Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій

Бакалаврська кваліфікаційна робота на тему:

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ КІЛЬКОСТІ СПАЛЕНИХ КАЛОРІЙ ПІД ЧАС ТРЕНУВАНЬ»

Виконала: студентка 4 курсу, групи 2ІСТ-21б
спеціальності 126 – «Інформаційні системи та технології»
_____ Анастасія ТРУСЮК
Керівник: Ph.D., ст. викладач каф. САІТ
_____Ольга ВОЙЦЕХОВСЬКА
«03» 06 2025 р.
Рецензент: к.т.н., доцент каф. КН
_____Ігор АРСЕНЮК
«07» 06 2025 р.

Допущено до захисту

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

«06» 06 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра системного аналізу та інформаційних технологій
Рівень вищої освіти – перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність 126 – «Інформаційні системи та технології»
Освітньо-професійна програма – Прикладні інформаційні технології

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. Віталій МОКІН
"28" 03 2025 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Трусюк Анастасії Сергіївні

1. Тема роботи: «Інформаційна технологія аналізу та прогнозування кількості спалених калорій під час тренувань»
керівник роботи: Войцеховська О. О., Ph.D., ст. викл. кафедри САІТ
затверджені наказом ВНТУ від «20» 03 2025 року № 97
2. Термін подання студентом роботи 31.05.25
3. Вихідні дані до роботи:
1) Kaggle Dataset «Calories Burnt Prediction».
4. Зміст текстової частини:
1) Загальна характеристика об'єкту досліджень;
2) Вибір оптимального рішення та налаштувань для розв'язання поставленої задачі;
3) Розроблення інформаційної технології аналізу та прогнозування кількості спалених калорій під час тренувань.
5. Перелік ілюстративного матеріалу.
1) Гістограми розподілу даних;
2) Кореляційна матриця;
3) Графік візуалізації роботи моделей "Лінійна регресія", "Дерево рішень", k-найближчих сусідів" на тестових даних;
4) Таблиця порівняння точності усіх моделей.

6. Консультанти розділів роботи:

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

7. Дата видачі завдання 28.03.25

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Загальна характеристика об'єкту дослідження	01.04	10.04	вик
2	Аналіз аналогічних методів та інформаційних технологій	10.04	20.04	вик
3	Вибір оптимального рішення та налаштувань для розв'язання поставленої задачі	20.04	30.04	вик
4	Розроблення інформаційної технології аналізу та прогнозування кількості спалених калорій під час тренувань	1.05	15.05	вик
5	Оформлення матеріалів до захисту БКР	15.05	30.05	вик

Студентка _____ Анастасія ТРУСЮК

Керівник роботи _____ Ольга ВОЙЦЕХОВСЬКА

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 83 сторінок формату А4, на яких є 55 рисунка, 11 таблиць, список використаних джерел містить 20 найменувань.

Метою роботи є підвищення точності прогнозування кількості спалених калорій під час тренування шляхом створення інформаційної технології та використання методів машинного навчання.

У бакалаврській кваліфікаційній роботі здійснено аналіз предметної області, пов’язаної з вивченням витрат калорій під час фізичних навантажень. Розглянуто основні чинники, які впливають на цей процес, зокрема фізіологічні особливості користувачів та характеристики тренувань. Досліджено сучасні методи прогнозування кількості спалених калорій і виконано обґрунтований вибір інформаційних технологій для створення ефективної прогностичної системи.

У роботі детально описано процес розроблення інформаційної технології на основі відкритого датасету з платформи Kaggle із застосуванням мови програмування Python. Було проведено розвідувальний аналіз даних, виконане їх очищення, нормалізацію та попередню обробку вхідної інформації. Розроблено й протестовано кілька моделей машинного навчання для прогнозування витрати калорій під час тренувань. На основі аналізу ефективності моделей було обрано оптимальний варіант, який забезпечив найвищу точність прогнозування.

Ключові слова: інформаційна технологія, калорії, тренування, машинне навчання, аналіз, прогнозування, Python.

ABSTRACT

The bachelor's qualification work consists of 83 pages of A4 format, with 55 figures, 11 tables, and a list of references containing 20 titles.

The aim of the work is to improve the accuracy of predicting the number of calories burned during exercise by creating information technology and using machine learning methods.

The bachelor's thesis analyses the subject area related to the study of calorie expenditure during physical activity. The main factors that influence this process are considered, including the physiological characteristics of users and training characteristics. Modern methods of predicting the number of calories burned are investigated and a reasonable choice of information technologies is made to create an effective prognostic system.

The paper describes in detail the process of developing an information technology based on an open dataset from the Kaggle platform using the Python programming language. An exploratory analysis of the data was carried out, its cleaning, normalisation and preliminary processing of the input information were performed. Several machine learning models were developed and tested to predict calorie consumption during exercise. Based on the analysis of the models' efficiency, the optimal variant was chosen, which provided the highest prediction accuracy.

Keywords: information technology, calories, training, machine learning, analysis, forecasting, Python.

ЗМІСТ

ВСТУП

Актуальність теми. У сучасному суспільстві, що все більше орієнтується на збереження здоров'я, підвищення фізичної активності та контроль над власною фізичною формою, постає потреба в об'єктивному, точному та автоматизованому способі обліку енергетичних витрат людини. Зростання популярності фітнес-трекерів, мобільних додатків та персоналізованих систем здоров'я зумовлює необхідність розробки інформаційних технологій, здатних не лише фіксувати біометричні показники, а й здійснювати аналітичне прогнозування на їх основі. Відтак, проблема прогнозування кількості спалених калорій під час фізичних навантажень є не лише інженерним чи прикладним завданням, а й міждисциплінарною проблемою на перетині інформатики, фізіології та медичної статистики.

Традиційні методи оцінювання калорійних витрат часто базуються на спрощених аналітичних формулах, які не враховують багатофакторність фізіологічних параметрів, індивідуальні відмінності та інтенсивність навантаження. Це знижує точність обчислень і ефективність прийняття рішень у таких галузях, як спортивна медицина, фітнес-індустрія та дієтологія. На цьому тлі використання засобів машинного навчання та аналітики великих даних відкриває нові перспективи для високоточних, адаптивних і масштабованих рішень, які здатні в режимі реального часу прогнозувати кількість витрачених калорій за біологічними, часовими та поведінковими показниками.

Мета і задачі дослідження. Метою дослідження є підвищення точності прогнозування кількості спалених калорій під час тренування шляхом створення інформаційної технології та використання методів машинного навчання.

Для досягнення поставленої мети передбачено виконання таких задач:

- провести аналіз предметної області;
- здійснити вибір оптимальних інформаційних технологій для

інформаційної технології аналізу та прогнозування кількості спалених калорій.

- здійснити розвідувальний аналіз даних;

- розробити інформаційну технологію для прогнозування аналізу та прогнозування кількості спалених калорій під час тренувань.

Об’єктом дослідження є процес розроблення інформаційної технології для прогнозування кількості калорій, спалених під час фізичних навантажень.

Предметом дослідження є методи машинного навчання і програмні засоби, які використовуються у процесі створення інформаційної технології для прогнозування витрат калорій під час тренувань.

1 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ОБ’ЄКТА ДОСЛІДЖЕНЬ

- 1.1 Аналіз об'єкту дослідження

Фізична активність є однією з фундаментальних складових життєдіяльності людини, що безпосередньо впливає на її фізіологічний стан, енергетичний обмін, психоемоційне самопочуття та загальну якість життя. З фізіологічної точки зору будь-яке м’язове скорочення супроводжується витратою енергії, джерелом якої є метаболічні процеси в організмі. Енергетичні витрати, або так зване енергоспоживання, визначаються кількістю калорій, що спалюються під час руху, навантаження, а також у стані спокою. Таким чином, точне оцінювання калорійної витрати є надзвичайно важливим як для спортсменів, так і для звичайних користувачів, які прагнуть регулювати масу тіла, підтримувати фізичну форму або дотримуватись здорового способу життя. Так, до прикладу, на рисунку 1.1. зображено фактори, що впливають на витрату калорій під час тренувань [1].

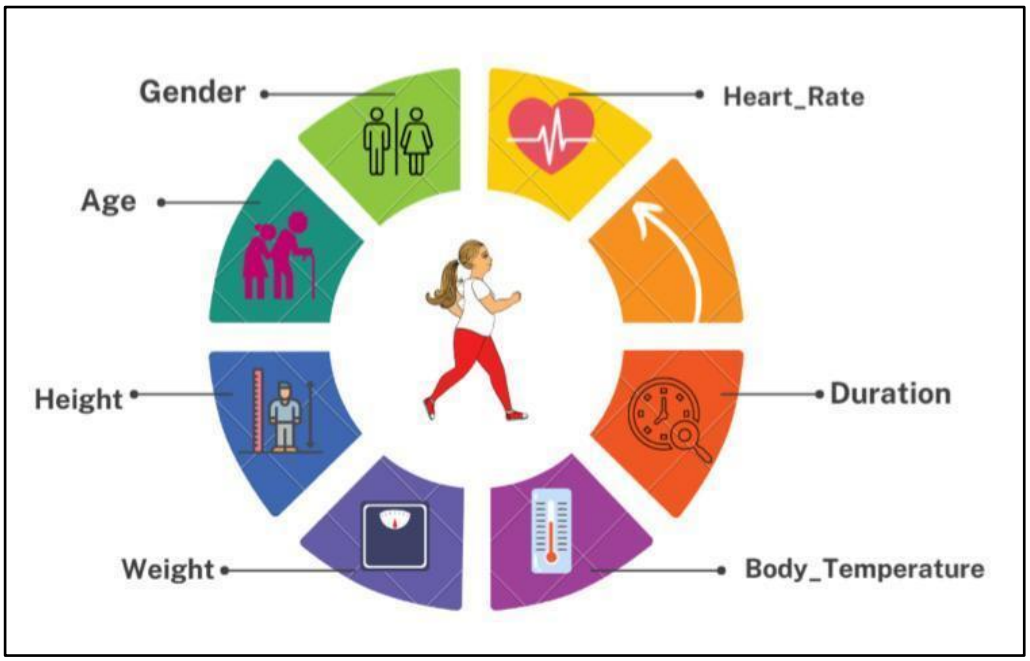

Рисунок 1.1 – Фактори, що впливають на витрату калорій під час тренувань

Поняття "калорійної витрати" в сучасній науковій літературі найчастіше розглядається в контексті енергетичного балансу, де спожиті калорії порівнюються з витраченими. При позитивному балансі (споживання > витрата) організм накопичує енергію, що зберігається у вигляді жирових відкладень; при негативному – спостерігається зменшення маси тіла. Енергетичні витрати людини загалом поділяються на три основні компоненти: базовий обмін речовин (метаболізм у стані спокою), термічний ефект їжі та енерговитрати, пов’язані з фізичною активністю [1].

Базовий обмін речовин – це мінімальний рівень енергії, необхідний для підтримання життєво важливих функцій організму у стані спокою, включаючи серцебиття, дихання, терморегуляцію, роботу внутрішніх органів тощо. На базовий обмін впливають такі чинники, як вік, стать, вага, зріст, генетичні особливості, гормональний фон. Для приблизного розрахунку базового метаболізму застосовуються емпіричні формули, зокрема формула Гарріса-Бенедикта або формула Міффліна-Сан Жеора. Наприклад, за формулою Гарріса-Бенедикта базовий обмін для чоловіків розраховується як:

$$BMR = 88{,}36 + (13{,}4 \times \text{вага у кг}) + (4{,}8 \times \text{зріст у см}) - (5{,}7 \times \text{вік у роках}). \quad (1.1)$$

Для жінок ця формула має інші коефіцієнти. Проте такі формули є статистичними узагальненнями й не враховують змін у серцевому ритмі, температурі тіла чи специфіці навантажень [2].

Іншою складовою є так званий термічний ефект їжі, що становить близько 10% від загального енергетичного обміну. Це енергія, яка витрачається організмом на травлення, засвоєння та метаболізм поживних речовин. Хоча цей параметр менш динамічний, ніж фізична активність, він також має вплив на загальний енергетичний баланс, особливо в осіб, які дотримуються спортивного харчування або проходять реабілітаційні дієти [3].

Наймінливішою та найважче контрольованою точному прогнозуванню частиною витрат калорій є фізична активність, яка включає як усвідомлене тренування (наприклад, пробіжки, фітнес-заняття, плавання), так і повсякденну активність (ходіння, стояння, пересування в побуті). Кількість калорій, які спалюються під час тренувань, залежить від цілого ряду параметрів: інтенсивності, тривалості, виду активності, а також фізіологічних характеристик індивіда – віку, статі, маси тіла, серцевого ритму, температури тіла. У зв'язку з цим виникає потреба в гнучких математичних моделях, здатних враховувати ці фактори й адаптуватися під конкретні дані користувача (табл. 1.1).

Таблиця 1.1 – Компоненти загального енергетичного обміну в організмі людини

Компонент	Частка від загального енергоспоживання	Коротка характеристика
Базовий обмін	60–75%	Витрати енергії у стані спокою
Термогенез їжі	10–15%	Енерговитрати на перетравлення їжі
Фізична активність	15–30%	Включає як свідомі навантаження, так і побутову активність

У другій половині XX століття з розвитком медико-біологічної кібернетики виникла ідея створення математичних моделей функціонування людського організму, зокрема моделей енергетичних процесів. Ці моделі поділяються на дві основні категорії: детерміновані (аналітичні) та емпіричні. Детерміновані моделі будуються на основі фізіологічних законів та рівнянь енергетичного обміну, таких як теплообмін, споживання кисню тощо. Натомість емпіричні моделі ґрунтуються на статистичних залежностях, виявлених у результаті спостережень за великими вибірками людей, і часто використовують методи регресійного аналізу або класифікації [4].

Особливої актуальності набули моделі, що інтегрують дані сенсорного моніторингу – такі як частота серцевих скорочень (пульс), тривалість активності, температура тіла, іноді навіть електрокардіографічні сигнали. Згідно з результатами ряду досліджень, найбільш інформативним біометричним параметром для оцінювання інтенсивності тренувального навантаження є саме серцевий ритм. Його значення дозволяє безпосередньо оцінити рівень аеробного або анаеробного навантаження (рис 1.2). Відомо, що енергетичні витрати зростають експоненціально з підвищенням частоти пульсу, а тривала активність із високим ритмом спричиняє не лише витрату глікогену, а й активацію ліполізу [5].

Рисунок 1.2 – Вхідні параметри для підрахунку калорій

У свою чергу, температура тіла слугує індикатором інтенсивності метаболічних процесів. Під час активних тренувань внутрішня температура зростає, що спричиняє додаткові витрати енергії на терморегуляцію (через потовиділення, розширення судин тощо). Включення показника температури тіла в модель дозволяє підвищити точність прогнозування, особливо для високоінтенсивних навантажень [6].

Завдяки сучасному прогресу в галузі інформаційних технологій стало можливим не лише реєструвати ці біометричні сигнали, але й зберігати, передавати та обробляти їх у великих обсягах. Системи інтернету речей (зокрема носимі пристрої) генерують потоки даних, які потребують розумного аналізу. Саме тут виникає потреба в інтелектуальних інформаційних системах, що використовують методи машинного навчання, здатні автоматично виявляти закономірності в даних, будувати адаптивні моделі та здійснювати прогнозування з високою точністю. Такий підхід є кардинально новим порівняно з класичними таблицями MET або формулами базового метаболізму.

У науковій літературі вже окреслено низку спроб використання методів штучного інтелекту в задачах прогнозування калорійної витрати. Найчастіше застосовуються регресійні моделі, зокрема лінійна регресія, дерево рішень, випадкові ліси, а також нейронні мережі. Основною перевагою цих методів є здатність моделювати складні взаємозв’язки між незалежними змінними (вік, вага, тривалість, пульс тощо) та залежною змінною – кількістю калорій, що спалюються [7].

Крім того, в новітніх дослідженнях все частіше акцентується увага на необхідності персоналізації моделей, тобто їхньої адаптації під конкретного користувача. У межах персоналізованого прогнозування враховуються історичні дані кожної людини, її попередній рівень активності, зміни у вазі, рівень підготовки, тривалість сну, гормональні особливості тощо. Такий підхід реалізується, зокрема, за допомогою алгоритмів адаптивного навчання та

глибокого навчання (нейронні мережі з багатьма шарами), які здатні обробляти великі масиви даних і самостійно вибирати найбільш релевантні ознаки [8].

Усе це свідчить про необхідність створення інтегрованих інформаційних рішень, які поєднують обробку біометричних сигналів, автоматичну підготовку даних, побудову моделі та її регулярне оновлення. Такий підхід дозволяє створити персоналізовану систему прогнозування енергетичних витрат, що здатна працювати в реальному часі та враховувати динамічні зміни фізіологічних параметрів людини.

У межах практичного застосування сучасних моделей оцінювання калорійної витрати постає питання не лише їх точності, а й відповідності міжнародним стандартам і нормативам, особливо коли йдеться про використання таких систем у медичних або спортивних установах. Наприклад, Всесвітня організація охорони здоров’я, а також Американський коледж спортивної медицини рекомендують використовувати валідовані інструменти та моделі з похибкою, яка не перевищує 10–15% від реальних енергетичних витрат, визначених за допомогою прямої калориметрії або спірометрії [9]. Однак, у реальних умовах повсякденного життя чи у побутових фітнес-програмах застосування прямих методів вимірювання енерговитрат є технічно складним, дорогим або просто неможливим. Саме тому виникає потреба в розробці альтернативних інформаційних технологій, які могли б поєднувати доступність, високу точність і адаптивність.

Одним із напрямів розвитку таких технологій є створення цифрових двійників фізіологічного профілю користувача. Ідея полягає в тому, щоб, спираючись на дані про вагу, зріст, вік, стать, фізіологічні реакції на навантаження (серцебиття, температура), побудувати віртуальну модель метаболічної реакції організму на фізичну активність. Цей підхід уже застосовується у спортивній медицині для визначення навантажень під час тренувань, у кардіології – для дозування реабілітаційної активності, а також у

смарт-системах споживчого сегмента, таких як Google Fit, Apple Health, Samsung Health тощо.

З технічної точки зору, практичне застосування такої моделі потребує чітко визначеної структури вхідних ознак , їх перетворення та масштабування, а також вибору адекватного типу алгоритму. У моделі прогнозування, наприклад, можуть використовуватися наступні вхідні змінні: вік, зріст, вага, стать, температура тіла, частота серцевих скорочень – тобто ті змінні, які й містяться у відкритому датасеті «Calories Burnt Prediction» з платформи Kaggle [10]. Такий набір параметрів є репрезентативним, дозволяє охопити основні біофізіологічні характеристики людини, а також забезпечити широку варіативність ситуацій – від низькоінтенсивних тренувань до високоінтенсивних навантажень.

Слід зазначити, що навіть найсучасніші алгоритми машинного навчання можуть давати хибні результати без належного попереднього аналізу даних. Для цього в сучасній науково-практичній парадигмі активно застосовуються методи візуального аналізу даних, виявлення викидів, вивчення кореляційних зв’язків між змінними, що дозволяє уникнути мультиколінеарності та інших статистичних спотворень. Такі підходи забезпечують стабільність та узгодженість моделей, що особливо важливо для задач прогнозування, які мають наслідки для здоров’я [11].

▪ 1.2 Аналіз аналогічних методів та інформаційних технологій

У сучасному суспільстві, де превентивна медицина, самостійний контроль за здоров’ям і фізична культура набувають особливого значення, інформаційні технології дедалі частіше стають невід’ємною складовою життєдіяльності. Серед таких технологій особливе місце посідають системи фітнес-моніторингу – програмно-апаратні комплекси, які забезпечують безперервний контроль за фізіологічними показниками людини з метою підтримання фізичного стану,

корекції навантажень і відстеження динаміки здоров’я. Ці системи базуються на концепціях персоніфікованої медицини, інтернету речей, мобільних обчислень і машинного навчання (табл. 1.2).

Таблиця 1.2 – Основні типи сенсорів у фітнес-моніторингу

Тип сенсора	Фізіологічний параметр	Пристрої, де використовується
Оптичний пульсометр	Частота серцевих скорочень	Смарт-годинники, браслети
Термометр	Температура тіла	Нагрудні монітори, біосенсори
Акселерометр	Рух, положення	Смартфони, фітнес-браслети

Фітнес-моніторинг – це комплексна інформаційна технологія, яка включає в себе сенсорну реєстрацію біометричних даних, їх обробку в реальному часі, інтерпретацію результатів і подання користувачеві у зрозумілому форматі. Головна мета таких систем полягає у підвищенні об’єктивності оцінювання фізичного навантаження та виявленні індивідуальних реакцій організму на тренування, що дає змогу коригувати інтенсивність вправ, зменшувати ризик перевантаження і досягати оптимальних результатів. Структурно будь-яка система фітнес-моніторингу складається з трьох основних компонентів: 1) сенсорний модуль збору даних; 2) обчислювальний модуль аналізу; 3) інтерфейс користувача або візуалізаційний блок.

На сьогодні на ринку існує кілька груп фітнес-систем, які можна класифікувати за ступенем складності та цільовим призначенням:

1. Побутові персональні трекери – це фітнес-браслети, годинники та смартфони, які реєструють базові параметри та відображають їх у мобільних застосунках. Найбільш відомі системи: Apple Watch, Xiaomi Mi Band, Samsung Galaxy Watch, Garmin Forerunner. Ці пристрої дозволяють вимірювати пульс,

кількість кроків, тривалість сну, а деякі – навіть ЕКГ та рівень кисню в крові.

2. Професійні спортивні платформи – комплексні рішення для тренерів, спортсменів і реабілітологів. Вони включають спеціалізовані датчики (нагрудні пульсометри, датчики VO2max, монітори рухів) і програмне забезпечення для глибокого аналізу тренувального процесу (наприклад, Polar Flow, Suunto App, TrainingPeaks).

3. Медичні телеметричні комплекси – використовуються у кардіології, реабілітаційній медицині та спортивній діагностиці. Зазвичай інтегруються з електронною медичною картою, мають функції екстреного оповіщення, віддаленого моніторингу та автоматичної генерації звітів.

Ці системи розрізняються за точністю, ціною, способом збору даних, інтегрованістю з іншими сервісами та рівнем автоматизації. Однак спільним для них є принцип роботи: перетворення фізіологічних показників у цифрові сигнали, їхня подальша обробка і аналіз із метою надання зворотного зв’язку користувачу (табл. 1.3).

Таблиця 1.3 – Популярні комерційні платформи фітнес-моніторингу

Назва платформи	Основна функціональність	Тип користувача
Apple Health	Відстеження активності, серця, сну	Масовий споживач
Garmin Connect	Спортивний аналіз, GPS	Професійні спортсмени
Polar Flow	Інтервальні тренування, VO2max	Тренери, реабілітологи

Однією з ключових проблем, з якою стикаються сучасні фітнес-моніторингові системи, є точність вимірювання та прогнозування. Незважаючи на активне впровадження сенсорних технологій, більшість комерційних пристроїв мають певну похибку, яка може коливатись у межах 10–25% залежно від умов використання, типу активності, фізіологічних особливостей користувача, а також зовнішніх факторів – таких як вологість, температура довкілля або рівень освітлення при оптичному вимірюванні пульсу. У

дослідженнях зазначається, що найвищу точність демонструють нагрудні пульсометри, тоді як зап’ястні пристрої частіше дають спотворення, особливо під час динамічних рухів [12].

Для підвищення точності все частіше застосовуються комбіновані підходи – інтеграція даних з кількох сенсорів, побудова багатошарових моделей машинного навчання, регулярне оновлення алгоритмів на основі накопиченої історії користувача. Саме тут на перший план виходять алгоритми адаптивного прогнозування, здатні навчатися на індивідуальних даних і формувати так званий "цифровий профіль користувача". Такі профілі вже реалізовані у деяких комерційних платформах, зокрема Apple Health та Fitbit Coach, де історія активності впливає на персональні поради щодо інтенсивності тренування та відновлення (табл 1.4) [13].

Таблиця 1.4 – Типи обробки даних у фітнес-системах

Етап обробки	Завдання	Приклад
Первинна фільтрація	Усунення шуму	Згладжування пульсу
Нормалізація	Приведення до єдиної шкали	Масштабування ваги/зросту
Моделювання	Побудова прогнозу	Нейронна мережа, XGBoost

Особливу перспективу мають нейромережеві архітектури, які дозволяють моделювати складні, часто нелінійні зв’язки між великою кількістю параметрів. Такі архітектури можуть бути як прямого поширення (багатошарові перцептрони), так і рекурентного типу (наприклад, довготривала короткочасна пам’ять), що робить їх придатними для аналізу послідовностей даних, зокрема часових рядів біометричних показників. Крім того, у деяких сучасних дослідженнях використовуються згорткові нейронні мережі для аналізу сигналів з акселерометрів або електрокардіограм [14].

Ще одним важливим напрямом є впровадження штучного інтелекту для

автоматичного виявлення аномалій у фізіологічних показниках. Наприклад, при занадто високому пульсі або критичних коливаннях температури система може подати сигнал тривоги, запропонувати завершити тренування або автоматично надіслати повідомлення лікарю – особливо якщо мова йде про літніх людей або осіб із серцево-судинними захворюваннями. Такі функції вже частково реалізовані у флагманських моделях Apple Watch, Garmin Venu, Polar Vantage, які мають вбудовані алгоритми аналізу варіабельності серцевого ритму [15].

Цей напрямок розвитку не лише відповідає сучасним потребам користувачів, а й створює фундамент для інтеграції фітнес-технологій у сферу електронної медицини, телемедицини, реабілітації, превентивної діагностики та медичного страхування. Надалі особливу увагу необхідно приділяти етичним аспектам використання таких систем: збереження приватності біометричних даних, прозорість алгоритмів, відповідальність за прийняті рішення на основі штучного інтелекту.

Прогнозування енергетичних витрат організму під час фізичної активності – це складна міждисциплінарна задача, що поєднує аспекти фізіології, спортивної медицини, математичного моделювання, інженерії обчислювальних систем і сучасних інформаційних технологій. У науковому середовищі ця проблема досліджується вже понад століття, однак з появою потужних комп’ютерів, сенсорної техніки та алгоритмів машинного навчання відбулося суттєве переосмислення як теоретичних підходів, так і прикладних рішень.

У класичній фізіологічній науці основним методом оцінки енергетичних витрат вважався калориметричний аналіз. Калориметрія як метод передбачає пряме або непряме вимірювання тепла, що виробляється організмом у процесі життєдіяльності. Пряма калориметрія передбачає облік теплового випромінювання в спеціальних камерах, тоді як непряма базується на аналізі дихання – співвідношення спожитого кисню та виділеного вуглекислого газу. Ці методи, хоча й точні, є малопридатними для повсякденного застосування через

високу вартість, складність технічного забезпечення та неможливість безперервного моніторингу в умовах активного руху.

З початку XXI століття у світовій науці почала формуватись нова парадигма – використання інтелектуальних алгоритмів для прогнозування калорійної витрати, зокрема методів штучного інтелекту та машинного навчання. Ці підходи дозволяють обробляти великі обсяги складноструктурованих даних, виявляти нелінійні закономірності та моделювати індивідуальні реакції організму на різні види навантажень. Першими методами, які були застосовані в цьому напрямі, стали лінійна регресія, регресійні дерева рішень, метод опорних векторів та наївний байєсівський класифікатор. Дослідження довели їхню ефективність при аналізі наборів даних, що містять основні біометричні параметри: вага, зріст, вік, частота серцевих скорочень, тривалість вправи тощо.

У подальших роботах акцент був зміщений у бік ансамблевих моделей – випадкових лісів, градієнтного бустингу та XGBoost, які забезпечують вищу точність за рахунок поєднання результатів кількох моделей та зниження ризику перенавчання. Наприклад, у дослідженні, проведеному Zhang Y., Liu S., Zhang J., було використано набір з 15000 записів, подібний до датасету «Calories Burnt Prediction», і доведено, що ансамблеві методи демонструють стабільну точність при прогнозуванні кількості витрачених калорій із похибкою не більш ніж 5–7% [15].

Паралельно відбувалося зростання інтересу до нейронних мереж, зокрема згорткових і рекурентних, які дозволяють моделювати часові залежності у даних, а також працювати з неповними або шумними даними. У контексті фітнес-моніторингу такі мережі можуть навчатися на потокових даних з носимих пристроїв, включаючи зміну серцевого ритму, кроки, положення тіла у просторі, що дає змогу створювати динамічні предиктивні моделі з адаптацією в реальному часі [16].

Варто зазначити, що в межах сучасних досліджень також зростає зацікавленість у використанні гібридних систем, які поєднують фізіологічні моделі з алгоритмами машинного навчання. Це дозволяє об’єднувати переваги обох підходів: теоретичну обґрунтованість і адаптивну здатність до даних. Зокрема, в проєктах фінансованих Європейською Комісією в межах програми Horizon 2020, розробляються моделі, які враховують як біофізичні основи метаболізму, так і патерни, виявлені на основі великих баз даних [17].

Особливої уваги заслуговує застосування глибокого навчання у цій сфері. Нейронні мережі з великою кількістю прихованих шарів – як рекурентного, так і згорткового типу – здатні моделювати часові залежності біометричних даних, враховувати їхній контекст і самонавчатися на нових даних у процесі використання. Це дозволяє реалізовувати концепцію онлайн-прогнозування – коли модель, вбудована у фітнес-пристрій або мобільний застосунок, з кожною новою тренувальною сесією удосконалює свої прогностичні характеристики. Окремі моделі вже продемонстрували здатність працювати в режимі реального часу з похибкою менше ніж 5%, що вважається дуже хорошим результатом у фізіологічному прогнозуванні(табл. 1.5) [16].

Таблиця 1.5 – Переваги та недоліки класичних і сучасних підходів до прогнозування

Підхід	Переваги	Недоліки
Формули (Гарріс-Бенедикт, MET)	Простота, швидкість	Неточність, не персоналізовані
Регресійні моделі	Врахування кількох параметрів	Чутливість до якості даних
Машинне навчання	Висока точність, адаптивність	Потребує великих наборів даних

Окремо слід відзначити потенціал відкритих наборів даних – таких як «Calories Burnt Prediction» на платформі Kaggle, який використовується у даній роботі. Він дозволяє не лише протестувати різні моделі, а й здійснити порівняльний аналіз підходів, що є надзвичайно цінним для академічних досліджень. Застосування цього набору в рамках машинного навчання дозволяє з високим ступенем достовірності відтворити залежності між статтю, віком, зростом, масою тіла, тривалістю вправи, пульсом, температурою тіла – і результативною змінною, тобто кількістю спалених калорій.

▪ 1.3 Постановка задачі прогнозування кількості спалених калорій

Змістовна постановка задачі прогнозування кількості спалених калорій ґрунтується на необхідності створення інформаційної технології, яка б дозволяла визначити енергетичні витрати організму людини під час фізичної активності з урахуванням багатьох біометричних, часових і фізіологічних змінних. В сучасних умовах така задача має не лише суто науковий, а й прикладний характер, оскільки результат її розв’язання може бути безпосередньо використаний у фітнес-індустрії, спортивній медицині, фізичній реабілітації, дієтології, а також у системах мобільного здоров’я.

Основною мотивацією для розв’язання цієї задачі є обмеження традиційних підходів, які спираються на спрощені формули або загальні статистичні таблиці (наприклад, значення метаболічних еквівалентів – MET), що не враховують індивідуальних особливостей організму. Навіть відомі формули Гарріса-Бенедикта чи Міффліна-Сан Жеора, які використовуються для оцінки базового обміну речовин, не здатні забезпечити достатню точність для практичного прогнозування витрат калорій під час тренувань, оскільки не враховують змін серцевого ритму, температури тіла, тривалості й інтенсивності навантаження у динаміці [2], [5].

З урахуванням розвитку технологій носимих пристроїв, які дозволяють фіксувати пульс, температуру тіла, кількість кроків, тривалість активності та інші показники у реальному часі, виникає можливість використати ці дані для побудови прогностичних моделей, що здатні адаптуватися до особливостей кожного користувача. В цьому контексті прогнозування кількості спалених калорій має розглядатися як задача регресійного аналізу, де залежна змінна (цільова) – це числове значення витрачених калорій, а незалежні змінні – це сукупність фізіологічних, демографічних і часових параметрів.

На практиці така задача реалізується через побудову функції, яка встановлює відображення між входом (параметрами тренування та фізіологічними ознаками користувача) та виходом (енергетичними витратами). Наприклад, змінними, що характеризують користувача, можуть бути: вік, стать, зріст, вага, пульс, температура тіла, тривалість активності, тип вправи. Така функція повинна враховувати не лише кількісні, але й якісні взаємозв’язки між змінними, а також їхню потенційну нелінійність та складну взаємодію.

Змістовно задачу прогнозування витрати калорій можна описати як необхідність розроблення інформаційної технології, яка:

- приймає на вхід багатовимірний масив біометричних та фізіологічних параметрів (вхідний вектор ознак);

- здійснює попередню обробку (очищення, нормалізацію, кодування змінних);

- застосовує до очищених даних обрану модель машинного навчання (регресійну або гібридну);

- формує на виході прогнозоване значення – оцінку кількості калорій, спалених під час фізичної активності;

- надає можливість візуалізації та порівняння результатів, а також оцінювання якості моделі за допомогою статистичних метрик.

Такий підхід дозволяє реалізувати персоналізоване прогнозування, що

значно підвищує точність і практичну придатність результатів. У результаті користувач може не лише отримати кількісну оцінку витрачених калорій, а й адаптувати програму тренувань відповідно до цілей (наприклад, схуднення, підтримання тонусу, спортивна підготовка тощо).

▪ 1.4 Висновки

У цьому розділі здійснено комплексний огляд теоретичних та практичних аспектів оцінювання калорійної витрати під час фізичної активності. Проаналізовано базові складові енергетичного обміну – базовий метаболізм, термічний ефект їжі та витрати на фізичну активність – й окреслено їхню роль у формуванні загальної калорійної витрати. Розглянуто класичні підходи і зазначено їхні обмеження через спрощеність моделей та відсутність урахування індивідуальних біометричних показників. Проведено огляд сучасних інформаційних систем фітнес-моніторингу, що використовують оптичні пульсометри, акселерометри й інші сенсори, та виділено ключові завдання обробки сигналів і візуалізації результатів. Формальною основою стало математичне формулювання задачі регресійного прогнозування калорійної витрати з чітким визначенням вхідного вектора ознак і цільової функції. У результаті обґрунтовано необхідність переходу від традиційних статистичних моделей до гнучких інформаційних технологій машинного навчання.

2 ВИБІР ОПТИМАЛЬНОГО РІШЕННЯ ТА НАЛАШТУВАНЬ ДЛЯ РОЗВ'ЯЗАННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

▪ 2.1 Аналіз мови та середовища програмування для розробки інформаційної технології прогнозування кількості спалених калорій

У сучасній практиці аналітики, науки про дані та машинного навчання використовується широкий спектр програмних засобів, які забезпечують підтримку задач статистичного аналізу, побудови моделей прогнозування, візуалізації результатів та інтеграції з сенсорними або мобільними платформами. Перед розробником, аналітиком або дослідником завжди постає завдання: обрати оптимальне програмне середовище, яке б поєднувало гнучкість, масштабованість, простоту реалізації, розширюваність та підтримку сучасних алгоритмів машинного навчання [7]. Саме тому в цьому підрозділі доцільно здійснити порівняльний аналіз найпопулярніших мов програмування та середовищ, що використовуються у сфері прогнозування фізіологічних показників, зокрема витрати калорій.

До ключових критеріїв, якими керуються при виборі мови програмування для задач прогнозування, належать:

- підтримка бібліотек і фреймворків машинного навчання;
- ефективність обробки числових і табличних даних;
- наявність зручних засобів візуалізації;
- активність спільноти та наявність документації;
- легкість у навчанні та розробці;
- інтеграція з зовнішніми джерелами даних (файли, API, бази даних);
- підтримка інтерактивного середовища розробки.

Нижче наведено порівняльну таблицю найбільш уживаних мов програмування для аналізу та прогнозування даних (табл. 2.1).

Таблиця 2.1 – Порівняння мов програмування для задач машинного навчання

Критерій	Python	R	Java	MATLAB
Призначення	Загального призначення, аналіз даних, гнучке ML	Статистика, візуалізація	Корпоративні системи, продуктивність	Інженерні розрахунки
Підтримка фреймворків	TensorFlow, Scikit-learn, PyTorch, Keras	caret, mlr, randomForest	Weka, Deeplearning4j	Neural Network Toolbox
Візуалізація	matplotlib, seaborn, plotly	ggplot2, lattice	Обмежена (зовнішні бібліотеки)	Вбудована
Простота для початківців	Висока	Середня	Низька	Середня
Інтеграція з веб і API	Чудова	Посередня	Потужна	Обмежена
Платформа	Кросплатформна	Кросплатформна	Кросплатформна	Ліцензійна
Вартість	Безкоштовна	Безкоштовна	Безкоштовна	Платна

Як видно з таблиці 2.1, мова Python є найбільш збалансованим вибором, оскільки поєднує в собі гнучкість, наявність великої кількості бібліотек машинного навчання, високий рівень інтеграції з іншими системами, а також простоту синтаксису, що важливо для швидкої реалізації прототипів моделей. На відміну від Java, яка більше підходить для промислових рішень, або MATLAB, орієнтованого на інженерну академічну спільноту, Python активно використовується як у наукових дослідженнях, так і у комерційних застосуваннях в галузі охорони здоров’я, фітнесу, реабілітації, біоінформатики [7].

Не менш важливою складовою є вибір середовища розробки, яке дозволяє

писати, тестувати та документувати код. Нижче наведено ще одну порівняльну таблицю середовищ програмування (табл 2.2).

Таблиця 2.2 – Порівняння середовищ програмування для аналітики та машинного навчання

Середовище	Опис	Переваги	Недоліки
Kaggle	Інтерактивне середовище Python	Миттєве виконання коду, візуалізація, Markdown	Може бути повільним для великих моделей
RStudio	IDE для мови R	Глибока інтеграція з R, графіки	Лише для мови R
Google Colab	Хмарне середовище на базі Jupyter	Безкоштовні GPU, доступ онлайн	Залежність від підключення до Інтернет
Visual Studio Code	Універсальне середовище	Плагіни для Python, інструменти розробки	Потребує конфігурації

Серед усіх перерахованих інструментів Kaggle займає провідну позицію у сфері аналітики даних. Воно забезпечує інтерактивність, можливість документування та візуалізації в одному вікні, що є особливо зручним при проведенні експериментів з даними, гіпер параметрами моделей та представленням результатів. Крім того, завдяки активній спільноті, цей інструмент має багатий набір розширень для роботи з великими обсягами інформації, базами даних, графіками тощо [17].

Враховуючи попередній аналіз, можна зробити висновок, що мова програмування Python у поєднанні з інтерактивним середовищем Kaggle є найоптимальнішим вибором для реалізації задачі прогнозування кількості спалених калорій на основі біометричних і поведінкових даних користувача. Цей вибір ґрунтується як на технічних перевагах, так і на практичних аспектах, що

підтверджується численними науковими та прикладними дослідженнями [17].

Передусім, Python забезпечує повний набір інструментів для реалізації повного циклу аналітичного процесу – від обробки даних до побудови моделей і візуалізації результатів. Зокрема, бібліотеки pandas та numpy дозволяють ефективно працювати з табличними та числовими даними, scikit-learn – реалізовує більшість класичних і сучасних алгоритмів машинного навчання, matplotlib, seaborn, plotly – забезпечують побудову високоякісних графіків і діаграм (рис 2.1). Крім того, бібліотеки xgboost, lightgbm, tensorflow і keras дозволяють реалізувати складні ансамблеві та нейромережеві архітектури, що робить Python універсальним інструментом у сфері штучного інтелекту [16].

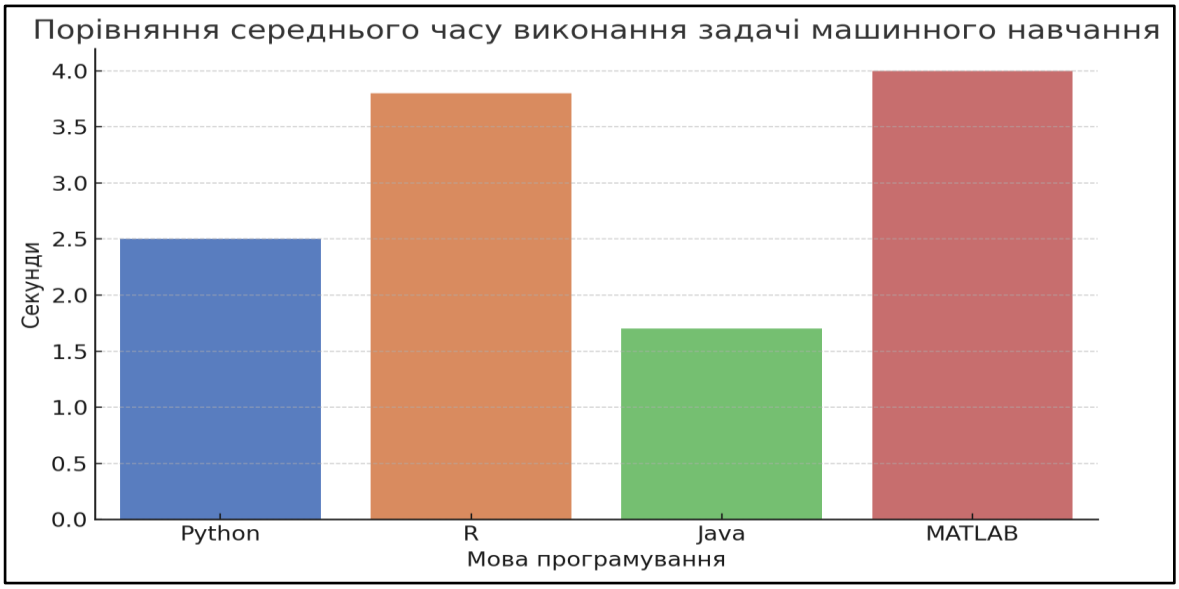

Рисунок 2.1 – Порівняння середнього часу виконання задачі машинного навчання

Однією з найважливіших переваг Python є його читабельність і лаконічність, що значно знижує поріг входу для студентів, молодих дослідників і прикладних аналітиків, а також спрощує підтримку коду в командній роботі. Крім того, Python є відкритим програмним забезпеченням, що робить його особливо привабливим для академічної та освітньої сфери, яка не завжди має доступ до платних продуктів на кшталт MATLAB чи SPSS (рис 2.2).

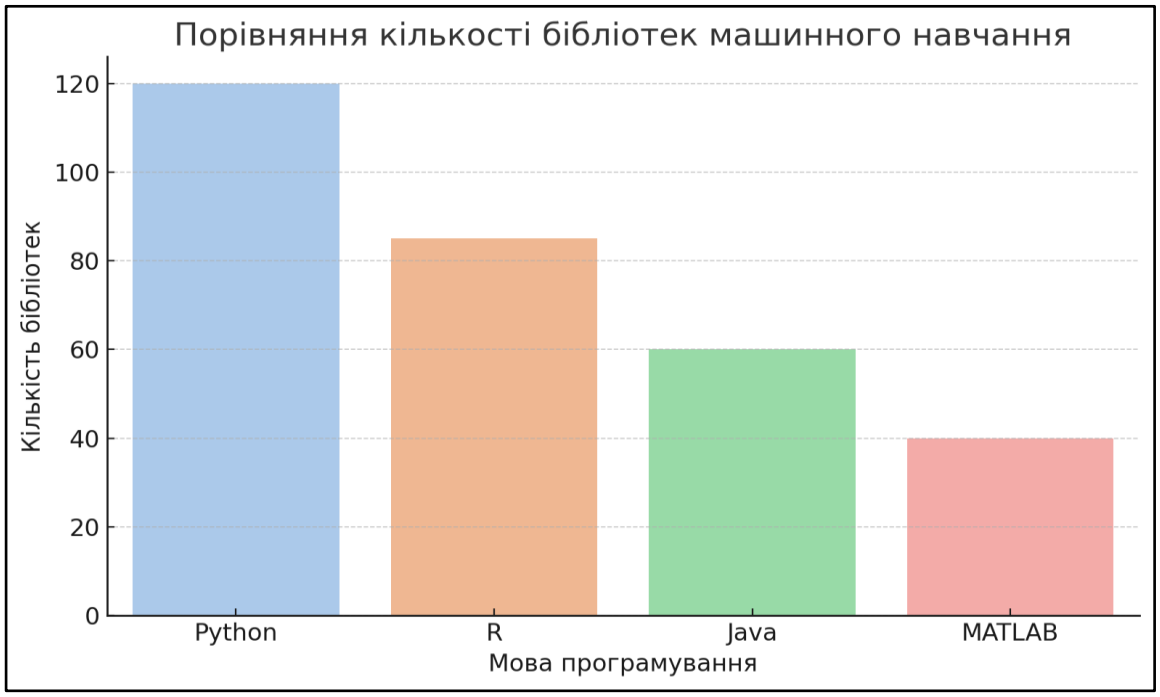

Рисунок 2.2 – Порівняння кількості бібліотек машинного навчання

Що стосується середовища Kaggle, то його переваги полягають у:

- інтерактивності – код виконується поетапно, що дозволяє швидко аналізувати результати та вносити зміни;

- можливості коментування – підтримка розмітки Markdown дозволяє інтегрувати текстові пояснення, формули та графіки безпосередньо в документ;

- репрезентативності результатів – ноутбуки зручно використовувати як звіти, навчальні посібники або документацію до моделей;

- гнучкій підтримці форматів візуалізації – включаючи інтерактивні графіки та HTML-вивід;

- підтримці великої кількості ядер обробки, зокрема Python, R, Julia, що робить його універсальним середовищем для аналітики.

Таким чином, вибір інструментального середовища Python + Kaggle є обґрунтованим як з точки зору наукової доцільності, так і з погляду практичної ефективності. Цей стек інструментів дозволяє не лише реалізувати всі етапи обробки та аналізу даних, але й забезпечує гнучкість, розширюваність, зручність відтворення результатів та прозорість аналітичного процесу.

2.2 Попередня обробка даних та розвідувальний аналіз

Для виконання поставленої задачі було обрано набір даних «Calories Burnt Prediction», розміщений на платформі Kaggle [10]. Цей датасет широко використовується у навчальних і прикладних цілях, оскільки містить повний спектр біометричних, часових та фізіологічних ознак, які можуть бути релевантними для побудови регресійної моделі передбачення кількості спалених калорій.

Усього набір даних містить 15000 записів, де кожен рядок відповідає окремій сесії тренування користувача. Кожен запис має дев’ять стовпців (ознак), з яких вісім є вхідними (незалежними) змінними, а одна – цільовою (залежною). Набір є збалансованим, дані подано у плоскому табличному форматі з однорідними типами змінних.

Нижче подано таблицю з детальною характеристикою кожного поля набору даних (табл 2.3).

Таблиця 2.3 – Опис змінних у датасеті «Calories Burnt Prediction»

Назва змінної	Тип даних	Одиниця виміру	Опис
User_ID	Цілочисельний	Умовний ідентифікатор	Ідентифікатор користувача, не використовується у моделі
Gender	Категоріальний	Чоловік/Жінка	Стать користувача (потребує кодування для моделі)
Age	Цілочисельний	Роки	Вік користувача на момент тренування
Height	Дійсне число	Сантиметри	Зріст користувача
Weight	Дійсне число	Кілограми	Вага користувача
Duration	Дійсне число	Хвилини	Тривалість фізичної активності

Heart_Rate	Дійсне число	Удари за хвилину	Середня частота серцевих скорочень під час сесії
Body_Temp	Дійсне число	Градуси Цельсія	Середня температура тіла під час тренування
Calories	Дійсне число	Кілокалорії	Кількість спалених калорій (цільова змінна)

Набір даних характеризується високим ступенем повноти та внутрішньої узгодженості. Категоріальні змінні, такі як «Gender», потребують перетворення за допомогою методу кодування, а числові – можуть бути стандартизовані або нормалізовані для зменшення впливу масштабів під час тренування моделей.

Для кращого розуміння особливостей змінних нижче наведено основні статистичні характеристики числових полів (табл 2.4).

Таблиця 2.4 – Описова статистика числових змінних у датасеті

Змінна	Мінімум	Максимум	Середнє значення	Стандартне відхилення
Age	20	74	37.1	11.2
Height (см)	123	222	169.8	10.5
Weight (кг)	36	132	74.5	14.2
Duration (хв)	1	30	14.8	8.3
Heart_Rate	74	114	92.1	10.1
Body_Temp	37.1	41.5	39.6	0.8
Calories	3	314	104.5	52.6

На основі наведеної статистики можна зробити низку висновків: по-перше, дані охоплюють широкий спектр фізіологічних параметрів, що дозволяє будувати моделі, орієнтовані як на молодих, так і на людей похилого віку. По-друге, розкиди значень не є надто великими, що свідчить про можливість

ефективного нормалізованого навчання без значного впливу екстремальних значень.

Отже, датасет «Calories Burnt Prediction» [10] є репрезентативним, структуровано повним та придатним для реалізації задачі регресійного прогнозування. Він охоплює як статичні (наприклад, стать, зріст), так і динамічні (пульс, температура тіла) характеристики користувача, що дозволяє створити адаптивну модель, здатну прогнозувати кількість витрачених калорій з високою точністю.

У межах підготовки даних для побудови прогнозної моделі було виконано низку ключових етапів попередньої обробки, спрямованих на досягнення коректності й узгодженості вхідного масиву ознак. Це необхідно для того, щоб зменшити вплив шуму, уникнути спотворення оцінок моделей і забезпечити стабільне та швидке навчання алгоритмів машинного навчання.

Спочатку було завантажено датасет «calories.csv» та виведено перші та останні рядки записів за допомогою методу head() бібліотеки pandas (рис 2.3).

```
# Завантаження даних
def load_data(filename):
    try:
        df = pd.read_csv(/kaggle/input/calories-burnt-prediction/calories.csv)
        print(f"Дані успішно завантажено. Розмір датасету: {df.shape}")
        return df
    except Exception as e:
        print(f"Помилка при завантаженні даних: {e}")
        return None
```

Рисунок 2.3 – Завантаження даних датасету

Це дозволяє отримати загальне уявлення про структуру таблиці, назви стовпців та формат значень (рис 2.4).

```
    User_ID  Gender  Age  Height  ...  Duration
Heart_Rate  Body_Temp  Calories
0  14733363    male   68   190.0  ...      29.0
       105.0       40.8     231.0
1  14861698  female   20   166.0  ...      14.0
        94.0       40.3      66.0
2  11179863    male   69   179.0  ...       5.0
        88.0       38.7      26.0
3  16180408  female   34   179.0  ...      13.0
       100.0       40.5      71.0
4  17771927  female   27   154.0  ...      10.0
        81.0       39.8      35.0

[5 rows x 9 columns]

...
        User_ID  Gender  Age  Height  ...  Duration
Heart_Rate  Body_Temp  Calories
14995  15644082  female   20   193.0  ...      11.0
        92.0       40.4      45.0
14996  17212577  female   27   165.0  ...       6.0
        85.0       39.2      23.0
14997  17271188  female   43   159.0  ...      16.0
        90.0       40.1      75.0
14998  18643037    male   78   193.0  ...       2.0
        84.0       38.3      11.0
14999  11751526    male   63   173.0  ...      18.0
        92.0       40.5      98.0

[5 rows x 9 columns]
```

Рисунок 2.4 – Вивід перших та останніх записів

Далі викликом info() було відображено типи даних кожного стовпця та наявність пропусків (рис 2.5, 2.6).

```python
# Базова інформація про датасет
print("Базова інформація про датасет:")
print(df.info())
print("\nСтатистика по числовим полям:")
print(df.describe())
print("\nПриклад рядків даних:")
print(df.head())
print("\n...")
print(df.tail())

# Перевірка на пропущені значення
print("\nПеревірка на пропущені значення:")
print(df.isnull().sum())
```

Рисунок 2.5 – Перевірка на пропущені значення

```
Перевірка на пропущені значення:
User_ID       0
Gender        0
Age           0
Height        0
Weight        0
Duration      0
Heart_Rate    0
Body_Temp     0
Calories      0
dtype: int64
```

Рисунок 2.6 – Результати перевірки

Виявилося, що датасет не містить відсутніх значень, що спрощує етап очищення, проте дозволяє одразу перейти до перетворення змінних. Створюємо копію даних та перетворюємо категоріальні змінні на числові (2.7).

```python
# Створюємо копію даних для аналізу без категоріальних змінних
df_numeric = df.copy()

# Перетворюємо категоріальні дані (стать) у числові
if 'Gender' in df_numeric.columns:
    df_numeric = pd.get_dummies(df_numeric, columns=['Gender'], drop_first=True)

# Видаляємо нечислові стовпці (User_ID)
if 'User_ID' in df_numeric.columns:
    df_numeric = df_numeric.drop('User_ID', axis=1)
```

Рисунок 2.7 – Перетворення категоріальних даних

Далі було проведено аналіз основних статистичних показників (мінімум, максимум, середнє, стандартне відхилення) для всіх кількісних полів (рис 2.8).

```python
# Дослідницький аналіз даних
def exploratory_analysis(df):
    print("\n1. ДОСЛІДНИЦЬКИЙ АНАЛІЗ ДАНИХ\n")

    # Базова інформація про датасет
    print("Базова інформація про датасет:")
    print(df.info())
    print("\nСтатистика по числовим полям:")
    print(df.describe())
    print("\nПриклад рядків даних:")
    print(df.head())
    print("\n...")
    print(df.tail())
```

Рисунок 2.8 – Аналіз основних статистичних показників

Результат аналізу основних статистичних показників представлено на рисунку 2.9.

```
Статистика по числовим полям:
            User_ID           Age  ...     Body_Temp
      Calories
count  1.500000e+04  15000.000000  ...  15000.000000
  15000.000000
mean   1.497736e+07     42.789800  ...     40.025453
     89.539533
std    2.872851e+06     16.980264  ...      0.779230
     62.456978
min    1.000116e+07     20.000000  ...     37.100000
      1.000000
25%    1.247419e+07     28.000000  ...     39.600000
     35.000000
50%    1.499728e+07     39.000000  ...     40.200000
     79.000000
75%    1.744928e+07     56.000000  ...     40.600000
    138.000000
max    1.999965e+07     79.000000  ...     41.500000
    314.000000
```

Рисунок 2.9 – Результат аналізу

Відповідно до візуалізації типу boxplot, побудованої для змінної "Калорії" у розрізі статі, було виконано аналіз з метою виявлення суттєвих викидів у наборі даних (рис 2.10).

```python
# Додатково: аналіз залежності калорій від статі
if 'Gender' in df.columns:
    plt.figure(figsize=(8, 6))
    sns.boxplot(x='Gender', y='Calories', data=df)
    plt.title('Залежність калорій від статі')
    plt.savefig('gender_calories.png')
    plt.close()
```

Рисунок 2.10 – Аналіз за типом boxplot

На рисунку 2.11 зображено результат аналізу за типом boxplot.

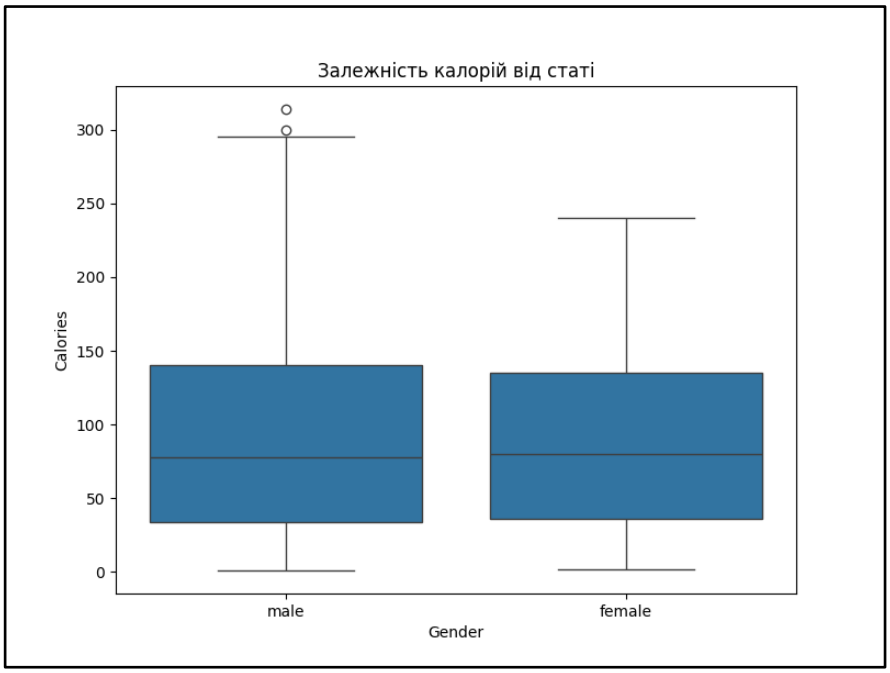

Рисунок 2.11 – Результат аналізу за типом boxplot

З рисунку 2.11 видно, що середнє значення калорій, витрачених під час тренувань, є приблизно однаковим для чоловіків і жінок. Діапазон значень також схожий, хоча у чоловіків спостерігається дещо ширший розкид, що свідчить про більшу варіативність у фізичних навантаженнях.

Для розуміння потенційної корисності ознак щодо прогнозування кількості витрачених калорій, доцільно провести попередній кореляційний аналіз (рис 2.12).

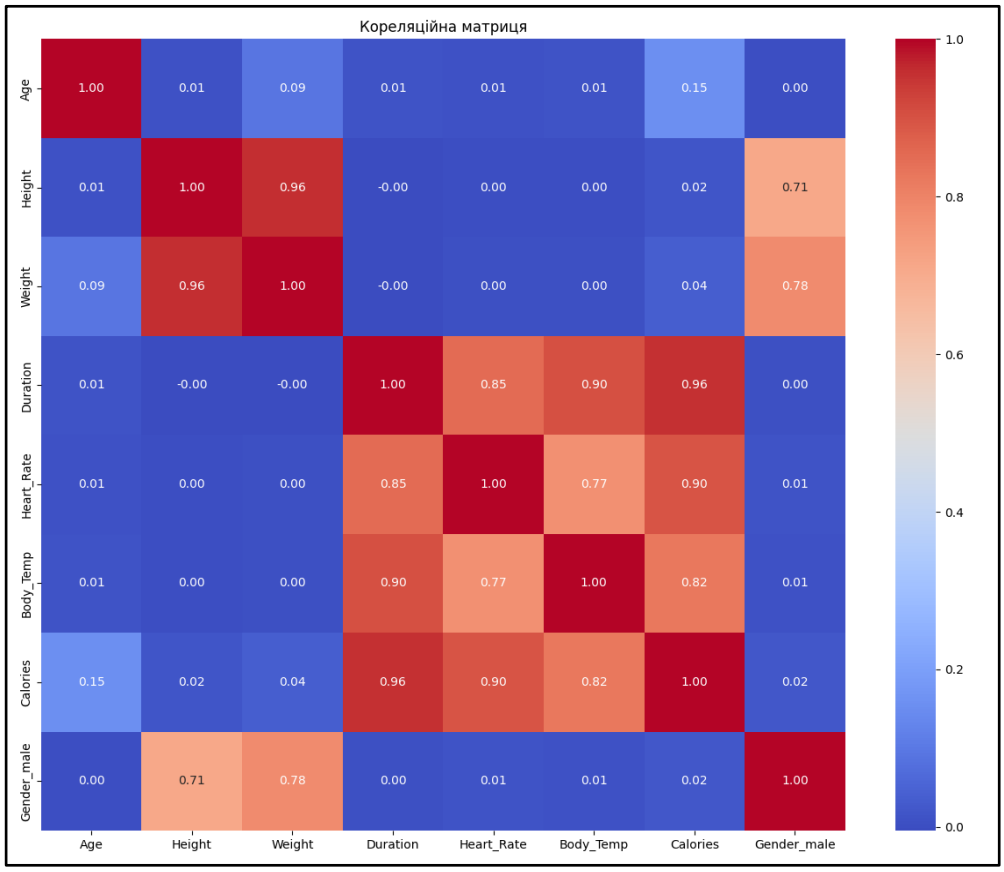

Рисунок 2.12 – Кореляційна матриця

За результатами аналізу бачимо, що найсильніші кореляції з цільовою змінною мають:

1. Duration (тривалість фізичної активності) – безпосередньо впливає на енерговитрати. Має значення 0,96 і є найсильнішою позитивною кореляцією. Це свідчить про те, що збільшення тривалості фізичної активності напряму веде до значного зростання витрати калорій.

2. Heart_Rate (середня частота серцевих скорочень під час сесії) – показник інтенсивності фізичного навантаження. Має значення 0,90 та має високу позитивну кореляцію. Підвищений пульс під час виконання вправ асоціюється зі зростанням енергозатрат.

3. Body_Temp (температура тіла) – важливий показник при розрахунку енерговитрат через масу тіла. Має значення 0,82 та свідчить про сильний зв’язок. Збільшення температури тіла під час фізичної активності є додатковим

показником інтенсивності навантаження, що впливає на витрату калорій.

Значення кореляцій демонструють, що модель прогнозування повинна зосередитися насамперед на тривалості, пульсі та вазі, а інші змінні можуть бути другорядними або використовуватись для побудови комплексних взаємозв'язків (табл. 2.5).

Таблиця 2.5 – Орієнтовні коефіцієнти кореляції з цільовою зміною Calories.

Змінна	Коефіцієнт кореляції (r)
Duration	0.96
Heart_Rate	0.90
Body_Temp	0.82
Age	0.15
Weight	0.04
Height	0.02

На рисунку 2.13 зображено розподіл числових змінних категорії "Вік".

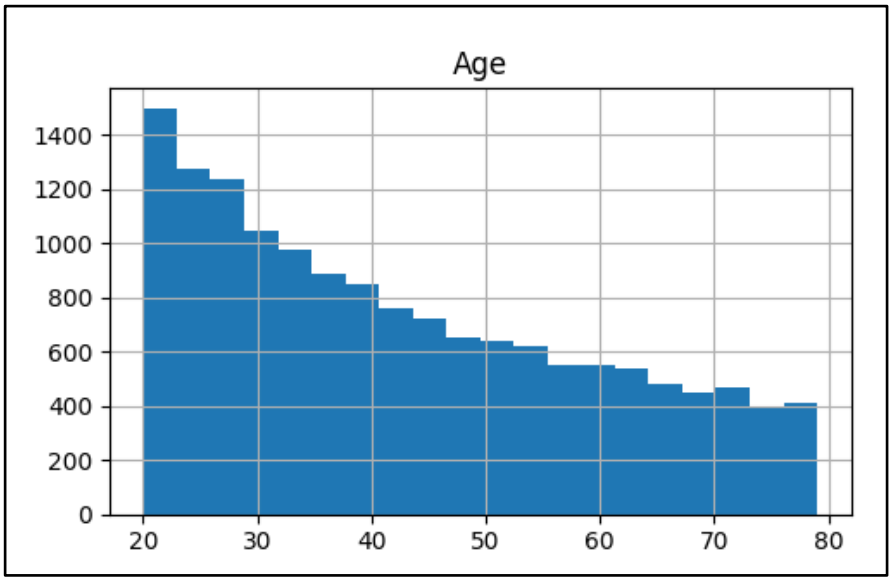

Рисунок 2.13 – Розподіл числових змінних категорії "Вік"

Розподіл вікових груп демонструє асиметричний характер, при цьому найбільшу частку становлять молоді користувачі у віковій категорії від 18 до 30 років. Із підвищенням вікового показника спостерігається поступове зменшення кількості зареєстрованих випадків.

На рисунку 2.14 зображено розподіл числових змінних категорії “Зріст”.

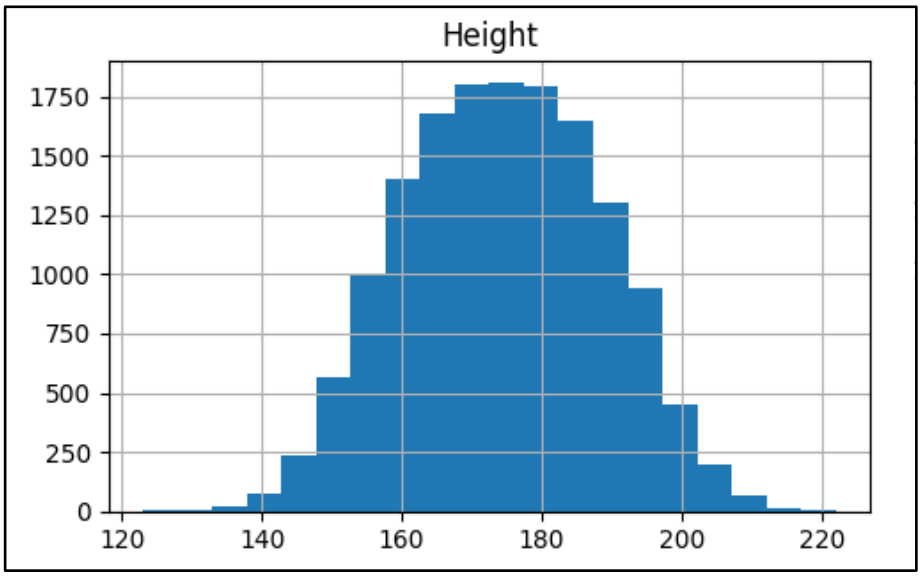

Рисунок 2.14 – Розподіл числових змінних категорії “Зріст”

Досліджувані дані категорії “Зріст” підпорядковуються нормальному розподілу з центральною тенденцією, що знаходиться в інтервалі 170–180 см. Такий розподіл засвідчує рівномірність та репрезентативність вибірки за показником зросту.

Далі розглянемо розподіл числових змінних категорії “Вага”, який представлений на рис. 2.15.

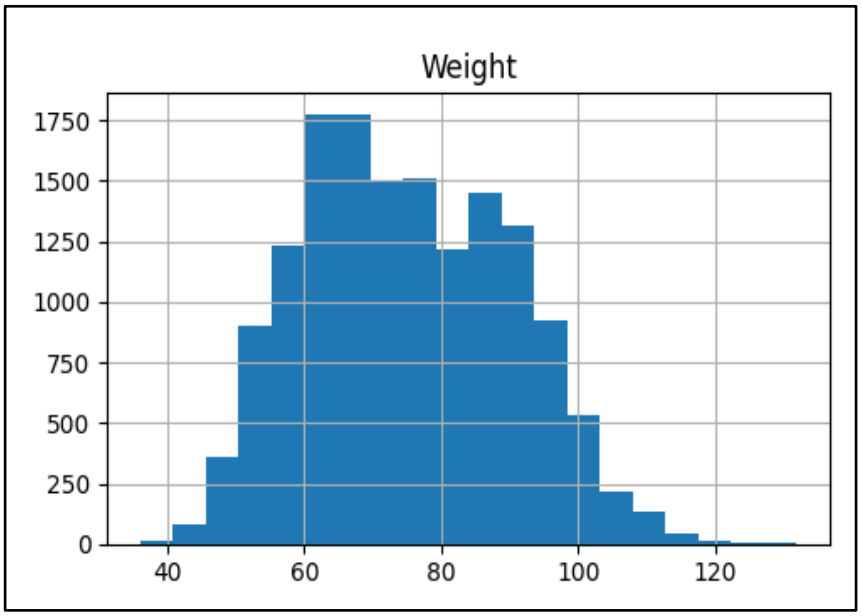

Рисунок 2.15 – Розподіл числових змінних категорії “Вага”

Як бачимо з рисунка, розподіл категорії "Вага" характеризується правосторонньою асиметрією. Основна частка користувачів належить до вагового інтервалу 60–80 кг. Водночас кількість записів із вагою понад 100 кг є незначною.

Розподіл числових змінних категорії “Тривалість тренування” зображено на рис. 2.16

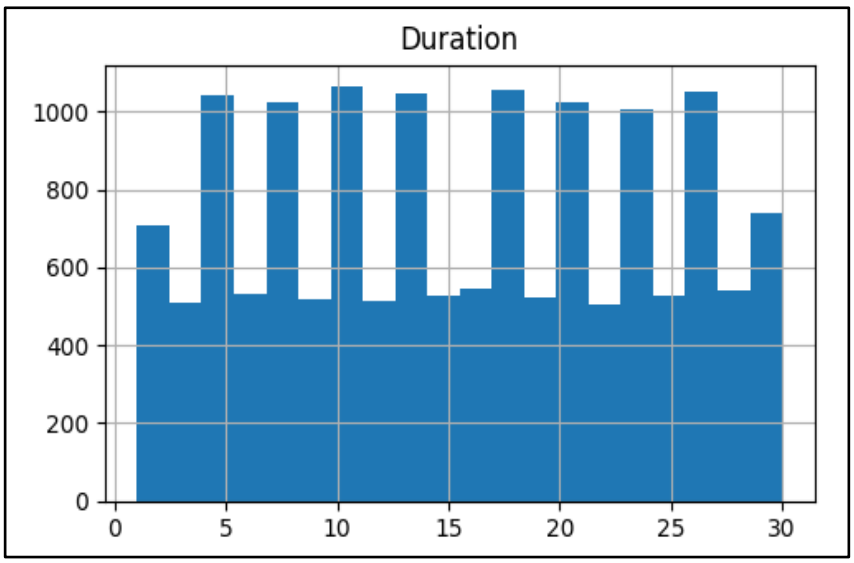

Рисунок 2.16 – Розподіл числових змінних категорії “Тривалість тренування”

З рисунка 2.6 бачимо, що розподіл категорії “Тривалість тренування” є рівномірним із періодичними піками, що може вказувати на сталість тривалості тренувальних сесій. Інформація охоплює інтервал до 30 хвилин.

Наступний розподіл, який проаналізуємо, це розподіл числових змінних категорії “Частота серцевих скорочень” (рис.2.17).

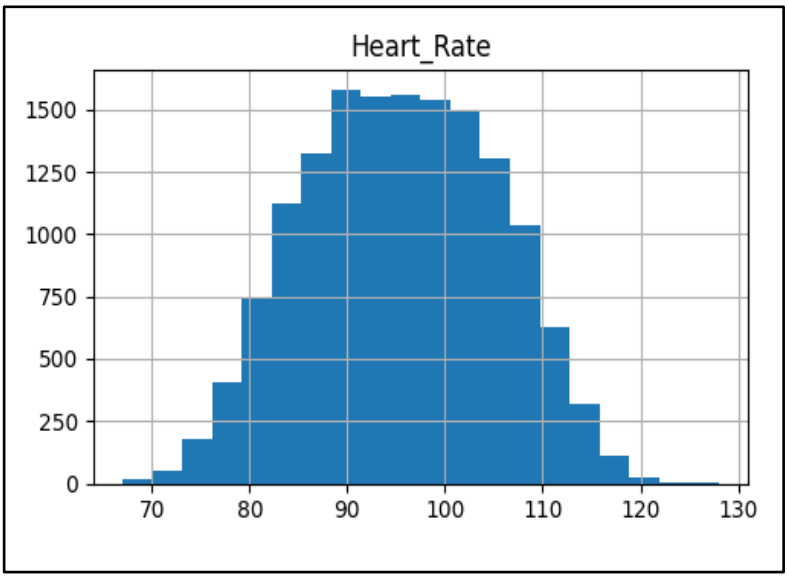

Рисунок 2.17 – Розподіл числових змінних категорії “Частота серцевих скорочень”

У розподілі частоти серцевих скорочень спостерігається нормальний характер із середнім показником, що варіюється в межах 90–100 ударів за хвилину. Такий інтервал є характерним для частоти серцевих скорочень під час помірного рівня фізичної активності.

Далі розглянемо розподіл числових змінних категорії “Температура тіла”, представлений на рисунку 2.18.

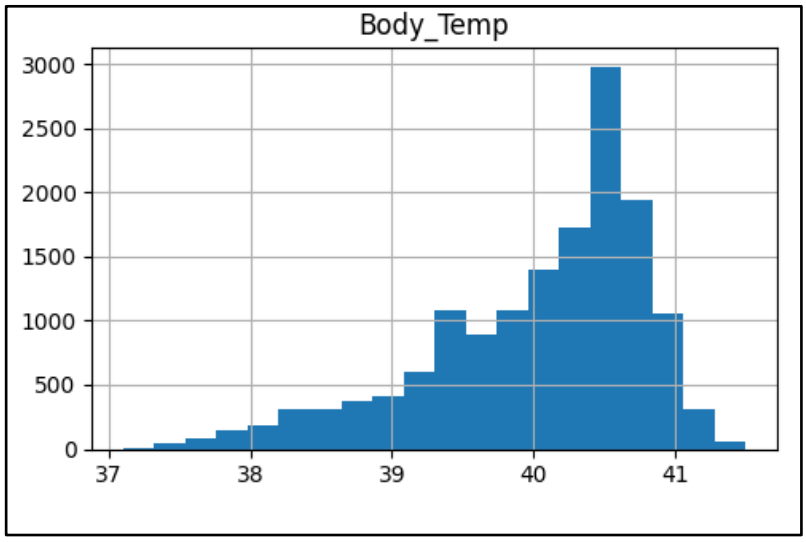

Рисунок 2.18 – Розподіл числових змінних категорії “Температура тіла”

Як видно з рисунка 2.18, категорія температура тіла характеризується асиметричним розподілом із максимумом, що наближається до 40°C. Показники нижче 38°C зустрічаються вкрай рідко, що свідчить про фізіологічний стан організму під час його активної діяльності.

На рисунку 2.19 зображений розподіл числових змінних категорії “Витрата калорій”.

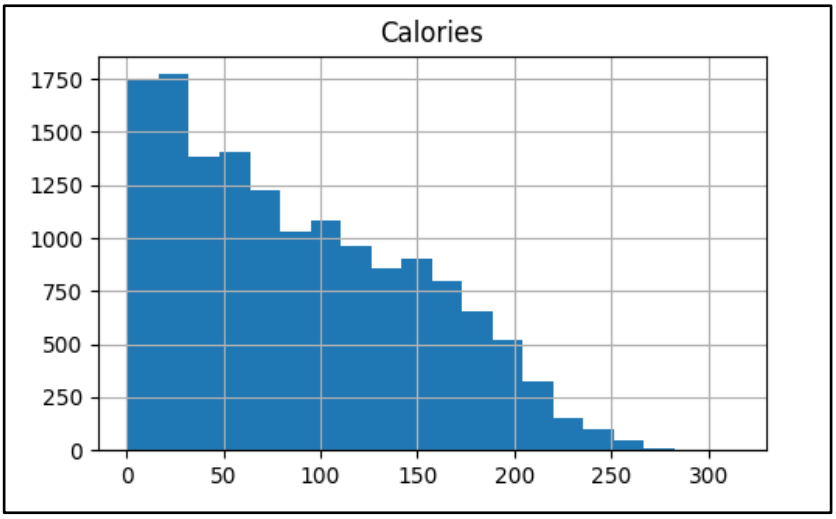

Рисунок 2.19 – Розподіл числових змінних категорії “Витрата калорій”

Аналіз розподілу категорії “Витрата калорій” свідчить, що переважна частина тренувань характеризується низьким рівнем енергетичних витрат (менше 100 ккал). Частота випадків із зростанням енерговитрат поступово зменшується.

▪ 2.3 Вибір моделей машинного навчання для прогнозування кількості спалених калорій

При виборі алгоритмів машинного навчання для задачі регресійного прогнозування кількості спалених калорій ми орієнтуємося на такі критерії:

1. Точність прогнозу – мінімізація середньоквадратичної помилки та середньої абсолютної помилки;
2. Стабільність – здатність моделі не перенавчатися на шумі даних;
3. Інтерпретованість – можливість пояснити вклад окремих ознак у прогноз;
4. Обчислювальні витрати – швидкість навчання та передбачення при даному обсязі даних;
5. Наявність готових імплементацій – підтримка у бібліотеках Python.

У рамках задачі прогнозування кількості спалених калорій варто звернути увагу на низку сучасних регресійних моделей, які широко використовуються в практиці аналізу даних. Для дослідження були розглянуті такі методи [19]:

1. Метод Випадковий ліс. Він являє собою ансамблеву модель машинного навчання, що складається з набору незалежних дерев рішень. Кожне дерево у цьому наборі будується на основі випадкової вибірки з доступних даних. Підсумковий прогноз формується шляхом усереднення вихідних результатів усіх дерев, що забезпечує стабільність і точність моделі. Серед ключових переваг такого алгоритму варто виділити високу ефективність, здатність визначати значущість ознак для побудови моделі, а також

універсальність у роботі як з числовими, так і зі змішаними наборами даних.

2. Метод «LGBM Regressor (Light Gradient Boosting Machine)». Цей метод є високоефективним алгоритмом градієнтного бустингу, розробленим для швидкої обробки великих обсягів даних і забезпечення високої продуктивності. На відміну від традиційного Gradient Boosting, LGBM застосовує гістограмний метод побудови дерев, що значно скорочує час навчання при збереженні точної роботи моделі.

3. Метод «XGB Regressor (Extreme Gradient Boosting)». Цей метод пропонує розширені можливості регуляризації, що дозволяє ефективно контролювати складність моделі та запобігати перенавчанню. XGB Regressor зазвичай демонструє кращі результати порівняно з іншими моделями на структурованих даних і є одним із найпоширеніших рішень у задачах, де потрібна висока точність прогнозування [19].

4. Лінійна регресія є фундаментальним методом, який припускає лінійну залежність цільової змінної від вхідних ознак. Модель описується функцією

$$\hat{y} = \beta_0 + \sum_{j=1}^{n} \beta_j x_j, \tag{2.1}$$

де β_j – вагові коефіцієнти, що оцінюються мінімізацією функції втрат, зазвичай середньоквадратичної похибки

$$L(\beta) = \frac{1}{m} \sum_{i=1}^{m} \left(\beta_0 + \sum_{j=1}^{n} \beta_j x_j^{(i)} - y^{(i)} \right)^2. \tag{2.2}$$

Лінійна регресія легка в інтерпретації – кожний коефіцієнт показує середній вплив відповідної ознаки [17].

5. Дерева рішень. Метод дерев рішень розбиває простір ознак на регіони шляхом послідовних бінарних розщеплень. При побудові дерева шукається змінна x_j та поріг s, які мінімізують сумарну дисперсію цільової змінної після розбиття

$$\min_{j,s} \sum_{x_i \le s} (y_i - \bar{y}_{\text{left}})^2 + \sum_{x_i > s} (y_i - \bar{y}_{\text{right}})^2. \quad (2.3)$$

Результатом є набір листів – областей, у кожному з яких прогнозом є середнє значення y по точках цього листа. Дерева легко інтерпретувати, але вони схильні до переобучення, якщо не обмежувати глибину [20].

6. Метод k-найближчих сусідів. Метод k-найближчих сусідів здійснює прогноз шляхом пошуку k точок у навчальній вибірці, які мають найменшу відстань до нової точки за допомогою метрики (часто — евклідова відстань).

Прогноз для нової точки обчислюється як середнє значення цільової змінної по її k найближчих сусідах.

Аналіз вхідних даних та методів машинного навчання дозволив обґрунтовано обрати такі моделі для подальшого дослідження: Лінійна регресія, Дерева рішень, Метод k найближчих сусідів.

У таблиці 2.6 наведено порівняльну характеристику всіх обраних моделей.

Таблиця 2.6 – Порівняльна характеристика обраних моделей

Модель	Точність	Стійкість до перенавчання	Обчислювальні витрати	Інтерпретованість
Лінійна регресія	Низька	Висока	Низькі	Висока
Дерево рішень	Середня	Низька (без обрізки)	Середні	Висока
Метод найближчих сусідів	Висока	Висока	Вищі	Помірна

▪ 2.4 Висновки

У цьому розділі обґрунтовано вибір технічної платформи та підготовлено дані для побудови прогнозної моделі. Першочергово проведено порівняльний аналіз мов програмування й середовищ розробки, у результаті якого Python у поєднанні з Kaggle визнано оптимальними інструментами завдяки широкому набору бібліотек, інтерактивності та зрозумілому синтаксису. Детально описано структуру набору даних «Calories Burnt Prediction»: наведено типи ознак, їхню роль і статистичні характеристики, а також виявлено кореляційні зв’язки ключових параметрів із цільовою змінною. Виконано теоретичний огляд низки регресійних моделей – від лінійної регресії та дерев рішень до методу найближчих сусідів – із наведенням ключових формул та порівняльною характеристикою. Таким чином, вибір моделей та налаштувань сформує надійну основу для подальшої оптимізації прогнозування кількості спалених калорій.

3. РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ КІЛЬКОСТІ СПАЛЕНИХ КАЛОРІЙ ПІД ЧАС ТРЕНУВАНЬ

▪ 3.1 Розроблення інформаційної технології

Для розробки інформаційної технології прогнозування кількості спалених калорій під час тренування необхідно створити комплекс методів машинного навчання, які попередньо навчатимуться на оброблених даних та проходитимуть тестування на встановлених наборах даних. У межах поставленої задачі обрано такі методи: Лінійна регресія, Дерево рішень, Метод найближчих сусідів.

Для налаштування зазначених моделей потрібна специфікація параметрів, на основі яких буде здійснюватися їх адаптація. Для визначення оптимальної комбінації параметрів вирішено застосувати метод GridSearchCV, який зарекомендував себе як ефективний та доступний у реалізації. У випадку, якщо точність моделі не відповідатиме вимогам дослідження, параметри буде скориговано до досягнення задовільного рівня результатів.

Після отримання результатів планується провести порівняльний аналіз моделей, з метою обрання найкращої для подальшого вивчення. Обрану модель буде піддано детальному аналізу шляхом оцінки впливу характеристик вхідних даних на процес навчання. Зокрема, буде проаналізовано ступінь внеску кожної окремої ознаки в процес прийняття рішень моделі під час класифікації даних.

На рисунку 3.1 представлено блок-схему алгоритму роботи інформаційної технології прогнозування кількості спалених калорій під час тренувань.

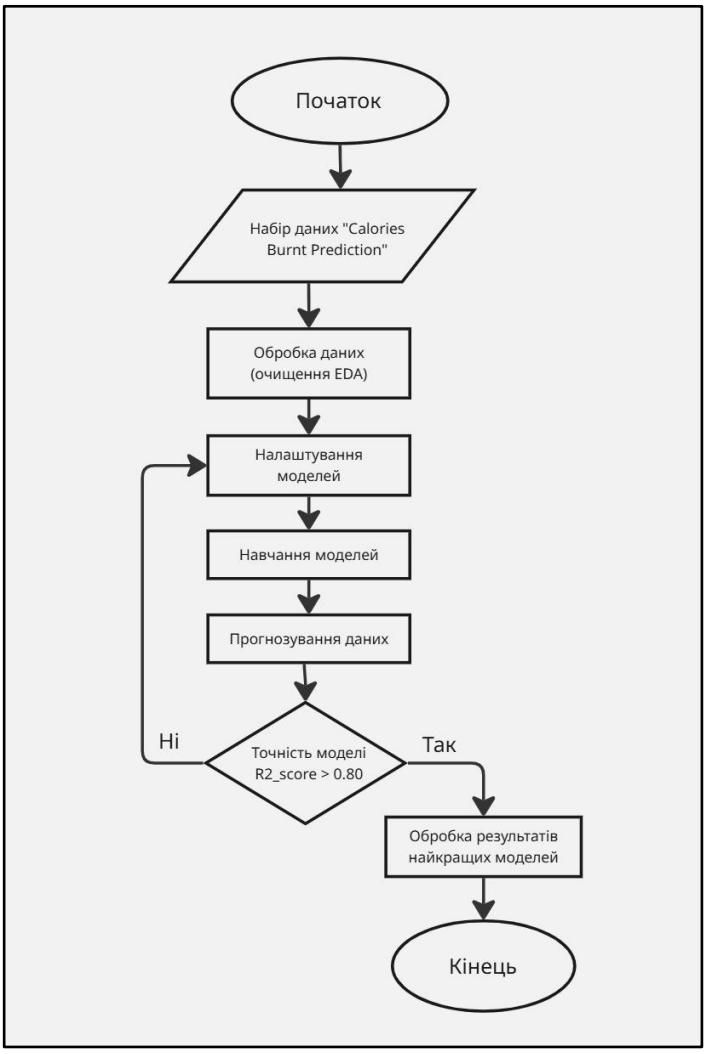

Рисунок 3.1 – Блок-схема алгоритму роботи інформаційної технології

На рисунку 3.1 представлено алгоритм функціонування інформаційної технології, спрямованої на прогнозування кількості калорій, витрачених під час фізичних тренувань. Процес розпочинається із отримання набору даних із платформи Kaggle, зокрема з набору "Calories Burnt Prediction". Першим етапом є попередня обробка даних, яка охоплює їх очищення, нормалізацію та первинний розвідувальний аналіз (Exploratory Data Analysis, EDA). Цей початковий аналіз слугує для виявлення базових закономірностей у даних та ідентифікації можливих аномалій.

У наступному етапі здійснюється налаштування моделей машинного навчання, що включає вибір відповідних алгоритмів та визначення початкових

гіперпараметрів. Далі алгоритми проходять тренування на відповідному тренувальному наборі даних. Після завершення навчання моделі застосовуються для прогнозування кількості спалених калорій на тестовому наборі даних.

Центральна частина алгоритму зосереджена на оцінці якості роботи моделі, яка визначається через метрику точності R^2_score. У разі, якщо модель демонструє точність вище порогового значення ($R^2 > 0.80$), відбувається подальша обробка результатів найкращих моделей із метою отримання підсумкових висновків. Якщо ж точність залишається недостатньою, система повертається до етапу налаштування алгоритмів для подальшої оптимізації параметрів і покращення результатів.

Зазначена блок-схема ілюструє структурований підхід до вирішення задачі прогнозування калорійних витрат. Вона забезпечує чітку послідовність ключових етапів – від збору даних до вибору найбільш ефективної моделі, сприяючи системному вирішенню поставленого завдання.

Для структурованого представлення функціональних можливостей розробленої інформаційної технології прогнозування кількості спалених калорій під час тренувань було використано нотацію UML (Unified Modeling Language), зокрема діаграму варіантів використання (Use Case), яка забезпечує візуалізацію основних сценаріїв взаємодії користувачів із системою.

На рисунку 3.2 представлено Use Case діаграму функціонування інформаційної технології прогнозування спалення калорій під час тренувань, на якій зображено основних учасників (акторів) — користувача (спортсмена або тренувального клієнта), фітнес-тренера та систему машинного навчання. Також представлено ключові функціональні можливості, реалізовані в межах створеної технології.

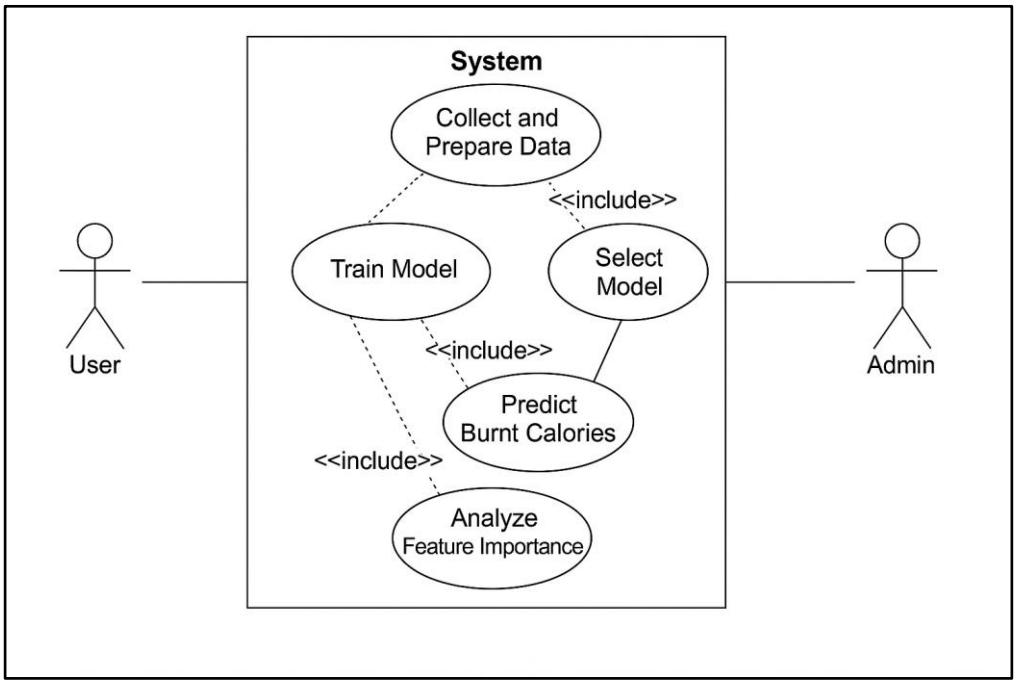

Рисунок 3.2 – UML Use Case діаграма функціонування інформаційної технології прогнозування спалення калорій під час тренувань

Серед основних варіантів використання системи виокремлюються такі:

1. Надання вхідних даних: Користувач або тренер вводить необхідні анкетні та фізіологічні дані (вік, вага, зріст, температура тіла, пульс, тривалість тренування тощо), які формують основу для аналізу.

2. Попередня обробка даних: Система здійснює перевірку введених даних, їх очищення та нормалізацію для забезпечення коректної роботи алгоритмів машинного навчання.

3. Тренування моделі: Тренер або адміністратор ініціює процес навчання моделі на основі зібраного датасету, що включає попередньо оброблені приклади тренувань і відповідні значення витрати калорій.

4. Прогнозування кількості спалених калорій: Після навчання модель здатна прогнозувати витрату калорій на основі введених параметрів конкретного тренування.

5. Аналіз важливості ознак: Система визначає, які саме вхідні параметри (наприклад, вага, пульс або тривалість тренування) найбільше впливають на

результат прогнозу.

6. Отримання результатів: Користувач або тренер отримує детальну інформацію щодо прогнозованої кількості спалених калорій, яка може бути використана для подальшого коригування плану тренувань.

Ця діаграма є основою для формування загального уявлення про функціональну структуру системи, що включає розподіл обов'язків між акторами та визначення ключових сценаріїв її використання. Такий підхід сприяє більш ефективному проектуванню інформаційної технології прогнозування кількості спалених калорій, забезпечує цілісність взаємодії між компонентами системи та створює основу для її подальшої розробки, масштабування й інтеграції в реальні фітнес-сервіси.

На рисунку 3.3 наведена UML-діаграма компонентів, яка ілюструє архітектуру інформаційної технології для прогнозування кількості спалених калорій під час фізичних тренувань. Система побудована за модульним підходом і складається з ключових компонентів, що реалізують основні етапи обробки даних.

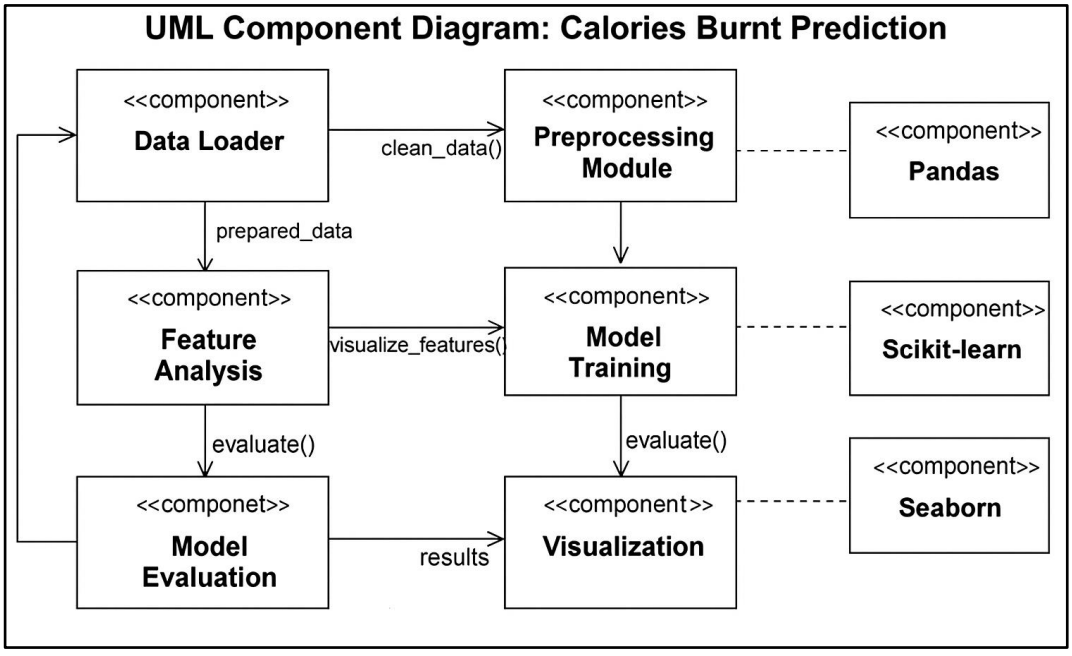

Рисунок 3.3 – UML діаграма компонентів

Компонент Data Loader відповідає за завантаження набору даних, що містять інформацію про фізичну активність користувачів. Після цього дані передаються до Preprocessing Module, де здійснюється їх очищення, нормалізація, а також базова обробка.

Наступним етапом є аналіз ознак у модулі Feature Analysis, який допомагає визначити, які фактори найбільше впливають на процес витрати калорій. Підготовлені ознаки потім передаються до модуля Model Training, де здійснюється навчання моделей машинного навчання.

Навчені моделі оцінюються в компоненті Model Evaluation через обчислення ряду метрик точності, таких як MAE, MSE та R^2. Отримані результати передаються до модуля Visualization, який забезпечує візуалізацію результатів у вигляді графіків.

Зазначена архітектура забезпечує структуровану організацію процесу аналізу даних і прогнозування, сприяючи зрозумілій взаємодії між окремими компонентами системи та надаючи можливість для її подальшої адаптації і вдосконалення.

▪ 3.2 Тренування моделей машинного навчання для прогнозування кількості спалених калорій

Для початку оптимізації параметрів моделей машинного навчання проведемо підготовку даних (рис 3.4).

```python
# Підготовка даних для моделювання
def prepare_data(df):
    print("\n2. ПІДГОТОВКА ДАНИХ ДЛЯ МОДЕЛЮВАННЯ\n")

    # Створюємо копію даних
    df_model = df.copy()

    # Видаляємо стовбчик User_ID, якщо він є
    if 'User_ID' in df_model.columns:
        df_model = df_model.drop('User_ID', axis=1)

    # Кодування категоріальних змінних
    if 'Gender' in df_model.columns:
        df_model = pd.get_dummies(df_model, columns=['Gender'], drop_first=True)

    # Вибір ознак та цільової змінної
    X = df_model.drop('Calories', axis=1)
    y = df_model['Calories']

    # Масштабування ознак
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
```

Рисунок 3.4 – Підготовка даних

Далі розділяємо наші дані на тренувальну та тестову вибірку (рис 3.5).

```python
# Розділення даних на тренувальну та тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

print(f"Розмір тренувальної вибірки: {X_train.shape}, Розмір тестової вибірки: {X_test.shape}")

return X, X_scaled, X_train, X_test, y_train, y_test, scaler
```

Рисунок 3.5 – Розділення даних

Наступним етапом проводимо аналіз даних за обраними методами. Спершу проведемо аналіз для тренувальних даних. Для аналізу розглянемо модель лінійної регресії та проведемо оцінку точності. Зважаючи на те, що модель лінійної регресії не передбачає наявності гіперпараметрів для налаштування, вона буде застосована у стандартному вигляді (рис. 3.6-3.7).

```python
# Лінійна регресія
def linear_regression_model(X_train, X_test, y_train, y_test):
    print("\n3. ЛІНІЙНА РЕГРЕСІЯ\n")

    # Навчання моделі
    model = LinearRegression()
    model.fit(X_train, y_train)
```

Рисунок 3.6 – Навчання моделі лінійної регресії на тренувальних даних

```python
# Прогнозування
y_pred = model.predict(X_test)

# Оцінка моделі
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f"Середньоквадратична помилка (MSE): {mse:.2f}")
print(f"Корінь середньоквадратичної помилки (RMSE): {rmse:.2f}")
print(f"Коефіцієнт детермінації (R²): {r2:.2f}")
```

Рисунок 3.7 – Прогнозування та оцінка Лінійної регресії

На рисунку 3.8 можемо побачити результат аналізу за обраним методом.

```
Тренувальна вибірка:
  R²: 0.9672
  RMSE: 11.27
  MAE: 8.31
  MSE: 126.95
```

Рисунок 3.8 – Результат оцінки методом “Лінійна регресія”

Точність моделі по метриці “Коефіцієнт детермінації” дорівнює 0.96, що означає, що модель прогнозує 96% даних і є гарно натренованою. На рисунку 3.9 можемо побачити візуалізацію прогнозування за методом «Лінійна регресія».

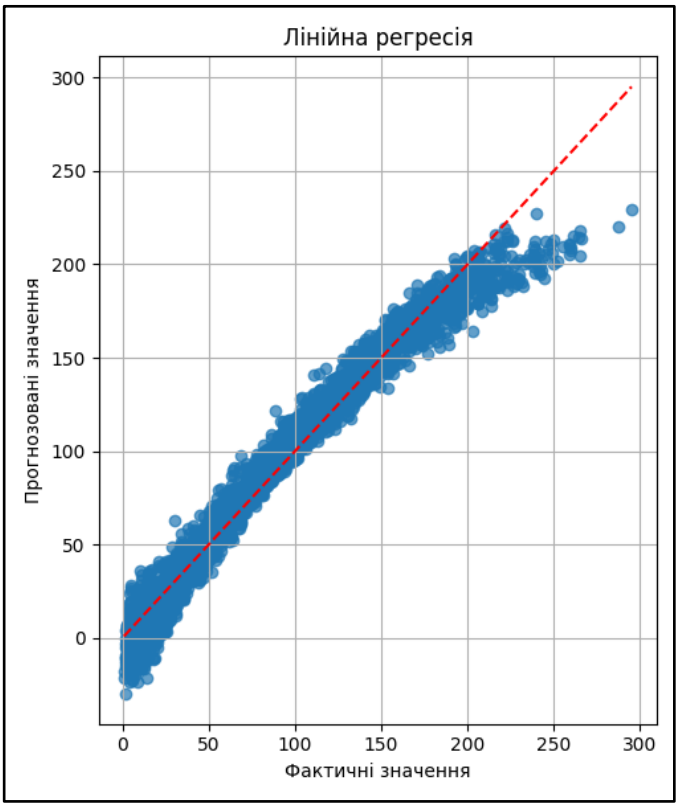

Рисунок 3.9 – Графік візуалізації роботи моделі “Лінійна регресія” на тренувальних даних

На рисунку 3.9 зображено графік розсіювання, який демонструє взаємозв’язок між фактичними значеннями та прогнозними оцінками, отриманими за допомогою моделі «Лінійна регресія». Розташування більшості точок поблизу червоної лінії, яка свідчить про ідеальні значення, означає високу точність моделі та її ефективність у забезпеченні кореляції між реальними та передбаченими даними. Такий результат доводить, що дана модель є успішним інструментом для виконання задачі регресійного прогнозування в рамках заданої проблематики.

Наступним етапом буде аналіз за методом дерево рішень. Вона є однією з перспективних моделей, яка має широкий спектр застосування. Для автоматизованої оптимізації її параметрів розробляється код, що реалізує модель регресії. Ця модель проходить етап навчання на тренувальних даних та використовується для прогнозування на тестових вибірках.

Оцінювання її якості включає такі метрики, як середньоквадратична похибка (MSE), корінь середньоквадратичної похибки (RMSE), коефіцієнт детермінації (R^2), а також проводиться крос-валідація. Додатково аналізується значущість ознак, які впливають на результати передбачення, що дозволяє краще зрозуміти внесок кожного з факторів у модель (3.10).

```
# Дерево рішень
def decision_tree_model(X_train, X_test, y_train, y_test):
    print("\n4. ДЕРЕВО РІШЕНЬ\n")

    # Навчання моделі
    model = DecisionTreeRegressor(random_state=42)
    model.fit(X_train, y_train)
```

Рисунок 3.10 – Навчання моделі на тренувальних даних

Наступним етапом проводимо прогнозування моделі та її оцінку (рис 3.11).

```
# Прогнозування
y_pred = model.predict(X_test)

# Оцінка моделі
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
```

Рисунок 3.11 – Прогнозування та оцінка моделі

Після етапу прогнозування та оцінки моделі можемо проаналізувати отриманий результат (рис 3.12)

```
R²: 0.9961
RMSE: 3.91
MAE: 2.72
MSE: 15.26
```

Рисунок 3.12 – Результат оцінки методом “Дерево рішень”

Дана модель має точність 0.99, що означає, що вона спрогнозувала 99% даних. Це, в свою чергу, є хорошою ознакою адекватності даної моделі. На рисунку 3.13 представлено графічну інтерпретацію прогнозування, виконаного за методом “Дерево рішень”.

.

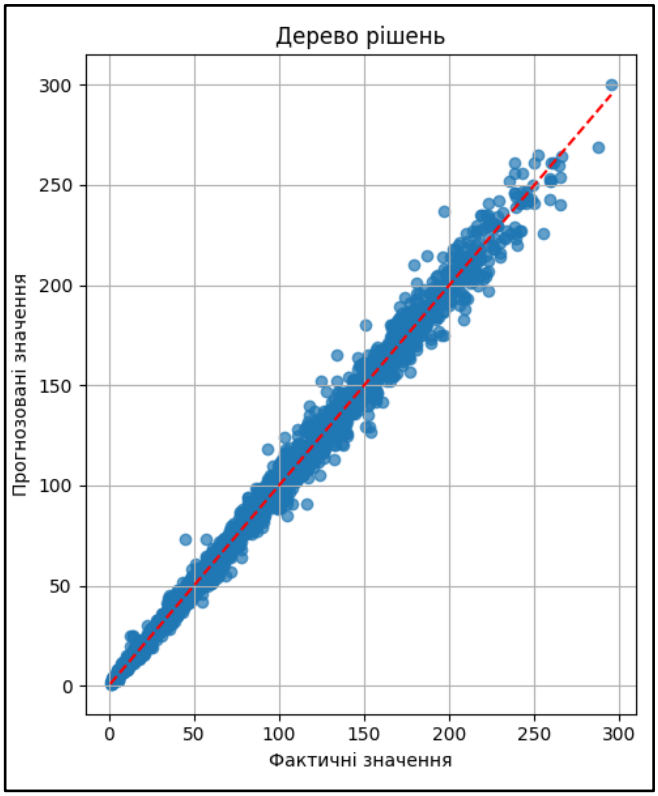

Рисунок 3.13 – Графік візуалізації роботи моделі “Дерево рішень” на тренувальних даних

На рисунку 3.13 представлено графік розсіювання, який ілюструє взаємозв’язок між фактичними значеннями та прогнозами, отриманими за допомогою моделі «Дерево рішень». Досить близьке розташування більшості точок до червоної лінії свідчить про високу точність моделі та її здатність ефективно відображати кореляцію між реальними та передбаченими даними.

Наступною для аналізу буде модель “Метод k-найближчих сусідів”. Цей метод машинного навчання, хоч і базовий, є одним із найефективніших і широко застосовується для розв'язання задач регресії та класифікації. Його суть полягає

в прогнозуванні значення цільової змінної, орієнтуючись на середнє значення чи більшість відповідей найближчих точок у багатовимірному просторі ознак.

Для побудови такої моделі створюється код, який дозволяє обрати оптимальну кількість сусідів (гіперпараметр k) і перевірити продуктивність моделі на тестових даних. Модель навчається на тренувальній вибірці й використовує передбачення, базуючись на евклідовій відстані або іншій обраній метриці (рис 3.14).

```python
# Метод найближчих сусідів (KNN)
def knn_model(X_train, X_test, y_train, y_test):
    print("\n5. МЕТОД НАЙБЛИЖЧИХ СУСІДІВ (KNN)\n")

    # Пошук оптимального k
    rmse_values = []
    k_values = range(1, min(21, len(X_train)))

    for k in k_values:
        model = KNeighborsRegressor(n_neighbors=k)
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        rmse = np.sqrt(mean_squared_error(y_test, y_pred))
        rmse_values.append(rmse)
```

Рисунок 3.14 – Пошук оптимального значення k

На рисунку 3.15 зображений фрагмент коду для проведення прогнозування та оцінки даної моделі.

```python
# Прогнозування
y_pred = model.predict(X_test)

# Оцінка моделі
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

print(f"Середньоквадратична помилка (MSE): {mse:.2f}")
print(f"Корінь середньоквадратичної помилки (RMSE): {rmse:.2f}")
print(f"Коефіцієнт детермінації (R²): {r2:.2f}")
```

Рисунок 3.15 – Прогнозування та оцінка моделі

На рисунку 3.16 можемо бачити результат коду для проведення прогнозування та оцінки даної моделі.

```
Оптимальне значення k: 10
```

```
Тренувальна вибірка:
  R²: 0.9947
  RMSE: 4.51
  MAE: 3.25
  MSE: 20.30
```

Рисунок 3.16 – Результат оцінки методом “k-найближчих сусідів”

Дана модель має точність 0.99, що означає, що вона спрогнозувала 99% тренувальних даних. Що свідчить, що модель є гарно натренована. На рисунку 3.17 наведена графічна візуалізація результатів прогнозування, здійсненого методом k-найближчих сусідів.

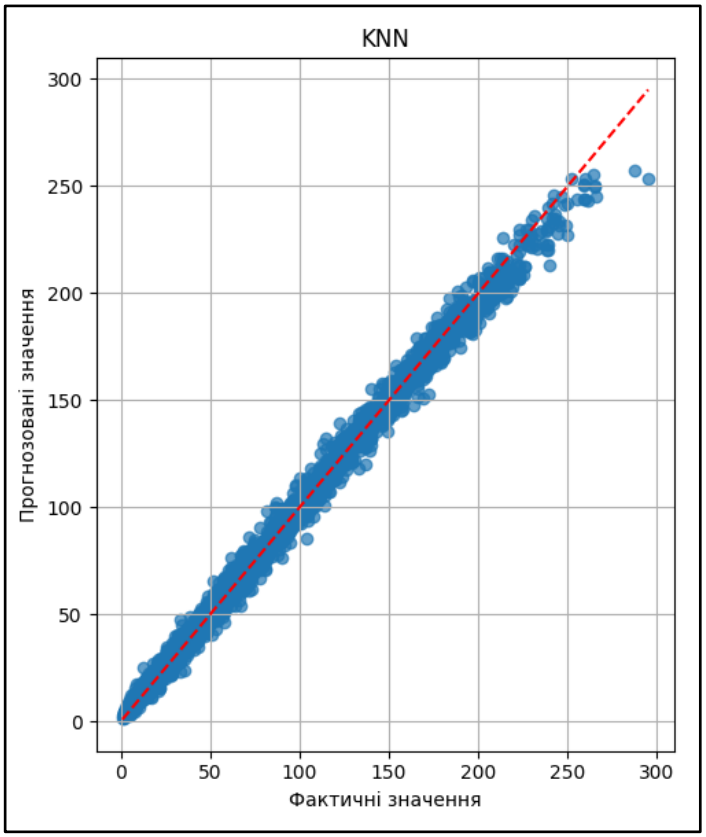

Рисунок 3.17 – Графік візуалізації роботи моделі “ k-найближчих сусідів” на тренувальних даних

На представленому зображенні відображено графік розсіювання, що демонструє зосередженість більшості точок у безпосередній близькості до червоної лінії, це означає високу точність моделі. Отримані результати підтверджують, що дана модель виступає як надійний інструмент для розв'язання задач регресійного прогнозування в рамках заданої проблематики та має найбільшу точність.

- 3.3 Застосування моделей на тестових даних

Після того, як ми протестували обрані моделі для аналізу на тренувальних даних, використовуємо їх для аналізу на тестових даних. Для початку завантажуємо дані та проводимо перевірку на наявність усіх потрібних колонок. Потім проводимо прогнозування за обраними методами. На рисунку 3.18 наведено приклад встановлення нових тестових даних.

```python
# Прогнозування на нових даних
def make_predictions(best_model_name, models, scaler, X):
    print("\n7. ПРОГНОЗУВАННЯ НА НОВИХ ДАНИХ\n")

    # Приклад нових даних
    new_data = pd.DataFrame({
        'Age': [30],
        'Height': [175.0],
        'Weight': [75.0],
        'Duration': [20.0],
        'Heart_Rate': [95.0],
        'Body_Temp': [40.0],
        'Gender_male': [1]  # 1 для чоловіка, 0 для жінки
    })

    print("Нові дані для прогнозування:")
    print(new_data)
```

Рисунок 3.18 – Приклад нових даних

На наступному етапі ми проводимо аналіз на наявність усіх потрібних даних у колонках (рис 3.19).

```
# Переконаємося, що у нових даних є всі потрібні колонки
for col in X.columns:
    if col not in new_data.columns:
        new_data[col] = 0

# Відкидаємо непотрібні колонки і забезпечуємо правильний порядок
new_data_X = new_data[X.columns]
```

Рисунок 3.19 – Перевірка на наявність потрібних колонок

Після перевірки даних, проводимо аналіз та прогнозування на цих відібраних тестових даних (3.20).

```
# Прогнозування
predictions = {}
best_prediction = None

for name, model in models.items():
    pred = model.predict(new_data_scaled)[0]
    predictions[name] = pred
    if name == best_model_name:
        best_prediction = pred

print("\nРезультати прогнозування:")
for model_name, prediction in predictions.items():
    print(f"{model_name}: {prediction:.2f} калорій")

print(f"\nПрогноз найкращої моделі ({best_model_name}): {best_prediction:.2f} калорій")
```

Рисунок 3.20 – Прогнозування за тестовими даними

На рисунку 3.21 зображено фрагмент коду, який проводить підготовку навчання моделей.

```
# Підготовка даних
global X, y_test
X, X_scaled, X_train, X_test, y_train, y_test, scaler = prepare_data(df)

# Навчання моделей
lr_model, lr_y_pred, lr_r2 = linear_regression_model(X_train, X_test, y_train, y_test)
dt_model, dt_y_pred, dt_r2 = decision_tree_model(X_train, X_test, y_train, y_test)
knn_model_obj, knn_y_pred, knn_r2 = knn_model(X_train, X_test, y_train, y_test)
```

Рисунок 3.21 – Підготовка на навчання моделей

На рисунку 3.22 можемо побачити результат тестової вибірки за методом «Лінійна регресія».

```
Тестова вибірка:
  R²: 0.9673
  RMSE: 11.49
  MAE: 8.44
  MSE: 132.00
```

Рисунок 3.22 – Результат тестової вибірки за методом “Лінійна регресія”

За результатами можемо бачити, що дана модель має точність 0.96, що означає що вона спрогнозувала 96% даних. А отже, модель можна вважати адекватною. На рисунку 3.23 наведена візуалізація аналізу за обраним методом на тестовій вибірці.

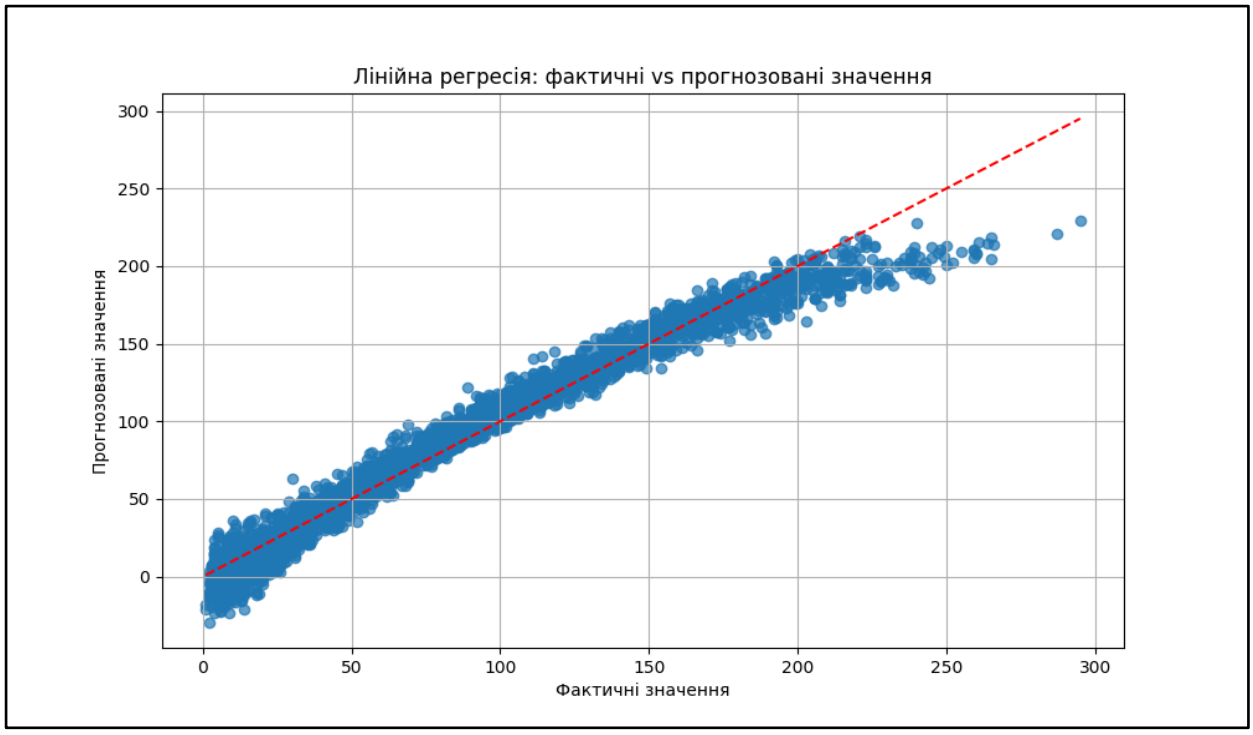

Рисунок 3.23 – Графік візуалізації роботи моделі “Лінійна регресія” на тестових даних

Наступними є результати аналізу за методом «Дерево рішень» (рис 3.24)

```
Тестова вибірка:
  R²: 0.9917
  RMSE: 5.79
  MAE: 3.95
  MSE: 33.48
```

Рисунок 3.24 – Результат тестової вибірки за методом “Дерево рішень”

Дана модель має точність 0.99, а отже вона спрогнозувала 99% даних, тому вона є гарно натренована. На рисунку 3.25 представлено графічну інтерпретацію результатів аналізу, виконаного за методом Дерево рішень на тестовій вибірці.

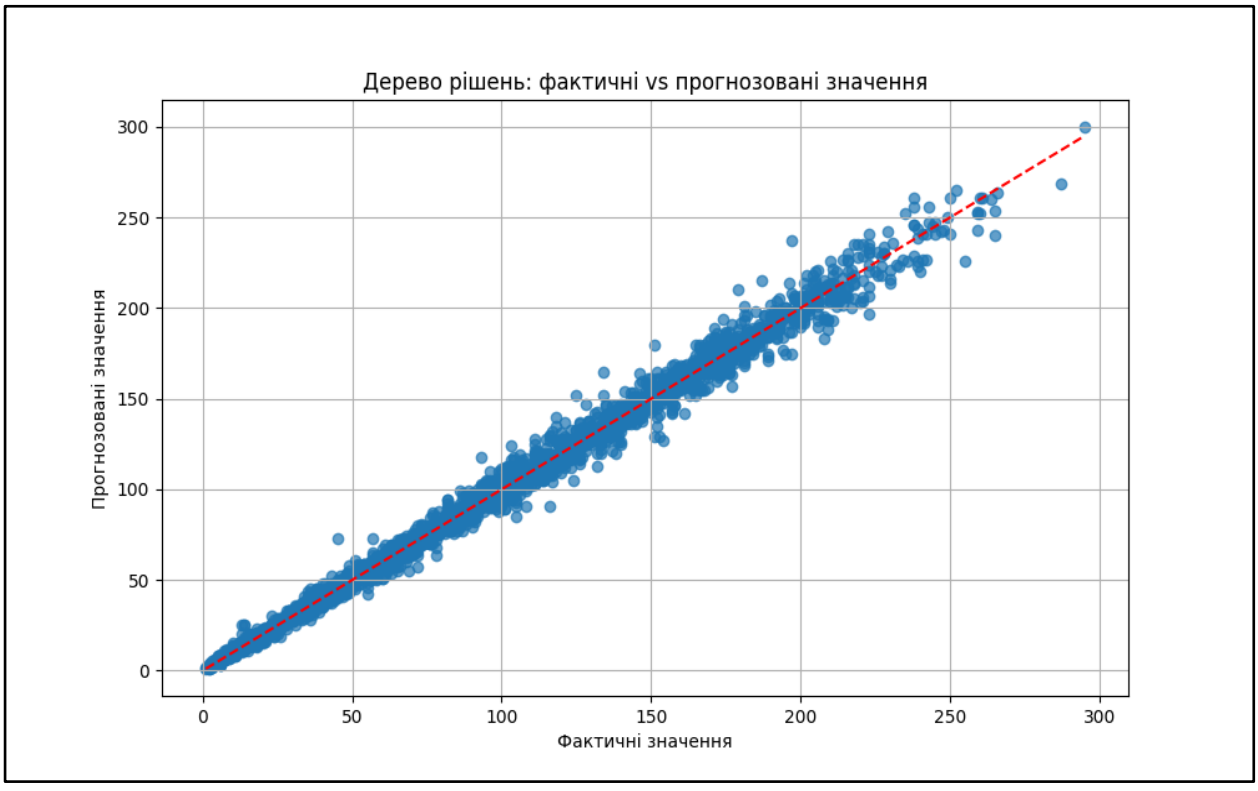

Рисунок 3.25 – Графік візуалізації роботи моделі “Дерево рішень” на тестових даних

На рисунку 3.26 ми можемо побачити результат аналізу методом «k-найближчих сусідів». Саме його результати є найкращими.

```
Тестова вибірка:
  R²: 0.9942
  RMSE: 4.85
  MAE: 3.55
  MSE: 23.51
```

Рисунок 3.26 – Результат тестової вибірки за методом “ k-найближчих сусідів”

За результатами, можемо побачити, що модель має точність 0.99 даних, а отже спрогнозувала 99% тестових даних. На рисунку 3.27 зображено графічну інтерпретацію результатів аналізу, проведеного з використанням обраного методу на тестовій вибірці.

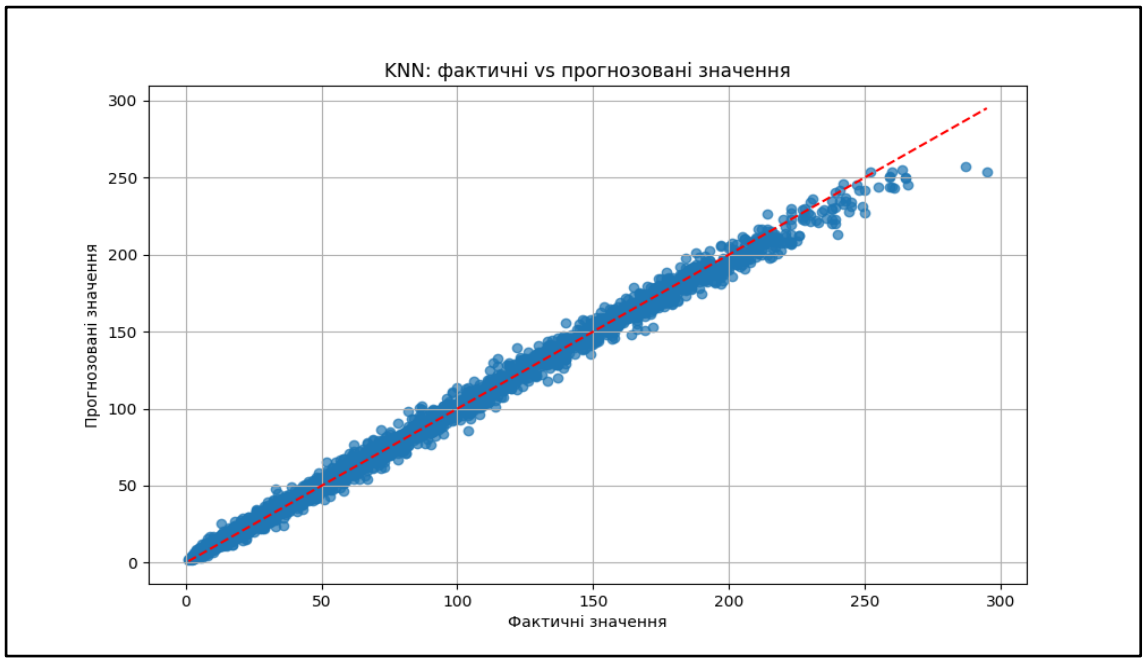

Рисунок 3.27 – Графік візуалізації роботи моделі “ k-найближчих сусідів” на тестових даних

Наступним етапом проведемо порівняння моделей для визначення найбільш ефективної з них (рис 3.28).

```python
# Порівняння моделей
def compare_models(results):
    print("\n6. ПОРІВНЯННЯ МОДЕЛЕЙ\n")

    models = list(results.keys())
    r2_scores = [results[model]['r2'] for model in models]

    # Таблиця результатів
    comparison_df = pd.DataFrame({
        'Модель': models,
        'R²': r2_scores
    }).sort_values(by='R²', ascending=False)

    print("Порівняння моделей за R²:")
    print(comparison_df)
```

Рисунок 3.28 – Порівняння моделей за таблицею

На рисунку 3.29 зображено фрагмент коду, що відповідає за візуалізацію порівняння обраних моделей та створення графіку до них.

```python
# Візуалізація порівняння моделей
plt.figure(figsize=(12, 6))
sns.barplot(x='Модель', y='R²', data=comparison_df)
plt.title('Порівняння моделей за коефіцієнтом детермінації (R²)')
plt.ylim(0, 1)
plt.grid(True, axis='y')
plt.tight_layout()
plt.savefig('model_comparison.png')
plt.close()
```

```python
# Графік прогнозів всіх моделей
plt.figure(figsize=(15, 6))

plt.subplot(1, 3, 1)
plt.scatter(y_test, results['Лінійна регресія']['y_pred'], alpha=0.7)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Фактичні значення')
plt.ylabel('Прогнозовані значення')
plt.title('Лінійна регресія')
plt.grid(True)

plt.subplot(1, 3, 2)
plt.scatter(y_test, results['Дерево рішень']['y_pred'], alpha=0.7)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Фактичні значення')
plt.ylabel('Прогнозовані значення')
plt.title('Дерево рішень')
plt.grid(True)

plt.subplot(1, 3, 3)
plt.scatter(y_test, results['KNN']['y_pred'], alpha=0.7)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Фактичні значення')
plt.ylabel('Прогнозовані значення')
plt.title('KNN')
plt.grid(True)

plt.tight_layout()
plt.savefig('all_models_comparison.png')
plt.close()
```

Рисунок 3.29 – Візуалізація порівняння моделей та створення графіків

На рисунку 3.30 представлено результат візуалізації порівняння моделей у форматі порівняльної таблиці.

```
Модель: KNN
  Тестова R²: 0.9942
  Тренувальна R²: 0.9947
  Тестова RMSE: 4.85
  CV R²: 0.9928 ± 0.0004
  Різниця R² (train-test): 0.0006
  Статус: Збалансована модель

Модель: Дерево рішень
  Тестова R²: 0.9917
  Тренувальна R²: 0.9961
  Тестова RMSE: 5.79
  CV R²: 0.9902 ± 0.0004
  Різниця R² (train-test): 0.0043
  Статус: Збалансована модель

Модель: Лінійна регресія
  Тестова R²: 0.9673
  Тренувальна R²: 0.9672
  Тестова RMSE: 11.49
  CV R²: 0.9670 ± 0.0008
  Різниця R² (train-test): -0.0001
  Статус: Можливе недонавчання
```

Рисунок 3.30 – Таблиця порівняння усіх моделей

За результатами тестування робимо висновки, що найкращим методом є “Метод k-найближчих сусідів”. Він найбільш точно спрогнозував дані за двома вибірками, а саме за тестовими та тренувальними даними (рис 3.31).

```
НАЙКРАЩА МОДЕЛЬ: KNN
Тестова R²: 0.9942
```

Рисунок 3.31 – Результат порівняння

На рисунку 3.32 зображено візуалізацію у форматі графіку із порівнянням усіх моделей за метриком R^2_score.

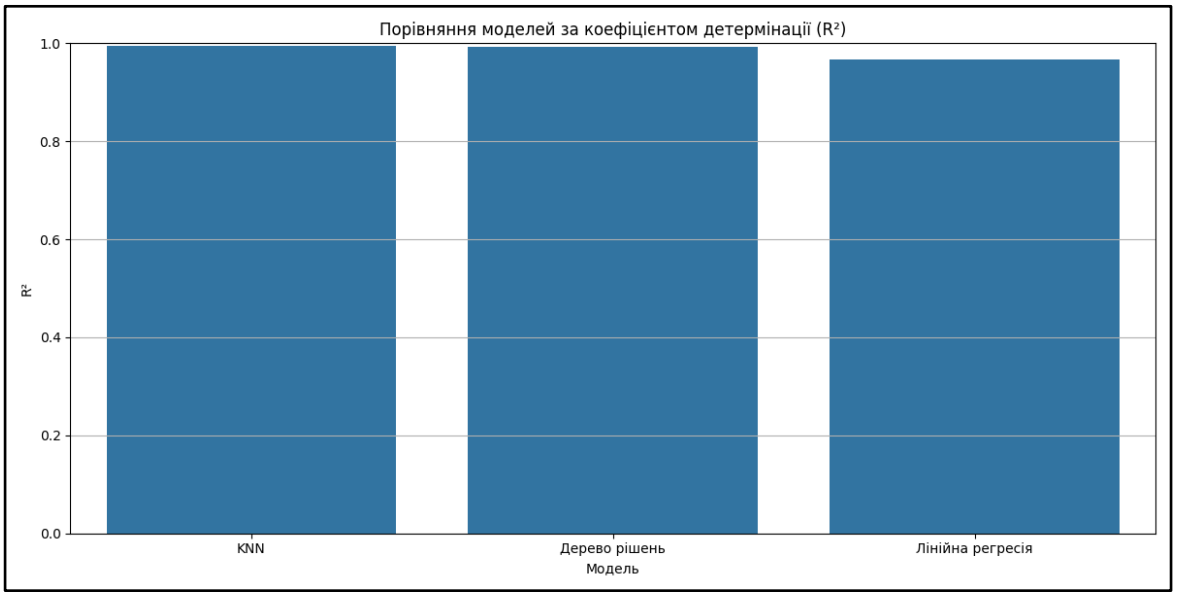

Рисунок 3.32 – Візуалізація порівняння усіх моделей

На рисунку 3.33 зображено графік важливості ознак за найкращим методом, а саме “Метод k-найближчих сусідів”.

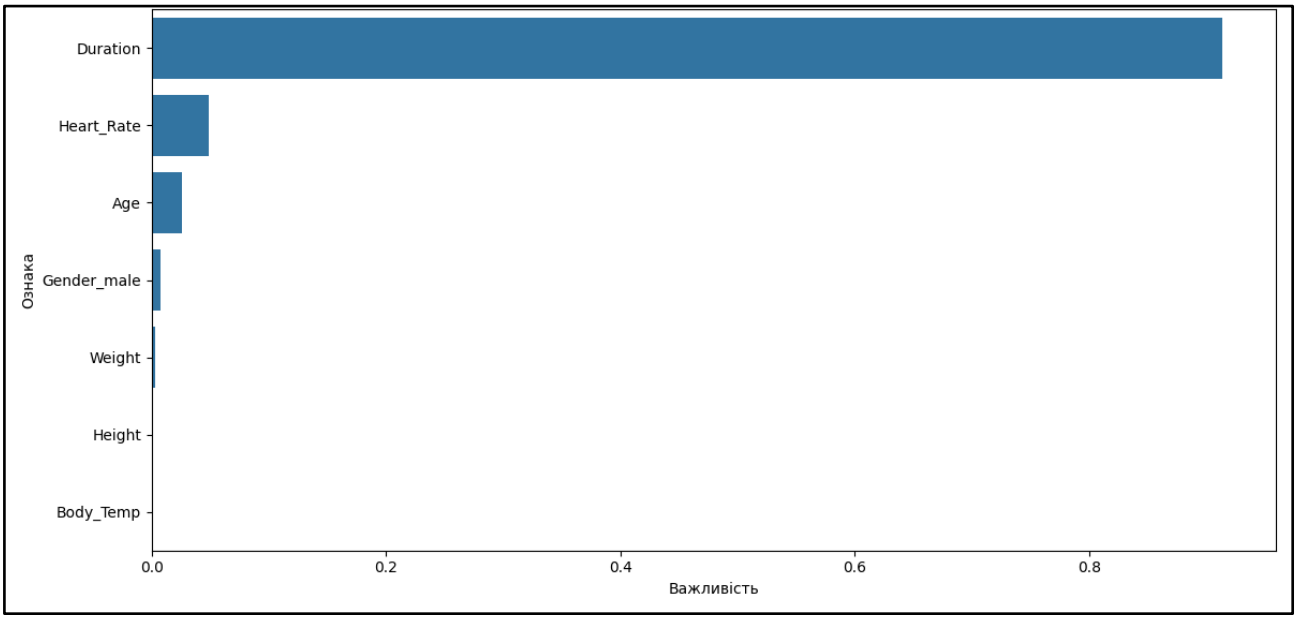

Рисунок 3.33 – Важливість ознак за найкращим методом “ k-найближчих сусідів”

За графіком можемо зробити висновки, що показник Duration має найбільшу важливість – приблизно 0,90. Це свідчить про те, що саме ця ознака є

домінуючим фактором у визначенні кількості спалених калорій. Такий результат є логічним, адже чим довше триває фізична активність, тим більше енергії витрачає організм. Heart_Rate (частота серцевих скорочень) посіла друге місце за важливістю, із значенням близько 0.06. Це означає, що високий пульс під час тренування також пов’язаний зі зростанням витрати калорій, хоча його вплив значно менший, ніж у Duration. Age (Вік) має дещо нижчий, але все ще помітний вплив, що може бути пов’язано з фізіологічними особливостями обміну речовин у людей різного віку та ваги, а також із різним рівнем навантаження та часу тренування.

3.4. Висновки

У цьому розділі розроблено інформаційну технологію для прогнозування кількості спалених калорій під час тренувань. Застосовано три моделі машинного навчання (а саме: Лінійна регресія, Дерево рішень, Метод найближчих сусідів) з метою прогнозування кількості спожитих калорій під час фізичних вправ.

Процес навчання моделей здійснювався на навчальному наборі даних, а якість їх роботи оцінювалась за такими метриками: середньоквадратична похибка (MSE), середньоабсолютна похибка (MAE) та коефіцієнт детермінації (R^2_score).

Після завершення навчання моделей їх було протестовано на тестовому наборі даних для оцінки їхньої адекватності, здатності до узагальнення й стійкості до перенавчання. На основі отриманих результатів було визначено найефективнішу модель.

Порівняльний аналіз показав, що модель k-найближчих сусідів продемонструвала найкращі результати у вирішенні задачі прогнозування калорій, що споживаються. Вона вирізняється високою точністю прогнозів і

стійкістю до перенавчання. Значення коефіцієнта детермінації (R^2_score) склали 0.99 для навчального та тестового набору даних, що підтверджує ефективність і надійність цієї моделі.

ВИСНОВКИ

У ході виконання бакалаврської кваліфікаційної роботи на тему «Інформаційна технологія аналізу та прогнозування кількості спалених калорій під час тренувань» було здійснено глибокий аналіз предметної галузі, присвяченої вивченню фізіологічних процесів та їх взаємозв’язку з енергетичними витратами під час фізичних навантажень. Особливу увагу було приділено вивченню чинників, які суттєво впливають на кількість спалених калорій: вік, вага, зріст, пульс, температура тіла, тривалість тренування та інші параметри.

Проаналізовано застосування сучасних інформаційних технологій у фітнесі та медичній сфері, з акцентом на алгоритми машинного навчання, що дають змогу точно прогнозувати кількість спалених калорій.

У другому розділі роботи проведено аналітичну обробку даних, побудовано гістограми розподілу числових змінних, зокрема таких як вік, зріст, вага, пульс, температура тіла, тривалість тренування та кількість спалених калорій. У процесі підготовки даних для моделювання (інженерії ознак) категоріальні змінні (наприклад, стать) були перетворені у числові формати, що дозволило забезпечити коректну роботу моделей машинного навчання. Надалі було відібрано найбільш інформативні ознаки для побудови моделі прогнозування кількості спалених калорій.

На наступному етапі було реалізовано створення та первинну апробацію моделей машинного навчання без параметричної оптимізації. Для порівняння результатів були використані: лінійна регресія, дерево рішень та метод найближчих сусідів.

У третьому розділі бакалаврської кваліфікаційної роботи розроблено інформаційну технологію для прогнозування кількості спалених калорій під час тренувань, яка базується на алгоритмах машинного навчання, а саме: Лінійна

регресія (Linear Regression), Дерево рішень (Decision Tree), K-найближчих сусідів (K- Nearest Neighbors).

Здійснено остаточне моделювання з використанням зазначених алгоритмів. Моделі тренувалися на навчальному наборі даних, а оцінювання їх ефективності здійснювалося за метриками MAE (mean absolute error), MSE (mean squared error) та R^2_score (коефіцієнт детермінації).

Після етапу тренування моделі було протестовано на тестовому наборі даних з метою оцінки їх узагальнювальної здатності. У результаті аналізу встановлено, що модель k-найближчих сусідів (K-Nearest Neighbors) забезпечує найкращі показники точності прогнозування кількості спалених калорій, водночас демонструючи високу стійкість до перенавчання. Значення R^2_score для цієї моделі склали 0.99 на тренувальних даних та тестових, що свідчить про її надійність і ефективність.

Таким чином, модель k-найближчих сусідів рекомендовано як оптимальне рішення для задач аналізу та прогнозування кількості спалених калорій під час тренувань, де ключовими є точність оцінки та стійкість результатів у практичному застосуванні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Малоштан Л. М., Рядних О. К., Жегунова Г. П. Фізіологія з основами анатомії людини : підруч. для студентів ВНЗ. Харків : Вид-во НФаУ; Золоті сторінки, 2003. 432 с. URL: http://dspace.nuph.edu.ua/handle/123456789/22328 (дата звернення: 5.05.2025).

2. Mifflin M. D., St Jeor S. T., Hill L. A. A new predictive equation for resting energy expenditure in healthy individuals. *American Journal of Clinical Nutrition*. 2000. Vol. 51, № 2. P. 241–247. URL: https://doi.org/10.1093/ajcn/51.2.241 (дата звернення: 5.05.2025).

3. Євлаша В. В., Потапова В. О., Радченка М. І., Савицької Н. Л. Повноцінне харчування: інноваційні аспекти технологій, енергоефективного виробництва, зберігання та маркетингу: монографія. Харків: ХДУХТ, 2016. 320 с. URL: https://localhost:4000/bitstreams/9af00e1c-8717-4bcb-903e-4cac734f0674/download (дата звернення: 7.05.2025).

4. Симоненко Т. І. Диференційна психофізіологія людини: навч. посіб. – Київ: ІПООД імені Івана Зязюна НАПН України, 2023. 174 с. URL: https://lib.iitta.gov.ua/id/eprint/733601/1/%D0%94%D0%B8%D1%84%D0%B5%D1%80%D0%B5%D0%BD%D1%86%D1%96%D0%B9%D0%BD%D0%B0%20%D0%BF%D1%81%D0%B8%D1%85%D0%BE%D0%BB%D0%BE%D0%BB%D0%B3%D1%96%D1%8F.pdf?utm_source=chatgpt.com (дата звернення: 10.05.2025).

5. Beatty A. L., Iribarren C., Kazi D. S. et al. Physical Activity Measured by a Wearable Device and the Risk of Cardiovascular Events: The Women's Health Study. *JAMA Network Open*. 2020. Vol. 3, № 9. e2011875. URL: https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2770184 (дата звернення: 17.05.2025).

6. Guyton A. C., Hall J. E. Textbook of Medical Physiology. 13th ed. Philadelphia : Elsevier, 2016. 1168 p. URL:

https://clinref.com/data/uploads/books/Guyton_and_Hall_Textbook_of_Medical_Physiology_14th_Ed.pdf?utm_source=chatgpt.com (дата звернення: 18.05.2025).

7. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2nd ed. Sebastopol : O'Reilly Media, 2019. 819 p. URL: https://archive.org/details/handson-machine-learning-with-scikit-2-e-1/page/n5/mode/2up?utm_source=chatgpt.com (дата звернення: 19.05.2025).

8. Кривуля Г. М. Інтелектуальний аналіз даних. Львів : Видавництво Львівської політехніки, 2017. 324 с. URL: https://ena.lpnu.ua:8443/server/api/core/bitstreams/e037c9a0-3623-435b-8e0d-f1c1fb09d240/content (дата звернення: 20.05.2025).

9. WHO. Global Recommendations on Physical Activity for Health. Geneva : World Health Organization, 2010. URL: https://www.who.int/publications/i/item/9789241599979 (дата звернення: 20.05.2025).

10. Calories Burnt Prediction Dataset. URL: https://www.kaggle.com/datasets/ruchikakumbhar/calories-burnt-prediction

11 . Tamura T., Maeda Y., Sekine M., Yoshida M. Wearable Photoplethysmographic Sensors – Past and Present. Electronics. 2014. Vol. 3, № 2. P. 282–302. DOI: https://doi.org/10.3390/electronics3020282 (дата звернення: 24.05.2025).

12. Bent B., Goldstein B. A., Kibbe W. A. et al. Investigating sources of inaccuracy in wearable optical heart rate sensors. *NPJ Digital Medicine*. 2020. Vol. 3. Article No. URL: https://doi.org/10.1038/s41746-020-0226-6 (дата звернення: 18.05.2025).

13. Zhang Y., Liu S., Zhang J. et al. Personalized Health Monitoring System Based on Smart Wearable Devices. *Healthcare*. 2021. Vol. 9, № 3. 288. URL: https://doi.org/10.3390/healthcare9030288 (дата звернення: 21.05.2025).

14. Roy N., Misra A., Cook D. J. et al. Infrastructure-assisted smartphone-based

activity recognition for real-time applications. *IEEE Transactions on Mobile Computing*. 2017. Vol. 17, № 2. P. 474–487. URL: https://doi.org/10.1109/TMC.2017.2706260 (дата звернення: 27.05.2025).

15. Apple Inc. Apple Watch Series 8 Health Features. Офіційний сайт. URL: https://www.apple.com/ua/apple-watch-series-8/ (дата звернення: 27.05.2025)

16. European Commission. Horizon 2020 Projects on Personalised Health Monitoring. URL: https://cordis.europa.eu/project/id/731391 (дата звернення: 23.05.2025)

17. Lane N. D., Georgiev P., Qendro L. DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments. *Proceedings of the ACM Conference on Ubiquitous Computing*. 2015. P. 283–294. URL: https://www.repository.cam.ac.uk/items/4dc23b3d-48e0-4390-9172-8ba16aa19920 (дата звернення: 20.05.2025)

18. Bishop C. M. Pattern Recognition and Machine Learning. New York: Springer, 2006. 738 p. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf (дата звернення: 20.05.2025)

19. Мокін В. Б., Дратований М. В. Наука про дані: машинне навчання та інтелектуальний аналіз даних. Навчальний посібник. Вінниця: ВНТУ, 2024. 263 с. URL: https://iq.vntu.edu.ua/repository/getfile.php/8163.pdf (дата звернення 23.05.2025)

20. Іванчук Я. В., Месюра В. І., Яровий А. А., Манжілевський О. Д. Інтелектуальний аналіз даних та машинне навчання. Частина 1. Базові методи та засоби аналізу даних. Навчальний посібник. Вінниця: ВНТУ, 2021. 69 с. URL: https://press.vntu.edu.ua/index.php/vntu/catalog/download/673/1186/2394-1?inline=1 (дата звернення 23.05.2025)

ДОДАТОК А

(обов’язковий)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

_____ д.т.н., проф. Віталій МОКІН

“____”_____ 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на бакалаврську кваліфікаційну роботу

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ КІЛЬКОСТІ СПАЛЕНИХ КАЛОРІЙ ПІД ЧАС ТРЕНУВАНЬ

08-34.БКР.024 00.000 ТЗ

Керівник: Ph. D., ст. викл. кафедри САІТ

_____ Ольга ВОЙЦЕХОВСЬКА

«__» _____ 2025 р.

Розробила студентка гр. 2ІСТ-21б

_____ Анастасія ТРУСЮК

«__» _____ 2025 р.

Вінниця 2025

1. Підстава для проведення робіт.
Підставою для виконання роботи є наказ №__ по ВНТУ від «__» _____2025р., та індивідуальне завдання на БКР, затверджене протоколом №__ засідання кафедри САІТ від «__» _____ 2025р.
2. Джерела розробки:
1) Calories Burnt Prediction using Machine Learning Algorithms. *Guinness Press Journal of Artificial Intelligence & Machine Learning.* 2024. Volume 1, Issue 1, PP. 29-36.URL: https://guinnesspress.org/articles/666bea5c04ca0-cie-3-52446-a.pdf;
2) Mifflin M. D., St Jeor S. T., Hill L. A. A new predictive equation for resting energy expenditure in healthy individuals. *American Journal of Clinical Nutrition*. 2000. Vol. 51, № 2. P. 241–247. URL: https://doi.org/10.1093/ajcn/51.2.241
3. Мета і призначення роботи. Метою дослідження є підвищення точності прогнозування кількості спалених калорій під час тренування шляхом створення інформаційної технології та використання методів машинного навчання.
4. Вихідні дані для проведення робіт: Kaggle Dataset «Calories Burnt Prediction Dataset». URL: https://www.kaggle.com/datasets/ruchikakumbhar/calories-burnt-prediction .
5. Методи дослідження: розвідувальний аналіз даних, інженерія ознак, методи машинного навчання, розробка UML-діаграм.
6. Етапи роботи і терміни їх виконання:
a) Загальна характеристика об'єкту дослідження _____ – _____
b) Аналіз аналогічних методів та інформаційних технологій _____ – _____
c) Вибір оптимального рішення та налаштувань для розв'язання поставленої задачі _____ – _____
d) Розроблення інформаційної технології _____ – _____
e) Оформлення матеріалів до захисту БКР _____ – _____
6. Очікувані результати та порядок реалізації
Отримання інформаційної технології для прогнозування кількості спалених калорій під час тренувань.
8. Вимоги до розробленої документації
Текстова та ілюстративна частини роботи оформлені у відповідності до вимог «Методичних вказівок до виконання бакалаврських кваліфікаційних робіт для студентів спеціальностей: 124 «Системний аналіз», 126 «Інформаційні системи та технології» (освітня програма «Прикладні інформаційні технології»)».
9. Порядок приймання роботи
Публічний захист «___» _____ 2025 р.
Початок розробки «___» _____ 2025 р.
Граничні терміни виконання БКР «___» _____ 2025 р.

Розробила студентка групи 2ІСТ-21б _____ Анастасія ТРУСЮК

ДОДАТОК Б

(обов’язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: «Інформаційна технологія аналізу та прогнозування кількості спалених калорій під час тренувань»

Тип роботи: бакалаврська кваліфікаційна робота

Підрозділ: кафедра САІТ, ФІІТА, гр. 2ІСТ-21б

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 1.19%

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне):

■ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

□ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

□ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Віталій МОКІН, зав. каф. САІТ _____
(підпис)

Євгеній КРИЖАНОВСЬКИЙ, доц. каф. САІТ _____
(підпис)

Особа, відповідальна за перевірку _____ Сергій ЖУКОВ
(підпис)

З висновком експертної комісії ознайомлений(-на)

Керівник _____ Ольга ВОЙЦЕХОВСЬКА
(підпис)

Здобувач _____ Анастасія ТРУСЮК
(підпис)

ДОДАТОК В

(довідниковий)

Фрагмент лістингу програми

Код файлу інформаційної технології:

```
# Завантаження даних
def load_data(filename):
    try:
        df = pd.read_csv(/kaggle/input/calories-burnt-prediction/calories.csv)
        print(f"Дані успішно завантажено. Розмір датасету: {df.shape}")
        return df
    except Exception as e:
        print(f"Помилка при завантаженні даних: {e}")
        return None
# Дослідницький аналіз даних
def exploratory_analysis(df):
    print("\n1. ДОСЛІДНИЦЬКИЙ АНАЛІЗ ДАНИХ\n")
    # Базова інформація про датасет
    print("Базова інформація про датасет:")
    print(df.info())
    print("\nСтатистика по числовим полям:")
    print(df.describe())
    print("\nПриклад рядків даних:")
    print(df.head())
    print("\n...")
    print(df.tail())
```

```
    # Перевірка на пропущені значення
    print("\nПеревірка на пропущені значення:")
    print(df.isnull().sum())
    # Створюємо копію даних для аналізу без категоріальних змінних
    df_numeric = df.copy()
    # Перетворюємо категоріальні дані (стать) у числові
    if 'Gender' in df_numeric.columns:
        df_numeric = pd.get_dummies(df_numeric, columns=['Gender'], drop_first=True)
# Видаляємо нечислові стовпці (User_ID)
    if 'User_ID' in df_numeric.columns:
        df_numeric = df_numeric.drop('User_ID', axis=1)
    # Кореляційна матриця (тільки для числових даних)
    plt.figure(figsize=(12, 10))
    correlation_matrix = df_numeric.corr()
    sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
    plt.title('Кореляційна матриця')
    plt.tight_layout()
    plt.savefig('correlation_matrix.png')
    plt.close()
    # Гістограми розподілу числових змінних
    df_numeric.hist(figsize=(15, 10), bins=20)
    plt.suptitle('Розподіл числових змінних')
    plt.tight_layout()
    plt.subplots_adjust(top=0.9)
    plt.savefig('histograms.png')
    plt.close()
```

```
 # Діаграма розсіювання для найбільш корельованих змінних з Calories
    corr_with_calories = correlation_matrix['Calories'].sort_values(ascending=False)
    print("\nКореляція змінних з Calories:")
    print(corr_with_calories)
    top_features = corr_with_calories[1:4].index.tolist()  # Топ 3 корельованих змінних (виключаючи саму 'Calories')
    plt.figure(figsize=(15, 5))
    for i, feature in enumerate(top_features):
        plt.subplot(1, 3, i + 1)
        sns.scatterplot(x=feature, y='Calories', data=df_numeric)
        plt.title(f'{feature} vs Calories')
    plt.tight_layout()
    plt.savefig('scatter_plots.png')
    plt.close()
    # Додатково: аналіз залежності калорій від статі
    if 'Gender' in df.columns:
        plt.figure(figsize=(8, 6))
        sns.boxplot(x='Gender', y='Calories', data=df)
        plt.title('Залежність калорій від статі')
        plt.savefig('gender_calories.png')
        plt.close()
# Підготовка даних для моделювання
def prepare_data(df):
    print("\n2. ПІДГОТОВКА ДАНИХ ДЛЯ МОДЕЛЮВАННЯ\n")
    # Створюємо копію даних
    df_model = df.copy()
    # Видаляємо стовбчик User_ID, якщо він є
```

```
    if 'User_ID' in df_model.columns:
        df_model = df_model.drop('User_ID', axis=1)
    # Кодування категоріальних змінних
    if 'Gender' in df_model.columns:
        df_model = pd.get_dummies(df_model, columns=['Gender'], drop_first=True)
    # Вибір ознак та цільової змінної
    X = df_model.drop('Calories', axis=1)
    y = df_model['Calories']
    # Масштабування ознак
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
    # Розділення даних на тренувальну та тестову вибірки
    X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
    print(f"Розмір тренувальної вибірки: {X_train.shape}, Розмір тестової вибірки: {X_test.shape}")
    return X, X_scaled, X_train, X_test, y_train, y_test, scaler
# Лінійна регресія
def linear_regression_model(X_train, X_test, y_train, y_test):
    print("\n3. ЛІНІЙНА РЕГРЕСІЯ\n")
    # Навчання моделі
    model = LinearRegression()
    model.fit(X_train, y_train)
    # Прогнозування
    y_pred = model.predict(X_test)
    # Оцінка моделі
    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
```

```
    r2 = r2_score(y_test, y_pred)
    print(f"Середньоквадратична помилка (MSE): {mse:.2f}")
    print(f"Корінь середньоквадратичної помилки (RMSE): {rmse:.2f}")
    print(f"Коефіцієнт детермінації (R²): {r2:.2f}")
    # Крос-валідація
    cv_scores = cross_val_score(model, X_train, y_train, cv=5, scoring='r2')
    print(f"Крос-валідація (R²): {cv_scores.mean():.2f} ± {cv_scores.std():.2f}")
    # Коефіцієнти моделі
    coef_df = pd.DataFrame({
        'Ознака': X.columns,
        'Коефіцієнт': model.coef_
    }).sort_values(by='Коефіцієнт', ascending=False)
    print("\nКоефіцієнти моделі:")
    print(coef_df)
# Візуалізація прогнозів
    plt.figure(figsize=(10, 6))
    plt.scatter(y_test, y_pred, alpha=0.7)
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
    plt.xlabel('Фактичні значення')
    plt.ylabel('Прогнозовані значення')
    plt.title('Лінійна регресія: фактичні vs прогнозовані значення')
    plt.grid(True)
    plt.savefig('linear_regression_plot.png')
    plt.close()
    # Візуалізація коефіцієнтів
    plt.figure(figsize=(12, 6))
    top_coef = coef_df.iloc[:10]
```

```
sns.barplot(x='Коефіцієнт', y='Ознака', data=top_coef)
plt.title('Топ-10 найважливіших ознак (Лінійна регресія)')
plt.tight_layout()
plt.savefig('linear_regression_coef.png')
plt.close()
return model, y_pred, r2
```

ДОДАТОК Г

(обов’язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АНАЛІЗУ ТА ПРОГНОЗУВАННЯ СПАЛЕННЯ КАЛОРІЙ ПІД ЧАС ТРЕНУВАННЯ

Нормоконтроль: к.т.н., доцент

_____ Сергій ЖУКОВ

«___» _____ 2025 р.

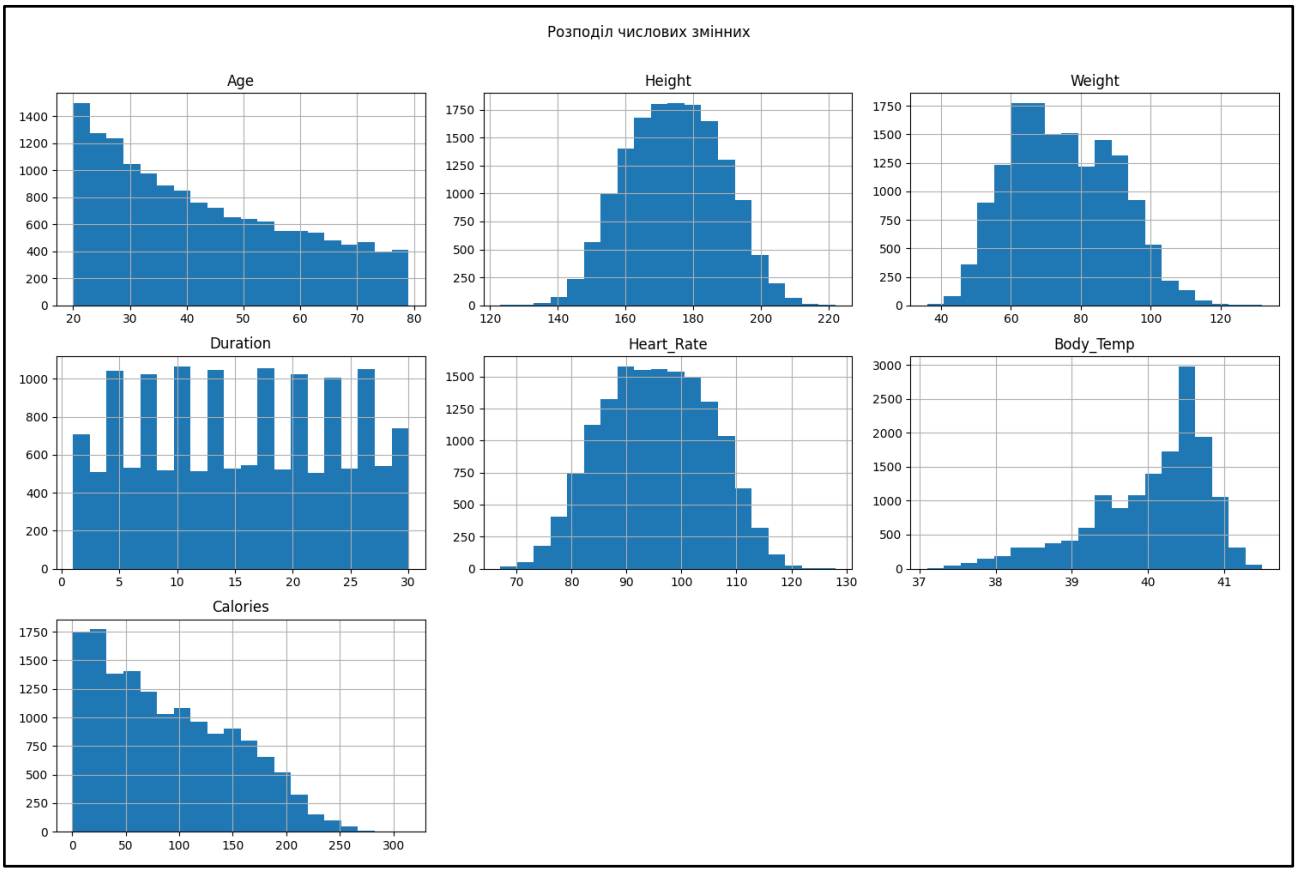

Рисунок Г.1 – Гістограми розподілу даних

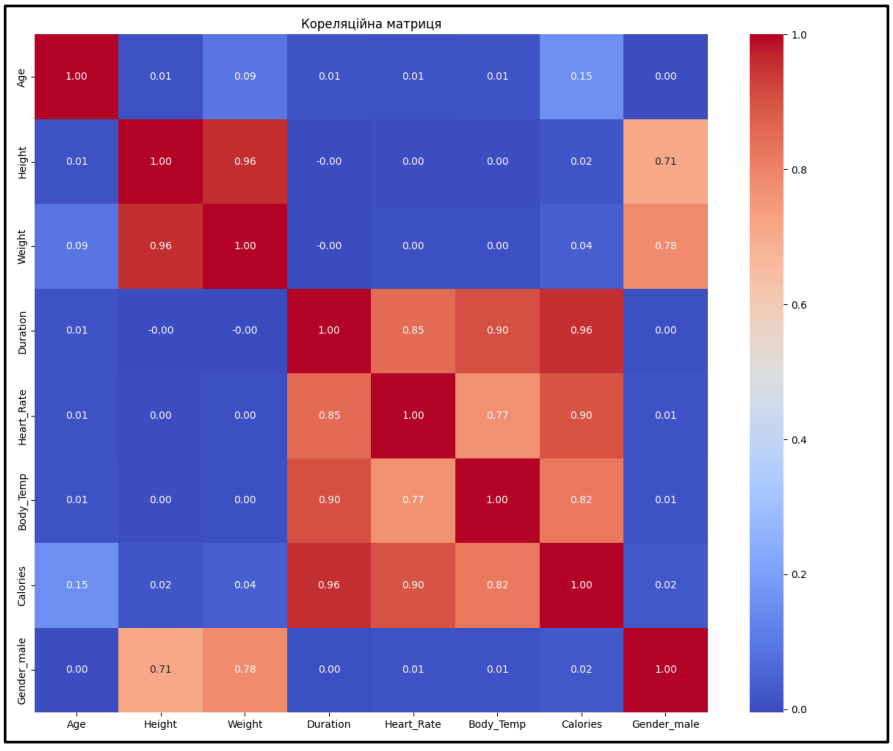

Рисунок Г.2 – Кореляційна матриця

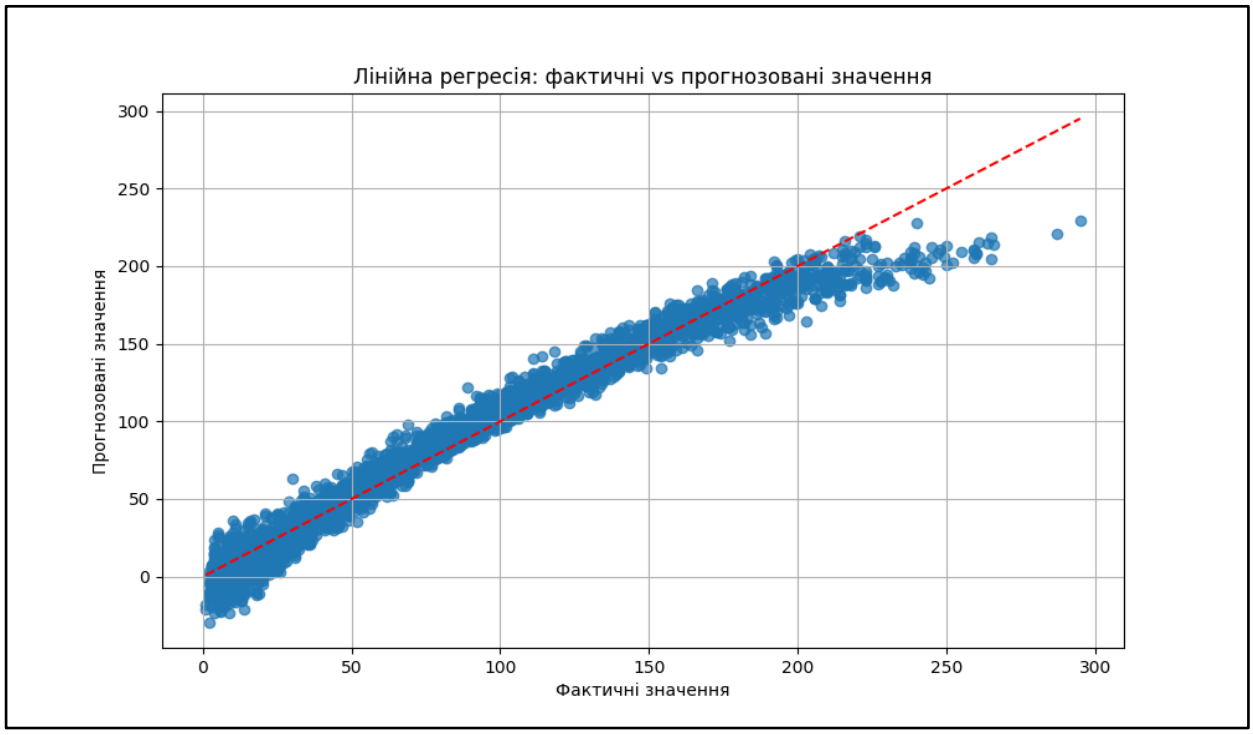

Рисунок Г.3 – Графік візуалізації роботи моделі “Лінійна регресія” на тестових даних

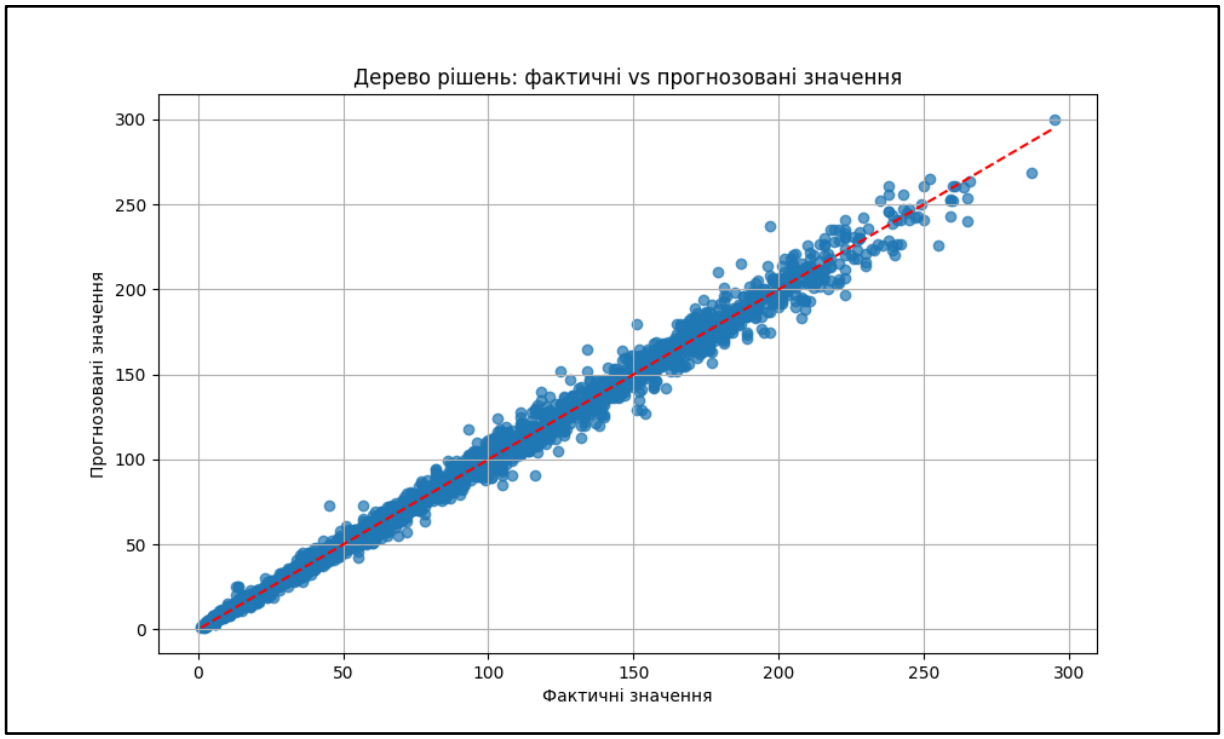

Рисунок Г.4 – Графік візуалізації роботи моделі “Дерево рішень” на тестових даних

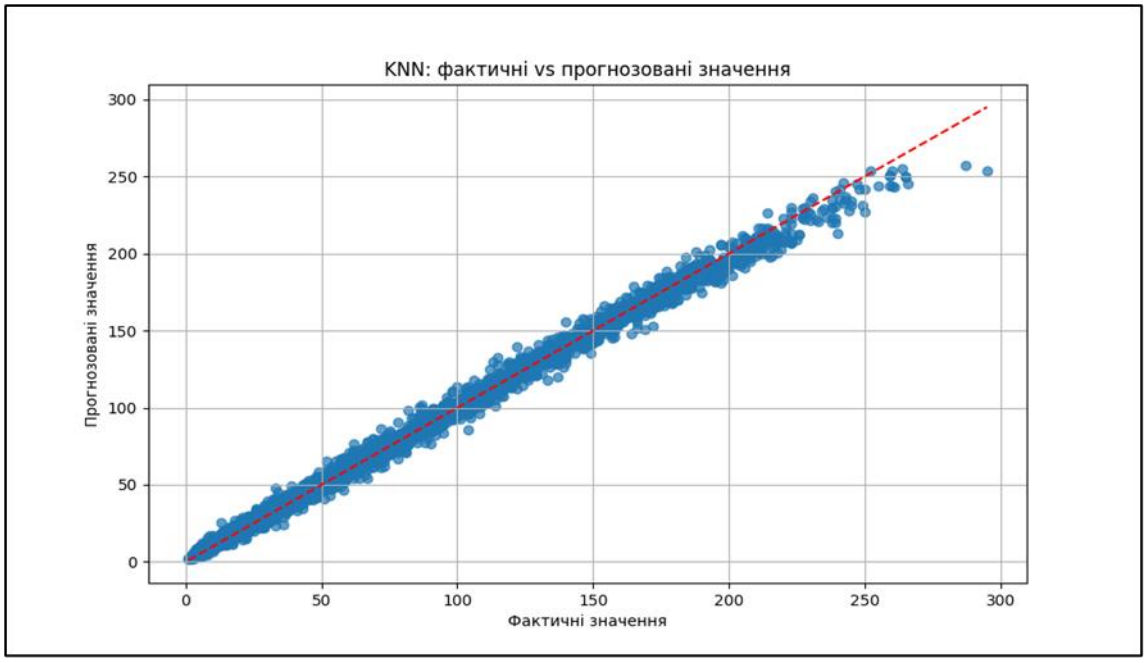

Рисунок Г.5 – Графік візуалізації роботи моделі “ k-найближчих сусідів” на тестових даних

```
Модель: KNN
  Тестова R²: 0.9942
  Тренувальна R²: 0.9947
  Тестова RMSE: 4.85
  CV R²: 0.9928 ± 0.0004
  Різниця R² (train-test): 0.0006
  Статус: Збалансована модель

Модель: Дерево рішень
  Тестова R²: 0.9917
  Тренувальна R²: 0.9961
  Тестова RMSE: 5.79
  CV R²: 0.9902 ± 0.0004
  Різниця R² (train-test): 0.0043
  Статус: Збалансована модель

Модель: Лінійна регресія
  Тестова R²: 0.9673
  Тренувальна R²: 0.9672
  Тестова RMSE: 11.49
  CV R²: 0.9670 ± 0.0008
  Різниця R² (train-test): -0.0001
  Статус: Можливе недонавчання
```

Рисунок Г.6 – Таблиця порівняння точності усіх моделей