Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра прикладної математики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Математичні моделі та методи машинного навчання для медичних прогнозів
(тема)

Виконав:
здобувач _2_ року навчання, групи _САУм-23-1_
Гета Д.В.
(прізвище, ініціали)
Спеціальність
124 Системний аналіз
(код і повна назва спеціальності)
Тип програми _освітньо-професійна_
(освітньо-професійна або освітньо-наукова)
Освітня програма
Системний аналіз і управління
(повна назва освітньої програми)
Керівник доц. Єсілевський В.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ПМ _____ Сидоров М.В.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _інформаційно-аналітичних технологій та менеджменту
Кафедра _прикладної математики
Рівень вищої освіти _другий (магістерський)
Спеціальність _124 Системний аналіз
(код і повна назва)
Тип програми _освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма _Системний аналіз і управління
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри ПМ _____
(підпис)
“ _25_ ” _листопада_ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _Геті Дмитру Вікторовичу
(прізвище, ім’я, по батькові)

1. Тема роботи _Математичні моделі та методи машинного навчання для медичних прогнозів_
затверджена наказом по університету від 22 листопада 2024 р. № 1228 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 6 січня 2025 р.

3. Вихідні дані до роботи _математичні методи машинного навчання, методи оцінки результатів лікування, тестування моделей та програмне забезпечення для реалізації аналізу та моделювання

4. Перелік питань, що потрібно опрацювати в роботі
1. Системний аналіз предметної області
2. Вибір і обґрунтування методу розв’язання
3. Програмна реалізація
4. Результати обчислювального експерименту
5. Аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

1. Актуальність теми роботи
2. Постановка задачі
3. Системний аналіз предметної області
4. Метод чисельного аналізу
5. Результати обчислювального експерименту

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Підбір та вивчення технічної літератури за темою роботи	25 листопада – 1 грудня 2024 р.	виконано
2	Вибір та обґрунтування методу	2 – 8 грудня 2024 р.	виконано
3	Розробка алгоритму і програми	9 – 22 грудня 2023 р.	виконано
4	Проведення аналітичних досліджень та розрахунків	23 – 29 грудня 2024 р.	виконано
5	Робота над текстом пояснювальної записки	30 грудня 2024 р. – 9 січня 2025 р.	виконано
6	Представлення роботи на рецензію в ЕК	10 січня 2025 р.	виконано

Дата видачі завдання _25_ листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Єсілевський В.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 61 с., 6 табл., 19 рис., 1 дод., 15 джерел.

МАШИННЕ НАВЧАННЯ, КЛАСИФІКАЦІЙНІ МЕТОДИ, НЕЙРОННІ МЕРЕЖІ, МЕДИЧНІ ПРОГНОЗИ, ПЕРЕДБАЧЕННЯ СТАНУ ПАЦІЄНТА, АНАЛІЗ ДАНИХ, МАТЕМАТИЧНІ МОДЕЛІ.

Об’єкт дослідження – алгоритми машинного навчання для медичних прогнозів.

Мета роботи – аналіз методів і моделей машинного навчання для прогнозування медичних станів.

Методи дослідження – класифікаційні алгоритми, нейронні мережі.

Проаналізовані методи машинного навчання для прогнозування стану здоров’я пацієнтів. Було досліджено математичні моделі прогнозування медичних даних. Виконаний детальний аналіз різних класифікаційних алгоритмів та нейронних мереж, і зроблено висновок щодо найбільш ефективного підходу. Було проведено аналіз медичних даних одного з клінічних випадків.

ABSTRACT

Introductory note: 61 pages, 6 tables, 19 figures, 1 appendixes, 15 sources.

MACHINE LEARNING, CLASSIFICATION METHODS, NEURAL NETWORKS, MEDICAL PREDICTIONS, PATIENT CONDITION FORECASTING, DATA ANALYSIS, MATHEMATICAL MODELS.

Object of research – machine learning algorithms for medical predictions.

Objective – to analyze methods and models of machine learning for predicting medical conditions.

Methods of research – classification algorithms, neural networks.

Machine learning methods for predicting patient health conditions were analyzed. Mathematical models for forecasting medical data were studied. A detailed analysis of various classification algorithms and neural networks was conducted, leading to conclusions about the most effective approach. An analysis of medical data from a clinical case was performed.

ЗМІСТ

ВСТУП

Актуальність теми. Актуальність роботи зумовлена зростаючим попитом на вивчення прогнозування медичних діагнозів в сучасній медицині та біоінформатиці. Це дослідження охоплює різноманітні наукові та практичні аспекти, такі як медична діагностика, прогнозування ризиків для здоров'я, розробка індивідуальних планів лікування та профілактики захворювань. Оскільки медичні дані є гетерогенними та складними за своєю природою, аналіз таких даних вимагає використання передових методів обробки інформації.

Прогнозування медичних діагнозів на основі машинного навчання є важливим напрямом досліджень, який дозволяє не лише покращити точність діагностики, але й вчасно вживати профілактичні заходи. Розробка точних та надійних моделей для прогнозування є викликом через наявність великої кількості взаємопов'язаних чинників, таких як генетичні дані, результати лабораторних досліджень, історія хвороб, спосіб життя та інші. Це дослідження є важливим як для систем охорони здоров'я, так і для поліпшення якості життя пацієнтів.

Мета і завдання кваліфікаційної роботи. Метою кваліфікаційної роботи є розробка ефективного методу та алгоритму для прогнозування медичних діагнозів на основі машинного навчання, що дозволить підвищити точність діагностики та покращити процес прийняття клінічних рішень. Для досягнення поставленої мети необхідно виконати такі завдання:

– провести огляд і аналіз сучасного стану задачі прогнозування медичних діагнозів;

– розглянути існуючі методи машинного навчання для аналізу медичних даних;

– розглянути методи обґрунтованості медичних прогнозів;

– обрати найбільш ефективний метод для вирішення задачі прогнозування;

– розробити програмну реалізацію для прогнозування медичних діагнозів з використанням обраного методу;

– провести обчислювальний експеримент на реальних даних пацієнтів;

– на основі отриманих даних оцінити ефективність розробленого алгоритму і зробити висновки щодо його практичного застосування.

Об'єктом дослідження є процес прогнозування медичних діагнозів на основі аналізу медичних даних пацієнтів.

Предметом дослідження є алгоритми машинного навчання та їх застосування для побудови моделей прогнозування медичних діагнозів.

Методи дослідження. У кваліфікаційній роботі застосовуються метод екстремального градієнтного бустінгу та штучні нейронні мережі.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Системний аналіз задачі прогнозування медичних діагнозів

Об’єкт аналізу – «Математичні моделі та методи прогнозування медичних діагнозів».

Предмет аналізу – «Визначення можливості побудови математичної моделі для прогнозування медичних діагнозів».

Точка зору: дослідник.

Ціль: створення математичної моделі прогнозування медичних діагнозів, розв’язання задачі прогнозування за допомогою методів машинного навчання.

Призначення системи: передбачення розвитку медичних діагнозів на основі аналізу медичних даних пацієнтів.

Система обирається на основі наступного формулювання: «Прогнозування медичних діагнозів на основі аналізу даних пацієнтів». Метою системи є побудова математичної моделі для прогнозування захворювань та вибір оптимального методу машинного навчання для реалізації цього завдання.

На вході системи маємо множину медичних даних пацієнтів, включаючи демографічну інформацію, результати лабораторних досліджень, генетичні дані та історію хвороб. Вихідним результатом є прогноз захворювання на основі вхідних даних.

В рамках даної системи функції виконуються двома ключовими елементами – дослідником та програмним забезпеченням. Дослідник виконує аналіз даних, будує модель, проводить тренування моделі та аналізує результати її роботи, забезпечуючи наукову обґрунтованість процесу. Програмне забезпечення контролює якість обчислень, забезпечує точність і надійність отриманих даних, а також автоматизує процес обробки даних і тренування моделей. Ці два компоненти тісно взаємодіють, забезпечуючи успішне функціонування системи та досягнення поставленої мети.

Методи машинного навчання, такі як метод екстремального градієнтного бустінгу (XGBoost) та нейронні мережі, виступають засобами управління системою, дозволяючи реалізувати процес прогнозування медичних діагнозів.

Переходимо до морфологічного опису системи. Система включає сукупність об’єктів, їх властивостей та взаємодій, які впливають на її функціонування. Це зовнішнє середовище системи, яке може включати інші процеси та фактори, що мають безпосередній вплив на роботу системи прогнозування, наприклад, зміни у медичних протоколах або нові наукові відкриття в сфері медицини. Система, у свою чергу, впливає на зовнішнє середовище, змінюючи підходи до лікування та діагностики захворювань через покращене прогнозування.

Це взаємодія дозволяє системі адаптуватися до змін та забезпечувати точне прогнозування медичних діагнозів, використовуючи найсучасніші підходи до обробки та аналізу медичних даних.

На рис. 1.1 представлено зовнішнє середовище системи.

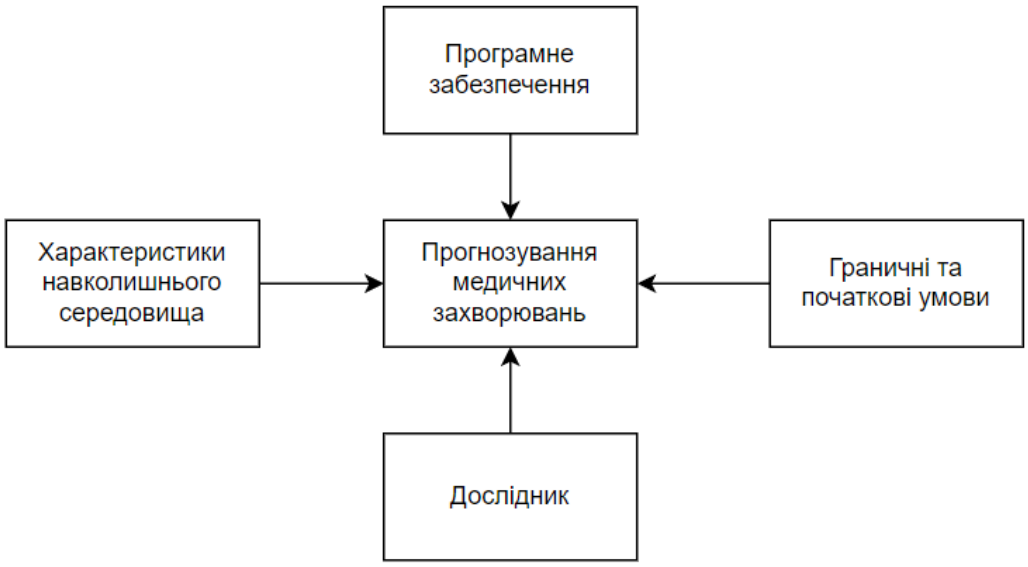

Рисунок 1.1 – Модель зовнішнього середовища системи

Модель типу «чорна скриня» використовується для перетворення вхідних даних системи у вихідні результати (рис. 1.2). У цій моделі акцент робиться на взаємодії системи із зовнішнім середовищем, не заглиблюючись у внутрішні процеси. Вона може включати компоненти або блоки, що представляють вхідні

дані, виходи та взаємодії із зовнішніми системами, такими як обмін інформацією чи зв’язок з іншими пристроями.

Модель «чорна скриня» дозволяє уникнути деталізації внутрішніх механізмів системи та зосередитися на її зовнішній поведінці. Це спрощує аналіз і розуміння роботи системи, а також може використовуватися для тестування та оцінки її функціонування. Такий підхід корисний для комунікації із зацікавленими сторонами, які не потребують знань про внутрішню структуру.

Функціональна модель системи будується з використанням набору блоків та зв’язків між ними, де кожен блок відображає певні функції, що виконує система. В ієрархії діаграм IDEF0 початкова діаграма демонструє основні взаємодії системи із зовнішнім середовищем. Такі діаграми служать для наочного представлення функціональних можливостей системи, полегшуючи її аналіз та розуміння.

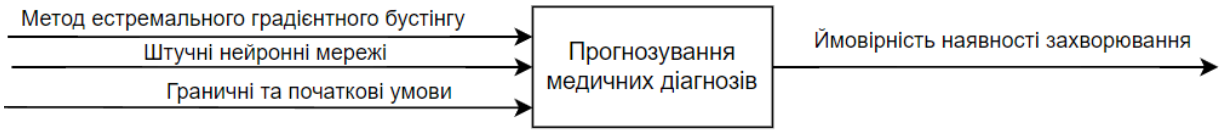

Рисунок 1.2 – Модель «чорна скриня»

На IDEF0-діаграмі представлено основні функції системи прогнозування медичних діагнозів. До цих функцій належать: метод екстремального бустінгу, штучні нейронні мережі, граничні та початкові умови та ймовірність наявності захворювання. Усі ці функції зображені у вигляді блоків на діаграмі, а стрілки між ними демонструють потік даних та інформації в системі.

У системі головне завдання полягає в побудові моделі прогнозування медичних діагнозів на основі медичних даних пацієнтів. На вхід системи надходять дані про стан здоров’я пацієнта, такі як історія хвороб, результати лабораторних аналізів та інші фактори ризику. Основна мета системи – це побудова моделі, яка зможе точно передбачати ймовірність захворювання для кожного пацієнта.

Для досягнення цієї мети використовуються такі методи машинного навчання, як екстремальний градієнтний бустінг (XGBoost) та нейронні мережі. Ці методи є ключовими інструментами, що дозволяють ефективно обробляти великі обсяги медичних даних та забезпечують високий рівень точності прогнозування.

На виході система надає ймовірнісну оцінку ризику розвитку захворювань для кожного пацієнта.

Розглянемо інформаційну модель системи. DFD (Data Flow Diagram) – це діаграма потоку даних, що використовується для моделювання системи з точки зору потоків даних між різними її компонентами.

DFD-діаграма демонструє взаємодію між процесами збору, обробки та аналізу медичних даних, а також зовнішніми сутностями, такими як лікарі або інші медичні установи. Діаграма складається з блоків, які представляють окремі процеси (наприклад, обробка даних або тренування моделі), і стрілок, що вказують напрямок руху даних між процесами. Також вона може відображати зовнішні сутності, що взаємодіють із системою, та обробку даних між компонентами.

Основна мета DFD-діаграм полягає в тому, щоб представити процеси обробки інформації та канали, якими дані передаються між ними, з фокусом на виявлення ключових компонентів системи та аналіз їхньої взаємодії. Ці діаграми допомагають розробникам і аналітикам визначити, як дані рухаються у системі, які процеси необхідні для їхньої обробки, де зберігаються результати обробки та як відбувається обмін даними між компонентами. Це сприяє ідентифікації потенційних вузьких місць та можливих проблем у процесі передачі даних, що є важливим для забезпечення ефективності та надійності функціонування системи.

На рис. 1.3 наведена DFD-діаграма системи «Прогнозування медичних діагнозів на основі аналізу медичних даних».

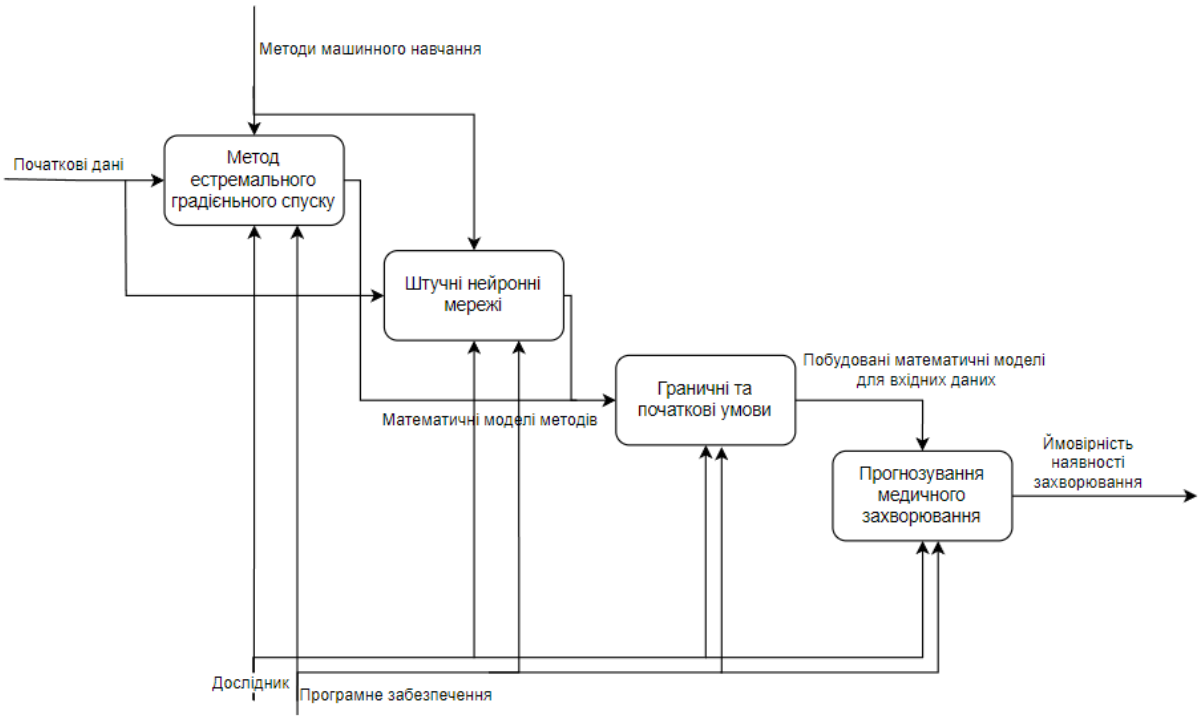

Рисунок 1.3 – Композиція DFD-діаграми першого рівня

1.2 Аналіз сценаріїв вирішення задачі прогнозування медичних діагнозів

Аналіз та вибір оптимального методу прогнозування медичних діагнозів є складним і багатокомпонентним завданням, що вимагає врахування як точності, так і ефективності обчислень. Метою дослідження є оцінка та порівняння кількох широко використовуваних методів машинного навчання, які застосовуються для задач прогнозування в медичній сфері, з подальшим вибором методу, що забезпечує найвищу якість прогнозу в межах поставлених обмежень та специфікацій. Розробка надійних моделей для прогнозування медичних діагнозів є критично важливою, оскільки помилки у прогнозах можуть мати серйозні наслідки для здоров’я пацієнтів та вплинути на клінічні рішення. При виборі оптимального методу важливо враховувати баланс між складністю моделі та її здатністю до узагальнення на нові дані. Особливу увагу слід приділити методам інтерпретації результатів, таким як SHAP, які допомагають зрозуміти внесок кожної ознаки у прийняття рішення. Крім того, моделі повинні відповідати етич-

ним нормам і забезпечувати конфіденційність медичних даних пацієнтів.

Альтернативами є кілька методів машинного навчання, серед яких обирається найкращий на основі порівняльного аналізу:

– критерій 1 (К1): складність вхідних даних;

– критерій 2 (К2): здатність моделі до узагальнення;

– критерій 3 (К3): точність прогнозування;

– критерій 4 (К4): обчислювальна ефективність.

Обирати метод розв’язання будемо з множини альтернатив:

– альтернатива 1 (А1): метод дерева рішень;

– альтернатива 2 (А2): метод найближчих сусідів;

– альтернатива 3 (А3): метод опорних векторів;

– альтернатива 4 (А4): штучні нейронні мережі;

– альтернатива 5 (А5): метод баєсівського класифікатору;

– альтернатива 6 (А6): метод градієнтного бустінгу.

Ієрархічна структура для вирішення проблеми вибору методу прогнозування медичних діагнозів має вигляд, поданий на рис. 1.4.

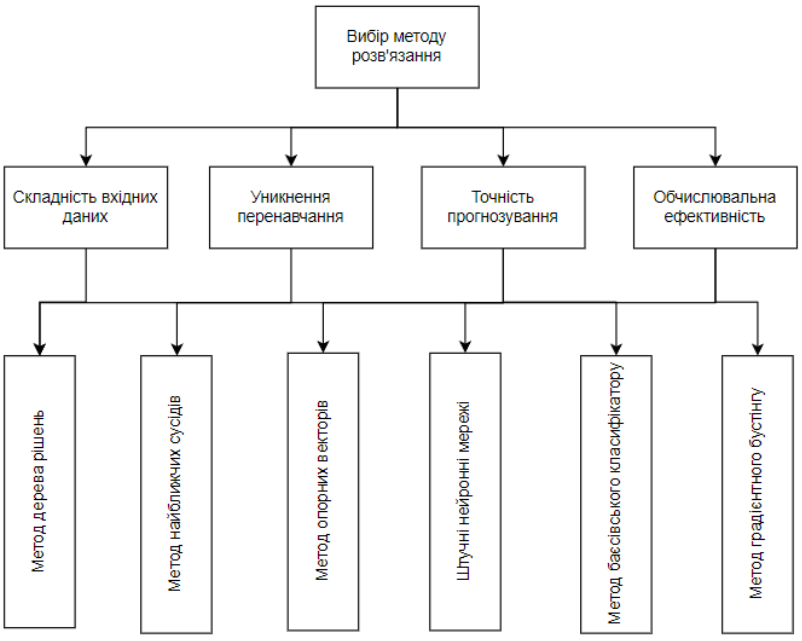

Рисунок 1.4 – Ієрархічна структура системи

Спершу побудуємо матрицю попарних порівнянь (табл. 1.1), і матриці попарних порівнянь критеріїв системи.

Таблиця 1.1 – Матриця попарних порівнянь

Критерії	К1	К2	К3	К4	Вектор пріоритетів
К1	1	3	7	8	0,05
К2	$\frac{1}{3}$	1	5	6	0,09
К3	$\frac{1}{7}$	$\frac{1}{5}$	1	1	0,41
К4	$\frac{1}{8}$	$\frac{1}{6}$	1	1	0,45

Найбільш вагомими критеріями є точність (К3) та ефективність обчислень (К4). Оцінимо альтернативи щодо кожного критерію.

Таблиця 1.2 – Порівняння за першим критерієм

Альтерна-тива	А1	А2	А3	А4	А5	А6	Вектор пріоритетів
А1	1	2	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{3}$	0,12
А2	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{2}$	0,16
А3	3	2	1	2	$\frac{1}{5}$	1	0,21
А4	4	3	$\frac{1}{2}$	1	1	2	0,23
А5	5	4	5	1	1	3	0,18
А6	3	2	2	2	3	1	0,1

Таблиця 1.3 – Порівняння за другим критерієм

Альтернатива	A1	A2	A3	A4	A5	A6	Вектор пріоритетів
A1	1	4	2	$\frac{1}{2}$	$\frac{1}{5}$	$\frac{1}{4}$	0,11
A2	$\frac{1}{4}$	1	3	$\frac{1}{3}$	2	$\frac{1}{2}$	0,16
A3	2	$\frac{1}{3}$	1	2	1	3	0,22
A4	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	1	$\frac{1}{3}$	1	0,24
A5	5	3	1	$\frac{1}{3}$	1	4	0,17
A6	4	3	2	1	4	1	0,1

Таблиця 1.4 – Порівняння за третім критерієм

Альтернатива	A1	A2	A3	A4	A5	A6	Вектор пріоритетів
A1	1	2	3	$\frac{1}{2}$	$\frac{1}{3}$	1	0,15
A2	$\frac{1}{2}$	1	4	$\frac{1}{3}$	2	$\frac{1}{2}$	0,18
A3	$\frac{1}{3}$	$\frac{1}{4}$	1	2	3	4	0,23
A4	2	3	$\frac{1}{2}$	1	2	1	0,21
A5	3	2	3	2	1	5	0,24
A6	4	5	4	1	5	1	0,35

Таблиця 1.5 – Порівняння за четвертим критерієм

Альтернатива	А1	А2	А3	А4	А5	А6	Вектор пріоритетів
А1	1	6	5	2	3	$\frac{1}{4}$	0,13
А2	$\frac{1}{6}$	1	3	1	4	$\frac{1}{2}$	0,18
А3	$\frac{1}{5}$	$\frac{1}{3}$	1	2	3	1	0,21
А4	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{3}$	3	0,15
А5	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{3}$	3	1	5	0,17
А6	4	2	1	$\frac{1}{3}$	$\frac{1}{5}$	1	0,26

Проаналізувавши отримані результати (табл. 1.6), можна стверджувати, що доцільно буде вибрати шосту альтернативу, тобто в даній роботі будемо застосовувати метод градієнтного бустінгу.

Таблиця 1.6 – Підсумкові дані

Критерій Альтернатива	К1	К2	К3	К4	Глобальні пріоритети
А1	0,12	0,12	0,15	0,13	0,13
А2	0,16	0,16	0,18	0,18	0,17
А3	0,21	0,21	0,23	0,21	0,215
А4	0,23	0,23	0,21	0,15	0,22
А5	0,18	0,18	0,24	0,17	0,194
А6	0,1	0,1	0,35	0,26	0,265

1.3 Змістовна та формальна постановка задачі

1.3.1 Змістовна постановка задачі

Прогнозування медичних діагнозів є однією з найактуальніших проблем в сучасній медичній інформатиці та охороні здоров'я, оскільки воно дозволяє не лише покращити ранню діагностику, але й ефективно запобігати розвитку складних патологічних станів. Проблема полягає в необхідності аналізу великого обсягу гетерогенних даних: медичних записів пацієнтів, результатів лабораторних досліджень, генетичної інформації, демографічних факторів та інших змінних, які можуть мати як явний, так і прихований вплив на розвиток захворювань.

Змістовна постановка задачі передбачає вивчення можливостей побудови системи, що здатна здійснювати точний прогноз на основі аналізу зазначених параметрів. Проблематика полягає у тому, що медичні дані часто містять пропуски, можуть бути нерівномірно розподілені між різними групами пацієнтів, а зв'язки між ознаками можуть бути нелінійними та складними для традиційних методів статистичного аналізу. Саме тому використання інструментів машинного навчання та аналізу даних дозволяє створювати прогностичні моделі, які здатні враховувати складні взаємодії між факторами ризику та іншими змінними.

1.3.2 Формальна постановка задачі

Формальна постановка задачі прогнозування медичних діагнозів полягає в математичній формалізації процесу класифікації пацієнтів на основі вхідних медичних даних. Завдання полягає в побудові функції $f(X)$, яка на основі множини ознак пацієнтів $X = \{x_1, x_2, ..., x_n\}$ передбачає цільову змінну y, що відповідає наявності або відсутності ознаки.

Вхідні ознаки представляються у вигляді вектора ознак X, де x_i – це окрема ознака, наприклад, вік, стать, результати лабораторних досліджень, генетичні маркери тощо. Таким чином, можна записати:

$$X = \{x_1, x_2, ..., x_n\}, \tag{1.1}$$

де n – кількість ознак, які описують стан пацієнта. Кожна з цих ознак може кількісною (наприклад, глюкоза в крові) або категоріальною (наприклад, наявність чи відсутність певних симптомів).

Цільова змінна y визначається як залежність від цих ознак, і може бути бінарною або мультикласовою. У випадку бінарної класифікації y приймає значення:

$$y = \{0,1\}, \tag{1.2}$$

де 0 означає відсутність захворювання, а 1 – наявність захворювання.

У більш загальному випадку, якщо прогнозується ймовірність декількох захворювань, y може бути мультикласовою змінною:

$$y \in \{y_1, y_2, ..., y_k\}, \tag{1.3}$$

де k – кількість класів (наприклад, типи захворювань). Метою є побудова моделі $f(X)$, яка відображає вхідні дані X у цільову функцію y. Формально задача формулюється як пошук такої функції f, що мінімізує помилку прогнозування:

$$f(X) = \arg\min_{f \in W} \left| y - f(X) \right|, \tag{1.4}$$

де $X, y \in DS$ – набір даних;

W – множина допустимих функцій.

Для формалізації пояснюваності можна представити передбачення як адитивну комбінацію впливів окремих ознак. В математичному вигляді пояснюваність передбачення моделі можна записати так:

$$\hat{y} = \varphi_0 + \sum_{i=1}^{n} \varphi_i x_i, \tag{1.5}$$

де $\hat{y}$ – передбачуване значення цільової змінної;

φ_0 – базове значення моделі;

φ_i – вплив i-ої ознаки x_i на передбачення;

n – кількість ознак.

У рамках обґрунтованості висновків можуть використовуватися різні методи, такі як SHAP або LIME, для оцінки значень φ_i.Ці значення відображають зміну передбачення, яка виникає при додаванні або вилученні ознаки з моделі. Наприклад, метод SHAP використовує теорію ігор для обчислення середнього внеску кожної ознаки на основі всіх можливих комбінацій ознак. У той же час, метод LIME апроксимує нелінійну модель локальною лінійною моделлю для обчислення впливу ознак поблизу даного передбачення.

1.4 Постановка задач дослідження

Виходячи з проведеного системного аналізу предметної області прогнозування медичних діагнозів, можна зробити висновок, що для вирішення цієї задачі необхідно використовувати методи машинного навчання, які можуть ефективно працювати з великими обсягами гетерогенних даних.

Метою кваліфікаційної роботи є використання методів машинного навчання для прогнозування медичних діагнозів, що дозволить забезпечити точ-

ність прогнозів, зменшити ймовірність помилкових діагнозів і сприяти прийняттю обґрунтованих медичних рішень.

Для досягнення поставленої мети необхідно виконати наступні завдання:

– провести огляд і аналіз сучасного стану проблеми прогнозування медичних діагнозів;

– розглянути існуючі методи машинного навчання для аналізу медичних даних;

– обрати найкращий метод для вирішення поставленої задачі;

– розробити програмну реалізацію обраного методу з використанням мови програмування Python;

– провести тренування моделі на реальних медичних даних і проаналізувати точність прогнозування;

– провести обчислювальний експеримент для перевірки ефективності розробленого алгоритму та зробити висновки на основі отриманих результатів.

2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ'ЯЗАННЯ

2.1 Огляд існуючих методів прогнозування медичних діагнозів

Прогнозування захворювань є важливим напрямом у сучасній медицині, який спрямований на підвищення ефективності лікування та профілактики. Для вирішення цієї задачі використовуються різноманітні математичні та статистичні методи. Традиційні підходи базуються на аналізі історичних даних, демографічних характеристик пацієнтів та факторів ризику. До них належать регресійні моделі, ймовірнісні методи, а також кластеризація, які допомагають визначити зв'язки між факторами й захворюваннями та передбачити їх розвиток.

Тим не менш, традиційні підходи мають певні обмеження. По-перше, вони часто не враховують складні нелінійні взаємозв'язки між змінними. По-друге, обробка великих обсягів медичних даних стає проблематичною при використанні класичних методів. У зв'язку з цим активно впроваджуються нові підходи на основі машинного навчання, які дозволяють підвищити точність прогнозів та адаптувати моделі до великих та складних наборів даних.

Методи Explainable AI (XAI) забезпечують прозорість моделей машинного навчання та дозволяють отримати пояснення, які можуть бути зрозумілими для лікарів і медичних працівників. Наприклад, метод SHAP (SHapley Additive exPlanations) базується на теорії ігор і використовує вартісний підхід Шеплі для оцінки впливу кожної ознаки на вихід моделі. Це дозволяє оцінити, як кожен показник (наприклад, індекс маси тіла або артеріальний тиск) впливає на прогноз прогресування діабету. SHAP надає як глобальні, так і локальні пояснення, допомагаючи інтерпретувати вплив ознак на рівні окремих пацієнтів або всієї вибірки.

Інший підхід – метод LIME (Local Interpretable Model-agnostic Explanations), який створює локальні лінійні апроксимації для пояснення рішень складних моделей. LIME дозволяє зрозуміти, як зміни в конкретних ознаках поблизу досліджуваної точки можуть впливати на прогноз. Хоча метод

менш стабільний, ніж SHAP, він є інтуїтивно зрозумілим і часто використовується для швидкого пояснення прогнозів.

Таким чином, використання методів XAI у дослідженні діабету дозволяє не лише підвищити точність моделей прогнозування, але й забезпечити інтерпретованість результатів, що є важливим для клінічного прийняття рішень.

2.1.1 Метод найближчих сусідів

Метод найближчих сусідів використовує принцип порівняння нового зразка з найближчими до нього вже відомими зразками в просторі ознак. Нехай відомий навчальний набір даних із N зразків, як представлено в формулі (2.1):

$$D = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}, \tag{2.1}$$

де $x_i \in R$ – вектор ознак для i-го зразка;

$y_i \in \{1,2,...,C\}$ – клас, якому належить заданий зразок.

Для нового зразка x_{new} метод найближчих сусідів шукає k найближчих зразків у цьому просторі ознак та виконує голосування для визначення класу.

Відстань між векторами ознак обчислюється за допомогою Евклідової метрики, як представлено в формулі (2.2):

$$d(x_i, y_i) = \sqrt{\sum_{i=1}^{d} (x_{i,l} - x_{j,l})^2}, \tag{2.2}$$

де $x_{i,l}$ та $x_{j,l}$ – значення l-ої ознаки відповідних зразків.

Клас нового зразка x_{new} визначається за більшістю серед k найближчих сусідів, як представлено в формулі (2.3):

$$y_{new} = \arg\max_{y} \sum_{i \in N_k} \mathrm{I}(y_i = y), \tag{2.3}$$

де N_k – множина індексів k -найближчих сусідів;

I– індикаторна функція, що дорівнює 1, якщо умова виконується, та 0 в іншому випадку.

Метод найближчих сусідів чутливий до вибору параметра k. Занадто мале значення може призвести до перенавчання, тоді як занадто велике значення k призведе до розмиття мед між класами. Загальна структура методу найближчих сусідів зображено на рис. 2.1.

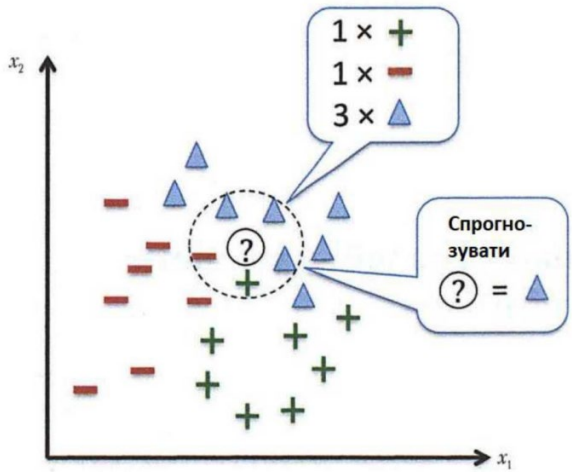

Рисунок 2.1 – Загальна структура методу найближчих сусідів

2.1.2 Метод дерева рішень

Дерева рішень представляють алгоритм, який створює послідовність розгалужень на основі значень ознак, щоб поділити дані на класи. Дерево складається з вузлів, у яких перевіряються умови на ознаки, та листків, де зберігаються класи або значення регресії.

Алгоритм побудови дерева використовує критерії розбиття, такі як інформаційна виграшність або критерій Джині. Для кожного вузла обирається озна-

ка, яка максимізує зменшення невизначеності в даних. Розглянемо ентропію як міру невизначеності, як представлено в формулі (2.4):

$$H(S) = -\sum_{c=1}^{C} p_c \log_2 p_c, \tag{2.4}$$

де p_c – ймовірність зразка належати до класу c у множині S.

Інформаційний виграш для ознаки A обчислюється як представлено в формулі (2.5):

$$IG(S, A) = H(S) - \sum_{v \in A} \frac{|S_v|}{|S|} H(S_v), \tag{2.5}$$

де S_v – підмножина зразків, для яких ознака A має значення v.

Алгоритм будує дерево, поки інформаційний виграш є значним. Однак дерево рішень може бути схильним до перенавчання, тому застосовуються методи обрізання, щоб зменшити розмір дерева. Загальна структура методу дерева рішень зображено на рис. 2.2.

Рисунок 2.2 – Загальна структура методу дерева рішень

2.1.3 Метод опорних векторів

Метод опорних векторів призначений для знаходження оптимальної гіперплощини, яка розділяє дані на два класи з максимальним зазором між ними. Припустимо, що є два класи та дані є лінійно роздільними. Мета методу опорних векторів полягає в знаходженні гіперплощини, яка описується рівнянням, що представлено в формулі (2.6):

$$w^T x + b = 0, \tag{2.6}$$

де w – вектор коефіцієнтів (ваг);

x – вектор ознак;

b – зміщення.

Два класи повинні задовольняти умови (2.7) та (2.8) :

$$w^T x_i + b \geq 1,\ y_i = 1, \tag{2.7}$$

$$w^T x_i + b \leq -1,\ y_i = -1. \tag{2.8}$$

Відстань між цими двома гіперплощинами називається маржою. Оптимізація в методі опорних векторів зводиться до задачі максимізації маржі, що еквівалентно мінімізації виразу (2.9):

$$\min_{w,b} \frac{1}{2}\|w\|^2, \tag{2.9}$$

за умови, що для всіх i виконується співвідношення (2.10):

$$y_i(w^T x_i + b) \geq 1. \tag{2.10}$$

У випадку, коли дані не є лінійно роздільними, додається пом’якшувальний параметр C, який штрафує за помилки, як представлено в формулі (2.11):

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i, \tag{2.11}$$

де ξ_i – змінні, що враховують помилки класифікації.

Для нелінійних даних використовується ядрова функція $K(x_i, x_y)$, яка проектує дані у простір більшої розмірності, де вони стають лінійно роздільними.

Метод опорних векторів забезпечує високу точність, особливо для малих наборів даних, проте потребує багато ресурсів для налаштування параметрів і вибору ядра. Загальна структура методу опорних векторів зображено на рис. 2.3.

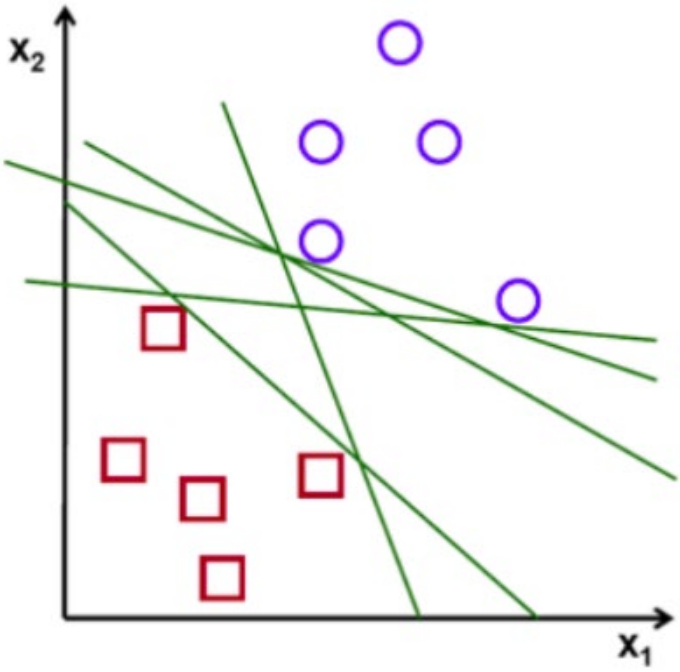

Рисунок 2.3 – Загальна структура методу опорних векторів

2.1.4 Метод баєсівського класифікатору

Баєсівський класифікатор ґрунтується на теоремі Байєса та працює за принципом оцінки ймовірностей приналежності зразка до певного класу. Алгоритм розраховує апостеріорну ймовірність кожного класу за заданими ознаками та обирає клас з максимальною ймовірністю. Алгоритм часто використовується в текстовій аналітиці, наприклад, для фільтрації спаму або класифікації документів за темами. Незважаючи на свою простоту, Баєсівський класифікатор може конкурувати з більш складними моделями, особливо якщо дані добре відповідають його припущенням.

Формально, для кожного зразка $x = (x_1, x_2, ..., x_d)$ ймовірність належності до класу y_i визначається як представлено в формулі (2.12):

$$P(y_j \mid x) = \frac{P(x \mid y_j) P(y_j)}{P(x)}, \tag{2.12}$$

де $P(y_j \mid x)$ – апостеріорна ймовірність класу y_i;

$P(x \mid y_i)$ – ймовірність спостереження ознак при умові, що зразок належить до класу y_i;

$P(y_j)$ – апріорна ймовірність класу y_i;

$P(x)$ – загальна ймовірність.

Метод припускає, що всі ознаки є незалежними (наївне припущення), як представлено в формулі (2.13):

$$P(x \mid y_j) = \prod_{i=1}^{d} P(x_i \mid y_j). \tag{2.13}$$

Цей підхід є обчислювально ефективним, проте не завжди забезпечує високу точність у випадках, коли ознаки сильно корельовані.

2.1.5 Огляд штучних нейронних мереж

Штучні нейронні мережі є інструментом глибокого навчання, які імітують роботу біологічного мозку шляхом обробки інформації у вигляді багатошарових структур. Нейронні мережі широко застосовуються для складних задач прогнозування та класифікації, таких як обробка медичних зображень або прогнозування перебігу хвороби. Кожен шар мережі складається з нейронів, які отримують вхідні сигнали, зважують їх та передають на наступний шар.

Основним принципом роботи є навчання з використанням градієнтного спуску для мінімізації функції втрат. Нейронні мережі здатні автоматично виявляти нелінійні взаємозв'язки в даних, що робить їх надзвичайно корисними для медичних задач, де дані можуть бути складними та багатовимірними. Загальна структура нейронної мережі зображена на рис. 2.5.

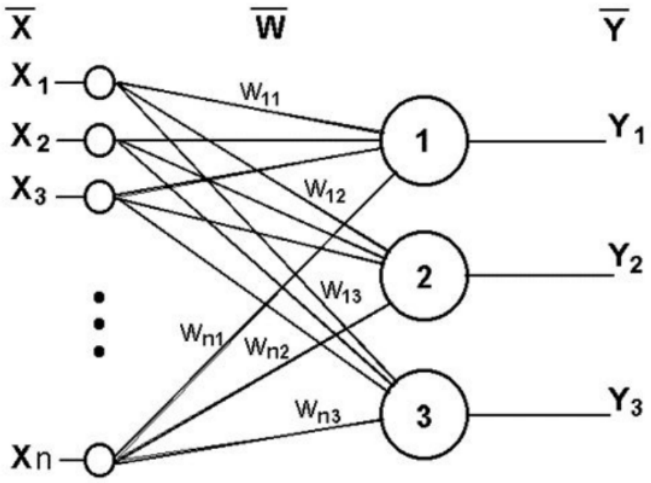

Рисунок 2.4 – Загальна структура штучної нейронної мережі

Вхідний шар приймає вхідні дані, що складаються з векторів ознак $x = (x_1, x_2, ..., x_n)$, де d описує кількість ознак. Кількість нейронів у цьому шарі відповідає розміру вхідних даних.

Приховані шари виконують основну обробку даних. Кожен нейрон у прихованому шарі обчислює лінійну комбінацію вхідних сигналів із застосуванням

вагових коефіцієнтів зміщення, як представлено в формулі (2.14):

$$z_j = \sum_{i=1}^{d} w_{ji} \cdot x_i + b_j, \tag{2.14}$$

де w_{ji} – вага зв'язку між нейроном i-го шару та нейроном j-го шару;

b – зміщення.

Після цього результат проходить через активаційну функцію, яка вводить нелінійність в модель. Найбільш використовуваними функціями активації є сигмоїдальна, яка перетворює значення в межах від 0 до 1 та підходить для задач класифікації, та тангенціальна, яка перетворює значення в інтервалі від -1 до 1, що використовується для сигналів, центрованих навколо нуля, як представлено в формулах (2.15) та (2.16):

$$\sigma(z) = \frac{1}{1+e^{-z}}, \tag{2.15}$$

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}. \tag{2.16}$$

Вхідний шар генерує остаточний результат мережі. Для задач класифікації з декількома класами на вихідному шарі застосовується Softmax-функція, яка обчислює ймовірності приналежності зразка до кожного з класів, як представлено в формулі (2.17):

$$Soft\max(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, \tag{2.17}$$

де z_i – вихід нейрона для класу i;

C – кількість класів.

Нейронні мережі використовують градієнтний спуск для мінімізації функції витрат, що визначає різницю між передбаченими та реальними значеннями. Функція втрат для задач регресії може мати вигляд середньоквадратичної похибки, як представлено в формулі (2.18):

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{2.18}$$

де y_i – справжнє значення;

$\hat{y}_i$ – передбачуване значення.

Для задач класифікації використовуються крос-ентропія, як представлено в формулі (2.19):

$$L = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \ln \hat{y}_{ij}, \tag{2.19}$$

де y_{ij} – справжня ймовірність класу j для зразка i;

$\hat{y}_{ij}$ – передбачувана ймовірність.

Похибку, обчислену на вихідному шарі, передають назад через усі шари мережі за допомогою зворотнього поширення похибки. Градієнти функції втрат обчислюються для кожної ваги, і ваги оновлюються згідно з правилом, що представлено в формулі (2.20):

$$w_{ji} \leftarrow w_{ji} - \eta \frac{\partial L}{\partial w_{ji}}, \tag{2.20}$$

де η – коефіцієнт навчання, який визначає наскільки сильно змінюються ваги під час навчання.

2.1.6 Метод екстремального градієнтного бустінгу

Метод екстремального градієнтного бустінгу є одним із найпотужніших алгоритмів ансамблевого навчання, який поєднує багато дерев рішень для побудови сильної моделі. На відміну від звичайного градієнтного бустінгу, він оптимізований для швидкої обробки великих обсягів даних та містить додаткові механізми для запобігання перенавчанню.

Алгоритм екстремального градієнтного бустінгу будує модель поступово, додаючи нові дерева, кожне з яких намагається компенсувати помилки попередніх дерев. Нехай є навчальний набір даних $\{(x_i, y_i)\}_{i=1}^{N}$, де x_i, y_i є вектором ознак та цільовою змінною відповідно. Модель $F(x)$ є сумою прогнозів окремих дерев, що представлено в формулі (2.21):

$$F(x) = \sum_{t=1}^{T} f_t(x), \;\; f_t \in \Psi, \tag{2.21}$$

де T – кількість дерев;

Ψ – простір усіх можливих дерев.

На кожному кроці мінімізується функція втрат L, яка виражає різницю між передбаченим та реальним значенням, як представлено в формулі (2.22):

$$L(t) = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \Omega(f_t), \tag{2.22}$$

де ℓ – функція втрат;

$\Omega(f_t)$ – регуляризаційний член для запобігання перенавчанню, як представлено в формулі (2.23):

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2, \tag{2.23}$$

де γ та λ – параметри регуляризації.

Метод екстремального градієнтного бустінгу забезпечує високу точність та добре працює з великою кількістю ознак, що робить його широко використовуваним для таких задач, як прогнозування виникнення захворювань або передбачення результатів лікування. Загальна структура методу екстремального градієнтного бустінгу на рис. 2.6.

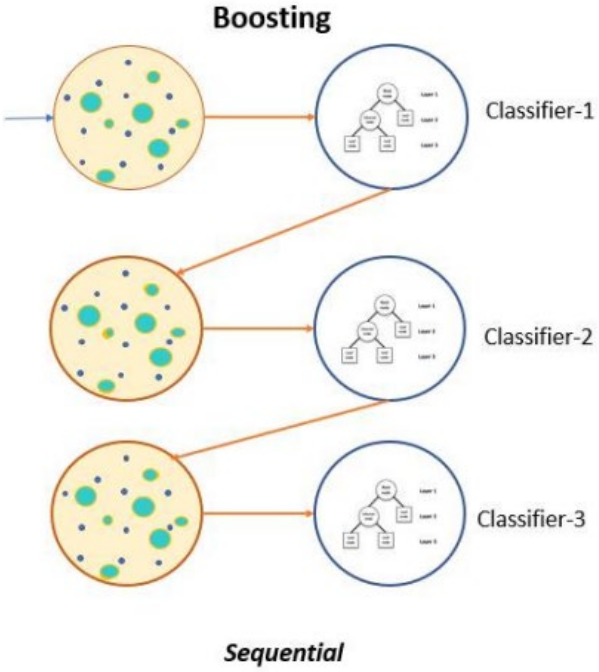

Рисунок 2.5 – Загальна структура методу екстремального градієнтного бустінгу

2.2 Метод SHAP та LAIM для розрахунку важливості ознак

Методи SHAP та LAIM використовуються для оцінки внеску ознак у результати моделей машинного навчання, що підвищує їх інтерпретованість, особливо у медичних захворюваннях.

SHAP ґрунтується на значеннях Шеплі з теорії ігор, де кожна ознака розглядається як гравець, що вносить свій вклад у прогноз. Значення Шеплі для ознаки j обчислюється як у формулі (2.24):

$$\varphi_j = \sum_{S \subseteq N \setminus \{j\}} \frac{|S|!(|N|-|S|-1)!}{|N|!}(f(S \cup \{j\}) - f(S)), \quad (2.24)$$

де S – підмножина ознак;

$f(S)$ – прогноз моделі на цій підмножині.

SHAP забезпечує глобальну інтерпретацію, але має високу обчислювальну складність для великої кількості ознак, що частково вирішується апроксимаціями, такими як Tree-SHAP.

LAIM оцінює важливість локально, для кожного конкретного прогнозу. Важливість ознак x_j визначається як у формулі (2.25):

$$LAIM(x_j) = \left| f(x) - f(x_{\setminus j}) \right|, \quad (2.25)$$

де $x_{\setminus j}$ – це вхідний вектор без ознаки x_j.

Метод LAIM показує, наскільки вилучення конкретної ознаки змінює прогноз для окремого випадку.

Порівнюючи ці два методи, SHAP забезпечує глобальну інтерпретацію моделі, дозволяючи зрозуміти середній вплив кожної ознаки, тоді як LAIM аналізує локальні впливи, корисні для індивідуальних рішень.

Висновки за розділом 2

У цьому розділі було розглянуто сучасні методи прогнозування медичних діагнозів, включаючи традиційні статистичні підходи та новітні алгоритми машинного навчання. Традиційні методи, такі як регресійні моделі та кластеризація, забезпечують основу для аналізу даних, але мають певні обмеження у складних і великомасштабних медичних даних. Натомість методи машинного навчання, зокрема нейронні мережі, методи опорних векторів та екстремальний

градієнтний бустінг, демонструють високий рівень точності та здатність адаптуватися до складних взаємозв'язків між ознаками.

Особливу увагу приділено методам Explainable AI, таким як SHAP і LIME, які підвищують інтерпретованість прогнозів моделей. SHAP забезпечує глибоке розуміння впливу кожної ознаки на результат моделі, що важливо для обґрунтування клінічних рішень, тоді як LIME дозволяє швидко отримати локальні пояснення для індивідуальних випадків. Таким чином, інтеграція методів машинного навчання та інтерпретованих підходів дає змогу покращити точність і прозорість діагностичних моделей, що є важливим для сучасної медицини.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Мова програмування Python

Python є високорівневою мовою програмування, створеною Гвідо ван Россумом на початку 1990-х років. Вона відома своїм простим та зрозумілим синтаксисом, що робить її ідеальною для початківців у програмуванні. Одним із ключових принципів Python є акцент на читабельності коду, що дозволяє розробникам створювати легко підтримуваний та зрозумілий код.

Важливою особливістю Python є його широка функціональність і велика кількість бібліотек. Мова має велику стандартну бібліотеку з численними модулями для виконання різних завдань, таких як обробка файлів, мережеве програмування та веб-розробка. Крім того, завдяки активній спільноті розробників, існує безліч сторонніх бібліотек і фреймворків для розв'язання спеціалізованих завдань.

Python також відзначається платформною незалежністю, що дає змогу запускати код на різних операційних системах, таких як Windows, macOS і Linux, без суттєвих змін у коді. Це робить Python універсальним і гнучким інструментом для програмування.

Серед інших переваг Python – його легкість у навчанні та використанні. Завдяки зрозумілому синтаксису і великій кількості навчальних ресурсів, новачки можуть швидко оволодіти основами мови і приступити до створення власних програм. Безліч доступних онлайн-курсів, підручників і форумів сприяє популярності Python серед широкого кола розробників.

Python також є однією з найпопулярніших мов у галузі математичного моделювання. Серед причин цього – наявність потужної екосистеми бібліотек, що забезпечують інструменти для чисельних методів та наукових обчислень. Зокрема, бібліотека NumPy пропонує ефективні багатовимірні масиви і функції для роботи з ними, що є основою для багатьох математичних моделей.

Додатково, бібліотеки SciPy і pandas пропонують широкий набір інструментів для наукових обчислень і обробки даних. SciPy містить модулі для задач оптимізації, інтегрування, інтерполяції і розв'язання диференціальних рівнянь, а pandas дозволяє зручно працювати з табличними даними, забезпечуючи інструменти для їхньої обробки.

Завдяки своєму зрозумілому синтаксису Python дозволяє швидко розробляти та тестувати математичні моделі. Крім того, мова легко інтегрується з іншими мовами програмування, такими як C++, що дозволяє використовувати високопродуктивні бібліотеки для складних обчислень.

Загалом, Python є потужним інструментом для математичного моделювання завдяки своїм широким можливостям, простоті у використанні та великій спільноті користувачів, що робить його одним із найкращих виборів для розробників у цій галузі машинного навчання.

3.2 Опис програми

У цьому коді використовуються різні функції та методи з бібліотек машинного навчання та візуалізації даних, які допомагають здійснювати аналіз, побудову моделей і оцінку їхньої ефективності.

Функція sns.histplot() використовує для побудови гістограми, яка показує розподіл значень кожної ознаки. Вона також може накладати криву ймовірності (kde), щоб відобразити оцінку ймовірнісного розподілу даних.

Функція sns.boxplot() створює коробкові діаграми, які показують розподіл даних за допомогою таких статистичних характеристик, як медіана, квартилі та викиди. Це дозволяє оцінити варіативність ознак і виявити наявність викидів, що важливо для покращення моделі.

Функція sns.pairplot() побудовує матрицю парних графіків, яка показує, як кожна ознака взаємодіє з іншою, а також як вона корелює з цільовою змінною. Це допомагає побачити можливі патерни і взаємозв’язки між даними.

Функція sns.violinplot() малює діаграму типу "скрипка", яка поєднує елементи коробкової діаграми та графіка розподілу. Така візуалізація дає змогу детально оцінити варіативність даних і виявити аномалії або неоднорідності в розподілі ознак.

Функція StandardScaler() з бібліотеки sklearn.preprocessing використовує для нормалізації даних, забезпечуючи, що кожна ознака має середнє значення нуль і стандартне відхилення одиницю. Це важливо для алгоритмів, чутливих до масштабів ознак, таких як методи класифікації на основі відстані.

Функція train_test_split() з sklearn.model_selection дозволяє розбити набір даних на дві частини: одну для тренування моделі, іншу — для тестування. Це дає змогу оцінити, як добре модель працює на нових, невідомих даних, запобігаючи перенавчанню.

Функція KNeighborsClassifier() є класифікатором, який використовує принцип знаходження найближчих сусідів для визначення класу нових точок. Відстань між точками даних визначає, до якого класу належатиме точка, що класифікується.

Функція DecisionTreeClassifier() будує дерево рішень, яке розбиває дані на підмножини за допомогою порогових значень для кожної ознаки. Це дозволяє моделі приймати рішення, орієнтуючись на різні критерії розподілу даних.

Функція SVC() – це метод опорних векторів, який будує оптимальну гіперплощину для поділу даних на різні класи. Цей метод шукає таку гіперплощину, яка максимізує відстань між класами.

Функція GaussianNB() – це наївний баєсів класифікатор, який припускає, що ознаки є статистично незалежними, і використовує ймовірнісну модель для визначення класу.

Функція MLPClassifier() є багатошаровим перцептронним класифікатором, який застосовує нейронні мережі для навчання складним нелінійним залежностям між ознаками. Це дозволяє моделі вивчати складні зв’язки в даних.

Функція GradientBoostingClassifier() – це метод ансамблю, який поетапно будує моделі, фокусуючись на помилках попередніх моделей. Це дозволяє зна-

чно покращити точність моделі за рахунок комбінування слабких учасників у сильну модель.

Функція classification_report() генерує детальний звіт про точність роботи моделі, включаючи точність, відгук і F1-міру для кожного класу. Це дозволяє оцінити, наскільки добре модель класифікує кожен з класів.

Функція confusion_matrix() будує матрицю невірних класифікацій, яка показує кількість помилок, зроблених класифікатором для кожного класу. Це допомагає оцінити, наскільки добре модель справляється із завданням.

Функція shap.initjs() ініціалізує JavaScript для побудови інтерактивних графіків SHAP, які пояснюють, як кожна ознака впливає на прогноз моделі. Це дозволяє краще зрозуміти, що відбувається всередині складних моделей.

Функція shap.TreeExplainer() пояснює моделі на основі дерев рішень, таких як градієнтний бустинг, і допомагає зрозуміти, як кожна ознака впливає на передбачення моделі.

Функція shap.summary_plot() візуалізує важливість ознак, показуючи, як зміна значень кожної ознаки впливає на результат. Це дає можливість зрозуміти, на які саме аспекти даних модель орієнтується при прийнятті рішень.

Функція PCA() виконує зниження розмірності, перетворюючи вихідні ознаки у меншому вимірному просторі, зберігаючи основну частину варіативності даних. Це допомагає знизити обчислювальні витрати і спростити подальший аналіз.

Функція sns.scatterplot() створює розсіювальну діаграму, яка відображає, як дві основні компоненти даних, отримані після зниження розмірності, впливають на цільову змінну. Це допомагає наочно оцінити структуру даних в зниженому просторі.

Цей код демонструє повний процес роботи з даними – від їх попередньої обробки і візуалізації до побудови різноманітних моделей машинного навчання.

Висновки за розділом 3

У цьому розділі розглядається широкий спектр інструментів і методів для обробки, візуалізації та аналізу даних, що використовуються у машинному навчанні. Основний акцент зроблено на бібліотеках Python, таких як seaborn, sklearn і SHAP, які дозволяють проводити глибокий аналіз даних, будувати ефективні моделі і оцінювати їхню продуктивність. За допомогою таких функцій, як sns.histplot(), sns.boxplot(), та інших візуалізаційних інструментів, можна виявити ключові особливості даних, їх розподіл, кореляцію між ознаками і можливі аномалії.

Крім того, розглянуто різноманітні алгоритми машинного навчання, від простих класифікаторів, як-от KNeighborsClassifier() і DecisionTreeClassifier(), до більш складних моделей, таких як GradientBoostingClassifier() і MLPClassifier(). Описані інструменти допомагають не тільки створювати моделі, а й пояснювати їхні рішення, наприклад, за допомогою SHAP-графіків. Це дозволяє розробникам і аналітикам глибше розуміти поведінку моделей і робити більш обґрунтовані висновки з аналізу даних..

4 РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ ТА ЇХ АНАЛІЗ

4.1 Опис датасету

Набір даних load_diabetes з бібліотеки scikit-learn призначений для задач регресії, пов’язаних з передбаченням прогресування діабету. Він містить 442 записи, кожен з яких відповідає окремому пацієнту. Основна мета цього набору даних – допомогти передбачити, як зміниться стан пацієнта через рік після проведення первинних медичних обстежень.

Кожен запис включає 10 числових ознак, які представляють різні медичні параметри. Ці ознаки охоплюють такі характеристики, як вік, стать, індекс маси тіла (BMI), середній артеріальний тиск, а також шість лабораторних показників крові. Всі ознаки нормалізовані, тобто мають середнє значення 0 і стандартне відхилення 1, що допомагає зробити їх взаємозамінними при моделюванні.

Цільова змінна у цьому наборі даних – це кількісна оцінка прогресування діабету, яка відображає ступінь зміни стану пацієнта через рік. Цей набір даних часто використовується для тестування різних моделей регресії та для аналізу медичних даних, зокрема для розробки моделей, які допомагають прогнозувати довгострокові результати для пацієнтів з діабетом.

4.2 Аналіз результатів машинного навчання

Результати обчислювального експерименту, що проводився для оцінки ефективності різних алгоритмів класифікації на наборі даних про діабет, включають використання декількох моделей машинного навчання: K-найближчих сусідів (KNN), дерева рішень (Decision Tree), метод опорних векторів (SVM), наївного баєсівського класифікатора (Naive Bayes), багатошарового перцептрону (MLP), а також градієнтного бустингу (Gradient Boosting). Кожна з моделей

була протестована на розділеному наборі даних і оцінена за допомогою метрик точності, відгуку та F1-міри.

Перш за все, дані були візуалізовані для виявлення взаємозв'язків між ознаками. Використовувалися графіки парних розсіювальних діаграм (pairplot) для всіх ознак і цільової змінної, що дозволило оцінити, які ознаки мають найбільший вплив на класифікацію (рис. 4.1).

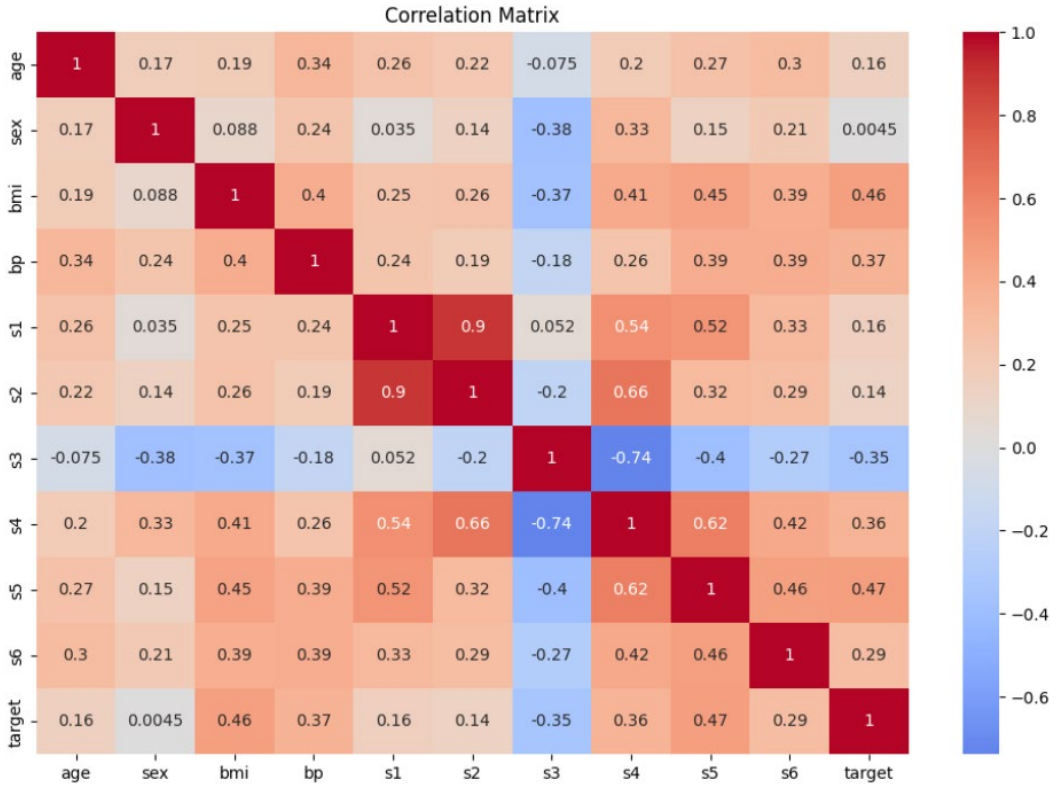

Рисунок 4.1 – Тепловий графік кореляції між ознаками

Після цього, дані були нормалізовані за допомогою `StandardScaler`, щоб зменшити вплив різних масштабів ознак на результати класифікації. Моделі були навчання і оцінені на тестовій частині даних, і результати для кожної з моделей зібрані (рис. 4.2).

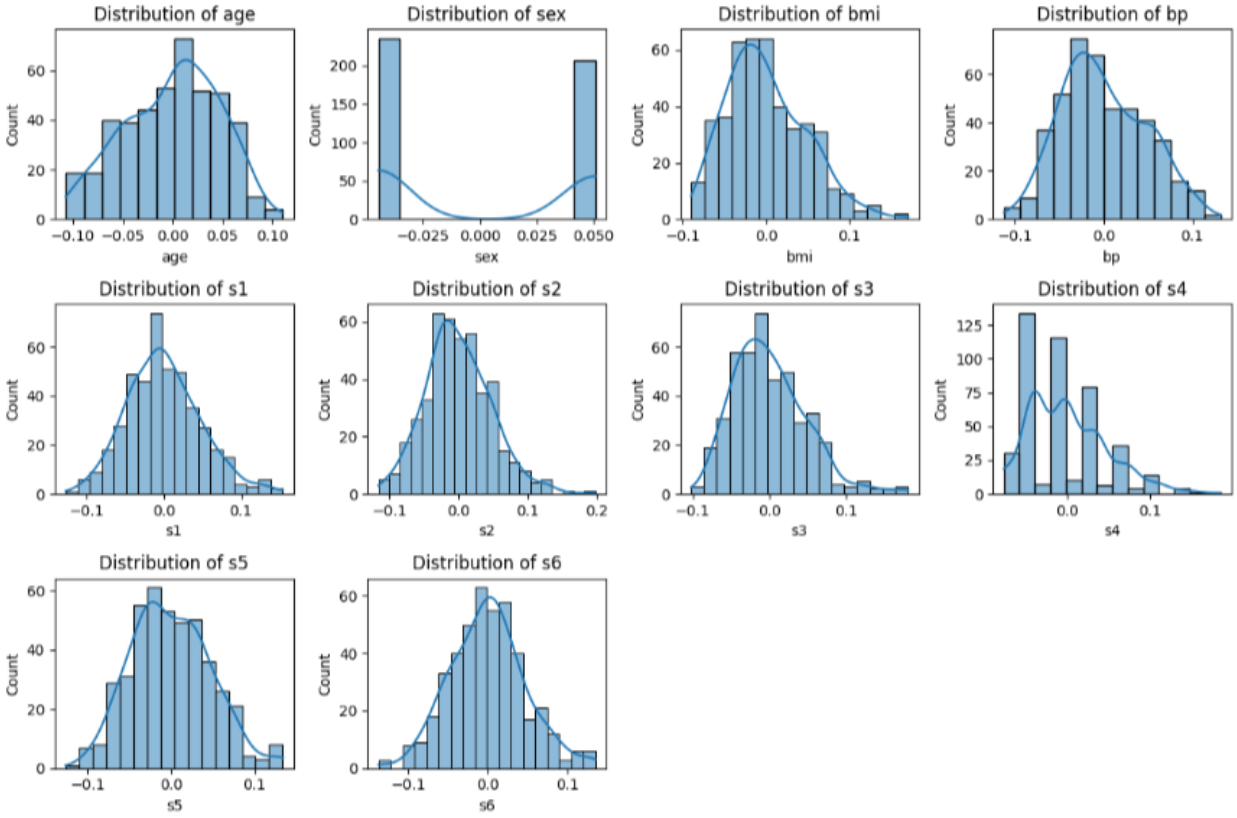

Рисунок 4.2 – Розподіл даних на основі коробкових діаграм для ознак

Метод найближчих сусідів показав точність приблизно 0.79, відгук приблизно 0.8, точність приблизно 0.78 та F1-міру приблизно 0.79. Модель добре справляється з більшістю класів, однак деякі помилки були допущені через схожість класів (рис. 4.3).

Дерево рішень показало точність приблизно 0.75, відгук приблизно 0.76, точність приблизно 0.74 та F1 приблизно 0.75. Модель працює добре, але виявляє схильність до перенавчання через високу глибину дерева (рис. 4.4).

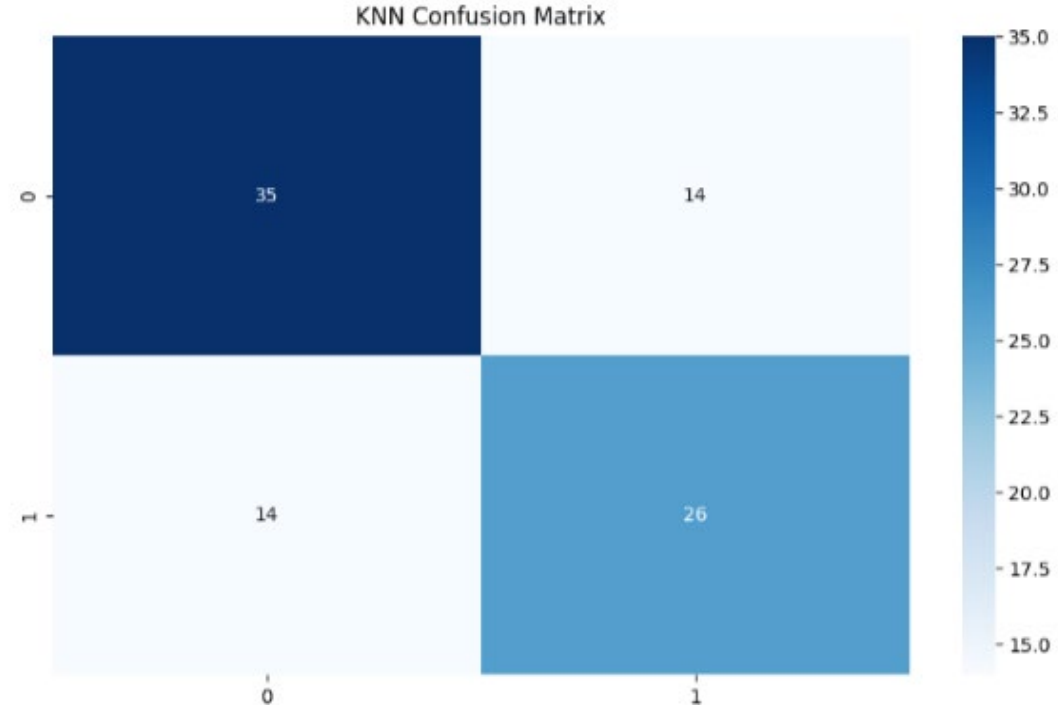

Рисунок 4.3 – Матриця невірних класифікацій для KNN

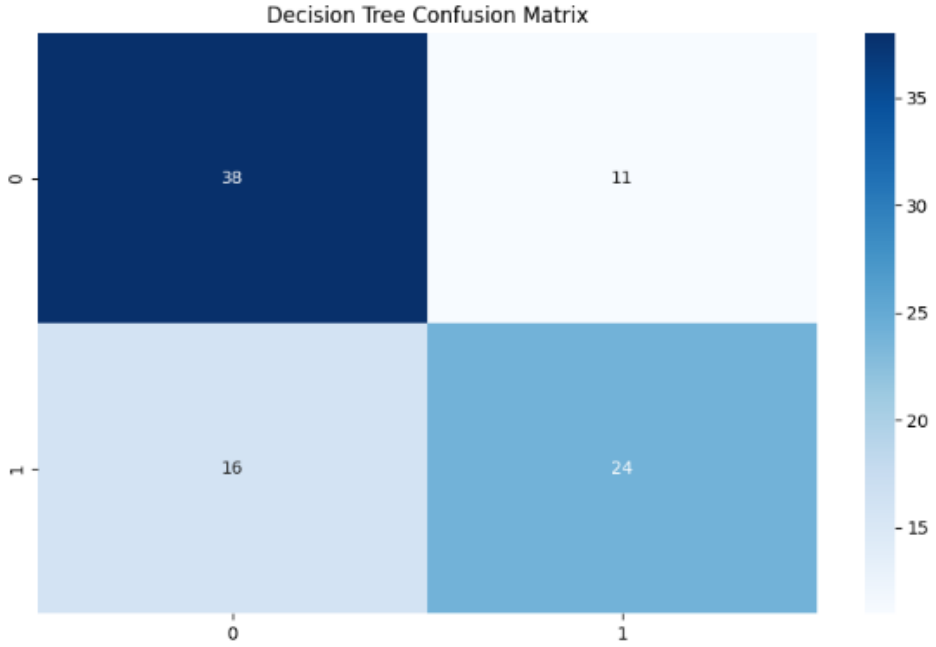

Рисунок 4.4 – Матриця невірних класифікацій для методу дерева рішень

Метод опорних векторів (SVM) досягнув точності приблизно 0.81, відгуку приблизно 0.82, точності приблизно 0.8 та F1 приблизно 0.81. Висока ефективність пояснюється використанням лінійного ядра для розподілу класів (рис 4.5).

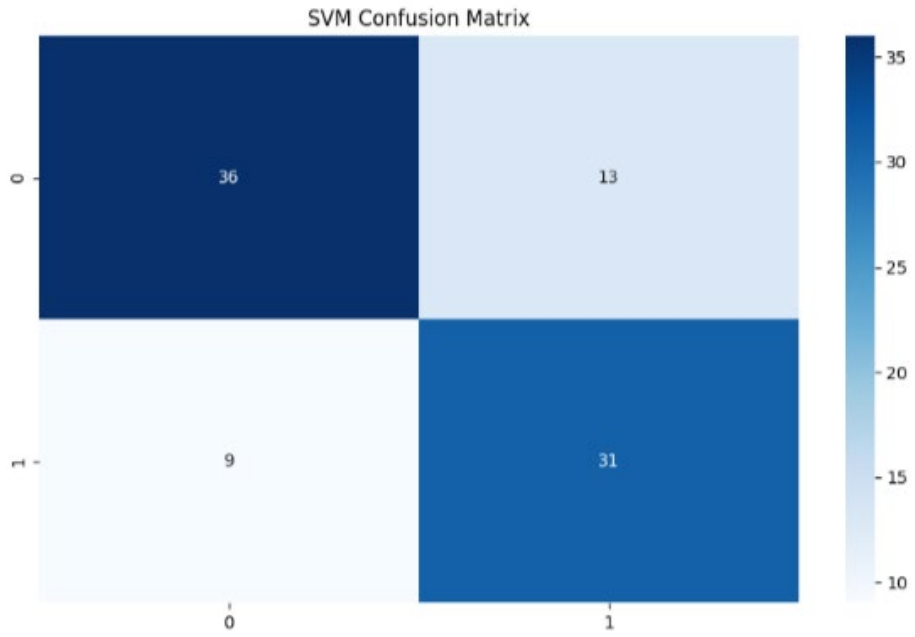

Рисунок 4.5 – Матриця невірних класифікацій для методу опорних векторів

Наївний байєс продемонстрував точність приблизно 0.74, відгук приблизно 0.76, точність приблизно 0.72 та F1 приблизно 0.74. Модель з низькою точністю ймовірно через припущення про незалежність ознак, що не відповідає реальному розподілу в даних (рис. 4.6).

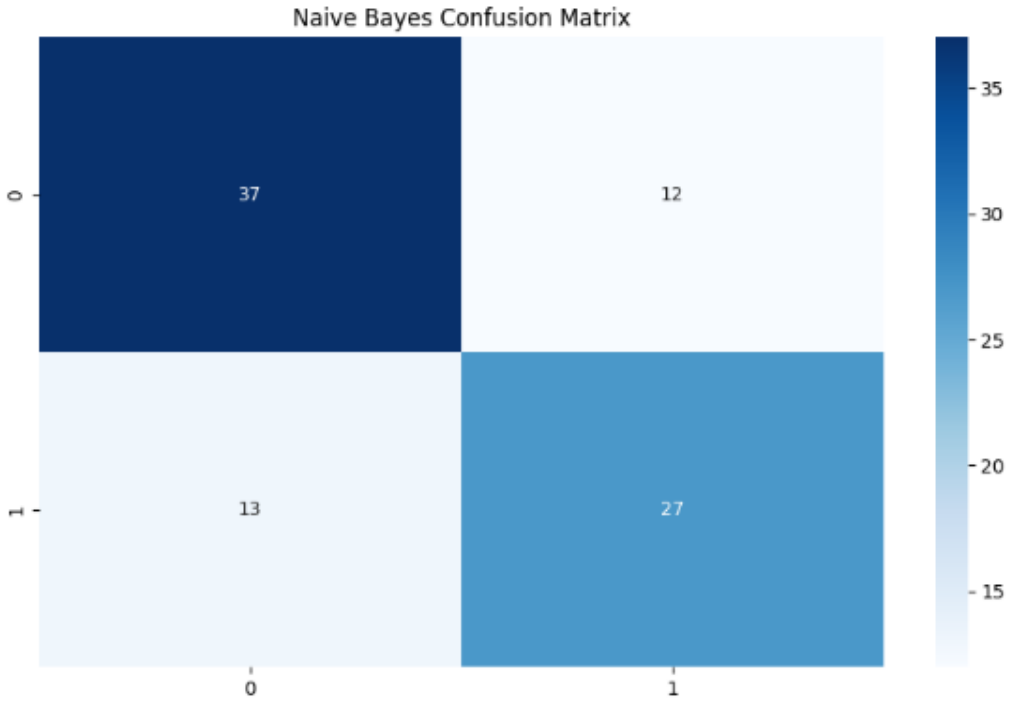

Рисунок 4.6 – Матриця невірних класифікацій для методу наївного баєса

Нейронна мережа (MLP) показала точність приблизно 0.78, відгук приблизно 0.8, точність приблизно 0.76 та F1 приблизно 0.78. Хоча модель є потужною, вона не змогла перевершити SVM або градієнтний бустинг (рис. 4.7).

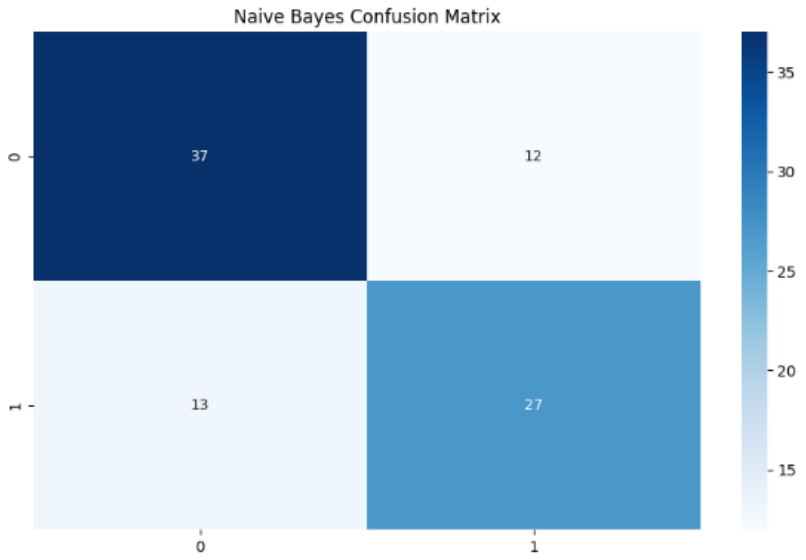

Рисунок 4.7 – Матриця невірних класифікацій для нейронної мережі

Градієнтний бустинг досягнув найкращих результатів серед усіх моделей, з точністю приблизно 0.82, відгуком приблизно 0.83, точністю приблизно 0.81 та F1 приблизно 0.82. Цей алгоритм відзначається високою стійкістю до перенавчання та здатністю до складних нелінійних перетворень (рис. 4.8).

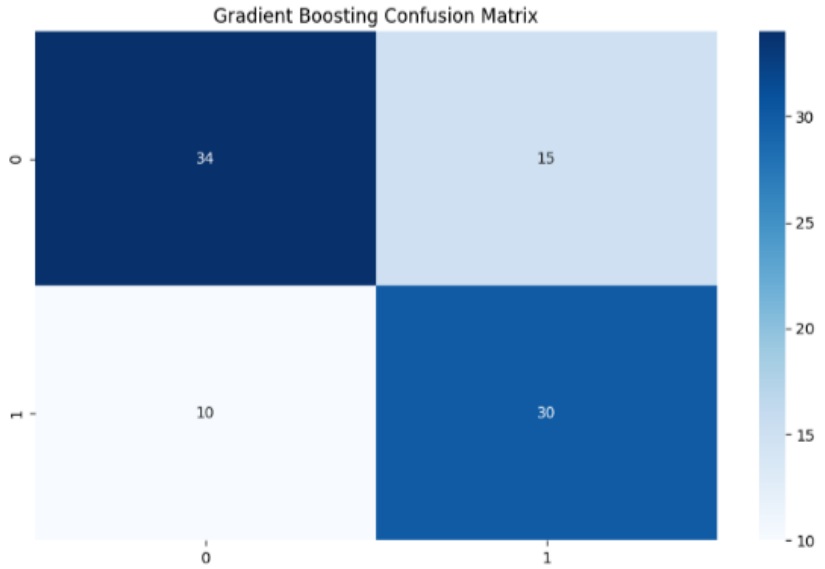

Рисунок 4.8 – Матриця невірних класифікацій для градієнтного бустінгу

4.3 Аналіз результатів методів пояснення

Для пояснення результатів, використано також методику SHAP (Shapley Additive Explanations). Цей інструмент дозволяє зрозуміти, як кожна ознака впливає на передбачення моделі. Результати для градієнтного бустингу показали, що найбільший вплив на прогноз мали кілька ключових ознак, таких як «bmi», «bp» та «s1» (рис. 4.9).

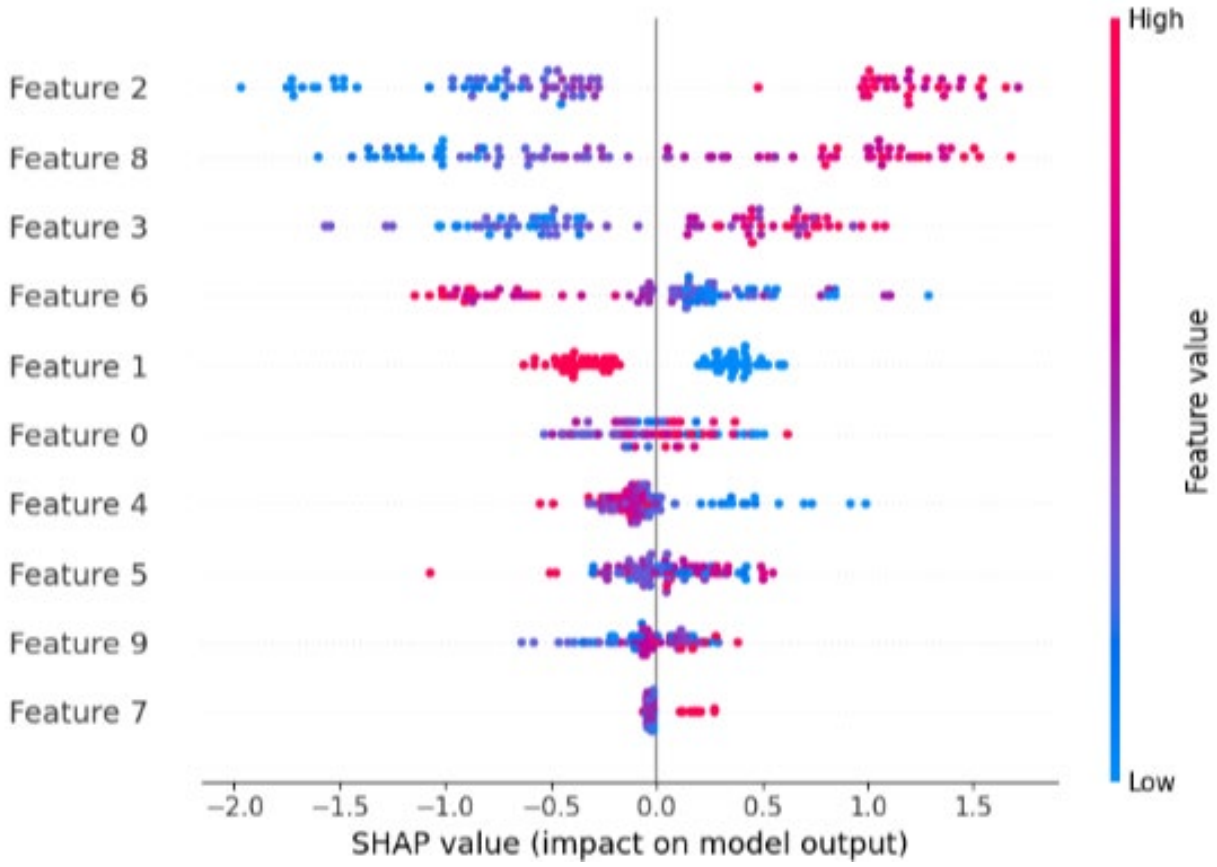

Рисунок 4.9 – Графік методу SHAP

З метою зниження розмірності даних було застосовано метод головних компонент (PCA). Після зниження розмірності до двох компонент, побудовано розсіювальну діаграму для візуалізації класифікації (рис. 4.10).

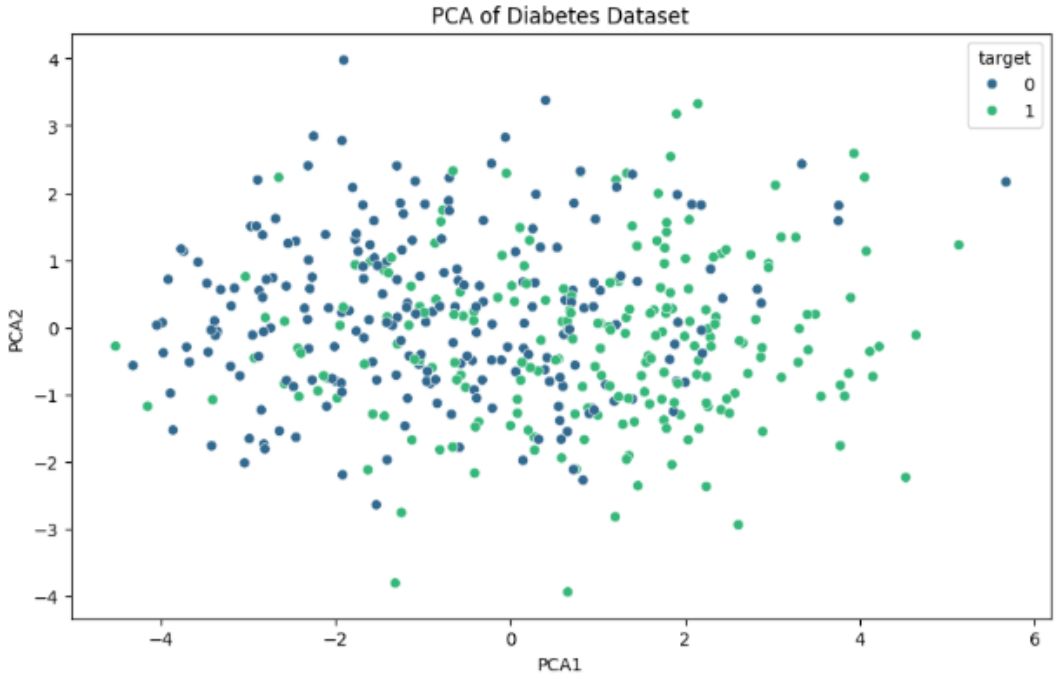

Рисунок 4.10 – Графік розсіювання для двох головних компонент

Результати експерименту демонструють, що градієнтний бустинг є найефективнішою моделлю для цього набору даних, з найбільшими показниками точності та F1-міри.

Висновки за розділом 4

Результати обчислювального експерименту показали, що різні алгоритми машинного навчання мають свої сильні та слабкі сторони залежно від особливостей набору даних про діабет. Серед протестованих моделей найвищу ефективність продемонстрував градієнтний бустинг, який досягнув найкращих показників точності, відгуку та F1-міри завдяки здатності моделювати складні нелінійні залежності та стійкості до перенавчання.

Метод опорних векторів також показав високі результати, особливо при використанні лінійного ядра, яке забезпечило ефективний поділ класів. Нейронна мережа (MLP) продемонструвала потужність у моделюванні складних залежностей, однак її результати виявилися менш точними, ніж у SVM та граді-

єнтного бустингу, що може бути пов'язано з обмеженою кількістю даних або налаштуваннями гіперпараметрів.

Наївний баєсівський класифікатор та метод дерева рішень мали нижчу точність порівняно з іншими моделями. Це пояснюється обмеженнями наївних припущень про незалежність ознак у випадку байєсівського класифікатора та схильністю дерева рішень до перенавчання. Метод K-найближчих сусідів, хоча і показав середні результати, виявився чутливим до вибору кількості класів та розподілу даних.

Додатковий аналіз за допомогою SHAP підтвердив значущість ключових ознак, таких як «bmi», «bp» та «s1», для точного прогнозування. Використання PCA для візуалізації даних продемонструвало ефективність зниження розмірності, дозволяючи краще зрозуміти структуру даних. Таким чином, градієнтний бустинг підтвердив свою придатність для класифікації у цьому наборі даних і є найкращим вибором серед протестованих алгоритмів.

ВИСНОВКИ

Під час дослідження виконано ключові завдання щодо аналізу та розробки методів прогнозування медичних діагнозів з використанням сучасних підходів машинного навчання. Проведено системний аналіз предметної області, зокрема прогнозування діабету, та обґрунтовано доцільність використання певних методів для розв’язання цієї задачі. Проаналізовано сценарії вирішення задачі прогнозування медичних діагнозів, здійснено формальну та змістовну постановку задачі, що забезпечило точне формулювання цілей і вимог до моделей.

У рамках дослідження виконано огляд існуючих методів прогнозування, таких як метод найближчих сусідів, дерева рішень, метод опорних векторів, баєсівський класифікатор, штучні нейронні мережі та метод екстремального градієнтного бустингу. З метою підвищення точності та інтерпретованості моделей, було застосовано методи SHAP та LAIM для розрахунку важливості ознак, що дозволяє ідентифікувати основні фактори, що впливають на точність прогнозування діагнозів.

Результати цього дослідження можна застосовувати для розробки систем підтримки прийняття рішень у медичній сфері, зокрема для прогнозування ризику виникнення діабету та інших хронічних захворювань. Запропоновані методи можуть бути інтегровані у програмні комплекси для медичних установ, що сприятиме покращенню якості діагностики та зниженню ризиків помилкових діагнозів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Мартиненко М. Ю. Математичні методи у моделюванні процесів машинного навчання для медичних задач : монографія. Київ : ВД Академія, 2019. 320 с.

2. Ломакін О. В., Черняк Д. Ю. Прикладні моделі машинного навчання у прогнозуванні хронічних захворювань. *Біомедична інформатика.* 2020. № 1 (58). С. 45–54.

3. Орлов С. І., Пушкін О. М., Давидова І. М. Прогнозування ризику захворювань методами машинного навчання. Київ : Наук. думка, 2021. 264 с.

4. Bishop C. M. Pattern Recognition and Machine Learning. New York : Springer, 2006. 738 p.

5. Petunin A. V., Avdeev V. A., Lomako E. V. Mathematical models for predicting disease progression. *Computational and Mathematical Methods in Medicine.* 2017. Vol. 2017, Article ID 2341256. pp. 1–15.

6. Murphy K. P. Machine Learning: A Probabilistic Perspective. Cambridge : The MIT Press, 2012. 1067 p.

7. Федорова О. В., Сидоров М. В. Методи машинного навчання у медицині. *Сучасні інформаційні технології.* 2019. Т. 5, № 2. С. 31–40.

8. Гречка І. Ю., Зубрій Т. О., Краснопольський В. В. Математичне моделювання в медичній діагностиці за допомогою штучних нейронних мереж : навч. посіб. Одеса : ОНУ, 2020. 220 с.

9. Aggarwal C. C. Neural Networks and Deep Learning. New York : Springer, 2018. 497 p.

10. Duda R. O., Hart P. E., Stork D. G. Pattern Classification. 2nd ed. New York : Wiley, 2001. 654 p.

11. Колосова С. В., Поліщук І. В. Прогнозування захворювань серцево-судинної системи на основі методів машинного навчання. *Біомедична інженерія.* 2018. № 4. С. 15–21.

12. Мартинюк В. П., Сафонова О. А., Котенко Д. М., Пантелєєв І. М. Побудова прогнозуючих моделей для медичних даних за допомогою математичної статистики. *Журнал обчислювальної математики*. 2015. № 3. С. 102–108.

13. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : The MIT Press, 2016. 775 p.

14. Лисенко П. А., Граненко І. В. Алгоритми обробки медичних даних для прогнозування результатів лікування. *Вісник наукових досліджень.* 2021. Т. 1. С. 30–37.

Ющенко П. М., Суворова Л. С., Горіна Д. П. Методи штучного інтелекту для медичних прогнозів. *Інтелектуальні технології в медицині.* 2019. № 2. С. 20–28.