

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»  
Факультет інформатики та обчислювальної техніки  
(повна назва інституту/факультету)

Кафедра інформатики та програмної інженерії  
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_  
(підпис) Едуард ЖАРІКОВ  
(ім'я прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

## Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інженерія програмного забезпечення  
інформаційних систем»

спеціальності «121 Інженерія програмного забезпечення»

на тему: Програмне забезпечення збору та аналізу даних про російських  
військовослужбовців, що беруть участь у війні проти України

Виконав студент IV курсу, групи ІП-92  
(шифр групи)

Ткаченко Вікторія Вікторівна  
(прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Керівник доцент, к.т.н., доц., Олійник Ю.О.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Консультант  
з графічної  
документації

доцент, к.т.н., доц., Ліщук К. І.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Рецензент

\_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_  
(підпис)

Засвідчую, що у цій дипломній роботі  
немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ –2023

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 Інженерія програмного забезпечення

Освітньо-професійна програма – Інженерія програмного забезпечення  
комп'ютеризованих систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_  
(підпис)      Едуард ЖАРІКОВ  
(ім'я прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
**на дипломний проєкт студенту**

Ткаченко Вікторії Вікторівні

(прізвище, ім'я, по батькові)

1. Тема проєкту      Програмне забезпечення збору та аналізу даних про російських  
військовослужбовців, що беруть участь у війні проти України

керівник проєкту      доцент, к.т.н., доц., Олійник Ю.О.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «31» травня 2023 р. № 2101-с

2. Термін подання студентом проєкту « 19 » червня 2023 року

3. Вихідні дані до проєкту: технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення: загальні положення, змістовний  
опис і аналіз предметної області, аналіз успішних ІТ-проектів, постановка  
задачі.

2) Моделювання та конструювання програмного забезпечення: моделювання та  
аналіз програмного забезпечення, архітектура програмного забезпечення,  
конструювання програмного забезпечення, аналіз безпеки даних.

3) Аналіз якості та тестування програмного забезпечення: аналіз якості ПЗ,  
Опис процесів тестування, опис контрольного прикладу..

4) Впровадження та супровід програмного забезпечення: розгортання  
програмного забезпечення, робота з програмним забезпеченням.

5. Перелік графічного матеріалу

1) Схема структурна варіантів використань \_\_\_\_\_

2) Схема структурна бізнес-процесів \_\_\_\_\_

3) Схема структурна класів програмного забезпечення \_\_\_\_\_

4) Схема розгортання \_\_\_\_\_

6. Консультанти розділів проєкту

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

7. Дата видачі завдання «10» березня 2023 року \_\_\_\_\_

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення рекомендованої літератури	10.03.2023	
2	Аналіз існуючих методів розв'язання задачі	21.03.2023	
3	Постановка та формалізація задачі	26.03.2023	
4	Розробка інформаційного забезпечення	02.04.2023	
5	Алгоритмізація задачі	16.04.2023	
6	Обґрунтування вибору використаних технічних засобів	16.04.2023	
7	Розробка програмного забезпечення	23.04.2023	
8	Налагодження програми	30.04.2023	
9	Виконання графічних документів	15.05.2023	
10	Оформлення пояснювальної записки	15.05.2023	
11	Подання ДП на попередній захист	19.05.2023	
12	Подання ДП рецензенту	11.06.2023	
13	Подання ДП на основний захист	19.06.2023	

Студент

\_\_\_\_\_  
(підпис)

Вікторія ТКАЧЕНКО

\_\_\_\_\_  
(ініціали, прізвище)

Керівник

\_\_\_\_\_  
(підпис)

Юрій ОЛІЙНИК

\_\_\_\_\_  
(ініціали, прізвище)

## АНОТАЦІЯ

Пояснювальна записка дипломного проєкту складається з чотирьох розділів, містить 25 таблиць, 38 рисунків та 9 джерел – загалом 64 сторінки.

Дипломний проєкт присвячений розробці програмного забезпечення збору даних про російських військовослужбовців, що беруть участь у війні проти України.

Об'єкт дослідження: програмне забезпечення для збору та аналізу даних про російських військовослужбовців, що беруть участь у війні проти України.

Предмет дослідження: процеси розроблення програмного забезпечення, що проводить пошук та аналіз даних, його впровадження і супроводження.

Мета дипломного проєкту: збільшення ефективності пошуку та аналізу даних про російських військовослужбовців, які беруть участь у війні проти України.

У першому розділі проведено аналіз предметної області та вивчено існуючі аналоги програмного забезпечення. Були визначені функціональні та нефункціональні вимоги до застосунку.

У другому розділі описана архітектура програмного продукту. Були наведені фреймворки та бібліотеки, які вважаються оптимальними для даного проєкту. Також надано детальний опис основних компонентів програми, опис вхідних та вихідних даних.

Третій розділ присвячений методам тестування розробленого застосунку. Були наведені загальні описи етапів та процесів тестування, а також представлені результати проведених тестів.

Четвертий розділ дипломного проєкту розглядає процес розгортання та експлуатації розробленого програмного забезпечення, а також питання підтримки та обслуговування.

Ключові слова: PYTHON, MACHINE LEARNING, ВЕБ-ДОДАТОК JAVASCRIPT, БД, СХОВИЩЕ.

## **ABSTRACT**

The explanatory note of the diploma project consists of four sections, contains 25 tables, 38 figures and 9 sources – in total 64 pages.

The diploma project is dedicated to the development of software for collecting data on Russian military personnel participating in the war against Ukraine.

The purpose of the diploma project is to increase the efficiency of searching and analyzing data on Russian military personnel participating in the war against Ukraine.

Research object: software for data collection and analysis on Russian military personnel participating in the war against Ukraine.

Research subject: processes of developing software that conducts data search and analysis, its implementation, and maintenance.

The first chapter includes an analysis of the subject area and a study of existing analogs of the software. Functional and non-functional requirements for the application were determined.

The second chapter describes the architecture of the software product. Optimal frameworks and libraries for the project are provided. A detailed description of the main components of the program, as well as input and output data, is given.

The third chapter focuses on the methods of testing the developed application. General descriptions of testing stages and processes are provided, along with the presentation of test results.

The fourth chapter of the diploma project covers the deployment and operation of the developed software, as well as support and maintenance issues.

**Keywords:** PYTHON, MACHINE LEARNING, JAVASCRIPT, WEB-APPLICATION, DATABASE, STORAGE.



Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“    ” \_\_\_\_\_ 2023 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ЗБОРУ ТА АНАЛІЗУ ДАНИХ ПРО  
РОСІЙСЬКИХ ВІЙСЬКОВОСЛУЖБОВЦІВ, ЩО БЕРУТЬ УЧАСТЬ У  
ВІЙНІ ПРОТИ УКРАЇНИ**

**Технічне завдання**

КП.ІІ-9222.045430.01.91

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_ Юрій ОЛІЙНИК

Нормоконтроль:

\_\_\_\_\_ Катерина ЛІЩУК

Виконавець:

\_\_\_\_\_ Вікторія ТКАЧЕНКО

Київ – 2023

## ЗМІСТ

1	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ .....	3
2	ПІДСТАВА ДЛЯ РОЗРОБКИ .....	4
3	ПРИЗНАЧЕННЯ РОЗРОБКИ.....	5
4	ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	6
4.1	Вимоги до функціональних характеристик .....	6
4.1.1	Користувацького інтерфейсу.....	6
4.1.2	Для користувача:.....	7
4.2	Вимоги до надійності.....	7
4.3	Умови експлуатації .....	7
4.3.1	Вид обслуговування.....	8
4.3.2	Обслуговуючий персонал .....	8
4.4	Вимоги до складу і параметрів технічних засобів .....	8
4.5	Вимоги до інформаційної та програмної сумісності .....	8
4.5.1	Вимоги до вхідних даних .....	8
4.5.2	Вимоги до вихідних даних .....	9
4.5.3	Вимоги до мови розробки.....	9
4.5.4	Вимоги до середовища розробки .....	9
4.5.5	Вимоги до представленню вихідних кодів .....	9
4.6	Вимоги до маркування та пакування .....	9
4.7	Вимоги до транспортування та зберігання.....	9
4.8	Спеціальні вимоги.....	9
5	ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ .....	10
5.1	Попередній склад програмної документації.....	10
5.2	Спеціальні вимоги до програмної документації .....	10
6	СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	11
7	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ .....	12



# 1 НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Програмне забезпечення збору даних про російських військовослужбовців, що беруть участь у війні проти України.

Галузь застосування: військова інформатика та аналітика військових конфліктів.

Наведене технічне завдання поширюється на розробку програмного забезпечення збору даних про російських військовослужбовців, що беруть участь у війні проти України КП.ІП-9222.045440.01.81, котра використовується для збору та аналізу даних про російських військових. Даний дипломний проєкт може бути застосований у сферах розвідки, аналітики та прийняття рішень в контексті воєнного конфлікту.

					КП.ІП-9222.045430.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

## 2 ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки програмного забезпечення збору даних про російських військовослужбовців, що беруть участь у війні проти України є завдання на дипломне проектування, затверджене кафедрою інформатики та програмної інженерії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

					КПІ.ІП-9222.045430.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

### 3 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для автоматизації збору даних про російських військовослужбовців та полегшенню роботи з ними.

Метою розробки є збільшення ефективності пошуку та аналізу даних про російських військовослужбовців, які беруть участь у війні проти України.

					КПІ.ІП-9222.045430.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

Програмне забезпечення повинно забезпечувати виконання наступних основних функцій:

#### 4.1.1 Користувацького інтерфейсу

Дипломний проєкт надаватиме наступні функції:

- доступ до списку російських військовослужбовців;
- інфографіка, що включає інтерактивні візуалізації, такі як діаграми, графіки та карти, для кращого розуміння висновків, отриманих в результаті аналізу;
- застосування методів ML в режимі реального часу для аналізу даних про військовослужбовців.

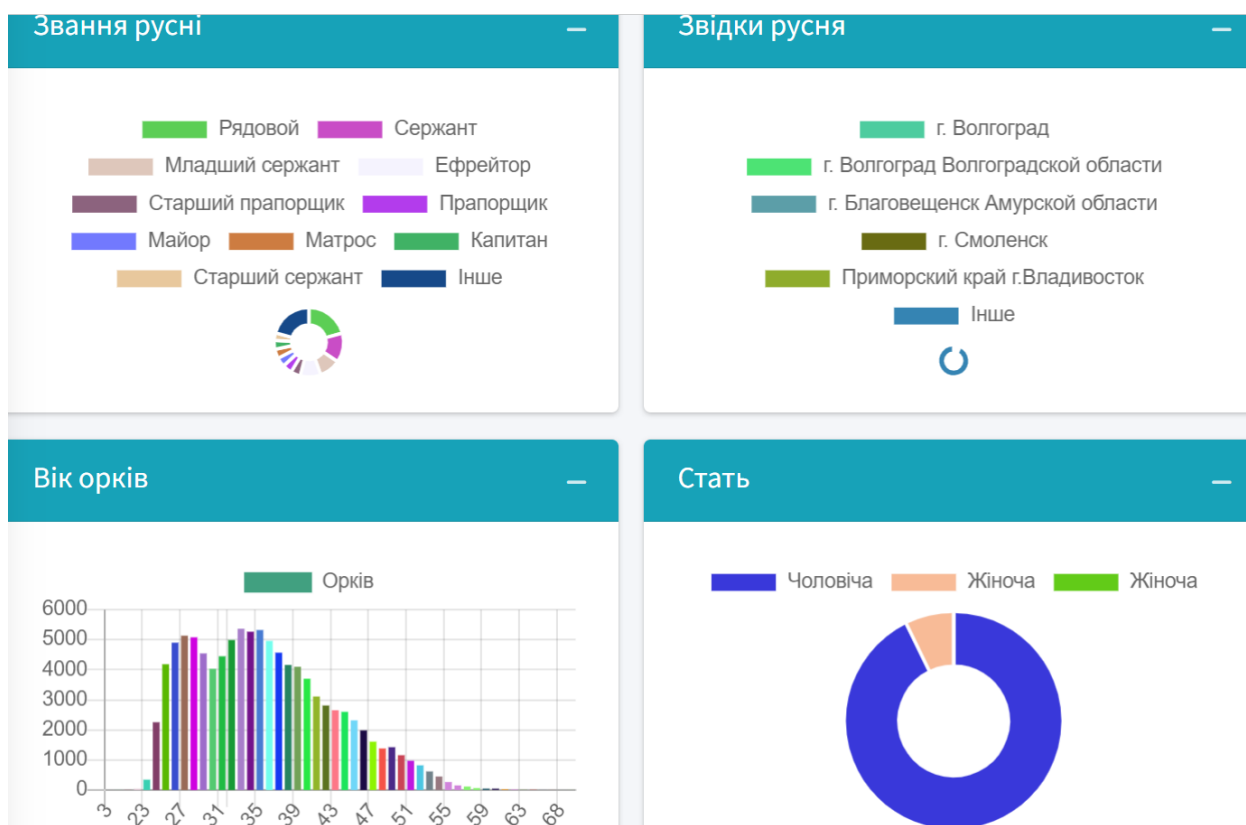


Рисунок 4.1 – Відображення інфографіки

ЯСТРУБ

ПРО НАС ГРАФІК СПИСОК АНАЛІЗ

СПИСОК ОРКІВ

«

«

1

2

3

4

5

6

7

8

9

10

—

220

»

»

№	ПІБ	Звання	Військова частина	Дата Народження	Більше Інформації
616	Козлов Иван Николаевич	Сержант	військова частина 51460	1984-04-21	Детальніше
639	Черняк Максим Эдуардович	Младший Сержант	військова частина 51460	1991-09-26	Детальніше
643	Щенников Сергей Олегович	Младший Сержант	військова частина 51460	1994-10-20	Детальніше
646	Кушнаренко Алексей Валерьевич	Младший Сержант	військова частина 51460	1994-08-02	Детальніше
647	Батоев Сергей Эрдынеевич	Младший Сержант	військова частина 51460	1992-10-27	Детальніше
661	Юшков Иван Алексеевич	Младший Сержант	військова частина 51460	1994-09-08	Детальніше
670	Ульянов Александр Сергеевич	Младший Сержант	військова частина 51460	1993-04-06	Детальніше
672	Урматов Андрей Максимович	Младший Сержант	військова частина 51460	1989-03-13	Детальніше

Рисунок 4.2 – Таблиця даних про російських військових

АНАЛІЗ					
<div> <div>ДОДАТИ ФАЙЛ</div> <div>ПРОВЕСТИ АНАЛІЗ</div> </div>					
<div> <div>«</div> <div>«</div> <div>1</div> <div>...</div> <div>10</div> <div>11</div> <div>12</div> <div>13</div> <div>14</div> <div>...</div> <div>20</div> <div>»</div> <div>»</div> </div>					
#	ПІБ	ЗВАННЯ	МІСЦЕ ПРОЖИВАННЯ	ДАТА НАРОДЖЕННЯ	
121	Попов Антон Павлович	Сержант	Немає інформації	1995-6-10	
122	Фаттахов Вагиф Нариман-оглы	Сержант	Немає інформації	1990-1-4	

Рисунок 4.3 – Завантажена та проаналізована таблиця даних про російських військових

#### 4.1.2 Для користувача:

- а) можливість перегляду таблиці з даними про військових;
- б) можливість завантажувати власні файли для аналізу;
- в) можливість перегляду результатів аналізу у вигляді візуалізованих графіків та діаграм.

#### 4.2 Вимоги до надійності

Передбачити контроль введення інформації та захист від некоректних дій користувача. Забезпечити цілісність інформації в базі даних.

#### 4.3 Умови експлуатації

Умови експлуатації згідно СанПін 2.2.2.542 – 96.

#### 4.3.1 Вид обслуговування

Вимоги до виду обслуговування не висуваються.

#### 4.3.2 Обслуговуючий персонал

Вимоги до обслуговуючого персоналу не висуваються.

#### 4.4 Вимоги до складу і параметрів технічних засобів

Мінімальна конфігурація технічних засобів:

- операційна система MacOS, Linux або Windows;
- версія NodeJS 12.18.3 та вище;
- версія NPM 7.1.1 та вище;
- текстовий редактор, бажано Microsoft VS Code.

#### 4.5 Вимоги до інформаційної та програмної сумісності

Вимоги до апаратної частини:

- комп'ютер, що має доступ до мережі інтернет;
- мінімальна роздільна здатність 800x600 та вище;
- процесор частотою не меншою за 1GHz.

Мінімальні вимоги для версій браузерів:

- Chrome 60+;
- Safari 10.1+ / iOS Safari 10.1+;
- Edge 12+;
- Firefox ESR+;
- Opera.

#### 4.5.1 Вимоги до вхідних даних

Вхідні дані повинні бути представлені в наступному форматі: таблиці з даними про військових у форматах .csv, .xlsx, .pdf.

					КПІ.ІП-9222.045430.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		8

#### 4.5.2 Вимоги до вихідних даних

Результати повинні бути представлені в наступному форматі: таблиці з даними про військових, графіки, що побудовані на основі цих даних.

#### 4.5.3 Вимоги до мови розробки

Розробку виконати на мові програмування Python для аналізу даних, та Node.js для розробки веб-застосунку.

#### 4.5.4 Вимоги до середовища розробки

Розробку виконати за допомогою середовища Microsoft VS Code.

#### 4.5.5 Вимоги до представлення вихідних кодів

Вихідний код програми має бути представлений у вигляді текстових файлів з розширенням .js для JavaScript або .py для Python.

#### 4.6 Вимоги до маркування та пакування

Вимоги до маркування та пакування не висуваються.

#### 4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються.

#### 4.8 Спеціальні вимоги

Розгорнути програмне забезпечення в Інтернеті для публічного доступу.

					КПІ.ІП-9222.045430.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		9

## 5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

### 5.1 Попередній склад програмної документації

У склад супроводжувальної документації повинні входити наступні документи на аркушах формату А4:

- а) пояснювальна записка;
- б) технічне завдання;
- в) текст програми;
- г) програма та методика тестування;
- д) керівництво користувача;
- е) керівництво програміста;
- ж) керівництво системного програміста.

Графічна частина повинна бути виконана на аркушах формату А3 та містити наступні документи:

- а) схема структурна варіантів використання;
- б) схема структурна бізнес-процесів;
- в) схема структурна класів програмного забезпечення;
- г) схема розгортання.

### 5.2 Спеціальні вимоги до програмної документації

Програмні модулі, котрі розробляються, повинні бути задокументовані, тобто тексти програм повинні містити всі необхідні коментарі.

					КПІ.ІП-9222.045430.01.91	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		10



## 6 СТАДІЇ І ЕТАПИ РОЗРОБКИ

№	Назва етапу	Строк	Звітність
1.	Вивчення літератури за тематикою проекту	21.02	
2.	Розробка технічного завдання	03.03	Технічне завдання
3.	Аналіз вимог та уточнення специфікацій	19.03	Специфікації програмного забезпечення
4.	Проектування структури програмного забезпечення, проектування компонентів	30.03	Схема структурна програмного забезпечення та специфікація компонентів (діаграма класів, схема алгоритму)
5.	Програмна реалізація програмного забезпечення	05.04	Тексти програмного забезпечення
6.	Тестування програмного забезпечення	10.04	Тести, результати тестування
7.	Розробка матеріалів текстової частини проекту	14.04	Пояснювальна записка
8.	Розробка матеріалів графічної частини проекту	20.04	Графічний матеріал проекту
9.	Оформлення технічної документації проекту	29.04	Технічна документація

## 7 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

					КПІ.ІП-9222.045430.01.91	Арк.
						12
Змін.	Арк.	№ докум.	Підп.	Дата.		

## **Пояснювальна записка до дипломного проєкту**

на тему: Програмне забезпечення збору та аналізу даних про російських  
військовослужбовців, що беруть участь у війні проти України

КП.ІП-9222.045430.01.81

Київ – 2023

## ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	6
1.1 Загальні положення.....	6
1.2 Змістовний опис і аналіз предметної області .....	10
1.3 Аналіз успішних ІТ-проектів.....	12
1.4 Розроблення функціональних вимог.....	17
1.5 Постановка задачі .....	24
Висновки до розділу .....	26
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	27
2.1 Моделювання та аналіз програмного забезпечення.....	27
2.2 Архітектура програмного забезпечення .....	28
2.3 Конструювання програмного забезпечення .....	29
2.4 Аналіз безпеки даних.....	40
Висновки до розділу .....	40
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..	42
3.1 Аналіз якості ПЗ.....	42
3.2 Опис процесів тестування .....	42
3.3 Опис контрольного прикладу.....	55
Висновки до розділу .....	56
4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .	57
4.1 Розгортання програмного забезпечення .....	57
4.2 Підтримка програмного забезпечення .....	59
4.3 Робота з програмним забезпеченням .....	59
Висновки до розділу .....	59
ВИСНОВКИ .....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	63
ДОДАТОК А ЗВІТ ПОДІБНОСТІ.....	64

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ML - Machine Learning - машинне навчання.

ETL - Extract, Transform, Load - витяг, перетворення та завантаження.

БД - База даних.

ПЗ - Програмне забезпечення.

AI - Artificial intelligence - штучний інтелект.

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

## ВСТУП

У сучасному світі, де швидкість та доступність інформації відіграють вирішальну роль, збір та аналіз даних стають невід'ємною складовою успішного функціонування та прийняття обґрунтованих рішень. Особливо важливою є ефективність цих процесів у сфері оборони та безпеки, де правильна та своєчасна інформація може мати вирішальний вплив на результати дій.

Ця область дослідження викликає особливий інтерес у зв'язку зі зростанням необхідності в ефективному зборі, аналізі та інтерпретації даних, пов'язаних з воєнним конфліктом. Забезпечення ефективної стратегії управління та прийняття рішень у цій сфері вимагає розробки та впровадження інноваційних методів та підходів до обробки цих даних.

Метою дипломного проєкту є збільшення ефективності пошуку та аналізу даних про російських військовослужбовців, які беруть участь у війні проти України. Це важлива проблема, яка вимагає комплексного підходу та високотехнологічних рішень задля забезпечення безпеки та захисту національних інтересів.

У рамках дипломного проєкту ми проведемо аналіз наявних рішень та перспектив розвитку в області збору та аналізу даних про військовослужбовців, а також розробимо власні методи та інструменти для поліпшення цих процесів. Буде досліджено сучасні технології збору даних, проведено розробку бази даних та розробку програмного забезпечення для ефективного аналізу цих даних.

Очікується, що результати проєкту допоможуть збільшити точність та швидкість збору інформації про військовослужбовців та сприятимуть більш глибокому розумінню їхніх дій та намірів. Це, у свою чергу, забезпечить більш об'єктивну оцінку ситуації, сприятиме прийняттю обґрунтованих рішень та забезпечить безпеку та захист інтересів нашої країни.

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

У цьому дипломному проєкті детально розглянуто кроки, методи та інструменти, які використовуються для досягнення мети. Буде проведено аналіз наявних рішень, розроблено власні методи та реалізовано програмне забезпечення, що дозволить здійснити збір, очищення та аналіз даних про російських військовослужбовців.

Результати дослідження та розробки мають великий потенціал для застосування в сфері оборони та безпеки, а також в інших сферах, де збір та аналіз даних відіграють важливу роль. Очікується, що вони стануть важливим внеском у покращення систем збору та аналізу даних, сприятимуть прийняттю обґрунтованих рішень та забезпечать ефективну захисту інтересів нашої країни. ПЗ може бути корисним для різних зацікавлених сторін, включаючи дослідників, аналітиків та організації, які займаються вирішенням цього важливого геополітичного конфлікту.

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

# 1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 1.1 Загальні положення

У зв'язку з повномасштабною війною росії проти України, в даний час тема аналізу даних про супротивника є особливо важливою.

Завдяки великій кількості висококваліфікованих фахівців в сфері кіберпростору, Україна має велику кількість інформації як про російських військовослужбовців, так і про об'єкти інфраструктури.

Цінність великої кількості персональних даних десятків тисяч російських військових, які брали участь у війні проти України, полягає насамперед у перспективі їх обробки та подальшого аналізу. Мета будь-якого аналізу даних – виявити та ідентифікувати закономірності в обсягах даних. Також важливо не тільки розпізнати закономірності, але й знайти їх причини.

Інформація про російських військовослужбовців дозволить встановити відповідність між особистими даними (такими як вік, регіон тощо) і військовим званням. Соціологи, психологи, політологи та інші експерти можуть далі інтерпретувати це листування разом з іншими статистичними даними, щоб шукати соціальні проблеми, які можуть бути чинниками цієї війни.

Щоб покращити обробку даних, необхідно використовувати ETL, що означає Extract, Transform, Load – процес, який широко використовується для інтеграції та зберігання даних. Він передбачає вилучення даних з різних джерел, їх перетворення відповідно до конкретних вимог і завантаження в цільову систему, таку як база даних або сховище даних. ETL складається з трьох етапів, що наведені нижче.

- а) Витяг: На цьому етапі дані витягуються з різнорідних джерел, таких як бази даних, електронні таблиці або файли.

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6



б) Перетворення: Після того, як дані витягнуті, вони піддаються перетворенню, щоб забезпечити їх послідовний і придатний для використання формат.

в) Завантаження: Після того, як дані були перетворені, вони завантажуються в цільову систему, таку як база даних або сховище даних.

ETL є критично важливим процесом в інтеграції та аналітиці даних, оскільки він дозволяє організаціям консолідувати та інтегрувати дані з різних джерел в уніфікований та узгоджений формат. Це гарантує, що дані є точними, повними та надійними для подальшого аналізу, звітності або прийняття рішень.

Процеси ETL можуть бути реалізовані за допомогою різних інструментів і технологій, починаючи від традиційних інструментів Extract, Transform, Load і закінчуючи сучасними платформами інтеграції даних і хмарними сервісами. Ці інструменти надають функціональні можливості для проєктування, планування та моніторингу робочих процесів ETL, а також для обробки складних перетворень даних і перевірки якості даних.

Загалом, ETL відіграє фундаментальну роль в інтеграції та зберіганні даних, дозволяючи організаціям ефективно видобувати, трансформувати та завантажувати дані, забезпечуючи їхню якість та придатність для подальшого використання в додатках та аналітиці.

Навчання з вчителем - це підхід до машинного навчання, коли алгоритм навчається на основі маркованих навчальних даних, щоб робити прогнози або приймати рішення. У цьому методі вхідні дані (також звані ознаками) супроводжуються відповідними цільовими мітками або виходами. Мета полягає в тому, щоб навчитися функції відображення, яка може передбачити правильний вихід, враховуючи нові, невидимі вхідні дані.

Алгоритми керованого навчання можна умовно поділити на регресію та класифікацію, що описано нижче.

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		7

- а) Регресія. Алгоритми регресії використовуються, коли цільова змінна є неперервною. Мета полягає в тому, щоб передбачити числове значення на основі вхідних даних. Приклади алгоритмів регресії включають лінійну регресію, регресію з використанням опорних векторів і регресію на основі дерева рішень.
- б) Класифікація. Алгоритми класифікації використовуються, коли цільова змінна є категоріальною або дискретною, а метою є віднесення вхідних даних до заздалегідь визначених класів або категорій. Поширені алгоритми класифікації включають логістичну регресію, дерева рішень, випадкові ліси та машини опорних векторів.

Некероване або навчання без вчителя - це підхід до машинного навчання, при якому алгоритм вивчає шаблони або структури з немаркованих даних. На відміну від навчання під наглядом, тут немає жодних цільових міток. Алгоритм досліджує внутрішню структуру даних, щоб виявити значущі закономірності, взаємозв'язки або групування.

Кластеризація - це поширений метод неконтрольованого навчання, який використовується для групування схожих точок даних. Деякі популярні алгоритми кластеризації включають кластеризацію за методом k-середніх, ієрархічну кластеризацію та DBSCAN - просторова кластеризація додатків з шумом на основі щільності[7].

Точність алгоритму визначається шляхом підрахунку кількості правильно класифікованих позитивних зразків (TP) та кількості неправильно класифікованих позитивних зразків (FP). Загальна формула для розрахунку precision (1.1) виглядає наступним чином:

$$\frac{TP}{TP+FP} \quad (1.1)$$

Де TP - кількість правильно класифікованих позитивних зразків, а FP - кількість неправильно класифікованих позитивних зразків.

Precision є метрикою, яка вимірює, наскільки точними є позитивні прогнози моделі. Вона визначається відношенням кількості правильно

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		8

класифікованих позитивних зразків до загальної кількості прогнозованих позитивних зразків. Формула для розрахунку точності:

Для обчислення точності, необхідно знати результати класифікації алгоритму і порівняти їх зі справжніми мітками класу. Зразки, які правильно були визначені як позитивні, вважаються True Positive (TP), тоді як зразки, які помилково були визначені як позитивні, вважаються False Positive (FP). За допомогою цих значень можна обчислити точність алгоритму.

Точність є важливою метрикою, особливо в задачах, де неправильна класифікація позитивного зразка може мати серйозні наслідки або де точність є більш пріоритетною, ніж повнота.

Звичайно, точність не є єдиною метрикою, і її слід оцінювати разом з іншими метриками, такими як повнота (recall), F1-середнє (f1-score) та support, для отримання більш повного уявлення про ефективність алгоритму класифікації.

Recall (повнота) вимірює, яку частину позитивних зразків модель змогла правильно ідентифікувати. Вона обчислюється як відношення кількості правильно класифікованих позитивних зразків до загальної кількості позитивних зразків. Формула для розрахунку повноти (1.2):

$$\frac{TP}{TP+FN} \quad (1.2)$$

де FN (False Negative) - кількість неправильно класифікованих негативних зразків.

F1-середнє (f1-score) є гармонічним середнім між точністю і повнотою. Воно враховує обидві метрики, і його значення відображає збалансовану ефективність моделі. F1-середнє обчислюється за формулою:

$$F1\text{-середнє} = 2 * (\text{точність} * \text{повнота}) / (\text{точність} + \text{повнота}).$$

$$\frac{2 \times (\text{точність} \times \text{повнота})}{\text{точність} + \text{повнота}} \quad (1.3)$$

Support відноситься до кількості зразків у кожному класі і використовується для підрахунку метрик. Він надає інформацію про розподіл зразків у різних класах.

## 1.2 Змістовний опис і аналіз предметної області

Аналіз даних про російських військовослужбовців має великий потенціал у різних сферах, включаючи військову стратегію, соціологію, психологію та політику. Він допомагає розкрити закономірності, виявити причини та встановити відповідності, що є важливими для розуміння ситуації та прийняття відповідних рішень. Однак перш ніж говорити про аналіз даних, варто провести їх збір та обробку.

Сьогодні збір даних хоч і не є проблемою, але даний процес досі залишається доволі затратним по часу. Варто зазначити, що у випадку з інформацією про військовослужбовців дані є персональними - і як було описано вище, їх використання дозволено в інтересах національної безпеки України, однак вони захищаються законом російської федерації, що означає, що отримати їх досить важко.

До того ж, отримана інформація про російських військовослужбовців може часто бути неповною, що призводить до неможливості використовувати для розв'язання задачі традиційні методи навчання з вчителем. Тож залишається актуальним питання виявлення ефективних методів інтелектуального аналізу зібраних даних.

Важливо зазначити, що вибір алгоритмів для конкретної задачі залежить від різних факторів, таких як характер набору даних, конкретна проблема, яку ви хочете вирішити, і вимоги до продуктивності. Експерименти та оцінка різних алгоритмів на вашому наборі даних допоможуть визначити найкращий підхід.

Повертаючись до проблеми з неповними даними, зазвичай всі неточності виправляються вручну людиною. При такому підході,

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		10

використовують бібліотеки, що містять методи для візуалізації та обробки даних, але оптимізація відбувається досить повільно

Іноді застосовують метод кластеризації як проміжний етап в аналізі даних. Тобто побудовані моделі для окремих кластерів, будуть більш точні, легкі й оптимізовані ніж ті, що використовуються для всіх даних разом.

Завдяки кластеризації можна більш якісно навчати моделі для класифікації. Вона дозволить зібрати й оцінити професійний та мотиваційний стан ворога, а проаналізувавши кількість солдатів в кожному кластері та дізнавшись їх приблизне місцезнаходження на фронті, можна більш ефективно вибудовувати тактику й використовувати ресурси.

Після успішної обробки даних зазвичай переходять до їх аналізу, що надає можливість для багатьох різних операцій, починаючи простим збором статистики, закінчуючи знаходженням різних колаборантів на нашій території та психологічним тиском на солдатів рф та на членів їх сім'ї.

Визначення найкращого алгоритму машинного навчання (ML) для конкретного набору даних вимагає аналізу різних факторів і експериментування з різними моделями. Для підбору підходящого алгоритму ML необхідно:

а) Розуміння задачі. Необхідно ретельно проаналізувати постановку задачі та мету, яку потрібно досягти. Які типи вхідних даних використовуються? Якого типу є вихідні дані - категоріальні, бінарні, числові? Чи є присутня залежність між вхідними та вихідними даними? Це допоможе зрозуміти, який тип задачі машинного навчання потрібен.

б) Розмір набору даних. Необхідно оцінити розмір набору даних. Якщо це великий набір даних, то моделі, які вимагають багато обчислювальних ресурсів або мають високу складність, можуть бути неефективними або недоцільними. У такому випадку, потрібно розглянути алгоритми, які швидко працюють з великими обсягами даних.

в) Типи функцій та відношень: Необхідно вивчити вхідні дані та встановити взаємозв'язки між ознаками. Якщо дані мають складні нелінійні відношення, то алгоритми, які використовують нелінійні моделі, такі як нейронні мережі або метод опорних векторів, можуть бути ефективними. Для лінійної залежності можна розглянути лінійну регресію або логістичну регресію.

г) Експериментування та оцінка моделей: На цьому етапі проводяться експерименти з різними алгоритмами ML та їх гіперпараметрами, оцінка ефективності кожної моделі за допомогою метрик якості, таких як точність, показник F1, середньоквадратична помилка і т.д. Необхідно порівняти результати та обрати алгоритм з найкращою продуктивністю для даного набору даних.

Підсумовуючи, визначення найкращого алгоритму ML є ітеративним процесом, який може залежати від унікальних особливостей вхідного набору даних. Тож чим більше проведено експериментів та чим більше буде оцінено різних моделей, тим краще можна зрозуміти, який алгоритм найкраще підходить для конкретної задачі.

### 1.3 Аналіз успішних ІТ-проектів

#### 1.3.1 RussianWar Criminals

На даний момент вже існують технічні рішення, що є в відкритому доступі. Один із них це проєкт RussianWar Criminals.[1] Він створений для ідентифікації воєнних злочинців, розроблений за співпраці Антикорупційного штабу України та командою незалежних професійних журналістів - агенцією «Слідство.Інфо». Представили даний проєкт у травні 2022 року. RussianWar Criminals має на меті систематизацію інформації про російських військовослужбовців в Україні та злочини, які вони скоїли.

У віртуальній онлайн-карті представлена детальна інформація про понад 138 тисяч військових, які належать до російської армії. Ця інформація включає

їх особисті дані, такі як ім'я, прізвище, дату народження та звання, а також відомості про приналежність до конкретних військових частин. Крім того, на карті також надаються дані про смерть та полон військовослужбовців. Завдяки цій інтерактивній онлайн-мапі користувачі можуть зручно переглядати та аналізувати інформацію про військових.

Для його створення була використана інформація з різних джерел, включаючи списки від Центру оборонних стратегій, Головного управління розвідки Міністерства оборони України, а також дані, отримані журналістами зі своїх джерел та бази даних МВС «Ищи своих».

Карта може бути використана для вивчення та аналізу військової ситуації. Також карту можна використовувати для спостереження за подіями, зв'язаними з російською армією. Деякі документи містили фотографії ідентифікованих російських військових і посилання на дані, що розкривають злочини.

Мета проєкту – систематизувати та каталогізувати доступну публічну інформацію про російську армію. Він допомагає журналістам та правоохоронним органам зібрати якомога більше даних про кожного російського військового злочинця.

Інтерактивна онлайн-карта «Російські військові злочинці» дозволяє користувачам знайти інформацію про військові частини російської армії. Ті підрозділи, які беруть участь у війні, виділені червоним кольором, а решта військових частин російської армії – зеленим.

Крім того, на мапі можна здійснювати пошук російських військових за прізвищем, ім'ям тощо. Після того як потрібний військовий знайдений, користувач може відкрити його профайл, де міститься інформація про його присутність в Україні (включаючи загибель або полон), скоєні воєнні злочини, якщо такі факти встановлені, а також посилання на джерела інформації та журналістські розслідування найавторитетніших українських редакцій. Розслідування розташовані у хронологічному порядку на окремій сторінці.

Важливо зазначити, що інформація на сайті проєкту постійно оновлюється і доповнюється. Головна мета проєкту полягає в задокументуванні кожного російського військового, що перебуває в Україні, і вчиненні всього можливого, щоб вони були притягнуті до відповідальності за свої злочини перед законом. Організатори вважають, що без накладання належних санкцій на воєнних злочинців, вони можуть повторювати свої звірства, і тому необхідно забезпечити, щоб жоден злочин не залишився безкарним.

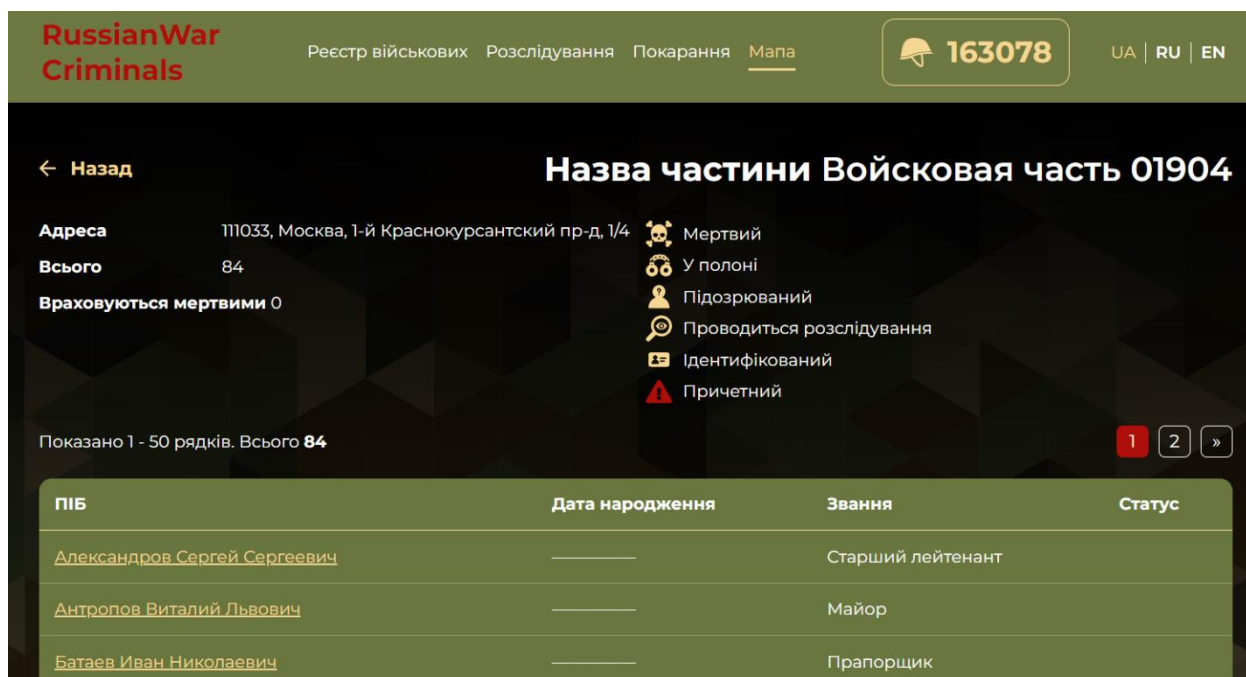


Рисунок 1.1 – Проєкт RussianWar Criminals

Ідентифікація злочинців та збір доказів - це важливі кроки в напрямку покарання військових, що скоюють злочини, та забезпечення безпеки та захисту прав людей. Організатори проєкту надіються, що залучення журналістів, активістів та правоохоронних органів допоможе зібрати якомога більше інформації про кожного російського воєнного злочинця і сприятиме приведенню їх до відповідальності.

Суспільна організація "Антикорупційний штаб" та агенція "Слідство.Інфо" прагнуть продовжувати свою роботу з ідентифікації злочинців, розкриття воєнних злочинів та приведення винних осіб до відповідальності. Вони активно співпрацюють зі своїми джерелами,



включаючи Центр оборонних стратегій, ГУР, РНБО та базу даних МВС "Шукай своїх". Крім того, журналісти з агенції "Слідство.Інфо" проводять свої власні розслідування та отримують інформацію з різних джерел.

### 1.3.2 InformNapalm

"InformNapalm" є міжнародною громадською ініціативою з вивчення російсько-українського конфлікту.[2] Проект, створений у 2014 році, присвячений аналізу інформації, що стосується воєнної діяльності, зокрема злочинів, вчинених російськими військовими на території України.

Вооружение и техника ВС РФ на Донбассе					
Вооружение и техника ВС РФ на Донбассе (по материалам расследований международного сообщества)					
Тип техники	Наименование	Место фиксации на Донбассе	Принадлежность к конкретному подразделению ВС РФ	Ссылки на OSINT-расследования и дополнительные материалы	
Броневедомобиль	ГАЗ-233014 "Тигр"	Луганская обл.	136-ая ОМСБр ВС РФ	<a href="https://informnapalm.org/23244-tigr-136-j-omsbr-dnr-lnr/">https://informnapalm.org/23244-tigr-136-j-omsbr-dnr-lnr/</a>	
Броневедомобиль	ГАЗ-39371 "Водник"	Сорокино (бывш. Краснодон, Луганская область)	не установлено	<a href="https://informnapalm.org/5087-general-na-bronevyke/">https://informnapalm.org/5087-general-na-bronevyke/</a>	
				<a href="https://informnapalm.org/">https://informnapalm.org/</a>	

Рисунок 1.2 – Проект InformNapalm

Головною метою "InformNapalm" є збір, аналіз та поширення відкритої інформації про російську військову присутність та дії в Україні. Члени проекту використовують різні джерела, такі як відкриті джерела інформації, соціальні медіа, власні розвідувальні дані та співробітництво з іншими організаціями та експертами.

У своїх матеріалах "InformNapalm" публікує факти про російських військових, їх розташування, використання спеціальної техніки та злочини, пов'язані з їхньою діяльністю. Вони також звертають увагу на пропагандистські дії та дезінформацію, які використовуються російськими органами влади.

InformNapalm має значний вплив на розкриття правди щодо російської агресії та воєнних злочинів, які відбуваються на території України. Їхні дослідження та докази широко використовуються журналістами, правозахисниками та міжнародними організаціями для документування порушень міжнародного прав.

### 1.3.3 Аналіз відомих програмних продуктів

Для порівняння роботи з аналогом можна скористатись таблицею 1.1.

Таблиця 1.1 – Порівняння з аналогами

Функціонал	Дипломний проект	RussianWar Criminals	InformNapalm	Пояснення
БД з інформацією про російських військовослужбовців	+	+	+	Наразі дипломний проект має найсвіжіші дані, в порівнянні з його аналогами
Інфографіка	+	містить лише інтерактивну мапу	містить інфографіку не по всій БД, а по окремим її категоріям	Дипломний проект доповнює функціонал обидвох аналогів

### Продовження таблиці 1.1

Аналіз даних за допомогою ML	+	-	-	Функціонал відсутній в аналогах, але присутній в дипломному проєкті
------------------------------	---	---	---	---

### 1.4 Розроблення функціональних вимог

Функціями програмного забезпечення є:

- доступ до списку військовослужбовців;
- перегляд інфографіки;
- пошук за ПІБ військового;
- завантаження файлу;
- приведення його до формату БД;
- застосування методів ML в режимі реального часу.

Більш детально функціонал можна побачити на рисунку 1.3.



Рисунок 1.3 – Діаграма варіантів використання

В таблицях 1.2 - 1.17 наведені варіанти використання програмного забезпечення.

Таблиця 1.2 – Варіант використання UC-O1

Use Case ID	UC-O1
Use Case Name	Переглянути список військовослужбовців
Actors	Користувач
Trigger	Користувач бажає переглянути список військовослужбовців.
Pre-conditions	Користувач перейшов на сторінку таблиці пошуку
Flow of Events	Відкривається таблиця. Користувач може переглядати список військовослужбовців.
Extension	-
Post-Condition	Користувач отримав список військовослужбовців.

Таблиця 1.3 – Варіант використання UC-O2

Use Case ID	UC-O2
Use Case Name	Переглянути інфографіку
Actors	Користувач
Trigger	Користувач бажає переглянути інфографіку.
Pre-conditions	Користувач перейшов на сторінку інфографіки.
Flow of Events	Відкривається сторінка інфографіки. Користувач може переглядати виведені графіки військовослужбовців.
Extension	При відсутності даних – інфографіка теж відсутня.
Post-Condition	Користувач має змогу переглядати інфографіку по даним, що є в базі даних.

Таблиця 1.4 – Варіант використання UC-O3

Use Case ID	UC-O3
Use Case Name	Шукати військовослужбовця за запитом
Actors	Користувач
Trigger	Користувач бажає знайти військовослужбовця за введеним запитом.

Продовження таблиці 1.4

Pre-conditions	Ввести пошуковий запит.
Flow of Events	Користувач натискає кнопку «Пошук». Відбувається формування та виведення таблиці результатів пошуку.
Extension	У випадку, якщо поле для запиту буде пустим, виведеться відповідне повідомлення.
Post-Condition	Користувач отримав список військовослужбовців, що відповідають введеному запиту.

Таблиця 1.5 – Варіант використання UC-O4

Use Case ID	UC-O4
Use Case Name	Переглянути інформацію про військовослужбовця
Actors	Користувач
Trigger	Користувач бажає переглянути більш детальну інформацію про військовослужбовця.
Pre-conditions	Користувач перейшов на сторінку таблиці пошуку.
Flow of Events	Відкривається таблиця. Користувач може обрати відповідне поле таблиці і натиснути кнопку «Детальніше». Після чого відкривається сторінка з детальною інформацією про відповідного військовослужбовця.
Extension	У випадку, якщо дані відсутні, виведеться відповідне значення..
Post-Condition	Користувач може переглядати детальну інформацію про відповідного військовослужбовця..

Таблиця 1.6 – Варіант використання UC-O5

Use Case ID	UC-O5
Use Case Name	Ввести пошуковий запит
Actors	Користувач

Продовження таблиці 1.6

Trigger	Користувач бажає знайти військовослужбовця за введеним запитом.
Pre-conditions	Користувач перейшов на сторінку таблиці пошуку та натискає на поле для запиту.
Flow of Events	Користувач може ввести запит для пошуку за ключовим словом.
Extension	У випадку, якщо поле для запиту буде пустим, виведеться відповідне повідомлення.
Post-Condition	Введено запит для пошуку.

Таблиця 1.7 – Варіант використання UC-О6

Use Case ID	UC-О6
Use Case Name	Провести аналіз
Actors	Користувач
Trigger	Користувач хоче провести аналіз.
Pre-conditions	-
Flow of Events	Користувач переходить на сторінку аналізу.
Extension	-
Post-Condition	Користувач може обирати параметри для аналізу даних.

Таблиця 1.8 – Варіант використання UC-О7

Use Case ID	UC-О6
Use Case Name	Здійснити класифікацію
Actors	Користувач
Trigger	Користувач хоче класифікувати військовослужбовців за вибраним параметром.

Продовження таблиці 1.8

Pre-conditions	Обрати параметр класифікації.
Flow of Events	Користувач натискає на кнопку «Аналіз». Відбувається генерація значення, виведення значення та точність згенерованої відповіді.
Extension	-
Post-Condition	Користувач може побачити результат аналізу відповідних військовослужбовців.

Таблиця 1.9 – Варіант використання UC-O8

Use Case ID	UC-O8
Use Case Name	Завантажити файл
Actors	Користувач
Trigger	Користувач хоче завантажити файл для подальшого аналізу.
Pre-conditions	Користувач на сторінці аналізу.
Flow of Events	Користувач натискає кнопку «Завантажити файл», обирає файл та натискає кнопку «Завантажити»
Extension	Файл має бути в розмірі .csv, при спробі завантажити інший розмір файлу, виводиться відповідне повідомлення.
Post-Condition	Файл завантажено.

Виходячи з наведених варіантів використання, ми можемо сформулювати наступні функціональні вимоги, наведені у таблиці 1.9.

Таблиця 1.10 – Функціональні вимоги:

ID	Назва	Опис
R1	Заповнення поля для запиту	Система повинна надавати можливість ввести запит за ключовим словом.

Продовження таблиці 1.10

R2	Перевірка довжини введеної фрази	Система повинна перевіряти довжину введеної фрази.
R3	Перевірка на пусте поле	Перед пошуком, запит обов'язково має бути введено.
R4	Виконання запиту .	Система повинна обробляти введену інформацію та надавати відповідний результат.
R5	Перевірка відповіді серверу	Система повинна обробляти відповідь серверної частини застосунку, та виводити відповідний результат.
R6	Перегляд виведеного результату	Система повинна надавати можливість переглядати результати запитів.
R7	Проведення аналізу	Система повинна надавати можливість аналізувати дані за вибраними параметрами.
R8	Завантаження файлу	Система повинна надавати можливість завантажувати файл.
R9	Перевірка формату файлу	Система повинна перевіряти введений формат файлу та надавати відповідний результат.

Відповідність між варіантами використання та визначеними функціональними вимогами можна зобразити у матриці трасування, яка наведена нижче в Таблиці 1.10.

Таблиця. 1.11 – Матриця трасування

	UC-O1	UC-O2	UC-O3	UC-O4	UC-O5	UC-O6	UC-O7	UC-O8
R 1								



Продовження таблиці 1.11

R 2								
R 3								
R 4								
R 5								
R 6								
R 7								
R 8								
R 9								

#### 1.4.1 Розроблення нефункціональних вимог

До програмного забезпечення збору даних про російських військовослужбовців, що беруть участь у війні проти України висунуті такі нефункціональні вимоги:

- а) модель класифікації/регресії, після тренування, повинна демонструвати точність не нижче 70% у визначенні класу для кожного елемента;
- б) вибрано українську мову для інтерфейсу;
- в) час, необхідний для внутрішніх обчислень та генерації відповіді, має бути менше 5 секунд;
- г) мінімальна роздільна здатність 800x600 та вище.

Мінімальні вимоги для версій браузерів:

- а) Chrome 60+;
- б) Safari 10.1+ / iOS Safari 10.1+;
- в) Edge 12+;
- г) Firefox ESR+;

д) Opera.

## 1.5 Постановка задачі

У цій роботі розглядається проблема аналізу даних про військовослужбовців, включаючи особисту інформацію, таку як освіта, звання та регіон походження, за допомогою різних методів машинного навчання. Метою є отримання цінної статистичної інформації та візуалізація результатів за допомогою веб-додатку.

За мету роботи покладено підвищення продуктивності процесу пошуку даних, а саме - зменшити час, структурувати знайдені дані за допомогою використання різних підходів та технік для аналізу знайденої інформації.

Поставлені наступні задачі:

- а) проектування та моделювання процесів збору та аналізу даних;
- б) пошук та виявлення нових ефективних методів аналізу даних;
- в) розробка програмного забезпечення збору та аналізу даних;
- г) оцінка ефективності запропонованого рішення.

Робота включає кілька ключових етапів. По-перше, буде проведено збір та підготовку даних. Це включає збір достовірних даних про солдатів з відповідних джерел та створення надійного сховища даних на СКБД PostgreSQL. Будуть реалізовані процеси вилучення, перетворення та завантаження (ETL) для очищення, інтеграції та перетворення даних у формат, придатний для аналізу.

Наступним важливим кроком є вибір оптимальних методів для класифікації, регресії та кластеризації. Для класифікації буде реалізовано та оцінено низку алгоритмів, включаючи логістичну регресію, дерева рішень, випадкові ліси, машини опорних векторів та нейронні мережі. Для оцінки моделей і визначення найбільш точного та інтерпретованого підходу будуть використовуватися такі показники ефективності, як точність, достовірність, F1 тощо.

Методи регресії, такі як лінійна регресія, регресія на основі опорних векторів, регресія на основі дерева рішень і регресія на основі нейронних мереж, будуть застосовані для прогнозування безперервних змінних у даних про солдатів. Для оцінки і вибору оптимальних регресійних моделей будуть використовуватися такі метрики, як середня квадратична помилка і R-квадрат.

Алгоритми кластеризації, включаючи K-середнє, ієрархічну кластеризацію, DBSCAN, будуть проаналізовані, після чого вибраний метод буде використаний для виявлення закономірностей і групування солдатів на основі схожості. Якість кластеризації буде оцінюватися за допомогою таких метрик, як оцінка силуету та сума квадратів всередині кластера.

Для ефективного представлення статистичних результатів буде розроблено веб-додаток з використанням Node.js.

Дипломний проєкт надаватиме наступні функції:

- а) доступ до списку російських військовослужбовців;
- б) інфографіка, що включає інтерактивні візуалізації, такі як діаграми, графіки та карти, для кращого розуміння висновків, отриманих в результаті аналізу;
- в) застосування методів ML в режимі реального часу для аналізу даних про військовослужбовців.

Запропонована робота має на меті надати змістовну інтерпретацію результатів, отриманих в результаті аналізу, сприяючи глибшому розумінню характеристик солдатів, включаючи їхній освітній рівень, звання, регіон та інші атрибути. Це дослідження може суттєво вплинути на процеси військового аналізу та прийняття рішень, надаючи цінну інформацію про досліджуваний набір даних.

## Висновки до розділу

Метою даного дипломного проєкту є збільшення пошукових засобів та засобів аналізу щодо збору даних про російських військовослужбовців, що беруть участь у війні проти України.

У даному розділі було здійснено комплексне дослідження існуючих аналогічних рішень та глибокий аналіз проблематики, що вимагала створення нового програмного забезпечення. На основі проведеного аналізу та обговорень було встановлено значну потребу в розробці даного проєкту.

Результати дослідження дозволили сформулювати чіткі та конкретні вимоги до програмного забезпечення. Це надало основу для постановки розумних та вимірюваних цілей проєкту, що сприяли ефективному виконанню завдань.

В рамках цього розділу були чітко визначені задачі, які були деталізовані та описані. Використання схем та таблиць сприяло зрозумілості, системності та структурованості процесу розробки та виконання проєкту.

Загалом, проведений аналіз, формулювання вимог та постановка задач стали надійною основою для успішного розвитку дипломного проєкту та його виконання відповідно до запланованих цілей та обмежень. Отримані результати дозволять ефективно розв'язати поставлені завдання та досягти мети проєкту.

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		26

## 2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Моделювання та аналіз програмного забезпечення

Для наочної та докладної демонстрації послідовності дій, необхідних для виконання процесу класифікації за допомогою програмного забезпечення, була обрана BPMN діаграма. Схему опису даного бізнес процесу можна побачити на рисунку 2.1:

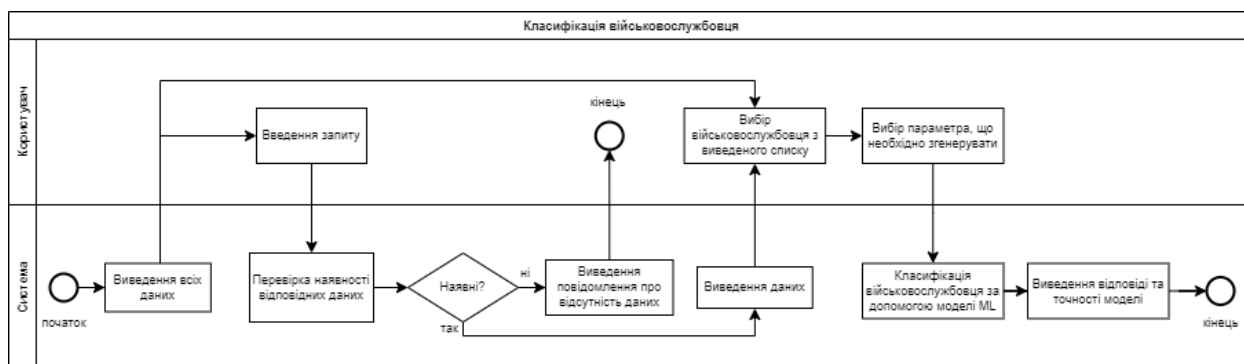


Рисунок 2.1 – Схема бізнес-процесу класифікації військових.

Опис послідовності класифікації військовослужбовця:

- користувач переходить на сторінку таблиці пошуку;
- система виводить дані про всіх військовослужбовців;
- якщо користувач вводить запит для пошуку конкретного військового, система перевіряє чи міститься інформація в базі даних і якщо інформація знайдена, то виводить відповідні значення, якщо ні – виводить відповідне повідомлення, про відсутність даних в системі;
- користувач обирає поле з відповідним військовим, після чого переходить на сторінку аналізу або на сторінку детальної інформації про цього військового;
- при переході на сторінку детальної інформації система виводить всю наявну інформацію про військового, користувач може перейти на сторінку аналізу;

- е) при переході на сторінку аналізу даних, користувач обирає параметр, що необхідно згенерувати;
- ж) система класифікує військовослужбовця, виводить результат обчислень та їх точність.

## 2.2 Архітектура програмного забезпечення

Програмне забезпечення складається з кількох модулів.

Перший модуль займається аналізом вхідних даних. При використанні системи кінцевим користувачем з даного модуля задіюється лише натренована модель ML для класифікації військовослужбовця.

Другий модуль займається створенням коду для фінального додатку. Для полегшення організації коду сторінок та елементів майбутнього веб-додатку використано шаблонізатор.

Враховуючи вищеописані модулі було прийнято рішення обрати монолітну архітектуру для проєкту. Монолітна архітектура надає наступні переваги:

- а) простота розробки та розгортання;
- б) простіше масштабування;
- в) зручність налагодження та тестування;
- г) менші накладні витрати.

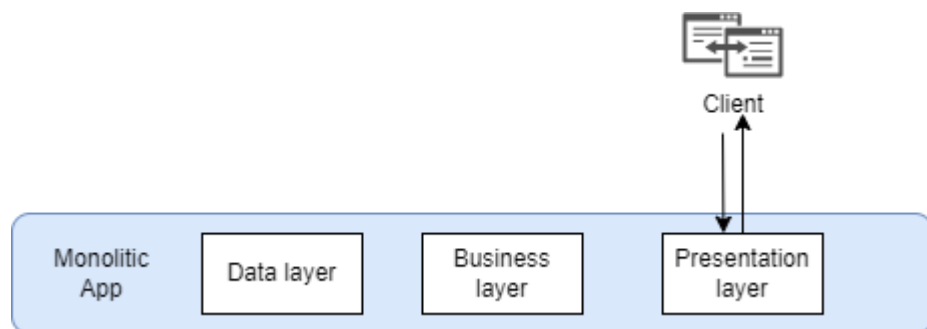


Рисунок 2.2 – Архітектура програмного забезпечення.

## 2.3 Конструювання програмного забезпечення

### 2.3.1 Вхідні дані

Вхідні дані грають ключову роль в вирішенні даної задачі. Для виконання проєкту було обрано 29 різних джерел даних, серед них:

- а) сайт Генерального Управління розвідкою України;
- б) сайт організації Anonymous;
- в) платформа Миротворець.

Для проектування та подальшої роботи з сховищем даних потрібно врахувати комплекс методів, які реалізують процес переносу початкових даних в аналітичний формат, який буде підтримуватись в сховищі даних та не порушуватиме цілісність системи. Для цього опишемо основні функції ETL процесів, що включають:

- а) процес завантаження;
- б) процес валідації даних;
- в) перетворення даних.

Після отримання даних з різних джерел і виконання перетворення, агрегації та очищення даних відбувається завершальний етап ETL, який полягає в завантаженні даних у сховище.

Процес завантаження передбачає передачу даних із проміжних таблиць у структури зберігання даних. Тому для прискорення завантаження даних необхідно заздалегідь перевірити їх цілісність і коректність, а потім розробити і вибрати найбільш оптимізовану процедуру зберігання даних. Під час завантаження даних у сховище переноситься не вся інформація з джерела, а лише інформація, яка змінилася протягом перехідного періоду з моменту останнього завантаження.

Для заповнення сховища даних російських військовослужбовців було реалізовано наступне:

- а) Для транспортування даних до Stage зони:

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		29

- 1) Мовою Python було реалізовано функцію яка перетворює файли xlsx та pdf у csv;
  - 2) Засобами DataGrip було транспортовано дані з файлів CSV до таблиць;
  - 3) Засобами SQL було здійснено форматування даних до єдиного формату.
- б) Отримано дані з Stage зони для заповнення сховища даних, за допомогою засобів SQL.
- в) Засобами мови SQL були очищені деякі поля головної таблиці. Також була очищена дата, записана у некоректному форматі і була приведена до єдиного формату YYYY-MM-DD.

Для підготовки тренування моделей машинного навчання було експортовано дані з бази даних у наступному вигляді

	#	orc_id	fio	age	bdate	bplace	nationality	gender	mstatus	address	...	description	mstatus_cl	edu_lvl_cl	mil_rank_cl	bdate_cl	mil_date_cl	pass_date_cl	contract_date_cl
0	1	439	Абрамцов Владимир Сергеевич	30.0	1991-08-19	NaN	NaN	m	NaN	NaN	...	NaN	NaN	NaN	Сержант	1991-08-19	NaN	2015-08-04	NaN
1	2	452	Фаттахов Вагиф Нариман-оглы	32.0	1990-04-01	NaN	NaN	m	NaN	NaN	...	NaN	NaN	NaN	Сержант	1990-04-01	NaN	2010-05-13	NaN
2	3	456	Тушевский Иван Георгиевич	25.0	1996-02-09	NaN	NaN	m	NaN	NaN	...	NaN	NaN	NaN	Сержант	1996-02-09	NaN	2010-03-11	NaN
3	4	458	Мезенин Виктор Вячеславович	27.0	1994-12-15	NaN	NaN	m	NaN	NaN	...	NaN	NaN	NaN	Сержант	1994-12-15	NaN	2009-03-05	NaN
4	5	97567	Худяк Денис Сергеевич	33.0	1989-02-06	NaN	NaN	m	NaN	NaN	...	NaN	NaN	NaN	NaN	1989-02-06	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
110067	110068	99495	Черненко Константин Игоревич	33.0	1988-08-28	NaN	NaN	m	NaN	NaN	...	NaN	NaN	NaN	NaN	1988-08-28	NaN	NaN	NaN
110068	110069	109315	Арцинеков Вячеслав Владимирович	NaN	1982-09-23	NaN	NaN	m	NaN	NaN	...	NaN	NaN	NaN	Матрос	1982-09-23	NaN	2015-06-29	NaN
110069	110070	109320	Мирошниченко Александр Анатольевич	NaN	1985-02-28	NaN	NaN	m	NaN	NaN	...	NaN	NaN	NaN	NaN	1985-02-28	NaN	NaN	NaN
110070	110071	109686	Сорокин Дмитрий Андреевич	NaN	1991-08-06	NaN	NaN	m	NaN	NaN	...	NaN	NaN	NaN	NaN	1991-08-06	NaN	NaN	NaN
110071	110072	109919	Мелентьев Александр Александрович	NaN	1988-09-18	NaN	NaN	m	NaN	NaN	...	NaN	NaN	NaN	Ефрейтор	1988-09-18	NaN	2008-12-22	NaN

Рисунок 2.3 – Приклад початкових даних

Спочатку було прийнято рішення про необхідність створення колонки регіон, яка формується за допомогою наступник колонок: місця народження, адресу реєстрації, адреси проживання, адреси видачі паспорту. Таким чином ми отримали більше рядків, які вміщують регіон, для того, щоб в подальшому можна було використовувати регіон при аналізі.



Після цього було необхідно позбутися від колонок, які заповнені на 60 та менше відсотків, а також дані, які не впливають на аналіз - у результаті отримали наступні дані

	age	region	nationality	gender	mstatus_cl	mil_rank_cl	edu_lvl_cl
0	30.0	Хабаровский край	nan	m	NaN	Сержант	NaN
1	32.0	Амурская область	nan	m	NaN	Сержант	NaN
2	25.0	Хабаровский край	nan	m	NaN	Сержант	NaN
3	27.0	Республика Бурятия	nan	m	NaN	Сержант	NaN
4	33.0	Республика Адыгея	nan	m	NaN	NaN	NaN
...	...	...	...	...	...	...	...
110067	33.0	Республика Адыгея	nan	m	NaN	NaN	NaN
110068	NaN	Кабардино-Балкарская Республика	nan	m	NaN	Матрос	NaN
110069	NaN	Республика Адыгея	nan	m	NaN	NaN	NaN
110070	NaN	Республика Адыгея	nan	m	NaN	NaN	NaN
110071	NaN	Республика Бурятия	nan	m	NaN	Ефрейтор	NaN

Рисунок 2.4 – Дані після видалення непотрібних колонок

Далі було необхідно перевести дані колонок зі званнями військовослужбовців та їх рівнем освіти у числові значення, а також очистити колонку національність від дублікатів, та значень, які були записані з орфографічними помилками. Як результат отримали наступні дані:

	age	region	nationality	gender	mstatus_cl	edu_lvl	mil_rank
0	30.0	Хабаровский край	nan	m	NaN	0	7
1	32.0	Амурская область	nan	m	NaN	0	7
2	25.0	Хабаровский край	nan	m	NaN	0	7
3	27.0	Республика Бурятия	nan	m	NaN	0	7
4	33.0	Республика Адыгея	nan	m	NaN	0	0
...	...	...	...	...	...	...	...
110067	33.0	Республика Адыгея	nan	m	NaN	0	0
110068	NaN	Кабардино-Балкарская Республика	nan	m	NaN	0	2
110069	NaN	Республика Адыгея	nan	m	NaN	0	0
110070	NaN	Республика Адыгея	nan	m	NaN	0	0
110071	NaN	Республика Бурятия	nan	m	NaN	0	3

Рисунок 2.5 – Дані після переведення деяких колонок у числові значення

Після даного етапу було прийняте рішення про необхідність використання One-hot encoding для того, щоб позбутися всіх категоріальних значень, натомість отримавши числові значення.

	age	edu_lvl	mil_rank	region_Алтайский край	region_Амурская область	region_Архангельская область	region_Астраханская область	region_Белгородская область	region_Брянская область	region_Владимирская область	...	nationality_украинец
0	30.0	0	7	0	0	0	0	0	0	0	...	0
1	32.0	0	7	0	1	0	0	0	0	0	...	0
2	25.0	0	7	0	0	0	0	0	0	0	...	0
3	27.0	0	7	0	0	0	0	0	0	0	...	0
4	33.0	0	0	0	0	0	0	0	0	0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
110067	33.0	0	0	0	0	0	0	0	0	0	...	0
110068	NaN	0	2	0	0	0	0	0	0	0	...	0
110069	NaN	0	0	0	0	0	0	0	0	0	...	0
110070	NaN	0	0	0	0	0	0	0	0	0	...	0
110071	NaN	0	3	0	0	0	0	0	0	0	...	0

Рисунок 2.6 – Дані після використання One-hot encoding

Також необхідно було позбутися пустих значень, для цього було використано метод K-Nearest Neighbors, який заповнює пропущені значень для кожної вибірки, використовуючи для цього своїх сусідів. Після даного етапу було отримано фінальний набір даних, який буде використовуватися для аналізу.

	age	edu_lvl	mil_rank	region_Алтайский край	region_Амурская область	region_Архангельская область	region_Астраханская область	region_Белгородская область	region_Брянская область	region_Владимирская область	...	nationality_украинец
0	30.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
1	32.0	0.0	7.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0
2	25.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
3	27.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
4	33.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...
110067	33.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
110068	26.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
110069	34.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
110070	34.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
110071	32.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0

Рисунок 2.7 – Дані підготовлені для кластеризації

### 2.3.2 Вихідні дані

На основі детального опису та проведеного аналізу предметної області було розроблено модель сховища даних російських військових. Модель сховища за типом зірки - відформатованої до однієї таблиці.

В таблиці наведені виміри дат, сімейних статусів, військових звань та рівня навчання.

```

CREATE TABLE soldier (
  id integer NOT NULL,
  fio character varying(100) NOT NULL,
  age character varying(20),
  bdate character varying(20),
  bplace character varying(100),
  nationality character varying(100),
  gender character varying(20) DEFAULT 'm',
  mstatus character varying(20),
  address character varying(250),
  reg_address character varying(200),
  tel character varying(20),
  inn character varying(20),
  bank character varying(512),
  vk character varying(512),
  sender character varying(100),I
  recipient character varying(100),
  tab_num character varying(20),
  v_ch character varying(100),
  branch character varying(100),
  mil_post character varying(1000),
  nickname character varying(100),
  mil_rank character varying(100),
  lnum character varying(20),
  battalion character varying(100),
  rota character varying(100),
  mil_ser character varying(20),
  mil_num character varying(20),
  mil_date character varying(20),
  mil_authority character varying(100),
  contract_end_date character varying(20),
  contract_auth character varying(100),
  contract_num character varying(20),
  contract_date character varying(20),
  edu_lvl character varying(64),
  study_place character varying(1000),
  specialty character varying(100),
  pass_ser character varying(20),
  pass_num character varying(20),
  pass_date character varying(20),
  pass_auth character varying(128),
  pass_code character varying(20),
  family character varying(512),
  children character varying(20),
  source character varying(100),
  photolink character varying(200),
  load_date datetime2(6) DEFAULT CURRENT_TIMESTAMP,
  description text,
  created_by_admin character varying,
  updated_by_admin character varying
);

```

Рисунок 2.8 – Схема сховища даних російських військових

На таблиці 2.1 видно детальний опис полів:

Таблиця. 2.1 – Опис полів таблиці Soldiers

ПІБ	fio
Вік	age
Дата народження	bdate
Місце народження	bplace
Національність	nationality
Стать	gender
Сімейний стан	mstatus
Адреса фактичного проживання	address
Дані про реєстрацію за місцем проживання	reg_address
Номер контактного телефону	tel
ПІН	inn

Продовження таблиці 2.1

Банківські реквізити	bank
Сторінка ВК	vk
Відправник	sender
Одержувач	recipient
Таб.№	tab_num
Військова частина	v_ch
Підрозділ	branch
Найменування військової посади	mil_post
Позивний	nickname
Військове звання	mil_rank
Особистий номер	lnum
Батальйон	battalion
Рота	rota
Серія (військовий квиток)	mil_ser
Номер (військовий квиток)	mil_num
Дата видачі (військовий квиток)	mil_date
Орган, який видав військовий квиток	mil_authority
Закінчення контракту	contract_end_date
Чий наказ	contract_auth
№ наказу	contract_num
Дата наказу	contract_date
Тип освіти	edu_lvl
Освітня установа	study_place
Спеціальність	specialty
Серія (паспорт)	pass_ser
Номер (паспорт)	pass_num
Дата видачі (паспорт)	pass_date
Орган, який видав паспорт	pass_auth
Код підрозділу	pass_code

### Продовження таблиці 2.1

Склад сім'ї	family
К-сть дітей	children
Ресурс	source
Дата завантаження	load_date

На основі вище описаних даних, була сконструйована база даних:

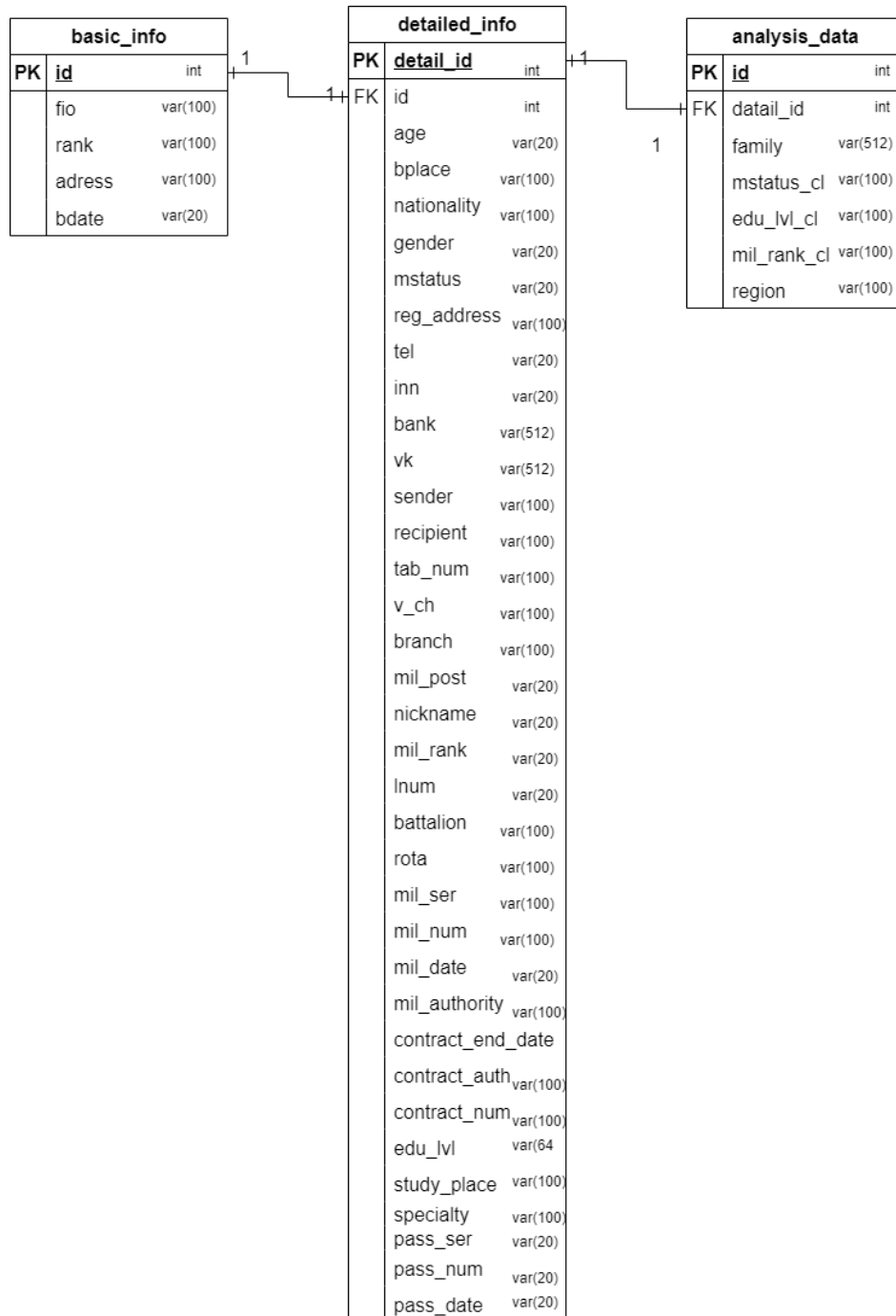


Рисунок 2.9 – Діаграма бази даних

### 2.3.3. Конструювання веб-додатку

Першим етапом є розробка надійної системи баз даних. Для задоволення цієї вимоги Postgre SQL використовуватиметься як система керування базою даних. Postgre SQL надає широкий спектр функцій, які добре підходять для ефективної обробки великих наборів даних.

Розроблена БД буде призначена для зберігання інформації про солдатів з належною організацією та інтеграцією даних. База даних буде структурована таким чином, щоб ефективно вміщувати дані про солдатів і забезпечувати їх гармонійну організацію та зв'язок між різними даними..

Наступний етап передбачає розробку веб-додатку з використанням Node.js для серверної частини додатку, та фреймворком React для створення інтерфейсу. Для аналізу даних про солдатів буде використовуватися Python як основна мова програмування.

Python[10] – інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією.[8] Мова програмування Python надає підтримку модулів і пакетів модулів, що сприяє створенню модульних структур і полегшує повторне використання коду. Інтерпретатор Python та стандартні бібліотеки доступні у форматі як скомпільованого коду, так і вихідного коду на різних платформах.

Python має важливу перевагу - він надає широку підтримку для роботи з даними шляхом використання різноманітних бібліотек. Під час розробки моделі були використані наступні пакети та бібліотеки:

а) Pandas[2] є пакетом Python з відкритим вихідним кодом, який пропонує зручні та високопродуктивні структури даних, а також інструменти для аналізу даних за допомогою мови програмування Python.

б) NumPy[3] – пакет для обробки масивів загального призначення.

Python забезпечує можливість працювати з високопродуктивними

багатовимірними об'єктами масиву та надає потужні інструменти для роботи з ними. Один з таких інструментів - це NumPy, який є ефективним контейнером для зберігання та маніпуляцій загальних багатовимірних даних.

- в) Matplotlib[4] – це бібліотека графіків для Python, яка надає об'єктно-орієнтований API для вбудовування графіків у програми.
- г) Scikit-Learn[1] – це надійна бібліотека машинного навчання для Python, яка надає функціональність для створення та тренування різноманітних алгоритмів класифікації, регресії та кластеризації, зменшення розмірності, вибору моделі, підготовки вхідних даних тощо. Вона надає цілий ряд алгоритмів навчання з вчителем і без вчителя через узгоджений інтерфейс на Python.

Для розробки основної логіки проекту використано Node.js та фреймворк React для інтерфейсу. Node.js[8] — це середовище виконання, яке дозволяє створювати масштабовані та ефективні серверні програми за допомогою JavaScript. Він пропонує такі переваги:

- а) за допомогою Node.js як інтерфейс, так і бек-енд програми можна написати на JavaScript, що спрощує розробку та покращує повторне використання коду.
- б) Node.js використовує V8 JavaScript, який забезпечує чудову продуктивність і дозволяє програмі ефективно обробляти велику кількість одночасних з'єднань.
- в) Node.js має величезну екосистему пакетів з відкритим кодом, доступних через npm (Node Package Manager), що дозволяє легко інтегрувати сторонні бібліотеки та модулі у проект.

React — популярна бібліотека JavaScript для створення інтерфейсів користувача. Він забезпечує такі переваги:

- а) React сприяє модульному та багаторазовому компонентному підходу до створення інтерфейсів користувача. Це полегшує керування кодовою базою та підтримку її.
- б) React використовує віртуальне представлення DOM, що покращує продуктивність за рахунок мінімізації фактичних маніпуляцій з DOM. Він ефективно оновлює та відтворює лише необхідні компоненти, що забезпечує швидшу та плавну роботу користувача.
- в) React слідує односпрямованому шаблону потоку даних, що полегшує відстеження та керування змінами даних. Це покращує передбачуваність коду та допомагає уникнути типових проблем із синхронізацією даних.

Як було описано раніше, робота складається з двох основних компонентів. Також окремим компонентом можна виділити модуль, що відповідає за інтерфейс додатку.

Умовно можна розбити проєкт на наступні сутності:

- а) Аналіз – містить вже натреновані моделі ML, за допомогою яких відбувається аналіз даних;
- б) Бізнес логіка – містить основну логіку веб-додатку, відповідає за зв'язок між модулем аналізу, модулем інтерфейсу. Визначає як саме додаток реагує на дії користувача;
- в) UI – відповідає за зовнішній вигляд додатку, взаємодію з користувачем тощо.

Деталізацію описаних компонентів можна розглянути на рисунку 2.4 та 2.5:



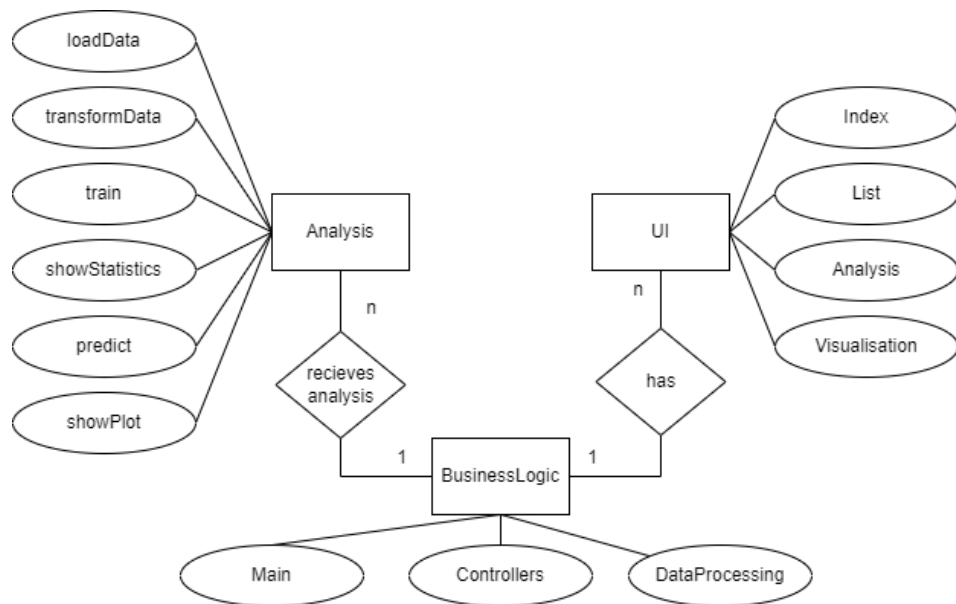


Рисунок 2.10 – ER діаграма сутностей Analysis, UI, Business Logic

Логіка проєкту складається з наступних класів:

- а) Program – клас, що відповідає за запуск програми;
- б) BusinessLogic – клас, що відповідає за основну логіку веб-проєкту, а саме маніпулювання даними, показ діаграм, таблиць, запуск аналізу даних і функціонування інших елементів веб-застосунку;
- в) Analysis – клас, що обробляє дані перед їх опрацюванням ML моделі, запускає модель, обробляє та виводить її прогноз, виводить статистику, тощо;
- г) Classification – клас, що містить модель класифікації, її основні функції такі як навчання, прогнозування, тощо;
- д) Regression – клас, що містить модель регресії, її основні функції такі як навчання, прогнозування.

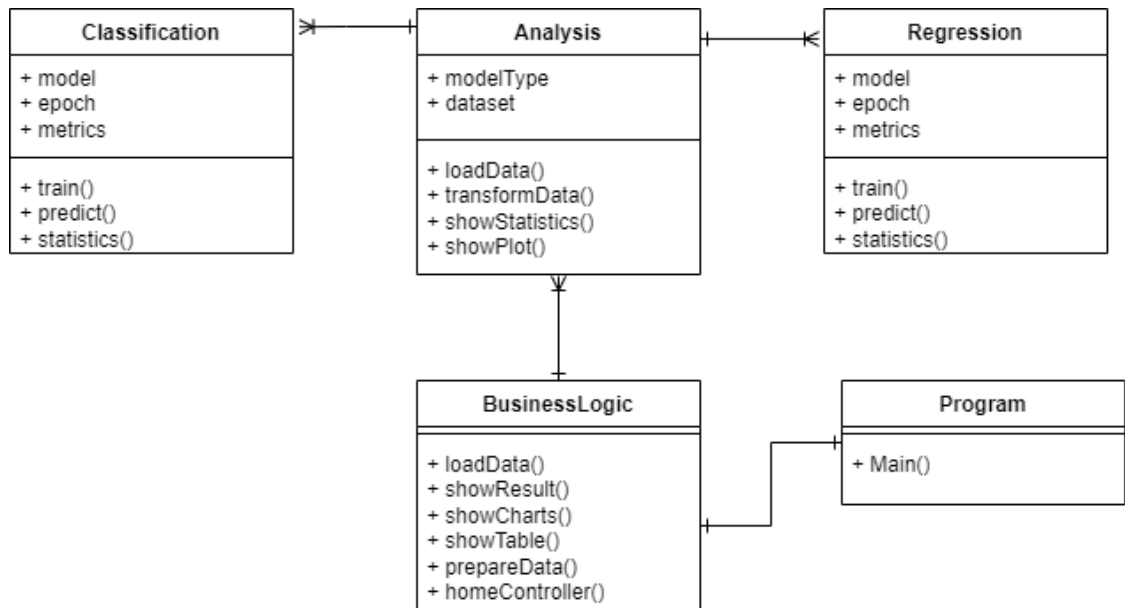


Рисунок 2.11 – Схема структурна класів програмного забезпечення

## 2.4 Аналіз безпеки даних

Будучи локальною програмою, яка не збирає жодних даних, наша система гарантує повну безпеку і конфіденційність користувача. Важливо зазначити, що весь наш код відкритий і доступний для перевірки будь-яким зацікавленим особам. Це сприяє прозорості та довірі до нашого програмного забезпечення.

### Висновки до розділу

В даному розділі було розглянуто наступні етапи розробки проєкту:

- підготовка даних, створення сховища для них та завантаження, з використанням ETL для пришвидшення даного процесу;
- підготовка моделей для аналізу даних, а саме підбір найкращих методів, їх тренування;
- розробка веб-додатку для візуалізації даних, взаємодії користувача з даними та їх аналізом за допомогою навчених моделей ML.

Зважаючи на вищеописані етапи розробки, було виділено декілька сутностей – Analysis, UI, Business Logic, їх деталізацію наведено в ER діаграмі та діаграмі класів.

Оскільки веб-додаток є відносно невеликим, було обрано монолітну архітектуру та мову програмування Node.js. При розробці на Node.js із React, можна використовувати сильні сторони обох технологій для створення масштабованої, продуктивної та модульної веб-програми. Node.js забезпечує надійне серверне середовище, а React дає змогу ефективно створювати динамічні та інтерактивні інтерфейси користувача.

Було обрано Python для розробки моделей машинного навчання через наявність широкого спектру бібліотек, спеціалізованих на цій галузі програмування. Мова програмування Python пропонує розгалужену екосистему інструментів та бібліотек, які сприяють зручному та ефективному створенню, навчанню та використанню моделей машинного навчання.

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		41

### 3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Аналіз якості ПЗ

Аналіз якості програмного забезпечення є невід'ємною складовою процесу розробки та впровадження програмного забезпечення. Цей аналіз дозволяє виявляти потенційні проблеми, помилки та недоліки, що можуть впливати на функціональність, надійність та ефективність програми.

Одним з методів аналізу якості програмного забезпечення є white-box (білоблочний) тестування. Цей підхід базується на детальному вивченні внутрішньої структури програми та алгоритмів, що використовуються. В рамках white-box тестування проводяться перевірки на відповідність заданим критеріям та правильність виконання коду.

Іншим методом є black-box (чорноблочний) тестування, яке спрямоване на перевірку функціональності програми без прямого доступу до її внутрішньої реалізації. В цьому випадку тестування здійснюється на основі специфікацій, вимог користувача та очікуваного поведінки програми.

Обидва методи, white-box та black-box, використовуються для забезпечення якості програмного забезпечення та виявлення потенційних проблем. Правильне використання цих методів дозволяє розробникам та тестувальникам ефективно оцінювати, валідувати та вдосконалювати програмне забезпечення з метою досягнення високої якості, надійності та задоволення потреб користувачів.

#### 3.2 Опис процесів тестування

З урахуванням вищезазначених принципів можна виокремити необхідні та достатні умови для забезпечення якості кінцевого продукту.

Важливо, щоб тести забезпечували аналіз якості програмного забезпечення протягом усього процесу розробки. Можна припустити, що

подальші дослідження дозволять створити єдину нейромережу, яка успішно впорається зі складним завданням класифікації даних. Тому раціональніше приділити більше уваги тестуванню кінцевої взаємодії з користувачем. Цей вид тестування ефективно відображатиме зміни якості програмного забезпечення й залишатиметься актуальним під час подальших вдосконалень внутрішньої архітектури.

На другому етапі, коли необхідно перевірити точність складного алгоритму, стає актуальним тестування окремих одиниць програми. Першим кроком є перевірка точності моделі, що використовується. Крім того, важливо оцінити точність натренованої моделі на тестовому наборі даних. Оскільки класифікація елементів за допомогою цієї моделі є лише проміжним етапом роботи програмного забезпечення, існує подальший постпроцесинг. Оцінка точності на кожному з цих етапів дозволить отримати уявлення про якість програми.

Таким чином, здійснення цих вимог тестування допоможе забезпечити високу якість програмного забезпечення, забезпечуючи аналіз його якості на різних етапах розробки та перевірку точності складних алгоритмів.

Дані процеси тестування можна подати у формі таблиць (Таблиця 3.1 – 3.7) та рисунків (Рисунок 3.1 – 3.14):

Таблиця 3.1 – Тест 1

Мета тесту	Перевірка точності натренованої моделі протягом епох тренування для подальшого обрання моделі.
Початковий стан	Оброблений датасет із нормалізованими даними про російських військовослужбовців
Вхідні дані	Модель Decision Tree
Опис проведеного тесту	Обчислення точності на перевірочному датасеті для моделі на кожній епосі тренування
Очікуваний результат	Значення точності більше 0.7
Фактичний результат	Значення точності 0.73

	precision	recall	f1-score	support
Солдаты, матросы, сержанты, старшины	0.33	0.21	0.26	1895
Прапорщики и мичманы	0.83	0.93	0.87	11889
младшие офицеры	0.00	0.00	0.00	12
старшие офицеры	0.27	0.03	0.06	1411
высшие офицеры	0.37	0.49	0.42	1654
accuracy			0.73	16861
macro avg	0.36	0.33	0.32	16861
weighted avg	0.68	0.73	0.69	16861

Trained Decision Tree in 89.18 s

Рисунок 3.1 – Результат виконання тесту 1 на моделі Decision Tree



Рисунок 3.2 – Результат виконання тесту 1 на моделі Decision Tree

Таблиця 3.2 – Тест 2

Мета тесту	Перевірка точності натренованої моделі протягом кожної із епох тренування для подальшого обрання моделі.
Початковий стан	Оброблений датасет із нормалізованими даними про російських військовослужбовців
Вхідні дані	Модель Nearest Neighbors
Опис проведеного тесту	Обчислення точності на перевірочному датасеті для моделі на кожній епосі тренування
Очікуваний результат	Значення точності більше 0.7
Фактичний результат	Значення точності 0.73

	precision	recall	f1-score	support
Солдаты, матросы, сержанты, старшины	0.33	0.22	0.27	1895
Прапорщики и мичманы	0.82	0.93	0.87	11889
младшие офицеры	0.00	0.00	0.00	12
старшие офицеры	0.32	0.06	0.11	1411
высшие офицеры	0.38	0.41	0.40	1654
accuracy			0.73	16861
macro avg	0.37	0.33	0.33	16861
weighted avg	0.68	0.73	0.69	16861

Trained Nearest Neighbors in 106.42 s

Рисунок 3.3 – Результат виконання тесту 2 на моделі Nearest Neighbors

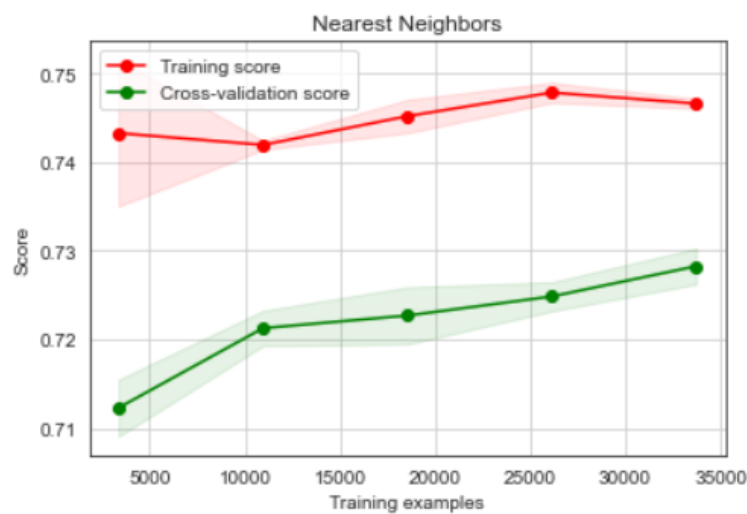


Рисунок 3.4 – Результат виконання тесту 2 на моделі Nearest Neighbors

Таблиця 3.3 – Тест 3

Мета тесту	Перевірка точності натренованої моделі протягом кожної із епох тренування для подальшого обрання моделі.
Початковий стан	Оброблений датасет із нормалізованими даними про російських військовослужбовців
Вхідні дані	Модель Naïve Bayes
Опис проведеного тесту	Обчислення точності на перевірочному датасеті для моделі на кожній епосі тренування
Очікуваний результат	Значення точності більше 0.7
Фактичний результат	Значення точності 0.7

	precision	recall	f1-score	support
Солдаты, матросы, сержанты, старшины	0.00	0.00	0.00	1895
Прапорщики и мичманы	0.71	1.00	0.83	11889
младшие офицеры	0.00	0.00	0.00	12
старшие офицеры	0.00	0.00	0.00	1411
высшие офицеры	0.00	0.00	0.00	1654
accuracy			0.71	16861
macro avg	0.14	0.20	0.17	16861
weighted avg	0.50	0.71	0.58	16861

Trained Naive Bayes in 3.42 s

Рисунок 3.5 – Результат виконання тесту 3 на моделі Naive Bayes

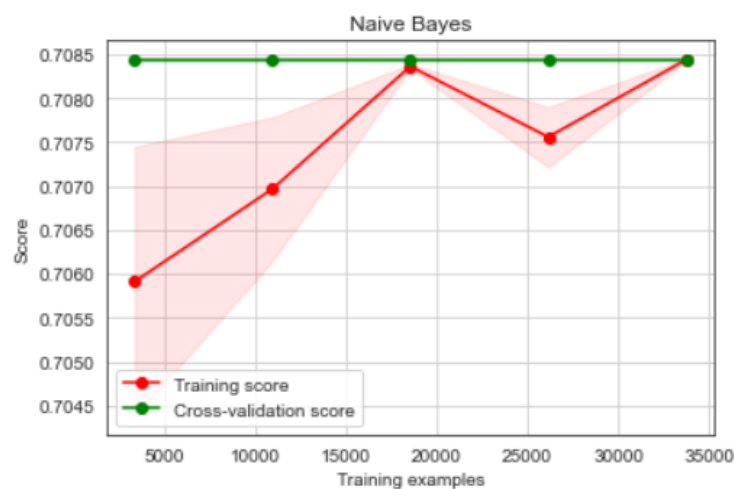


Рисунок 3.6 – Результат виконання тесту 3 на моделі Naive Bayes

Таблица 3.4 – Тест 4

Мета тесту	Перевірка точності натренованої моделі протягом кожної із епох тренування для подальшого обрання моделі.
Початковий стан	Оброблений датасет із нормалізованими даними про російських військовослужбовців
Вхідні дані	Модель Logistic Regression
Опис проведеного тесту	Обчислення точності на перевірконому датасеті для моделі на кожній епосі тренування
Очікуваний результат	Значення точності більше 0.7
Фактичний результат	Значення точності 0.72



	precision	recall	f1-score	support
Солдаты, матросы, сержанты, старшины	0.30	0.14	0.19	1895
Прапорщики и мичманы	0.79	0.95	0.86	11889
младшие офицеры	0.00	0.00	0.00	12
старшие офицеры	0.27	0.03	0.05	1411
высшие офицеры	0.38	0.38	0.38	1654
accuracy			0.72	16861
macro avg	0.35	0.30	0.30	16861
weighted avg	0.65	0.72	0.67	16861

Trained Logistic Regression in 21.35 s

Рисунок 3.7 – Результат виконання тесту 4 на моделі Logistic Regression

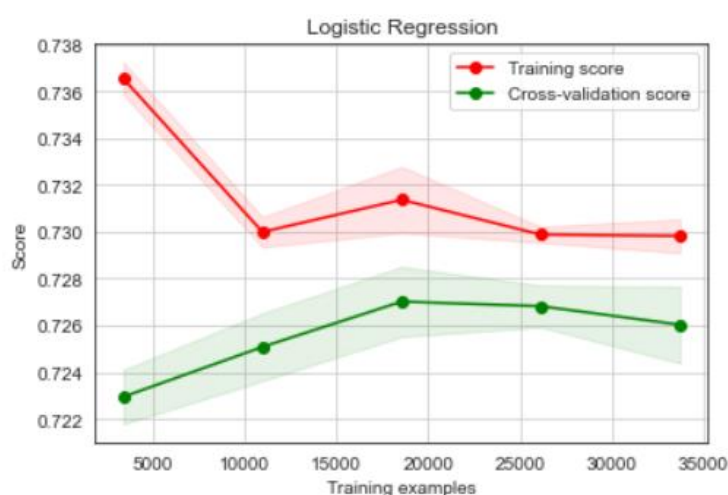


Рисунок 3.8 – Результат виконання тесту 4 на моделі Logistic Regression

Таблиця 3.5 – Тест 5

Мета тесту	Перевірка точності натренованої моделі протягом кожної із епох тренування для подальшого обрання моделі.
Початковий стан	Оброблений датасет із нормалізованими даними про російських військовослужбовців
Вхідні дані	Модель Gradient Boosting
Опис проведеного тесту	Обчислення точності на перевірочному датасеті для моделі на кожній епосі тренування
Очікуваний результат	Значення точності більше 0.7
Фактичний результат	Значення точності 0.74

	precision	recall	f1-score	support
Солдаты, матросы, сержанты, старшины	0.37	0.20	0.26	1895
Прапорщики и мичманы	0.83	0.94	0.88	11889
младшие офицеры	1.00	0.08	0.15	12
старшие офицеры	0.39	0.03	0.06	1411
высшие офицеры	0.39	0.52	0.44	1654
accuracy			0.74	16861
macro avg	0.59	0.35	0.36	16861
weighted avg	0.69	0.74	0.70	16861

Trained Gradient Boosting in 554.66 s

Рисунок 3.9 – Результат виконання тесту 5 на моделі Gradient Boosting

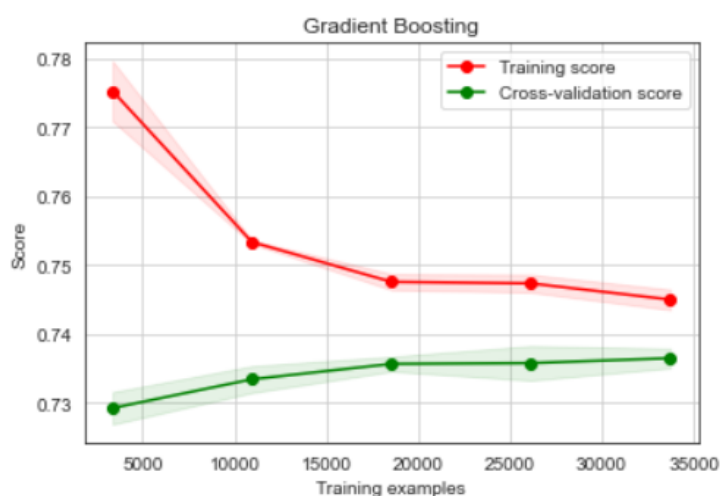


Рисунок 3.10 – Результат виконання тесту 5 на моделі Gradient Boosting

Таблица 3.6 – Тест 6

Мета тесту	Перевірка точності натренованої моделі протягом кожної із епох тренування для подальшого обрання моделі.
Початковий стан	Оброблений датасет із нормалізованими даними про російських військовослужбовців
Вхідні дані	Модель Random Forest
Опис проведеного тесту	Обчислення точності на перевірочному датасеті для моделі на кожній епосі тренування
Очікуваний результат	Значення точності більше 0.7
Фактичний результат	Значення точності 0.74

	precision	recall	f1-score	support
Солдаты, матросы, сержанты, старшины	0.36	0.19	0.25	1895
Прапорщики и мичманы	0.82	0.94	0.88	11889
младшие офицеры	0.00	0.00	0.00	12
старшие офицеры	0.40	0.02	0.03	1411
высшие офицеры	0.39	0.51	0.44	1654
accuracy			0.74	16861
macro avg	0.39	0.33	0.32	16861
weighted avg	0.69	0.74	0.69	16861

Trained Random Forest in 1527.46 s

Рисунок 3.11 – Результат виконання тесту 6 на моделі Random Forest

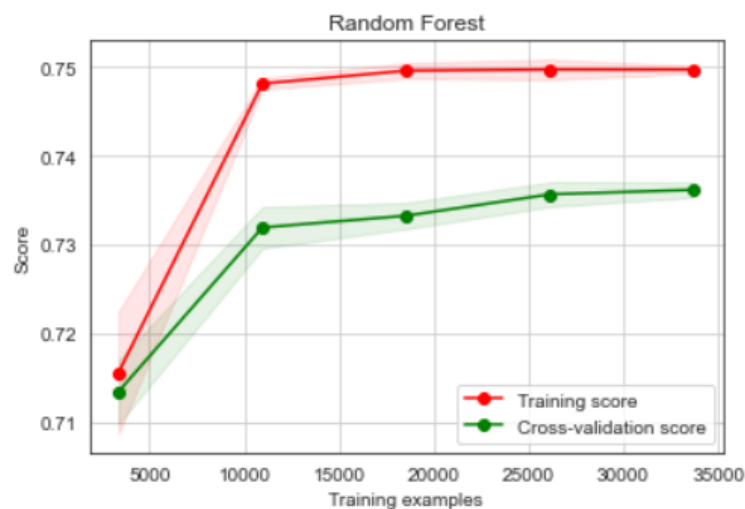


Рисунок 3.12 – Результат виконання тесту 6 на моделі Random Forest

Таблица 3.7 – Тест 7

Мета тесту	Перевірка точності натренованої моделі протягом кожної із епох тренування для подальшого обрання моделі.
Початковий стан	Оброблений датасет із нормалізованими даними про російських військовослужбовців
Вхідні дані	Модель Multilayer Perceptron
Опис проведеного тесту	Обчислення точності на перевірочному датасеті для моделі на кожній епісі тренування
Очікуваний результат	Значення точності більше 0.7
Фактичний результат	Значення точності 0.73

	precision	recall	f1-score	support
Солдаты, матросы, сержанты, старшины	0.35	0.18	0.24	1895
Прапорщики и мичманы	0.83	0.93	0.88	11889
младшие офицеры	0.00	0.00	0.00	12
старшие офицеры	0.39	0.04	0.07	1411
высшие офицеры	0.37	0.52	0.44	1654
accuracy			0.73	16861
macro avg	0.39	0.34	0.33	16861
weighted avg	0.69	0.73	0.69	16861

Trained Multilayer Perceptron in 347.92 s

Рисунок 3.13 – Результат виконання тесту 7 на моделі Multilayer Perceptron

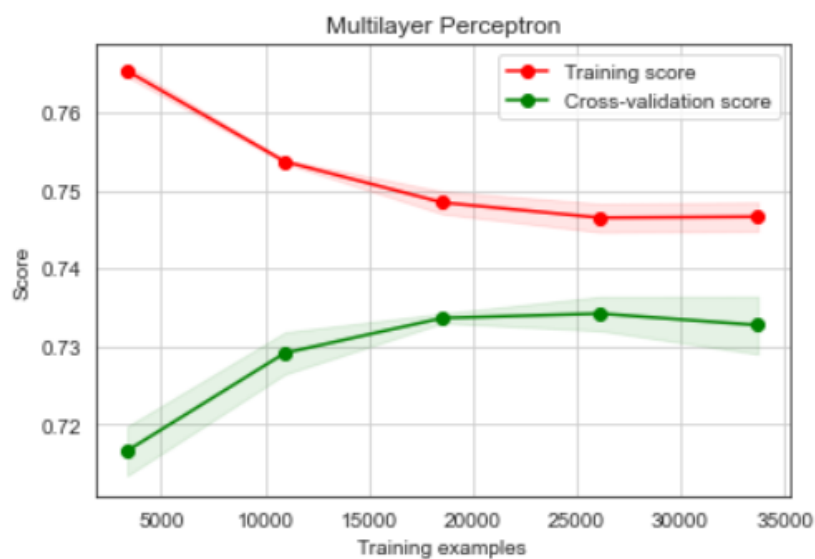


Рисунок 3.14 – Результат виконання тесту 7 на моделі Multilayer Perceptron

	classifier	train_score	test_score	training_time
4	Gradient Boosting	0.745463	0.736137	554.664317
5	Random Forest	0.749318	0.735366	1527.462086
6	Multilayer Perceptron	0.746135	0.733231	347.924942
0	Decision Tree	0.750247	0.728189	89.182042
1	Nearest Neighbors	0.746570	0.726766	106.419494
3	Logistic Regression	0.728698	0.723445	21.345301
2	Naive Bayes	0.708434	0.705118	3.423046

Рисунок 3.15 – Результат виконання тестів 1-7 – таблиця точностей алгоритмів класифікації.

Як можна зауважити з рисунку 3.15 точність натренованих моделей є відносно однаковою. Однак, спираючись на метрики precision, recall, f1-score, support, можна зробити наступні висновки.

- а) Всі алгоритми відповідають вимозі точності в 70%.
- б) Було недостатньо даних зі значенням «молодші офіцери», тому такі моделі як Decision Tree, Nearest Neighbors, Naive Bayes, Random Forest, Multilayer Perceptron – не натреновані класифікувати правильно дане значення. Висновок про це можна зробити проаналізувавши метрики precision та recall, які дорівнюють 0.
- в) Naive Bayes модель навчилася класифікувати лише значення «прапорщики та мічмани», а оскільки дані містять 70% саме цих значень, то модель має точність 70%, однак дана модель не може визначити інші значення, а отже використовувати її не можна.
- г) Всі інші значення були відносно однаково класифіковані усіма алгоритмами.

Зважаючи на вищеописані заключення для класифікації буде використано алгоритм Gradient Boosting.

Для регресії було використано 5 алгоритмів: Linear Regression, Gradient Boosting regressor, Lasso, Ridge та Elastic Net. Однак, результати тестування що зображені на рисунку 3.16 показали, що жоден з них, не набрав значення точності в 70%:

	method	r2_train	r2_test	MAE_train	MAE_test	MSE_train	MSE_test	training_time
0	Linear Regression	0.100510	0.104541	6.688248	6.683132	65.239260	0.104541	0.008601
1	Gradient Boosting regressor	0.293196	0.303414	5.972551	5.933864	50.750223	0.303414	196.522444
2	Lasso	0.089047	0.091207	6.877613	6.877304	66.210753	0.091207	2.573364
3	Ridge	0.100510	0.104535	6.688253	6.683175	65.239715	0.104535	2.193004
4	Elastic Net	0.080692	0.082211	6.945956	6.947412	66.866099	0.082211	2.212514

Рисунок 3.16 – Результат виконання тестів– таблиця точностей алгоритмів регресії

Найвище значення було отримано за допомогою алгоритму Gradient Boosting regressor – 30%, однак дане значення не відповідає раніше описаним вимогам до програмного забезпечення. По цій причині, методи регресії виключені з застосунку для користувача.

Подальше тестування проведено над веб-застосунком, яким користуватиметься кінцевий користувач:

Таблиця 3.8 – Тест 8

Мета тесту	Перевірка пошуку за ПІБ військового..
Початковий стан	Користувач на сторінці «Список»
Вхідні дані	Веб-застосунок підключений до бази даних
Опис проведеного тесту	Введення в пошукове поле ПІБ «Кузнецов Владислав Юрьевич» та натискання кнопки «Enter»
Очікуваний результат	Виведення всіх військовослужбовців з даним ПІБ.
Фактичний результат	Виведено всіх військовослужбовців з даним ПІБ.

Таблиця 3.9 – Тест 9

Мета тесту	Перевірка пошуку за ПІБ військового, що не має в базі даних.
Початковий стан	Користувач на сторінці «Список»
Вхідні дані	Веб-застосунок підключений до бази даних
Опис проведеного тесту	Введення в пошукове поле ПІБ «Кузнецова Наталия Сергеевна» та натискання кнопки «Enter»
Очікуваний результат	Виведення повідомлення про відсутність військовослужбовця з таким іменем.
Фактичний результат	Виведено повідомлення про відсутність військовослужбовця з таким іменем.

Таблиця 3.10 – Тест 10

Мета тесту	Перевірка відображення інфографіки.
Початковий стан	Користувач на сторінці «Графіки»
Вхідні дані	Веб-застосунок підключений до бази даних
Опис проведеного тесту	На сторінку виведені графіки. При наведенні на елементи графіків відображається відповідне значення кількості осіб.
Очікуваний результат	Відображено графіки. При наведенні на чоловічу статтю графіку «Стать» виводиться число 99543
Фактичний результат	Виведено повідомлення про відсутність військовослужбовця з таким іменем.

Таблиця 3.11 – Тест 11

Мета тесту	Перевірка відображення інфографіки.
Початковий стан	Користувач на сторінці «Графіки»
Вхідні дані	Веб-застосунок підключений до бази даних
Опис проведеного тесту	На сторінку виведені графіки. При наведенні на елементи графіків відображається відповідне значення кількості осіб.
Очікуваний результат	Відображено графіки. При наведенні на відповідний стовпчик графіку «Вік» виводиться число 22
Фактичний результат	Виведено число 22.

Таблиця 3.12 – Тест 12

Мета тесту	Перевірка завантаження файлу.
Початковий стан	Користувач на сторінці «Аналіз»
Вхідні дані	Веб-застосунок підключений до бази даних
Опис проведеного тесту	Завантажити файл формату .csv у поле «Завантажте файл формату .csv» та натиснути «Завантажити».

Продовження таблиці 3.12

Очікуваний результат	Відображено файл, що був завантажений.
Фактичний результат	Відображено файл, що був завантажений.

Таблиця 3.13 – Тест 13

Мета тесту	Перевірка завантаження файлу з недопустимим форматом даних.
Початковий стан	Користувач на сторінці «Аналіз»
Вхідні дані	Веб-застосунок підключений до бази даних
Опис проведеного тесту	Завантажити файл формату відмінного від .csv у поле «Завантажте файл формату .csv» та натиснути «Завантажити».
Очікуваний результат	Відображено повідомлення про те, що формат файлу невірний.
Фактичний результат	Відображено повідомлення про те, що формат файлу невірний.

Таблиця 3.14 – Тест 14

Мета тесту	Перевірка аналізу файлу та виведення даних в таблицю.
Початковий стан	Користувач на сторінці «Аналіз», файл з даними завантажено.
Вхідні дані	Веб-застосунок підключений до бази даних, файл формату .csv завантажено у відповідне поле.
Опис проведеного тесту	Відбувається натискання кнопки «Проаналізувати файл».
Очікуваний результат	Відображається таблиця з переліком розпарсених даних по відповідним до бази даних стовпчикам та відповідна інфографіка.
Фактичний результат	Відображено таблицю з переліком розпарсених даних та відповідна інфографіка.



### 3.3 Опис контрольного прикладу

Контрольні приклади використовуються для перевірки, чи працює програмне забезпечення належним чином і відповідає вимогам. Вони допомагають виявляти помилки, неправильну поведінку та несправності в програмі. Контрольні приклади можуть бути використані під час ручного тестування, автоматизованого тестування або тестування з використанням спеціалізованих фреймворків і інструментів.

На рисунках 3.17-3.20 наведено приклади роботи веб-застосунка та тести, що були описані в таблицях (Таблиця 3.8 – 3.14):

Кузнецов Владислав Юрьевич					
#	ПІБ	Звання	Місце проживання	Дата народження	Більше інформації
751	Кузнецов Владислав Юрьевич	Младший сержант		1996-9-20	<a href="#">Дізнатися більше</a>
59968	Кузнецов Владислав Юрьевич			1993-01-26	<a href="#">Дізнатися більше</a>

Рисунок 3.17 – Результат виконання пошуку в базі даних за ПІБ

Таблиця пошуку					
Кузнецова Наталия Сергеевна					
Немає записів					
#	ПІБ	Звання	Місце проживання	Дата народження	Більше інформації

Рисунок 3.18 – Результат виконання пошуку в базі даних за ПІБ

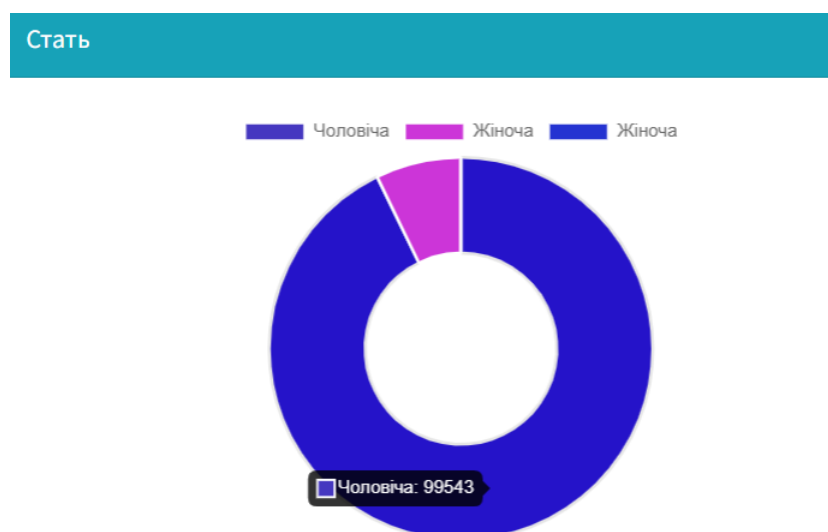


Рисунок 3.19 – Перегляд компоненту інфографіки

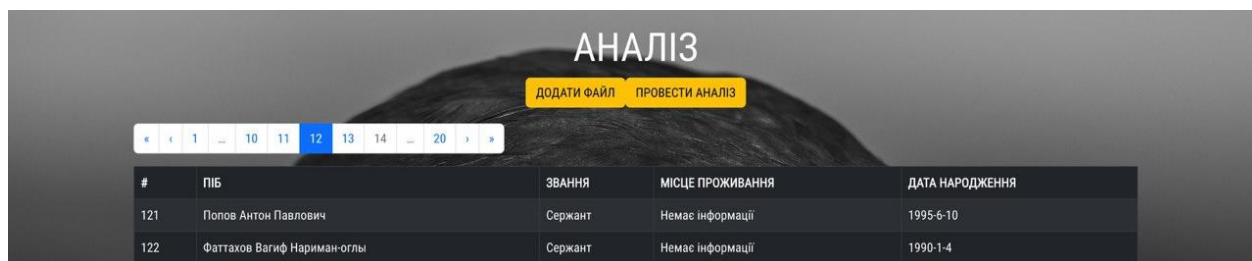


Рисунок 3.20 – Перегляд завантаженого файлу

## Висновки до розділу

В даному розділі було обрано оптимальну стратегію тестування програмного забезпечення, яка спрямована на перевірку точності використаних алгоритмів та подальший супровід системи. В результаті було встановлено, що наступні критерії є необхідними і достатніми для досягнення цілей.

- Тестування взаємодії з кінцевим користувачем, що використовує веб-клієнт додатку. Цей тест спрямований на перевірку правильності роботи системи під час взаємодії з реальними користувачами через веб-інтерфейс.
- Перевірка точності моделей на кожній епосі для визначення оптимального значення. Цей крок дозволяє аналізувати точність моделей на різних етапах їх навчання та визначити оптимальний момент для зупинки навчання моделі.
- Оцінка точності натренованих моделей на тестовому датасеті. Цей етап дозволяє оцінити якість моделей на незалежному наборі даних та зрозуміти, наскільки добре вони виконують своє завдання.

Ці стратегічні кроки забезпечують необхідну перевірку та оцінку точності використаних алгоритмів і дозволяють забезпечити ефективний супровід системи.

## 4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Розгортання програмного забезпечення

Розгортання React-застосунку на платформі Heroku[9] є простим та зручним. Heroku - це хмарна платформа, що дозволяє розгортати, керувати та масштабувати веб-додатки.

Процес розгортання React-застосунку на Heroku починається зі створення облікового запису на Heroku та створення нового додатку. Потім, використовуючи Git, можна підключитися до репозиторію Heroku та надіслати свій код на платформу. Heroku автоматично розпізнає React-застосунок і налаштовує його для запуску.

Крім того, Heroku надає додаткові можливості, такі як логування, моніторинг та масштабування, що дозволяють ефективно управляти та підтримувати веб-застосунок.

Розгортання React-застосунку на Heroku є зручним способом зробити додаток доступним для широкої аудиторії. Завдяки простоті процесу та надійності платформи Heroku, можна швидко розгорнути застосунок і сконцентруватися на його розвитку та функціоналі.

Розгортання програмного забезпечення для збору та аналізу даних про російських військовослужбовців можна побачити на рисунках (Рисунок 4.1-4.4):

```
PS C:\Users\vikyc\Desktop\Diplom-master\Diplom-master> heroku login
» Warning: Our terms of service have changed: https://dashboard.heroku.com/terms-of-service
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/ded3fca0-9e44-4d60-8de7-f5c11f0cbaad?requestor=SFmyNTY.g2gDbQAAAA05NS4xNTguNDM
figBYgABUYA.ebwnI0asdyP_xRP4H0MEarZrQyXpBu70bu2sMn0zkLo
logging in... done
logged in as vikyc13082002@gmail.com
```

Рисунок 4.1 – розгортання застосунку – вхід в обліковий аккаунт Heroku

```

vikyc@DESKTOP-M8SD616 MINGW64 ~/Desktop/Diplom-master/Diplom-master (master)
$ git commit -m "created"
[master (root-commit) baae24b] created
44 files changed, 37336 insertions(+)
create mode 100644 .gitignore
create mode 100644 README.md
create mode 100644 config/env.js
create mode 100644 config/getHttpsConfig.js
create mode 100644 config/jest/babelTransform.js
create mode 100644 config/jest/cssTransform.js
create mode 100644 config/jest/fileTransform.js
create mode 100644 config/modules.js
create mode 100644 config/paths.js
create mode 100644 config/webpack.config.js

```

Рисунок 4.2 – розгортання застосунку – git commit

```

vikyc@DESKTOP-M8SD616 MINGW64 ~/Desktop/Diplom-master/Diplom-master (master)
$ git push heroku master
Enumerating objects: 66, done.
Counting objects: 100% (66/66), done.
Delta compression using up to 8 threads
Compressing objects: 100% (60/60), done.
Writing objects: 100% (66/66), 6.54 MiB | 1.29 MiB/s, done.
Total 66 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Updated 44 paths from f978105
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-22 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:

```

Рисунок 4.3 – розгортання застосунку на Heroku

```

remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:      Procfile declares types      -> (none)
remote:      Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:      Done: 107M
remote: -----> Launching...
remote:      Released v3
remote:      https://test120.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/test120.git
 * [new branch]      master -> master

```

Рисунок 4.4 – розгортання застосунку на Heroku успішно завершено

## 4.2 Підтримка програмного забезпечення

У рамках дипломного проєкту була використана підтримка програмного забезпечення за допомогою Git та Heroku. Git був обраний як система керування версіями для відстеження змін у кодовій базі та спільної роботи над проєктом. За допомогою Git було можливо створювати гілки, злити різні гілки та керувати версіями проєкту.

Heroku використовувалася як хмарна платформа для розгортання та супроводу веб-додатків. Інтеграція з Git дозволила автоматично розпізнавати зміни у коді та налаштовувати середовище Heroku для запуску додатку.

Завдяки цій комбінації Git і Heroku, можна ефективно керувати розробкою, зберігати резервні копії коду, спільно працювати над новими функціями та легко розгорнути оновлення нашого програмного забезпечення.

## 4.3 Робота з програмним забезпеченням

У документі "Керівництво користувача" наведено детальну інструкцію щодо роботи з програмним забезпеченням.

### Висновки до розділу

У розділі було проведено аналіз та вивчення відповідних методів, інструментів та практик, що відображають етапи життєвого циклу програмного забезпечення після його розробки.

В результаті вивчення було встановлено, що розгортання ПЗ на платформі Heroku є простим та зручним процесом завдяки його хмарній архітектурі та інтеграції з Git. Цей процес дозволяє ефективно розгорнути розроблену систему та забезпечити доступ до неї широкому колу користувачів.

Крім того, підтримка програмного забезпечення за допомогою Git та Heroku забезпечує зручну систему контролю версій, яка дозволяє керувати розробкою та внесенням змін у програмне забезпечення. Git надає можливість

ефективно працювати з кодом, робити зміни, відкочувати до попередніх версій та спільно працювати з іншими розробниками. Heroku, у свою чергу, дозволяє зручно розгортати та керувати додатками, забезпечуючи їхню безперебійну роботу та масштабування.

Завдяки розгорнутій інструкції, яка описана в документі "Керівництво користувача", користувачі отримують докладну інформацію та кроки, необхідні для успішної роботи з програмним забезпеченням. Це забезпечує зручну та безпроблемну взаємодію користувачів з системою, дозволяючи їм ефективно використовувати її можливості та досягати поставлених цілей.

В цілому, етапи розгортання, супроводу та експлуатації програмного забезпечення є важливими компонентами для успішної втілення проекту. Застосування сучасних методів та інструментів, таких як Heroku та Git, дозволяє забезпечити ефективне управління програмним забезпеченням, забезпечити його надійність та доступність для користувачів. Правильно розгорнута та підтримувана система дозволяє забезпечити задоволення вимог та очікувань користувачів, забезпечуючи успішну реалізацію дипломного проєкту.

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		60

## ВИСНОВКИ

Основною метою нашого дипломного проєкту було поліпшення ефективності процесу пошуку та аналізу даних, пов'язаних з російськими військовослужбовцями. Шлях до досягнення поставленої мети включав ряд кроків та етапів.

Під час вивчення першого розділу дипломного проєкту було проведено аналіз наявних рішень та визначено перспективи розробки в області збору та аналізу даних про російських військовослужбовців, що беруть участь у війні проти України.

Аналіз рішень вказує на значимість та актуальність цієї тематики, оскільки збільшується необхідність в ефективному зборі, аналізі та інтерпретації даних, пов'язаних з воєнним конфліктом. Розробка та застосування інноваційних методів та підходів до обробки цих даних відіграють ключову роль у забезпеченні ефективної стратегії управління та прийняття рішень.

Виявлено, що недостатньо уваги приділялося автоматизованому збору даних та їх інтеграції з масштабованими системами аналізу. Однак, розробка та впровадження нових технологій, таких як методи штучного інтелекту та машинного навчання, можуть значно покращити процес збору та аналізу даних, забезпечуючи більш точну та швидку обробку інформації.

В другому розділі дипломного проєкту було ретельно проаналізовано процес досягнення поставленої мети, яким є збір та очищення даних, розробка бази даних з використанням Postgre та розробка архітектури нашого застосунку.

Проаналізувавши шлях до рішення поставленого завдання, було виявлено, що ефективний збір інформації є ключовим етапом у процесі розробки програмного забезпечення для аналізу даних про російських військовослужбовців. Було виконано докладне дослідження методів та технологій збору даних, а також виконано розробку відповідних модулів та

інструментів для автоматизації цього процесу. Такий підхід дозволить гарантувати високу якість і достовірність зібраних даних, що має велике значення для подальшого аналізу та прийняття обґрунтованих рішень. Розробка бази даних з використанням Postgre виявилась оптимальним варіантом для зберігання та управління великим обсягом даних, що були зібрані. Було ретельно спроектована структура бази даних, враховуючи специфічні вимоги і потреби досліджуваної області. Це забезпечить ефективний доступ до інформації, швидкість обробки запитів та збереження даних у надійному та масштабованому середовищі.

Розробка архітектури застосунку була проведена з урахуванням принципів модульності, масштабованості та ефективності. Було використано сучасні технології та практики, такі як React для фронтенду та Node.js для бекенду. Застосунок був розроблений з урахуванням вимог щодо безпеки, ефективності та зручного користування, що робить його готовим до використання у реальних умовах.

У третьому розділі дипломного проєкту була проведена детальна оцінка ефективності та доцільності різних методів тестування з метою вибору найбільш відповідних і необхідних для аналізу якості програмного забезпечення.

Після ретельного аналізу було вибрано методи тестування, які відповідають потребам і є необхідними та достатніми для аналізу якості програмного забезпечення.

Розділ четвертий ретельно розглядає процес розгортання та експлуатації розробленого програмного забезпечення а, також розділ охоплює питання підтримки та обслуговування програмного забезпечення.

Результати нашого дипломного проєкту свідчать про досягнення поставленої мети. Використовуючи наші розроблені методи та інструменти, ми здатні збирати та аналізувати дані про російських військовослужбовців з більшою ефективністю та точністю.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) RussianWar Criminals [Електронний ресурс] // – Режим доступу до ресурсу: <https://rwc.shtab.net/ru>.
- 2) InformNapalm [Електронний ресурс] // – Режим доступу до ресурсу: [https://informnapalm.org/db/russian-invasion/#lang=ua&page=m\\_unit](https://informnapalm.org/db/russian-invasion/#lang=ua&page=m_unit).
- 3) Pandas Documentation [Електронний ресурс] // – Режим доступу до ресурсу: <https://pandas.pydata.org/docs/index.html>.
- 4) NumPy Documentation [Електронний ресурс] // – Режим доступу до ресурсу: <https://numpy.org/doc/stable/>.
- 5) Matplotlib Documentation [Електронний ресурс] // – Режим доступу до ресурсу: <https://matplotlib.org/stable/index.html>.
- 6) Scikit-learn: Machine learning in Python – scikit-learn 1.1 [Електронний ресурс] / Документація. – Режим доступу до ресурсу: <https://scikitlearn.org/stable/>
- 7) Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). У Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M. A density-based algorithm for discovering clusters in large spatial databases with noise Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. с. 226–231.
- 8) Node.js v 17.9.1 Documentation. 2022. [Електронний ресурс]. – Режим доступу: <https://nodejs.org/docs/latest-v17.x/api/>
- 9) Heroku Documentation. 2023. [Електронний ресурс]. – Режим доступу: <https://devcenter.heroku.com/categories/reference>

## ДОДАТОК А ЗВІТ ПОДІБНОСТІ



Ім'я користувача:  
Лісовиченко Олег Іванович

Дата перевірки:  
04.06.2023 11:34:42 EEST

Дата звіту:  
04.06.2023 11:58:21 EEST

ID перевірки:  
1015410539

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
76913

Назва документа: ІП-92\_Ткаченко\_ПЗ

Кількість сторінок: 59 Кількість слів: 9309 Кількість символів: 74529 Розмір файлу: 2.07 MB ID файлу: 1015073846

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

**8.28%**  
**Схожість**

Найбільша схожість: 4.09% з джерелом з Бібліотеки (ID файлу: 1011468903)

5.4% Джерела з Інтернету 106 ..... Сторінка 61

8.07% Джерела з Бібліотеки 364 ..... Сторінка 63

**0% Цитат**

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

**0%**  
**Вилучень**

Немає вилучених джерел

**Модифікації**

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування 15 сторінок

					КПІ.ІП-9222.045430.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		64

Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“    ” \_\_\_\_\_ 2023 р.

Програмне забезпечення збору та аналізу даних про російських  
військовослужбовців, що беруть участь у війні проти України

**Текст програми**

КП.ІП-9222.045430.03.12

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_  Юрій ОЛІЙНИК

Нормоконтроль:

\_\_\_\_\_ Катерина ЛІЩУК

Виконавець:

\_\_\_\_\_ Вікторія ТКАЧЕНКО

Київ – 2023

## Файл clean\_data.py

```
import pandas as pd
import numpy as np

orcs = pd.read_csv('orcs.csv', index_col=0,
                  usecols=['orc_id', 'age', 'bplace', 'nationality', 'mstatus_cl', 'address',
                          'reg_address', 'mil_rank_cl', 'edu_lvl_cl', 'pass_auth', 'family'],
                  encoding='utf-8')

orcs['region'] = np.nan
orcs.sort_index(inplace=True)
orcs.head()
orcs.mstatus_cl.value_counts()

list = json.dumps(list, indent=8, ensure_ascii=False)
# Writing to sample.json
with open("data.json", "w") as outfile:
    outfile.write(list)

import json
df = pd.read_csv('orcs.csv', low_memory=False)
df.head()
diction = df.to_dict('records')
list = json.dumps(diction, ensure_ascii=False)
list.head()

import json
df = pd.read_json('data.json')
print(df)
orcs.dropna(subset=['mil_rank_cl', 'age'], inplace=True)
orcs.age = orcs.age.astype(int)
orcs.head()
orcs.isna().sum()

areas = \
['Республика Адыгея',
'Республика Башкортостан',
'Республика Бурятия',
'Республика Алтай',
'Республика Дагестан',
'Республика Ингушетия',
'Кабардино-Балкарская Республика',
'Республика Калмыкия',
'Карачаево-Черкесская Республика',
'Республика Карелия',
'Республика Коми',
```

					КПІ.ІП-9222.045430.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		2

'Республика Марий Эл',  
 'Республика Мордовия',  
 'Республика Саха (Якутия)',  
 'Республика Северная Осетия',  
 'Республика Татарстан',  
 'Республика Тыва',  
 'Удмуртская Республика',  
 'Республика Хакасия',  
 'Чеченская Республика',  
 'Чувашская Республика - Чувашия',  
 'Алтайский край',  
 'Краснодарский край',  
 'Красноярский край',  
 'Приморский край',  
 'Ставропольский край',  
 'Хабаровский край',  
 'Амурская область',  
 'Архангельская область',  
 'Астраханская область',  
 'Белгородская область',  
 'Брянская область',  
 'Владимирская область',  
 'Волгоградская область',  
 'Вологодская область',  
 'Воронежская область',  
 'Ивановская область',  
 'Иркутская область',  
 'Калининградская область',  
 'Калужская область',  
 'Камчатский край',  
 'Кемеровская область',  
 'Кировская область',  
 'Костромская область',  
 'Курганская область',  
 'Курская область',  
 'Ленинградская область',  
 'Липецкая область',  
 'Магаданская область',  
 'Московская область',  
 'Мурманская область',  
 'Нижегородская область',  
 'Новгородская область',

					КПІ.ІП-9222.045430.03.12	Арк.
Вмін.	Арк.	№ докум.	Підп.	Дата.		3

'Новосибирская область',  
 'Омская область',  
 'Оренбургская область',  
 'Орловская область',  
 'Пензенская область',  
 'Пермский край',  
 'Псковская область',  
 'Ростовская область',  
 'Рязанская область',  
 'Самарская область',  
 'Саратовская область',  
 'Сахалинская область',  
 'Свердловская область',  
 'Смоленская область',  
 'Тамбовская область',  
 'Тверская область',  
 'Томская область',  
 'Тульская область',  
 'Тюменская область',  
 'Ульяновская область',  
 'Челябинская область',  
 'Забайкальский край',  
 'Ярославская область',  
 'Москва',  
 'Санкт-Петербург',  
 'Еврейская автономная область',  
 'Ненецкий автономный округ',  
 'Ханты-Мансийский автономный округ',  
 'Чукотский автономный округ',  
 'Ямало-Ненецкий автономный округ',  
 'Украина',  
 'Беларусь'  
 ]  
 mil\_ranks = \  
 {  
     'Солдаты, матросы, сержанты, старшины' : ['рядовой', 'ефрейтор', 'младший сержант', 'сержант',  
   'старший сержант', 'старшина', 'матрос',  
   'старший матрос', 'старшина 2 статьи',  
   'старшина 1 статьи', 'главный старшина',  
   'главный корабельный старшина', 'старшина сержант'],  
     'Прапорщики и мичманы' : ['прапорщик', 'старший прапорщик', 'мичман', 'старший мичман'],  
     'младшие офицеры' : ['младший лейтенант', 'лейтенант', 'старший лейтенант',

					КПІ.ІП-9222.045430.03.12	Арк.
Вмін.	Арк.	№ докум.	Підп.	Дата.		4

```

        'капитан', 'капитан-лейтенант'],
    'старшие офицеры' : ['майор', 'подполковник', 'полковник', 'капитан 3 ранга',
        'капитан 2 ранга', 'капитан 1 ранга', 'старший офицер'],
    'высшие офицеры' : ['генерал-майор', 'генерал-лейтенант', 'генерал-полковник', 'генерал армии',
        'контр-адмирал', 'вице-адмирал', 'адмирал', 'адмирал флота']
}

from fuzzywuzzy import fuzz
import random

for i in orcs.index:
    try:
        # clear region
        if not pd.isna(orcs['bplace'][i]):
            orcs.loc[i, 'region'] = orcs.loc[i, 'bplace']
        else:
            if not pd.isna(orcs['reg_address'][i]):
                orcs.loc[i, 'region'] = orcs.loc[i, 'reg_address']
            else:
                if not pd.isna(orcs['address'][i]):
                    orcs.loc[i, 'region'] = orcs.loc[i, 'address']
                else:
                    if not pd.isna(orcs['pass_auth'][i]):
                        orcs.loc[i, 'region'] = orcs.loc[i, 'pass_auth']

        # clear family
        if not pd.isna(orcs['family'][i]):
            orcs.loc[i, 'family'] = True
        else:
            if orcs['mstatus_cl'][i] in ['Женат', 'Замужем']:
                orcs.loc[i, 'family'] = True
            else:
                orcs.loc[i, 'family'] = False if random.random() > 0.74 else True

        # clear education
        if pd.isna(orcs['edu_lvl_cl'][i]):
            orcs.loc[i, 'edu_lvl_cl'] = 'среднее'
        else:
            if orcs['edu_lvl_cl'][i] in ['ВВУЗ', 'ВУЦ']:
                orcs.loc[i, 'edu_lvl_cl'] = 'высшее военное'
            if orcs['edu_lvl_cl'][i] == 'Высшее':
                orcs.loc[i, 'edu_lvl_cl'] = 'высшее невоенное'
            else:

```

```

        orcs.loc[i, 'edu_lvl_cl'] = 'среднее'

# clear military rank
for j in mil_ranks:
    if orcs.loc[i, 'mil_rank_cl'].lower() in mil_ranks[j]:
        orcs.loc[i, 'mil_rank_cl'] = j
        break

# clear region
ratios = {j: fuzz.token_set_ratio(orcs['region'][i], areas[j]) for j in range(len(areas))}
orcs.loc[i, 'region'] = areas[max(ratios, key=ratios.get)]
except:
    pass

#clear nationality
orcs.nationality = orcs.nationality.apply(lambda x: 'русский' if pd.isna(x) else x.lower())

orcs.drop(['bplace', 'reg_address', 'address', 'pass_auth', 'mstatus_cl'], axis=1, inplace=True)
orcs.to_csv('orcs_cleaned.csv', encoding='utf-8')
orcs.isna().sum()

import pandas as pd
import numpy as np

orcs = pd.read_csv('orcs_cleaned.csv', index_col=0,
                  encoding='utf-8')
orcs.describe()

orcs['age_range'] = np.nan;
for i in orcs.index:
    try:
        #age_range
        if not pd.isna(orcs['age'][i]):
            if(orcs.loc[i, 'age']<21):
                orcs.loc[i, 'age_range'] = '18-20'
                continue
            if(orcs.loc[i, 'age']<26):
                orcs.loc[i, 'age_range'] = '21-25'
                continue
            if(orcs.loc[i, 'age']<31):

```

					КПІ.ІП-9222.045430.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6



```

        orcs.loc[i, 'age_range'] = '26-30'
        continue
    if(orcs.loc[i, 'age']<36):
        orcs.loc[i, 'age_range'] = '31-35'
        continue
    if(orcs.loc[i, 'age']<41):
        orcs.loc[i, 'age_range'] = '36-40'
        continue
    if(orcs.loc[i, 'age']<46):
        orcs.loc[i, 'age_range'] = '41-45'
        continue
    if(orcs.loc[i, 'age']<51):
        orcs.loc[i, 'age_range'] = '46-50'
        continue
    if(orcs.loc[i, 'age']<56):
        orcs.loc[i, 'age_range'] = '51-55'
        continue
    if(orcs.loc[i, 'age']<61):
        orcs.loc[i, 'age_range'] = '56-60'
        continue
    if(orcs.loc[i, 'age']<66):
        orcs.loc[i, 'age_range'] = '61-65'
        continue
    if(orcs.loc[i, 'age']<71):
        orcs.loc[i, 'age_range'] = '66-70'
        continue

except:
    pass

orcs.to_csv('orcs_cleaned.csv', encoding='utf-8')

```

## Файл classifier.py

```

import pandas as pd
import numpy as np

orcs = pd.read_csv('orcs_cleaned.csv', index_col=0, encoding='utf-8')
orcs.family = orcs.family.astype('int')

orcs

orcs.describe()

orcs.age.hist()

```

```

import matplotlib.pyplot as plt
import seaborn as sns

sns.set_style('white')
ax = sns.countplot(x='family', data=orcs).set_title('Family')
ax.axes.set_xlabel("")
ax.axes.set_ylabel("")
plt.show()

from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler

targets = orcs.mil_rank_cl
targets = LabelEncoder().fit_transform(targets)

features = orcs.drop(['mil_rank_cl', 'age_range'], axis=1)
features['edu_lvl_cl'] = LabelEncoder().fit_transform(features['edu_lvl_cl'])
features = pd.get_dummies(features, columns=['family', 'region', 'nationality'])

features = MinMaxScaler().fit_transform(features)
features.shape

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(features, targets, random_state=50, test_size=0.1)

from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.neural_network import MLPClassifier

dict_classifiers = {

    "Nearest Neighbors":
        {
            'classifier': KNeighborsClassifier(),
            'name': 'KNN',
            'params': [

```

					КПІ.ІП-9222.045430.03.12	Арк.
Вмін.	Арк.	№ докум.	Підп.	Дата.		8

```

        {
            'n_neighbors': [5, 10, 20],
            'weights': ['uniform', 'distance']
        }
    ]
},

"Decision Tree":
{
    'classifier': DecisionTreeClassifier(),
    'name': 'DT',
    'params': [
        {
            'max_depth': [10, 50, 100, None],
            'criterion': ['gini', 'entropy'],
            'splitter': ['best', 'random'],
            'min_samples_split': [2, 5, 10],
            'min_samples_leaf': [10, 50, 100]
        }
    ]
},

"Random Forest":
{
    'classifier': RandomForestClassifier(),
    'name': 'RF',
    'params': [
        {
            'n_estimators': [50, 100, 200],
            'max_depth': [10, 50, 100, None],
            'criterion': ['gini', 'entropy'],
            'min_samples_split': [2, 5, 10],
            'min_samples_leaf': [10, 50, 100]
        }
    ]
},

"Naive Bayes":
{
    'classifier': GaussianNB(),
    'name': 'NB',
    'params': [

```

```

        {
            'var_smoothing': np.logspace(0, -9, 10)
        }
    ],
},

```

"Logistic Regression":

```

{
    'classifier': LogisticRegression(),
    'name': 'LR',
    'params': [
        {
            "C": np.logspace(-3, 3, 5),
            "penalty": ["l1", "l2", "elasticnet"]
            # l1 lasso l2 ridge
        }
    ]
},

```

"Multilayer Perceptron":

```

{
    'classifier': MLPClassifier(),
    'name': 'MLP',
    'params': [
        {
            'hidden_layer_sizes': [(50, 50, 50),
                                   (50, 100, 50),
                                   (100,)]
        }
    ]
},

```

"Gradient Boosting":

```

{
    'classifier': GradientBoostingClassifier(),
    'name': 'GB',
    'params': [
        {
            'loss': ['deviance', 'exponential'],
            'n_estimators': [100, 200],
        }
    ]
}

```

					КПІ.ІП-9222.045430.03.12	Арк.
						10
Змін.	Арк.	№ докум.	Підп.	Дата.		

```

    }
}

from sklearn.model_selection import learning_curve

def plot_learning_curve(estimator, title, X, y, ylim=None, cv=None, n_jobs=1):
    """
    Generate a simple plot of the test and training learning curve.

    Parameters
    -----
    estimator : object type that implements the "fit" and "predict" methods
        An object of that type which is cloned for each validation.

    title : string
        Title for the chart.

    X : array-like, shape (n_samples, n_features)
        Training vector, where n_samples is the number of samples and
        n_features is the number of features.

    y : array-like, shape (n_samples) or (n_samples, n_features), optional
        Target relative to X for classification or regression;
        None for unsupervised learning.

    ylim : tuple, shape (ymin, ymax), optional
        Defines minimum and maximum yvalues plotted.

    cv : integer, cross-validation generator, optional
        If an integer is passed, it is the number of folds (defaults to 3).
        Specific cross-validation objects can be passed, see
        sklearn.cross_validation module for the list of possible objects

    n_jobs : integer, optional
        Number of jobs to run in parallel (default 1).
    """
    plt.figure()
    plt.title(title)

    if ylim is not None:
        plt.ylim(*ylim)

```

```

plt.xlabel("Training examples")
plt.ylabel("Score")

train_sizes, train_scores, test_scores = learning_curve(estimator, X, y, cv=cv, n_jobs=n_jobs)

train_scores_mean = np.mean(train_scores, axis=1)
train_scores_std = np.std(train_scores, axis=1)
test_scores_mean = np.mean(test_scores, axis=1)
test_scores_std = np.std(test_scores, axis=1)

plt.grid()

plt.fill_between(train_sizes, train_scores_mean - train_scores_std,
                 train_scores_mean + train_scores_std, alpha=0.1, color="r")
plt.fill_between(train_sizes, test_scores_mean - test_scores_std,
                 test_scores_mean + test_scores_std, alpha=0.1, color="g")

plt.plot(train_sizes, train_scores_mean, 'o-', color="r", label="Training score")
plt.plot(train_sizes, test_scores_mean, 'o-', color="g", label="Cross-validation score")

plt.legend(loc="best")
return plt

import time
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import pickle

num_classifiers = len(dict_classifiers.keys())

def batch_classify(X_train, Y_train, X_test, Y_test, classifier, verbose=True):
    """
    Classify the data using the given classifier

    Args:
        X_train (_type_): Training data
        Y_train (_type_): Training labels
        X_test (_type_): Test data
        Y_test (_type_): Test labels

```

classifier (\_type\_): Classifier to use  
 verbose (bool, optional): Print progress to stdout. Defaults to True.

Returns: (\_type\_): Classifier score

"""

```
t_start = time.perf_counter()
grid = GridSearchCV(estimator=dict_classifiers[classifier]['classifier'],
                    param_grid=dict_classifiers[classifier]['params'],
                    cv=3, n_jobs=3, verbose=verbose, refit=True, scoring='accuracy')

estimator = grid.fit(X_train, Y_train)
t_diff = time.perf_counter() - t_start

filename = f'{dict_classifiers[classifier]["name"]}.sav'
pickle.dump(estimator, open(filename, 'wb'))

train_score = estimator.score(X_train, Y_train)
test_score = estimator.score(X_test, Y_test)

predicted = estimator.predict(x_test)
expected = y_test

confusion = confusion_matrix(expected, predicted)
print(confusion, end="\n\n")

names = [str(i) for i in orcs.mil_rank_cl.unique()]
print(classification_report(expected, predicted, target_names=names), end="\n\n")
# Accuracy - number of correct predictions over all predictions
# Precision - how many of the positive predictions made are correct (true positives)
# Recall - how many of the positive cases the classifier correctly predicted,
#         over all the positive cases in the data
# F1 - harmonic mean of precision and recall

results = pd.Series()

results.loc[0] = classifier
results.loc[1] = train_score
results.loc[2] = test_score
results.loc[3] = t_diff

if verbose:
    print("Trained {c} in {f:.2f} s".format(c=classifier, f=t_diff))
```

```

plot_learning_curve(estimator, "{}".format(classifier), X_train, Y_train, cv=3, n_jobs=3)

return results

import warnings

warnings.filterwarnings("ignore")

df_results = pd.DataFrame(np.zeros(shape=(0, 4)),
                           columns = ['classifier', 'train_score', 'test_score', 'training_time'])

dt = batch_classify(x_train, y_train, x_test, y_test, 'Decision Tree')

df_results = df_results.append({'classifier': dt[0], 'train_score': dt[1],
                                'test_score': dt[2], 'training_time': dt[3]},
                                ignore_index=True)

warnings.filterwarnings("ignore")

knn = batch_classify(x_train, y_train, x_test, y_test, 'Nearest Neighbors')

df_results = df_results.append({'classifier': knn[0], 'train_score': knn[1],
                                'test_score': knn[2], 'training_time': knn[3]},
                                ignore_index=True)

warnings.filterwarnings("ignore")

nb = batch_classify(x_train, y_train, x_test, y_test, 'Naive Bayes')

df_results = df_results.append({'classifier': nb[0], 'train_score': nb[1],
                                'test_score': nb[2], 'training_time': nb[3]},
                                ignore_index=True)

warnings.filterwarnings("ignore")

lr = batch_classify(x_train, y_train, x_test, y_test, 'Logistic Regression')

df_results = df_results.append({'classifier': lr[0], 'train_score': lr[1],
                                'test_score': lr[2], 'training_time': lr[3]},
                                ignore_index=True)

```



```
warnings.filterwarnings("ignore")

gbc = batch_classify(x_train, y_train, x_test, y_test, 'Gradient Boosting')

df_results = df_results.append({'classifier': gbc[0], 'train_score': gbc[1],
                                'test_score': gbc[2], 'training_time': gbc[3]},
                                ignore_index=True)

warnings.filterwarnings("ignore")

rf = batch_classify(x_train, y_train, x_test, y_test, 'Random Forest')

df_results = df_results.append({'classifier': rf[0], 'train_score': rf[1],
                                'test_score': rf[2], 'training_time': rf[3]},
                                ignore_index=True)

warnings.filterwarnings("ignore")

mlp = batch_classify(x_train, y_train, x_test, y_test, 'Multilayer Perceptron')

df_results = df_results.append({'classifier': mlp[0], 'train_score': mlp[1],
                                'test_score': mlp[2], 'training_time': mlp[3]},
                                ignore_index=True)

df_results = df_results.sort_values(by=['test_score'], ascending=False)
df_results
```

### Файл regression.py

```
import pandas as pd
import numpy as np

orcs = pd.read_csv('orcs_cleaned.csv', index_col=0, encoding='utf-8')
orcs

features = orcs.drop(['age'], axis=1)
features = features.drop(['age_range'], axis=1)
features

from sklearn.preprocessing import LabelEncoder

orcs_encoded= orcs
lables = orcs_encoded.columns
size = len(lables)
```

```

for i in range(size):
    orcs_encoded[labels[i]] = LabelEncoder().fit_transform(orcs_encoded[labels[i]])
orcs_encoded

import matplotlib.pyplot as plt

plt.figure(figsize=(10,6))
plt.hist(orcs.age, bins=50, ec='black', color='b')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()

pd.plotting.scatter_matrix(orcs_encoded[['age', 'age_range', 'nationality', 'family', 'edu_lvl_cl', 'mil_rank_cl',
'region']], figsize=(18,18))
print("")

orcs_encoded

mask = np.zeros_like(orcs_encoded.corr())
triangle_indeces = np.triu_indices_from(mask)
mask[triangle_indeces] = True
mask

import seaborn as sns

Mask = np.zeros_like(orcs_encoded.corr())
triangle_indeces = np.triu_indices_from(Mask)
Mask[triangle_indeces] = True

plt.figure(figsize=(16,10))
sns.heatmap(orcs_encoded.corr(), mask=Mask, annot = True, annot_kws={"size" : 14})
sns.set_style('white')
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
plt.show()

targets = orcs_encoded.age
features = orcs_encoded
features = features.drop(['age_range'], axis=1)
features = features.drop(['age'], axis=1)

from sklearn.model_selection import train_test_split

```

```
x_train, x_test, y_train, y_test = train_test_split(features, targets, random_state=42)
```

```
import statsmodels.api as sm
```

```
X_incl_const = sm.add_constant(x_train)
```

```
model = sm.OLS(y_train, X_incl_const)
```

```
result= model.fit()
```

```
print(result.summary())
```

```
r=result.predict(X_incl_const)
```

```
r=np.array(result)
```

```
sns.regplot(result.fittedvalues.copy(), y_train)
```

```
plt.show()
```

```
x_incl_const = sm.add_constant(x_train)
```

```
model = sm.OLS(y_train, x_incl_const)
```

```
result = model.fit()
```

```
pd.DataFrame({'coef': result.params, 'p-values': round(result.pvalues,3)})
```

```
"""Бачимо, що p-values у колонок nationality, family, та edu_lvl_cl - завелике. Перевіримо на  
мультиколеніарність"""
```

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

```
x_incl_const = sm.add_constant(x_train)
```

```
model = sm.OLS(y_train, X_incl_const)
```

```
result = model.fit()
```

```
vif = [variance_inflation_factor(exog=x_incl_const.values, exog_idx = i) for i in range(x_incl_const.shape[1])]
```

```
pd.DataFrame({'coef_name': x_incl_const.columns, 'vif': np.around(vif, 2)})
```

```
"""Всі дані мають vif < 5, тому спробуємо побудувати модель використовуючи BIC"""
```

```
x_incl_const = sm.add_constant(x_train)
```

```
model = sm.OLS(y_train, x_incl_const)
```

```
result = model.fit()
```

```
print('BIC: \t', result.bic)
```

```
print('R-squared:', result.rsquared)
```

					КПІ.ІП-9222.045430.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		17

```
"""Спробуємо по черзі прибрати поля що мають великий P-value"""
```

```
x_incl_const = sm.add_constant(x_train)
x_incl_const = x_incl_const.drop(['nationality'], axis=1)
```

```
model = sm.OLS(y_train, x_incl_const)
result = model.fit()
```

```
print('BIC: \t', result.bic)
```

```
print('R-squared:', result.rsquared)
```

```
x_incl_const = sm.add_constant(x_train)
x_incl_const = x_incl_const.drop(['nationality', 'family'], axis=1)
```

```
model = sm.OLS(y_train, x_incl_const)
result = model.fit()
```

```
print('BIC: \t', result.bic)
```

```
print('R-squared:', result.rsquared)
```

```
"""результат покращився, але незначно :( Залишимо як і було."""
```

```
regression_results = pd.DataFrame(columns = ['method', 'r2_train', 'r2_test', 'MAE_train', 'MAE_test', 'MSE_train',
'MSE_test', 'training_time'])
```

```
x_train, x_test, y_train, y_test = train_test_split(features, targets, random_state=42)
```

```
import warnings
```

```
warnings.filterwarnings("ignore")
```

```
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
```

```
def get_metrics(y_train, y_train_prediced, y_test, y_test_prediced):
    r2_train = r2_score(y_train, y_train_prediced)
    r2_test = r2_score(y_test, y_test_prediced)
    mae_train = mean_absolute_error(y_train, y_train_prediced)
```

```

mae_test = mean_absolute_error(y_test, y_test_prediced)
mse_train = mean_squared_error(y_train, y_train_prediced)
mse_test = mean_squared_error(y_test, y_test_prediced)
return r2_train, r2_test, mae_train, mae_test, mse_train, mse_test

from sklearn.linear_model import LinearRegression
from sklearn.metrics import confusion_matrix
import time

t_start = time.perf_counter()
model = LinearRegression().fit(x_train, y_train)
t_diff = time.perf_counter() - t_start

y_train_prediced = model.predict(x_train)
y_test_prediced = model.predict(x_test)

r2_train, r2_test, MAE_train, MAE_test, MSE_train, MSE_test = get_metrics(y_train, y_train_prediced, y_test,
y_test_prediced)
regression_results = regression_results.append({'method': 'Linear Regression', 'r2_train': r2_train, 'r2_test': r2_test,
'MAE_train': MAE_train, 'MAE_test': MAE_test, 'MSE_train': MSE_train, 'MSE_test': MSE_test, 'training_time': t_diff},
ignore_index=True)

from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import GridSearchCV

params_dict = {
    'params': [
        {
            "learning_rate": [0.05, 0.1, 0.2],
            "min_samples_split": np.linspace(0.1, 0.5, 4),
            "min_samples_leaf": np.linspace(0.1, 0.5, 4),
            "max_depth": [3, 5],
            "alpha": [0.5, 0.7, 1.0],
            "max_features": [0.1, 0.5, 0.9],
        }
    ]
}

grid = GridSearchCV(GradientBoostingRegressor(), params_dict['params'],
                    refit=True, cv=3, scoring='accuracy', n_jobs=3)

t_start = time.perf_counter()
model = grid.fit(x_train, y_train)

```

```

t_diff = time.perf_counter() - t_start

y_train_prediced = model.predict(x_train)
y_test_prediced = model.predict(x_test)

r2_train, r2_test, MAE_train, MAE_test, MSE_train, MSE_test = get_metrics(y_train, y_train_prediced, y_test,
y_test_prediced)
regression_results = regression_results.append({'method': 'Gradient Boosting egressor', 'r2_train': r2_train, 'r2_test':
r2_test, 'MAE_train': MAE_train, 'MAE_test': MAE_test, 'MSE_train': MSE_train, 'MSE_test': r2_test, 'training_time':
t_diff}, ignore_index=True)

from sklearn.linear_model import Lasso
params_dict = {
    'params': [
        {
            "alpha": np.logspace(0, 100, 100),
        }
    ]
}

grid = GridSearchCV(Lasso(), params_dict['params'],
                    refit=True, cv=3, scoring='accuracy', n_jobs=3)

t_start = time.perf_counter()
model = grid.fit(x_train, y_train)
t_diff = time.perf_counter() - t_start

y_train_prediced = model.predict(x_train)
y_test_prediced = model.predict(x_test)

r2_train, r2_test, MAE_train, MAE_test, MSE_train, MSE_test = get_metrics(y_train, y_train_prediced, y_test,
y_test_prediced)
regression_results = regression_results.append({'method': 'Lasso', 'r2_train': r2_train, 'r2_test': r2_test, 'MAE_train':
MAE_train, 'MAE_test': MAE_test, 'MSE_train': MSE_test, 'MSE_test': r2_test, 'training_time': t_diff},
ignore_index=True)

from sklearn.linear_model import Ridge
params_dict = {
    'params': [
        {
            "alpha": np.logspace(0, 100, 100),
        }
    ]
}

```

```

}

grid = GridSearchCV(Ridge(), params_dict['params'],
                    refit=True, cv=3, scoring='accuracy', n_jobs=3)

t_start = time.perf_counter()
model = grid.fit(x_train, y_train)
t_diff = time.perf_counter() - t_start

y_train_prediced = model.predict(x_train)
y_test_prediced = model.predict(x_test)

r2_train, r2_test, MAE_train, MAE_test, MSE_train, MSE_test = get_metrics(y_train, y_train_prediced, y_test,
y_test_prediced)
regression_results = regression_results.append({'method': 'Ridge', 'r2_train': r2_train, 'r2_test': r2_test, 'MAE_train':
MAE_train, 'MAE_test': MAE_test, 'MSE_train': MSE_train, 'MSE_test': r2_test, 'training_time': t_diff},
ignore_index=True)

from sklearn.linear_model import ElasticNet

params_dict = {
    'params': [
        {
            "alpha": np.logspace(0, 20, 10),
            "l1_ratio": np.linspace(0.1, 0.9, 9),
        }
    ]
}

grid = GridSearchCV(ElasticNet(), params_dict['params'],
                    refit=True, cv=3, scoring='accuracy', n_jobs=3)

t_start = time.perf_counter()
model = grid.fit(x_train, y_train)
t_diff = time.perf_counter() - t_start

y_train_prediced = model.predict(x_train)
y_test_prediced = model.predict(x_test)

r2_train, r2_test, MAE_train, MAE_test, MSE_train, MSE_test = get_metrics(y_train, y_train_prediced, y_test,
y_test_prediced)
regression_results = regression_results.append({'method': 'Elastic Net', 'r2_train': r2_train, 'r2_test': r2_test,
'MAE_train': MAE_train, 'MAE_test': MAE_test, 'MSE_train': MSE_train, 'MSE_test': r2_test, 'training_time': t_diff},
ignore_index=True)

```

regression\_results

### Файл app.js

```
import './App.css';
import OrcsList from './components/pages/OrcsList'
import {Routes, Route, Link} from 'react-router-dom'
import GlobalLayout from './components/globalLayout'
import 'bootstrap/dist/css/bootstrap.min.css';
import InfoGraphics from './components/pages/infoGraph';
//import Analysis from './components/pages/Analysis';
import AboutUs from './components/pages/aboutUs';

function App() {
  return (
    <Routes>
      <Route path="/" element={<GlobalLayout/>}>
        <Route index path="/" element={<AboutUs/>} ></Route>
        <Route path='InfoGraphics' element={<InfoGraphics/>}></Route>
        <Route path='orcsList' element={<OrcsList/>} ></Route>

      </Route>
    </Routes>
  );
}

export default App;
```

### Файл app.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

### Файл index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import { BrowserRouter } from 'react-router-dom';
import App from './App';
// import './asstets/styles/app.scss';
import './index.css';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
```

					КПІ.ІП-9222.045430.03.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		22



```

<React.StrictMode>
  <BrowserRouter>
    <App />
  </BrowserRouter>
</React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();

const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
  }
};

export default reportWebVitals;

```

#### Файл reportWebVitals.js

```

const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
  }
};

export default reportWebVitals;

```

#### Файл index.jsx

```

import { Button } from 'react-bootstrap';
import indexModule from './aboutUs/index.module.css'
import { motion } from 'framer-motion'

const textAnimnation = {
  hidden: {
    x: -100,
    opacity: 0,

```

					КПІ.ІП-9222.045430.03.12	Арк.
						23
Змін.	Арк.	№ докум.	Підп.	Дата.		

```

    },
    visible: {
      x: 0,
      opacity: 1,
    },
  },
}

const AboutUs = () => {
  return (
    <section className={indexModule.section}>
      <div className={`$${indexModule.main} container`} >
        <div className={indexModule.mainText}>
          <motion.h1 variants={textAnimnation}
            initial='hidden'
            whileInView='visible'

            className={indexModule.h1}>ЯСТРУБ</motion.h1>
          <motion.h2 variants={textAnimnation} className={indexModule.h2}>
            Проект, який спрямований покарати кожного причетного до українського геноциду
          </motion.h2>
          <Button variant="warning">ДИВИТИСЯ СПИСОК</Button>{' '}
        </div>
      </div>
    </section>
  )
};

export default AboutUs;

```

Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“    ” \_\_\_\_\_ 2023 р.


Програмне забезпечення збору та аналізу даних про російських  
військовослужбовців, що беруть участь у війні проти України

**Програма та методика тестування**

КП.ІП-9222.045430.04.51

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_  Юрій ОЛПІЙНИК

Нормоконтроль:

\_\_\_\_\_ Катерина ЛІЩУК

Виконавець:

\_\_\_\_\_ Вікторія ТКАЧЕНКО

Київ – 2023

## ЗМІСТ

1	ОБ’ЄКТ ВИПРОБУВАНЬ .....	3
2	МЕТА ТЕСТУВАННЯ.....	4
3	МЕТОДИ ТЕСТУВАННЯ .....	5
4	ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ.....	6

					КПІ.ІП-9222.045430.04.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		2

## 1 ОБ'ЄКТ ВИПРОБУВАНЬ

Об'єктом випробування є програмне забезпечення збору та аналізу даних про російських військовослужбовців, що беруть участь у війні проти України.

					КПІ.ІП-9222.045430.04.51	Арк.
						3
Змін	Арк.	№ докум.	Підп.	Дата.		

## 2 МЕТА ТЕСТУВАННЯ

Метою тестування є наступне:

- перевірка правильності роботи програмного забезпечення відповідно до функціональних вимог;
- перевірка швидкості роботи веб-додатку;
- перевірка сумісності веб-додатку з останніми версіями сучасних браузерів (Chrome, Opera, Firefox, ...);
- знаходження проблем, помилок і недоліків з метою їх усунення;
- перевірка зручності графічного інтерфейсу.

					КПІ.ІП-9222.045430.04.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		4

### 3 МЕТОДИ ТЕСТУВАННЯ

Для тестування програмного забезпечення використовувався метод сірої скриньки. У результаті було проведено тестування наступних рис програмного забезпечення:

- тестування кросплатформності програмного забезпечення;
- тестування UI;
- функціональне тестування;
- нефункціональне тестування.

Також було проведено тестування моделей ML.

- Перевірка точності моделей на кожній епосі для визначення оптимального значення. Цей крок дозволяє аналізувати точність моделей на різних етапах їх навчання та визначити оптимальний момент для зупинки навчання моделі.
- Оцінка точності натренованих моделей на тестовому датасеті. Цей етап дозволяє оцінити якість моделей на незалежному наборі даних та зрозуміти, наскільки добре вони виконують своє завдання.

					КПІ.ІП-9222.045430.04.51	Арк.
						5
Змін	Арк.	№ докум.	Підп.	Дата.		

## 4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування веб-додатку відбувається шляхом:

- тестування кросплатформності програмного забезпечення;
- мануального тестування із використанням «крайніх меж» та недопустимих значень в полях для вводу;
- мануального тестування на відповідність функціональних вимог технічному завданню;
- перевірка користування у різних браузерах;
- користування системи та висновком щодо її зручності;
- звіряння графічного інтерфейсу із макетами.

					КПІ.ІП-9222.045430.04.51	Арк.
						6
Змін	Арк.	№ докум.	Підп.	Дата.		



Факультет інформатики та обчислювальної техніки  
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Едуард ЖАРІКОВ

“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ЗБОРУ ТА АНАЛІЗУ ДАНИХ ПРО  
РОСІЙСЬКИХ ВІЙСЬКОВОСЛУЖБОВЦІВ, ЩО БЕРУТЬ УЧАСТЬ У  
ВІЙНІ ПРОТИ УКРАЇНИ**

**Керівництво користувача**

КП.ІП-9222.045430.05.34

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_ Юрій ОЛІЙНИК

Нормоконтроль:

\_\_\_\_\_ Катерина ЛІЩУК

Виконавець:

\_\_\_\_\_ Вікторія ТКАЧЕНКО

Київ – 2023

## ЗМІСТ

1	ПРИЗНАЧЕННЯ ПРОГРАМИ .....	3
2	ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ.....	4
2.1	Системні вимоги для коректної роботи .....	4
2.2	Завантаження застосунку .....	4
2.3	Перевірка коректної роботи .....	4
3	ВИКОНАННЯ ПРОГРАМИ.....	5

					КПІ.ІП-9222.045430.05.34	Арк.
						2
Змін.	Арк.	№ докум.	Підп.	Дата.		

## 1 ПРИЗНАЧЕННЯ ПРОГРАМИ

«Яструб» - це веб-додаток для збору та аналізу даних про російських військовослужбовців, що беруть участь у війні проти України. Кінцева збірка програмного забезпечення представляє собою повністю функціональний і готовий до використання продукт. У репозиторії на GitHub зберігаються всі вихідні файли коду, що використовуються в процесі розробки. Вміст репозиторію включає окремі модулі, бібліотеки, скрипти, конфігураційні файли, залежності та інші компоненти, необхідні для збірки та встановлення програмного забезпечення.

					КПІ.ІП-9222.045430.05.34	Арк.
						3
Змін.	Арк.	№ докум.	Підп.	Дата.		

## 2 ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ

### 2.1 Системні вимоги для коректної роботи

Для успішної роботи даного застосунку необхідне виконання наступних вимог:

- комп'ютер, що має доступ до мережі інтернет;
- процесор частотою не меншою за 1GHz;
- мінімальна роздільна здатність 800x600 та вище.

Мінімальні вимоги для версій браузерів:

- Chrome 60+;
- Safari 10.1+ / iOS Safari 10.1+;
- Edge 12+;
- Firefox ESR+;
- Opera.

### 2.2 Завантаження застосунку

Розгортання веб-додатку описано в 4 розділі пояснювальної записки дипломного проєкту КПІ.ІП-9222.045430.01.81. Підтримка програмного забезпечення реалізована за допомогою Git та Heroku.

### 2.3 Перевірка коректної роботи

По завершенню розгортання додатка необхідно перейти за посиланням наданим Heroku. Після переходу по посиланню повинна відобразитись початкова сторінка застосунку.

					КПІ.ІП-9222.045430.05.34	Арк.
						4
Змін.	Арк.	№ докум.	Підп.	Дата.		

### 3 ВИКОНАННЯ ПРОГРАМИ

Початкову сторінку застосунку можна побачити на рисунку 3.1:

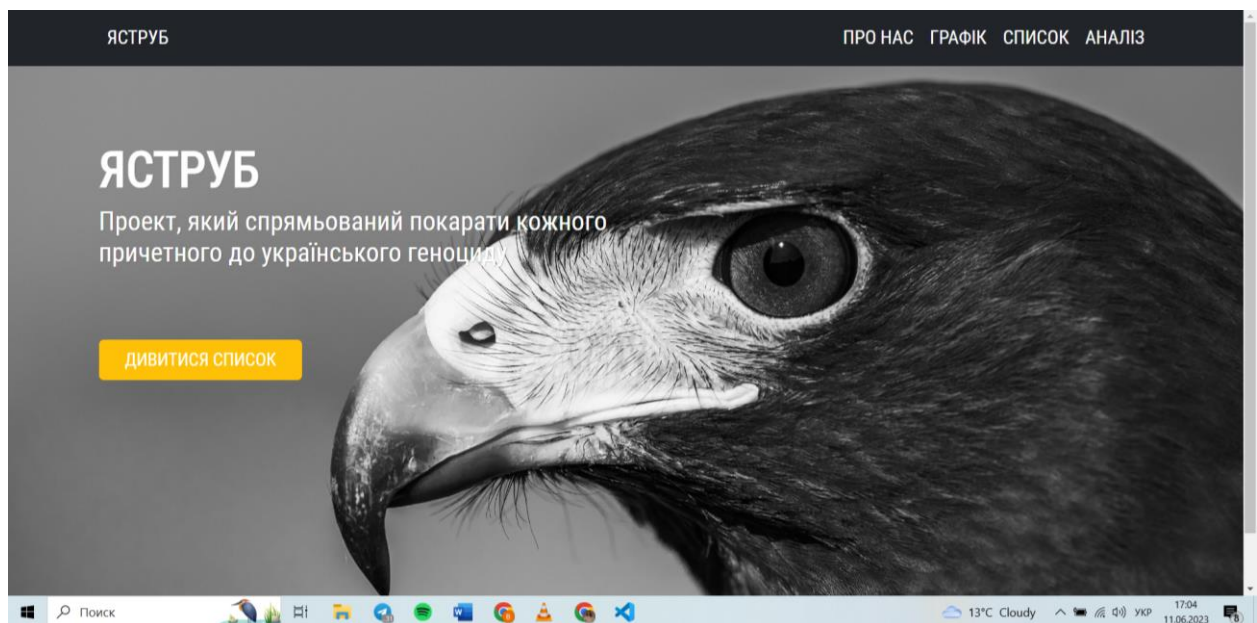


Рисунок 3.1 – Початкова сторінка додатку

Користувач має змогу перейти на будь яку сторінку, що його цікавить. Для перегляду бази даних про російських військослужбовців можна скористатись таблицею, яку можна знайти на сторінці «Список», що зображено на рисунку 3.2:

ЯСТРУБ

ПРО НАС ГРАФІК СПИСОК АНАЛІЗ

Таблиця пошуку

Search

< 1 2 3 4 5 6 7 8 9 10 ... 2145 2146 >

#	ПІБ	Звання	Місце проживання	Дата народження	Більше інформації
1	Ендовицкий Павел Иванович	капитан	Куйбышевская область г. Толятти	05.07.1989	<a href="#">Дізнатися більше</a>
2	Кравцов Сергей Викторович	старший сержант	Кировская область, Верхнекамский район, пос. Тупрунка	12.12.1988	<a href="#">Дізнатися більше</a>
3	Мункуев Тумэн Мункоевич	младший сержант	Читинская области Дульдургинский район с. Чиндалей	07.01.1994	<a href="#">Дізнатися більше</a>
4	Киргизов Игорь Андреевич	матрос	Приморский край Яковлевский район с.Варфоломеевка	06.11.1995	<a href="#">Дізнатися більше</a>
5	Леорда Артур Владимирович	матрос	Приморский край,г.Владивосток	29.07.2002	<a href="#">Дізнатися більше</a>

Рисунок 3.2 – Сторінка «Список»

Для пошуку по базі даних, можна скористатись пошуковим полем. Для цього необхідно ввести ПІБ та натиснути «Enter». Результати пошуку будуть виведені на ту ж саму сторінку, що можна побачити на рисунку 3.3

Кузнецов Владислав Юрьевич					
#	ПІБ	Звання	Місце проживання	Дата народження	Більше інформації
751	Кузнецов Владислав Юрьевич	Младший сержант		1996-9-20	<a href="#">Дізнатися більше</a>
59968	Кузнецов Владислав Юрьевич			1993-01-26	<a href="#">Дізнатися більше</a>

Рисунок 3.3 – Результати пошуку за ПІБ військового

При відсутності військового в базі даних, буде виведено повідомлення «Немає записів». Даний випадок можна розглянути на рисунку 3.4:

Таблиця пошуку					
Кузнецова Наталия Сергеевна					
Немає записів					
#	ПІБ	Звання	Місце проживання	Дата народження	Більше інформації

Рисунок 3.4 – Результати пошуку за ПІБ військового

Для перегляду інфографіки можна скористатись сторінкою «Графіки». До перегляду користувачу представлені графіки, що наведені на рисунку 3.5:

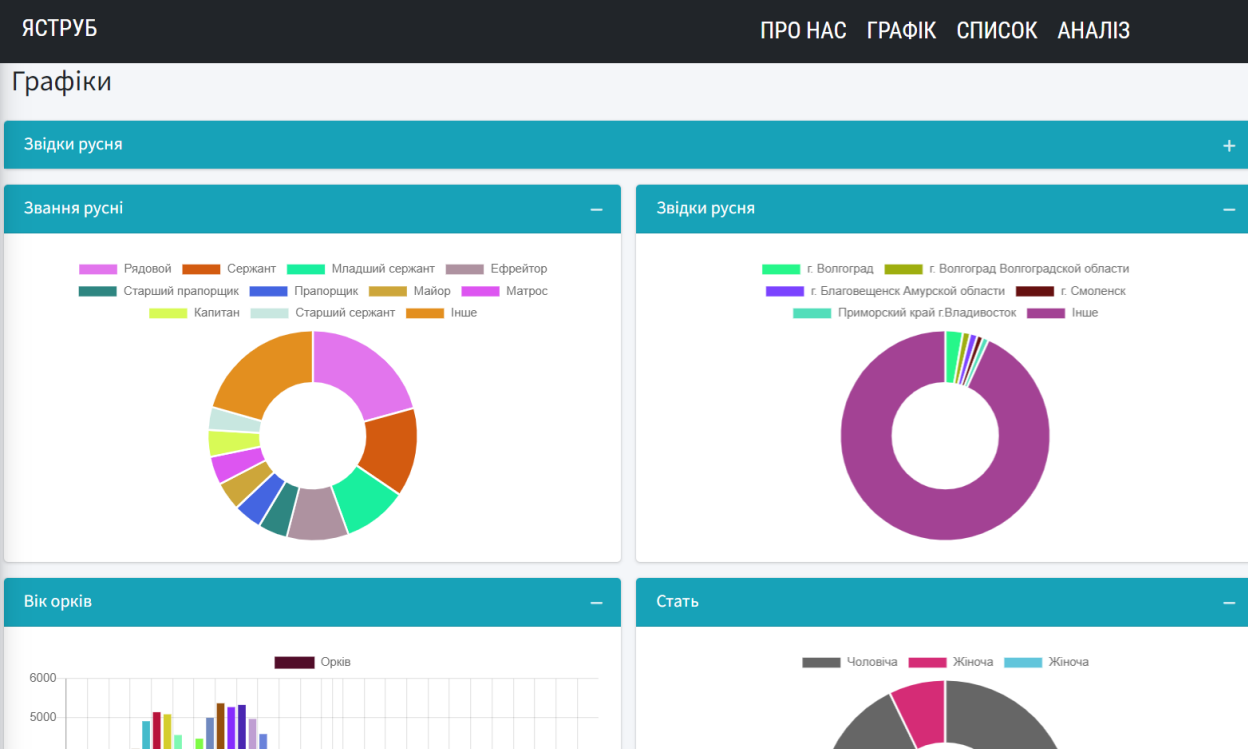
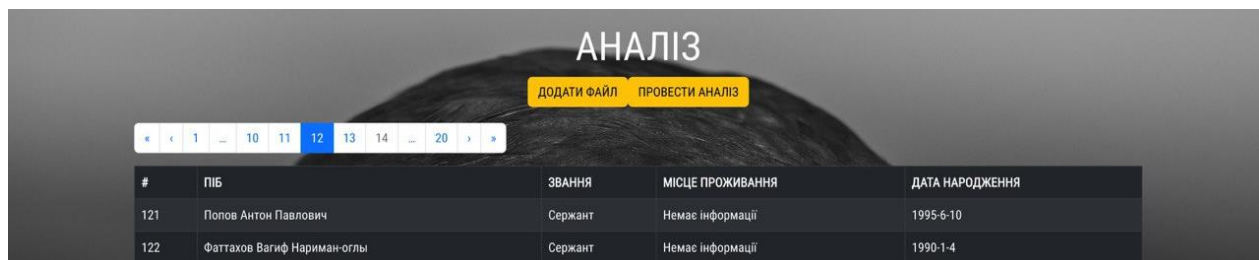


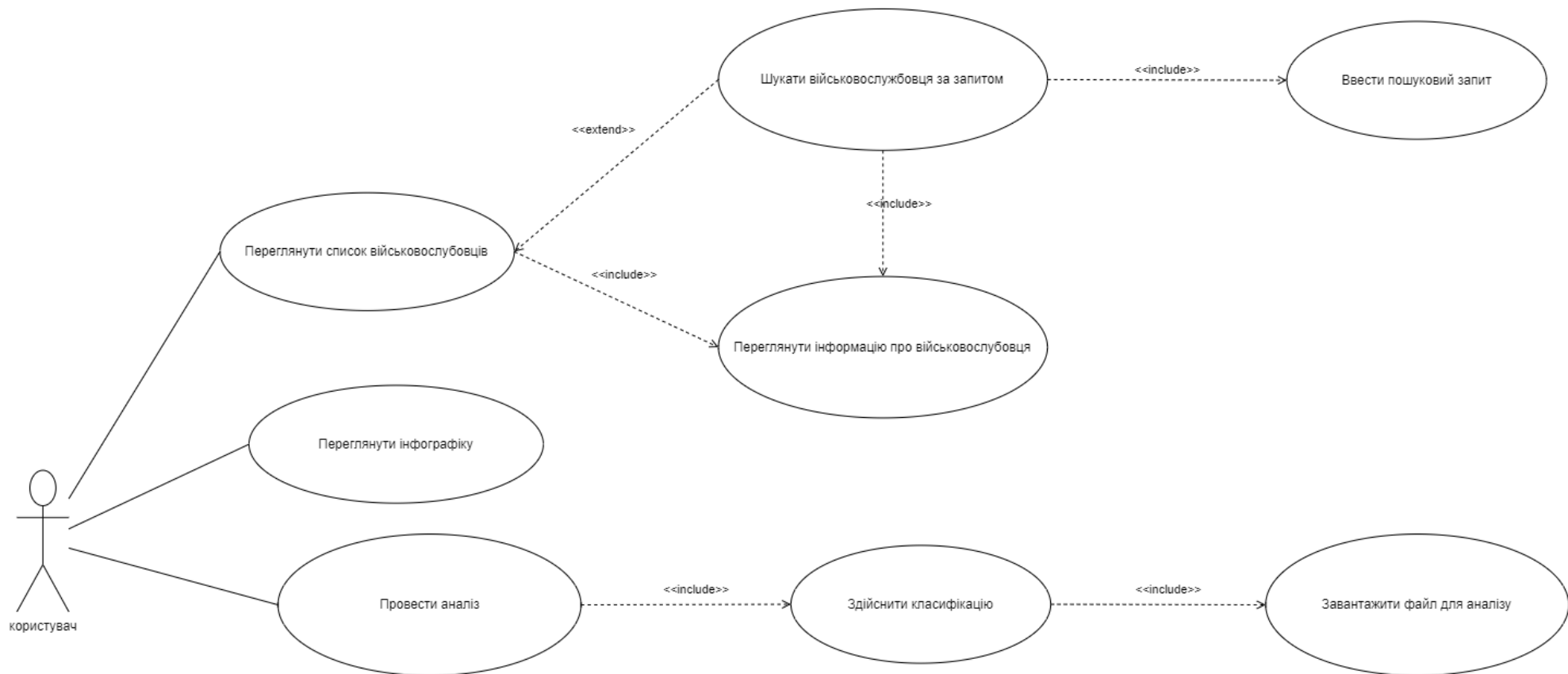
Рисунок 3.5 – Перегляд інфографіки

Якщо користувач хоче розпарсити та проаналізувати свої дані, їх можна завантажити на сторінці «Аналіз» за допомогою кнопки «Додати файл». Після чого необхідно натиснути кнопку «Провести аналіз». При відсутності даних необхідних для тої чи іншої колонки, в комірку буде записано «немає інформації». На рисунку 3.6 зображено результати:



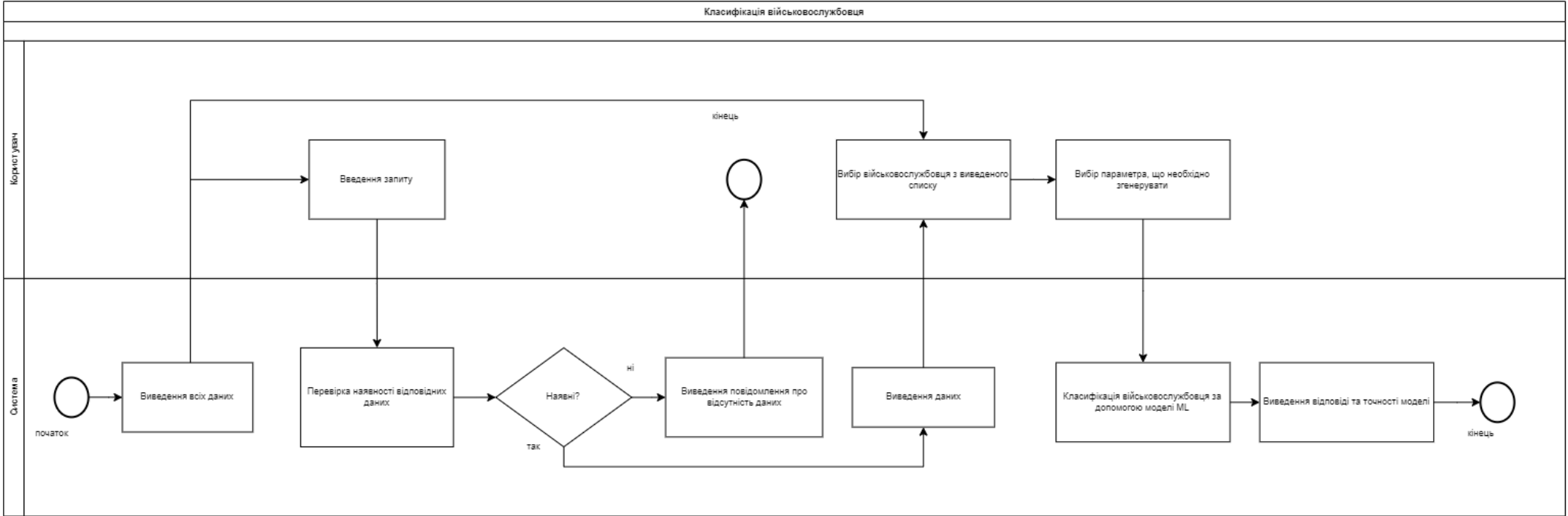
#	ПІБ	ЗВАННЯ	МІСЦЕ ПРОЖИВАННЯ	ДАТА НАРОДЖЕННЯ
121	Попов Антон Павлович	Сержант	Немає інформації	1995-6-10
122	Фаттахов Вагиф Нариман-оглы	Сержант	Немає інформації	1990-1-4

Рисунок 3.6 – Перегляд завантаженого та обробленого файлу

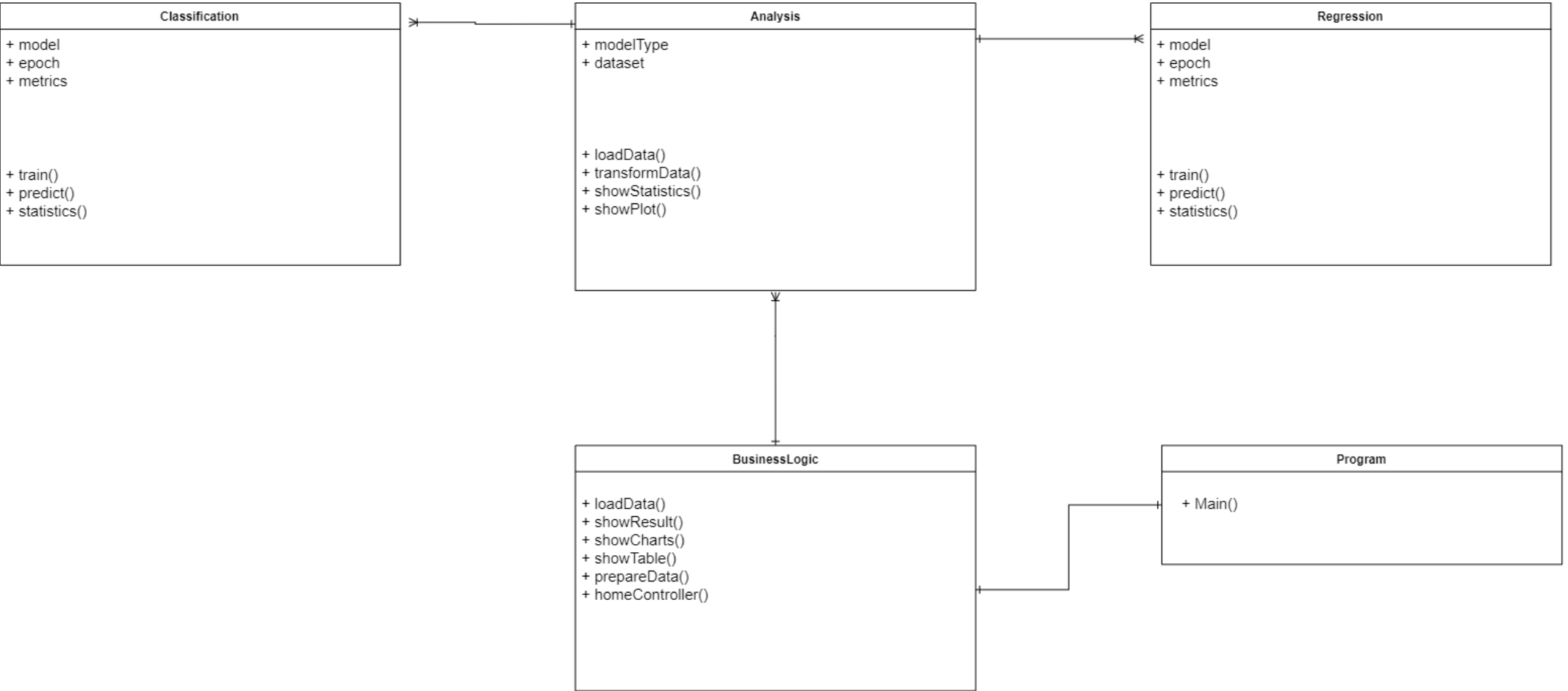


					КПІ.ІП-9222.045430.06.99.ССВ				
					Схема структурна варіантів використань	Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Ткаченко В.В.							
Перевірів		Олійник Ю.О.							
Т. кон.									
Н. кон.		Ліщук К.І.			Програмне забезпечення збору та аналізу даних про російських військовослужбовців що беруть участь у війні в Україні	КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІП-92			
Затвердив		Жаріков Е.В.							

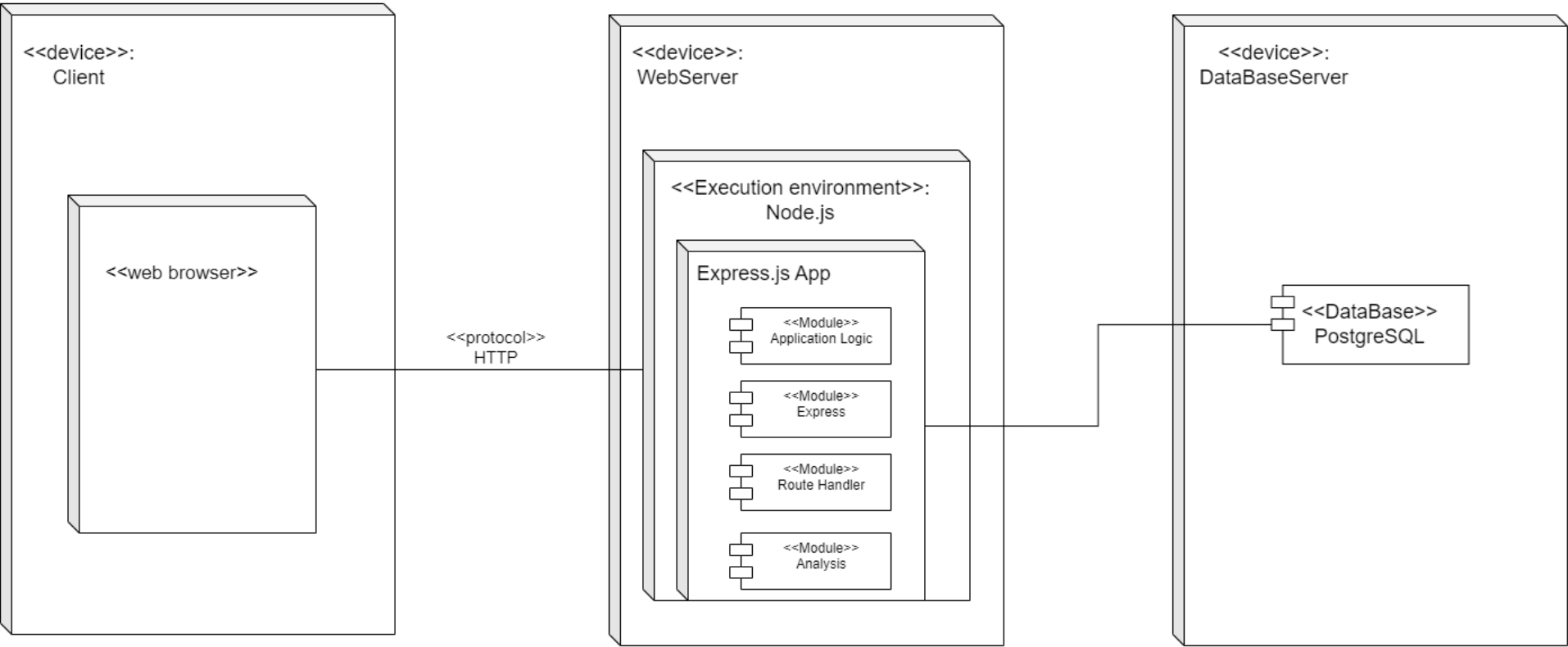




						КПІ.ІПТ-9222.045430.07.99.CCB			
						Схема структурна бізнес-процесів			
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Ткаченко В.В.							
Перевірів		Олійник Ю.О.							
Т. кон.									
Н. кон.		Ліщук К.І.							
Затвердив		Жаріков Е.В.							
Програмне забезпечення збору та аналізу даних про російських військовослужбовців, що беруть участь у війні в Україні						КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІП-92			
						Літера		Маса	Масштаб
						Аркуш		Аркушів	



						КПІ.ІП-9222.045430.08.99.ССК					
Зм.	Арк.	№ документа	Підпис	Дата	Схема структурна класів програмного забезпечення	Літера			Маса	Масштаб	
Розробив		Ткаченко В.В.									
Перевішив		Олійник Ю.О.									
Т. кон.						Аркуш			Аркушів		
Н. кон.		Ліщук К.І.			Програмне забезпечення збору та аналізу даних про російських військовослужбовців що беруть участь у війні в Україні	КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІП-92					
Затвердив		Жаріков Е.В.									



					КПІ.ІП-9222.045430.09.99.CCP				
					Схема структурна розгортання програмного забезпечення	Літера	Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Ткаченко В.В.							
Перевірів		Олійник Ю.О.							
Т. кон.						Аркуш	Аркушів		
Н. кон.		Ліщук К.І.			Програмне забезпечення збору та аналізу даних про російських військовослужбовців, що беруть участь у війні в Україні	КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІП-92			
Затвердив		Жаріков Е.В.							