%D0%B4%D1%80%D1%96%D0%B9%20%D0%AF.%20%D0%86.%20%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D0%B8%20%D0%BE%D1%85%D0%BE%D1%80%D0%BE%D0%BD%D0%B8%20%D0%BF%D1%80%D0%B0%D1%86%D1%96%202014.pdf (дата звернення: 14.06.2026)

24. Жидецький В. Ц. Основи охорони праці : підручник. Львів : Афіша, 2005. 320 с. URL: https://xn--e1ajqk.kiev.ua/wp-content/uploads/2019/12/ZHideczkij-V.CZ.-Osnovi-ohoroni-praczi.-Pidruchnik-1.pdf/. (дата звернення: 14.06.2026).

ДОДАТКИ

ДОДАТОК А

Тези конференції

Міністерство освіти і науки України
Тернопільський національний технічний університет
імені Івана Пулюя
Маріборський університет (Словенія)
Технічний університет в Кошице (Словаччина)
Каунаський технологічний університет (Литва)
Львівський національний університет
імені Івана Франка
Гірничо-металургійна академія ім. Станіслава Сташиця (Польща)
Луцький національний технічний університет
Чернівецький національний університет
імені Юрія Федьковича
Вроцлавський економічний університет (Польща)
Університет технологій та економіки
імені Хелени Ходковської (Польща)
Донбаська державна машинобудівна академія

Студентське наукове товариство

IX МІЖНАРОДНА

студентська науково - технічна конференція

"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ПИТАННЯ"

24-25 квітня 2026 р.

(збірник тез конференції)

Тернопіль 2026

IX Міжнародна студентська науково - технічна конференція
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ НАУКИ. АКТУАЛЬНІ ПИТАННЯ"

УДК 621.326
Карпюк К. – ст. гр. СП-41
Тернопільський національний технічний університет імені Івана Пулюя

АНАЛІЗ ТА ВИЯВЛЕННЯ ЗАКОНОМІРНОСТЕЙ У ДАНИХ

Науковий керівник: д. ф.-м. н., професор, зав. кафедри Петрик М.Р.

Karpiuk K.
Ternopil Ivan Puluj National Technical Univercity

ANALYSIS AND PATTERN DETECTION IN DATA

Supervisor: Doctor of Physical and Mathematical Sciences, Professor, Head of the Department, M. R. Petryk

Ключові слова: аналіз даних, виявлення закономірностей, статистика.
Keywords: data analysis, pattern detection, statistics.

Дані стали важливим ресурсом у сучасному світі, який використовується для підтримки прийняття рішень у науці, бізнесі та техніці. Зростання обсягів інформації, що генерується різноманітними інформаційними системами, сенсорами та користувачами, зумовлює необхідність її ефективної обробки та аналізу. Однак самі по собі дані не мають цінності без їх належного аналізу та інтерпретації. Виявлення закономірностей дозволяє перетворити дані на корисну інформацію, що може бути використана для прогнозування, оптимізації процесів та підвищення ефективності діяльності. Таким чином, аналіз даних виступає важливим інструментом отримання нових знань і підтримки прийняття обґрунтованих рішень. Тому аналіз даних є одним із ключових напрямів сучасних досліджень.

Основу сучасного аналізу даних становлять статистичні методи, які дозволяють досліджувати структуру наборів даних, оцінювати характеристики змінних та визначати взаємозв'язки між ними. До таких методів належать описова статистика, кореляційний аналіз, регресійне моделювання та аналіз трендів. Кожен із цих підходів має свої особливості та обмеження: кореляційні методи дозволяють оцінити силу зв'язку між змінними, але не визначають причинно-наслідкові залежності; регресійні моделі забезпечують можливість прогнозування, проте значною мірою залежать від якості та повноти вхідних даних; методи згладжування дозволяють виявляти загальні тенденції, але можуть втрачати локальні особливості та короткострокові коливання. Окрім того, результати аналізу можуть бути чутливими до наявності шуму, викидів та неоднорідностей у даних. Тому комплексне використання декількох методів є необхідним для отримання більш повної та об'єктивної картини досліджуваних процесів [1].

Виявлення закономірностей у даних передбачає не лише застосування окремих методів, але й їх поєднання з метою отримання більш надійних та інтерпретованих результатів. Особливу роль відіграє порівняльний аналіз різних підходів, який дозволяє оцінити їх ефективність, стійкість до шуму та чутливість до змін обсягу вибірки. Зокрема, різні методи кореляційного аналізу можуть давати відмінні результати залежно від характеру залежності між змінними, а параметри моделей можуть суттєво впливати на точність та узагальнювальну здатність результатів. Важливим також є врахування масштабу даних та їх попередньої обробки, оскільки ці фактори

безпосередньо впливають на результати аналізу. Таким чином, дослідження поведінки методів аналізу є важливою складовою процесу виявлення закономірностей [2].

Практичне значення аналізу даних полягає у можливості його застосування для вирішення широкого кола задач, включаючи прогнозування, оптимізацію процесів та підтримку прийняття рішень. Методи виявлення закономірностей використовуються в економіці, медицині, соціальних науках, інженерії та інших галузях, де необхідно працювати з великими обсягами інформації. Вони дозволяють виявляти приховані залежності, оцінювати вплив різних факторів та формувати обґрунтовані висновки на основі даних. Подальший розвиток підходів до аналізу даних спрямований на підвищення точності, інтерпретованості та адаптивності методів, що дозволить більш ефективно використовувати дані як стратегічний ресурс.

Література:

1. Montgomery D. C., Runger G. C. Applied Statistics and Probability for Engineers, 7th Edition Evaluation Copy. Wiley, 2017. 848 p.

2. An Introduction to Statistical Learning / G. James et al. New York, NY : Springer US, 2021. URL: https://doi.org/10.1007/978-1-0716-1418-1 (date of access: 14.04.2026).

ДОДАТОК Б

Посилання на репозиторій GitHub

https://github.com/libertasss9/dyplomna