

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри _____ Є. О. Давиденко

підпис

«_____» _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

**«СИСТЕМА ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ
СПОРТИВНИХ РЕЗУЛЬТАТІВ НА ОСНОВІ
МАШИННОГО НАВЧАННЯ»**

Спеціальність «Інженерія програмного забезпечення»

121 – КРМ.1 – 608м.21810823

Здобувач

_____ О. О. Постриган

підпис

«_____» _____ 2024 р.

Керівник канд. техн. Наук, доцент

_____ Г. В. Горбань

підпис

«_____» _____ 2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет імені Петра Могили
Факультет комп'ютерних наук
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри інженерії програмного
забезпечення, канд.техн.наук, доцент,

_____Є. О. Давиденко

«_____» 2023 р.

ЗАВДАННЯ
на виконання роботи магістра

Видано здобувачу групи 608м факультету комп'ютерних наук

_____Постриган Олексій Олексійович

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

«Система для аналізу та прогнозування спортивних результатів на основі машинного навчання».

Затверджена наказом по ЧНУ від «10» _____ листопада _____ 2023 р. № 234

2. Строк представлення кваліфікаційної роботи «27» _____ лютого 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Вхідні дані до роботи – функціональні та нефункціональні вимоги до програмного забезпечення системи для аналізу та прогнозування спортивних результатів на основі машинного навчання. Результат – програмне забезпечення для аналізу та прогнозування спортивних результатів на основі машинного навчання.

4. Перелік питань, що підлягають розробці:

– проаналізувати ринок наявних технологій і рішень для вирішення завдань;

– зібрати та обробити дані тренувань;

– виконати проектування та моделювання системи;

- використати моделі машинного навчання для аналізу отриманих даних;
- оцінити ефективність моделей та їх можливість впровадження у практику академічного веслування;
- апробувати отримані результати.

5. Перелік графічних матеріалів:

Презентація _____.

Керівник роботи _____ канд. техн. наук., доцент Г. В. Горбань
(посада, прізвище, ім'я, по батькові)

(підпис)

Завдання прийнято до виконання

_____ Постриган Олексій Олексійович
(прізвище, ім'я, по батькові студента)

(підпис)

Дата видачі завдання «26» _____ листопада _____ 2023 р.

КАЛЕНДАРНИЙ ПЛАН Виконання кваліфікаційної роботи

Тема: «Система для аналізу та прогнозування спортивних результатів на основі машинного навчання».

№	Найменування роботи	Початок	Закінчення	Примітки
1	Розробка та затвердження завдання на практику	27.11.2023	01.12.2023	Виконано
2	Огляд літератури за темою роботи	04.12.2023	08.12.2023	Виконано
3	Аналіз предметної області	11.12.2023	13.12.2023	Виконано
4	Аналіз ринку наявних технологій і рішень для вирішення завдань	14.12.2023	15.12.2023	Виконано
5	Розробка вимог до системи для аналізу та прогнозування спортивних результатів на основі машинного навчання	18.12.2023	20.12.2023	Виконано
6	Проектування системи для аналізу та прогнозування спортивних результатів на основі машинного навчання	21.12.2023	11.01.2024	Виконано
7.	Реалізація та налагодження системи	12.01.2024	24.01.2024	Виконано
8.	Відгук керівника КРМ	25.01.2024	31.01.2024	Виконано
9.	Оформлення КРМ та презентації	01.02.2024	07.02.2024	Виконано
10.	Попередній захист	08.02.2024	08.02.2024	Виконано
11.	Рецензування	09.02.2024	10.02.2024	Виконано
12.	Завершення оформлення КРМ та презентації	11.02.2024	25.02.2024	Виконано
13.	Захист кваліфікаційної роботи	27.02.2024	27.02.2024	Виконано

Розробив здобувач Постриган Олексій Олексійович _____
(прізвище, ім'я, по батькові студента) (підпис)

«26» листопада 2023р.

Керівник роботи канд. техн. наук, доцент Горбань Гліб Валентинович _____
(посада, прізвище, ім'я, по батькові) (підпис)

«26» листопада 2023р.

АНОТАЦІЯ

До кваліфікаційної роботи магістра

«Система для аналізу та прогнозування спортивних результатів на основі машинного навчання»

Здобувач 608м гр.: Постриган Олексій Олексійович

Керівник: канд. техн. наук, доцент Горбань Гліб Валентинович

Тема кваліфікаційної роботи є актуальною, оскільки застосування машинного навчання для аналізу та прогнозування результатів академічного веслування відіграє важливу роль в інформатизації вищої освіти. Машинне навчання відкриває можливості для покращення підготовки спортсменів та оптимізації тренувального процесу. Крім того, воно підкреслює важливість використання сучасних технологій для досягнення вищих спортивних результатів та підвищення конкурентоспроможності навчальних закладів.

Об'єктом дослідження є процес аналізу та прогнозування показників спортсменів з академічного веслування та результатів їх тренувань на веслувальному тренажері.

Предметом дослідження є методи машинного навчання для прогнозування на основі аналізу спортивних результатів.

Мета кваліфікаційної роботи – покращення якості прогнозування спортивних результатів шляхом застосування машинного навчання для аналізу та прогнозування їх результатів.

Для досягнення поставленої мети було розв'язано наступні завдання:

- проаналізовано ринок наявних технологій і рішень для вирішення завдань;
- зібрано та оброблено дані тренувань;
- виконано проектування та моделювання системи;
- використано моделі машинного навчання для аналізу отриманих даних;
- оцінено ефективність моделей та їх можливість впровадження у практику академічного веслування;
- апробовано отримані результати.

В ході виконання даної кваліфікаційної роботи магістра було проведено аналіз предметної області систем для аналізу та прогнозування спортивних результатів на основі машинного навчання.

В першому розділі роботи були досліджені інтелектуальні підходи, такі як кластерний аналіз, алгоритми регресії, метод опорних векторів, метод нечіткої логіки та нейронні мережі, що дозволило визначити оптимальний набір інструментів для подальшої розробки системи.

У другому розділі була описана розробка функціональної моделі системи та побудовано контекстну діаграму, що деталізує внутрішні процеси системи. Це забезпечило чітке розуміння логіки та взаємозв'язків компонентів системи для ефективного прогнозування спортивних результатів.

Третій розділ роботи присвячено технологіям, необхідним для реалізації системи, зокрема Python, бібліотекам Pandas та Scikit-learn. Створення діаграм прецедентів та сценаріїв використання сприяло кращому розумінню функціоналу системи та взаємодії з нею. Розробка макетів інтерфейсу дозволила візуалізувати функціонал системи та спростити її використання.

У четвертому розділі було успішно описано процес розробки застосунку та проведено аналіз результатів, що отримані за допомогою алгоритмів кластеризації та прогнозування на основі машинного навчання. Це дозволило оцінити ефективність розробленої системи та її здатність до точного прогнозування спортивних результатів.

КРБ викладена на 70 сторінок, вона містить 4 розділи, 19 ілюстрації, 8 таблиць, 22 джерелач в переліку посилань.

Ключові слова: машинне навчання, застосунок, ІС, кластеризація, спорт.

ABSTRACT

of the Master's Thesis

"System for analysis and prediction of sports results based on machine learning"

Student of group 608M: Postryhan Oleksii Oleksiiiovych

Supervisor: PhD in Dr.Sc., Professor Hlib Horban

The topic of the qualification work is relevant, as the application of machine learning for analysis and forecasting of the results of academic rowing plays an important role in the informatization of higher education. Machine learning opens up opportunities for improving the preparation of athletes and optimizing the training process. In addition, it emphasizes the importance of using modern technologies to achieve higher sports results and increase the competitiveness of educational institutions.

The object of research is the process of analysis and forecasting of the indicators of athletes from academic rowing and the results of their training on the rowing simulator.

The subject of research is machine learning methods for forecasting based on the analysis of sports results.

The purpose of the qualification work is to improve the quality of forecasting sports results by applying machine learning for analysis and forecasting of their results.

To achieve the goal, it is necessary to solve the following tasks:

- the market of existing technologies and solutions for task resolution was analyzed;
- training data was collected and processed;
- the system design and modeling have been completed;
- machine learning models were used to analyze the obtained data;
- the effectiveness of the models and their potential for implementation in academic rowing practice were evaluated;
- the obtained results were tested.

In the course of completing this master's qualification work, an analysis of the subject area of systems for analyzing and predicting sports results based on machine

learning was conducted. In the first section of the work, intelligent approaches such as cluster analysis, regression algorithms, support vector machines, fuzzy logic methods, and neural networks were investigated, allowing for the determination of an optimal set of tools for further system development.

The second section described the development of the functional model of the system and constructed a context diagram detailing the internal processes of the system. This ensured a clear understanding of the logic and interrelationships of system components for effective sports result prediction.

The third section focused on the technologies necessary for system implementation, including Python, Pandas, and Scikit-learn libraries. Creating use case diagrams and usage scenarios improved the understanding of the system's functionality and interaction. Interface mockups facilitated visualizing the system's features and simplified its use.

In the fourth section, the successful process of application development was described, and an analysis of the results obtained using clustering and machine learning-based prediction algorithms was conducted. This allowed for evaluating the effectiveness of the developed system and its ability to accurately predict sports results.

The work is 70 pages long and includes 4 sections, 19 illustrations, 8 tables, and 22 sources.

Keywords: Artificial Intelligence, Machine Learning, IS, sport.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ	4
ВСТУП.....	5
1 ДОСЛІДЖЕННЯ СИСТЕМ ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ СПОРТИВНИХ РЕЗУЛЬТАТІВ	7
1.1 Сутність систем для аналізу та прогнозування спортивних результатів	7
1.2 Класифікація систем для аналізу та прогнозування спортивних результатів	10
1.3 Інтелектуальні підходи до аналізу та оптимізації спортивних результатів за допомогою машинного навчання	11
1.3.1 Кластерний аналіз	11
1.3.2 Алгоритми регресії	12
1.3.3 Метод опорних векторів.....	13
1.3.4 Метод нечіткої логіки.....	13
1.3.5 Нейронні мережі	14
1.4 Аналіз аналогічних систем.....	15
1.4.1 Soccermetrics	16
1.4.2 IBM Watson for Wimbledon	16
1.5 Постановка задачі.....	17
1.5.1 Опис основних функцій системи	17
1.6 Специфікації вимог до програмного забезпечення	17
2 АНАЛІЗ НАБОРУ ДАНИХ ТА ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ.....	23
2.1 Опис набору даних.....	23
2.2 Функціональна модель системи	25
Висновок до розділу 2	29
3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	30
3.1 Опис використаних технологій	30
3.1.1 Python	30

3.1.2 Pandas	31
3.1.3 Scikit-learn	32
3.1.4 Tkinter	33
3.2 Макети інтерфейсу	34
3.2.1 Головна сторінка	35
3.2.2 Вікно редагування рядків таблиці	36
3.3 Діаграма прецедентів та сценарії використання	38
3.4 Діаграма класів	44
3.4.1 Опис класу AthleteDataMode	45
3.4.2 Опис класу GUI	46
3.4.3 Опис класу EditWindow	46
Висновок до розділу 3	47
4. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	48
4.1 Аналіз результатів	50
4.1.1 Групування студентів за їх характеристиками	50
4.1.2 Розрахунок середнього віку за національністю	51
4.2 Прогнозування результатів	52
4.2.1 Прогнозування часу проходження студентом 500-та метрів	52
4.2.2 Прогнозування статі студента за його фізичними параметрами з використанням методу опорних векторів	54
Висновок до розділу 4	55
ВИСНОВКИ	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	58
ДОДАТОК А Код імпорту, виведення, редагування, видалення та збереження даних	61
ДОДАТОК Б Код прогнозування статі студента за його фізичними параметрами з використанням методу опорних векторів	68

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ

- IT – інформаційні технології;
- IS – інформаційна система;
- ML – Machine Learning;
- AI – Artificial Intelligence;
- SL – Supervised learning;
- CSV – Comma-Separated Values;

ВСТУП

У зв'язку з розвитком інформаційних технологій відкриваються нескінченні можливості для застосування у різних сферах, які впливають на всі аспекти життя, також беручі до уваги вищу освіту та спорт. Зокрема, академічне веслування, спорт, що вимагає від учасників великого фізичного і технічного навантаження, також відчуває на собі вплив інформаційних технологій. Застосування машинного навчання (Machine Learning) стає дедалі більш важливим інструментом у цій галузі.

Тема кваліфікаційної роботи є актуальною, оскільки застосування машинного навчання для аналізу та прогнозування результатів академічного веслування відіграє важливу роль в інформатизації вищої освіти. Машинне навчання відкриває можливості для покращення підготовки спортсменів та оптимізації тренувального процесу. Крім того, воно підкреслює важливість використання сучасних технологій для досягнення вищих спортивних результатів та підвищення конкурентоспроможності навчальних закладів.

Об'єктом дослідження є процес аналізу та прогнозування показників спортсменів з академічного веслування та результатів їх тренувань на веслувальному тренажері.

Предметом дослідження є методи машинного навчання для прогнозування на основі аналізу спортивних результатів.

Мета кваліфікаційної роботи – покращення якості прогнозування спортивних результатів шляхом застосування машинного навчання для аналізу та прогнозування їх результатів.

Для досягнення поставленої мети необхідно розв'язати наступні завдання:

- проаналізувати ринок наявних технологій і рішень для вирішення завдань;
- зібрати та обробити дані тренувань;
- виконати проектування та моделювання системи;
- використати моделі машинного навчання для аналізу отриманих даних;

– оцінити ефективність моделей та їх можливість впровадження у практику академічного веслування;

– апробувати отримані результати.

Апробація результатів. Окремі аспекти дослідження були висвітлені на Всеукраїнській науково-практичній конференції молодих вчених, аспірантів і студентів «Інформаційні технології та інженерія – 2024»:

– Постриган О. О., Горбань Г. В. Система для аналізу та прогнозування спортивних результатів на основі машинного навчання // «Інформаційні технології та інженерія – 2024». Всеукраїнська науково-практична конференція молодих вчених, аспірантів і студентів. «Досвід та тенденції розвитку суспільства в Україні: глобальний, національний та регіональний аспекти : тези. Миколаїв : Вид-во ЧНУ імені Петра Могили, 2024. С. 57-59.

1 ДОСЛІДЖЕННЯ СИСТЕМ ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ СПОРТИВНИХ РЕЗУЛЬТАТІВ

1.1 Сутність систем для аналізу та прогнозування спортивних результатів

Спорт – це не лише здоровий спосіб життя, але і величезна індустрія, у якій конкурують мільйони людей, команд та організацій, тому для досягнення успіху в спорті необхідно не тільки талант та удача, а й науковий підхід до планування, підготовки та прогнозування спортивних результатів, а особливо з результатів з академічного веслування. У цьому контексті, використання інформаційних технологій, а особливо машинного навчання відіграє важливу роль, оскільки дозволяють оброблювати велику кількість даних, виявити зв'язки та моделювати відповідні ситуації.

Академічне веслування – це вид спорту, який передбачає змагання на спеціальних човнах, пливучи за допомогою весел. Академічне веслування відрізняється від інших видів веслування кріпленням весел до човна за допомогою півобертальних петель, а спортсмени при цьому сидять спиною до напрямку руху [1].

Академічне веслування вимагає від спортсменів високого рівня фізичної і функціональної підготовки, технічної майстерності та психологічної стійкості. Особливо потрібно зазначити синхронізацію роботи всіх членів екіпажу, яка в свою чергу залежить від сумісності морфологічних, фізіологічних та психологічних характеристик. Для підвищення ефективності тренувального процесу в академічному веслуванні застосовуються сучасні методи моделювання, які дозволяють визначити оптимальну структуру спортивної діяльності та підготовки спортсменів. Аналіз даних тренувань в академічному веслуванні може бути здійснено за допомогою різних джерел інформації.

Аналіз даних тренувань в академічному веслуванні є важливим інструментом для контролю та корекції тренувального процесу, виявлення

переваг та недоліків їх підготовки та прогнозування майбутніх спортивних результатів. Для аналізу даних тренувань в академічному веслуванні зазвичай використовуються протоколи змагань та тренувань, відеозаписи, анкети, педагогічні спостереження, фізіологічні та біомеханічні показники, дані з веслувальних тренажерів тощо [2].

Беручи до уваги дослідження [2], то можна побачити, що одним з найпоширеніших і найефективніших засобів для аналізу даних тренувань в академічному веслуванні є веслувальні тренажери Concept2, які дозволяють симулювати реальні умови веслування на човні, вимірювати різні параметри веслувальної діяльності, такі як час, відстань, швидкість, потужність, частота гребків, сила тяги, кут повороту весла тощо, зберігати, переглядати і порівнювати дані тренувань, змагатися з іншими спортсменами онлайн або офлайн, використовувати різні програми і ігри для різноманітності тренувань.

Потрібно зазначити, що веслувальні тренажери Concept2 оснащені моніторами PM5, які є «мозком» тренажера і надають дані тренувань, легку навігацію, попередньо запрограмовані і користувацькі тренування та інші функції.

Для аналізу даних тренувань, є можливість експортувати дані з монітора PM5 на флешку, для подальшої обробки. Дані тренувань експортуються в форматі CSV. Головною проблемою експорту даних з тренажера є те, що дані попередньо зберігаються не в дуже гарному форматі, оскільки формуються у вигляді тренувань за увесь рік, тому для подальшого аналізу отриманих даних, потрібно провести первинну обробку цих даних [3].

Машинне навчання – це розділ штучного інтелекту, який в свою чергу вивчає методи побудови алгоритмів, здатних навчатися з урахуванням даних, без явного завдання правил. Машинне навчання зазвичай використовується для таких задач, як регресія, класифікація та кластеризація [5].

Одним з напрямків, де інформаційні технології розкривають свій потенціал, є аналіз та прогнозування спортивних результатів, використовуючи системи, які базуються на машинному навчанні [4].

Машинне навчання – це передова технологія, яка має великий потенціал для інтеграції у різноманітні галузі, особливо ті, які пов'язані з аналізом та прогнозуванням. [4] Однією з таких областей є прогнозування спортивних результатів, яке все частіше обговорюється як перспективна область у спортивній науці [3].

Важливою ланкою у цьому контексті є інформаційна система, яка забезпечує обробку та зберігання даних, які було отримано внаслідок спортивних тренувань, проведених з використанням тренажерів та вбудованих датчиків. Ці системи вирізняються інтеграцією передових технологій машинного навчання та аналітики даних, що в свою чергу створює ефективний інструмент для аналізу та прогнозування спортивних результатів [2-4].

Системи для аналізу та прогнозування спортивних результатів об'єднують в собі методології на основі набору результатів спортивних тренувань. Суть систем для аналізу та прогнозування спортивних результатів пролягає в інтеграції методологій на основі даних для отримання цінної інформації про результативність, стратегії та тенденції у сфері спорту. Вони використовують передові технології та статистичні методи для обробки величезних обсягів даних, дозволяючи командам, тренерам і аналітикам приймати обґрунтовані рішення та прогнози.

Інтеграція машинного навчання та аналітики даних у розробці спортивних інформаційних систем дозволяє автоматизовано оцінювати значення параметрів спортивних даних. Використання концепції Data Mining підвищує ефективність аналізу та прогнозу спортивних результатів на основі накопичених даних [2].

Тому можна зробити висновок, що подібний підхід до аналізу спортивних результатів створює передумови для більш точних прогнозів та сприяє оптимізації тренувальних процесів.

1.2 Класифікація систем для аналізу та прогнозування спортивних результатів

Класифікація систем для аналізу та прогнозування спортивних результатів на основі машинного навчання відбувається за декількома критеріями, такими як вид спорту, тип даних, які використовуються для аналізу, та тип машинного навчання, якій використовуються для прогнозування результатів.

Системи для аналізу та прогнозування спортивних результатів на основі МН можна класифікувати за декількома критеріями, такими як:

– вид спорту. Залежно від специфіки та правил кожного виду спорту, можуть використовуватися різні джерела даних, методи аналізу та прогнозування, а також різні цілі та завдання. Наприклад, для командних видів спорту, таких як футбол, баскетбол, хокей тощо, можна аналізувати та прогнозувати результати матчів, турнірні таблиці, статистику гравців, індивідуальні та колективні показники ефективності тощо. Для індивідуальних видів спорту, таких як легка атлетика, плавання, теніс тощо, можна аналізувати та прогнозувати результати змагань, рейтинги спортсменів, технічні та тактичні характеристики виконання вправ тощо;

– тип даних. Залежно від доступності та якості даних, можуть використовуватися різні формати та джерела даних, такі як: історичні дані про результати попередніх змагань, статистичні дані про характеристики спортсменів та команд, біометричні дані про фізичний стан та здоров'я спортсменів, відео- та аудіодані про хід змагань, текстові дані про новини, коментарі, соціальні мережі тощо;

– тип машинного навчання. Залежно від типу завдання та типу даних, можуть використовуватися різні методи та алгоритми МН, такі як: навчання з учителем (класифікація, регресія тощо), навчання без учителя (кластеризація, зниження розмірності, асоціативні правила тощо), навчання з підсиленням (Q-навчання, SARSA, глибоке навчання з підсиленням тощо), глибоке навчання

(згорткові нейронні мережі, рекурентні нейронні мережі, генеративні змагальні мережі тощо).

1.3 Інтелектуальні підходи до аналізу та оптимізації спортивних результатів за допомогою машинного навчання

Сучасна область досліджень спортивних результатів все більше орієнтується на використання сучасних методів штучного інтелекту для аналізу та прогнозування спортивних результатів, а саме з академічного веслування. Серед цих методів виокремлюються кластерний аналіз, алгоритми регресії, метод опорних векторів, метод К-найближчих сусідів, асоціативні правила, нейронні мережі та методи нечіткої логіки. Ці методи використовуються для класифікації, кластеризації та прогнозування конкретних спортивних даних [3].

1.3.1 Кластерний аналіз

Кластерним аналізом називають метод групування об'єктів, що мають спільні характеристики, в кластери. Він зазвичай використовується для визначення схожості між об'єктами та виявлення структури в наборі даних [6].

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

де $D(x, y)$ – це відстань між об'єктами x та y ;

n – кількість характеристик.

Для прикладу, кластерний аналіз можна використати для групування спортсменів за їхніми фізичними параметрами та результатами тестів, що в свою чергу дозволяє тренерам ідентифікувати групи зі схожими характеристиками та визначити індивідуальні підходи для тренувань.

Класифікацію даних можна назвати фундаментальним елементом машинного навчання, яка має на меті можливість передбачити результат цільової змінної на основі попередньо невідомих даних, що у випадку аналізу та

прогнозування спортивних результатів, може допомогти передбачити результат тренування з академічного веслування [4].

1.3.2 Алгоритми регресії

Алгоритми регресії зазвичай використовуються для моделювання відносин між залежними Y та незалежними X змінними. Регресивні моделі поділяються на лінійну, множинну, поліноміальну та логістичну регресію.

Лінійна регресія є простою моделлю, за допомогою якої можна визначити відношення між залежною змінною Y та однією чи більше незалежними X . Беручи до уваги контекст аналізу спортивних результатів, то її можна інтерпретувати наступною формулою [7]:

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon \quad (2)$$

де Y – це результат тренування;

X_1 – фактор, який впливають на тренування;

β_0, β_1 – коефіцієнти регресії;

ε – помилка.

Множинна регресія в свою чергу використовується тоді, коли є більше одного чинника, який впливає на незалежну змінну. Формула виглядатиме наступним чином:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon \quad (3)$$

де X_1, X_2, \dots, X_n виступають у ролі факторів, які впливають на результат тренування.

Поліноміальна регресія використовується для моделювання нелінійних відносин між факторами та залежною змінною. Формула виглядатиме наступним чином:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_n X^n + \varepsilon$$

де тепер X^2, X^3, \dots, X^n – це ступені чинника X .

Використання логістичної регресії в свою чергу дозволяє прогнозувати ймовірність настання події з двома можливими результатами. Розрахувати можна її за наступною формулою:

$$P(Y = 1) = \frac{1}{1 + e^{(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}} \quad (4)$$

де $P(Y = 1)$ – ймовірність події;

e – число Ейлера;

X_1, X_2, \dots, X_n – фактори, які впливають на тренування;

$\beta_0, \beta_1, \dots, \beta_n$ – коефіцієнти регресії.

1.3.3 Метод опорних векторів

Метод опорних методів визначає оптимальну гіперплощину, яка найкращим чином розділяє дані у просторі. Використовується для класифікації та регресії [8].

$$y(x) = w^T x + b \quad (5)$$

де w – це вектор ваг;

x – вхідні дані;

b – зсув;

$y(x)$ – прогнозоване значення.

1.3.4 Метод нечіткої логіки

Метод нечіткої логіки надає можливість врахувати невизначеність та нечіткість в даних, що в свою чергу робить цей метод корисним в аналізі спортивних результатів, де визначення чітких меж може бути складним завданням [9].

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (6)$$

де $\mu_{A \cap B}$ – ступінь незалежності елемента x до об'єднаної нечіткої множини $A \cap B$;

$\mu_A(x), \mu_B(x)$ – це ступені незалежності елемента до множини A та B .

1.3.5 Нейронні мережі

Нейронні мережі – це комплексні алгоритми, які спроектовані для подальшої імітації процесу розпізнавання образів, які зазвичай виконує людина. Вони аналізують та оброблюють числові шаблони, які отримуються з даних, які потім перетворює у вектори, а здатність нейронних мереж кластеризувати та класифікувати отримані дані, надає їм можливість групувати інформацію на основі подібності або проводити класифікацію [10].

Глибинні нейронні мережі та штучні нейронні мережі використовуються під час розробки систем, які можуть передбачати результати спортивних подій, в тому числі і з академічного веслування.

Розглядаючи методи машинного навчання, слід відзначити два основних напрямки, а саме навчання з вчителем та навчання без вчителя.

Навчання з вчителем (англ. Supervised learning) – SL це один із способів машинного навчання, який передбачає введення вхідних та вихідних змінних для подальшого визначення математичної функції, яка зображує залежність між вхідними та вихідними даними. Основна мета – створити найточнішу модель функції, яка здатна передбачати значення вихідних змінних для нових вхідних даних. Навчання з вчителем полягає у прогнозуванні цільових змінних на основі вхідних. Процес навчання триває до досягнення моделлю певного рівня точності у передбаченні. Серед відомих прикладів навчання з вчителем варто виділити такі методи як лінійна регресія, дерева рішень, випадковий ліс, метод найближчих сусідів та логістична регресія [11].



Рисунок 1.1 – Схема навчання зі вчителем та без

Навчання без вчителя відрізняється тим, що воно не вимагає наявності цільової змінної для формування моделі; шаблони створюються виключно на основі вхідних даних. Сутність цього методу полягає в кластеризації точок даних в кластери на основі їхньої схожості. Обмеження цього алгоритму полягає в тому, що цей метод не може дати точних прогнозів, але до його переваг можна віднести те, що він самостійно може виділяти конкретні типи вихідних даних [12].

1.4 Аналіз аналогічних систем

Беручи до уваги специфічність теми кваліфікаційної роботи та відсутність аналогічних систем саме до аналізу даних з академічного веслування, то в якості аналогічних систем до розроблюваної системи програмних застосунків проаналізовано програмні рішення для виконання аналізу результатів, але в інших видах спорту. На цей момент існує маленька кількість систем для аналізу та прогнозування спортивних результатів на основі машинного навчання, які розроблені для різних видів спорту, завдань, даних та методів, проте найбільш розповсюдженими є Soccermetrics та IBM Watson for Wimbledon.

1.4.1 Soccermetrics

Soccermetrics – це вебзастосунок, який використовує алгоритми машинного навчання для аналізу та прогнозування футбольних результатів. Система дозволяє прогнозувати результати матчів, використовуючи моделі машинного навчання, такі як логістична регресія та нейронна мережа.

Переваги Soccermetrics [13]:

- система надає доступ до великої бази даних з результатами матчів;
- можливість прогнозувати результати матчів
- використовує різні моделі машинного навчання;
- виводить дані у вигляді, числових оцінок, ймовірностей та графіків.

Недоліки Soccermetrics [13]:

- система працює лише з футболістом;
- система іноді не працює.

1.4.2 IBM Watson for Wimbledon

IBM Watson for Wimbledon – це система, яка використовує машинне навчання для аналізу та прогнозування результатів з тенісу. Система дозволяє прогнозувати результати, використовуючи моделі машинного навчання, такі як логістична регресія та нейронна мережа.

Переваги IBM Watson for Wimbledon [14]:

- система надає доступ до великої бази даних з результатами матчів;
- можливість прогнозувати результати;
- використовує різні моделі машинного навчання.

Недоліки IBM Watson for Wimbledon [14]:

- система працює лише з тенісом;
- система іноді не працює.

1.5 Постановка задачі

1.5.1 Опис основних функцій системи

Система для аналізу та прогнозування спортивних результатів на основі машинного навчання, яка розробляється в ході кваліфікаційної роботи магістра, має наступний набір функціональних можливостей:

- можливість імпорту даних з файлу;
- виведення отриманих даних, для кращого спостереження за змінами та результатами обробки;
- додавання нових даних, що в свою чергу дозволяє отримувати нові дані для аналізу;
- можливість видалення застарілих, або непотрібних даних;
- очистка даних від сміття та пропусків чи неправильних значень;
- аналіз результатів;
- прогнозування результатів;
- візуалізація результатів прогнозу;
- представлення результатів обробки у вигляді діаграм.

Ці функціональні можливості спрямовані на створення високоефективної та користувацько-орієнтованої системи, яка допоможе в аналізі та прогнозуванні спортивних результатів за допомогою машинного навчання.

1.6 Специфікації вимог до програмного забезпечення

ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

Призначення системи (застосунку), для якої розробляється програмне забезпечення

Застосунок призначений для аналізу та прогнозування спортивних результатів з академічного веслування за допомогою машинного навчання.

Погодження, що ухвалені в програмній документації

Для створення застосунку та його злагодженої роботи було погоджено рішення використати допоміжні бібліотеки – Tkinter, Scikit-learn, Pandas, NumPy, Matplotlib/Seaborn та інтерактивне середовище для виконання та візуалізації коду Jupyter Notebooks.

Межі проєкту ПЗ

Дедлайн роботи над застосунком – 02.02.2024 р.

ЗАГАЛЬНИЙ ОПИС

Сфера застосування

Застосунок призначений для використання в університетських програмах та спортивних навчальних закладах для підвищення ефективності тренувань та аналізу результатів у галузі академічного веслування.

Характеристики користувачів

Основні характеристики користувачів: мати при собі смартфон, планшет або персональний комп'ютер з підключенням до мережі інтернет.

Загальна структура системи

Програмне забезпечення включає в себе модулі для збору даних, аналізу прогнозування результатів, а також інтерфейс користувача для взаємодії з системою.

Загальні обмеження

Система обмежена використанням лише для аналізу академічного веслування. Також вона залежить від якості та обсягу вхідних даних.

ФУНКЦІЇ СИСТЕМИ ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ СПОРТИВНИХ РЕЗУЛЬТАТІВ НА ОСНОВІ МАШИННОГО НАВЧАННЯ

Функція імпорту даних з датасету

Опис функції

Система повинна надавати можливість імпортувати дані тренувань з файлу CSV.

Вхідна та вихідна інформація

Вхідна інформація – файл CSV із структурованими даними тренувань;

Вихідна інформація – оновлені дані системи.

Функціональні вимоги

Доступ до файлу, перевірка та імпорт даних в систему.

Функція візуалізація графіків

Опис функції

Система повинна забезпечувати інтерактивну візуалізацію графіків для аналізу тренувань та прогнозованих даних.

Вхідна та вихідна інформація

Вхідна інформація – результати тренувань, прогнозовані дані;

Вихідна інформація – графіки.

Функціональні вимоги

Відображення графіків для різних параметрів тренувань.

Функція редагування даних

Опис функції

Застосунок повинен дозволяти редагувати дані тренувань.

Вхідна та вихідна інформація

Вхідна інформація – дані тренувань;

Вихідна інформація – відредаговані дані.

Функціональні вимоги

Доступ до даних.

Функція видалення даних тренувань

Опис функції

Застосунок повинен надавати функціонал з видалення непотрібних даних.

Вхідна та вихідна інформація

Вхідна інформація – непотрібні дані;

Вихідна інформація – видалення даних.

Функціональні вимоги

Доступ до даних.

Функція аналізу результатів тренувань

Опис функції

Система повинна аналізувати дані з результатами тренувань спортсменів.

Вхідна та вихідна інформація

Вхідна інформація – дані про тривалість, інтенсивність, антропометричні показники і тд.;

Вихідна інформація – проаналізовані дані.

Функціональні вимоги

Здійснення аналізу результатів тренувань та генерація звіту для тренера

Функція прогнозування результатів тренувань

Опис функції

Застосунок повинен використовувати моделі машинного навчання для прогнозування майбутніх результатів тренувань.

Вхідна та вихідна інформація

Вхідна інформація – дані тренувань;

Вихідна інформація – прогнозовані результати майбутніх тренувань.

Функціональні вимоги

Застосування моделей машинного навчання для прогнозування результатів. Виведення прогнозованих результатів.

ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

Джерела і зміст вхідної інформації (даних)

В даному застосунку джерелом вхідної інформації є користувач.

Нормативно-довідкова інформація (класифікатори, довідники тощо)

Вимоги до даного пункту відсутні.

Вимоги до способів організації, збереження та ведення інформації

В якості бази даних для програмного забезпечення обрано файл CSV та бібліотеку Pandas.

ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

Комп'ютер чи ноутбук може бути з будь-якими характеристиками, але не

нижче наступних:

- ОС: Windows 11
- CPU: Intel Core i3-6006u 2.00GHz;
- RAM: 16GB;
- SSD: 256GB;

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмної системи

Вимоги до даного пункту відсутні.

Мережне програмне забезпечення

Для створення застосунку потрібно встановити наступні компоненти:

- ОС Windows 11;
- Редактор коду: Visual Studio Code;
- Пакетний менеджер pip;

Програмне забезпечення ведення інформаційної бази

Робота з набором даних буде виконуватись за допомогою Pandas.

Мова і технологія розробки ПЗ

Застосунок повинен бути розроблений за допомогою бібліотек Tkinter, Pandas, Matplotlib, Sklearn. Мови розробки Python.

ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ

Інтерфейс користувача

Інтерфейс застосунку повинен задовольняти усі вимоги UX/UI, що дуже позитивно вплине на розуміння роботи застосунку.

Адаптивний інтерфейс

Адаптивним інтерфейсом є девайс, яким користувач користується (персональний комп'ютер, ноутбук).

Програмний інтерфейс

Tkinter.

ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Доступність

Кожен користувач, який бажає скористатися послугами застосунку може користуватися ним безкоштовно.

Супроводження

Застосунок повинен постійно вдосконалюватися та супроводжуватися.

Переносимість

Програмне забезпечення повинне працювати на комп'ютері з операційною системою Windows 11.

Продуктивність

Швидкість є найважливішим фактором при використанні даного програмного забезпечення. Продуктивність застосунку вимірюється у кількості запитів в секунду. Відправлення запиту не повинне перевищувати 2-х секунд.

Надійність

Усі дані користувачів повинні залишатися приватними і не повинні потрапити до сторонніх користувачів. Кожен користувач обов'язково повинен отримати доступ до своїх даних лише після авторизації у системі.

Безпека

Дані користувача при реєстрації та авторизації повинні буди зашифровані та захищені.

Висновки до розділу 1

В першому розділі було проведено аналіз предметної області, де розглянуто сутність та класифікацію систем для аналізу та прогнозування спортивних результатів. Окреслено інтелектуальні підходи, такі як кластерний аналіз, алгоритми регресії, метод опорних векторів, метод нечіткої логіки та нейронні мережі. Поставлена задача була детально розглянута, включаючи опис основних функцій системи. Також проведено аналіз аналогічних систем, таких як Soccermetrics та IBM Watson for Wimbledon. Завершено розділ специфікаціями вимог до програмного забезпечення, що встановлює основні вимоги до розробки системи для аналізу та прогнозування спортивних результатів.

2 АНАЛІЗ НАБОРУ ДАНИХ ТА ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ

2.1 Опис набору даних

Дані тренувань було виведено з монітору RM5, який було підключено до тренажеру Concept2. Після очистки даних від зайвого (тестування проводилось у певні дати протягом 3 днів, а монітор зберігав дані протягом одного року) було отримано єдиний файл у форматі *xlsx*. Під час дослідження, в якому взяли участь 195 студентів ЧНУ імені Петра Могили було отримано набір даних з результатами тренувань, вміст якого продемонстровано в табл. 2.1.

Таблиця 2.1 – Набір даних

Стовпчик	Опис
Id	Унікальний ідентифікатор студента, який було отримано з протоколу тестування.
Name	Стать студента.
Nationality	Національність.
Age	Вік студента.
Height	Зріст, який було виміряно перед тестуванням.
Weight	Вага, яку було виміряно перед тестуванням.
HeartRateBefore	Показник пульсу перед виконанням тесту.
HeartRateAfter	Показник пульсу після виконання тесту.
Date	Дата проходження тесту студентом.
ConceptNumber	Номер веслувального тренажеру.
Meters	Загальна відстань пройденої дистанції в метрах.
Avg SPM	Середня кількість помахів весла.
/500m	Час на проходження 500-ста метрів.
Cal/hr	Потужність помаху весла в калоріях.
Watt	Потужність помаху весла у ваттах.
Meter_2min	Відстань у метрах, яку студент проходить за перші дві хвилини.

Продовження таблиці 2.1

Avg SPM_2min	Середня кількість помахів весла за перші дві хвилини.
/500m_2min	Відстань у метрах, які студент пройшов за перші дві хвилини виконання тесту.
Avg SPM_2min	Середня кількість помахів весла за перші дві хвилини виконання тесту.
/500m_2min	Час на проходження 500-ста метрів за відстанню, пройденою у перші дві хвилини.
Cal/hr_2min	Потужність помаху весла в калоріях у годину за відстанню, яку було пройдено за перші дві хвилини.
Watt_2min	В ваттах потужність помаху весла за відстанню, яку було пройдено за перші дві хвилини.
Meters_4min	Відстань у метрах, яку студент проходить за третю та четверту хвилини.
Avg SPM_4min	Середня кількість помахів весла за третю та четверту хвилини.
/500m_4min	Відстань у метрах, які студент пройшов за третю та четверту хвилини виконання тесту.
Avg SPM_4min	Середня кількість помахів весла за третю та четверту хвилини виконання тесту.
/500m_4min	Час на проходження 500-ста метрів за відстанню, пройденою за третю та четверту хвилини.
Cal/hr_4min	Потужність помаху весла в калоріях у годину за відстанню, яку було пройдено за третю та четверту хвилини.
Watt_4min	В ваттах потужність помаху весла за відстанню, яку було пройдено за третю та четверту хвилини.
Meters_6min	Відстань у метрах, яку студент проходить за п'яту та шосту хвилини.

Кінець таблиці 2.1

Avg SPM_6min	Середня кількість помахів весла за п'яту та шосту хвилини виконання тесту.
/500m_6min	Час на проходження 500-ста метрів за відстанню, пройденою за п'яту та шосту хвилини.
Cal/hr_6min	Потужність помаху весла в калоріях у годину за відстанню, яку було пройдено за п'яту та шосту хвилини.
Watt_6min	В ваттах потужність помаху весла за відстанню, яку було пройдено за п'яту та шосту хвилини.

В подальшій роботі буде використано ці дані для аналізу та прогнозування спортивних результатів з академічного веслування.

2.2 Функціональна модель системи

Для моделювання структури та функціоналу кожного процесу використовуються методи, які ґрунтуються на основі об'єктно орієнтованих та структурних підходах до моделювання. Основним методом моделювання цих процесів є функціональне моделювання.

SADT (IDEF0) – це класичний метод процесного підходу до керування, який в свою чергу працює за принципом структуризації діяльності процесу відповідно до його внутрішніх процесів [15].

Структурно-функціональне моделювання поєднує в собі правила та процедури, які розкладають основний процес на дрібні компоненти, дозволяючи графічно відобразити алгоритм керування та ідентифікувати проблемні аспекти. За допомогою IDEF0-моделі є можливість отримати специфікацію всіх дій, що відбуваються в процесі, і визначити їх взаємозв'язок. Для подальшого створення моделей найкращим варіантом є застосунок Erwin Process Modeler [15].

Для опису функціональної моделі системи використано IDEF0 діаграму. Для А-0 визначено наступні правила, які керують процесом:

– нормативні акти, що регулюють використання та обробку даних спортивних результатів;

– особливості системи керування набором даних для зберігання та обробки інформації.

В якості вхідних даних використовуються:

– дані про спортивні події та результати;

– запити користувачів щодо аналізу конкретних спортивних подій або прогнозування результатів.

В якості механізмів керування виступають:

– алгоритми машинного навчання для аналізу та прогнозування спортивних результатів.

В якості кінцевих результатів виступають:

– прогнози результатів спортивних подій;

– аналітичні звіти.

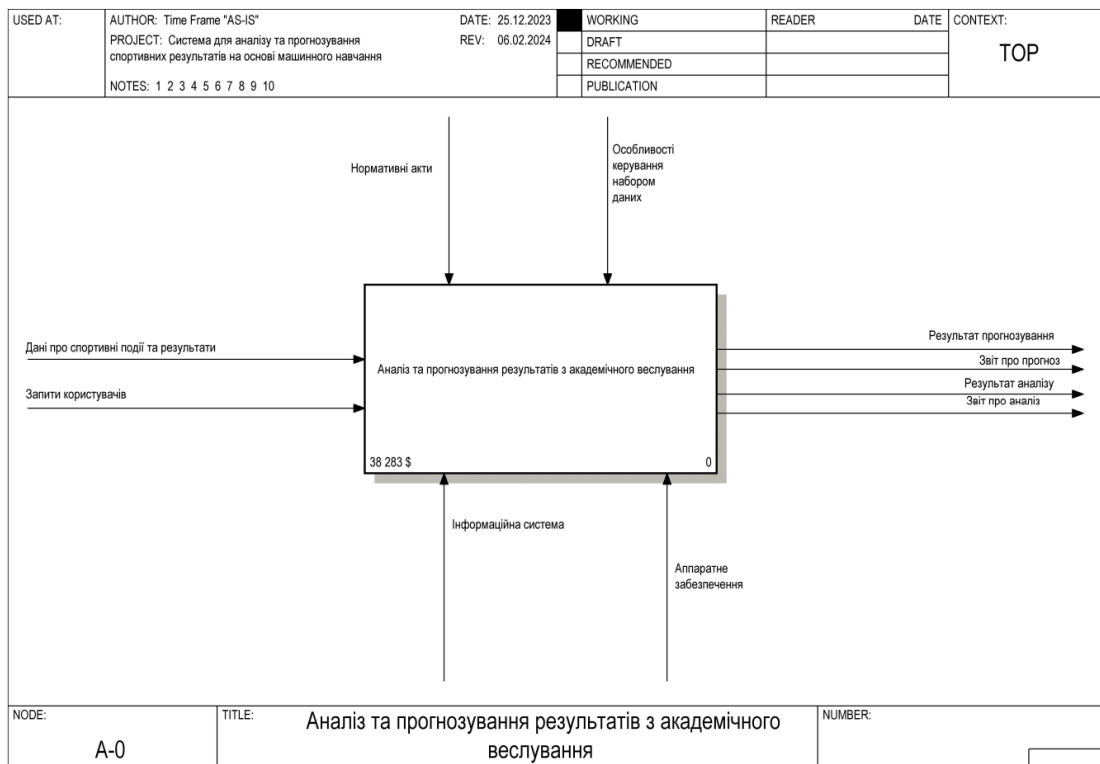


Рисунок 2.1 – Контекстна діаграма аналізу та прогнозування спортивних результатів з академічного веслування

Користувачі системи виконують запит до системи щодо аналізу певних спортивних подій, або прогнозування результатів. Виконавши аналіз та прогнозування система виводить результат у вигляді звіту з прогнозом результатів, або аналізом.

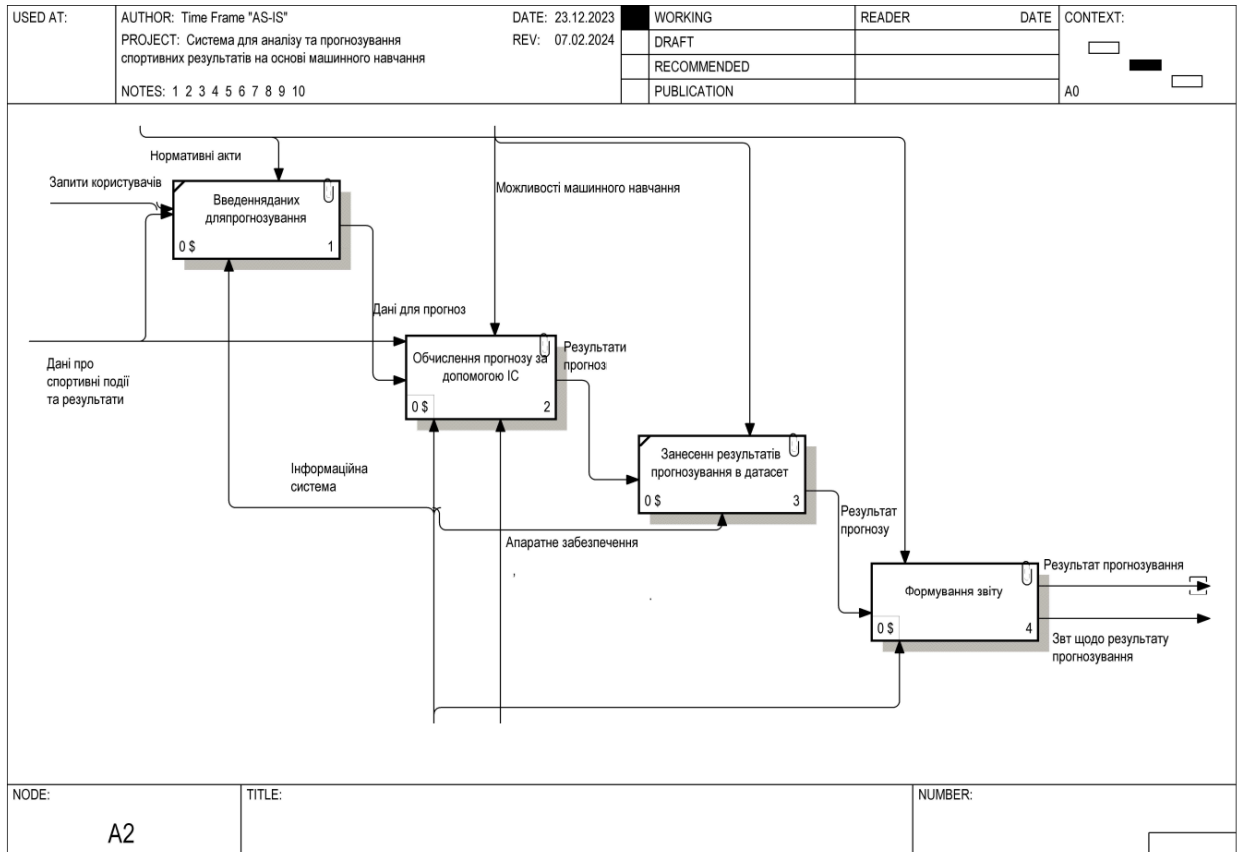


Рисунок 2.2 – Діаграма декомпозиції

Контекстна діаграма розкладається на більш дрібні компоненти для того, щоб детально показати внутрішні процеси, які відбуваються в системі. Всі зв'язки та залежності, які були визначені в батьківській діаграмі, передаються на діаграму декомпозиції.

Батьківська діаграма декомпонується на наступні блоки:

- введення даних для подальшого прогнозування;
- обробка прогнозування ІС;
- занесення результатів прогнозування в набір даних;
- формування звіту з результатами прогнозування.

Кожен з цих блоків виконує відповідні функції та процеси, що необхідні для успішного прогнозування спортивних результатів. Деталізація дозволяє краще розібратися у внутрішніх механізмах та взаємозв'язках між різними частинами системи.

Для кращого розуміння процесів що відбуваються в системі декомповано діяльність «Розрахунку прогнозу інформаційною системою». Сама діаграма декомпозиції зображена на рис. 2.3.

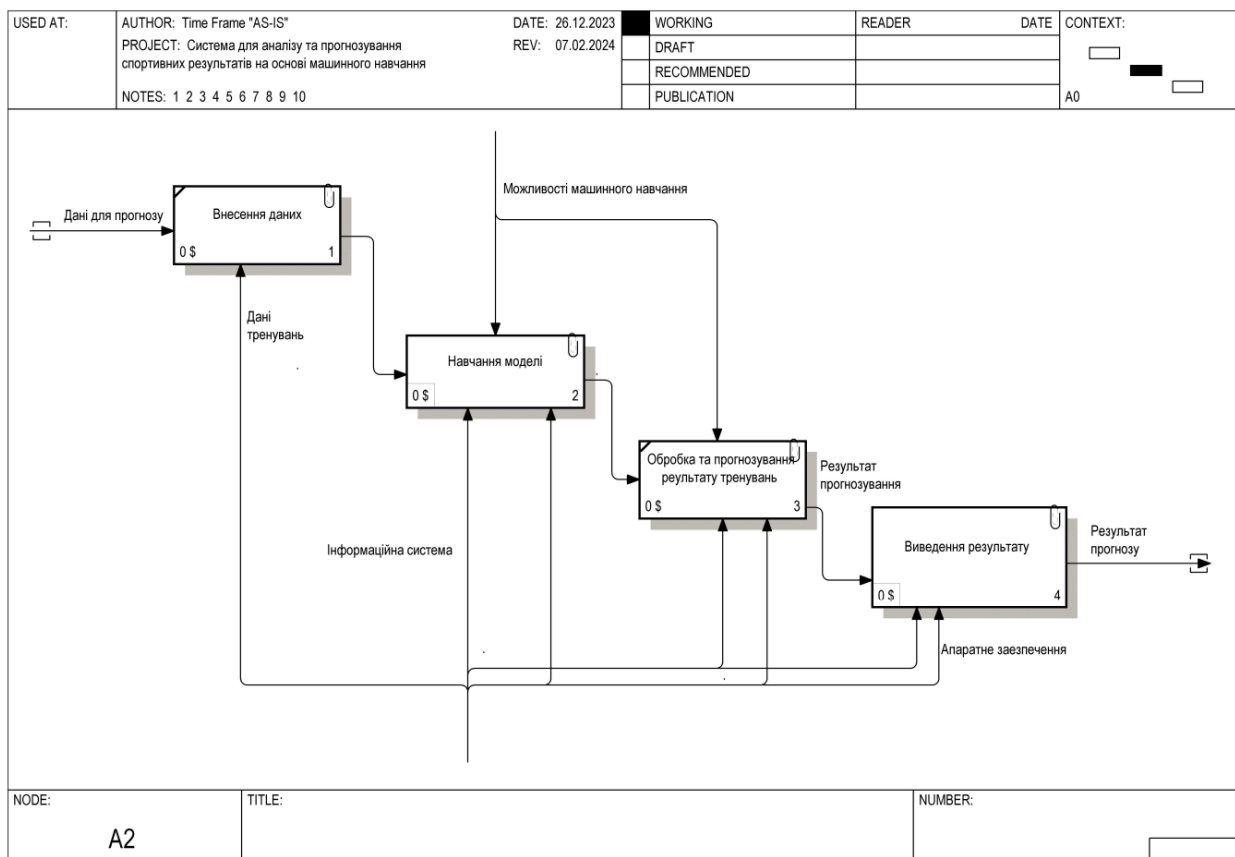


Рисунок 2.3 – Діаграма декомпозиції процесу «обробки прогнозу інформаційною системою»

Виконано аналіз основних процесів, що проходять в системі та продемонстровано їх на діаграмах.

Висновок до розділу 2

В другому розділі було описано набір даних, що використовується для моделювання та прогнозування спортивних результатів з академічного веслування. Дані були отримані з монітору PM5, підключеного до тренажеру Concept2, та очищені від зайвої інформації. Набір даних включає інформацію про студентів, таку як ідентифікатор, стать, національність, вік, зріст, вагу, показники пульсу перед і після тренування, дату, номер веслувального тренажеру, відстань, кількість помахів весла, час та потужність помаху.

Для подальшого аналізу та прогнозування спортивних результатів було побудовано функціональну модель системи. Використано метод функціонального моделювання на основі IDEF0, що дозволило детально відобразити процеси системи та їх взаємозв'язки. Кожен блок функціональної моделі виконує певні завдання, необхідні для прогнозування результатів.

Далі було розроблено контекстну діаграму та діаграму декомпозиції, що деталізує внутрішні процеси системи та показує їх структуру. Ці діаграми допомагають краще зрозуміти взаємозв'язки та логіку роботи системи для ефективного прогнозування спортивних результатів.

3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Опис використаних технологій

У зв'язку з великою кількістю вхідних та вихідних даних, розробка системи повинна бути спрямована на оптимізацію швидкості обробки даних. Це означає, що під час написання програмного забезпечення необхідно використовувати ефективні алгоритми та оптимізовані структури даних, щоб забезпечити швидку обробку інформації в реальному часі. Крім того, важливо враховувати можливість паралельної обробки даних та використання розподіленого обчислення для подальшого збільшення продуктивності системи. Такий підхід дозволить забезпечити ефективну роботу застосунку навіть при великому обсязі даних, що надходять та обробляються.

3.1.1 Python

Python – це високорівнева мова програмування загального призначення, яка була створена в 1991 році Гвідо ван Россумом. Вона характеризується простим синтаксисом та високим рівнем читабельності, що робить її ідеальним вибором для початківців та досвідчених розробників. Python підтримує багато парадигм програмування, включаючи процедурне, об'єктно-орієнтоване, функціональне та аспектно-орієнтоване програмування [16].



Рисунок 3.1 – Лого Python

Основні риси Python включають велику кількість стандартних бібліотек, які надають широкий спектр функціональності для різних завдань, таких як робота з файлами, мережами, обробка рядків, обробка даних та багато іншого.

Крім того, Python є платформонезалежною мовою, що означає, що програми, написані на Python, можуть працювати на різних операційних системах, таких як Windows, macOS та різних дистрибутивах Linux.

Python широко використовується в різних галузях, включаючи веброзробку, наукові дослідження, аналіз даних, штучний інтелект, машинне навчання, автоматизацію та багато іншого, тому саме її використання є найдоцільнішим варіантом при створенні системи для аналізу та прогнозування спортивних результатів на основі машинного навчання.

Переваги *Python* [16]:

- **читабельність коду.** Синтаксис мови програмування легкий та зрозумілий, що в свою чергу полегшує розробникам швидше писати код;
- **велика спільнота.** Python має велику спільноту, яка активно розвивається, що в свою чергу дозволяє отримати доступ до великої кількості інформації на різних ресурсах та форумах, які в майбутньому можна використовувати для вирішення виникаючих проблем;
- **велика кількість сторонніх бібліотек.** Це дозволяє пришвидшити розробку програмних застосунків.

3.1.2 Pandas

Pandas – це потужна бібліотека для обробки та аналізу даних в мові програмування Python, яка надає зручні та ефективні структури даних дозволяючи обробляти великі обсяги даних [17].



Рисунок 3.2 – Лого Pandas

Переваги *Pandas* [17]:

– *структури даних*. *Pandas* надає можливість працювати з двома основними структурами даних – *DataFrame* та *Series*. *Dataframe* представляє собою таблицю з даними, яка схожа за структурою на таблиці в базах даних. *Series* – це одновимірний масив з індексами, що надає удосконалену роботу з даними. Ці структури дозволяють легко та ефективно вирішувати завдання з обробки та аналізу даних;

– *робота з даними*. *Pandas* надає велику кількість функціоналу для фільтрації, групування, агрегації та злиття даних, що в свою чергу дозволяє розробникам легко виконувати операції з обробки даних;

– *зчитування та запис даних*. *Pandas* дозволяє виконувати зчитування та запис у різноманітних форматах, таких як CSV, Excel, SQL, JSON і тд. Це дозволяє полегшити процедуру імпорту даних з датасету та експортувати результати аналізу для подальшого використання.

3.1.3 Scikit-learn

Scikit-learn – це бібліотека машинного навчання для мови програмування Python, яка надає простий та ефективний інтерфейс для роботи з різноманітними алгоритмами машинного навчання [18].



Рисунок 3.3 – Лого Scikit-learn

Переваги *Scikit-learn* [18]:

– *простий та легкий у використанні*. Scikit-learn має простий та легкий у використанні інтерфейс, що дозволяє швидко та ефективно застосовувати різноманітні алгоритми машинного навчання навіть для новачків. Вона має добре документовану API, яка робить роботу з бібліотекою ще більш простою;

– *широкий вибір алгоритмів*. Scikit-learn включає в себе великий набір алгоритмів машинного навчання, включаючи методи класифікації, регресії, кластеризації, зменшення розмірності, відбору ознак, навчання без учителя, навчання з учителем тощо. Це дозволяє легко експериментувати з різними алгоритмами та знаходити найбільш підходящий для конкретної задачі.

3.1.4 Tkinter

Tkinter – це стандартний модуль Python, який використовується для створення графічного інтерфейсу користувача (GUI). Вона базується на Tk GUI toolkit, що походить з Tcl programming language [19].

Переваги Tkinter [19]:

– *простота використання*. Tkinter надає простий і інтуїтивно зрозумілий спосіб створення GUI в Python. Він має чистий і зрозумілий синтаксис, що робить його відмінним вибором для початківців програмістів та тих, хто хоче швидко створити простий інтерфейс;

– *кросплатформенність*. Tkinter вбудований у стандартну бібліотеку Python, що означає, що він доступний на всіх платформах, які підтримуються Python (Windows, macOS, Linux). Це забезпечує кросплатформенну сумісність ваших додатків без додаткової установки сторонніх бібліотек або інструментів;

– *обширна документація та активна спільнота*. Tkinter широко використовується і має обширну документацію, включаючи багато посібників, прикладів коду та навчальних матеріалів. Крім того, існує активна спільнота розробників, які готові допомогти і підтримати початківців у освоєнні цієї бібліотеки.

3.2 Макети інтерфейсу

Макети інтерфейсу – це візуальні моделі або концепції, що показують організацію та вигляд користувацького інтерфейсу програми, вебзастосунку або іншого програмного продукту. Вони можуть бути відображені у вигляді малюнків, креслень, прототипів або навіть інтерактивних демонстрацій [21].

Макети інтерфейсу відіграють важливу роль у процесі розробки програмного забезпечення з наступними користувачами [21]:

- візуалізація концепцій. Вони дозволяють команді розробників та дизайнерів візуалізувати концепції та ідеї щодо вигляду та функціональності продукту перед тим, як розпочати активний процес розробки;

- тестування концепцій. За допомогою макетів можна швидко та витратно виконати тестування концепцій з користувачами. Це дозволяє виявити потенційні проблеми або відмінності між очікуваннями користувачів та реалізацією продукту;

- визначення структури та організації. Макети допомагають визначити оптимальну структуру та організацію елементів інтерфейсу для забезпечення легкого користування та зручного доступу до функціональності;

- взаємодія з клієнтом. Макети дозволяють замовнику або клієнту зрозуміти, яким буде кінцевий продукт, і внести необхідні зміни чи корективи на ранніх стадіях розробки, що зменшує ризики та витрати.

Беручи до уваги наведену інформацію, то можна зробити висновок, що макети інтерфейсу є важливим інструментом у процесі розробки програмного забезпечення, який допомагає командам розробників та дизайнерів ефективно спілкуватися, тестувати та вдосконалювати концепції продукту перед його реалізацією.

Система аналізу та прогнозування результатів академічного веслування розроблена для надання користувачам зручного інструменту для обробки та аналізу даних з веслування. Головний екран застосунку включає в себе інтерфейс з кнопками для імпорту даних з файлу, збереження, видалення, аналізу та

прогнозування результатів. Також на головному екрані розміщена таблиця з даними, де користувач може переглядати та редагувати інформацію. Функціонал застосунку дозволяє ефективно керувати та аналізувати великі обсяги даних з академічного веслування, що сприяє покращенню спортивних результатів та підвищенню ефективності тренувань.

3.2.1 Головна сторінка

Головна сторінка системи для аналізу та прогнозування спортивних результатів з академічного веслування на основі машинного навчання містить наступні елементи:

– кнопка «Завантажити файл». Ця кнопка дозволяє користувачу завантажити файл з даними про спортивні результати з академічного веслування у систему для подальшого аналізу та обробки;

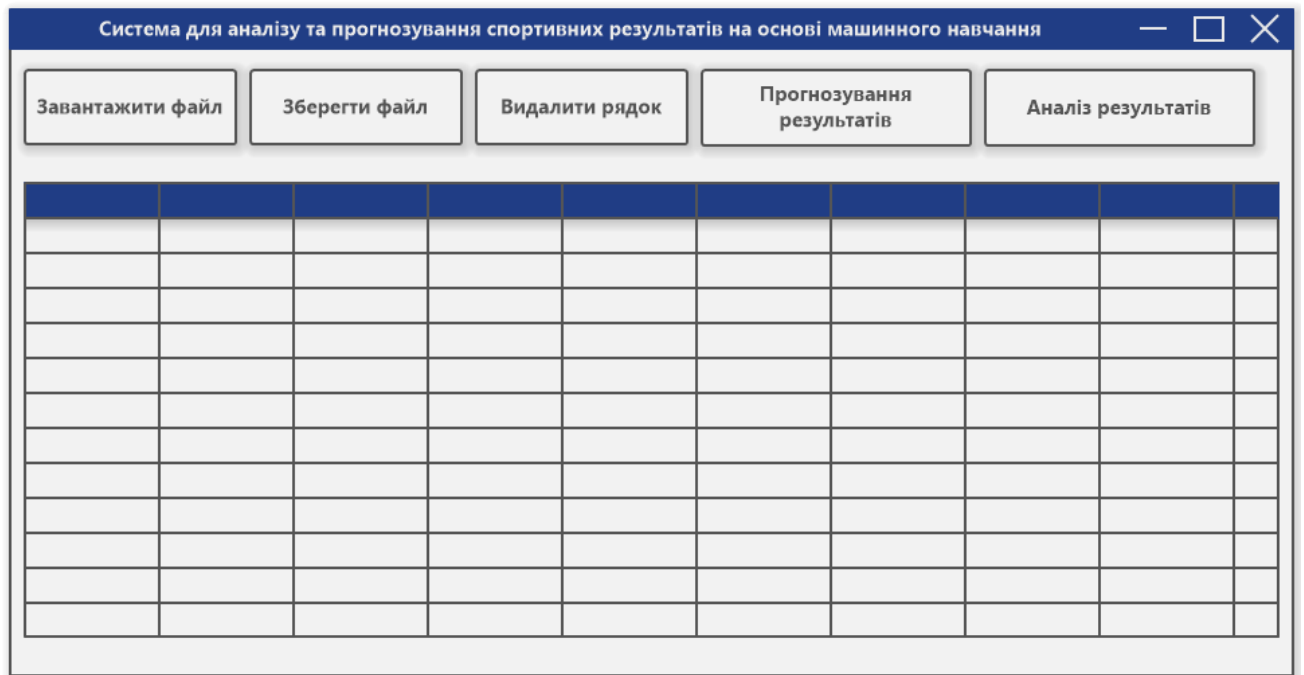


Рисунок 3.1 – Макет головного екрану

– кнопка «Зберегти файл». Після внесення змін у дані або після аналізу та прогнозування результатів користувач може скористатися цією кнопкою для збереження файлу з оновленими даними або результатами;

– кнопка «Видалити строку»: За допомогою цієї кнопки користувач може видалити вибраний рядок з таблиці даних. Це може бути корисним для виправлення помилок або очищення непотрібних записів;

– кнопка «Прогнозування результатів». Ця кнопка запускає процес прогнозування спортивних результатів на основі машинного навчання. Система аналізує наявні дані та дає прогнози щодо майбутніх результатів;

– кнопка «Аналіз результатів тренувань». Після завантаження даних користувач може скористатися цією кнопкою для аналізу результатів тренувань. Система надає звіт або візуалізації, що допомагають зрозуміти основні відмінності спортивних результатів.

3.2.2 Вікно редагування рядків таблиці

Вікно редагування рядка з таблиці в системі для аналізу та прогнозування спортивних результатів з академічного веслування на основі машинного навчання надає можливість користувачам зручно та легко вносити зміни до існуючих даних. Ось основні елементи та функціонал цієї сторінки:

– назва комірки та її вміст. В цьому вікні відображаються дані про спортивні результати з академічного веслування;

– форма редагування. Для редагування даних користувач повинен обрати потрібний рядок в таблиці та відкрити форму редагування. Ця форма містить поля, що відповідають кожному атрибуту даних, таким як ім'я, вік, зріст, вага та інші параметри, які можна змінювати;

– кнопка збереження. Після внесення змін користувач натискає кнопку «Зберегти», щоб застосувати зміни до вибраного рядка даних. Це дозволяє оновити існуючий запис із новими даними;

– кнопка скасування. Кнопка «Скасувати» дозволяє користувачеві відмінити редагування та залишити дані без змін, якщо він вирішить не зберігати внесені зміни;

Вікно для редагування даних

ID	<input type="text" value="Вміст"/>	<input type="button" value="Зберегти"/>
Date_Time_x	<input type="text" value="Вміст"/>	
Meters	<input type="text" value="Вміст"/>	
Avg SPM	<input type="text" value="Вміст"/>	
/500m	<input type="text" value="Вміст"/>	
Cal/hr	<input type="text" value="Вміст"/>	
Watt	<input type="text" value="Вміст"/>	
Number_x	<input type="text" value="Вміст"/>	
Meters_2min	<input type="text" value="Вміст"/>	
SPM_2min	<input type="text" value="Вміст"/>	
/500m_2min	<input type="text" value="Вміст"/>	
Cal/hr_2min	<input type="text" value="Вміст"/>	
Watt_2min	<input type="text" value="Вміст"/>	
Meters_4min	<input type="text" value="Вміст"/>	
SPM_4min	<input type="text" value="Вміст"/>	
...	<input type="text" value="Вміст"/>	

Рисунок 3.2 – Вікно редагування вмісту рядка таблиці

- валідація даних. Система може виконувати перевірку на коректність введених даних, так що користувач буде попереджений про будь-які помилки чи некоректні значення перед збереженням;
- зручний інтерфейс. Сторінка має зручний інтерфейс з інтуїтивно зрозумілими елементами керування, що дозволяє користувачам легко та ефективно взаємодіяти з даними для редагування;

3.3 Діаграма прецедентів та сценарії використання

Діаграми прецедентів – це важливий інструмент у аналізі та проєктуванні програмного забезпечення, особливо у методології UML (Unified Modeling Language). Вони використовуються для моделювання функціональності системи з точки зору її взаємодії з різними користувачами або зовнішніми сутностями [20].

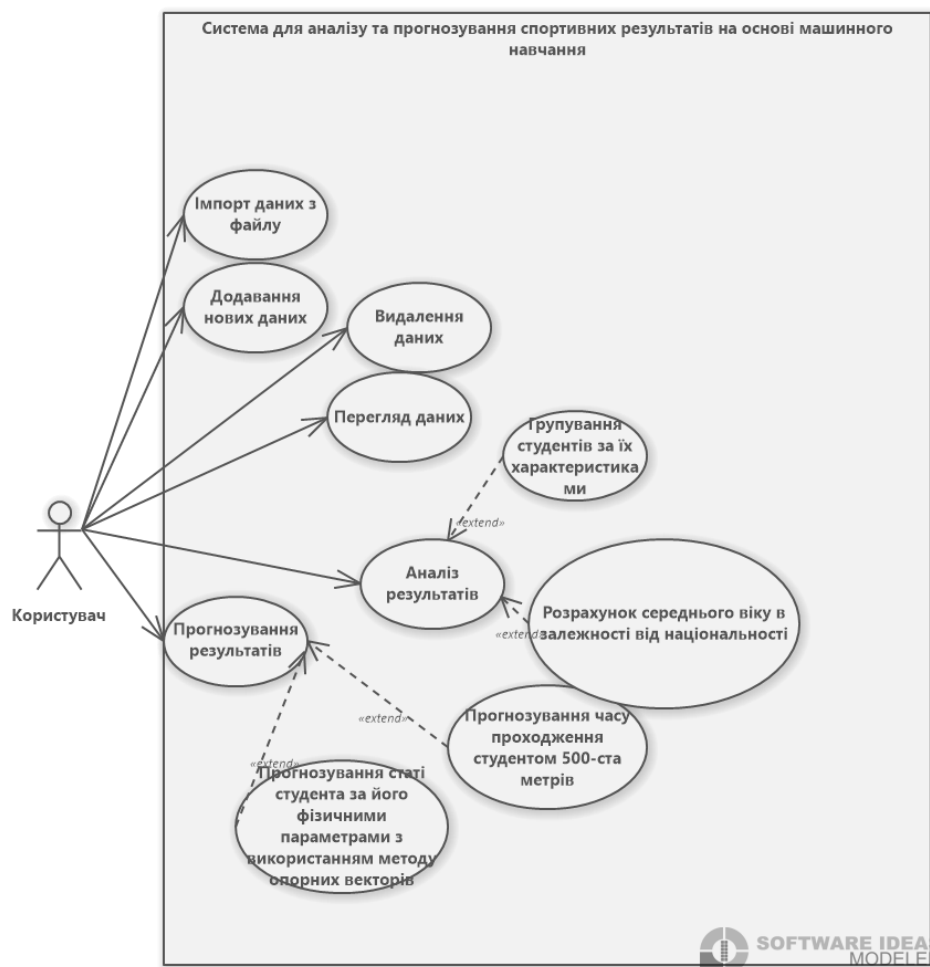


Рисунок 3.3 – Діаграма використання

Основна мета діаграм прецедентів полягає в тому, щоб [20]:

– уточнити функціональні вимоги системи. Діаграми прецедентів дозволяють ідентифікувати основні дії, які можуть виконувати користувачі або зовнішні системи щодо програмного забезпечення;

– визначити акторів та їх взаємодію із системою: актори – це ролі чи сутності, які взаємодіють із системою. Діаграма прецедентів допомагає визначити цих акторів та описати їх можливі дії в контексті системи;

– підготувати основу для детального аналізу та проєктування: діаграми прецедентів надають візуальне представлення функціональності системи, що є основою для подальшого детального аналізу та проєктування компонентів програмного забезпечення;

– комунікація з зацікавленими сторонами. Вони також можуть використовуватись для комунікації з клієнтами або зацікавленими сторонами щодо функціональних вимог до системи.

Сценарії використання є важливою частиною процесу розробки програмного забезпечення. Вони допомагають виявити можливі проблеми та недоліки у роботі застосунку ще до того, як вона потрапить до кінцевого користувача.

Таблиця 3.1 – Сценарій імпорту датасету.

Діючі особи	Користувач, система.
Мета	Імпортування файлу з набором даних тренувань.
Передумова	Користувач запустив застосунок.
Успішний сценарій:	
<ul style="list-style-type: none"> – користувач натискає кнопку «підключити набір даних»; – обирає потрібний файл; – натискає кнопку «додати»; – система імпортує дані з файлу; – користувач отримує повідомлення про успішний імпорт даних. 	

Кінець таблиці 3.1

Сценарій успішний. Користувач імпортував набір даних з датасету.	
1a	Відсутній файл. Система демонструє відповідне повідомлення. Результат: користувач не може імпортувати файл.
2a	Розширення файлу не підтримується. Система демонструє відповідне повідомлення. Результат: користувач не може імпортувати файл.
Усі розширення успішно виконані.	

Таблиця 3.2 – Виведення даних з таблиці.

Діючі особи	Користувач, система.
Мета	Виведення даних тренувань.
Передумова	Користувач виконав імпорт даних з файлу.
Успішний сценарій: <ul style="list-style-type: none"> – користувач натискає кнопку вивести дані тренувань; – користувач отримує результати тренувань. 	
Сценарій успішний. Користувач вивів результати тренувань.	
1a	Відсутній файл. Система демонструє відповідне повідомлення. Результат: користувач не може вивести результати тренувань.
Усі розширення успішно виконані.	

Таблиця 3.3 – Додавання нових даних

Діючі особи	Користувач, система.
Мета	Додавання нових даних для аналізу.

Кінець таблиці 3.3

Передумова	Користувач виконав імпорт даних з файлу або система має попередні дані для аналізу.
Успішний сценарій:	
<ul style="list-style-type: none"> - користувач натискає кнопку «додати дані»; - вводить нові дані через відповідну форму або завантажує файл з новими даними; - натискає кнопку «зберегти»; - система додає нові дані до існуючого набору даних; - користувач отримує повідомлення про успішне додавання нових даних. 	
Сценарій успішний. Користувач успішно додав нові дані для аналізу.	
1a	Відсутність нових даних. Система демонструє відповідне повідомлення. Результат: користувач не може додати порожні дані для аналізу.
2a	Помилковий формат даних. Система демонструє відповідне повідомлення про помилку в форматі. Результат: користувач не може додати дані через неправильний формат.
Усі розширення успішно виконані.	

Таблиця 3.4 – Аналіз результатів

Діючі особи	Користувач, система.
Мета	Аналіз результатів обробки даних.
Передумова	Дані оброблені та готові для аналізу.
Успішний сценарій:	
<ul style="list-style-type: none"> - користувач вибирає параметри для аналізу; - система виконує аналіз оброблених даних; - користувач отримує результати аналізу. 	

Кінець таблиці 3.4

Сценарій успішний. Користувач успішно проаналізував результати.	
1a	Система демонструє відповідне повідомлення. Результат: користувач не може аналізувати дані без попередньої обробки.
2a	Система повідомляє користувача про неправильний вибір параметрів для аналізу. Результат: користувач коригує параметри.
Усі розширення успішно виконані.	

Таблиця 3.5 – Прогнозування результатів

Діючі особи	Користувач, система.
Мета	Прогнозування майбутніх результатів на основі оброблених даних.
Передумова	Існують оброблені дані для аналізу та прогнозування.
Успішний сценарій:	
<ul style="list-style-type: none"> - користувач обирає необхідні параметри; - система виконує прогноз на основі оброблених даних; - користувач отримує прогнозовані результати. 	
Сценарій успішний. Користувач успішно отримав прогнозовані результати.	
1a	Відсутність оброблених даних. Система демонструє відповідне повідомлення. Результат: користувач не може отримати прогноз без попередньої обробки.
Усі розширення успішно виконані.	

Таблиця 3.6 – Візуалізація результатів прогнозу

Діючі особи	Користувач, система.
Мета	Представлення прогнозованих результатів у вигляді діаграм.
Передумова	Існують прогнозовані дані.
Успішний сценарій: <ul style="list-style-type: none"> – користувач обирає параметри для візуалізації; – система створює діаграму на основі прогнозованих даних; – користувач отримує діаграму з прогнозованими результатами. 	
Сценарій успішний. Користувач успішно отримав діаграму з прогнозованими результатами.	
1a	Відсутність прогнозованих даних. Система демонструє відповідне повідомлення. Результат: користувач не може побудувати діаграму без прогнозованих даних.
Усі розширення успішно виконані.	

Таблиця 3.7 – Експорт даних в файл.

Діючі особи	Користувач, система.
Мета	Експорт набор даних тренувань в файл.
Передумова	Користувач запустив застосунок.
Успішний сценарій: <ul style="list-style-type: none"> – користувач натискає кнопку «Зберегти дані»; – обирає місце куди необхідне зберегти файл; – натискає кнопку «зберегти»; – система експортує дані в файл; – користувач отримує повідомлення про успішний експорт даних. 	

Кінець таблиці 3.7

Сценарій успішний. Користувач експортував набір даних з датасету.	
1a	Відсутні дані. Система демонструє відповідне повідомлення. Результат: користувач не може експортувати дані.
2a	Розширення файлу не підтримується. Система демонструє відповідне повідомлення. Результат: користувач не може експортувати файл.
Усі розширення успішно виконані.	

Всі ці сценарії успішно виконані під час тестування, якщо вони відповідають очікуваному результату. Це допомагає забезпечити, що програма працює правильно та надійно, зменшуючи можливість помилок та ризиків для користувачів

3.4 Діаграма класів

Діаграма класів – це графічний інструмент, що належить до родини діаграм UML, і який призначений для ілюстрації структури системи з точки зору класів, їх атрибутів, методів та взаємозв'язків між ними. Це потужний інструмент, який дозволяє візуально відобразити компоненти програми і зрозуміти, як вони взаємодіють один з одним. У діаграмі класів кожен клас зображений у вигляді прямокутника, де вказуються його атрибути та методи. Зв'язки між класами позначаються стрілками, що вказують напрямок зв'язку та його тип. Цей інструмент є важливим для аналізу та проектування програмних систем [22].



Рисунок 3.4 – Діаграма класів

3.4.1 Опис класу AthleteDataMode

AthleteDataMode – забезпечує обробку та аналіз даних, включаючи побудову прогнозів та їх візуалізацію, а також за кластеризацію атлетів за характеристиками.

Атрибути:

– data – містить дані про спортивні тренування;

Операції:

– __init__() – конструктор класу, що ініціалізує об'єкт AthleteDataModel з наданими даними;

– predict_watt() – метод для побудови та візуалізації прогнозів втрати енергії;

– perform_clustering() – метод для кластеризації атлетів за характеристиками.

3.4.2 Опис класу GUI

GUI – забезпечує створення графічного інтерфейсу користувача для взаємодії з даними спортсменів, включаючи завантаження даних, відображення їх у вікні програми та надання можливості виконання операцій, таких як побудова прогнозів та кластеризація.

Атрибути:

- `button_frame` – фрейм для розміщення кнопок;
- `load_data_button` – кнопка для завантаження даних;
- `predict_watt_button` – кнопка для виклику методу `predict_watt()`;
- `cluster_button` – кнопка для виклику методу `perform_clustering()`;
- `tree` – об'єкт `Treeview` для відображення даних.

Операції:

- `__init__()` – конструктор класу, що налаштовує графічний інтерфейс користувача;
- `load_data()` – метод для завантаження даних з файлу;
- `on_double_click()` – метод, що викликається при подвійному клацанні на рядку даних для відкриття вікна редагування;
- `predict_watt()` – метод для виклику методу `predict_watt()` з класу `AthleteDataModel`;
- `perform_clustering()` – метод для виклику методу `perform_clustering()` з класу `AthleteDataModel`.

3.4.3 Опис класу EditWindow

`EditWindow` – забезпечує створення та відображення вікна для редагування даних, а також за збереження внесених змін.

Атрибути:

- `values` – значення редагованих даних;
- `columns` – назви стовпців даних;

Операції:

- `__init__()` – конструктор класу, що створює вікно редагування;
- `save()` – метод для збереження редагованих даних.

Висновок до розділу 3

В третьому розділі було розглянуто технології, необхідні для проектування програмного забезпечення системи аналізу та прогнозування спортивних результатів на основі машинного навчання. Зважаючи на обсяг вхідних та вихідних даних, головний акцент був зроблений на оптимізацію швидкості обробки даних та забезпечення ефективної роботи системи, навіть за великих обсягів інформації.

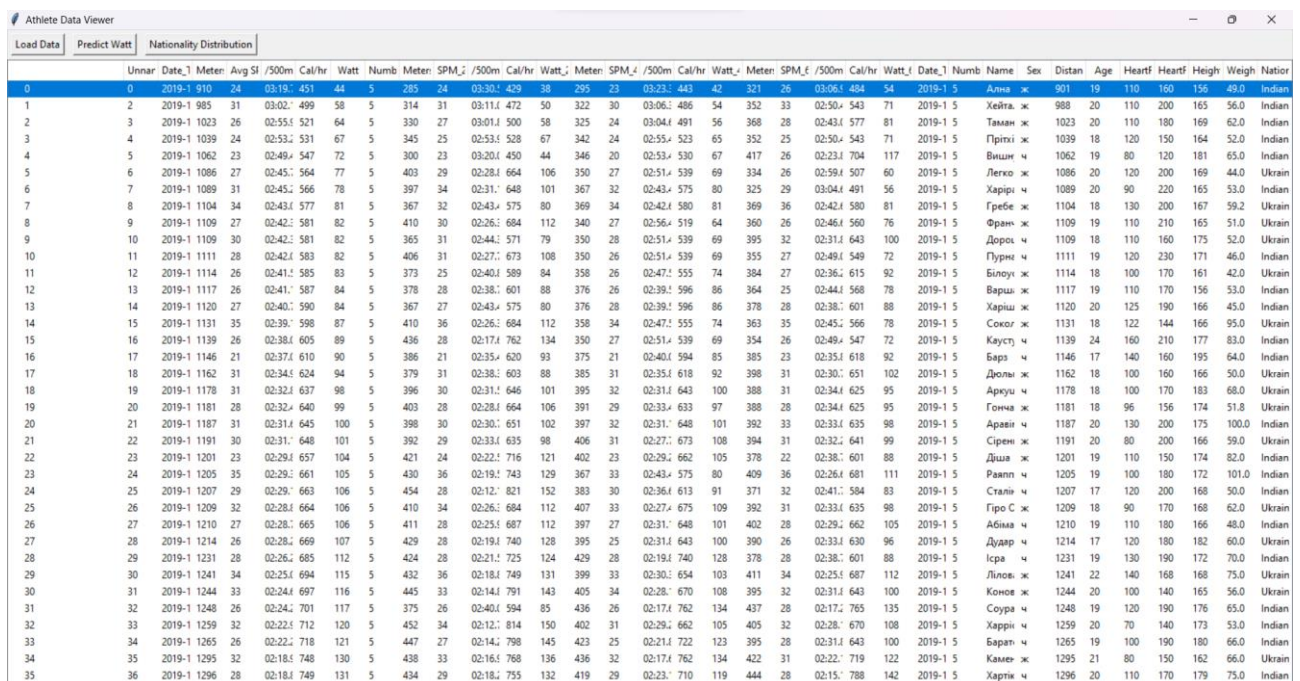
В рамках цього розділу були розглянуті та описані ключові технології, необхідні для розробки системи, зокрема Python, який вибрано як основну мову програмування з урахуванням його читабельності та великої кількості сторонніх бібліотек. Також було описано бібліотеки Pandas та Scikit-learn, які використовуються для обробки та аналізу даних, а також для реалізації алгоритмів машинного навчання відповідно. Для кращого розуміння функціоналу системи були розроблені діаграми прецедентів, що відображають взаємодію користувачів із системою, а також написані сценарії використання, які дозволяють детально описати можливі дії користувачів та взаємодію із системою.

Створено макети інтерфейсу та опис роботи функціоналу. Для кращого представлення роботи системи та її інтерфейсу були створені макети інтерфейсу, що демонструють вигляд та функціональність системи. Також було проведено детальне опис роботи функціоналу системи, що включає в себе операції з обробки та аналізу даних, а також застосування алгоритмів машинного навчання для прогнозування спортивних результатів.

4. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Імпорт, редагування, додавання та видалення даних

При натисканні кнопки «Load Data» викликається метод *load_data()*. Він відкриває діалогове вікно для вибору файлу за допомогою *filedialog.askopenfilename()*. Після вибору файлу викликається *pd.read_csv()* для завантаження даних. Якщо дані успішно завантажені, вони відображаються у дереві (Treeview). Колонки та рядки виводяться на екран у вигляді таблиці за допомогою методів *self.tree["columns"], self.tree.heading(), self.tree.column()* та *self.tree.insert()*. Приклад виведення даних з файлу можна переглянути на рис. 4.1.



Year	Date	Meter	Avg SI	500m	Cal/hr	Watt	Numb	Meter	SPM, /500m	Cal/hr	Watt, Meten	SPM, /500m	Cal/hr	Watt, Meten	SPM, /500m	Cal/hr	Watt, Date, Numb	Name	Sex	Distan	Age	Hearf	Hearf	Heigh	Weigh	Nator								
0	2019-1	910	24	03:19:	451	44	5	285	24	03:30:	429	38	295	23	03:23:	443	42	321	26	03:06:	484	54	2019-1	5	Альн	ж	901	19	110	160	156	49.0	Indian	
1	2	2019-1	985	31	03:02:	499	58	5	314	31	03:11:	472	50	322	30	03:06:	486	54	352	33	02:50:	543	71	2019-1	5	Хейла	ж	988	20	110	200	165	56.0	Indian
2	3	2019-1	1023	26	02:55:	521	64	5	330	27	03:01:	500	58	325	24	03:04:	491	56	368	28	02:43:	577	81	2019-1	5	Таман	ж	1023	20	110	180	169	62.0	Indian
3	4	2019-1	1039	24	02:53:	531	67	5	345	25	02:53:	528	67	342	24	02:55:	523	65	352	25	02:50:	543	71	2019-1	5	Прітї	ж	1039	18	120	150	164	52.0	Indian
4	5	2019-1	1062	23	02:49:	547	72	5	300	23	03:20:	450	44	346	20	02:53:	530	67	417	26	02:23:	704	117	2019-1	5	Виши	ч	1062	19	80	120	181	65.0	Indian
5	6	2019-1	1086	27	02:45:	564	77	5	403	29	02:28:	664	106	350	27	02:51:	539	69	334	26	02:59:	507	60	2019-1	5	Легю	ж	1086	20	120	200	169	44.0	Ukrain
6	7	2019-1	1089	31	02:45:	566	78	5	397	34	02:31:	648	101	367	32	02:43:	575	80	325	29	03:04:	491	56	2019-1	5	Харіч	ч	1089	20	90	220	165	53.0	Indian
7	8	2019-1	1104	34	02:43:	577	81	5	367	32	02:43:	575	80	369	34	02:42:	580	81	369	36	02:42:	580	81	2019-1	5	Гребе	ж	1104	18	130	200	167	59.2	Ukrain
8	9	2019-1	1109	27	02:42:	581	82	5	410	30	02:26:	684	112	340	27	02:56:	519	64	360	26	02:46:	560	76	2019-1	5	Фран	ж	1109	19	110	210	165	51.0	Ukrain
9	10	2019-1	1109	30	02:42:	581	82	5	365	31	02:44:	571	79	350	28	02:51:	539	69	395	32	02:31:	643	100	2019-1	5	Доро	ч	1109	18	110	160	175	52.0	Ukrain
10	11	2019-1	1111	28	02:42:	583	82	5	406	31	02:27:	673	108	350	26	02:51:	539	69	355	27	02:49:	549	72	2019-1	5	Пурн	ч	1111	19	120	230	171	46.0	Indian
11	12	2019-1	1114	26	02:41:	585	83	5	373	25	02:40:	589	84	358	26	02:47:	555	74	384	27	02:36:	615	92	2019-1	5	Білоу	ж	1114	18	100	170	161	42.0	Ukrain
12	13	2019-1	1117	26	02:41:	587	84	5	378	28	02:38:	601	88	376	26	02:39:	596	86	364	25	02:44:	568	78	2019-1	5	Варш	ж	1117	19	110	170	156	53.0	Indian
13	14	2019-1	1120	27	02:40:	590	84	5	367	27	02:43:	575	80	376	28	02:39:	596	86	378	28	02:38:	601	88	2019-1	5	Харш	ж	1120	20	125	190	166	45.0	Indian
14	15	2019-1	1131	35	02:39:	598	87	5	410	36	02:26:	684	112	358	34	02:47:	555	74	363	35	02:45:	566	78	2019-1	5	Соко	ж	1131	18	122	144	166	95.0	Ukrain
15	16	2019-1	1139	26	02:38:	605	89	5	436	28	02:17:	762	134	350	27	02:51:	539	69	354	26	02:49:	547	72	2019-1	5	Кауст	ч	1139	24	160	210	177	83.0	Indian
16	17	2019-1	1146	21	02:37:	610	90	5	386	21	02:35:	620	93	375	21	02:40:	594	85	385	23	02:35:	618	92	2019-1	5	Бар	ч	1146	17	140	160	195	64.0	Indian
17	18	2019-1	1152	31	02:38:	624	94	5	379	31	02:38:	603	88	385	31	02:35:	618	92	398	31	02:30:	651	102	2019-1	5	Доло	ж	1152	18	100	160	166	50.0	Ukrain
18	19	2019-1	1178	31	02:32:	637	98	5	396	30	02:31:	646	101	395	32	02:33:	633	100	388	31	02:34:	625	95	2019-1	5	Арку	ч	1178	18	100	170	183	60.0	Ukrain
19	20	2019-1	1181	28	02:32:	640	99	5	403	28	02:28:	664	106	391	29	02:33:	643	97	388	28	02:34:	625	95	2019-1	5	Гонч	ж	1181	18	96	156	174	51.8	Ukrain
20	21	2019-1	1187	31	02:31:	645	100	5	398	30	02:30:	651	102	397	32	02:31:	648	101	392	33	02:33:	635	98	2019-1	5	Аран	ч	1187	20	130	200	175	100.0	Indian
21	22	2019-1	1191	30	02:31:	648	101	5	392	29	02:33:	635	98	406	31	02:27:	673	108	394	31	02:32:	641	99	2019-1	5	Сирн	ж	1191	20	80	200	166	59.0	Ukrain
22	23	2019-1	1201	23	02:29:	657	104	5	421	24	02:22:	716	121	402	23	02:29:	662	105	378	22	02:38:	601	88	2019-1	5	Діш	ж	1201	19	110	150	174	82.0	Indian
23	24	2019-1	1205	35	02:29:	661	105	5	430	36	02:19:	743	129	367	33	02:49:	575	80	409	36	02:26:	681	111	2019-1	5	Рвал	ч	1205	19	100	180	172	101.0	Indian
24	25	2019-1	1207	29	02:29:	663	106	5	454	28	02:12:	821	152	383	30	02:36:	613	91	371	32	02:41:	584	83	2019-1	5	Стал	ч	1207	17	120	200	168	50.0	Indian
25	26	2019-1	1209	32	02:28:	664	106	5	410	34	02:26:	684	112	407	33	02:27:	675	109	392	31	02:33:	635	98	2019-1	5	Гро	С	1209	18	90	170	168	62.0	Ukrain
26	27	2019-1	1210	27	02:28:	665	106	5	411	28	02:25:	687	112	397	27	02:31:	648	101	402	28	02:29:	662	105	2019-1	5	Абіа	ч	1210	19	110	180	166	48.0	Indian
27	28	2019-1	1214	26	02:28:	669	107	5	429	28	02:19:	740	128	395	25	02:31:	643	100	390	26	02:33:	630	96	2019-1	5	Дуд	ч	1214	17	120	180	182	60.0	Ukrain
28	29	2019-1	1231	28	02:26:	685	112	5	424	28	02:21:	725	124	429	28	02:19:	740	128	378	28	02:38:	601	88	2019-1	5	Іср	ч	1231	19	130	190	172	70.0	Indian
29	30	2019-1	1241	34	02:25:	694	115	5	432	36	02:18:	749	131	399	33	02:30:	654	103	411	34	02:25:	687	112	2019-1	5	Ліло	ж	1241	22	140	168	168	75.0	Ukrain
30	31	2019-1	1244	33	02:24:	697	116	5	445	33	02:14:	791	143	405	34	02:28:	670	108	395	32	02:31:	643	100	2019-1	5	Коне	ж	1244	20	100	140	165	56.0	Ukrain
31	32	2019-1	1248	26	02:24:	701	117	5	375	26	02:40:	594	85	436	26	02:17:	762	134	437	28	02:17:	765	135	2019-1	5	Соур	ч	1248	19	120	190	176	65.0	Indian
32	33	2019-1	1259	32	02:22:	712	120	5	452	34	02:12:	814	150	402	31	02:29:	662	105	405	32	02:28:	670	108	2019-1	5	Харр	ч	1259	20	70	140	173	53.0	Indian
33	34	2019-1	1265	26	02:22:	718	121	5	447	27	02:14:	798	145	423	25	02:21:	722	123	395	28	02:31:	643	100	2019-1	5	Бар	ч	1265	19	100	190	180	66.0	Indian
34	35	2019-1	1295	32	02:18:	748	130	5	438	33	02:16:	768	136	436	32	02:17:	762	134	422	31	02:22:	719	122	2019-1	5	Кале	ж	1295	21	80	150	162	66.0	Ukrain
35	36	2019-1	1296	28	02:18:	749	131	5	434	29	02:18:	755	132	419	29	02:23:	710	119	444	28	02:15:	788	142	2019-1	5	Харт	ч	1296	20	110	170	179	73.0	Indian

Рисунок 4.1 – Виведення імпортованих даних з файлу

При подвійному клацанні на рядок таблиці викликається метод *on_double_click()*. У цьому методі створюється вікно редагування «EditWindow», яке містить усі значення рядка та поля для їх зміни. Користувач може запровадити нові значення біля поля редагування. Після натискання кнопки *Save* нові значення зберігаються в *DataFrame self.master.data*, а відображена

інформація в таблиці оновлюється, що і можна переглянути на рис. 4.2.

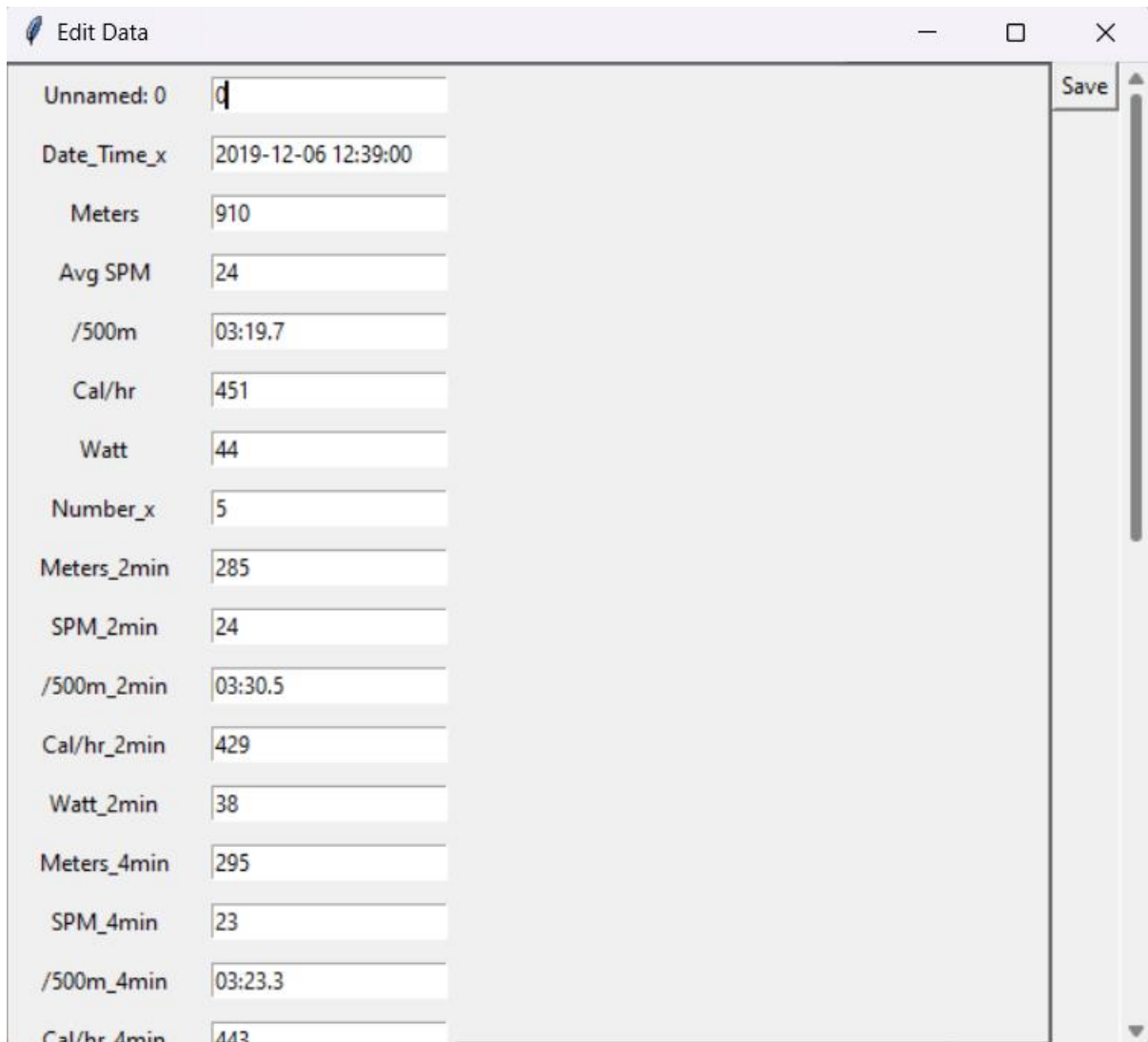


Рисунок 4.2 – Вікно редагування даних в рядку таблиці

При натисканні кнопки «Save As» викликається метод *save_as()*. Цей метод відкриває діалогове вікно для вибору розташування та назви файлу для збереження за допомогою *filedialog.asksaveasfilename()*. Після вибору шляху до файлу дані зберігаються у вказаних файлах за допомогою *self.data.to_csv()*.

При натисканні кнопки «Delete Row» викликається метод *delete_row()*. Користувачу спочатку показується діалогове вікно підтвердження видалення рядка. Якщо видалення підтверджено, вибраний рядок у таблиці видаляється за допомогою *self.tree.delete()*.

4.1 Аналіз результатів

4.1.1 Групування студентів за їх характеристиками

Дані про студентів завантажуються з файлу, після чого вибираються характеристики для аналізу, такі як вік, розмір, вага, пульс тощо. Ці характеристики нормалізуються для однорідності масштабу. Визначається кількість кластерів, у які будуть групуватися спортсмени (у даному випадку – 3).

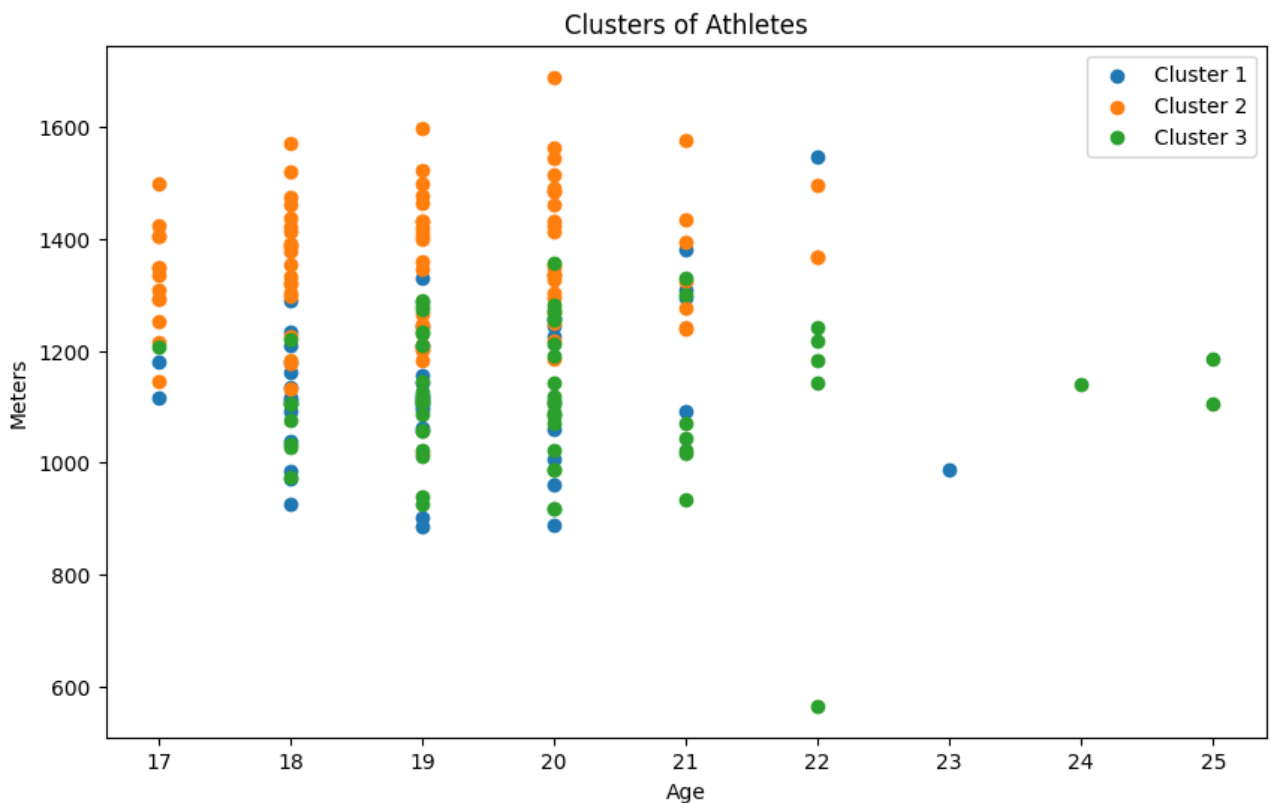


Рисунок 4.3 – Групування студентів за їх характеристиками

```

super()._check_params_vs_input(X, default_n_init=10)
Cluster 1:
      Unnamed: 0      Meters      Avg SPM      Cal/hr      Watt      Number_x \
count  91.000000   91.000000   91.000000   91.000000   91.000000   91.000000
mean  105.131868  1355.560440  28.626374  826.758242  153.219780  6.241758
std   58.344229   124.276145   3.230577  142.794751  41.527448  1.250250
min   15.000000   1016.000000  21.000000  516.000000  63.000000  5.000000
25%   49.500000   1267.500000  26.500000  720.500000  122.000000  5.000000
50%   110.000000  1354.000000  28.000000  812.000000  149.000000  6.000000
75%   165.000000  1432.500000  31.000000  906.500000  176.500000  8.000000
max   195.000000  1690.000000  38.000000  1296.000000  290.000000  8.000000

      Meters_2min  SPM_2min  Cal/hr_2min  Watt_2min  ...  Cal/hr_6min \
count  91.000000   91.000000   91.000000   91.000000  ...   91.000000
mean  471.395604   29.516484   905.472527  176.076923  ...   794.571429
std   52.880763    3.563278   199.708974   58.004833  ...   135.656522
min   355.000000   21.000000   549.000000   72.000000  ...   496.000000
25%   432.500000   27.000000   750.500000  131.500000  ...   699.500000
50%   477.000000   29.000000   905.000000  176.000000  ...   775.000000
75%   509.000000   32.000000  1035.000000  213.500000  ...   882.000000
max   618.000000   42.000000  1616.000000  382.000000  ...  1125.000000

      Watt_6min  Number_y  Distance      Age  HeartRateBefore \
count  91.000000   91.000000   91.000000   91.000000   91.000000
mean  143.835165   6.241758  1356.527473  18.989011   114.175824
std   39.361082   1.250250   124.779569   1.303794   14.065398
...
75%   170.000000  171.500000  59.000000    2.0
max   180.000000  181.000000  75.000000    2.0

[8 rows x 26 columns]
    
```

Рисунок 4.4 – Групування студентів за їх характеристиками

Застосовується алгоритм кластеризації k-середніх для розділення спортсменів на відповідні кластери. Кожному спортсмену додається мітка кластера, до якого він належить. Надалі виводиться статистична інформація про характеристики спортсменів у кожному кластері, а також проводиться візуалізація кластерів на графіку, що дозволяє візуально оцінити розподіл спортсменів за характеристиками та кластерами, що і продемонстровано на рис. 4.1. Реалізацію можна переглянути в додатку А.

4.1.2 Розрахунок середнього віку за національністю

Для розрахунку середнього віку за національністю використовується метод *groupby* для групування даних за національністю та обчислення середнього віку для кожної групи.

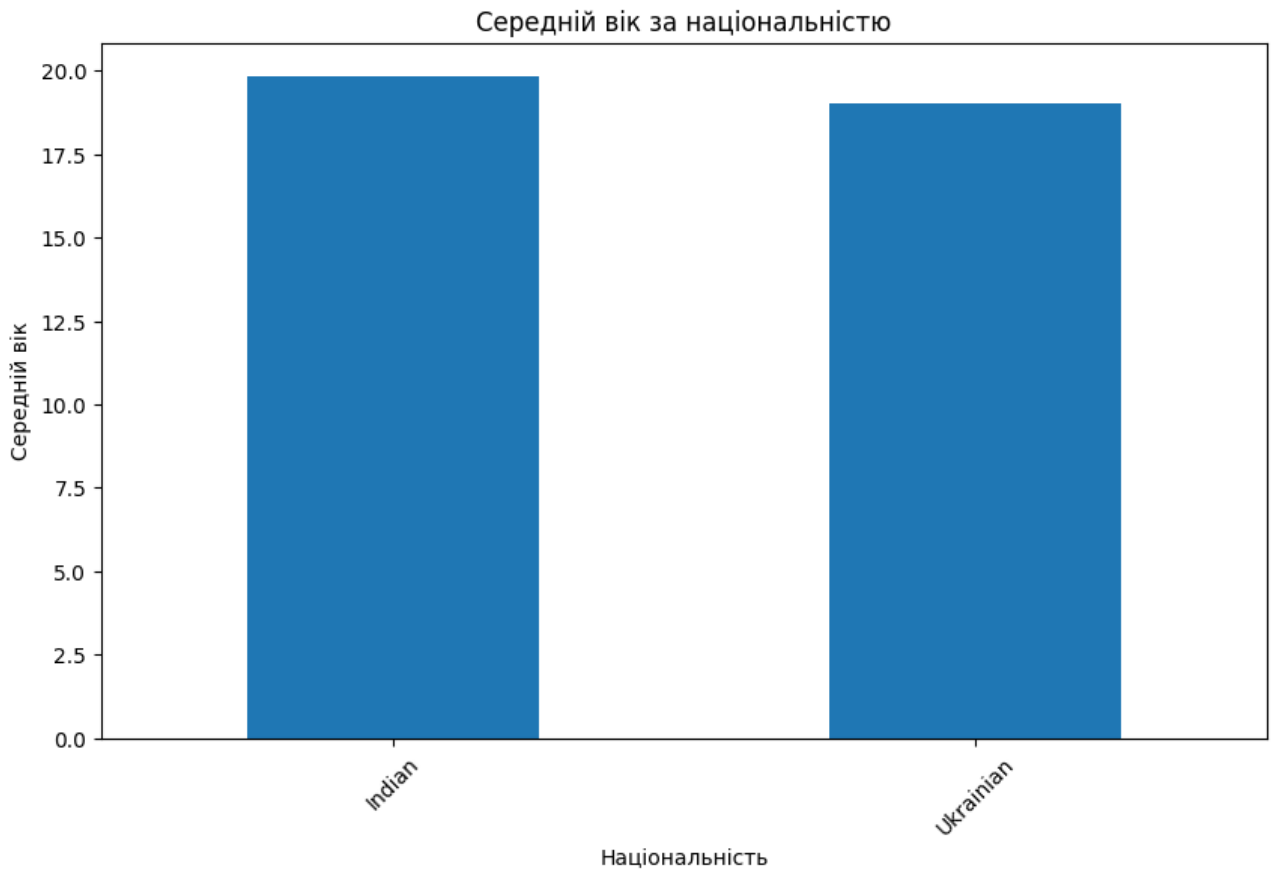


Рисунок 4.5 – Середній вік за національністю

Метод *groupby* групує дані за полем «Nationality», а потім використовується метод `mean()` для обчислення середнього віку для кожної групи. Результатом буде серія, де індексами будуть національності, а значеннями – середній вік для кожної національності.

4.2 Прогнозування результатів

4.2.1 Прогнозування часу проходження студентом 500-та метрів

Для розв’язання задачі прогнозування часу на проходження 500 метрів на весловому тренажері використовується алгоритм машинного навчання на основі довільного лісу. Алгоритм починається із завантаження даних про студентів, включаючи їх характеристики (зріст, вага, пульс до та після тренування) та показники виконання вправ на тренажері (наприклад, середня швидкість, кількість помахів веслом). Для підготовки даних до навчання моделі

використовується перетворення часових даних у стовпці з цільовою змінною (час на проходження 500 метрів) з формату часу в хвилину та секундах у кількість секунд. Далі дані поділяються на навчальний та тестовий набори. Навчання моделі здійснюється за допомогою *RandomForestRegressor*, який навчається на навчальних даних для прогнозування часу на проходження 500 метрів на основі характеристик студентів та їх результатів на тренажері. Для оцінки якості моделі використовують середньоквадратичну помилку (MSE) на тестовому наборі даних. Візуалізація результатів зображена на рис. 4.1 у вигляді scatter plot, що в свою чергу дозволяє порівняти фактичні значення часу на проходження 500 метрів із передбаченими моделлю, що допомагає оцінити точність моделі та її придатність для прогнозування результатів тренувань студентів. Приклад реалізації продемонстровано в додатку А.

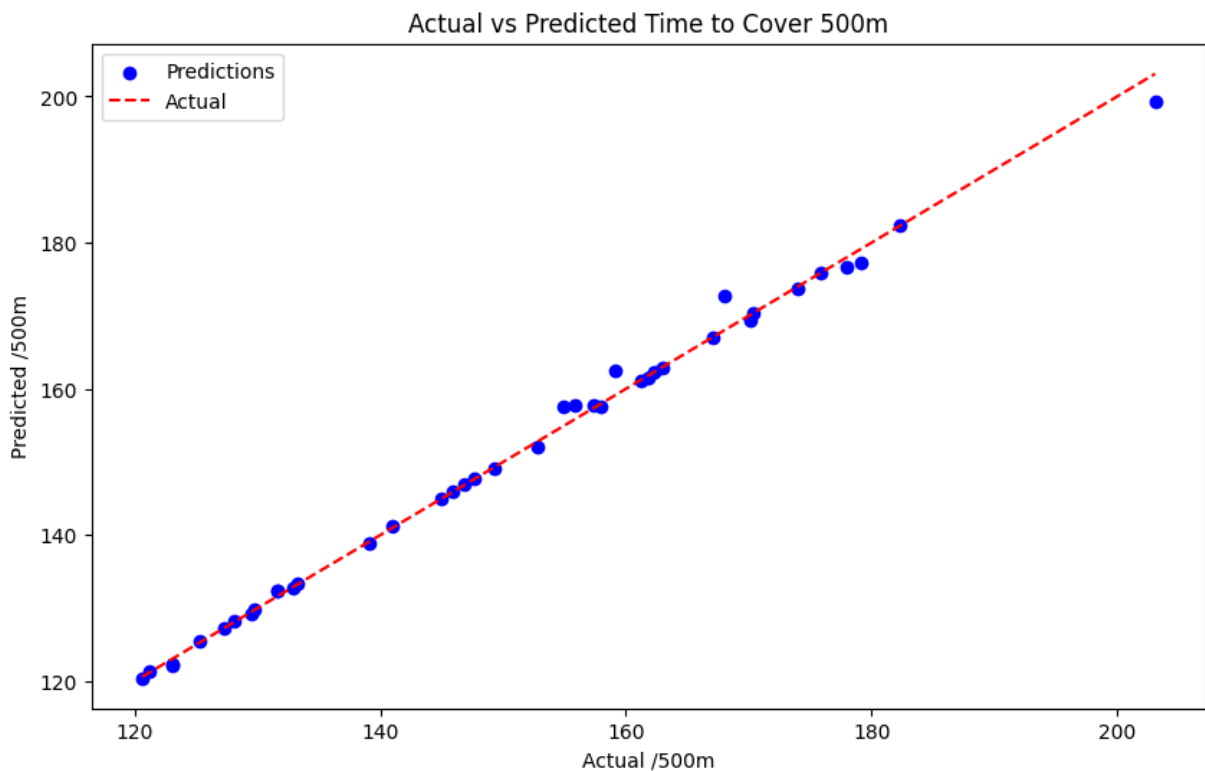


Рисунок 4.6 – Прогнозування часу на проходження 500-та метрів

Прогнозування часу на проходження 500 метрів є досить корисним у контексті оцінки фізичної підготовленості студентів або учасників спортивних

заходів, планування тренувального процесу, оптимізації програм тренувань та виявлення факторів, що впливають на швидкість виконання вправ. Точна модель прогнозування може допомогти тренерам та спортсменам краще зрозуміти та покращити їх фізичну підготовку та результати.

4.2.2 Прогнозування статі студента за його фізичними параметрами з використанням методу опорних векторів

Для прогнозування статі студента за його фізичними параметрами після імпорту даних з файлу вибираються фізичні параметри (вік, зріст, вага, показники пульсу до та після тестування) для прогнозування статі студента. Далі дані розділяються на тренувальний та тестовий набори відповідно, щоб оцінити ефективність моделі. Застосовується модель опорних векторів (SVM) з лінійним ядром для класифікації. Після навчання моделі вона використовується для прогнозування статі студентів у тестовому наборі. На завершення оцінюється точність моделі, а також виводиться інформація про прогнозовані статі перших 10 студентів для подальшого аналізу. Приклад реалізації продемонстровано в додатку Б.

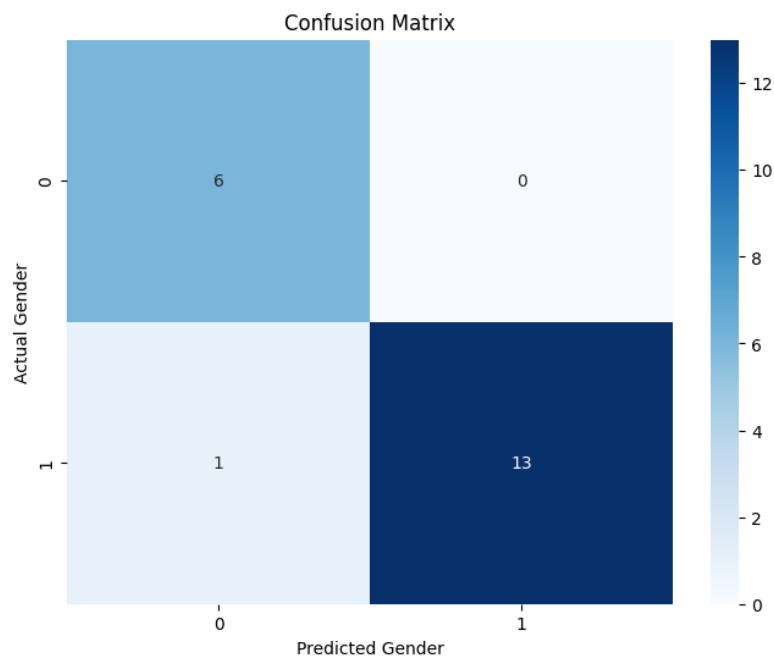


Рисунок 4.7 – Матриця помилок

```
** Accuracy: 0.95
Predicted Genders:
Student 1: ч
Student 2: ж
Student 3: ч
Student 4: ч
Student 5: ж
Student 6: ж
Student 7: ж
Student 8: ч
Student 9: ч
Student 10: ч
```

Рисунок 4.8 – Прогнозовані статі перших 10 студентів

Матриця помилок допомагає зрозуміти, скільки і які помилки робить модель у прогнозуванні класів.

Висновок до розділу 4

В ході виконання четвертого розділу було описано процес розробки застосунку. Також описано процес групування студентів за їх характеристиками та аналіз результатів, використовуючи алгоритм кластеризації k-середніх та візуалізацію результатів на графіку, та процес прогнозування часу проходження студентом 500-та метрів та прогнозування статі студента за його фізичними параметрами, з використанням відповідних алгоритмів машинного навчання. Результати моделей оцінюються на основі точності та візуалізуються для кращого розуміння.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи магістра було виконано поставлені завдання та на їх основі сформовано звіт, який має у собі систематизацію отриманих результатів.

Для досягнення поставленої мети було розв'язано наступні завдання:

- проаналізовано ринок наявних технологій і рішень для вирішення завдань;
- зібрано та оброблено дані тренувань;
- виконано проектування та моделювання системи;
- використано моделі машинного навчання для аналізу отриманих даних;
- оцінено ефективність моделей та їх можливість впровадження у практику академічного веслування;
- апробовано отримані результати.

Проведено аналіз предметної області, розглянути класифікацію систем для аналізу спортивних результатів та розглянув різні методи машинного навчання, які можуть бути використані для прогнозування. Кластерний аналіз, алгоритми регресії, метод опорних векторів, метод нечіткої логіки та нейронні мережі виявилися основними інструментами для розв'язання завдань у даній області.

Аналіз аналогічних систем, зокрема Soccermetrics, дав змогу визначити передовий досвід та інновації, які можна впровадити у власний проєкт. Важливим етапом було визначення специфікацій вимог до програмного забезпечення та чітке визначення задачі, що стало фундаментом подальшого проектування системи.

В першому розділі роботи були досліджені інтелектуальні підходи, такі як кластерний аналіз, алгоритми регресії, метод опорних векторів, метод нечіткої логіки та нейронні мережі, що дозволило визначити оптимальний набір інструментів для подальшої розробки системи.

У другому розділі була описана розробка функціональної моделі системи та побудовано контекстну діаграму, що деталізує внутрішні процеси системи. Це

забезпечило чітке розуміння логіки та взаємозв'язків компонентів системи для ефективного прогнозування спортивних результатів.

Третій розділ роботи присвячено технологіям, необхідним для реалізації системи, зокрема Python, бібліотекам Pandas та Scikit-learn. Створення діаграм прецедентів та сценаріїв використання сприяло кращому розумінню функціоналу системи та взаємодії з нею. Розробка макетів інтерфейсу дозволила візуалізувати функціонал системи та спростити її використання.

У четвертому розділі було успішно описано процес розробки застосунку та проведено аналіз результатів, що отримані за допомогою алгоритмів кластеризації та прогнозування на основі машинного навчання. Це дозволило оцінити ефективність розробленої системи та її здатність до точного прогнозування спортивних результатів.

Отже, результати дослідження та розробки, представлені у даній дипломній роботі, свідчать про успішну реалізацію системи для аналізу та прогнозування спортивних результатів на основі машинного навчання, яка може бути використана для покращення тренувального процесу та досягнення кращих результатів у спортивній діяльності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Академічне веслування – ВУЄ. URL: https://vue.gov.ua/%D0%90%D0%BA%D0%B0%D0%B4%D0%B5%D0%BC%D1%96%D1%87%D0%BD%D0%B5_%D0%B2%D0%B5%D1%81%D0%BB%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F. (дата звернення: 03.12.2023).
2. Горбань Г.В., Кандиба І.О., Антипова К.О., Кірей К.О. Первинний та візуальний аналіз даних спортивних результатів з академічного веслування засобами мови Python з використанням бібліотек Pandas, Matplotlib та Seaborn // Таврійський науковий вісник. Серія: Технічні науки. 2022, Вип. 3, С. 27-37.
3. N. Abdelhamid, A. Ayesh, F. Thabtah, S. Ahmadi, W. Hadi. MAC: A multiclass associative classification algorithm. WorldScinet, 2, 2012. pp. 95-110.
4. Горбань Г.В., Кандиба І.О., Фісун М.Т. Первинна обробка даних результатів спортивних тренувань студентів на веслувальних тренажерах Concept2 для подальшого аналізу за допомогою бібліотеки Pandas // Таврійський науковий вісник. Серія: Технічні науки. 2022, Вип. 2, С. 33-44.
5. Ghasemzadeh H., Jafari R. Coordination analysis of human movements with body sensor networks: A signal processing model to evaluate baseball swings. IEEE Sensors Journal. 2011. Vol 11(3). pp. 603-610.
6. Кластерний аналіз, багатовимірний статистичний аналіз. Основні поняття, Визначення, цілі та алгоритм кластерного аналізу. https://stud.com.ua/75019/statistika/klasterniy_analiz (дата звернення: 03.12.2023).
7. Пояснення регресії та класифікації в машинному навчанні. URL: <https://techukraine.net/%d0%bf%d0%be%d1%8f%d1%81%d0%bd%d0%b5%d0%bd%d0%bd%d1%8f%d1%80%d0%b5%d0%b3%d1%80%d0%b5%d1%81%d1%96%d1%97%d1%82%d0%b0%d0%ba%d0%bb%d0%b0%d1%81%d0%b8%d1%84%d1%96%d0%ba%d0%b0%d1%86%d1%96%d1%97-%d0%b2/> (дата звернення: 04.12.2023).
8. Черепанський М.В., Молодецька К.В. Доцільність використання методу опорних векторів для вирішення задач класифікації в машинному навчанні //

Науковий вісник Житомирського державного технологічного університету. 2017. Вип. 2. С. 67-106.

9. Теорія алгоритмів та математична логіка. URL: https://elearning.sumdu.edu.ua/free_content/lectured:075b2e8a0bfe48bcef0ab3106c6d51679abc41f9/latest/117286/index.html . (дата звернення: 05.12.2023).

10. Що таке штучний інтелект: коротко, простими словами. URL: <https://bizmag.com.ua/shho-take-shtuchnyj-intelekt/>. (дата звернення: 07.12.2023).

11. Мілян Н. Аналіз методів машинного навчання з вчителем // Міжнародна студентська науково-технічна конференція «Природничі та гуманітарні науки. Актуальні питання». 2017. Вип. 3. С. 51-52.

12. Що таке штучний інтелект: коротко, простими словами. URL: (дата звернення: 11.12.2023).

13. Soccermetics. URL: <https://soccermetics.app/> . (дата звернення: 15.12.2023).

14. IBM Watson for Wimbledon. URL: <https://soccermetics.app/> . (дата звернення: 16.12.2023).

15. Радченко Д.М. Структурно-функціональне моделювання бізнес процесу управління фінансовим плануванням на підприємстві // Молодий вчений. 2016. Вип. 6 (33). С. 94-99.

16. Загальні поширені запитання щодо Python – Python 3.12.2 documentation. URL: <https://docs.python.org/uk/3/faq/general.html> . (дата звернення: 25.12.2023).

17. Ось чому Pandas є найпопулярнішою бібліотекою аналізу даних Python URL: <https://techukraine.net/%d0%be%d1%81%d1%8c-%d1%87%d0%be%d0%bc%d1%83-pandas-%d1%94%d0%bd%d0%b0%d0%b9%d0%bf%d0%be%d0%bf%d1%83%d0%bb%d1%8f%d1%80%d0%bd%d1%96%d1%88%d0%be%d1%8e%d0%b1%d1%96%d0%b1%d0%bb%d1%96%d0%be%d1%82%d0%b5/>. (дата звернення: 26.12.2023).

18. Scikit-learn Tutorials — scikit-learn 1.4.1 documentation. URL: <https://scikit-learn.org/stable/tutorial/index.html>. (дата звернення: 27.12.2023).
19. tkinter – Інтерфейс Python до Tcl/Tk. URL: <https://docs.python.org/uk/3/library/tkinter.html>. (дата звернення: 30.12.2023).
20. Use case diagrams. URL: <https://www.quality-assurance-group.com/use-case-diagrams/> (дата звернення: 05.01.2024).
21. Що таке мокап і як його використовують дизайнери?. URL: <https://web-promo.ua/ua/blog/sho-take-mokap-i-yak-jogo-vikoristovuyut-dizajneri/> (дата звернення: 05.01.2024).
22. Застосування uml (частина 3). діаграма класів - class diagram. URL: https://duikt.edu.ua/ua/news-1-626-8002-zastosuvannya-uml-chastina-3-diagrama-klasiv----class-diagram_kafedra-kompyuternih-nauk-ta-informaciynih-tehnologiy (дата звернення: 07.01.2024).

ДОДАТОК А

Код імпорту, виведення, редагування, видалення та збереження даних

```
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.cluster import KMeans

class AthleteDataModel:
    def __init__(self, data):
        self.data = data

    def predict_watt(self):
        # Підготовка даних для моделі
        features = ['Height', 'Weight', 'HeartRateBefore', 'HeartRateAfter', 'Meters', 'Avg
SPM', 'Cal/hr']
        target = 'Watt'

        X = self.data[features]
        y = self.data[target]

        # Розділення даних на тренувальний та тестовий набори
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

        # Навчання моделі RandomForestRegressor
```

```
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Оцінка якості моделі на тестовому наборі даних
predictions = model.predict(X_test)
mse = mean_squared_error(y_test, predictions)
print("Mean Squared Error:", mse)

# Візуалізація фактичних значень та прогнозів
plt.figure(figsize=(10, 6))
plt.scatter(y_test, predictions, color='blue', label='Predictions')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], '--k', color='red',
label='Actual')
plt.xlabel('Actual Watt')
plt.ylabel('Predicted Watt')
plt.title('Actual vs Predicted Watt')

# Створення віджету для відображення графіки в Tkinter
root = tk.Tk()
canvas = FigureCanvasTkAgg(plt.gcf(), master=root)
canvas.draw()
canvas.get_tk_widget().pack()
root.mainloop()

def perform_clustering(self):
    # Вибірка характеристик для кластерного аналізу
    features = self.data[['Age', 'Height', 'Weight', 'HeartRateBefore', 'HeartRateAfter',
'Meters']]
```

```
# Нормалізація даних
normalized_features = (features - features.mean()) / features.std()

# Визначення кількості кластерів (груп)
num_clusters = 3

# Застосування алгоритму кластеризації
kmeans = KMeans(n_clusters=num_clusters)
kmeans.fit(normalized_features)

# Додавання стовпця із мітками кластерів до даних
self.data['Cluster'] = kmeans.labels_

# Вивід інформації про кожен кластер
for cluster_num in range(num_clusters):
    cluster_data = self.data[self.data['Cluster'] == cluster_num]
    print(f"Cluster {cluster_num + 1}:")
    print(cluster_data.describe())

# Візуалізація кластерів
plt.figure(figsize=(10, 6))
for cluster_num in range(num_clusters):
    cluster_data = self.data[self.data['Cluster'] == cluster_num]
    plt.scatter(cluster_data['Age'], cluster_data['Meters'], label=f'Cluster
{cluster_num + 1}')
plt.xlabel('Age')
plt.ylabel('Meters')
plt.title('Clusters of Athletes')
plt.legend()
```

```
# Створення віджету для відображення графіки в Tkinter
root = tk.Tk()
canvas = FigureCanvasTkAgg(plt.gcf(), master=root)
canvas.draw()
canvas.get_tk_widget().pack()
root.mainloop()

class GUI(tk.Tk):
    def __init__(self):
        super().__init__()

        self.title("Athlete Data Viewer")
        self.geometry("800x600")

        self.button_frame = tk.Frame(self)
        self.button_frame.pack(side=tk.TOP, fill=tk.X)

        self.load_data_button = tk.Button(self.button_frame, text="Load Data",
command=self.load_data)
        self.load_data_button.pack(side=tk.LEFT, padx=5, pady=5)

        self.predict_watt_button = tk.Button(self.button_frame, text="Predict Watt",
command=self.predict_watt)
        self.predict_watt_button.pack(side=tk.LEFT, padx=5, pady=5)

        self.cluster_button = tk.Button(self.button_frame, text="Perform Clustering",
command=self.perform_clustering)
        self.cluster_button.pack(side=tk.LEFT, padx=5, pady=5)
```

```
self.tree = ttk.Treeview(self)
self.tree.pack(fill=tk.BOTH, expand=True)

self.tree.bind("<Double-1>", self.on_double_click)

self.data = None
self.model = None

def load_data(self):
    file_path = filedialog.askopenfilename(filetypes=[("CSV files", "*.csv")])
    if file_path:
        self.data = pd.read_csv(file_path)

        self.tree.delete(*self.tree.get_children())

        if not self.data.empty:
            self.tree["columns"] = list(self.data.columns)
            for col in self.tree["columns"]:
                self.tree.heading(col, text=col)
                self.tree.column(col, width=100)

            for index, row in self.data.iterrows():
                self.tree.insert("", "end", text=index, values=list(row))

        self.model = AthleteDataModel(self.data)

def on_double_click(self, event):
    item = self.tree.selection()[0]
```



```
values = self.tree.item(item, "values")
columns = self.tree["columns"]
self.edit_window = EditWindow(self, values, columns)
self.edit_window.grab_set()

def predict_watt(self):
    if self.model:
        self.model.predict_watt()

def perform_clustering(self):
    if self.model:
        self.model.perform_clustering()

class EditWindow(tk.Toplevel):
    def __init__(self, parent, values, columns):
        super().__init__(parent)

        self.title("Edit Data")

        self.values = values
        self.columns = columns

        self.canvas = tk.Canvas(self)
        self.canvas.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

        self.scrollbar = tk.Scrollbar(self, orient=tk.VERTICAL,
command=self.canvas.yview)
        self.scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
```

```
self.canvas.configure(yscrollcommand=self.scrollbar.set)
self.canvas.bind('<Configure>', lambda e:
self.canvas.configure(scrollregion=self.canvas.bbox("all")))

self.frame = tk.Frame(self.canvas)
self.canvas.create_window((0, 0), window=self.frame, anchor='nw')

self.entries = []
for i, value in enumerate(self.values):
    label = tk.Label(self.frame, text=self.columns[i])
    label.grid(row=i, column=0, padx=5, pady=5)

    entry = tk.Entry(self.frame)
    entry.grid(row=i, column=1, padx=5, pady=5)
    entry.insert(0, value)

    self.entries.append(entry)

self.save_button = tk.Button(self, text="Save", command=self.save)
self.save_button.pack()

def save(self):
    new_values = [entry.get() for entry in self.entries]
    selected_index = self.master.tree.index(self.master.tree.selection()[0])
    self.master.data.iloc[selected_index, :] = new_values
    self.master.tree.item(self.master.tree.selection()[0], values=new_values)
    self.destroy()
```

ДОДАТОК Б

Код прогнозування статі студента за його фізичними параметрами з використанням методу опорних векторів

```
# Вибір фізичних параметрів для прогнозування статі
X = data[['Age', 'Height', 'Weight', 'HeartRateBefore', 'HeartRateAfter']]
y = data['Sex'] # Цільова змінна - стать студента

# Розділення даних на тренувальний та тестовий набори
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1,
random_state=100)

# Побудова моделі SVM
svm_model = SVC(kernel='linear', random_state=101)
svm_model.fit(X_train, y_train)

# Прогнозування на тестовому наборі
y_pred = svm_model.predict(X_test)

# Оцінка точності моделі
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Виведення інформації про прогнозовані статі студентів
print("Predicted Genders:")
for idx, gender in enumerate(y_pred[:10]): # Виведення перших 10 прогнозів
    print(f"Student {idx+1}: {gender}")

# Побудова матриці помилок
conf_matrix = confusion_matrix(y_test, y_pred)
```

```
# Візуалізація матриці помилок
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted Gender')
plt.ylabel('Actual Gender')
plt.title('Confusion Matrix')
plt.show()
```