

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

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра програмної інженерії

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Бакалавр

(назва освітнього ступеня)

на тему: **Розробка програмного забезпечення та тестування системи
формування персоналізованого плану тренувань з використанням
методів машинного навчання**

Виконав(ла): студент(ка) 4 курсу, групи СП-42
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

(підпис)

Кобель Б. М.

(прізвище та ініціали)

Керівник

(підпис)

Цебрій О.Р.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Стоянов Ю.М.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Петрик М.Р.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль
2026

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

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

Кафедра програмної інженерії
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ (підпис) _____ (прізвище та ініціали)
« » 20__ р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавра
(назва освітнього ступеня)

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

студенту Кобель Богдан Михайлович
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка програмного забезпечення та тестування системи формування персоналізованого плану тренувань з використанням методів машинного навчання

Керівник роботи Цебрій О.Р., канд. фіз.-мат. наук, доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «__» _____ 20__ року № _____

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Предметна область, завдання, вимоги та специфікація, програмне рішення, методичні вказівки

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступна частина

Аналіз предметної області та теоретичних основ

Визначення методики реалізації моделі

Реалізація моделі

Визначення основних аспектів охорони праці та безпеки життєдіяльності

Висновки роботи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

Слайди презентації та діаграми процесів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці			
Нормоконтроль	Стоянов Ю.М. к.т.н., доц. каф. ПІ		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	<i>Розробка технічного завдання</i>	<i>6.04 – 12.04</i>	
2	<i>Робота над першим розділом «АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ»</i>	<i>13.04 – 26.04</i>	
3	<i>Робота над другим розділом «ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ МОНІТОРИНГУ ТА ПРОГНОЗУВАННЯ ЗАБРУДНЕННЯ ПОВІТРЯ»</i>	<i>27.04 – 03.05</i>	
4	<i>Робота над третім розділом «ТЕСТУВАННЯ, ОЦІНЮВАННЯ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ»</i>	<i>04.05 – 17.05</i>	
5	<i>Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»</i>	<i>18.05 – 24.05</i>	
6	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	<i>25.05 – 7.06</i>	
7	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	<i>8.06 – 14.06</i>	
8	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	<i>15.06 – 21.06</i>	
9	<i>Захист кваліфікаційної роботи бакалавра</i>		

Студент

_____ (підпис)

Кобель Б.М.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Цебрій О.Р.

_____ (прізвище та ініціали)

АНОТАЦІЯ

Розробка програмного забезпечення та тестування системи формування персоналізованого плану тренувань з використанням методів машинного навчання // Кваліфікаційна робота освітнього рівня «Бакалавр» // Кобаль Богдан Михайлович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СП-42 // Тернопіль, 2026 // С. ____, рис. – ____, табл. – ____, додат. – ____, бібліогр. – ____.

Ключові слова: персоналізовані тренування; Random Forest; фітнес; тренувальний план; вебзастосунок; аналіз даних.

Кваліфікаційна робота присвячена розробці інтелектуальної системи формування персоналізованих планів тренувань на основі методів машинного навчання.

У роботі проаналізовано підходи до персоналізації тренувальних програм і рекомендаційних систем, сформовано вимоги та спроектовано архітектуру програмної системи.

Для навчання моделі використано відкритий набір даних Kaggle. Рекомендаційний механізм реалізовано на основі алгоритму Random Forest Classifier для визначення типу тренувального плану та рівня навантаження.

Об'єкт дослідження – процес формування персоналізованих тренувальних програм.

Предмет дослідження – методи та програмні засоби формування тренувальних рекомендацій із використанням машинного навчання.

Практичним результатом є вебсистема, що забезпечує введення параметрів користувача та автоматичне формування персоналізованого плану тренувань.

ABSTRACT

Software development and testing of a system for generating a personalized training plan using machine learning methods // Bachelor's Qualification Thesis // Bohdan Kobel // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, Group SP-42 // Ternopil, 2026 // P. ____, fig. – ____, tabl. – ____, annexes – ____, references – ____.

Keywords: personalized training; Random Forest; fitness; training plan; web application; data analysis.

The qualification thesis is devoted to the development of an intelligent system for generating personalized training plans based on machine learning methods.

The thesis analyzes approaches to the personalization of training programs and recommendation systems, defines system requirements, and designs the architecture of the software system.

An open Kaggle dataset was used for training the model. The recommendation mechanism was implemented using the Random Forest Classifier algorithm to determine the type of training plan and the recommended workload level.

The object of research is the process of generating personalized training programs.

The subject of research is methods and software tools for generating training recommendations using machine learning techniques.

The practical result of the work is a web-based system that provides user parameter input and automatic generation of personalized training plans.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИЗНАЧЕННЯ ВИМОГ ДО ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.....	12
1.1 Аналіз сучасних підходів до персоналізації тренувальних програм.....	12
1.2 Огляд існуючих програмних рішень для планування тренувань.....	14
1.3 Аналіз методів машинного навчання для побудови рекомендаційних систем.....	16
1.4 Формування функціональних та нефункціональних вимог до системи.....	18
1.5 Визначення сценаріїв використання та користувацьких ролей.....	22
1.6 Постановка задачі розробки інтелектуальної системи.....	25
1.7 Висновки до першого розділу.....	26
2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ.....	28
2.1 Проєктування архітектури програмної системи.....	28
2.2 Проєктування структури даних та бази даних.....	31
2.3 Розробка модуля підготовки та обробки даних.....	35
2.4 Розробка моделі машинного навчання для формування персоналізованих планів тренувань.....	37
2.5 Реалізація серверної частини системи.....	42
2.6 Розробка користувацького інтерфейсу.....	45
2.7 Висновки до розділу 2.....	48
3 ТЕСТУВАННЯ, ВПРОВАДЖЕННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ.....	49
3.1 Формування набору даних для навчання та тестування моделі.....	49
3.2 Оцінювання ефективності та результати роботи моделі машинного навчання	

3.3 Тестування функціональних можливостей та аналіз якості роботи програмної системи.....	53
3.4 Впровадження системи та рекомендації щодо використання.....	55
3.5 Висновки до розділу 3.....	56
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....	57
4.1 Інженерно-психологічні принципи професійного добору.....	57
4.2 Інженерно-технічні рішення з охорони праці.....	59
4.3 Висновки до четвертого розділу.....	62
ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
ДОДАТКИ.....	68
Додаток А.....	69
Додаток Б.....	81

ПЕРЕЛІК СКОРОЧЕНЬ

- ML – Machine Learning (машинне навчання)
- AI – Artificial Intelligence (штучний інтелект)
- DL – Deep Learning (глибинне навчання)
- EDA – Exploratory Data Analysis (розвідувальний аналіз даних)
- CSV – Comma-Separated Values (текстовий формат даних)
- LR – Logistic Regression (логістична регресія)
- DT – Decision Tree (дерево рішень)
- kNN – k-Nearest Neighbors (метод k найближчих сусідів)
- SVM – Support Vector Machine (метод опорних векторів)
- NB – Naive Bayes (наївний байєсівський класифікатор)
- NN – Neural Network (нейронна мережа)
- RF – Random Forest (випадковий ліс)
- GBM – Gradient Boosting Machine (градієнтний бустинг)
- XGB – XGBoost (Extreme Gradient Boosting)
- LGBM – LightGBM (Light Gradient Boosting Machine)
- CB – CatBoost (Categorical Boosting)
- SMOTE – Synthetic Minority Oversampling Technique (синтетичне збільшення вибірки меншості)
- ROC – Receiver Operating Characteristic (крива робочих характеристик приймача)
- AUC – Area Under Curve (площа під кривою)
- API – Application Programming Interface (інтерфейс прикладного програмування)
- CPU – Central Processing Unit (центральний процесор)
- RAM – Random Access Memory (оперативна пам'ять)
- JSON – JavaScript Object Notation (формат даних JSON)

ВСТУП

Персоналізація тренувального процесу є одним із важливих напрямів розвитку сучасних інформаційних технологій у сфері фітнесу та здорового способу життя. В умовах зростання популярності спортивних занять та цифрових сервісів користувачі потребують інструментів, які дозволяють формувати індивідуальні тренувальні програми з урахуванням їх фізичних характеристик, рівня підготовки та поставлених цілей. Використання універсальних тренувальних планів часто не забезпечує необхідної ефективності, оскільки не враховує особливості конкретної людини та може призводити до перевантаження або недостатньої інтенсивності занять.

Сучасні інформаційні системи дозволяють накопичувати значні обсяги даних про користувачів, їх фізичний стан, результати тренувань та спортивні вподобання. Аналіз таких даних створює передумови для автоматизації процесу підбору тренувальних програм та формування персоналізованих рекомендацій. Традиційні підходи до створення тренувальних планів базуються переважно на експертних знаннях тренерів та використанні заздалегідь підготовлених шаблонів, що обмежує можливості їх адаптації до потреб конкретного користувача.

Для розв'язання цієї проблеми дедалі ширше використовуються методи машинного навчання, які дозволяють автоматично виявляти закономірності у даних та формувати рекомендації на основі характеристик користувачів. Застосування алгоритмів машинного навчання дає можливість підвищити рівень персоналізації тренувальних програм, автоматизувати процес прийняття рішень та забезпечити більш ефективний підбір фізичних навантажень відповідно до поставлених цілей.

Актуальність теми кваліфікаційної роботи зумовлена необхідністю розробки інтелектуальних програмних засобів для автоматизованого формування персоналізованих тренувальних планів. Використання методів машинного навчання дозволяє підвищити якість рекомендацій, скоротити час підбору

тренувальних програм та забезпечити індивідуальний підхід до кожного користувача.

Метою кваліфікаційної роботи є розробка інтелектуальної системи формування персоналізованого плану тренувань на основі методів машинного навчання.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз предметної області персоналізації тренувальних програм;
- дослідити сучасні підходи до побудови рекомендаційних систем у сфері фітнесу;
- проаналізувати методи машинного навчання для формування тренувальних рекомендацій;
- сформулювати функціональні та нефункціональні вимоги до програмної системи;
- спроектувати архітектуру програмного рішення та структуру бази даних;
- реалізувати модуль підготовки та обробки даних;
- розробити модель машинного навчання для формування рекомендацій;
- реалізувати серверну частину та користувацький інтерфейс системи;
- провести тестування та оцінювання ефективності роботи розробленої системи.

Об'єктом дослідження є процес формування персоналізованих тренувальних програм на основі характеристик користувачів.

Предметом дослідження є методи, моделі та програмні засоби формування персоналізованих тренувальних рекомендацій із використанням технологій машинного навчання.

Під час виконання роботи використано методи системного аналізу, машинного навчання, класифікації даних, об'єктно-орієнтованого проектування,

проектування баз даних, розробки вебзастосунків та тестування програмного забезпечення.

Практичне значення одержаних результатів полягає у розробці програмної системи, яка дозволяє автоматизувати процес формування персоналізованих тренувальних планів на основі індивідуальних характеристик користувачів. Розроблена система може використовуватися у фітнес-клубах, спортивних центрах, навчальних закладах або для індивідуального планування тренувального процесу.

Кваліфікаційна робота присвячена дослідженню методів машинного навчання для задач персоналізації тренувань та розробці інтелектуальної системи формування тренувальних рекомендацій. У роботі проведено аналіз сучасних підходів до персоналізації тренувальних програм, спроектовано архітектуру програмної системи, реалізовано рекомендаційний механізм на основі алгоритму Random Forest та виконано оцінювання ефективності розробленого програмного забезпечення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИЗНАЧЕННЯ ВИМОГ ДО ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

Сучасні інформаційні технології активно впроваджуються у сферу фітнесу та спортивної підготовки, забезпечуючи можливість автоматизації процесів планування, моніторингу та аналізу тренувальної діяльності. Особливої актуальності набувають інтелектуальні системи, здатні формувати персоналізовані рекомендації на основі індивідуальних характеристик користувача, його фізичного стану, цілей тренувань та історії попередніх занять.

Використання методів машинного навчання дозволяє підвищити ефективність процесу персоналізації тренувальних програм завдяки аналізу великих обсягів даних та виявленню прихованих закономірностей між параметрами користувача і результативністю виконання фізичних вправ. У даному розділі проведено аналіз сучасних підходів до формування персоналізованих тренувальних програм, досліджено існуючі програмні рішення, розглянуто методи машинного навчання, які можуть бути використані для побудови рекомендаційної системи, а також сформовано функціональні та нефункціональні вимоги до розроблюваної інтелектуальної системи [1].

1.1 Аналіз сучасних підходів до персоналізації тренувальних програм

Персоналізація тренувальних програм є одним із ключових напрямів розвитку сучасних фітнес-технологій. На відміну від традиційних універсальних програм тренувань, персоналізований підхід враховує індивідуальні особливості користувача, зокрема вік, стать, фізичну підготовку, антропометричні показники, стан здоров'я, спортивний досвід та поставлені цілі. Такий підхід дозволяє підвищити ефективність тренувального процесу, знизити ризик травмування та покращити мотивацію користувача до регулярних занять.

Сучасні підходи до персоналізації тренувань можна умовно поділити на декілька груп, структура яких наведена на рисунку 1.1.

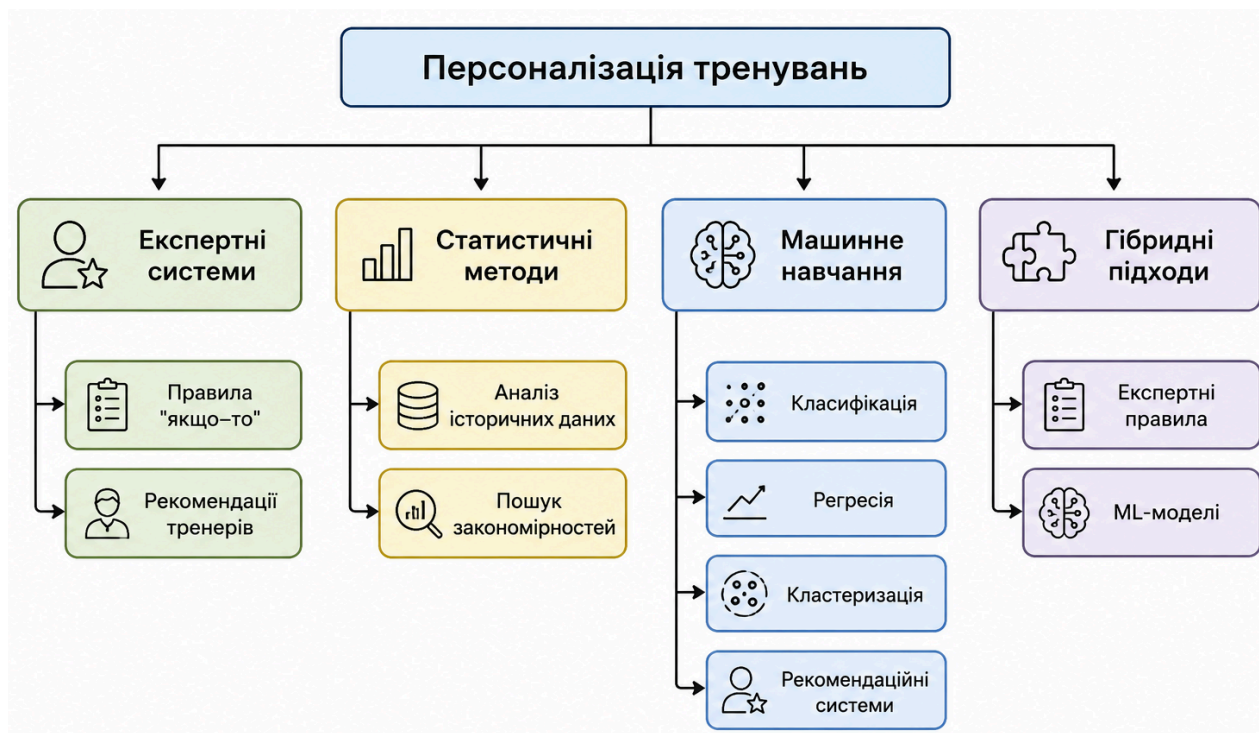


Рисунок 1.1 – Класифікація сучасних підходів до персоналізації тренувальних програм

Традиційні експертні системи базуються на знаннях професійних тренерів та заздалегідь визначених правилах формування тренувальних програм. Такі системи використовують набори логічних правил типу «якщо–то», які дозволяють генерувати рекомендації на основі введених користувачем параметрів. Основною перевагою даного підходу є простота реалізації та зрозумілість процесу формування рекомендацій. Водночас ефективність таких систем суттєво залежить від повноти бази знань і не дозволяє враховувати складні залежності між параметрами користувача [2].

Статистичні підходи використовують результати аналізу накопичених даних про тренування та фізичний розвиток користувачів. На основі статистичних моделей визначаються типові закономірності, які можуть бути використані для побудови рекомендацій. Проте такі методи обмежені у здатності адаптуватися до індивідуальних особливостей кожного користувача [3].

Найбільш перспективним напрямом є застосування методів машинного навчання. У таких системах моделі автоматично навчаються на історичних даних користувачів та можуть прогнозувати оптимальні параметри тренувань залежно від поставлених цілей. Для цього використовуються алгоритми класифікації, регресії, кластеризації та рекомендаційні моделі. Перевагою даного підходу є можливість автоматичного вдосконалення рекомендацій у процесі накопичення нових даних [4].

Окрему групу складають гібридні системи, які поєднують експертні правила та методи машинного навчання. Такі рішення дозволяють забезпечити як надійність рекомендацій на початкових етапах використання системи, так і їх подальшу адаптацію до індивідуальних особливостей користувача.

Аналіз сучасних підходів показав, що найбільший потенціал для створення ефективної системи персоналізації тренувань мають саме методи машинного навчання та гібридні підходи, оскільки вони забезпечують високу адаптивність, масштабованість та здатність враховувати індивідуальні характеристики користувачів.

1.2 Огляд існуючих програмних рішень для планування тренувань

Стрімкий розвиток цифрових технологій сприяв появі великої кількості програмних продуктів, призначених для автоматизації процесу планування та контролю фізичної активності. Сучасні системи дозволяють користувачам формувати індивідуальні тренувальні програми, відстежувати результати занять, аналізувати показники фізичної активності та отримувати рекомендації щодо подальшого тренувального процесу. Значна частина таких рішень реалізована у вигляді мобільних застосунків або веб-платформ, що забезпечують постійний доступ до персональних даних користувача [5].

Для визначення функціональних можливостей, які доцільно реалізувати у розроблюваній системі, було проведено аналіз найбільш популярних програмних

рішень для планування тренувань. Порівняльна характеристика основних систем наведена в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика сучасних програмних рішень для планування тренувань

Програмне рішення	Платформа	Персоналізація	Використання ШІ	Формування плану тренувань	Аналітика прогресу
Nike Training Club	Android, iOS	Так	Ні	Так	Так
Freeletics	Android, iOS, Web	Так	Частково	Так	Так
Fitbod	Android, iOS	Так	Так	Так	Так
Strong	Android, iOS	Обмежена	Ні	Частково	Так
JEFIT	Android, iOS, Web	Так	Ні	Так	Так
Gymshark Training	Android, iOS	Обмежена	Ні	Так	Частково

Серед розглянутих рішень найбільшого поширення набули платформи Nike Training Club та Freeletics, які пропонують широкий набір готових тренувальних програм для користувачів різного рівня підготовки. Дані системи використовують персоналізацію на основі анкетних даних користувача та обраних цілей тренування [6].

Застосунок Fitbod є одним із прикладів використання алгоритмів штучного інтелекту для автоматичного формування тренувальних планів. Система аналізує попередню історію тренувань, рівень навантаження на окремі групи м'язів та адаптує майбутні рекомендації відповідно до прогресу користувача.

Платформа Strong орієнтована переважно на ведення журналу тренувань та моніторинг спортивних результатів. Незважаючи на наявність інструментів статистичного аналізу, можливості автоматичного формування персоналізованих програм у даному рішенні є обмеженими.

Застосунок JEFIT поєднує функції планування тренувань, відстеження фізичних показників та використання великої бази вправ. Водночас більшість

рекомендацій у системі формується на основі шаблонів, що знижує рівень індивідуалізації [7].

Проведений аналіз показав, що більшість сучасних програмних продуктів забезпечують базову персоналізацію тренувань, проте лише окремі рішення використовують методи машинного навчання для автоматичної адаптації рекомендацій до змін фізичного стану користувача. Крім того, існуючі системи часто є комерційними продуктами із закритими алгоритмами прийняття рішень, що ускладнює оцінювання ефективності використовуваних підходів. Це підтверджує доцільність розробки власної інтелектуальної системи формування персоналізованих планів тренувань на основі методів машинного навчання.

1.3 Аналіз методів машинного навчання для побудови рекомендаційних систем

Рекомендаційні системи є одним із найбільш поширених напрямів застосування методів машинного навчання. Їх основним призначенням є автоматичне формування рекомендацій на основі аналізу характеристик користувачів, їх поведінки та накопичених історичних даних. У контексті планування тренувань рекомендаційна система повинна визначати оптимальні вправи, інтенсивність навантаження, тривалість занять та послідовність виконання тренувальних програм відповідно до індивідуальних особливостей користувача [8].

Сучасні рекомендаційні системи використовують широкий спектр алгоритмів машинного навчання, кожен з яких має власні переваги та обмеження. Для вибору найбільш доцільного підходу до реалізації інтелектуальної системи персоналізації тренувань було проведено порівняльний аналіз основних методів машинного навчання, результати якого наведено в таблиці 1.2.

Методи класифікації використовуються для віднесення користувачів до певних груп залежно від рівня фізичної підготовки, досвіду тренувань або поставлених цілей. Після визначення класу користувача система може

рекомендувати відповідний набір вправ та параметрів тренування. До найбільш поширених алгоритмів даної групи належать дерева рішень, Random Forest, метод опорних векторів та нейронні мережі [9].

Таблиця 1.2 – Порівняльна характеристика методів машинного навчання для побудови рекомендаційних систем

Метод	Призначення	Переваги	Недоліки
Класифікація	Визначення категорії користувача	Простота інтерпретації результатів	Обмежена гнучкість рекомендацій
Регресія	Прогнозування числових параметрів тренування	Висока точність прогнозів	Чутливість до якості даних
Кластеризація	Групування схожих користувачів	Виявлення прихованих закономірностей	Складність вибору кількості кластерів
Колаборативна фільтрація	Пошук рекомендацій на основі схожих користувачів	Висока якість персоналізації	Проблема «холодного старту»
Контентна фільтрація	Аналіз характеристик користувача та вправ	Не потребує великої кількості користувачів	Обмежене різноманіття рекомендацій
Нейронні мережі	Побудова складних прогнозних моделей	Висока точність та адаптивність	Значні обчислювальні витрати
Ансамблеві методи	Комбінування результатів моделей	Підвищення точності прогнозів	Складність реалізації

Методи регресії дозволяють прогнозувати числові параметри тренувального процесу, зокрема рекомендовану тривалість заняття, кількість підходів, повторень або величину навантаження. Такі моделі забезпечують більш гнучке налаштування тренувальної програми відповідно до індивідуальних характеристик користувача [10].

Алгоритми кластеризації використовуються для автоматичного виявлення груп користувачів із подібними характеристиками. Отримані кластери можуть застосовуватися для формування типових профілів спортсменів та подальшого

використання рекомендацій, які добре зарекомендували себе для аналогічних груп користувачів.

Особливе місце займають нейронні мережі та методи глибинного навчання, які здатні виявляти складні нелінійні залежності між великою кількістю параметрів. Такі моделі забезпечують високу точність прогнозування, однак потребують значних обчислювальних ресурсів та великих обсягів навчальних даних.

Для задач побудови рекомендаційних систем найбільш поширеними є такі підходи:

- контентно-орієнтована фільтрація, яка формує рекомендації на основі характеристик користувача та властивостей об'єктів;
- колаборативна фільтрація, що використовує інформацію про поведінку схожих користувачів;
- гібридні рекомендаційні системи, які поєднують декілька методів для підвищення точності рекомендацій;
- моделі на основі нейронних мереж, здатні автоматично виявляти приховані закономірності у даних;
- ансамблеві методи машинного навчання, які комбінують результати декількох моделей для отримання більш стабільних рекомендацій.

Перевагою використання методів машинного навчання у сфері персоналізації тренувань є можливість постійного вдосконалення рекомендацій у процесі накопичення нових даних про користувача. Завдяки цьому система здатна адаптуватися до змін фізичної форми спортсмена та забезпечувати більш ефективне планування тренувального процесу.

1.4 Формування функціональних та нефункціональних вимог до системи

Формування вимог є одним із ключових етапів розробки програмної системи, оскільки саме на цьому етапі визначаються основні можливості

майбутнього програмного продукту, обмеження його функціонування та очікувані показники якості. Для інтелектуальної системи формування персоналізованого плану тренувань вимоги повинні враховувати як потреби кінцевого користувача, так і особливості використання методів машинного навчання для аналізу вхідних даних та генерації рекомендацій [11].

Функціональні вимоги описують конкретні дії, які повинна виконувати система. До таких дій належать реєстрація користувача, введення персональних параметрів, формування тренувального плану, перегляд рекомендацій, збереження історії тренувань та оновлення рекомендацій на основі змін у профілі користувача. Перелік основних функціональних вимог до системи наведено в таблиці 1.3.

Таблиця 1.3 – Функціональні вимоги до інтелектуальної системи формування персоналізованого плану тренувань

Ідентифікатор	Назва вимоги	Опис вимоги	Пріоритет
1	2	3	4
FR-01	Реєстрація користувача	Система повинна забезпечувати створення облікового запису користувача із введенням базових персональних даних.	Високий
FR-02	Авторизація користувача	Система повинна забезпечувати безпечний вхід користувача до особистого кабінету.	Високий
FR-03	Заповнення профілю користувача	Система повинна надавати можливість введення віку, статі, ваги, зросту, рівня фізичної підготовки та тренувальної цілі.	Високий
FR-04	Вибір тренувальної мети	Система повинна дозволяти користувачу вибрати основну мету тренувань: схуднення, набір м'язової маси, підтримка форми або розвиток витривалості.	Високий
FR-05	Формування персоналізованого плану тренувань	Система повинна автоматично формувати індивідуальний тренувальний план на основі даних профілю користувача та результатів роботи ML-моделі.	Високий
FR-06	Рекомендація вправ	Система повинна рекомендувати перелік вправ відповідно до рівня підготовки користувача, його цілі та доступного обладнання.	Високий

Продовження таблиці 1.3

1	2	3	4
FR-07	Розрахунок параметрів тренування	Система повинна визначати рекомендовану кількість підходів, повторень, тривалість тренування та рівень інтенсивності.	Високий
FR-08	Перегляд сформованого плану	Система повинна надавати користувачу можливість переглядати сформований тренувальний план у зручному форматі.	Високий
FR-09	Збереження історії тренувань	Система повинна зберігати інформацію про сформовані плани та виконані тренування користувача.	Середній
FR-10	Оновлення рекомендацій	Система повинна оновлювати тренувальні рекомендації після зміни параметрів користувача або його тренувальної цілі.	Високий
FR-11	Перегляд прогресу	Система повинна надавати користувачу можливість переглядати базові показники прогресу.	Середній
FR-12	Адміністрування бази вправ	Система повинна дозволяти адміністратору додавати, редагувати та видаляти вправи в базі даних.	Середній
FR-13	Перевірка коректності введених даних	Система повинна виконувати валідацію введених користувачем даних.	Високий
FR-14	Надання пояснення рекомендацій	Система повинна відображати коротке пояснення, чому користувачу запропоновано відповідний план тренувань.	Середній

Нефункціональні вимоги визначають якісні характеристики програмної системи, зокрема її продуктивність, надійність, зручність використання, безпеку, масштабованість та супроводжуваність. Для інтелектуальної системи планування тренувань важливим є забезпечення швидкої обробки запитів, захисту персональних даних користувачів та стабільної роботи модуля машинного навчання. Крім того, такі вимоги забезпечують відповідність системи сучасним стандартам розробки програмного забезпечення та очікуванням кінцевих користувачів. Нефункціональні вимоги до системи наведено в таблиці 1.4.

Таблиця 1.4 – Нефункціональні вимоги до інтелектуальної системи формування персоналізованого плану тренувань

Ідентифікатор	Категорія вимоги	Опис вимоги	Критерій перевірки
1	2	3	4
NFR-01	Продуктивність	Система повинна формувати персоналізований план тренувань за прийнятний для користувача час.	Час формування плану не повинен перевищувати 5 секунд.
NFR-02	Надійність	Система повинна стабільно працювати під час виконання основних користувацьких сценаріїв.	Відсутність критичних помилок під час тестування основних функцій.
NFR-03	Доступність	Система повинна бути доступною користувачу через вебінтерфейс.	Система відкривається у сучасному веббраузері.
NFR-04	Зручність використання	Інтерфейс системи повинен бути зрозумілим для користувачів без спеціальної технічної підготовки.	Основні дії виконуються без додаткових інструкцій.
NFR-05	Безпека	Система повинна забезпечувати захист облікових записів користувачів.	Паролі зберігаються у захищеному вигляді, доступ до профілю можливий лише після авторизації.
NFR-06	Конфіденційність	Система повинна обмежувати доступ до персональних даних користувача.	Дані профілю доступні лише власнику облікового запису та адміністратору.
NFR-07	Масштабованість	Архітектура системи повинна дозволяти розширення функціональних можливостей.	Можливість додавання нових типів вправ, цілей тренувань та ML-моделей без повної зміни архітектури.
NFR-08	Супроводжуваність	Програмний код повинен бути структурованим і придатним для подальшого розвитку.	Код поділено на логічні модулі, використано зрозумілі назви класів, функцій і змінних.
NFR-09	Сумісність	Система повинна коректно працювати у популярних браузерах.	Підтримка Google Chrome, Mozilla Firefox, Microsoft Edge та Safari.

Продовження таблиці 1.4

1	2	3	4
NFR-10	Точність рекомендацій	Система повинна формувати рекомендації, які відповідають заданим параметрам користувача.	Рекомендовані вправи та навантаження відповідають рівню підготовки, цілі та обмеженням користувача.
NFR-11	Відмовостійкість	У разі помилки система повинна повідомляти користувача про проблему без втрати введених даних.	При некоректному запиті відображається повідомлення про помилку.
NFR-12	Портативність	Система повинна мати можливість розгортання на локальному сервері або хмарній платформі.	Проєкт може бути запущений у стандартному серверному середовищі.

Таким чином, сформовані функціональні та нефункціональні вимоги визначають основні можливості й якісні характеристики майбутньої інтелектуальної системи. Функціональні вимоги описують поведінку системи під час взаємодії з користувачем, тоді як нефункціональні вимоги встановлюють обмеження щодо продуктивності, безпеки, зручності використання та надійності. Надалі ці вимоги будуть використані як основа для проєктування архітектури системи, структури бази даних, модуля машинного навчання та користувацького інтерфейсу [12].

1.5 Визначення сценаріїв використання та користувацьких ролей

Одним із важливих етапів аналізу вимог до програмної системи є визначення користувацьких ролей та сценаріїв взаємодії із системою. Даний етап дозволяє формалізувати функціональні можливості програмного забезпечення, визначити межі відповідальності кожного учасника процесу та встановити перелік дій, які можуть виконуватися в системі. Для цього в інженерії програмного забезпечення широко використовуються діаграми варіантів використання (Use

Case Diagram), що відображають взаємодію між користувачами та функціональними можливостями системи.

Для розроблюваної інтелектуальної системи формування персоналізованого плану тренувань було визначено три основні ролі:

- Користувач — основний учасник системи, який створює обліковий запис, заповнює персональний профіль, вказує тренувальну мету, отримує персоналізовані рекомендації, переглядає сформований план тренувань та відстежує власний прогрес;

- Адміністратор — відповідає за підтримку працездатності системи та адміністрування бази вправ, включаючи додавання, редагування та видалення інформації про тренування;

- ML-модуль — програмний компонент системи, який реалізує алгоритми машинного навчання та забезпечує автоматичне формування рекомендацій на основі даних користувача.

Загальна діаграма варіантів використання для розроблюваної системи наведена на рисунку 1.2.

Як показано на рисунку 1.2, центральним сценарієм роботи системи є формування персоналізованого плану тренувань. Для виконання даного сценарію користувач повинен пройти процедуру реєстрації та авторизації, після чого заповнити профіль із зазначенням особистих характеристик та тренувальної мети.

Сценарій формування персоналізованого плану включає перевірку коректності введених даних, генерацію рекомендацій щодо вправ та розрахунок параметрів тренування. У процесі роботи система використовує можливості модуля машинного навчання для аналізу характеристик користувача та формування індивідуальних рекомендацій.

Після завершення генерації тренувального плану користувач отримує можливість переглянути сформовані рекомендації та зберегти їх для подальшого використання. За необхідності система може виконувати оновлення рекомендацій у випадку зміни параметрів користувача, його фізичної форми або тренувальної мети.

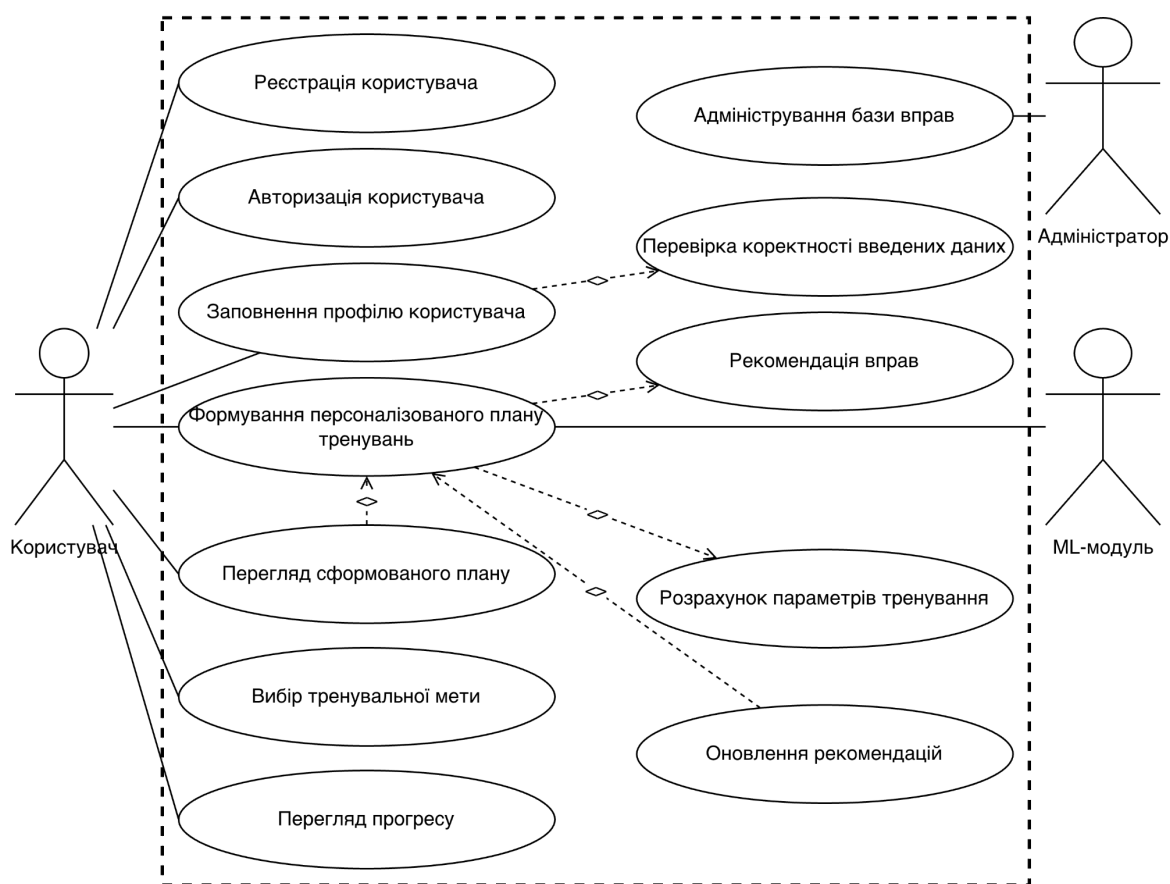


Рисунок 1.2 – Діаграма варіантів використання інтелектуальної системи формування персоналізованого плану тренувань

Окремим напрямом функціонування системи є моніторинг прогресу користувача. Даний сценарій дозволяє відстежувати результати тренувань та використовувати накопичену інформацію для подальшого вдосконалення рекомендацій.

Адміністратор взаємодіє із системою через сценарій адміністрування бази вправ. Він забезпечує актуальність інформації про доступні вправи, їх характеристики та можливі параметри виконання, що безпосередньо впливає на якість рекомендацій, сформованих системою.

Таким чином, визначені користувацькі ролі та сценарії використання дозволяють сформувавши загальне бачення функціонування інтелектуальної системи та встановити взаємозв'язки між її основними компонентами. Отримані

результати є основою для подальшого проєктування архітектури програмного забезпечення та деталізації функціональних компонентів системи.

1.6 Постановка задачі розробки інтелектуальної системи

Проведений аналіз предметної області, існуючих програмних рішень та сучасних методів машинного навчання показав актуальність використання інтелектуальних технологій для автоматизації процесу формування тренувальних програм. Більшість існуючих систем використовують готові шаблони тренувань або обмежені механізми персоналізації, що не дозволяє повною мірою враховувати індивідуальні характеристики користувача та його поточний фізичний стан [13].

У зв'язку з цим виникає задача розробки інтелектуальної системи формування персоналізованого плану тренувань, яка на основі введених користувачем даних та методів машинного навчання буде автоматично генерувати індивідуальні рекомендації щодо фізичної активності. Система повинна враховувати основні характеристики користувача, зокрема вік, стать, вагу, зріст, рівень фізичної підготовки та поставлену тренувальну мету, а також забезпечувати можливість перегляду отриманих рекомендацій і подальшого оновлення тренувального плану.

Об'єктом розробки є процес формування персоналізованих тренувальних програм для користувачів із різним рівнем фізичної підготовки.

Предметом розробки є методи та засоби створення інтелектуальної програмної системи, що використовує алгоритми машинного навчання для автоматичного підбору тренувальних рекомендацій.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз предметної області персоналізованого планування тренувань;
- дослідити існуючі програмні рішення та підходи до персоналізації тренувального процесу;

- проаналізувати сучасні методи машинного навчання, придатні для побудови рекомендаційної системи;
- сформулювати функціональні та нефункціональні вимоги до програмної системи;
- розробити архітектуру інтелектуальної системи формування персоналізованих тренувальних планів;
- спроектувати структуру бази даних для зберігання інформації про користувачів, справи та результати роботи системи;
- реалізувати програмний модуль формування рекомендацій на основі методів машинного навчання;
- розробити користувацький інтерфейс для взаємодії із системою;
- провести тестування програмного забезпечення та оцінити коректність роботи реалізованих функцій;
- виконати аналіз результатів роботи системи та оцінити можливість її практичного використання.

Таким чином, результатом виконання кваліфікаційної роботи повинна стати інтелектуальна програмна система, здатна автоматизувати процес формування персоналізованих тренувальних планів та забезпечити користувачів індивідуальними рекомендаціями на основі аналізу їхніх характеристик і тренувальних цілей.

1.7 Висновки до першого розділу

У першому розділі було проведено аналіз предметної області персоналізованого планування тренувань та досліджено сучасні підходи до формування індивідуальних тренувальних програм. Встановлено, що використання персоналізованих рекомендацій дозволяє підвищити ефективність тренувального процесу, врахувати особливості користувача та забезпечити досягнення поставлених спортивних цілей.

Під час аналізу існуючих програмних рішень було виявлено, що більшість сучасних систем підтримують базову персоналізацію тренувань, проте значна частина з них використовує шаблонні підходи або обмежені механізми адаптації рекомендацій. Водночас застосування методів машинного навчання відкриває можливість автоматичного аналізу даних користувачів та формування більш гнучких і точних рекомендацій.

У роботі було розглянуто основні методи машинного навчання, які можуть бути використані для побудови рекомендаційної системи, зокрема методи класифікації, регресії, кластеризації, колаборативної та контентної фільтрації, а також нейронні мережі. Проведений аналіз показав доцільність використання рекомендаційного підходу на основі методів машинного навчання для задачі формування персоналізованих тренувальних планів.

На основі результатів аналізу були сформовані функціональні та нефункціональні вимоги до програмної системи, визначено основні користувацькі ролі та сценарії взаємодії із системою. Також було сформульовано задачу розробки інтелектуальної системи формування персоналізованого плану тренувань та визначено перелік завдань, необхідних для досягнення поставленої мети.

Отримані результати створюють основу для подальшого проектування архітектури програмної системи, структури бази даних, моделі машинного навчання та програмної реалізації інтелектуальної системи, що буде розглянуто у наступному розділі.

2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

На основі результатів аналізу предметної області, проведеного в попередньому розділі, було визначено основні вимоги до інтелектуальної системи формування персоналізованого плану тренувань. Реалізація таких вимог потребує розроблення програмної архітектури, проєктування структури даних, створення модулів обробки інформації та інтеграції алгоритмів машинного навчання для автоматичного формування рекомендацій.

У даному розділі розглядаються питання проєктування та програмної реалізації інтелектуальної системи. Зокрема, описано архітектуру програмного забезпечення, структуру бази даних, механізми взаємодії між окремими компонентами системи, а також особливості реалізації модуля машинного навчання. Крім того, наведено рішення щодо розробки серверної частини, користувацького інтерфейсу та інтеграції всіх програмних компонентів у єдину систему, здатну формувати персоналізовані тренувальні плани відповідно до характеристик і цілей користувача [14].

2.1 Проєктування архітектури програмної системи

Архітектура програмної системи визначає структуру її компонентів, механізми взаємодії між ними та принципи обробки даних. Грамотно спроектована архітектура забезпечує масштабованість, модифікованість, надійність та зручність подальшого супроводу програмного забезпечення. Для інтелектуальної системи формування персоналізованого плану тренувань особливо важливим є забезпечення ефективної взаємодії між користувацьким інтерфейсом, серверною частиною, базою даних та модулем машинного навчання.

Під час проєктування системи було обрано багаторівневу архітектуру, яка дозволяє розділити логіку представлення даних, бізнес-логіку та рівень зберігання інформації. Такий підхід спрощує розробку окремих компонентів, забезпечує їх

незалежність та створює можливість подальшого розширення функціональності системи [15].

Загальна архітектура основних компонентів розробленої програмної системи наведена на рисунку 2.1.

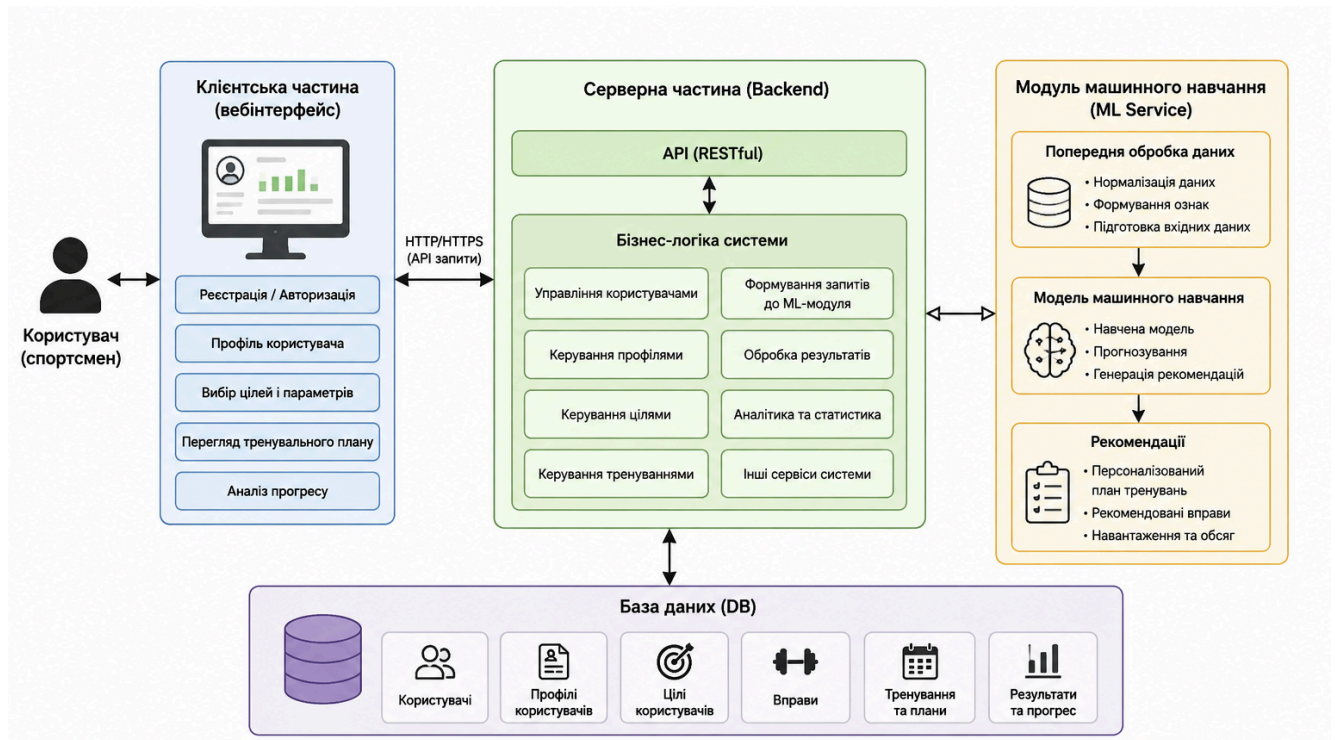


Рисунок 2.1 – Архітектура інтелектуальної системи формування персоналізованого плану тренувань

Як показано на рисунку 2.1, основним користувачем системи є спортсмен або користувач, який взаємодіє з програмою через вебінтерфейс. За допомогою інтерфейсу користувач може створювати обліковий запис, заповнювати персональний профіль, вказувати тренувальні цілі, переглядати сформовані рекомендації та аналізувати власний прогрес [16].

Для детальнішого представлення процесу обробки інформації в системі на рисунку 2.2 наведено схему потоку даних від моменту введення користувачем параметрів до отримання персоналізованих рекомендацій.

Основні програмні компоненти системи та їх функціональне призначення наведено в таблиці 2.1.

Таблиця 2.1 – Основні компоненти архітектури системи

Компонент	Призначення
Вебінтерфейс	Взаємодія користувача із системою
Backend API	Обробка запитів та бізнес-логіка
ML-модуль	Формування персоналізованих рекомендацій
База даних	Зберігання інформації про користувачів та тренування
Модуль аналітики	Аналіз прогресу користувача

Користувацький інтерфейс взаємодіє із серверною частиною через програмний інтерфейс прикладного програмування (API). Серверна частина виконує обробку запитів користувача, реалізує бізнес-логіку системи, забезпечує доступ до даних та координує роботу інших компонентів програмного забезпечення.

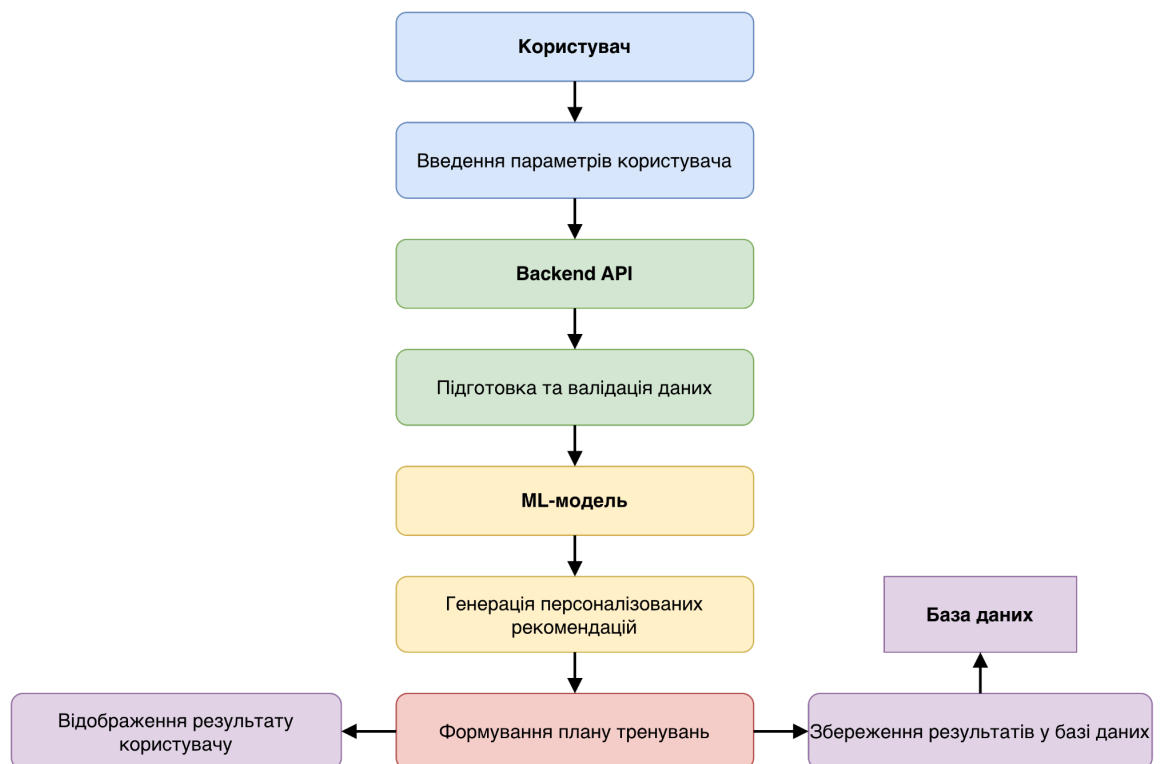


Рисунок 2.2 – Схема потоку даних у системі формування персоналізованого плану тренувань

Одним із ключових елементів архітектури є модуль машинного навчання,

який відповідає за формування персоналізованих рекомендацій. Даний модуль отримує інформацію про користувача, аналізує його характеристики та на основі навченої моделі генерує індивідуальний план тренувань. Результати роботи модуля передаються серверній частині та відображаються користувачу через інтерфейс системи [17].

Для зберігання інформації використовується база даних, у якій розміщуються відомості про користувачів, їхні профілі, тренувальні цілі, вправи, історію тренувань та результати формування рекомендацій. Централізоване зберігання даних забезпечує можливість подальшого аналізу інформації та повторного використання накопичених даних для покращення роботи моделей машинного навчання.

Запропонована архітектура забезпечує чіткий розподіл відповідальності між компонентами системи та дозволяє незалежно вдосконалювати окремі модулі без впливу на інші складові програмного забезпечення. Крім того, використання окремого модуля машинного навчання створює можливість подальшого підвищення точності рекомендацій шляхом заміни або донавчання моделей без зміни основної структури системи [18].

Таким чином, розроблена архітектура забезпечує ефективну взаємодію між усіма компонентами програмної системи та створює основу для реалізації функціональних можливостей інтелектуальної системи формування персоналізованого плану тренувань.

2.2 Проектування структури даних та бази даних

Одним із ключових компонентів інтелектуальної системи формування персоналізованого плану тренувань є підсистема зберігання даних. Від якості проектування структури бази даних залежить ефективність роботи всієї системи, швидкість доступу до інформації, цілісність даних та можливість подальшого розширення функціональних можливостей програмного забезпечення.

Для реалізації системи було обрано реляційну модель зберігання даних. Реляційні бази даних залишаються одним із найбільш поширених рішень для створення інформаційних систем завдяки високій надійності, підтримці механізмів забезпечення цілісності даних, можливості реалізації складних зв'язків між сутностями та ефективній роботі з транзакціями. Як систему керування базами даних було обрано PostgreSQL, яка є сучасною відкритою реляційною СКБД, що забезпечує високу продуктивність, масштабованість та підтримку складних SQL-запитів.

Для відображення структури даних системи було розроблено ER-діаграму, яка демонструє основні сутності бази даних та зв'язки між ними. ER-діаграму бази даних наведено на рисунку 2.3.

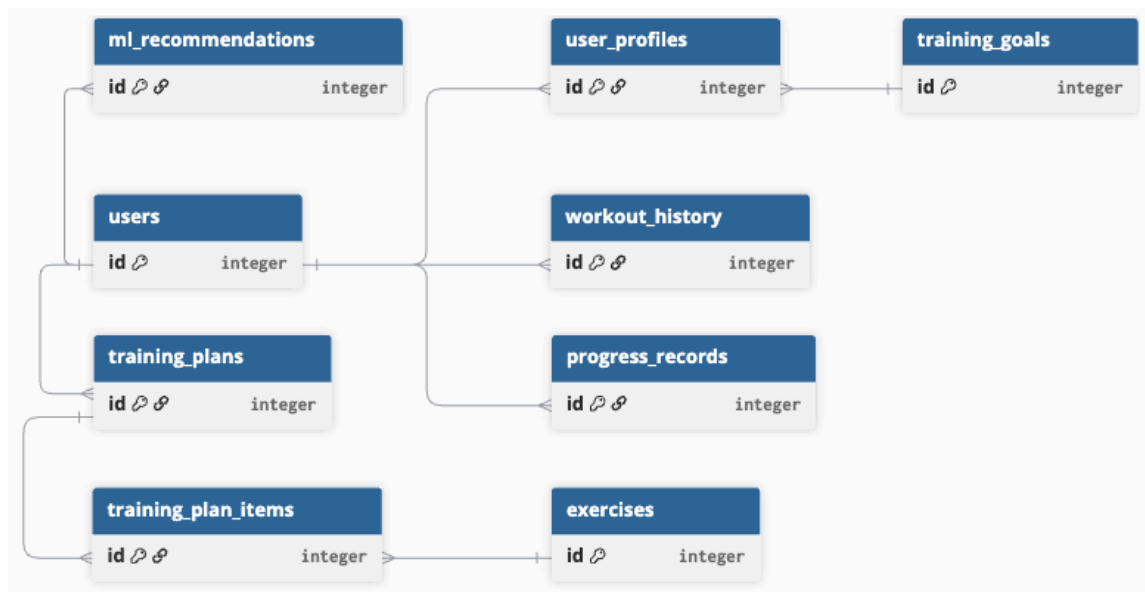


Рисунок 2.3 – ER-діаграма бази даних інтелектуальної системи формування персоналізованого плану тренувань

Основною сутністю системи є користувач, навколо якого формуються всі інші інформаційні об'єкти. Кожен користувач має власний профіль, тренувальні плани, історію тренувань, показники прогресу та результати роботи рекомендаційного модуля [19].

Для детального опису структури бази даних було сформовано перелік основних таблиць та їх призначення. Основні таблиці бази даних наведено в таблиці 2.2.

Таблиця 2.2 – Основні таблиці бази даних системи

Таблиця	Призначення
users	Зберігання облікових записів користувачів системи
user_profiles	Зберігання персональних та фізичних характеристик користувачів
training_goals	Довідник тренувальних цілей
exercises	Довідник вправ та їх характеристик
training_plans	Зберігання сформованих персоналізованих тренувальних планів
training_plan_items	Зберігання вправ, що входять до складу тренувальних планів
workout_history	Історія виконання тренувань користувачами
progress_records	Зберігання показників прогресу користувачів
ml_recommendations	Зберігання результатів роботи модуля машинного навчання

Таблиця users використовується для зберігання інформації про зареєстрованих користувачів системи. Вона містить дані, необхідні для авторизації та ідентифікації користувача, зокрема ім'я, адресу електронної пошти, пароль та роль у системі.

Таблиця user_profiles містить персональні характеристики користувачів, які використовуються під час формування рекомендацій. До таких характеристик належать вік, стать, зріст, вага, рівень фізичної підготовки, досвід тренувань та бажана кількість тренувань на тиждень.

Таблиця training_goals реалізує довідник цілей тренувального процесу. Вона дозволяє стандартизувати можливі цілі користувачів та забезпечує можливість подальшого розширення системи без зміни структури інших таблиць.

Таблиця exercises використовується для зберігання інформації про вправи, які можуть бути рекомендовані користувачам. Для кожної вправи зберігається її назва, опис, рівень складності, цільова група м'язів та необхідне обладнання.

Таблиця `training_plans` містить загальну інформацію про сформовані плани тренувань. Кожен запис відповідає окремому персоналізованому плану, який було сформовано для конкретного користувача.

Таблиця `training_plan_items` забезпечує деталізацію тренувального плану. У ній зберігаються вправи, що входять до складу плану, а також параметри їх виконання: кількість підходів, повторень, тривалість та інтенсивність.

Таблиця `workout_history` використовується для накопичення інформації про виконанні тренування. Вона дозволяє аналізувати активність користувача та використовувати накопичені дані для подальшого вдосконалення рекомендацій.

Таблиця `progress_records` призначена для фіксації змін фізичних показників користувача та результатів тренувального процесу. Інформація з даної таблиці використовується для оцінювання ефективності тренувань та моніторингу досягнення поставлених цілей.

Таблиця `ml_recommendations` зберігає результати роботи модуля машинного навчання. Вона містить сформовані рекомендації, параметри моделі та результати прогнозування, що дозволяє аналізувати ефективність роботи рекомендаційного механізму.

Під час проектування бази даних використовувалися сучасні підходи до розробки інформаційних систем, зокрема нормалізація структури даних, розподіл інформації за логічними сутностями та використання зовнішніх ключів для забезпечення цілісності зв'язків між таблицями. Такий підхід дозволяє мінімізувати дублювання інформації, підвищити ефективність обробки запитів та забезпечити простоту подальшого супроводу системи.

Таким чином, використання реляційної бази даних PostgreSQL та запропонованої структури таблиць забезпечує надійне зберігання інформації, підтримку складних взаємозв'язків між сутностями та створює основу для ефективної роботи інтелектуальної системи формування персоналізованого плану тренувань.

2.3 Розробка модуля підготовки та обробки даних

Ефективність роботи будь-якої системи машинного навчання значною мірою залежить від якості підготовки вхідних даних. Навіть використання сучасних алгоритмів не дозволяє отримати якісні результати у випадку наявності помилок, пропусків або некоректних значень у наборі даних. Тому одним із важливих компонентів розроблюваної інтелектуальної системи є модуль підготовки та обробки даних, який забезпечує перетворення введеної користувачем інформації у формат, придатний для подальшого використання моделлю машинного навчання.

Основним призначенням модуля є автоматизація процесів збору, перевірки, очищення, перетворення та підготовки даних перед їх передачею до рекомендаційного механізму. Загальну послідовність роботи модуля підготовки та обробки даних наведено на рисунку 2.4.



Рисунок 2.4 – Етапи роботи модуля підготовки та обробки даних

Як показано на рисунку 2.4, процес підготовки даних складається з декількох послідовних етапів, кожен з яких виконує окрему функцію в загальному процесі формування персоналізованих рекомендацій [20].

– Збір даних користувача. На даному етапі система отримує від користувача основні параметри, необхідні для формування тренувального плану. До таких параметрів належать вік, стать, зріст, вага, рівень фізичної підготовки, тренувальна мета та бажана кількість тренувань на тиждень. Якість зібраних даних безпосередньо впливає на точність подальших рекомендацій.

– Валідація даних. Після введення інформації виконується перевірка коректності значень. Система контролює наявність обов'язкових полів, допустимі діапазони числових значень та правильність формату введених даних. Даний етап дозволяє уникнути помилок під час подальшої обробки інформації.

– Очищення даних. На цьому етапі усуваються дублікати, некоректні записи та пропущені значення. За необхідності система може використовувати значення за замовчуванням або виконувати автоматичне коригування окремих параметрів.

– Перетворення даних. Введена користувачем інформація переводиться у формат, придатний для роботи алгоритмів машинного навчання. Категоріальні параметри кодуються числовими значеннями, а текстові характеристики перетворюються у відповідні ознаки моделі.

– Нормалізація ознак. Для забезпечення коректної роботи алгоритмів машинного навчання числові параметри приводяться до єдиного масштабу. Це дозволяє уникнути домінування окремих ознак та покращує якість прогнозування.

– Формування вектора ознак. Після завершення попередньої обробки всі підготовлені параметри об'єднуються в єдину структуру даних, яка використовується як вхідний набір ознак для моделі машинного навчання.

– Передача даних до ML-модуля. Сформований вектор ознак передається до рекомендаційного модуля, який виконує прогнозування та визначає найбільш доцільний варіант тренувального плану.

– Збереження результатів обробки. Підготовлені дані та результати

роботи моделі зберігаються у базі даних для подальшого аналізу, формування історії рекомендацій та повторного використання під час навчання моделей.

Запропонований модуль підготовки та обробки даних забезпечує стандартизацію інформації, зменшує вплив помилок введення та підвищує якість роботи рекомендаційної системи. Завдяки використанню сучасних підходів до попередньої обробки даних створюються умови для стабільної роботи моделей машинного навчання та формування більш точних персоналізованих тренувальних рекомендацій.

Для реалізації модуля підготовки та обробки даних було використано мову програмування Python та набір спеціалізованих бібліотек для роботи з даними і машинним навчанням. Основна обробка табличних даних здійснюється за допомогою бібліотеки Pandas, яка забезпечує ефективні засоби для завантаження, очищення, фільтрації та перетворення інформації. Для виконання чисельних обчислень та роботи з багатовимірними масивами використовується бібліотека NumPy. Процеси нормалізації ознак, кодування категоріальних параметрів та підготовки вхідних даних для моделей машинного навчання реалізовано засобами бібліотеки Scikit-learn. Для збереження та отримання інформації з бази даних PostgreSQL використовується ORM-бібліотека SQLAlchemy, що забезпечує зручну взаємодію між програмним кодом та реляційною базою даних. Запропонований набір технологій є одним із найбільш поширених у сучасних системах аналізу даних та машинного навчання, що забезпечує високу продуктивність, масштабованість і зручність подальшого супроводу програмного забезпечення.

2.4 Розробка моделі машинного навчання для формування персоналізованих планів тренувань

Одним із ключових компонентів інтелектуальної системи формування персоналізованого плану тренувань є модель машинного навчання, яка відповідає за автоматичне визначення типу тренувального плану та рівня навантаження для користувача. На відміну від шаблонного підходу, модель машинного навчання дає

змогу враховувати індивідуальні характеристики користувача, його фізичну підготовку, тренувальну мету, доступне обладнання та можливі обмеження.

У межах даної роботи задача формування персоналізованого тренувального плану розглядається як задача класифікації. На основі вхідних параметрів користувача система повинна віднести його до певної категорії тренувальної програми, наприклад: програма для початкового рівня, програма для зниження ваги, програма для набору м'язової маси, програма для розвитку витривалості або програма для підтримки фізичної форми. Такий підхід є достатнім для бакалаврської кваліфікаційної роботи, оскільки дозволяє реалізувати інтелектуальний механізм рекомендацій без надмірного ускладнення архітектури системи.

Для вибору оптимального класифікатора було розглянуто декілька поширених алгоритмів машинного навчання, які можуть бути застосовані для роботи з табличними даними користувачів. Порівняльну характеристику класифікаторів наведено в таблиці 2.3.

Таблиця 2.3 – Порівняльна характеристика класифікаторів для формування персоналізованих тренувальних планів

Класифікатор	Переваги	Недоліки	Доцільність використання
1	2	3	4
Decision Tree	Простота реалізації, зрозуміла логіка прийняття рішень	Схильність до перенавчання, низька стабільність результатів	Доцільний для базового прототипу
Logistic Regression	Швидке навчання, проста інтерпретація результатів	Обмежена здатність працювати зі складними нелінійними залежностями	Доцільна лише для простих залежностей
K-Nearest Neighbors	Простий принцип роботи, не потребує складного навчання	Повільна робота на великих даних, чутливість до масштабу ознак	Обмежено доцільний

Продовження таблиці 2.3

1	2	3	4
Support Vector Machine	Висока точність для окремих задач класифікації	Складність налаштування, нижча інтерпретованість	Доцільний для малих наборів даних
Gradient Boosting	Висока точність прогнозування	Складніше налаштування, більший ризик перенавчання	Доцільний для розширеної реалізації
Random Forest	Висока точність, стійкість до шуму, робота з різними типами ознак, менша схильність до перенавчання	Більша складність порівняно з одним деревом рішень	Найбільш доцільний для даної системи

На основі проведеного порівняння для реалізації моделі машинного навчання було обрано Random Forest Classifier. Даний алгоритм належить до ансамблевих методів машинного навчання та працює шляхом побудови множини дерев рішень. Кожне дерево формує власний прогноз, а остаточний результат визначається шляхом голосування між деревами. Такий підхід дозволяє підвищити стабільність моделі та зменшити ризик помилкової класифікації.

Вибір Random Forest Classifier обумовлений тим, що система працює переважно з табличними даними користувача: віком, зростом, вагою, рівнем фізичної підготовки, тренувальною метою, досвідом занять та доступним обладнанням. Для таких даних Random Forest є ефективним і практичним рішенням, оскільки не потребує складної математичної підготовки ознак, добре працює з нелінійними залежностями та забезпечує достатню точність для задачі персоналізації тренувань.

Окремо в системі передбачено врахування можливих протипоказань або обмежень користувача. Такі обмеження доцільно обробляти не лише засобами машинного навчання, а й за допомогою модуля правил. Це пояснюється тим, що питання безпеки тренувань повинні мати чітку логіку прийняття рішень. Наприклад, при наявності проблем із колінними суглобами система може виключати вправи з високим ударним навантаженням, а при низькому рівні

підготовки — обмежувати інтенсивність тренування. Таким чином, у системі використовується гібридний підхід: модуль правил відповідає за безпечну фільтрацію вправ, а Random Forest Classifier — за персоналізоване визначення типу тренувального плану.

Вхідними даними для моделі машинного навчання є параметри профілю користувача. До них належать:

- вік користувача;
- стать;
- зріст;
- вага;
- рівень фізичної підготовки;
- досвід тренувань;
- кількість бажаних тренувань на тиждень;
- тренувальна мета;
- доступне обладнання;
- наявність обмежень або протипоказань.

Загальну схему роботи моделі машинного навчання наведено на рисунку 2.5.

Як показано на рисунку 2.5, спочатку система отримує параметри користувача та виконує їх попередню обробку. Далі модуль правил перевіряє можливі обмеження та виключає небажані варіанти вправ або надмірні рівні навантаження. Після цього підготовлені дані передаються до Random Forest Classifier, який визначає рекомендований тип тренувального плану. На основі результату класифікації система формує персоналізований план тренувань, що відповідає характеристикам користувача.

Для програмної реалізації моделі було використано мову Python та бібліотеки Scikit-learn, Pandas, NumPy і Joblib. Бібліотека Pandas застосовується для роботи з табличними даними, NumPy — для числових обчислень, Scikit-learn — для створення, навчання та тестування класифікатора Random Forest, а Joblib — для збереження навченої моделі та її повторного використання в системі.



Рисунок 2.5 – Схема роботи моделі машинного навчання для формування персоналізованого тренувального плану

Таким чином, використання Random Forest Classifier у поєднанні з модулем

правил дозволяє реалізувати зрозумілий, безпечний та ефективний механізм формування персоналізованих тренувальних планів. Обраний підхід є доцільним для бакалаврської роботи, оскільки забезпечує баланс між якістю рекомендацій, простотою реалізації, інтерпретованістю результатів та можливістю подальшого розширення системи.

2.5 Реалізація серверної частини системи

Серверна частина системи забезпечує обробку запитів користувачів, взаємодію з базою даних, виконання бізнес-логіки та інтеграцію з модулем машинного навчання. Для реалізації серверної частини було використано платформу Node.js та фреймворк Express.js, які забезпечують високу швидкість обробки HTTP-запитів і дозволяють реалізувати REST API для взаємодії між клієнтською частиною, базою даних та ML-модулем.

Архітектура серверної частини побудована за модульним принципом. Основними компонентами є контролери, сервіси, модуль доступу до даних та модуль інтеграції з системою машинного навчання. Такий підхід забезпечує простоту супроводу програмного забезпечення та можливість подальшого розширення функціональності.

Загальну структуру взаємодії основних компонентів серверної частини наведено на рисунку 2.6.

Основними компонентами серверної частини є:

- UserController — обробка запитів користувача;
- TrainingController — керування тренувальними планами;
- RecommendationService — формування персоналізованих рекомендацій;
- MLService — взаємодія з моделлю Random Forest;
- DatabaseService — робота з базою даних PostgreSQL;
- UserRepository — доступ до даних користувачів;
- TrainingRepository — доступ до даних тренувань;

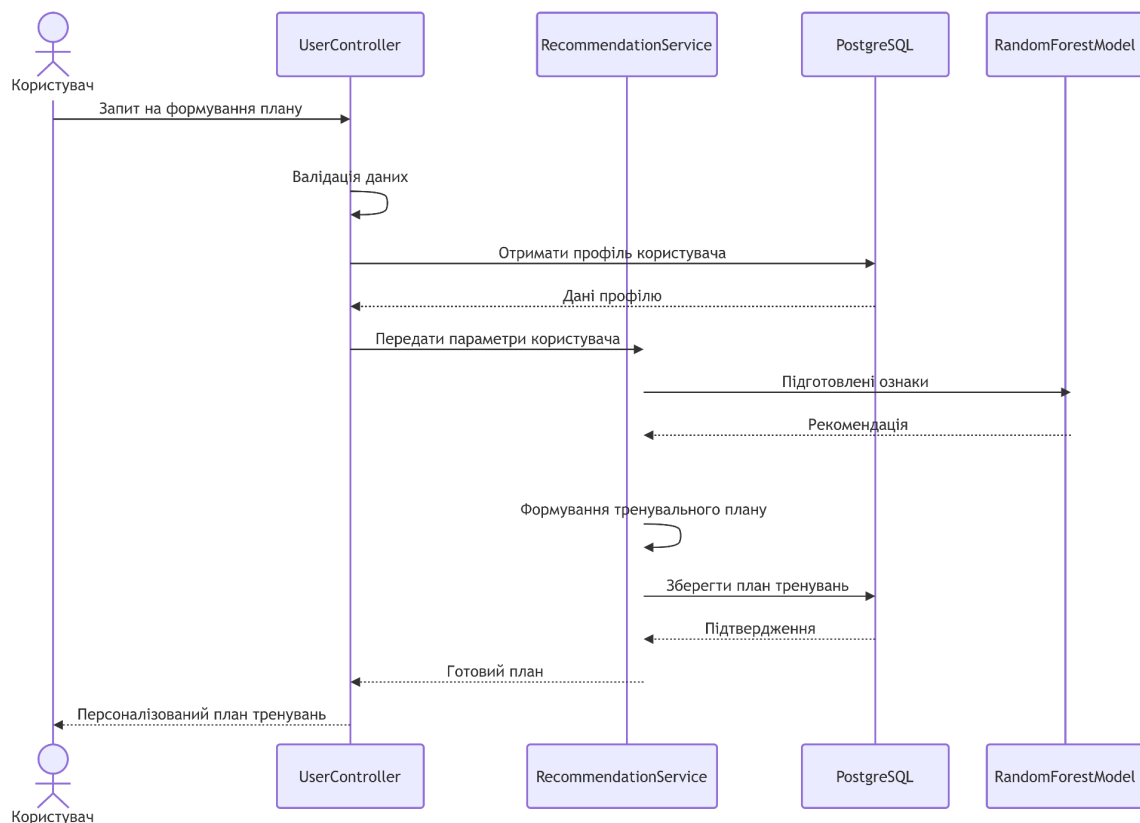


Рисунок 2.7 – Діаграма послідовності формування тренувального плану

Процес роботи серверної частини складається з таких кроків:

1. Користувач надсилає запит на формування тренувального плану.
2. Контролер отримує та перевіряє вхідні дані.
3. Сервіс отримує профіль користувача з бази даних.
4. Дані передаються до ML-модуля.
5. Модель Random Forest формує рекомендацію.
6. Сервер створює персоналізований тренувальний план.
7. Результат зберігається в базі даних.
8. Готовий план повертається користувачу через API.

Реалізована серверна частина забезпечує централізовану обробку бізнес-логіки системи, взаємодію з базою даних та інтеграцію алгоритмів машинного навчання. Використання архітектурного підходу на основі REST API дозволяє легко масштабувати систему та забезпечує незалежність клієнтської частини від внутрішньої логіки сервера.

2.6 Розробка користувацького інтерфейсу

Користувацький інтерфейс є важливою складовою програмної системи, оскільки забезпечує взаємодію користувача з функціональними можливостями застосунку. Під час проєктування інтерфейсу основну увагу було приділено простоті використання, зрозумілій навігації та мінімізації кількості дій, необхідних для отримання персоналізованого тренувального плану.

Оскільки розроблювана система орієнтована на широке коло користувачів із різним рівнем підготовки, інтерфейс було реалізовано відповідно до принципів мінімалізму та зручності використання. Усі основні функції системи зосереджені у декількох логічно пов'язаних екранах, що дозволяє користувачу швидко пройти весь процес формування та перегляду персоналізованого плану тренувань.

Першим етапом роботи із системою є заповнення профілю користувача. На даному екрані користувач вводить персональні дані, які надалі використовуються модулем машинного навчання для формування рекомендацій. Форма містить основні параметри, необхідні для роботи системи, зокрема вік, стать, зріст, вагу, рівень фізичної підготовки, тренувальну мету, бажану кількість тренувань на тиждень та можливі обмеження або протипоказання. Інтерфейс заповнення профілю користувача наведено на рисунку 2.8.

Після введення необхідних параметрів користувач може перейти до формування персоналізованого тренувального плану. На даному етапі система виконує обробку введених даних, передає їх до моделі машинного навчання та відображає отримані результати. Користувач отримує коротку інформацію про сформовану рекомендацію, яка містить тип тренувального плану, рекомендований рівень навантаження та кількість тренувань на тиждень. Інтерфейс формування персоналізованого тренувального плану наведено на рисунку 2.9.

Профіль користувача
Заповніть інформацію про себе для створення персоналізованого плану тренувань

Основна інформація

Вік: 25 | Стать: Чоловік | Зріст (см): 180

Вага (кг): 80 | Рівень фізичної підготовки: Середній

Тренувальні параметри

Ціль тренувань: Набір м'язової маси | Кількість тренувань на тиждень: 4

Додаткова інформація

Доступне обладнання: Тренажерний зал | Обмеження / протипоказання: Відсутні

[Зберегти профіль](#)

Рисунок 2.8 – Інтерфейс заповнення профілю користувача

Формування персонального плану
Натисніть кнопку, щоб наша система на основі ваших параметрів сформувала оптимальний тренувальний план за допомогою моделі Random Forest.

[Сформувати план](#)

Ваші параметри

Вік	25	Ціль тренувань	Набір м'язової маси
Стать	Чоловік	Тренувань на тиждень	4
Зріст	180 см	Обладнання	Тренажерний зал
Вага	80 кг	Обмеження	Відсутні
Рівень підготовки	Середній		

Результат моделі Random Forest

<p>Тип плану</p> <p>Силовий</p> <p>План спрямований на набір м'язової маси та збільшення сили.</p>	<p>Рівень навантаження</p> <p>Середній</p> <p>Оптимальне навантаження для вашого рівня підготовки.</p>	<p>Частота тренувань</p> <p>4 рази на тиждень</p> <p>Рекомендована частота для досягнення вашої цілі.</p>
-----------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------

Рисунок 2.9 – Інтерфейс формування персоналізованого тренувального плану

Після завершення формування рекомендацій користувач отримує доступ до готового тренувального плану. На даному екрані відображається перелік вправ, розподілених за тренувальними днями, а також основні параметри виконання кожної вправи, зокрема кількість підходів, повторень та рівень інтенсивності. Крім того, система дозволяє переглядати поточний прогрес виконання тренувальної програми, що дає можливість оцінювати ефективність занять та контролювати досягнення поставленої мети. Інтерфейс перегляду тренувального плану та прогресу користувача наведено на рисунку 2.10.

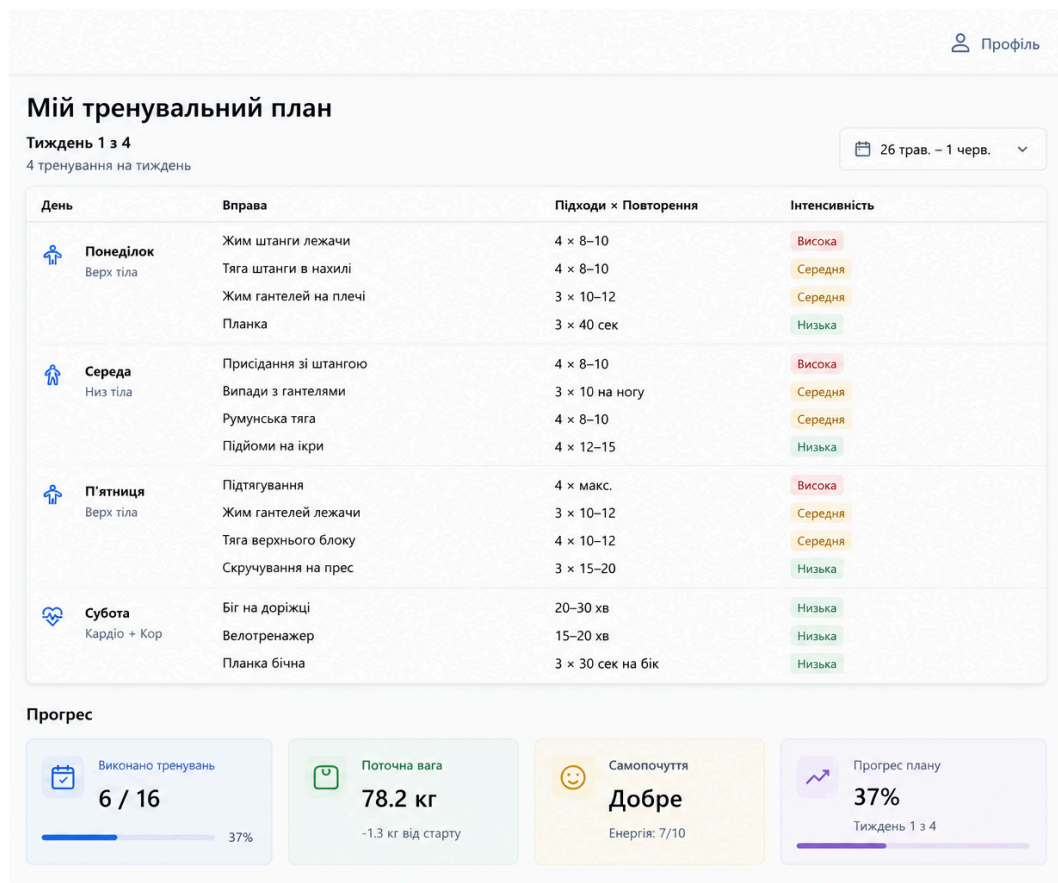


Рисунок 2.10 – Інтерфейс перегляду тренувального плану та прогресу користувача

Під час реалізації користувацького інтерфейсу було використано сучасні вебтехнології HTML5, CSS3 та JavaScript. Для побудови компонентів інтерфейсу застосовувався фреймворк React, який забезпечує створення інтерактивних

користувацьких елементів та ефективно оновлення даних без перезавантаження сторінки. Взаємодія з серверною частиною реалізована через REST API із використанням асинхронних HTTP-запитів.

Таким чином, розроблений користувацький інтерфейс забезпечує зручну взаємодію із системою на всіх етапах її використання: від введення персональних параметрів до отримання готового тренувального плану та контролю прогресу користувача. Запропоноване рішення відповідає вимогам простоти використання, наочності представлення інформації та забезпечує комфортну роботу з інтелектуальною системою формування персоналізованих планів тренувань.

2.7 Висновки до розділу 2

У другому розділі було виконано проектування та розробку основних компонентів інтелектуальної системи формування персоналізованого плану тренувань. Розроблено архітектуру програмної системи, визначено структуру взаємодії між клієнтською частиною, сервером, базою даних та модулем машинного навчання. Спроектовано структуру реляційної бази даних PostgreSQL, яка забезпечує зберігання інформації про користувачів, тренувальні плани, історію тренувань та результати роботи рекомендаційного модуля.

Також було розроблено модуль підготовки та обробки даних, обрано та обґрунтовано використання алгоритму Random Forest Classifier для класифікації користувачів за типом тренувального плану та рівнем навантаження. Реалізовано серверну частину системи на основі REST API та спроектовано користувацький інтерфейс, який забезпечує зручне введення даних, формування рекомендацій та перегляд отриманого тренувального плану. Отримані результати створюють основу для практичної реалізації та подальшого тестування розробленої інтелектуальної системи.

3 ТЕСТУВАННЯ, ВПРОВАДЖЕННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ

Якість роботи моделі машинного навчання значною мірою залежить від якості набору даних, який використовується для її навчання та тестування. Для розробки інтелектуальної системи формування персоналізованого плану тренувань було сформовано набір даних, що містить інформацію про користувачів, їх фізичні характеристики, рівень підготовки та тренувальні цілі. Дані використовуються для навчання моделі Random Forest Classifier, яка визначає найбільш доцільний тип тренувального плану для конкретного користувача.

3.1 Формування набору даних для навчання та тестування моделі

Якість роботи моделі машинного навчання значною мірою залежить від якості набору даних, який використовується для її навчання та тестування. Для розробки інтелектуальної системи формування персоналізованого плану тренувань було сформовано набір даних, що містить інформацію про користувачів, їх фізичні характеристики, рівень підготовки та тренувальні цілі. Дані використовуються для навчання моделі Random Forest Classifier, яка визначає найбільш доцільний тип тренувального плану для конкретного користувача.

Для навчання та тестування моделі машинного навчання було використано відкритий набір даних «Fitness Exercises Using BFP & BMI», розміщений на платформі Kaggle. Даний датасет містить інформацію про фізичні характеристики користувачів, показники фізичної підготовки, рівень активності та тренувальні цілі. Крім того, набір даних містить інформацію про рекомендовані типи фізичних навантажень, що дозволяє використовувати його для побудови систем рекомендацій у сфері фітнесу та персоналізованих тренувань.

Для потреб розроблюваної системи було виконано адаптацію структури набору даних. Із вихідного датасету були відібрані параметри, які безпосередньо впливають на формування персоналізованого тренувального плану, а також

сформовано цільову ознаку, що використовується під час навчання моделі Random Forest Classifier. Отриманий набір даних дозволяє моделі встановлювати взаємозв'язок між характеристиками користувача та рекомендованим типом тренувальної програми.

Набір даних містить параметри, які безпосередньо впливають на процес формування тренувальних рекомендацій. До таких параметрів належать вік, стать, зріст, вага, рівень фізичної підготовки, досвід тренувань, кількість бажаних тренувань на тиждень, тренувальна мета, доступне обладнання та наявність протипоказань або обмежень.

Загальну структуру набору даних наведено в таблиці 3.1.

Таблиця 3.1 – Структура набору даних для навчання моделі

Ознака	Тип даних	Опис
age	Числовий	Вік користувача
gender	Категоріальний	Стать користувача
height_cm	Числовий	Зріст користувача
weight_kg	Числовий	Вага користувача
fitness_level	Категоріальний	Рівень фізичної підготовки
experience	Категоріальний	Досвід тренувань
workouts_per_week	Числовий	Кількість тренувань на тиждень
goal	Категоріальний	Тренувальна мета
equipment	Категоріальний	Доступне обладнання
limitations	Категоріальний	Наявність обмежень
plan_class	Категоріальний	Цільова змінна (клас плану)

Перед навчанням моделі було виконано попередню підготовку даних. Вона включала перевірку коректності значень, усунення пропущених даних, кодування категоріальних ознак та приведення числових параметрів до формату, придатного для використання алгоритмами машинного навчання. Після завершення обробки кожен запис набору даних перетворювався у вектор ознак, який подавався на вхід моделі Random Forest.

Приклад записів набору даних наведено в таблиці 3.2.

Таблиця 3.2 – Приклад даних для навчання моделі

Вік	Стать	Рівень підготовки	Ціль	Тренувань /тиждень	Клас плану
24	Чоловік	Середній	Набір маси	4	Muscle_Gain_Intermediate
31	Жінка	Початковий	Схуднення	3	Weight_Loss_Beginner
27	Чоловік	Просунутий	Витривалість	5	Endurance_Advanced
22	Жінка	Початковий	Підтримка форми	2	Fitness_Beginner

Перед навчанням моделі було виконано попередню підготовку даних, яка включала перевірку коректності значень, очищення даних, кодування категоріальних ознак та формування вектора ознак. Детальний опис етапів підготовки та обробки даних наведено у підрозділі 2.3, де розглянуто структуру відповідного програмного модуля та послідовність перетворення вхідної інформації.

Для оцінювання ефективності моделі набір даних було поділено на навчальну та тестову вибірки. Навчальна вибірка використовувалася для побудови моделі Random Forest Classifier, а тестова – для перевірки її здатності коректно працювати на нових даних. У роботі використано співвідношення 80% даних для навчання та 20% для тестування, що є одним із найбільш поширених підходів у задачах класифікації.

Таким чином, сформований набір даних містить необхідний перелік характеристик користувачів та забезпечує можливість навчання і подальшого оцінювання якості роботи моделі машинного навчання. Отримана вибірка дозволяє перевірити здатність системи формувати персоналізовані тренувальні рекомендації відповідно до індивідуальних параметрів користувачів.

3.2 Оцінювання ефективності та результати роботи моделі машинного навчання

Після завершення навчання моделі Random Forest Classifier було проведено оцінювання її ефективності на тестовій вибірці. Для оцінювання якості

класифікації використовувалися стандартні метрики машинного навчання: Accuracy, Precision, Recall та F1-score. Дані показники дозволяють комплексно оцінити здатність моделі коректно визначати рекомендований тип тренувального плану на основі характеристик користувача.

Навчання моделі виконувалося на 80% даних сформованої вибірки, тоді як решта 20% використовувалися для тестування. Для побудови моделі використовувався алгоритм Random Forest Classifier із параметрами, наведеними в таблиці 3.3.

Таблиця 3.3 – Основні параметри моделі Random Forest

Параметр	Значення
Кількість дерев (n_estimators)	100
Максимальна глибина дерева (max_depth)	10
Мінімальна кількість зразків у вузлі (min_samples_split)	2
Мінімальна кількість зразків у листі (min_samples_leaf)	1
Критерій розбиття	Gini
Test Size	20 %

Результати оцінювання моделі наведено в таблиці 3.4.

Таблиця 3.4 – Результати оцінювання моделі

Метрика	Значення
Accuracy	0,89
Precision	0,88
Recall	0,87
F1-score	0,88

Отримані результати свідчать про достатньо високу якість класифікації користувачів за типами тренувальних програм. Значення Accuracy на рівні 89% демонструє, що модель правильно визначає рекомендований тип тренувального плану для більшості користувачів тестової вибірки. Водночас близькі значення

Precision, Recall та F1-score свідчать про збалансованість моделі та відсутність суттєвих проблем із перенавчанням.

Отримані результати підтверджують доцільність використання алгоритму Random Forest Classifier для задачі формування персоналізованих тренувальних рекомендацій. Модель забезпечує достатній рівень точності, стабільно працює на тестових даних та може бути використана як основа для рекомендаційного механізму розробленої програмної системи.

3.3 Тестування функціональних можливостей та аналіз якості роботи програмної системи

Після завершення розробки програмної системи було проведено комплексне тестування її функціональних можливостей та оцінювання якості сформованих тренувальних рекомендацій. Основною метою тестування була перевірка коректності роботи окремих модулів системи, взаємодії між компонентами, а також оцінювання здатності системи формувати персоналізовані тренувальні плани відповідно до характеристик користувача.

Тестування виконувалося на рівні основних функціональних компонентів системи: модуля керування користувачами, модуля підготовки даних, рекомендаційного модуля на основі Random Forest, модуля генерації тренувальних планів та користувацького інтерфейсу. Результати проведеного тестування наведено в таблиці 3.5.

Отримані результати свідчать про високу надійність розробленої програмної системи. Із 90 проведених тестових сценаріїв успішно виконано 88, що відповідає загальному рівню успішності 97,8 %. Виявлені недоліки не впливають на основну функціональність системи та можуть бути усунуті в процесі подальшого вдосконалення програмного забезпечення.

Таблиця 3.5 – Результати функціонального тестування системи

Компонент системи	Кількість тестів	Успішно пройдено	Успішність, %
Реєстрація та авторизація користувачів	8	8	100
Керування профілем користувача	10	10	100
Підготовка та обробка даних	12	12	100
Робота моделі Random Forest	15	14	93,3
Генерація тренувальних планів	12	12	100
Збереження даних у базі даних	8	8	100
REST API	10	10	100
Користувацький інтерфейс	15	14	93,3
Загалом	90	88	97,8

Окрему увагу було приділено оцінюванню якості сформованих тренувальних планів. Для цього було виконано серію тестових запусків системи для користувачів із різними характеристиками та тренувальними цілями. Аналіз результатів показав, що сформовані рекомендації відповідають заданим параметрам користувачів та враховують рівень фізичної підготовки, бажану кількість тренувань і поставлену тренувальну мету.

Для оцінювання якості сформованих рекомендацій було використано експертний аналіз отриманих результатів. Основні результати оцінювання наведено в таблиці 3.6.

Таблиця 3.6 – Результати оцінювання якості тренувальних рекомендацій

Показник	Значення
Відповідність тренувальній меті	92 %
Відповідність рівню підготовки	89 %
Коректність підбору навантаження	90 %
Узгодженість структури плану	94 %
Середня оцінка якості рекомендацій	91,25 %

Результати оцінювання показують, що система забезпечує достатньо високий рівень персоналізації сформованих тренувальних програм. Найкращі

результати отримано за показником узгодженості структури тренувального плану, тоді як незначні відхилення спостерігаються під час визначення оптимального рівня навантаження для користувачів із близькими характеристиками.

Таким чином, проведене тестування підтвердило працездатність усіх основних компонентів програмної системи та ефективність використання алгоритму Random Forest для формування персоналізованих тренувальних рекомендацій. Отримані результати свідчать про можливість практичного використання розробленої системи для автоматизованого формування індивідуальних тренувальних планів.

3.4 Впровадження системи та рекомендації щодо використання

Розроблена інтелектуальна система може бути використана як програмний засіб підтримки прийняття рішень під час формування персоналізованих тренувальних програм. Система орієнтована на користувачів різного рівня фізичної підготовки та може застосовуватися у фітнес-клубах, спортивних центрах, навчальних закладах або для індивідуального використання.

Для впровадження системи необхідно розгорнути серверну частину на вебсервері або хмарній платформі, налаштувати базу даних PostgreSQL та забезпечити доступ користувачів через вебінтерфейс. Навчена модель Random Forest завантажується під час запуску серверної частини та використовується для автоматичного формування рекомендацій у режимі реального часу.

Під час практичного використання рекомендується періодично оновлювати набір даних для навчання моделі, а також розширювати базу вправ та тренувальних шаблонів. Це дозволить підвищити якість персоналізації рекомендацій та адаптувати систему до ширшого кола користувачів. Перспективним напрямом подальшого розвитку є інтеграція з мобільними застосунками та носимими пристроями для автоматичного збору даних про фізичну активність користувачів.

3.5 Висновки до розділу 3

У третьому розділі було сформовано набір даних для навчання та тестування моделі машинного навчання, виконано оцінювання ефективності алгоритму Random Forest Classifier та проаналізовано результати його роботи. Отримані значення метрик Accuracy, Precision, Recall та F1-score підтвердили доцільність використання обраного алгоритму для задачі формування персоналізованих тренувальних рекомендацій.

Також було проведено тестування функціональних можливостей програмної системи та оцінювання якості сформованих тренувальних планів. Результати тестування підтвердили коректність роботи основних компонентів системи, а сформовані рекомендації продемонстрували високий рівень відповідності характеристикам користувачів та поставленим тренувальним цілям. Отримані результати свідчать про готовність розробленої системи до практичного використання та можливість її подальшого розвитку.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

У даному розділі розглядаються основні вимоги охорони праці та безпеки життєдіяльності, яких необхідно дотримуватися під час виконання робіт з використанням комп'ютерної техніки. Особливу увагу приділено організації робочого місця та забезпеченню безпечних умов праці відповідно до чинних нормативних документів [21].

4.1 Інженерно-психологічні принципи професійного добору

Під час розробки та експлуатації програмних систем важливе значення має врахування інженерно-психологічних принципів професійного добору працівників. Інженерна психологія вивчає особливості взаємодії людини з технічними системами та спрямована на забезпечення безпечної, ефективної й комфортної діяльності оператора. У свою чергу, безпека життєдіяльності розглядає умови, за яких професійна діяльність не створює загроз для здоров'я та працездатності людини [22].

У контексті розробки інтелектуальної системи формування персоналізованого плану тренувань основним працівником виступає розробник програмного забезпечення, діяльність якого пов'язана з тривалою роботою за комп'ютером, аналізом інформації, проектуванням програмних компонентів та прийняттям технічних рішень. Відповідно до вимог безпеки життєдіяльності та охорони праці, ефективність такої діяльності значною мірою залежить від професійно важливих психофізіологічних характеристик працівника [21].

Основною метою професійного добору є забезпечення відповідності індивідуальних можливостей працівника вимогам конкретної діяльності. З точки зору безпеки життєдіяльності це дозволяє знизити ризик професійних помилок, підвищити продуктивність праці та мінімізувати негативний вплив психоемоційних навантажень на здоров'я людини [24].

Для фахівців у сфері програмної інженерії особливого значення набувають такі професійно важливі якості:

- здатність до тривалої концентрації уваги;
- високий рівень логічного та аналітичного мислення;
- розвинена оперативна пам'ять;
- здатність швидко обробляти великі обсяги інформації;
- стійкість до монотонної роботи;
- емоційна врівноваженість;
- відповідальність за прийняття рішень;
- здатність працювати в умовах обмеженого часу;
- навички командної взаємодії та професійної комунікації.

З позиції безпеки життєдіяльності недостатній рівень розвитку зазначених якостей може призводити до виникнення помилок під час розробки програмного забезпечення, збільшення психоемоційного напруження, перевтоми та професійного вигорання. Безпека життєдіяльності передбачає своєчасне виявлення таких ризиків і врахування їх під час професійного добору. Особливо актуальним це є під час виконання складних завдань, пов'язаних із використанням алгоритмів машинного навчання, аналізом даних та тестуванням програмних компонентів [22].

Важливим інженерно-психологічним принципом є узгодження інформаційного навантаження з можливостями людини. Під час розробки програмного забезпечення спеціаліст одночасно працює з великою кількістю інформаційних об'єктів: програмним кодом, документацією, базами даних, результатами тестування та іншими технічними матеріалами. З точки зору безпеки життєдіяльності перевищення допустимого рівня інформаційного навантаження може спричиняти зниження уваги, збільшення кількості помилок та погіршення загального психофізіологічного стану працівника [24].

З метою забезпечення безпечної професійної діяльності розробника програмного забезпечення та дотримання принципів безпеки життєдіяльності доцільно дотримуватись таких принципів професійного добору:

- відповідність рівня кваліфікації складності виконуваних завдань;
- врахування психофізіологічних особливостей працівника;
- оцінювання стійкості до розумових навантажень;
- перевірка здатності до тривалої концентрації уваги;
- оцінювання навичок роботи з інформаційними системами;
- визначення рівня комунікативних та організаційних здібностей;
- оцінювання готовності до безперервного професійного навчання.

Законодавство України у сфері охорони праці визначає необхідність створення безпечних умов праці та врахування людського фактора під час організації трудової діяльності [23]. Безпека життєдіяльності є важливою складовою цього процесу, оскільки спрямована на збереження життя, здоров'я та працездатності працівників. У сучасних умовах цифровізації особливого значення набуває забезпечення відповідності між вимогами професійної діяльності та психофізіологічними можливостями працівника, що є одним із ключових принципів профілактики професійних ризиків [25].

Таким чином, інженерно-психологічні принципи професійного добору спрямовані на забезпечення ефективної та безпечної діяльності фахівців, які працюють із програмними системами. Безпека життєдіяльності виступає важливим критерієм оцінювання умов праці та професійної придатності працівника. Для розробника інтелектуальної системи формування персоналізованого плану тренувань дотримання зазначених принципів дозволяє знизити ймовірність помилок, підвищити продуктивність праці та забезпечити належний рівень безпеки життєдіяльності під час виконання професійних обов'язків [21–25].

4.2 Інженерно-технічні рішення з охорони праці

Під час розробки інтелектуальної системи формування персоналізованого плану тренувань основним робочим середовищем є автоматизоване робоче місце розробника програмного забезпечення. Відповідно до вимог охорони праці,

організація такого робочого місця повинна забезпечувати безпечні та комфортні умови праці, запобігати виникненню професійних захворювань, зменшувати вплив шкідливих виробничих факторів та сприяти збереженню працездатності працівника [23].

Інженерно-технічні рішення з охорони праці являють собою комплекс організаційних та технічних заходів, спрямованих на усунення або мінімізацію небезпечних і шкідливих виробничих факторів. Для працівників ІТ-сфери основними факторами ризику є тривала робота за комп'ютером, статичне навантаження на опорно-руховий апарат, напруження органів зору, недостатня рухова активність, а також психоемоційне навантаження [22].

З метою забезпечення вимог охорони праці робоче місце розробника повинно відповідати таким вимогам:

- робочий стіл повинен забезпечувати достатню площу для розміщення комп'ютерної техніки та допоміжного обладнання;
- конструкція робочого крісла повинна передбачати регулювання висоти сидіння, нахилу спинки та підтримку поперекового відділу хребта;
- монітор необхідно розташовувати на відстані 50–70 см від очей користувача;
- верхня межа екрана повинна знаходитися на рівні очей або трохи нижче;
- клавіатура та маніпулятор типу «миша» повинні розташовуватись таким чином, щоб забезпечувати природне положення кистей рук;
- повинна бути забезпечена можливість зміни робочої пози протягом робочого дня [24].

Особливу увагу в охороні праці приділяють забезпеченню нормативних параметрів виробничого середовища. Для приміщень, у яких експлуатується комп'ютерна техніка, необхідно підтримувати:

- температуру повітря в межах 20–24 °С;
- відносну вологість повітря 40–60 %;
- швидкість руху повітря не більше 0,1–0,2 м/с;

– рівень шуму, що не перевищує встановлених нормативних значень [24].

Важливим напрямом охорони праці є забезпечення безпечної експлуатації електрообладнання. У процесі розробки програмного забезпечення використовуються персональні комп'ютери, монітори, мережеве обладнання та інші електронні пристрої, тому необхідно передбачити:

- використання справних електромереж та сертифікованого обладнання;
- наявність захисного заземлення електрообладнання;
- застосування автоматичних вимикачів захисту;
- своєчасне проведення технічного обслуговування обладнання;
- недопущення експлуатації пошкоджених кабелів та розеток [23].

Одним із ключових завдань охорони праці є забезпечення раціонального режиму праці та відпочинку. Тривала робота за комп'ютером призводить до зорового стомлення та зниження працездатності працівника. Для зменшення негативного впливу рекомендується організовувати регламентовані перерви, виконувати вправи для очей та періодично змінювати характер діяльності [22].

Для забезпечення вимог пожежної безпеки в приміщенні необхідно передбачити:

- наявність первинних засобів пожежогасіння;
- вільний доступ до евакуаційних виходів;
- використання справної електромережі;
- дотримання правил експлуатації електрообладнання;
- проведення інструктажів з питань пожежної безпеки та охорони праці [23].

Важливим елементом системи охорони праці є проведення інструктажів та навчання працівників. Кожен працівник повинен бути ознайомлений із вимогами охорони праці, правилами безпечної роботи з комп'ютерною технікою, порядком дій у разі виникнення аварійних ситуацій та заходами пожежної безпеки [25].

Таким чином, інженерно-технічні рішення з охорони праці для розробника інтелектуальної системи формування персоналізованого плану тренувань

спрямовані на створення безпечних умов праці, збереження здоров'я працівників та зниження рівня професійних ризиків. Дотримання вимог охорони праці під час організації робочого місця, експлуатації обладнання та планування режиму праці забезпечує належний рівень безпеки праці та підвищує ефективність професійної діяльності [21–25].

4.3 Висновки до четвертого розділу

У процесі виконання кваліфікаційної роботи було розглянуто питання безпеки життєдіяльності та охорони праці, що стосуються професійної діяльності розробника програмного забезпечення. Проведений аналіз інженерно-психологічних принципів професійного добору дозволив визначити основні психофізіологічні вимоги до фахівців, діяльність яких пов'язана з розробкою програмних систем. Встановлено, що врахування людського фактора, раціональна організація трудового процесу та дотримання вимог безпеки життєдіяльності сприяють підвищенню ефективності роботи та зниженню професійних ризиків.

Також були розглянуті інженерно-технічні рішення з охорони праці, спрямовані на забезпечення безпечних умов праці під час роботи з комп'ютерною технікою. У межах розробки інтелектуальної системи формування персоналізованого плану тренувань враховано вимоги охорони праці щодо організації робочого місця, безпечної експлуатації обладнання, дотримання санітарно-гігієнічних норм та режимів праці й відпочинку. Це дозволяє забезпечити належний рівень безпеки праці, збереження здоров'я працівників та відповідність розробленого програмного продукту сучасним вимогам безпеки життєдіяльності та охорони праці.

ВИСНОВКИ

У кваліфікаційній роботі розв'язано актуальну задачу розробки інтелектуальної системи формування персоналізованого плану тренувань на основі методів машинного навчання. Актуальність роботи обумовлена необхідністю автоматизації процесу підбору тренувальних програм з урахуванням індивідуальних характеристик користувачів, рівня їх фізичної підготовки та поставлених тренувальних цілей. Проведено аналіз предметної області, сучасних підходів до персоналізації тренувань, існуючих програмних рішень та методів машинного навчання для побудови рекомендаційних систем. На основі проведеного аналізу сформовано функціональні та нефункціональні вимоги до програмної системи, визначено її основні ролі, сценарії використання та структуру.

У роботі спроектовано архітектуру програмної системи, структуру реляційної бази даних та механізм взаємодії між її основними компонентами. Розроблено модуль підготовки та обробки даних, серверну частину системи на основі REST API та користувацький інтерфейс, який забезпечує введення параметрів користувача, формування рекомендацій та перегляд готових тренувальних планів. Для зберігання даних використано систему керування базами даних PostgreSQL, а для реалізації рекомендаційного механізму обрано алгоритм Random Forest Classifier, який забезпечує класифікацію користувачів за типом тренувальної програми та рекомендованим рівнем навантаження.

Для навчання та тестування моделі використано відкритий набір даних із платформи Kaggle, адаптований до потреб розроблюваної системи. За результатами експериментальних досліджень модель Random Forest продемонструвала значення Accuracy = 89 %, Precision = 88 %, Recall = 87 % та F1-score = 88 %, що свідчить про достатньо високу якість класифікації користувачів. Також було проведено функціональне тестування програмної системи, під час якого виконано 90 тестових сценаріїв, з яких успішно пройдено 88. Загальний рівень успішності тестування склав 97,8 %, що підтверджує

коректність роботи основних модулів та стабільність функціонування програмного забезпечення.

Практичне значення роботи полягає у створенні програмної системи, яка дозволяє автоматизувати процес формування персоналізованих тренувальних рекомендацій та підвищити ефективність підбору тренувальних програм для користувачів різного рівня підготовки. Результати проведених досліджень підтвердили доцільність використання методів машинного навчання для задач персоналізації тренувань, а розроблена система продемонструвала достатній рівень точності, функціональності та готовності до практичного використання. Таким чином, мету кваліфікаційної роботи досягнуто, а всі поставлені завдання виконано в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Михалик Д. М., Цуприк Г. Б., Бревус В. М. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за освітньо-професійною програмою «Інженерія програмного забезпечення» спеціальності 121 – «Інженерія програмного забезпечення» всіх форм навчання. Тернопіль : ТНТУ ім. І. Пулюя, 2024. 45 с.
2. Катренко А. В. Системний аналіз об'єктів та процесів комп'ютеризації : навч. посіб. Львів : Новий Світ-2000, 2022. 424 с.
3. Hyndman R. J., Athanasopoulos G. Forecasting: Principles and Practice. 3rd ed. Melbourne : OTexts, 2021. 442 p.
4. Пасічник В. В., Шаховська Н. Б. Сховища даних : навчальний посібник. Львів : Видавництво Львівської політехніки, 2021. 356 с.
5. Мороз О. О., Субботін С. О. Методи та системи штучного інтелекту : навчальний посібник. Запоріжжя : ЗНТУ, 2020. 332 с.
6. Литвин В. В., Висоцька В. А. Проектування інформаційних систем : навчальний посібник. Львів : Новий Світ-2000, 2021. 380 с.
7. Різник О. Я., Яцишин В. В. Машинне навчання та аналіз даних : навчальний посібник. Тернопіль : ТНТУ ім. І. Пулюя, 2024. 310 с.
8. Субботін С. О. Нейронні мережі та їх застосування : навчальний посібник. Запоріжжя : ЗНТУ, 2021. 290 с.
9. Глибовець М. М., Олецький О. В. Штучний інтелект : підручник. Київ : Видавничий дім «КМ Академія», 2022. 412 с.
10. Боднар Р. Т., Лужецький В. А. Основи аналізу часових рядів : навчальний посібник. Вінниця : ВНТУ, 2023. 244 с.
11. Hyndman R. J., Athanasopoulos G. Forecasting: Principles and Practice. 3rd ed. Melbourne : OTexts, 2021. 442 p.
12. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow. 3rd ed. Sebastopol : O'Reilly Media, 2022. 851 p.

13. Chollet F. Deep Learning with Python. 2nd ed. Shelter Island : Manning Publications, 2021. 504 p.
14. Brownlee J. Deep Learning for Time Series Forecasting. Melbourne : Machine Learning Mastery, 2018. 479 p.
15. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. 800 p.
16. James G., Witten D., Hastie T., Tibshirani R. An Introduction to Statistical Learning. 2nd ed. New York : Springer, 2021. 607 p.
17. Raschka S., Liu Y., Mirjalili V. Machine Learning with PyTorch and Scikit-Learn. Birmingham : Packt Publishing, 2022. 770 p.
18. McKinney W. Python for Data Analysis. 3rd ed. Sebastopol : O'Reilly Media, 2022. 579 p.
19. VanderPlas J. Python Data Science Handbook. 2nd ed. Sebastopol : O'Reilly Media, 2023. 620 p.
20. Müller A. C., Guido S. Introduction to Machine Learning with Python. Sebastopol : O'Reilly Media, 2016. 400 p.
21. Методичні вказівки для написання розділу «Безпека життєдіяльності, основи охорони праці» в кваліфікаційних роботах здобувачів освітнього рівня бакалавр / уклад. О. Я. Гурик, І. Б. Окіп. Тернопіль : ТНТУ ім. І. Пулюя, 2023.
22. Грибан В. Г., Фоменко А. Є., Казначеев Д. Г. Безпека життєдіяльності та охорона праці : підручник. Дніпро : Дніпропетровський державний університет внутрішніх справ, 2022. 440 с.
23. Про охорону праці : Закон України від 14.10.1992 № 2694-XII (у редакції станом на 2025 р.). Відомості Верховної Ради України. Київ, 2025. URL: <https://zakon.rada.gov.ua/laws/show/2694-12>.
24. НПАОП 0.00-7.11-12. Загальні вимоги стосовно забезпечення роботодавцями охорони праці працівників. Чинна редакція. Київ : Міністерство економіки України. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=58109.

25. НПАОП 0.00-7.23-23. Мінімальні вимоги щодо безпеки та здоров'я працівників на роботі. Київ : Міністерство економіки України, 2023.

ДОДАТКИ

Додаток А

Основні програмні файли

Лістинг А.1 – Модель машинного навчання Random Forest для формування тренувальних рекомендацій

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score,
classification_report
import joblib

# Завантаження набору даних
data = pd.read_csv("fitness_dataset.csv")

# Кодування категоріальних ознак
encoders = {}

categorical_columns = [
    "gender",
    "fitness_level",
    "goal",
    "equipment",
    "limitations",
    "plan_class"
]

for column in categorical_columns:
    encoder = LabelEncoder()
    data[column] = encoder.fit_transform(data[column])
    encoders[column] = encoder

# Формування вхідних ознак та цільової змінної
X = data[
```

```
[
    "age",
    "gender",
    "height_cm",
    "weight_kg",
    "fitness_level",
    "experience_years",
    "workouts_per_week",
    "goal",
    "equipment",
    "limitations"
]

y = data["plan_class"]

# Розподіл даних на навчальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42
)

# Створення та навчання моделі
model = RandomForestClassifier(
    n_estimators=100,
    max_depth=10,
    random_state=42
)

model.fit(X_train, y_train)

# Оцінювання якості моделі
y_pred = model.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)

print("Accuracy:", accuracy)
print(classification_report(y_test, y_pred))

# Збереження навченої моделі
joblib.dump(model, "random_forest_model.pkl")

# Функція прогнозування
def predict_training_plan(user_data):

    prediction = model.predict([user_data])

    plan_class = encoders["plan_class"].inverse_transform(
        prediction
    )[0]

    return plan_class

# Приклад використання
sample_user = [
    25,      # age
    1,      # gender
    180,    # height_cm
    80,     # weight_kg
    2,     # fitness_level
    2,     # experience_years
    4,     # workouts_per_week
    1,     # goal
    1,     # equipment
    0      # limitations
]

result = predict_training_plan(sample_user)
```

```

print("Recommended plan:", result)

<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="UTF-8">
<title>Прогнозування попиту</title>
<link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}">
</head>
<body>
<div class="layout">
<aside class="sidebar">
<h2>Demand Forecast</h2>
<nav>
<a href="/">Завантаження даних</a>
<a href="/prepare-page">Підготовка даних</a>
<a href="/forecast-page" class="active">Прогноз</a>
</nav>
</aside>
<main class="content">
<section class="card">
<h1>Прогнозування попиту</h1>
<p>Система формує прогноз попиту на основі підготовлених
часових рядів.</p>
<div class="form-row">
<label for="model">Модель прогнозування</label>
<select id="model">
<option value="lstm">LSTM</option>
<option value="arima">ARIMA</option>
</select>
</div>
<div class="form-row">
<label for="period">Період прогнозування</label>
<select id="period">
<option value="quarter">Поточний квартал</option>

```

```

<option value="month">Наступний місяць</option>
</select>
</div>
<button onclick="runForecast()">Сформувати прогноз</button>
</section>
<section class="card">
<h2>Результати прогнозування</h2>
<div id="status">Очікується запуск прогнозування</div>
<div class="metrics">
<div>
<span>MAE</span>
<strong id="mae">-</strong>
</div>
<div>
<span>RMSE</span>
<strong id="rmse">-</strong>
</div>
<div>
<span>MAPE</span>
<strong id="mape">-</strong>
</div>
</div>
<div class="chart-placeholder">
Графік фактичних та прогнозованих значень
</div>
</section>
</main>
</div>
<script>
function runForecast() {
  document.getElementById("status").innerText="Виконується
прогнозування...";
  fetch("/forecast", {method:"POST"})
  .then(response=>response.json())
  .then(data=>{
    if(data.status==="success") {

```

```

document.getElementById("status").innerText=data.message;
document.getElementById("mae").innerText=data.MAE;
document.getElementById("rmse").innerText=data.RMSE;
document.getElementById("mape").innerText=data.MAPE+"%";
}else{
document.getElementById("status").innerText=data.message;
}
})
.catch(())=>{
document.getElementById("status").innerText="Помилка під час
прогнозування";
});
}
</script>
</body>
</html>

```

Лістинг А.2 – Модуль підготовки та обробки даних користувача

```

import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.impute import SimpleImputer

class DataPreprocessor:

    def __init__(self):
        self.encoders = {}

    def load_data(self, file_path):
        """
        Завантаження набору даних.
        """
        return pd.read_csv(file_path)

    def remove_duplicates(self, data):
        """
        Видалення дублікатів.
        """
        return data.drop_duplicates()

    def handle_missing_values(self, data):
        """
        Обробка пропущених значень.
        """

```

```

numeric_columns = data.select_dtypes(
    include=["int64", "float64"]
).columns

categorical_columns = data.select_dtypes(
    include=["object"]
).columns

numeric_imputer = SimpleImputer(strategy="mean")
categorical_imputer = SimpleImputer(strategy="most_frequent")

data[numeric_columns] = numeric_imputer.fit_transform(
    data[numeric_columns]
)

data[categorical_columns] =
categorical_imputer.fit_transform(
    data[categorical_columns]
)

return data

def encode_categorical_features(self, data):
    """
    Кодування категоріальних ознак.
    """

    categorical_columns = [
        "gender",
        "fitness_level",
        "goal",
        "equipment",
        "limitations"
    ]

    for column in categorical_columns:

        encoder = LabelEncoder()

        data[column] = encoder.fit_transform(
            data[column]
        )

        self.encoders[column] = encoder

    return data

def prepare_features(self, data):
    """
    Формування набору ознак.
    """

```

```

features = data[
    [
        "age",
        "gender",
        "height_cm",
        "weight_kg",
        "fitness_level",
        "experience_years",
        "workouts_per_week",
        "goal",
        "equipment",
        "limitations"
    ]
]

return features

def process_dataset(self, file_path):
    data = self.load_data(file_path)
    data = self.remove_duplicates(data)
    data = self.handle_missing_values(data)
    data = self.encode_categorical_features(data)
    features = self.prepare_features(data)
    return features

# Приклад використання
preprocessor = DataPreprocessor()
prepared_data = preprocessor.process_dataset(
    "fitness_dataset.csv"
)

print(prepared_data.head())

```

Лістинг А.3 – REST API для формування персоналізованого тренувального плану

```

const express = require("express");
const router = express.Router();

const TrainingService = require("../services/TrainingService");

// Формування персоналізованого тренувального плану
router.post("/generate-plan", async (req, res) => {
    try {
        const userData = req.body;

```

```

        if (
            !userData.age ||
            !userData.height_cm ||
            !userData.weight_kg ||
            !userData.fitness_level ||
            !userData.goal
        ) {
            return res.status(400).json({
                message: "Не всі обов'язкові параметри
заповнено"
            });
        }

        const trainingPlan =
            await
            TrainingService.generatePersonalizedPlan(userData);

        return res.status(200).json({
            message: "Тренувальний план успішно сформовано",
            plan: trainingPlan
        });

    } catch (error) {
        return res.status(500).json({
            message: "Помилка формування тренувального плану",
            error: error.message
        });
    }
});

// Отримання збереженого тренувального плану користувача
router.get("/plan/:userId", async (req, res) => {
    try {
        const userId = req.params.userId;

        const plan = await TrainingService.getUserPlan(userId);

        if (!plan) {
            return res.status(404).json({
                message: "Тренувальний план не знайдено"
            });
        }

        return res.status(200).json(plan);

    } catch (error) {
        return res.status(500).json({
            message: "Помилка отримання тренувального плану",
            error: error.message
        });
    }
});

```

```
module.exports = router;
```

Лістинг А.4 – Модель бази даних та структура сутностей системи

```
from sqlalchemy import (
    Column,
    Integer,
    String,
    Float,
    ForeignKey,
    DateTime,
    Text
)

from sqlalchemy.orm import relationship
from sqlalchemy.ext.declarative import declarative_base
from datetime import datetime

Base = declarative_base()

class User(Base):
    __tablename__ = "users"

    id = Column(Integer, primary_key=True)
    full_name = Column(String(100))
    email = Column(String(100), unique=True)
    password_hash = Column(String(255))
    created_at = Column(DateTime, default=datetime.utcnow)

    profile = relationship(
        "UserProfile",
        back_populates="user",
        uselist=False
    )

    training_plans = relationship(
        "TrainingPlan",
        back_populates="user"
    )

class UserProfile(Base):
    __tablename__ = "user_profiles"

    id = Column(Integer, primary_key=True)
    user_id = Column(Integer, ForeignKey("users.id"))

    age = Column(Integer)
    gender = Column(String(20))
    height_cm = Column(Float)
    weight_kg = Column(Float)
```

```

fitness_level = Column(String(50))
experience_years = Column(Integer)

user = relationship(
    "User",
    back_populates="profile"
)

class TrainingPlan(Base):
    __tablename__ = "training_plans"

    id = Column(Integer, primary_key=True)
    user_id = Column(Integer, ForeignKey("users.id"))

    plan_type = Column(String(100))
    intensity = Column(String(50))
    workouts_per_week = Column(Integer)

    created_at = Column(
        DateTime,
        default=datetime.utcnow
    )

    user = relationship(
        "User",
        back_populates="training_plans"
    )

    exercises = relationship(
        "Exercise",
        back_populates="training_plan"
    )

class Exercise(Base):
    __tablename__ = "exercises"

    id = Column(Integer, primary_key=True)

    training_plan_id = Column(
        Integer,
        ForeignKey("training_plans.id")
    )

    name = Column(String(100))
    sets = Column(Integer)
    repetitions = Column(Integer)
    intensity = Column(String(50))

    training_plan = relationship(
        "TrainingPlan",

```

```
        back_populates="exercises"
    )

class Recommendation(Base):
    __tablename__ = "recommendations"

    id = Column(Integer, primary_key=True)

    user_id = Column(Integer,
                     ForeignKey("users.id"))

    recommended_plan = Column(String(100))

    confidence_score = Column(Float)

    generated_at = Column(
        DateTime,
        default=datetime.utcnow
    )

class TrainingHistory(Base):
    __tablename__ = "training_history"

    id = Column(Integer, primary_key=True)

    user_id = Column(Integer,
                     ForeignKey("users.id"))

    training_date = Column(DateTime)

    completed_exercises = Column(Integer)

    notes = Column(Text)
```

Додаток Б

Теза конференції

УДК 621.326

Кобель Б. – ст. гр. СП-42

Тернопільський національний технічний університет імені Івана Пулюя

РОЗРОБКА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ФОРМУВАННЯ ПЕРСОНАЛІЗОВАНОГО ПЛАНУ ТРЕНУВАНЬ НА ОСНОВІ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Науковий керівник: к.ф.-м.н., доцент Цебрій О. Р.

Kobel B.

Ternopil Ivan Puluj National Technical University

DEVELOPMENT OF AN INTELLIGENT SYSTEM FOR GENERATING PERSONALIZED TRAINING PLANS USING MACHINE LEARNING METHODS

Supervisor: PhD in Physics and Mathematics, Associate Professor Tsebrii O.

Ключові слова: машинне навчання, персоналізація, план тренувань

Keywords: machine learning, personalization, training plan

Формування ефективного плану тренувань є складною задачею, яка потребує врахування індивідуальних особливостей користувача, таких як рівень фізичної підготовки, цілі тренувань, фізіологічні параметри та історія навантажень. Традиційні підходи до побудови тренувальних програм зазвичай базуються на загальних рекомендаціях або досвіді тренера, що не завжди дозволяє досягти оптимальних результатів для кожної конкретної людини. У цьому контексті використання методів машинного навчання відкриває можливості для автоматизації процесу персоналізації та підвищення ефективності тренувань.

Сучасні інтелектуальні системи використовують різні підходи, зокрема алгоритми класифікації, регресії та рекомендаційні моделі, для аналізу даних користувачів і формування індивідуальних програм. Застосування моделей, таких як Random Forest, Gradient Boosting, нейронні мережі та методи колаборативної фільтрації, дозволяє враховувати складні залежності між параметрами користувача та результатами тренувань. При цьому важливу роль відіграє якість вхідних даних, які можуть включати інформацію про фізичну активність, біометричні показники, режим сну та харчування.

Окрім цього, дедалі більшого поширення набувають гібридні підходи, які поєднують класичні алгоритми машинного навчання з методами глибинного навчання. Такі моделі дозволяють аналізувати часові ряди (наприклад, історію тренувань або показники з носимих пристроїв) та виявляти приховані патерни у поведінці користувача. Використання рекурентних нейронних мереж та трансформерів забезпечує більш точне прогнозування прогресу та адаптацію плану тренувань у динаміці.

Експериментальні дослідження показують, що персоналізовані системи рекомендацій здатні підвищувати ефективність тренувань, зменшувати ризик перенавантаження та покращувати мотивацію користувачів. Основними метриками оцінювання є точність рекомендацій, рівень досягнення цілей користувача та стабільність прогресу в часі [1].

Важливим аспектом є побудова адекватної моделі представлення користувача, яка дозволяє формалізувати його фізичний стан, обмеження та цілі у вигляді набору параметрів. До таких параметрів можуть належати вік, маса тіла, рівень підготовки, попередній досвід тренувань, наявність травм або медичних обмежень. Формування такого профілю дозволяє системі не лише генерувати базовий план, але й забезпечувати його коректну адаптацію у процесі використання. При цьому важливо враховувати, що частина даних може бути неповною або неточною, тому моделі повинні бути стійкими до пропусків і шуму.

Окрему роль відіграє моделювання динаміки тренувального процесу, де система повинна враховувати накопичений ефект навантажень і відновлення. Для цього використовуються підходи, що дозволяють оцінювати втому, адаптацію організму та ризик перенавантаження на основі історичних даних. Аналіз часових залежностей дає змогу прогнозувати оптимальні інтервали між тренуваннями, змінювати інтенсивність і обсяг вправ, а також своєчасно вносити корективи у план. Це дозволяє перейти від статичних програм до адаптивних стратегій, які змінюються разом із користувачем.

Також важливим є врахування поведінкових факторів користувача, таких як мотивація, регулярність виконання вправ та рівень залученості. Інтелектуальна система може аналізувати ці показники та формувати рекомендації, які не лише оптимальні з фізіологічної точки зору, але й реалістичні для виконання. Наприклад, у випадку пропусків тренувань або зниження активності система може автоматично знижувати навантаження або змінювати структуру плану, щоб підтримати поступовий прогрес. Такий підхід дозволяє підвищити довгострокову ефективність тренувань і зменшити ймовірність відмови користувача від занять.

Разом з тим, існують проблеми, пов'язані з неповнотою або шумністю даних, а також необхідністю адаптації моделей до різних категорій користувачів. Тому актуальними залишаються задачі підвищення робастності моделей, забезпечення конфіденційності даних та оптимізації обчислювальних ресурсів для роботи систем у реальному часі [2].

Додатково слід зазначити, що сучасні підходи до побудови таких систем активно використовують методи нормалізації та масштабування даних, а також механізми регуляризації, що дозволяє підвищити стабільність моделей. Важливим є також впровадження адаптивних алгоритмів, які здатні змінювати структуру тренувального плану залежно від прогресу користувача, що забезпечує більш гнучкий та ефективний підхід до тренувального процесу.

Таким чином, поєднання сучасних методів машинного навчання, ефективної обробки даних та адаптивних алгоритмів формує основу для створення інтелектуальних систем, здатних генерувати персоналізовані плани тренувань з урахуванням індивідуальних характеристик користувача. У результаті формується комплексна система підтримки прийняття рішень, яка може бути інтегрована у мобільні додатки або фітнес-платформи, забезпечуючи підвищення ефективності тренувань, зниження ризиків травм та покращення загального фізичного стану користувачів.

Література:

1. Wang, J., Zhang, Y. (2022). Personalized recommendation systems based on machine learning: A review. — IEEE Access.
2. Chen, L., Chen, G., Wang, F. (2023). Recommender systems based on deep learning: A comprehensive review. — ACM Computing Surveys.