

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))
Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:
«Інформаційна технологія діагностування стану здоров'я людини»

Виконав: студент 2-го курсу, гр. 2КН-23м
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

Пасічник І.В.
(прізвище та ініціали)

Керівник: д.т.н., проф. каф. КН
Іванчук Я. В.
(прізвище та ініціали)

« 05 » 12 2023 р.

Опонент: д.т.н., проф. зав. каф. КСУ
Ковтун В. В.
(прізвище та ініціали)

« 05 » 12 2024 р.

Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
(прізвище та ініціали)
« 06 » 12 2024 р.

м. Вінниця - 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо - професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
проф., д.т.н. Яровий А.А.
« 11 » 10 2024 р.

ЗАВДАННЯ

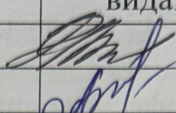
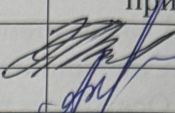
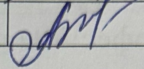
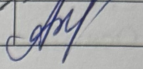
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Пасічнюку Івану Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна технологія діагностування стану здоров'я людини
керівник роботи: д.т.н., професор кафедри КН Іванчук Я.В.
затверджені наказом вищого навчального закладу від «17» 09 2024 року №310.
2. Строк подання студентом роботи 11.11 2024 р.
3. Вихідні дані до роботи: точність моделі не менше 80%; час прогнозування не повинен перевищувати 60 секунд для забезпечення ефективної роботи користувачів; має бути інтуїтивно зрозумілий інтерфейс для полегшення адаптації нових співробітників до системи.
4. Зміст текстової частини: вступ; Аналіз предметної області; розробка інформаційної технології діагностування стану здоров'я людини; програмна реалізація інформаційної технології діагностування стану здоров'я людини; економічна частина; висновок; перелік використаних джерел; додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): Основні класичні методи діагностики; принцип роботи методу градієнтного бустингу дерев рішень; принцип роботи методу випадкового лісу; принцип роботи методу дерева рішень; приклади роботи програми.

6. Консультанти розділів роботи

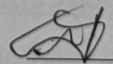
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Іванчук Я. В., проф. каф. КН		
4	Адлер О. О., доц. каф. ЕПВМ		

7. Дата видачі завдання 02.09.2024

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ з/п	Назва та зміст етапу	Термін виконання		Приміт
		початок	кінець	
1	Аналіз предметної області	02.09.24	09.09.24	
2	Проектування інформаційної технології діагностування стану здоров'я людини	10.09.24	28.09.24	
3	Програмна реалізація інформаційної технології діагностування стану здоров'я людини	29.09.24	18.10.24	
4	Економічна частина	19.10.24	01.11.24	
5	Апробація та/або впровадження результатів дослідження	2.11.24	4.11.24	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	05.11.24	11.11.24	

Студент


(підпис)

Пасічнюк

(прізвище та іні

Керівник роботи


(підпис)

Іванчук

(прізвище та іні

АНОТАЦІЯ

УДК 614.2

Пасічнюк І. В. Інформаційна технологія діагностування стану здоров'я людини. Магістерська кваліфікаційна робота за спеціальністю 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Вінниця: ВНТУ, 2024. 131 с. Українською мовою. Бібліогр.: 46 назв; рис.: 33; табл. 18.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології для прогнозування ризику серцевих захворювань на основі методів машинного навчання. Проаналізовано сучасні підходи до діагностики стану здоров'я людини, а також досліджено фактори ризику, які найбільше впливають на розвиток серцево-судинних захворювань. Серед існуючих алгоритмів машинного навчання для вирішення завдання було обрано метод градієнтного бустингу дерев рішень, який забезпечує високу точність класифікації. Розроблено програму, яка реалізує модель прогнозування на основі обраного методу та додатковий модуль для шифрування даних, що гарантує конфіденційність інформації про пацієнтів. Реалізація виконана мовою програмування C# у середовищі Visual Studio 2022. Розроблена система досягла точності 86.64%, F1 Score – 87.76% та AUC – 94.34%, що підтверджує її ефективність у прогнозуванні ризику серцевих захворювань.

Ключові слова: прогнозування здоров'я, серцеві захворювання, градієнтний бустинг дерев рішень, машинне навчання, шифрування даних.

ABSTRACT

Pasichniuk I.V. Information technology for diagnosing the state of human health. Master's qualification work on specialty 122 "Computer Science", educational program "Computer Science". Vinnytsia: VNTU, 2023. 105 p. In Ukrainian. Bibliography: 46 titles; Fig.: 33; table 12.

This master's thesis is devoted to the development of information technology for predicting the risk of heart diseases based on machine learning methods. Modern approaches to the diagnosis of the state of human health were analyzed, as well as the risk factors that most affect the development of cardiovascular diseases were investigated. Among the existing machine learning algorithms, the method of gradient boosting of decision trees was chosen to solve the problem, which ensures high classification accuracy. A program has been developed that implements a prediction model based on the chosen method and an additional module for data encryption, which guarantees the confidentiality of patient information. The implementation was performed in the C# programming language in the Visual Studio 2022 environment. The developed system achieved an accuracy of 86.64%, an F1 Score of 87.76% and an AUC of 94.34%, which confirms its effectiveness in predicting the risk of heart diseases.

Keywords: health prediction, heart diseases, gradient boosting of decision trees, machine learning, data encryption.

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд сучасних підходів до діагностики стану здоров'я людини.....	8
1.2 Вивчення факторів ризику серцевих захворювань та їх впливу на прогнозування стану здоров'я	14
1.3 Огляд існуючих програмних рішень.....	20
1.4 Постановка задачі	26
1.5 Висновок до розділу 1	28
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ СТАНУ ЗДОРОВ'Я ЛЮДИНИ	29
2.1 Аналіз та обґрунтування вибору методу машинного навчання для прогнозування стану здоров'я	29
2.2 Опис та обґрунтування вибору необхідного тренувального набору даних.....	38
2.3 Математична модель прогнозування та класифікації ризиків	40
2.4 Опис метрик для оцінки якості прогнозування та точності моделі.....	44
2.5 Висновок до розділу 2	47
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДІАГНОСТИКИ СТАНУ ЗДОРОВ'Я.....	48
3.1 Обґрунтування вибору мови програмування та технологічних засобів розробки	48
3.2 Реалізація програмного модуля для прогнозування ризику серцевих захворювань з використанням алгоритмів машинного навчання.....	52
3.3 Реалізація додаткових модулів системи	60
3.4 Тестування програми.....	65
3.5 Аналіз результатів роботи програми прогнозування стану здоров'я людини	72
3.6 Висновок до розділу 3	75
4 ЕКОНОМІЧНА ЧАСТИНА	77
4.1 Проведення комерційного та технологічного аудиту інформаційної технології діагностування стану здоров'я.....	77

4.2 Розрахунок витрат на здійснення науково-дослідної розробки інформаційної технології діагностування стану здоров'я.....	78
4.3 Розрахунок економічної ефективності науково-технічної розробки інформаційної технології діагностування стану здоров'я за її можливої комерціалізації потенційним інвестором	85
4.4 Висновок до розділу 4	89
ВИСНОВКИ.....	90
Додаток А (обов'язковий)_Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	97
Додаток Б (довідниковий)_Лістинг програми	98
Додаток В (Обов'язковий)_ІЛЮСТРАТИВНА ЧАСТИНА.....	112
Додаток Г (довідниковий)_Інструкція користувача	120

ВСТУП

Актуальність теми дослідження. В сучасному світі інформаційні технології набувають все більшого значення у сфері охорони здоров'я, оскільки вони здатні вирішувати низку критично важливих завдань, пов'язаних із забезпеченням точності та швидкості діагностування. Завдяки застосуванню штучного інтелекту (ШІ) та машинного навчання (МН), з'являється можливість підвищення рівня обробки та аналізу великих обсягів медичних даних, що дозволяє значно покращити діагностичні процедури, а також виявляти захворювання на ранніх стадіях. Однією з ключових проблем, яка постає перед системами охорони здоров'я, є необхідність швидкого і точного діагностування стану пацієнта, що потребує залучення новітніх технологій для обробки інформації та створення систем прогнозування.

Особливо актуальною ця проблема є для України, де впровадження технологій на основі ШІ та машинного навчання має перспективу не лише з точки зору покращення якості медичного обслуговування, але й оптимізації витрат. Сьогодні медичні установи все частіше стикаються з проблемою браку кваліфікованих кадрів, високим навантаженням на лікарів, а також недосконалістю методів обробки та зберігання медичної інформації. У такому контексті розвиток технологій автоматизованої діагностики стає невід'ємною складовою інноваційних рішень для охорони здоров'я, здатних допомогти лікарям та знизити ризик діагностичних помилок [2, 3].

Використання моделей глибокого навчання дозволяє з високою точністю аналізувати зображення, сигнали та інші медичні дані, що сприяє не лише кращому діагностуванню, але й прогнозуванню можливих ускладнень. При цьому, розвиток хмарних технологій та інтернету речей (IoT) забезпечує віддалений доступ до діагностичних результатів, що є надзвичайно важливим у період пандемій та інших кризових ситуацій. Окрім того, критичний аналіз світового досвіду показує, що розвинені країни активно застосовують ШІ для створення систем підтримки медичних рішень, що включають не лише

діагностику, але й рекомендації щодо лікування, розробку терапевтичних програм та моніторинг пацієнтів у реальному часі. На відміну від багатьох традиційних методів, ці технології здатні швидко адаптуватися до змін у стані пацієнта та ефективно використовувати інформацію для підтримки рішень лікарів.

В Україні, де питання здоров'я нації є стратегічним пріоритетом, впровадження таких технологій може значно покращити стан охорони здоров'я, особливо в умовах, коли медична допомога у віддалених регіонах часто обмежена. Завдяки застосуванню автоматизованих систем діагностики, які враховують локальні особливості та потреби, можливо створити умови для більш доступної та якісної медицини. Використання алгоритмів машинного навчання дозволить адаптувати існуючі світові моделі до особливостей українського населення, враховуючи екологічні, демографічні та інші фактори.

Зв'язок роботи з науковими програмами, планами, темами.

Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка прикладних інтелектуальних інформаційних технологій та систем» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження є розробка інформаційної технології прогнозування стану здоров'я людини, яка дозволить здійснювати автоматичну діагностику ризиків розвитку серцевих захворювань на основі аналізу медичних даних з використанням методів машинного навчання.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести аналіз сучасних підходів до діагностики та прогнозування стану здоров'я людини, зокрема серцево-судинних захворювань, для визначення ключових методів і критеріїв оцінки;

- дослідити основні фактори ризику серцевих захворювань, їх вплив на стан здоров'я пацієнта та необхідність врахування цих факторів у прогнозах моделях;
- розробити математичну модель прогнозування ризиків із застосуванням обґрунтованого методу машинного навчання та побудувати класифікаційний алгоритм для оцінки ймовірності розвитку захворювань;
- реалізувати програмний модуль для прогнозування ризиків та провести тестування розробленої інформаційної технології з метою оцінки її точності у діагностиці.

Об'єкт дослідження – це процес діагностування ризику серцевих захворювань пацієнтів з використанням методів машинного навчання.

Предмет дослідження – це алгоритми та програмні засоби, що застосовуються для визначення ризиків серцево-судинних захворювань шляхом аналізу медичних показників, таких як артеріальний тиск, рівень холестерину, індекс маси тіла, наявність шкідливих звичок та інші фактори.

Методи дослідження. У даній роботі застосовано такі методи наукових досліджень: метод системного аналізу для детального вивчення структури та компонентів інформаційної технології діагностики; методи машинного навчання, зокрема алгоритми класифікації, для побудови моделі прогнозування ризику серцевих захворювань; методи математичної статистики для аналізу та оцінки ефективності створеної моделі; методи об'єктно-орієнтованого програмування для реалізації програмного забезпечення, що дозволяє автоматизувати процес прогнозування на основі вхідних даних.

Наукова новизна одержаних результатів полягає у наступному:

- вперше запропоновано комплексну інформаційну технологію діагностики ризиків серцевих захворювань, яка інтегрує численні медичні показники (таких як артеріальний тиск, індекс маси тіла, рівень холестерину тощо) для підвищення точності прогнозування за допомогою алгоритмів машинного навчання, що дозволяє виявляти потенційні ризики з більшою достовірністю порівняно з традиційними методами;

- удосконалено процес класифікації медичних даних за рахунок впровадження процедури нормалізації та обробки відсутніх значень, що дозволило підвищити ефективність навчання та точність моделі, а також мінімізувати вплив нерелевантних даних на прогнозування стану здоров'я пацієнтів.

Практичне значення одержаних результатів полягає у наступному:

- розроблено програмне забезпечення для діагностування ризиків серцево-судинних захворювань на основі машинного навчання, що може використовуватися у медичних закладах для підвищення точності та оперативності діагностичних рішень;
- запропоновані методи обробки та аналізу даних дозволяють інтегрувати отримане рішення у хмарні сервіси або мобільні додатки, забезпечуючи віддалений моніторинг стану здоров'я пацієнтів, що є особливо актуальним для віддалених регіонів;
- розроблена технологія може бути використана в освітніх процесах як практичний приклад застосування методів машинного навчання у сфері медицини, що сприяє формуванню компетенцій у майбутніх спеціалістів у галузі штучного інтелекту та охорони здоров'я.

Достовірність теоретичних положень магістерської кваліфікаційної роботи обґрунтована чіткістю постановки задач, обґрунтованим вибором методів машинного навчання для досягнення поставленої мети, а також коректним застосуванням математичних і статистичних методів для оцінки точності прогнозування. Додаткова перевірка отриманих результатів шляхом тестування на реальних медичних даних підтверджує адекватність моделі, що розроблена, та збіжність її прогнозів з очікуваними результатами в рамках тестових сценаріїв.

Апробація результатів роботи. Результати були апробовані на науково-технічній конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)» [1].

Публікації. За результатами досліджень опубліковано тези доповіді [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд сучасних підходів до діагностики стану здоров'я людини

У медицині поняття «здоров'я» часто асоціюється з нормою, яка визначається сукупністю середніх фізіологічних показників: температура тіла, рівень глюкози в крові, кількість еритроцитів, артеріальний тиск та інші параметри. Однак, ці значення нормальних показників можуть значно варіюватися залежно від віку, статі, способу життя, клімату і навіть географічних умов, що свідчить про їхню відносність. Середні показники "норми" не завжди відображають здатність організму до адаптації та опору впливу зовнішніх чинників, таких як фізичні навантаження, екологічні умови або наявність захворювань. З огляду на це, «здоров'я» варто розглядати не як статичний стан, а як динамічний процес, що характеризується адаптаційними можливостями організму, його здатністю до пристосування та функціональної активності. Це дозволяє ширше зрозуміти поняття «здоров'я» як баланс між станом організму та його здатністю ефективно реагувати на різні зовнішні та внутрішні виклики [6].

Хвороба, в свою чергу, традиційно розглядається як протилежний стан до здоров'я, де, за визначенням ВООЗ, йдеться про будь-які суб'єктивні або об'єктивні відхилення від нормального фізіологічного стану. Проте таке визначення є радше загальним і не завжди застосовується в практичному розумінні стану пацієнта. Хвороба не заперечує існування здоров'я як такого – часто виявляється не хвороба, а дефіцит певних аспектів здоров'я. Між повним здоров'ям і хворобою існує безліч проміжних станів, що можна трактувати як особливі форми адаптації організму, які знаходяться десь на межі між здоров'ям і захворюванням. Ці стани показують важливість індивідуальних резервів та можливостей організму до адаптації, що дозволяє більш точно та обґрунтовано підходити до діагностики та прогнозування здоров'я людини.

Діагностика стану здоров'я є важливим етапом у профілактиці, виявленні та лікуванні захворювань. З розвитком цифрових технологій та ШІ у медицині з'являються нові підходи, що дозволяють підвищити точність і швидкість діагностики. Сучасні методи діагностики базуються на застосуванні великих даних, алгоритмів МН, хмарних технологій, IoT та різноманітних сенсорів, що забезпечують моніторинг та обробку медичних даних[7].

Класичні методи діагностики – це сукупність стандартних медичних процедур та тестів, які використовуються для оцінки стану здоров'я пацієнта. Вони базуються на фізіологічних параметрах і дозволяють отримати інформацію про різні аспекти функціонування організму, необхідну для постановки діагнозу. Ці методи є невід'ємною частиною медичної практики, оскільки забезпечують базову та важливу інформацію для лікаря, від якої залежить подальше обстеження пацієнта. Традиційно їх виконують у медичних установах під наглядом фахівців.

На рисунку 1.1 наведена діаграма основних класичних методів діагностики, що демонструє ключові підходи до обстеження пацієнта.

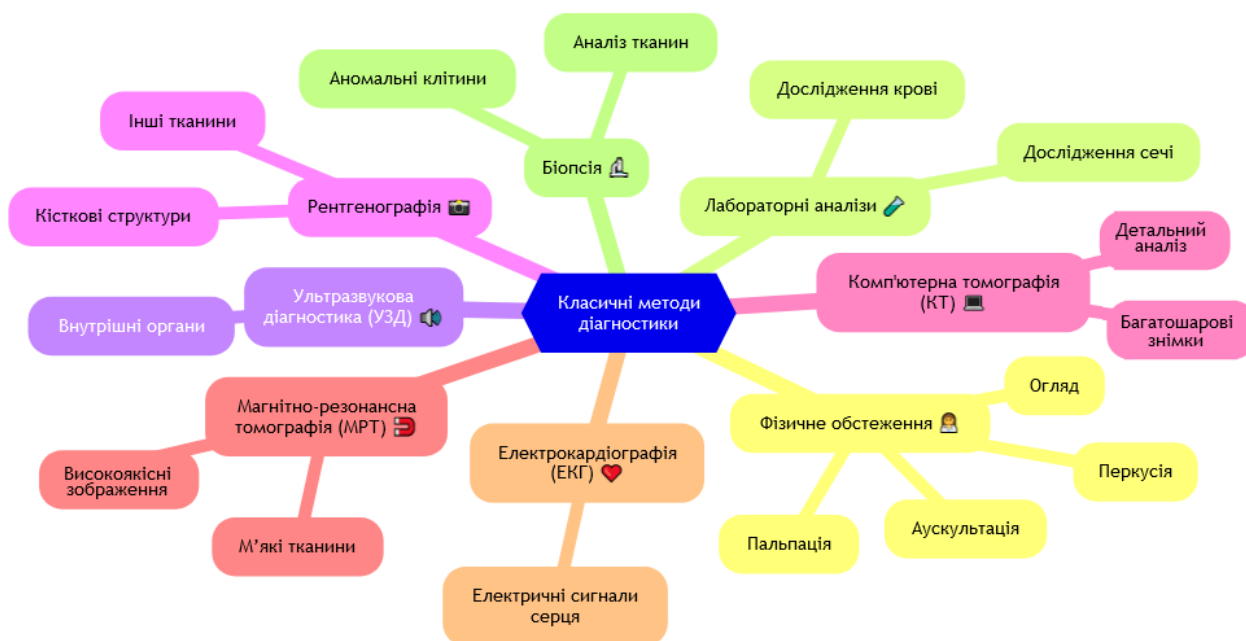


Рисунок 1.1 – Основні класичні методи діагностики

До основних класичних методів діагностики відносяться:

- фізичне обстеження, яке включає огляд, пальпацію, перкусію і аускультацию для оцінки зовнішнього стану органів і систем;
- лабораторні аналізи, до яких відносяться дослідження біоматеріалів (крові, сечі) для виявлення показників, що відображають стан здоров'я;
- ультразвукова діагностика (УЗД), яка використовує звукові хвилі для створення зображень внутрішніх органів;
- рентгенографія для застосування рентгенівських променів для дослідження кісткових структур та інших тканин;
- комп'ютерна томографія (КТ), яка робить багат шарові знімки для детального аналізу внутрішніх органів та структур;
- магнітно-резонансна томографія (МРТ), яка використовує магнітне поле для створення високоякісних зображень м'яких тканин;
- електрокардіографія (ЕКГ), яка реєструє електричних сигналів серця для діагностики його роботи;
- біопсія, яка проводить забір тканин для лабораторного дослідження з метою виявлення аномальних клітин.

Перелічені методи забезпечують основну інформацію про фізіологічні та анатомічні аспекти здоров'я пацієнта, надаючи лікарю змогу побачити загальну картину стану організму. Незважаючи на появу новітніх технологій, класичні методи діагностики залишаються актуальними, оскільки дозволяють точно та комплексно оцінити стан пацієнта. Вони є важливим етапом перед детальнішим обстеженням за допомогою спеціалізованих методів [8].

ІІІ та МН – це сучасні технології, які надають можливість автоматизувати діагностичні процеси та підвищити точність прогнозування стану здоров'я пацієнтів. Ці методи базуються на алгоритмах, здатних аналізувати великі обсяги медичних даних і знаходити приховані закономірності, що можуть бути непомітними при традиційних підходах. Використання ІІІ та МН дозволяє створювати моделі для діагностики різноманітних захворювань, зокрема

серцево-судинних, онкологічних та неврологічних розладів, а також прогнозувати ймовірність ускладнень на основі індивідуальних медичних показників пацієнта [9].

На рисунку 1.2 представлена діаграма, що демонструє основні методи діагностики, що використовують штучний інтелект та машинне навчання. Ці технології включають різні підходи, такі як нейронні мережі, обробка природної мови, глибоке навчання та інші методи, що допомагають підвищити ефективність та швидкість діагностичних процесів.



Рисунок 1.2 – Методи діагностики на основі ШІ та МН

До основних методів діагностики на основі ШІ та МН:

- нейронні мережі – використовуються для обробки складних медичних зображень і аналізу структур;
- глибоке навчання – залучає багаторівневі моделі для високоточного розпізнавання патологій на знімках;
- обробка природної мови (NLP) – аналізує медичні записи для автоматичного виділення ключової інформації[10];

- класифікація – визначає ризик розвитку захворювань на основі групування подібних випадків;
- регресійний аналіз – прогнозує ймовірність розвитку захворювань за допомогою числових показників;
- кластеризації – групують пацієнтів за схожими характеристиками для персоналізованого лікування;
- аналіз часу життя (Survival Analysis) – прогнозує тривалість життя пацієнтів із різними діагнозами[11];
- аналіз аномалій – виявляє нетипові медичні показники, що можуть вказувати на захворювання.

Застосування цих методів забезпечує високу точність діагностики та дозволяє вчасно виявляти потенційні ризики для здоров'я пацієнта. ШІ та машинне навчання допомагають автоматизувати процеси, підвищують швидкість обробки медичних даних і забезпечують індивідуальний підхід до кожного пацієнта. Ці методи стають незамінними в сучасній медицині, забезпечуючи лікарів цінною інформацією для прийняття оптимальних рішень.

IoT та сенсори для моніторингу здоров'я – це система взаємозв'язаних пристроїв, які дозволяють у реальному часі відстежувати життєві показники пацієнтів і передавати їх на медичні платформи для аналізу. Ці технології допомагають забезпечити постійний контроль за здоров'ям, особливо для людей з хронічними захворюваннями або тих, хто потребує регулярного моніторингу. Сенсори, вбудовані у пристрої, такі як смарт-годинники чи фітнес-трекери, вимірюють такі показники, як частота серцевих скорочень, рівень кисню в крові, рівень глюкози тощо, а IoT забезпечує обмін цими даними між пристроями та системами.

На рисунку 1.3 зображено основні методи використання IoT та сенсорів у медичній діагностиці. Ці технології охоплюють різні типи датчиків і пристроїв, що дозволяють отримувати широкий спектр інформації для оцінки стану здоров'я в режимі реального часу.



Рисунок 1.3 – Методи діагностики на основі IoT

Основні методи моніторингу здоров'я на основі IoT та сенсорів:

- сенсори серцевого ритму – вимірюють частоту серцевих скорочень для контролю за серцевою діяльністю;
- пульсоксиметри – контролюють рівень кисню в крові, що важливо для пацієнтів з респіраторними захворюваннями;
- глюкометри – вимірюють рівень глюкози, особливо важливо для пацієнтів з діабетом;
- сенсори температури – відстежують температуру тіла, дозволяючи швидко виявити ознаки запалення;
- датчики артеріального тиску – постійно вимірюють тиск для моніторингу гіпертонії;
- фітнес-трекери – відстежують рівень фізичної активності та загальний стан здоров'я;
- смарт-ваги – аналізують вагу і склад тіла для оцінки загальної фізичної форми;
- системи для моніторингу сну – відстежують цикли сну, допомагаючи виявити проблеми з відпочинком та відновленням.

Застосування IoT та сенсорів у діагностиці забезпечує постійний моніторинг стану здоров'я пацієнтів, дозволяючи лікарям оперативно реагувати на будь-які відхилення від норми. Ці технології сприяють індивідуальному підходу до лікування, забезпечуючи вчасну профілактику та корекцію лікувальних заходів, що особливо важливо для пацієнтів із хронічними хворобами [12].

Сучасні підходи до діагностики стану здоров'я людини інтегрують інноваційні технології та методи обробки даних, що підвищує їхню точність і оперативність. Використання ІІ та МН дозволяє аналізувати медичні показники більш точно та швидко, що є критичним для профілактики та своєчасного лікування. Інтернет речей і хмарні технології забезпечують моніторинг здоров'я у реальному часі, що відкриває нові можливості для віддаленого медичного обслуговування. Ці технології створюють нові перспективи для підвищення ефективності діагностичних процесів та забезпечення якісної медичної допомоги для всіх пацієнтів, незалежно від їхнього місця проживання чи доступу до медичних установ.

1.2 Вивчення факторів ризику серцевих захворювань та їх впливу на прогнозування стану здоров'я

Серцево-судинні захворювання є однією з провідних причин смертності у світі, що вимагає детального вивчення факторів ризику для їх ефективного прогнозування та профілактики. Основними факторами, які впливають на ризик розвитку серцевих захворювань, є фізіологічні показники, спосіб життя, генетична схильність, а також соціально-економічні умови. Застосування методів машинного навчання дозволяє здійснити аналіз та інтеграцію цих факторів для створення моделей прогнозування, що сприяє ранньому виявленню потенційних загроз здоров'ю.

До основних факторів ризику серцевих захворювань відносяться:

Артеріальний тиск (АТ) – це сила, з якою кров тисне на стінки судин під час її циркуляції по тілу. Він є одним з основних показників стану серцево-судинної системи і використовується для оцінки ризику розвитку серцевих захворювань, таких як гіпертонія, інфаркт міокарда та інсульт. Показники артеріального тиску зазвичай поділяються на систолічний (вимірюється під час скорочення серця) і діастолічний (вимірюється у фазі розслаблення). Регулярний моніторинг АТ дозволяє вчасно виявити відхилення від норми та запобігти серйозним ускладненням для здоров'я.

На основі показників артеріального тиску формується загальна картина стану пацієнта, що дозволяє лікарям оцінити ризики серцево-судинних захворювань. У таблиці 1.1 наведені середні значення артеріального тиску в різних категоріях пацієнтів, що дозволяє краще зрозуміти, як різні рівні тиску можуть впливати на стан здоров'я.

Таблиця 1.1 – Вплив артеріального тиску на ризик серцевих захворювань

Категорія пацієнта	Систолічний тиск (мм рт. ст.)	Діастолічний тиск (мм рт. ст.)	Ризик серцевих захворювань	Рекомендовані заходи
Нормальний рівень	120–129	80–84	Низький	Підтримка здорового способу життя
Підвищений рівень	130–139	85–89	Середній	Контроль за тиском, зміна раціону
Гіпертонія I ступеня	140–159	90–99	Високий	Прийом ліків, регулярний моніторинг
Гіпертонія II ступеня	160–179	100–109	Дуже високий	Медикаментозна терапія
Гіпертонія III ступеня	>180	>110	Критичний	Невідкладна допомога

Показники артеріального тиску є важливим індикатором стану серцево-судинної системи. Регулярне вимірювання та оцінка АТ допомагають

попередити розвиток серцевих захворювань, а також дозволяють лікарям планувати індивідуальні профілактичні або лікувальні заходи для кожного пацієнта. Контроль рівня АТ є основним елементом у попередженні ризиків, пов'язаних із серцево-судинними ускладненнями [13].

Індекс маси тіла (ІМТ) – це числовий показник, який визначається співвідношенням маси тіла до квадрату зросту людини (вимірюється в $\text{кг}/\text{м}^2$) і використовується для оцінки ваги та можливих ризиків для здоров'я. ІМТ є широко застосовуваним інструментом для класифікації маси тіла в категорії, такі як недостатня вага, нормальна вага, надмірна вага та ожиріння. Ожиріння, яке визначається високим рівнем ІМТ, пов'язане з підвищеним ризиком розвитку серцево-судинних захворювань, оскільки зайва вага створює додаткове навантаження на серце і кровоносні судини.

У таблиці 1.2 наведені категорії ІМТ та відповідні ризики для здоров'я, що допомагає лікарям оцінити вплив маси тіла на серцево-судинну систему пацієнта.

Таблиця 1.2 –ІМТ та ризик серцево-судинних захворювань

Категорія ІМТ	ІМТ ($\text{кг}/\text{м}^2$)	Ризик	Рекомендовані заходи
Недостатня вага	<18.5	Низький	Збалансоване харчування
Нормальна вага	18.5–24.9	Низький	Підтримка здорового способу життя
Надмірна вага	25.0–29.9	Помірний	Контроль харчування, фізична активність
Ожиріння I ступеня	30.0–34.9	Високий	Програма зниження ваги
Ожиріння II ступеня	35.0–39.9	Дуже високий	Медичний контроль та корекція ваги
Ожиріння III ступеня	≥ 40	Критичний	Інтенсивна терапія, можливо хірургія

ІМТ є важливим показником для оцінки ризику серцево-судинних захворювань, оскільки зайва вага негативно впливає на роботу серця та судин.

Контроль і корекція ІМТ допомагають зменшити ймовірність розвитку серцевих ускладнень і забезпечити тривале здоров'я пацієнтів, особливо тих, хто знаходиться у високоризикових категоріях [14].

Шкідливі звички, такі як куріння та вживання алкоголю, є значущими факторами ризику для серцево-судинної системи. На основі частоти куріння та рівня вживання алкоголю можна оцінити ризики для серцево-судинної системи і надати пацієнтам рекомендації щодо корекції цих звичок. У таблиці 1.3 наведені категорії ризику для серцевого здоров'я залежно від рівня вживання тютюну та алкоголю.

Таблиця 1.3 – Вплив шкідливих звичок на серцево-судинні захворювання

Шкідлива звичка	Рівень вживання	Ризик серцевих захворювань	Рекомендовані заходи
Куріння	Легке (до 5 цигарок/день)	Помірний	Скорочення споживання
Куріння	Середнє (5-15 цигарок/день)	Високий	Програма відмови від куріння
Куріння	Високе (>15 цигарок/день)	Дуже високий	Медична підтримка для відмови
Алкоголь	Помірне	Низький	Контроль вживання
Алкоголь	Регулярне	Високий	Обмеження до рекомендованих норм
Алкоголь	Надмірне	Дуже високий	Консультація з лікарем, відмова

Даний підхід до оцінки ризику допомагає виявити взаємозв'язок між шкідливими звичками і серцевими захворюваннями, а також розробити рекомендації щодо їх усунення або зменшення негативного впливу. Куріння і надмірне вживання алкоголю суттєво збільшують ризик серцевих патологій, і корекція цих звичок є важливим елементом профілактики серцево-судинних захворювань [15].

Генетична схильність і спадковість відіграють важливу роль у розвитку серцево-судинних захворювань, оскільки певні гени можуть підвищувати ризик

таких станів, як гіпертонія, атеросклероз, кардіоміопатія та серцеві напади. Наявність в анамнезі родичів з серцевими захворюваннями значно збільшує ймовірність розвитку подібних проблем у наступних поколіннях. Ці генетичні особливості впливають на метаболізм, обмін речовин, рівень холестерину та артеріальний тиск, що може сприяти прискоренню розвитку серцевих патологій. Важливо зазначити, що генетична схильність не є вироком, але вона створює передумови для виникнення захворювань за наявності інших факторів ризику, таких як малорухливий спосіб життя, шкідливі звички та неправильне харчування.

У таблиці 1.4 наведені категорії ризику серцевих захворювань залежно від наявності спадкових факторів, що дозволяє оцінити рівень ризику та необхідність профілактичних заходів.

Таблиця 1.4 – Вплив генетичної схильності на ризик захворювань

Сімейна історія	Ризик захворювань у родичів	Ризик для пацієнта	Рекомендовані заходи
Відсутність серцевих захворювань	Низький	Низький	Підтримка здорового способу життя
Один випадок серед родичів	Помірний	Середній	Профілактичні заходи
Кілька випадків серед родичів	Високий	Підвищений	Регулярний моніторинг стану серця
Батьки із серцевими захворюваннями	Дуже високий	Високий	Медичний контроль, рання профілактика
Родичі з інфарктом у ранньому віці	Критичний	Дуже високий	Посилений медичний контроль
Відомі генетичні мутації	Високий	Максимальний	Генетичне консультування, комплексна терапія

Цей підхід допомагає оцінити індивідуальний ризик розвитку серцевих захворювань на основі сімейної історії та генетичних факторів. Наявність

генетичної схильності вимагає регулярного моніторингу та профілактичних заходів для зниження ризику, а також допомагає лікарям розробити індивідуальні рекомендації для пацієнтів. Генетична спадковість є одним із найважливіших чинників, який визначає необхідність ранньої діагностики та профілактики [16].

Соціально-економічні фактори відіграють значну роль у розвитку серцево-судинних захворювань, оскільки рівень доходу, освітній рівень, умови праці та соціальна підтримка впливають на загальний стан здоров'я.

На основі соціально-економічних показників можна оцінити загрозу для серцево-судинного здоров'я та визначити, які групи населення потребують додаткової підтримки або профілактичних заходів. У таблиці 1.5 наведено вплив соціально-економічних факторів на ризик серцевих захворювань, що допомагає класифікувати різні групи населення за рівнем ризику.

Таблиця 1.5 – Вплив соціально-економічних факторів

Соціально-економічний фактор	Категорія	Ризик серцевих захворювань	Рекомендовані заходи
Рівень доходу	Високий	Низький	Підтримка здорового способу життя
Рівень доходу	Низький	Високий	Профілактичні програми, доступ до медицини
Освітній рівень	Вищий	Помірний	Розширення профілактичних знань
Освітній рівень	Початковий	Високий	Програми підвищення обізнаності
Умови праці	Стабільні, без стресу	Низький	Підтримка здорового балансу
Умови праці	Нестабільні, високий стрес	Високий	Психологічна підтримка, управління стресом
Соціальна підтримка	Висока	Низький	Підтримка громадських ініціатив

Цей підхід до оцінки впливу соціально-економічних факторів допомагає виявити групи ризику та розробити відповідні стратегії для зменшення їхнього впливу на здоров'я. Соціально-економічні показники тісно пов'язані з ризиком серцево-судинних захворювань, а їхній контроль та врахування можуть допомогти у створенні ефективних програм профілактики та забезпеченні кращого доступу до медичних послуг [17].

Вивчення факторів ризику серцевих захворювань є важливим етапом для створення моделей прогнозування, що здатні виявляти індивідуальні ризики для пацієнтів на основі об'єктивних медичних даних. Використання методів машинного навчання дозволяє інтегрувати різноманітні фактори ризику у єдину систему, яка враховує фізичні, поведінкові та генетичні особливості пацієнтів. Такий підхід сприяє підвищенню точності прогнозування, надає можливість вчасного втручання та сприяє розвитку персоналізованої медицини, що особливо важливо для попередження серйозних ускладнень та зниження рівня смертності від серцево-судинних захворювань.

1.3 Огляд існуючих програмних рішень

Розвиток інформаційних технологій у сфері медицини значно покращив процеси діагностики, прогнозування та лікування серцево-судинних захворювань. Існуючі програмні рішення для моніторингу здоров'я пацієнтів, аналізу медичних даних та прогнозування ризиків захворювань базуються на штучному інтелекті, машинному навчанні та обробці великих даних. Кожне з рішень має свої особливості та функціональні можливості, що дозволяють медичним працівникам приймати точніші рішення та підвищувати рівень персоналізованого підходу.

Ada – мобільний застосунок, розроблений компанією Ada Health, що використовує штучний інтелект для оцінки симптомів та надання персоналізованих рекомендацій щодо здоров'я (рис. 1.4). Користувач вводить інформацію про свої симптоми, після чого застосунок задає серію уточнюючих

питань, адаптуючи їх на основі попередніх відповідей. На основі введених даних та медичної бази знань, яка охоплює тисячі захворювань, Ada генерує можливі діагнози та рекомендації щодо подальших дій, таких як самостійний догляд або звернення до лікаря.

Архітектура застосунку складається з інтерфейсу користувача, що забезпечує взаємодію з користувачем, та серверної частини, де розміщені алгоритми штучного інтелекту та медична база знань. Алгоритми використовують методи машинного навчання для аналізу симптомів та визначення ймовірних діагнозів. Застосунок доступний на різних платформах і підтримує кілька мов, що робить його зручним для широкого кола користувачів.



Рисунок 1.4 – Приклад інтерфейсу системи «Ada» [18]

Переваги Ada:

- швидкий доступ до інформації. Ada надає швидкий і зручний спосіб отримати рекомендації щодо стану здоров'я без необхідності негайного відвідування лікаря;

- індивідуальний підхід [19]. Завдяки алгоритмам штучного інтелекту, додаток адаптує питання на основі відповідей, що дозволяє краще оцінювати симптоми;
- розширена медична база. Включає знання про тисячі захворювань, що підвищує точність прогнозів;
- доступність. Підтримує кілька мов і працює на різних платформах, роблячи додаток зручним для користувачів по всьому світу.

Недоліки Ada:

- відсутність прямого медичного діагнозу [20]. Ada надає лише можливі варіанти діагнозу, але не замінює професійний медичний висновок;
- можливість помилкових рекомендацій. Незважаючи на алгоритми, штучний інтелект може помилятися, що може призвести до хибних висновків;
- залежність від точності введених даних. Якість рекомендацій залежить від правильності та повноти введених користувачем симптомів;
- обмежений функціонал. Не надає безперервного моніторингу або медичної підтримки у реальному часі.

Отже, Ada – це корисний інструмент для первинної оцінки стану здоров'я, який дозволяє користувачам швидко отримати можливі діагнози та рекомендації щодо подальших дій. Однак, через обмеження в точності і можливість помилкових рекомендацій, додаток слід використовувати як допоміжний інструмент, а не як заміну професійного медичного обстеження.

SkinVision – мобільний застосунок, призначений для раннього виявлення ознак раку шкіри. Користувач фотографує підозрілу ділянку шкіри, після чого алгоритм на основі штучного інтелекту аналізує зображення та надає оцінку ризику протягом 30 секунд.

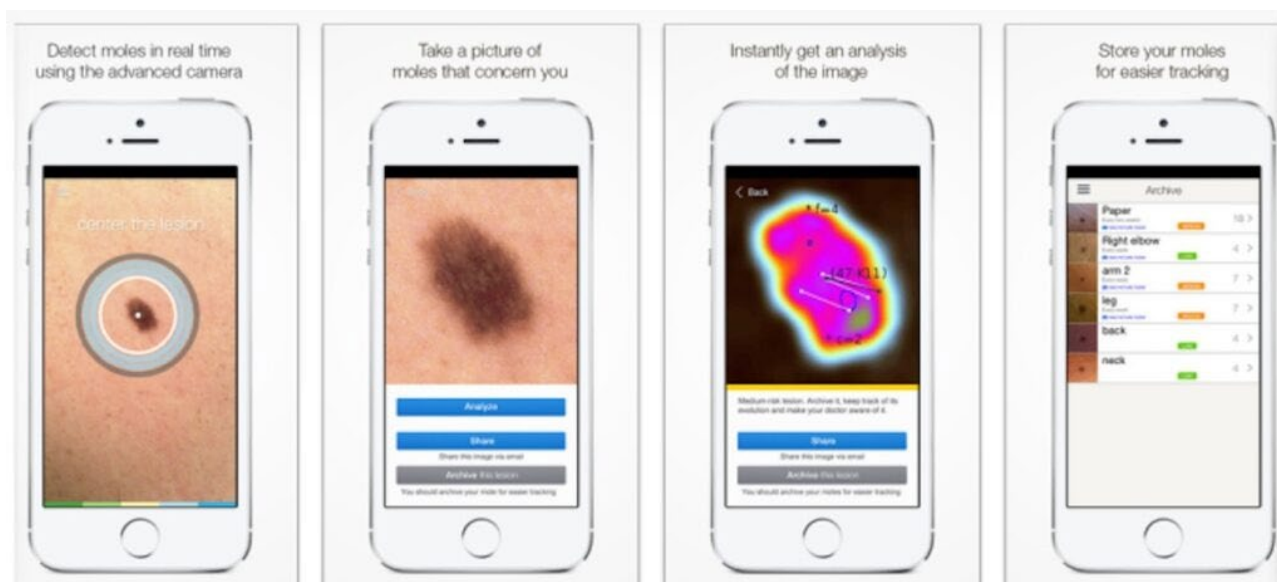


Рисунок 1.5 – Приклад інтерфейсу системи «SkinVision» [21]

Архітектура застосунку складається з інтерфейсу користувача, що забезпечує взаємодію з користувачем, та серверної частини, де розміщені алгоритми машинного навчання та база даних медичних зображень. Алгоритм навчений на великій кількості клінічно підтверджених зображень шкірних уражень, що дозволяє йому розпізнавати патерни, характерні для різних типів раку шкіри. Застосунок також надає можливість зберігати зображення для відстеження змін у часі та встановлювати нагадування для регулярних перевірок.

Переваги SkinVision:

- раннє виявлення раку шкіри. Дозволяє швидко оцінити ризик, допомагаючи користувачам виявити потенційно небезпечні новоутворення на ранніх стадіях;
- висока доступність. Користувачі можуть самостійно використовувати додаток у домашніх умовах, що спрощує моніторинг стану шкіри;
- збереження та відстеження змін [22]. Застосунок дозволяє зберігати зображення шкіри та встановлювати нагадування для регулярних перевірок;
- швидкість. Оцінка ризику займає лише кілька секунд, що є зручним для користувачів.

Недоліки SkinVision:

- обмеження в точності. Як і всі алгоритми, модель може помилятися в оцінці, особливо якщо фото зроблено у неякісних умовах освітлення;
- не замінює професійний огляд. SkinVision є допоміжним інструментом і не може замінити огляд дерматолога;
- залежність від якості зображення [23]. Недосконале фото може вплинути на точність оцінки, що може дати користувачеві хибне відчуття безпеки або навпаки;
- вартість послуг. Деякі функції доступні тільки у платній версії, що може бути недоліком для деяких користувачів.

Отже, SkinVision є корисним додатком для тих, хто прагне регулярно відстежувати стан своєї шкіри і своєчасно виявляти можливі проблеми. Завдяки зручності та швидкості оцінки, він допомагає користувачам отримувати початкову інформацію про ризики, проте не замінює професійного медичного огляду.

KardiaMobile – мобільний застосунок від компанії AliveCor, призначений для моніторингу серцевої діяльності, особливо корисний для людей із ризиком серцевих аритмій. Застосунок використовується разом із компактним пристроєм KardiaMobile, який знімає електрокардіограму і передає дані на смартфон. KardiaMobile дозволяє проводити ЕКГ-сканування в будь-якому місці, зберігати результати, отримувати миттєві попередні висновки та ділитися даними з лікарем.



Рисунок 1.6 – Приклад інтерфейсу системи «KardiaMobile» [24]

Архітектура включає інтерфейс користувача для запуску тестування, перегляду і збереження результатів, а також серверну частину з алгоритмами для обробки і попереднього аналізу даних ЕКГ. Дані передаються за допомогою Bluetooth, що забезпечує зручну інтеграцію з мобільними пристроями та можливість зберігати історію для тривалого моніторингу.

Переваги KardiaMobile:

- мобільність і зручність. Дозволяє користувачам проводити ЕКГ у будь-якому місці, не прив'язуючись до клініки;
- швидка обробка результатів [25]. Забезпечує миттєвий попередній аналіз серцевої діяльності, допомагаючи вчасно виявити можливі аритмії;
- доступ до історії даних. Дозволяє зберігати результати і відстежувати зміни в стані серця з часом;
- інтеграція з медичними консультаціями. Можливість ділитися даними з лікарем для отримання професійної оцінки.

Недоліки KardiaMobile:

- не забезпечує безперервний моніторинг. Пристрій призначений для короткочасних вимірювань, а не для постійного стеження за серцем;

- відсутність повного діагностичного потенціалу [26]. Не замінює комплексного медичного обстеження у кардіолога;
- залежність від правильної інтерпретації. Попередній аналіз може бути неточним без подальшої оцінки лікарем;
- обмеження в функціоналі. Працює тільки з певними мобільними пристроями і вимагає спеціального обладнання (пристрою KardiaMobile).

Отже, KardiaMobile – зручний і ефективний інструмент для моніторингу серцевої діяльності в домашніх умовах. Він дозволяє користувачам своєчасно виявляти можливі порушення ритму, але через обмеження в функціоналі і відсутність безперервного моніторингу не замінює професійного медичного обстеження.

Виходячи з проведеного аналізу, можна зробити висновок, що існуючі програмні рішення для діагностики серцево-судинних захворювань мають значні переваги, проте також характеризуються певними обмеженнями, такими як висока вартість, складність у використанні та впровадженні, а також обмежений доступ до безкоштовних версій. Розробка нового рішення на базі штучного інтелекту з простішим функціоналом та легкістю інтеграції у робочі процеси медичних установ дозволить знизити бар'єри для використання таких технологій широким колом пацієнтів і лікарів. Безкоштовна версія програми забезпечить доступність для закладів з обмеженими ресурсами та користувачів, які потребують регулярного моніторингу, але не мають можливості оплачувати дорогі ліцензії. Таке рішення сприятиме поширенню технологій моніторингу здоров'я на нові ринки, підвищуючи ефективність профілактики серцево-судинних захворювань і знижуючи навантаження на медичну систему.

1.4 Постановка задачі

Завдання розробки інформаційної технології для діагностування стану здоров'я людини полягає у створенні програмного продукту, що дозволить автоматизувати процес прогнозування ризиків серцево-судинних захворювань

на основі обраних користувачем даних. Ця система повинна використовувати сучасні методи машинного навчання для точного аналізу індивідуальних параметрів здоров'я, надаючи персоналізовані прогнози на основі фізіологічних показників та інших факторів ризику.

До ключових вимог для розроблюваної системи належать:

- автоматизація процесу завантаження та обробки даних. Система має підтримувати завантаження даних тренування з обраного користувачем файлу, забезпечуючи зручний інтерфейс для вибору файлів, що спрощує процес підготовки до тренування моделі;
- точність і надійність прогнозування. Використання алгоритмів машинного навчання повинно забезпечити точне прогнозування, дозволяючи системі враховувати багатофакторний аналіз на основі введених користувачем вхідних даних і зберігаючи при цьому стабільність результатів;
- збереження та повторне використання моделей. Після тренування модель повинна зберігатися для подальшого використання, що дасть змогу застосовувати її для наступних прогнозів без необхідності повторного навчання, забезпечуючи стабільний і швидкий доступ до результатів;
- зручність введення даних для прогнозування. Система повинна мати інтуїтивно зрозумілий інтерфейс для введення вхідних даних користувачем, що забезпечить можливість швидкого та простого отримання прогнозу.

Така система дозволить значно підвищити ефективність діагностування ризиків серцево-судинних захворювань, автоматизуючи процес завантаження, тренування, збереження та використання моделей для прогнозування. Рішення стане зручним інструментом для персоналізованого підходу до моніторингу здоров'я, що особливо важливо для пацієнтів з індивідуальними потребами та дозволить оптимізувати використання медичних ресурсів.

1.5 Висновок до розділу 1

У рамках даного розділу було проведено комплексний аналіз предметної області для розробки інформаційної технології прогнозування стану здоров'я людини, що включає огляд сучасних підходів до діагностики, дослідження факторів ризику та оцінку існуючих програмних рішень. Було детально розглянуто класичні методи діагностики, зокрема фізичне обстеження, лабораторні аналізи, УЗД, рентгенографію, КТ, МРТ, ЕКГ та біопсію, що дозволяють отримати базові дані для оцінки здоров'я пацієнтів. Проаналізовано можливості штучного інтелекту та машинного навчання у медичній діагностиці, включаючи методи нейронних мереж, глибокого навчання, класифікації, кластеризації та аналізу аномалій, а також IoT-рішення, що використовують різноманітні сенсори для моніторингу життєвих показників. Досліджено фактори ризику серцево-судинних захворювань, серед яких артеріальний тиск, індекс маси тіла, шкідливі звички, генетична схильність та соціально-економічні умови, які суттєво впливають на здоров'я і які важливо враховувати в прогнозуванні. Проведено огляд існуючих програмних рішень, таких як Ada, SkinVision та KardiaMobile, з виділенням їхніх переваг і недоліків, що дозволило сформулювати вимоги до нового програмного продукту. Визначення задачі розробки дало змогу окреслити ключові вимоги та можливості нової системи, які буде використано у наступному розділі для побудови інформаційної технології, здатної забезпечити автоматизоване, точне та ефективне прогнозування серцево-судинних захворювань на основі обраних користувачем даних.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПРОГНОЗУВАННЯ СТАНУ ЗДОРОВ'Я ЛЮДИНИ

2.1 Аналіз та обґрунтування вибору методу машинного навчання для прогнозування стану здоров'я

Сучасні технології машинного навчання забезпечують високий рівень точності в задачах, пов'язаних із прогнозуванням стану здоров'я, завдяки здатності моделювати складні взаємозв'язки між численними ознаками та виявляти приховані закономірності в медичних даних. Основною метою даного підрозділу є обґрунтування вибору оптимального методу машинного навчання, який найкраще підходить для задачі прогнозування здоров'я людини. Цей вибір залежить від особливостей медичних даних, вимог до точності прогнозів, інтерпретованості результатів, а також обчислювальної складності алгоритму.

Метод градієнтного бустингу дерев рішень (ГБДР) є одним із найпопулярніших і ефективних підходів у машинному навчанні для вирішення завдань класифікації та регресії. Його суть полягає у побудові ансамблю слабких моделей, якими зазвичай виступають дерева рішень, які працюють послідовно, щоб покращити точність прогнозування [27]. Градієнтний бустинг додає кожне нове дерево, яке коригує помилки попередніх дерев, тим самим зменшуючи похибку моделі загалом. На відміну від простого бустингу, цей метод використовує градієнт спаду похибки, що дозволяє кожному наступному дереву ефективніше «виправляти» попередні помилки.

Даний метод широко застосовується для роботи з великими масивами даних, особливо у завданнях, де необхідна висока точність прогнозування. Зокрема, у медицині цей метод можна використовувати для прогнозування стану здоров'я пацієнтів, оцінки ризику розвитку різних захворювань та діагностики на основі історичних даних. Завдяки своїй здатності поступово поліпшувати результати через навчання на помилках попередніх моделей, градієнтний бустинг може враховувати множинні фактори ризику та інші медичні показники,

що впливають на стан здоров'я пацієнта. Наприклад, у задачах прогнозування ризику серцево-судинних захворювань, метод може обробляти дані про артеріальний тиск, рівень холестерину, історію хвороб та інші показники, що дозволяє моделі більш точно оцінити ймовірність виникнення ускладнень.

На рисунку 2.1 показано схему принципу роботи ГБДР. Схема ілюструє процес поступового навчання дерев рішень з урахуванням ваг даних і складання прогнозів в ансамбль для отримання остаточного прогнозу, що є результатом всього ансамблю слабких класифікаторів.

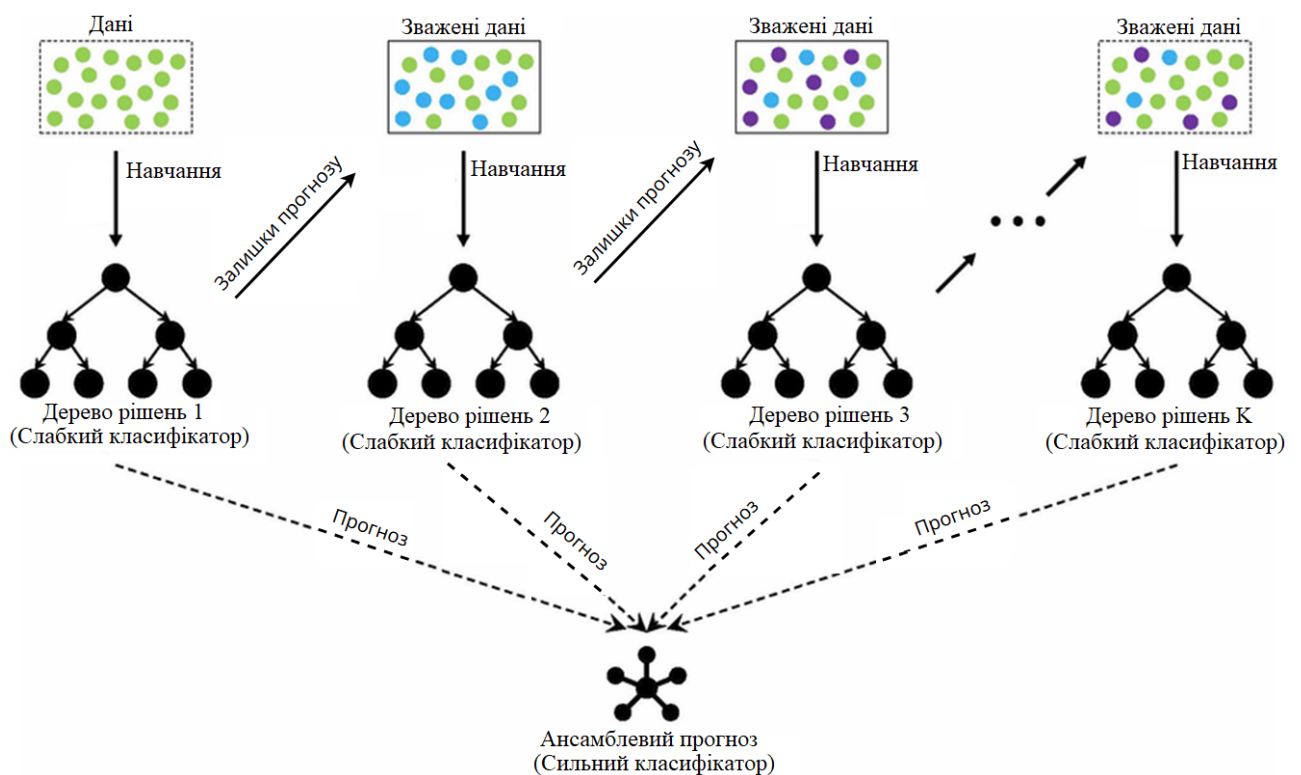


Рисунок 2.1 – Принцип роботи методу градієнтного бустингу дерев рішень [28]

На першому етапі, початкові дані використовуються для тренування першого дерева рішень. Це дерево є слабким класифікатором, тобто воно здатне виконувати певні прогнози, але з обмеженою точністю. Після цього алгоритм оцінює похибку, яка була допущена під час прогнозування, і змінює ваги даних так, щоб у наступному дереві більше уваги приділялося тим прикладам, які були неправильно класифіковані попереднім деревом. Це забезпечує адаптивне навчання, де кожне нове дерево спрямоване на корекцію помилок попереднього.

Другий і третій рівні дерев рішень зображені аналогічно, де кожне дерево тренується на даних з урахуванням змінених ваг. Цей процес повторюється для кожного з дерев до тих пір, поки не буде досягнуто визначеної кількості ітерацій або поки модель не досягне прийнятного рівня точності. На останньому етапі, кожне з дерев вносить свій прогноз у загальний ансамбль, результати якого об'єднуються в єдиний сильний класифікатор, що дає остаточний прогноз.

Переваги методу градієнтного бустингу дерев рішень:

- висока точність. Градієнтний бустинг зазвичай забезпечує кращу точність порівняно з багатьма іншими методами, оскільки він комбінує велику кількість слабких моделей (дерев рішень), що дозволяє мінімізувати помилки;
- гнучкість в обробці даних [29]. Метод добре працює з різними типами даних і здатний обробляти як числові, так і категоріальні дані, що робить його універсальним для різних задач, зокрема медичних;
- можливість обробки складних зв'язків між ознаками [30]. Градієнтний бустинг ефективно знаходить нелінійні взаємозв'язки між ознаками, що дозволяє враховувати різні фактори, які можуть впливати на прогноз, наприклад, медичні показники для оцінки стану здоров'я.

Недоліки методу ГБДР:

- високі обчислювальні витрати [31]. Навчання моделі градієнтного бустингу є досить ресурсомістким, що потребує значних обчислювальних потужностей, особливо для великих обсягів даних;
- ризик перенавчання. Оскільки метод намагається максимально зменшити похибку, він може перенавчитися на тренувальних даних, що погіршує його здатність до узагальнення на нових даних;
- складність у налаштуванні гіперпараметрів. Градієнтний бустинг має велику кількість гіперпараметрів, які впливають на продуктивність моделі, і їх налаштування потребує часу та експертизи, що ускладнює процес моделювання.

Отже, ГБДР є потужним і гнучким інструментом, що може забезпечити високу точність у задачах прогнозування та класифікації, зокрема в медичній сфері для оцінки стану здоров'я пацієнтів. Проте він вимагає значних ресурсів

для навчання та налаштування, а також може страждати від перенавчання. Вибір даного методу доцільний у випадках, коли необхідна висока точність і є можливість використовувати необхідні обчислювальні ресурси.

Метод випадкового лісу (ВЛ) є одним із методів ансамблевого навчання, який базується на використанні множини дерев рішень. Ідея полягає в тому, щоб побудувати велику кількість дерев рішень на основі різних випадкових підмножин даних і ознак, а потім об'єднати результати кожного дерева, використовуючи голосування для задач класифікації або усереднення для задач регресії [32]. Цей метод є надзвичайно потужним, стійким до перенавчання та здатним обробляти великий обсяг даних із різними типами ознак.

ВЛ активно використовується у різних сферах, таких як фінанси для аналізу ризиків, маркетинг для сегментації клієнтів, і, зокрема, у медицині для прогнозування стану здоров'я та діагностики захворювань. У медичних додатках ВЛ може бути корисним для класифікації пацієнтів за групами ризику, прогнозування результатів лікування та оцінки ймовірності розвитку певних захворювань. Завдяки своїй здатності ефективно працювати з великою кількістю ознак і обробляти пропущені значення, випадковий ліс може допомогти виявляти складні закономірності в медичних даних, наприклад, аналізуючи такі показники, як рівень цукру в крові, тиск, вікові параметри, і використовувати ці дані для складання прогнозу щодо стану здоров'я пацієнта.

На рисунку 2.2 наведено схему принципу роботи випадкового лісу. Схема показує, як кожне дерево рішень у лісі приймає рішення на основі одного і того ж зразка даних. Кожне дерево, навчене на випадковій вибірці даних, класифікує зразок у певний клас.

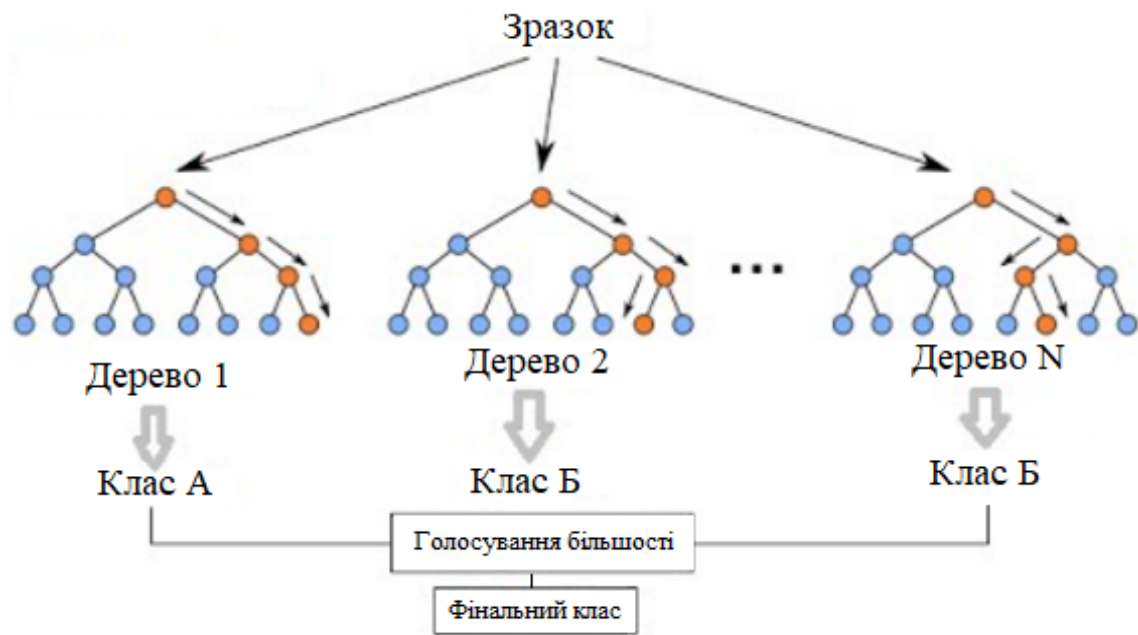


Рисунок 2.2 – Принцип роботи методу випадкового лісу [33]

На схемі показано, що перше дерево класифікує зразок у клас А, тоді як друге та третє дерева (разом із наступними деревами) класифікують його як клас Б. Після цього відбувається етап голосування, де обирається клас, який був визначений більшістю дерев. У результаті клас, за який проголосувала більшість дерев, стає фінальним класом для цього зразка. Цей підхід дозволяє знизити ймовірність помилок і підвищити точність прогнозування, оскільки випадковий ліс, на відміну від окремих дерев, є більш стійким до похибок та перенавчання.

Переваги методу випадкового лісу:

- стійкість до перенавчання. Завдяки використанню багатьох дерев, кожне з яких навчається на різних випадкових вибірках, випадковий ліс менш схильний до перенавчання, що робить його більш стійким та надійним у прогнозуванні;
- гнучкість до різних типів даних [34]. Метод може працювати з числовими, категоріальними даними та навіть із даними з пропущеними значеннями, що робить його універсальним для різноманітних задач;
- інтерпретація важливості ознак. Випадковий ліс дозволяє оцінити важливість ознак, використаних для класифікації або регресії, що особливо

корисно в медичних задачах, де важливо розуміти, які показники найбільше впливають на прогноз.

Недоліки методу випадкового лісу:

- високі обчислювальні витрати [35]. Метод вимагає значних обчислювальних ресурсів, особливо при роботі з великими наборами даних та великою кількістю дерев, що може сповільнювати обробку;

- складність у налаштуванні [36]. Випадковий ліс має низку гіперпараметрів, таких як кількість дерев і глибина кожного дерева, які потребують налаштування для досягнення оптимальної продуктивності, що може бути складним і вимагати багато часу;

Отже, метод ВЛ є потужним інструментом для задач класифікації та регресії, особливо у випадках, де важлива стійкість до перенавчання і потрібна робота з різноманітними типами даних. Завдяки можливості визначати важливість ознак, випадковий ліс знаходить застосування в медицині, де потрібно розуміти вплив окремих показників на стан здоров'я. Однак висока обчислювальна складність і потреба в ретельному налаштуванні гіперпараметрів можуть обмежувати його застосування в деяких випадках.

Метод дерева рішень (ДР) є одним із найпоширеніших підходів у машинному навчанні, який використовується для розв'язання задач класифікації та регресії. Дерева рішень базуються на розбитті даних на підмножини, що зменшує невизначеність і дозволяє дійти до кінцевого рішення, яке представлено у вигляді листового вузла [37]. Цей метод особливо корисний тим, що він інтуїтивно зрозумілий, легко інтерпретується і не потребує попередньої нормалізації даних.

ДР застосовуються у багатьох сферах, зокрема у фінансовій аналітиці, маркетингу, бізнес-аналізі, а також у медицині для діагностики та прогнозування стану здоров'я. У медичній сфері дерева рішень можуть використовуватися для прогнозування розвитку захворювань або визначення груп ризику. Наприклад, за наявності даних про пацієнта, таких як вік, рівень артеріального тиску, рівень холестерину, історія хвороб, дерево рішень може допомогти спрогнозувати

ймовірність виникнення певного захворювання. Кожна характеристика пацієнта може розглядатися як ознака, яка поступово звужує можливості діагнозу та веде до кінцевого рішення.

На рисунку 2.3 показано схему принципу роботи дерева рішень. Процес починається з кореневого вузла, де розташовується початковий зразок. У кореневому вузлі відбувається перевірка певної ознаки, яка визначає, в який напрямок рухатиметься зразок. Залежно від значення ознаки, зразок переходить до наступного вузла, який може бути або вузлом рішення, або листовим вузлом.

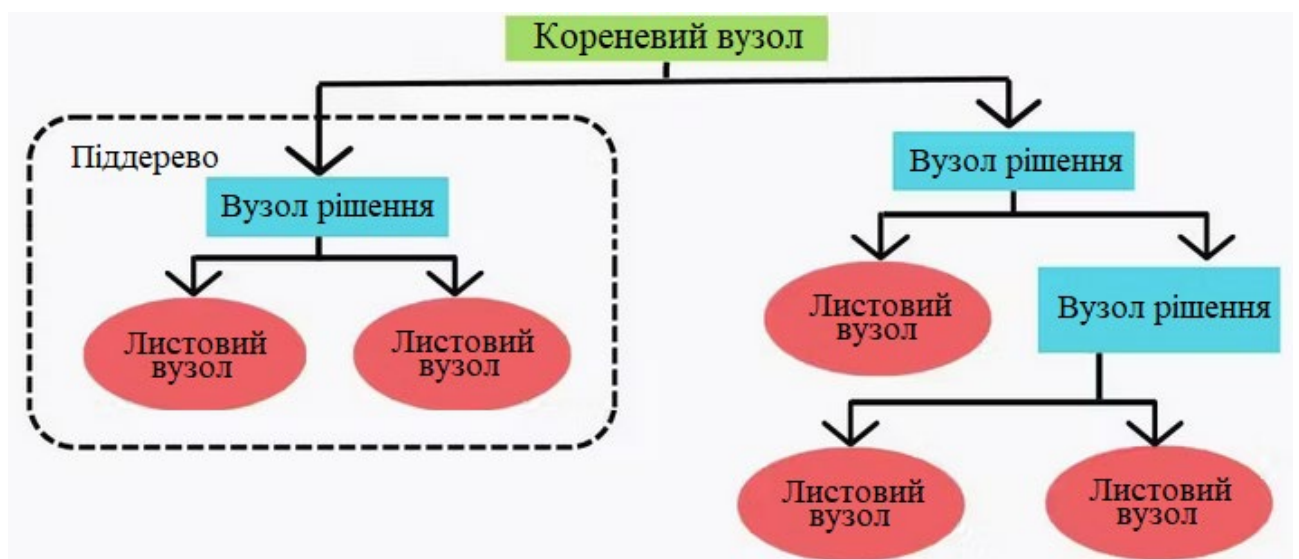


Рисунок 2.3 – Принцип роботи методу дерева рішень [38]

Вузол рішення продовжує процес розгалуження, тобто поділяє зразок далі на основі іншої ознаки, ведучи до додаткових вузлів або листових вузлів. Цей процес триває до тих пір, поки зразок не досягне листового вузла, який є кінцевим результатом аналізу дерева для даного зразка.

На схемі представлено два піддерева, що включають вузли рішень і листові вузли. У лівому піддереві від кореневого вузла через один вузол рішення зразок досягає одного з двох листових вузлів. У правому піддереві структура дещо складніша: від кореневого вузла зразок проходить через два вузли рішення, перш ніж досягти одного з кількох листових вузлів. Кожен листовий вузол відповідає за певний клас або значення, яке є остаточним результатом для цього зразка.

Переваги методу ДР:

- інтуїтивно зрозуміла інтерпретація. Дерева рішень легко візуалізувати та зрозуміти, оскільки структура дерева чітко показує процес прийняття рішень на кожному етапі;
- мінімальна підготовка даних [39]. На відміну від інших алгоритмів, дерево рішень не вимагає нормалізації даних або перетворення їх у певний формат. Воно може працювати з категоріальними і числовими даними, що спрощує процес попередньої обробки;

Недоліки методу дерева рішень:

- схильність до перенавчання [40]. Дерева рішень можуть легко перенавчитися на тренувальних даних, особливо якщо дерево є занадто глибоким;
- нестабільність [41]. Дерева рішень чутливі до змін у даних. Невеликі зміни у вхідних даних можуть значно змінити структуру дерева, що знижує стабільність методу;
- неоптимальність для складних зв'язків. Дерево рішень не завжди здатне ефективно моделювати складні нелінійні зв'язки між ознаками.

Отже, метод ДР є ефективним і зрозумілим інструментом для розв'язання задач класифікації та регресії, особливо у випадках, коли важлива інтерпретація результату, як-от у медичній сфері для пояснення прогнозів стану здоров'я. Однак його обмеження, зокрема схильність до перенавчання та нестабільність, можуть знижувати його ефективність у більш складних задачах. У таких випадках варто розглянути ансамблеві методи, які можуть підвищити точність і стабільність прогнозів.

Щоб краще зрозуміти особливості кожного з методів, доцільно провести їх порівняння за основними характеристиками. Кожен з них має свої унікальні властивості, які впливають на точність, стабільність, стійкість до перенавчання та вимоги до обчислювальних ресурсів. У таблиці 2.1 наведено порівняння цих методів за характеристиками, що найчастіше визначають їх застосування на практиці.

Таблиця 2.1 – Порівняння характеристик методів машинного навчання

Характеристика	Гرادієнтний бустинг дерев рішень	Випадковий ліс	Дерева рішень
Точність	Висока	Середня - висока	Середня
Стійкість до перенавчання	Висока (при належних налаштуваннях)	Висока	Низька
Вимоги до обчислювальних ресурсів	Високі	Середні	Низькі
Інтерпретація результатів	Помірна	Помірна - висока	Висока
Складність у налаштуванні	Висока	Середня	Низька
Стійкість до змін у даних	Висока	Висока	Низька
Можливість обробки складних взаємозв'язків	Висока	Середня	Низька

Метод ГБДР є оптимальним вибором для розробки системи через його високу точність та здатність ефективно обробляти складні взаємозв'язки між ознаками. Завдяки стійкості до перенавчання, він забезпечує надійність результатів навіть у випадках із великою кількістю ознак і взаємозалежними факторами, що особливо важливо в медичних застосуваннях, де точність прогнозу має критичне значення. Незважаючи на високі вимоги до обчислювальних ресурсів, цей метод здатен гарантувати стабільні результати, що робить його вигідним вибором для задач, де якість результату важливіша за швидкість обробки. ГБДР також демонструє високу стійкість до змін у даних, що є додатковою перевагою для використання в умовах реальних даних, де можуть спостерігатися різноманітні варіації.

2.2 Опис та обґрунтування вибору необхідного тренувального набору даних

Для розробки системи прогнозування ризику серцево-судинних захворювань був обраний набір даних Heart Disease Health Indicators Dataset із платформи Kaggle [42]. Це обґрунтовано тим, що серцево-судинні захворювання є провідною причиною смертності у Сполучених Штатах, щороку забираючи близько 647 000 життів. Набір даних містить інформацію про найбільш поширені фактори ризику серцево-судинних захворювань, такі як високий артеріальний тиск, високий рівень холестерину, діабет, паління та інші соціально-демографічні й поведінкові характеристики. Ці фактори є критично важливими для побудови точних моделей прогнозування, адже більшість пацієнтів дізнаються про наявність захворювання вже після виникнення симптомів, що підкреслює важливість превентивних заходів і точних оцінок ризику.

Даний набір даних базується на щорічному телефонному опитуванні Behavioral Risk Factor Surveillance System (BRFSS), яке проводить Центр з контролю та профілактики захворювань США. Опитування охоплює понад 400 000 американців і містить дані про поведінкові ризики, хронічні захворювання та використання медичних послуг. Даний набір даних є підготовленою версією BRFSS за 2015 рік і включає 253 680 респондентів, що дозволяє зменшити проблему дисбалансу класів (всього 23 893 респонденти мають серцево-судинні захворювання, тоді як 229 787 не мають). Завдяки великому обсягу даних, цей набір є підходящим для застосування в алгоритмах машинного навчання, які потребують багато даних для підвищення точності прогнозів.

У таблиці 2.2 наведено детальний опис ключових атрибутів набору даних Heart Disease Health Indicators Dataset, які використовуються для моделювання та аналізу ризиків серцево-судинних захворювань. Ці атрибути включають як медичні показники, що мають безпосередній зв'язок із захворюваннями серця,

так і поведінкові, соціально-демографічні характеристики, які допомагають виявити додаткові ризики та чинники, що впливають на загальний стан здоров'я пацієнтів.

Таблиця 2.2 – Опис ключових атрибутів набору даних

Атрибут	Опис	Значення
1	2	3
HeartDiseaseorAttack	Наявність захворювання серця або серцевого нападу	1 - наявність, 0 - відсутність
HighBP	Високий артеріальний тиск	1 - так, 0 - ні
HighChol	Високий рівень холестерину	1 - так, 0 - ні
CholCheck	Перевірка рівня холестерину протягом останніх 5 років	1 - так, 0 - ні
BMI	Індекс маси тіла, який показує співвідношення ваги до зросту	Числове значення
Smoker	Куріння	1 - так, 0 - ні
Stroke	Інсульт	1 - так, 0 - ні
Diabetes	Діабет	0 - відсутній, 1 - на ранній стадії, 2 - розвинутий
PhysActivity	Фізична активність за останні 30 днів	1 - так, 0 - ні
Fruits	Споживання фруктів щодня	1 - так, 0 - ні
Veggies	Споживання овочів щодня	1 - так, 0 - ні
HvyAlcoholConsump	Вживання великої кількості алкоголю	1 - так, 0 - ні
AnyHealthcare	Наявність медичного страхування	1 - так, 0 - ні
NoDocbcCost	Відмова від візиту до лікаря через вартість послуг	1 - так, 0 - ні
GenHlth	Загальний стан здоров'я	1 - відмінний, 2 - добрий, 3 - задовільний, 4 - посередній, 5 - поганий
MentHlth	Кількість днів з поганим психічним здоров'ям за останній місяць	Числове значення
PhysHlth	Кількість днів з поганим фізичним здоров'ям за останній місяць	Числове значення

Продовження таблиці 2.2

1	2	3
DiffWalk	Проблеми з ходьбою	1 - так, 0 - ні
Sex	Стать	0 - жіноча, 1 - чоловіча
Age	Вікова група	1 - 18-24, 2 - 25-29, ..., 13 - 80+
Education	Рівень освіти	1 - не завершено середню, 2 - середня, ..., 6 - вища
Income	Рівень доходу	1 - < \$10,000, 2 - \$10,000–\$15,000, ..., 8 - > \$75,000

Вибір цього набору даних обумовлений його широким охопленням важливих факторів ризику, які є вирішальними при прогнозуванні серцево-судинних захворювань. Наявність таких атрибутів, як високий тиск, холестерин, рівень фізичної активності, діабет та інші, дозволяє створити точну модель для оцінки ризику захворювань. Більше того, великий обсяг даних дає можливість побудувати стійку модель на основі ГБДР, що забезпечить надійний прогноз і допоможе в превентивній медицині. Цей набір даних також містить показники поведінки та соціально-демографічні характеристики, що дозволяє здійснити більш детальний і персоналізований аналіз ризику.

2.3 Математична модель прогнозування та класифікації ризиків

Діагностування ризику серцевих захворювань є критично важливим завданням у сучасній медицині. Застосування методів машинного навчання, зокрема градієнтного бустингу дерев рішень, дозволяє створити ефективні моделі прогнозування на основі великої кількості клінічних та демографічних ознак. Нижче описано математичну модель для прогнозування ризику серцевих захворювань, використовуючи алгоритм градієнтного бустингу дерев рішень.

Нехай задано вибірку даних із N спостережень:

$$\{(x_i, y_i)\}_{i=1}^N, \quad (2.1)$$

де $x_i \in R$ – вектор ознак i -го спостереження, а $y_i \in \{0,1\}$ – мітка класу, де 1 відповідає наявності серцевого захворювання, а 0 – його відсутності.

Вектор ознак x_i складається з наступних атрибутів:

- високий артеріальний тиск (x_{i1});
- високий рівень холестерину (x_{i2});
- перевірка рівня холестерину (x_{i3});
- індекс маси тіла (x_{i4});
- куріння (x_{i5});
- інсульт (x_{i6});
- діабет (x_{i7});
- фізична активність (x_{i8});
- споживання фруктів (x_{i9});
- споживання овочів (x_{i10});
- вживання алкоголю (x_{i11});
- наявність медичного страхування (x_{i12});
- відмова від візиту до лікаря через вартість (x_{i13});
- загальний стан здоров'я (x_{i14});
- погане психічне здоров'я (x_{i15});
- погане фізичне здоров'я (x_{i16});
- проблеми з ходьбою (x_{i17});
- стать (x_{i18});
- вікова група (x_{i19});
- рівень освіти (x_{i20});
- рівень доходу (x_{i21}).

Потрібно побудувати модель $f(x)$, яка прогнозує ймовірність наявності серцевого захворювання для заданого вектору ознак x :

$$\hat{y} = f(x) , \quad (2.2)$$

У разі наявності пропущених значень у ознаках, вони замінюються середнім значенням відповідної ознаки:

$$x_{ij} = \begin{cases} x_{ij}, & \text{якщо дані доступні} \\ \bar{x}_j, & \text{якщо дані відсутні} \end{cases} , \quad (2.3)$$

де $\bar{x}_j$ – середнє значення ознаки j -го стовпця по всій вибірці.

Для забезпечення однакового масштабу ознак здійснюється нормалізація методом середнього значення та дисперсії:

$$x_{ij}^{norm} = \frac{x_{ij} - \bar{x}_j}{S_i} , \quad (2.4)$$

де S_i – стандартне відхилення ознаки j .

Гradientний бустинг дерев рішень будує модель як суму M базових моделей (дерев рішень):

$$f(x) = \sum_{m=1}^M \gamma_m \cdot h_m(x) , \quad (2.5)$$

де:

$h_m(x)$ – базова модель (дерево рішень) на m -й ітерації;

γ_m – коефіцієнт навчання або вага m -ї базової моделі.

Використовується логістична функція втрат, яка підходить для задачі бінарної класифікації:

$$L(y, f(x)) = \ln(1 + e^{-yf(x)}), \quad (2.5)$$

де $y \in \{-1, 1\}$. Для цього вихідні мітки перетворюються: якщо $y_i = 0$, то $y'_i = -1$, якщо $y_i = 1$, то $y'_i = 1$.

Модель ініціалізується константою, яка мінімізує функцію втрат:

$$f_0(x) = \arg \min_{\rho} \sum_{i=1}^N L(y'_i, \rho), \quad (2.6)$$

Для $m=1$ до M виконується обчислення негативних градієнтів:

$$r_{im} = - \left(\frac{\partial L(y'_i, f_{m-1}(x_i))}{\partial f_{m-1}(x_i)} \right) = y'_i - \hat{y}_i, \quad (2.7)$$

де $\hat{y}_i = \sigma f_{m-1}(x_i)$, а $\sigma(z) = \frac{1}{1+e^{-z}}$ – сигмоїдна функція.

Далі навчається дерево $h_m(x)$ для прогнозування r_{im} .

Для цього знаходиться оптимальний коефіцієнт γ_m шляхом мінімізації:

$$\gamma_m = \arg \min_{\gamma} \sum_{i=1}^N L(y'_i, f_{m-1}(x_i) + \gamma h_m(x_i)), \quad (2.8)$$

Модель у свою чергу оновлюється за формулою:

$$f_m(x) = f_{m-1}(x) + \gamma_m \cdot h_m(x), \quad (2.9)$$

Після навчання повної моделі $f_m(x)$, для нового спостереження x обчислюється ймовірність наявності захворювання:

$$P(y = 1|x) = \sigma(f_M(x)) = \frac{1}{1 + e^{-f_M(x)}}, \quad (2.10)$$

Якщо $(y = 1|x) \geq 0.5$, то прогнозується наявність ризику серцевого захворювання.

Побудована математична модель на основі градієнтного бустингу дерев рішень дозволяє ефективно прогнозувати ризик серцевих захворювань, використовуючи різноманітні клінічні та демографічні ознаки. Застосування цієї моделі може сприяти своєчасному виявленню пацієнтів з підвищеним ризиком та прийняттю відповідних профілактичних заходів. Це, в свою чергу, може знизити рівень смертності від серцево-судинних захворювань та покращити якість життя пацієнтів.

2.4 Опис метрик для оцінки якості прогнозування та точності моделі

Для об'єктивної оцінки якості побудованої моделі та визначення її здатності до точного прогнозування використовуються кілька основних метрик. Вибір цих метрик обумовлений необхідністю врахування різних аспектів роботи моделі, таких як загальна точність, здатність правильно класифікувати позитивні та негативні класи, а також баланс між точністю та повнотою в умовах класового дисбалансу. Нижче описані метрики є стандартними у машинному навчанні для задач класифікації, що дозволяє не лише оцінити якість прогнозування, але й порівнювати модель з іншими методами або моделями.

Точність є однією з найбільш базових метрик, яка визначає частку правильних прогнозів серед загальної кількості спостережень. Точність обчислюється за формулою:

$$Accuracy = \frac{1}{N} \sum_{i=1}^N I(y_i = \hat{y}_i), \quad (2.11)$$

де I – індикаторна функція, яка дорівнює 1, якщо прогноз моделі $\hat{y}_i$ збігається з реальним значенням y_i , і 0, якщо не збігається. Ця метрика показує загальну ефективність моделі, але не завжди є достатньо інформативною в умовах класового дисбалансу, оскільки вона не враховує, наскільки добре модель розпізнає конкретні класи.

$F1$ -міра є гармонійним середнім між точністю ($Precision$) та повнотою ($Recall$), і використовується для оцінки балансу між цими показниками. $F1$ -міра особливо корисна в умовах, коли важливо уникати як хибних позитивів, так і хибних негативів. Формула для розрахунку $F1$ -міри виглядає так:

$$F1 = \frac{Precision \times Recall}{Precision + Recall}, \quad (2.12)$$

Точність ($Precision$) обчислюється за формулою:

$$Precision = \frac{TP}{TP + FP}, \quad (2.13)$$

де TP – це кількість випадків, коли модель правильно визначила позитивний клас, а FP – кількість випадків, коли модель помилково класифікувала негативний клас як позитивний.

Повнота ($Recall$) обчислюється за формулою:

$$Recall = \frac{TP}{TP + FN}, \quad (2.14)$$

де FN – це кількість випадків, коли модель помилково класифікувала позитивний клас як негативний.

Можна сказати, що $F1$ -міра є корисною метрикою, коли потрібно враховувати як точність, так і повноту, що дозволяє моделі досягти балансу між

пропущенням важливих позитивів та запобіганням надмірній кількості хибних позитивів.

Площа під *ROC*-кривою (*AUC*) є метрикою, яка показує здатність моделі розрізняти позитивний і негативний класи на різних порогах. *ROC*-крива обчислюється за формулою:

$$AUC = \int_0^1 TPR(FPR^{-1}(t))dt, \quad (2.15)$$

де *TPR* – True Positive Rate (справжній позитивний рівень), який визначає відсоток коректно класифікованих позитивних випадків, а *FPR* – False Positive Rate (хибний позитивний рівень), який показує частку неправильно класифікованих негативних випадків. Ця інтегральна площа під кривою дозволяє оцінити середню здатність моделі розрізняти між класами при зміні порогу класифікації. *AUC* наближається до 1 для ідеальних моделей і до 0.5 для моделей, що класифікують випадковим чином.

Кожна із розглянутих метрик відіграє важливу роль в оцінці якості моделі. Точність є базовою метрикою, яка показує загальний рівень правильності класифікації, проте її ефективність знижується в умовах класового дисбалансу. *F1*-міра забезпечує більш точне уявлення про продуктивність моделі в умовах, де важливо враховувати як точність, так і повноту, що особливо актуально в задачах із високою значущістю позитивного класу, наприклад, у медичній діагностиці. *AUC* дозволяє оцінити здатність моделі правильно розрізняти класи незалежно від обраного порогу, що є цінним для задач, де важлива точність у широкому діапазоні можливих порогів. Разом ці метрики надають комплексне розуміння ефективності моделі, дозволяючи врахувати різні аспекти класифікації та вибрати найбільш відповідний підхід для конкретної задачі.

2.5 Висновок до розділу 2

У рамках даного розділу було здійснено повний аналіз та обґрунтування вибору методів і даних для побудови системи прогнозування стану здоров'я людини. Проведено порівняння методів машинного навчання, таких як градієнтний бустинг дерев рішень, випадковий ліс та дерева рішень. На основі аналізу було обрано метод градієнтного бустингу дерев рішень завдяки його високій точності та здатності ефективно обробляти складні взаємозв'язки між ознаками, що є важливим для завдань медичної діагностики. Для тренування моделі було обрано датасет з платформи Kaggle, що містить критично важливі показники здоров'я для прогнозування серцево-судинних захворювань, такі як високий тиск, рівень холестерину, фізична активність, діабет та інші фактори ризику. Окрім цього, математично описано обраний метод градієнтного бустингу дерев рішень, що забезпечує основу для його коректного застосування в алгоритмі моделі. Також описано метрики для оцінки якості моделі, зокрема точність, F1-міру та площу під ROC-кривою, які дозволять об'єктивно оцінити ефективність моделі в наступному етапі роботи. Отримані результати даного розділу закладають основу для реалізації та тестування інформаційної технології діагностики стану здоров'я у наступному розділі.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДІАГНОСТИКИ СТАНУ ЗДОРОВ'Я

3.1 Обґрунтування вибору мови програмування та технологічних засобів розробки

Для створення ефективної та надійної системи для діагностування стану здоров'я людини важливим аспектом є вибір мови програмування та технологічних засобів, що дозволять забезпечити високу продуктивність, легкість підтримки та можливість інтеграції з іншими системами. Серед багатьох сучасних мов програмування Python, Java та C# виділяються своїми унікальними можливостями та функціональністю, що робить їх потенційно придатними для реалізації подібної системи.

Python – це інтерпретована мова з простою та зрозумілою синтаксичною структурою, яка робить її ідеальною для швидкого прототипування та наукових досліджень [43]. Вона забезпечує широкий доступ до бібліотек для обробки даних і машинного навчання, таких як TensorFlow та Keras, що є ключовим для систем, які виконують аналіз даних у сфері медицини. Крім того, Python дозволяє легко інтегруватися з іншими інструментами, що значно розширює її можливості для реалізації складних аналітичних рішень.

Java, зі своєї сторони, є універсальною мовою програмування з високим рівнем надійності та безпеки, яка має широку підтримку на різних платформах [44]. Завдяки об'єктно-орієнтованому підходу та високій продуктивності Java є придатною для реалізації масштабованих і безпечних систем, які часто застосовуються в корпоративному та медичному середовищах.

C# також є потужною мовою програмування, яка тісно інтегрована з платформою .NET, забезпечуючи високу продуктивність і підтримку сучасних функцій для розробки десктопних і веб-додатків [45]. C# надає розробникам потужні засоби для обробки даних і паралельних обчислень, що є цінним для систем, які потребують великої обчислювальної потужності, таких як медичні

діагностичні системи. Окрім того, підтримка сучасних технологій та інструментів Microsoft робить C# популярним вибором для реалізації безпечних та інтегрованих рішень у сфері охорони здоров'я.

Аналізуючи кожну з мов за такими параметрами, як продуктивність, підтримка багатопоточності, інтеграція з інструментами, рівень безпеки та зручність розробки, можна сформулювати повніше уявлення про їхні можливості в контексті розробки медичних додатків. У таблиці 3.1 наведено порівняльний аналіз основних характеристик Python, Java та C#, що є важливими при виборі мови для реалізації інформаційної системи діагностування стану здоров'я.

Таблиця 3.1 – Порівняння можливостей мов програмування

Характеристика	Python	Java	C#
Продуктивність	Середня, залежить від бібліотек	Висока, стабільна	Висока, оптимізована на .NET
Підтримка багатопоточності	Обмежена, GIL ускладнює паралелізм	Широка підтримка, добре масштабована	Широка підтримка на .NET, ефективна
Інтеграція з інструментами	Широка, особливо в наукових середовищах	Підтримка на різних платформах	Глибока інтеграція з інструментами Microsoft
Рівень безпеки	Відносно низький, потребує додаткових бібліотек	Високий, підтримка різних рівнів безпеки	Високий, .NET підтримує сучасні механізми захисту
Зручність розробки	Висока, завдяки простому синтаксису	Помірна, потребує детальнішого коду	Висока, надає інструменти для швидкої розробки

Вибір мови програмування C# для розробки системи діагностики стану здоров'я є обґрунтованим завдяки її високій продуктивності та ефективній багатопоточній обробці на платформі .NET, що є критичними для обчислювально-інтенсивних медичних додатків. Глибока інтеграція з інструментами Microsoft сприяє створенню комплексних та безпечних рішень, що відповідають сучасним вимогам до медичного програмного забезпечення.

Після вибору мови програмування важливо обрати оптимальне середовище розробки, яке забезпечить зручність, ефективність та продуктивність у створенні медичної інформаційної системи. Серед можливих середовищ для розробки на мові C# виділяються Visual Studio 2022, Visual Studio Code та Rider, кожне з яких має свої унікальні можливості та функціональні переваги.

Visual Studio 2022 є потужним і комплексним середовищем розробки, що надає велику кількість інструментів для C# і глибоко інтегрується з платформою .NET [46]. Це середовище підтримує широкі можливості для налагодження, тестування та автоматизації, що полегшує роботу над великими проектами. Visual Studio 2022 також має інтелектуальне автозаповнення та розширену підтримку багатоплатформної розробки, що особливо корисно при створенні масштабованих медичних систем.

Visual Studio Code є легким і гнучким редактором, що підходить для швидкого редагування та розробки кросплатформних додатків [47].

Rider, розроблений компанією JetBrains, є інноваційним середовищем розробки з широким набором інструментів для роботи з C# і .NET, що також включає інтелектуальні функції рефакторингу коду. Rider відомий своєю високою швидкістю і плавною інтеграцією з продуктами JetBrains, такими як ReSharper, що дозволяє підвищити ефективність кодування. Це середовище є чудовим вибором для розробників, яким потрібна гнучкість і продуктивність при роботі над комплексними проектами.

У таблиці 3.2 представлено порівняльний аналіз, який включає такі критерії, як підтримка інструментів налагодження, можливості автоматизації тестування, інтеграція з платформою .NET, зручність для роботи з великими проектами та доступність додаткових засобів для роботи з C#.

Таблиця 3.2 – Порівняння можливостей середовищ розробки

Характеристика	Visual Studio 2022	Visual Studio Code	Rider
Підтримка інструментів налагодження	Повна, включає розширене налагодження	Обмежена, залежить від встановлених розширень	Добре розвинена, але менш комплексна
Автоматизація тестування	Вбудована, включає спеціалізовані інструменти	Потребує встановлення додаткових модулів	Підтримка на середньому рівні
Інтеграція з платформою .NET	Глибока інтеграція, оптимізована	Обмежена, через розширення	Підтримка .NET, інтеграція з JetBrains
Зручність для великих проектів	Висока, оптимізовано для масштабних проектів	Помірна, підходить для менших проектів	Добра, але обмежена для дуже великих проектів
Додаткові засоби для роботи з C#	Розширена підтримка інструментів і бібліотек	Залежить від розширень	Підтримка через JetBrains інструменти

Вибір Visual Studio 2022 для розробки системи діагностування стану здоров'я є обґрунтованим через її потужний набір вбудованих інструментів, що оптимально підтримують платформу .NET та мову програмування C#. Це середовище надає розширені можливості для налагодження і автоматизації тестування, що є критичним для створення стабільних і безпечних медичних додатків. Visual Studio 2022 також забезпечує високу продуктивність і зручність при роботі з великими проектами, полегшуючи управління складними структурами коду. Завдяки широкій інтеграції з різноманітними бібліотеками та інструментами для C#, середовище дозволяє швидко та ефективно реалізовувати необхідні функції для аналізу і обробки медичних даних, що є ключовим аспектом у системах діагностування.

3.2 Реалізація програмного модуля для прогнозування ризику серцевих захворювань з використанням алгоритмів машинного навчання

Для створення системи прогнозування ризику серцевих захворювань важливим компонентом є модуль машинного навчання, який здатен обробляти та аналізувати дані, навчаючи моделі для точного оцінювання ризиків. Цей модуль має забезпечувати зручне завантаження даних, тренування алгоритмів та подальше збереження моделей для майбутніх прогнозів. Реалізація такого модуля потребує інтеграції з інтерфейсом, що дозволить легко додавати нові дані для навчання, вибирати моделі та налаштовувати процес прогнозування, а також забезпечить високу точність і надійність у прогнозах, що є критично важливим у медичних системах. На рис. 3.1 представлено фрагмент коду, який відповідає за функцію вибору файлу з даними для тренування.

```
private void OpenBtn_Click(object sender, EventArgs e) {
    // Створення діалогового вікна для відкриття файлу
    OpenFileDialog openFileDialog = new OpenFileDialog();
    // Налаштування властивостей діалогового вікна
    openFileDialog.Filter = "Text files (*.csv)|*.csv|All files (*.*)|*.*";
    openFileDialog.FilterIndex = 2;
    openFileDialog.RestoreDirectory = true;
    // Відображення діалогового вікна та обробка результату
    if (openFileDialog.ShowDialog() == DialogResult.OK) {
        _Path = openFileDialog.FileName;
        FileNameTextBox.Text = openFileDialog.FileName;
    }
}
```

Рисунок 3.1 – Код вибору файлу з даними для тренування моделі

Код активується натисканням кнопки і відкриває діалогове вікно типу OpenFileDialog, що надає можливість обрати файл з набором даних для подальшого навчання моделі. Налаштовується фільтр для файлів, щоб відобразити користувачеві лише найбільш релевантні формати (зокрема, CSV), що спрощує вибір відповідного файлу для завантаження в систему. Після підтвердження вибору шлях до файлу зберігається у змінній _Path, а текстове поле інтерфейсу оновлюється, відображаючи ім'я обраного файлу, що дає користувачеві впевненість у правильному виборі даних для тренування моделі.

На початковому етапі проводиться налаштування модуля для тренування моделей машинного навчання створюється об'єкт `MLContext`, який виступає основним інструментом для взаємодії з бібліотекою `ML.NET` (рис. 3.2).

```
// Створення контексту ML  
mlContext = new MLContext(seed: 0);
```

Рисунок 3.2 – Створення контексту `ML.NET` для тренування моделі

Даний контекст ініціалізується з використанням фіксованого значення параметра `seed`, що задає початкове значення для генератора випадкових чисел. Встановлення цього параметра сприяє повторюваності експериментів та результатів моделі, що є важливим аспектом для забезпечення стабільності та надійності у процесі навчання та тестування моделі прогнозування ризику.

На етапі завантаження даних для тренування моделі використовується метод `LoadFromTextFile`, що є частиною контексту `mlContext` і дозволяє імпортувати дані з текстового файлу у форматі `CSV` (рис. 3.3).

```
dataView = mlContext.Data.LoadFromTextFile<HeartData>(  
    path: _Path,  
    hasHeader: true,  
    separatorChar: ',');
```

Рисунок 3.3 – Завантаження даних для тренування моделі

Параметр `path` вказує шлях до файлу, що містить дані, необхідні для навчання моделі, а також визначає роздільник у вигляді коми, характерний для `CSV`-формату.

Для підготовки даних до навчання та оцінки моделі виконується розділення набору даних на тренувальну і тестову вибірки за допомогою методу `TrainTestSplit` (рис. 3.4). Зазначений параметр `testFraction`, значення якого дорівнює 0.2, визначає, що 20% даних буде виділено для тестування, тоді як решта 80% буде використана для тренування моделі.

```
var splitData = mlContext.Data.TrainTestSplit(dataView, testFraction: 0.2);
```

Рисунок 3.4 – Розділення набору даних

Цей підхід сприяє об'єктивному оцінюванню точності моделі, оскільки модель спочатку навчається на основній частині даних, а потім перевіряється на окремій тестовій вибірці, що дозволяє уникнути надмірного узагальнення та отримати більш стабільні результати прогнозування.

Метод «PrintClassMetrics» виконує аналіз і виведення кількості зразків, що належать до кожного класу в наборі даних, для оцінки розподілу даних за ризиком серцевих захворювань (рис. 3.5).

```
public void PrintClassMetrics(MLContext mlContext, IDataView data) {
    ReportTextBox.Text +=("\r\nПідрахунок кількості зразків для кожного класу...");
    var dataEnumerable =
        mlContext.Data.CreateEnumerable<HeartData>(data, reuseRowObject: false);
    var classCounts = dataEnumerable
        .GroupBy(d => d.HeartDiseaseorAttack)
        .Select(group => new {
            Class = group.Key,
            Count = group.Count()
        });
    foreach (var item in classCounts) {
        ReportTextBox.Text +=($" \r\nКлас {(item.Class == 1 ?
            "Є ризик" : "Немає ризику")}: {item.Count} зразків");
    }
}
```

Рисунок 3.5 – Код методу «PrintClassMetrics»

Спершу виконується перетворення IDataView у перелік об'єктів класу HeartData, що дозволяє зручно працювати з кожним записом у наборі. Потім дані групуються за полем HeartDiseaseorAttack, яке відображає наявність або відсутність ризику захворювання, і для кожної групи підраховується кількість зразків. Результати виводяться в текстове поле ReportTextBox, де для кожного класу додається повідомлення, що вказує кількість зразків та позначення ризику, що забезпечує користувачу наочний огляд структури даних.

Щоб оцінити продуктивність процесу навчання моделі, починається вимірювання часу за допомогою об'єкта Stopwatch, який фіксує тривалість цього процесу. Після запуску таймера виконується виклик методу

BuildAndTrainModel, який будує та тренує модель, використовуючи тренувальний набір даних із попередньо розділеного масиву (рис. 3.6).

```
var stopwatch = Stopwatch.StartNew();
// Побудова та тренування моделі
trainedModel = BuildAndTrainModel(mlContext, splitData.TrainSet);
```

Рисунок 3.6 – Вимірювання часу та запуск процесу тренування моделі

Даний підхід дозволяє не лише створити модель для прогнозування ризику, але й отримати точні часові показники, що допомагають оцінити ефективність обчислень і ресурсоемність операції навчання.

У фрагменті коду, представленому на рисунку 3.7, визначається масив featureColumns, який містить перелік полів, що використовуватимуться як вхідні ознаки для моделі машинного навчання.

```
var featureColumns = new[]
{
    nameof(HeartData.HighBP), nameof(HeartData.HighChol),
    nameof(HeartData.CholCheck), nameof(HeartData.BMI),
    nameof(HeartData.Smoker), nameof(HeartData.Stroke),
    nameof(HeartData.Diabetes), nameof(HeartData.PhysActivity),
    nameof(HeartData.Fruits), nameof(HeartData.Veggies),
    nameof(HeartData.HvyAlcoholConsump), nameof(HeartData.AnyHealthcare),
    nameof(HeartData.NoDocbcCost), nameof(HeartData.GenHlth),
    nameof(HeartData.MentHlth), nameof(HeartData.PhysHlth),
    nameof(HeartData.DiffWalk), nameof(HeartData.Sex),
    nameof(HeartData.Age), nameof(HeartData.Education),
    nameof(HeartData.Income)
};
```

Рисунок 3.7 – Визначення масиву featureColumns

Кожна з ознак представляє певний аспект здоров'я або поведінкові характеристики пацієнта, такі як рівень артеріального тиску (HighBP), індекс маси тіла (BMI), наявність діабету (Diabetes), та інші фактори, які можуть впливати на ризик серцевих захворювань. Така структура ознак забезпечує комплексний підхід до тренування моделі, оскільки дозволяє врахувати широкий спектр характеристик пацієнтів, що підвищує точність та адаптивність моделі для прогнозування ризику.

Код на рисунку 3.8 формує конвеєр обробки та навчання моделі, який забезпечує комплексну підготовку даних і їхнє використання для тренування класифікатора. Спочатку ознаки об'єднуються в єдину колонку під назвою Features, що дозволяє обробляти їх разом. Далі відбувається заміна відсутніх значень, нормалізація даних для зменшення розкиду значень та покращення стабільності навчання.

```
var trainingPipeline = mlContext.Transforms.Concatenate("Features", featureColumns)
    .Append(mlContext.Transforms.ReplaceMissingValues("Features"))
    .Append(mlContext.Transforms.NormalizeMeanVariance("Features"))
    .Append(mlContext.Transforms.Conversion.ConvertType(
        outputColumnName: "Label",
        inputColumnName: nameof(HeartData.HeartDiseaseorAttack),
        outputKind: DataKind.Boolean))
    .Append(mlContext.BinaryClassification.Trainers.FastTree(
        labelColumnName: "Label",
        featureColumnName: "Features"));
```

Рисунок 3.8 – Код конвеєра обробки та навчання моделі

Колонка з вихідною змінною HeartDiseaseorAttack перетворюється у формат булевого типу з назвою Label, що дозволяє зручно працювати з метою класифікації. Заключним етапом є застосування алгоритму FastTree для бінарної класифікації, який виконує тренування моделі на основі підготовлених ознак, дозволяючи прогнозувати ризик серцевих захворювань на основі наданих характеристик.

У фрагменті коду на рисунку 3.9 виконується тренування моделі, шляхом застосування підготовленого конвеєра trainingPipeline до навчального набору даних trainData.

```
var model = trainingPipeline.Fit(trainData);
```

Рисунок 3.9 – Тренування моделі

Команда Fit запускає процес навчання, де алгоритм оптимізує параметри моделі на основі тренувальних даних, створюючи структуру, здатну здійснювати точні прогнози. Після завершення цього етапу модель зберігає інформацію про

патерни в даних, що стане основою для подальшого прогнозування ризику серцевих захворювань для нових пацієнтів.

Метод «Evaluate» призначений для оцінювання якості навченої моделі, надаючи користувачеві інформацію про її продуктивність (рис. 3.10). Спочатку виводиться повідомлення в текстове поле RaportTBox, що процес оцінки моделі розпочався. За допомогою методу Transform модель застосовується до тестових даних, генеруючи прогнози, на основі яких обчислюються основні метрики. Далі метрика Accuracy показує точність моделі в класифікації, F1 Score дає уявлення про баланс між точністю та повнотою, а AUC відображає здатність моделі відрізняти класи за шкалою від 0 до 1. Результати оцінки додаються до текстового поля, дозволяючи користувачеві швидко ознайомитися з ключовими показниками якості моделі.

```
private void Evaluate(MLContext mlContext, ITransformer model, IDataView testData) {
    ReportTBox.Text += ("Оцінка моделі...");
    var predictions = model.Transform(testData);
    var metrics =
        mlContext.BinaryClassification.Evaluate(predictions, labelColumnName: "Label");
    ReportTBox.Text += ($"\r\nТочність: {metrics.Accuracy:P2}");
    ReportTBox.Text += ($"\r\nF1 Score: {metrics.F1Score:P2}");
    ReportTBox.Text += ($"\r\nAUC: {metrics.AreaUnderRocCurve:P2}");
}
```

Рисунок 3.10 – Код методу «Evaluate»

Наступним кроком реалізовано форму «Діагностування», що призначена для надання користувачу можливості здійснювати аналіз ризику серцевих захворювань на основі введених даних, а також взаємодії з побудованими моделями машинного навчання (рис. 3.11).

Рисунок 3.11 – Реалізація форми «Діагностування»

На формі реалізовано механізми для введення даних, вибору моделей та відображення результатів, що забезпечує зручність і інформативність процесу діагностики.

У формі оголошуються й ініціалізуються основні змінні та об'єкти, необхідні для функціонування форми «Діагностування» (рис. 3.12).

```
private ValidationMy _Validation = new ValidationMy();
private Models _SelectedModels = new Models();
private MLContext context = new MLContext();
private PredictionEngine<HeartData, HeartPrediction> predictionEngine;

private ModelsProvider _ModelsProvider = new ModelsProvider();
private List<Models> _ModelsList = new List<Models>();
private bool _IsModelsLoad = false;
private LogsProvider _LogsProvider = new LogsProvider();
private DiabetesApp _DiabetesApp = new DiabetesApp();
private List<Diabetes> _DiabetesL = new List<Diabetes>();
private GenHlthApp _GenHlthApp = new GenHlthApp();
private List<GenHlth> _GenHlthL = new List<GenHlth>();
```

Рисунок 3.12 – Оголошення основних змінних та об'єктів

Створюється об'єкт `_Validation` для перевірки введених користувачем даних і `_SelectedModels` для зберігання обраної моделі. Ініціалізується `MLContext`, що забезпечує доступ до інструментів машинного навчання, та

predictionEngine, який використовується для прогнозування ризику на основі введених даних. ModelsProvider і _ModelsList відповідають за завантаження й управління переліком доступних моделей, причому _IsModelsLoad вказує, чи були моделі завантажені в систему. Також додаються об'єкти _LogsProvider для ведення записів логів і _DiabetesApp та _GenHlthApp, які взаємодіють з відповідними списками _DiabetesL і _GenHlthL, що містять інформацію про додаткові параметри стану здоров'я користувача для більш точного діагностування.

Метод, наведений на рисунку 3.13 обробляє подію зміни значення у випадальному списку ModelsCBox, що дозволяє користувачу вибрати бажану модель для діагностики.

```
private void ModelsCBox_SelectedValueChanged(object sender, EventArgs e) {
    if (_IsModelsLoad && IsModelExist()) {
        _SelectedModels = _ModelsProvider.SelectedModelsById(
            Convert.ToInt32(ModelsCBox.SelectedValue));
        LoadData(_SelectedModels.ModelsFileModel);
    }
}
```

Рисунок 3.13 – Оголошення основних змінних та об'єктів

На початку перевіряється, чи були моделі завантажені та чи існує обрана модель у системі. Якщо обидві умови виконуються, об'єкт _SelectedModels оновлюється, отримуючи відповідну модель через метод SelectedModelsById з використанням обраного ідентифікатора. Після цього викликається метод LoadData, який завантажує дані з файлу моделі, забезпечуючи її готовність до подальшого прогнозування.

Метод PredictBtn_Click запускає процес прогнозування ризику серцевих захворювань, активуючись при натисканні кнопки «Прогнозування» (рис. 3.14). Спершу здійснюється перевірка, чи всі дані введені правильно і чи обрана модель доступна для використання. Після цього створюється об'єкт HeartData, заповнений значеннями з інтерфейсу – кожен вхідний параметр, такий як артеріальний тиск, індекс маси тіла, рівень фізичної активності тощо,

передається як ознака для моделі, включаючи обрані опції та значення з полів вводу. Далі викликається метод Predict, який застосовує модель до введених даних і обчислює прогноз.

```
private void PredictBtn_Click(object sender, EventArgs e) {
    // Перевіряємо, чи всі введені дані коректні
    if (IsAllHeartDataCorrect() && IsModelExist()) {
        // Створюємо об'єкт HeartData з даними на основі введених значень
        HeartData testHeartData = new HeartData {
            HighBP = HighBPChBox.Checked ? 1f : 0f,
            HighChol = HighCholChBox.Checked ? 1f : 0f,
            CholCheck = CholCheckChBox.Checked ? 1f : 0f,
            BMI = (float)Convert.ToDouble(BMITBox.Text),
            Smoker = SmokerChBox.Checked ? 1f : 0f,
            Stroke = StrokeChBox.Checked ? 1f : 0f,
            Diabetes = (float)Convert.ToDouble(DiabetesCBox.SelectedValue),
            PhysActivity = PhysActivityChBox.Checked ? 1f : 0f,
            Fruits = FruitsChBox.Checked ? 1f : 0f,
        };
        // Прогнозування на основі введених даних
        var prediction = predictionEngine.Predict(testHeartData);
        MonitoringTBox.Clear();
        // Формуємо результат прогнозування
        var answer = new StringBuilder();
        answer.AppendLine("\r\n--- Прогнозування ---");
        answer.AppendLine($"{"\r\nРекомендований клас: {{prediction.Prediction ? "Є ризик" : "Немає ризику"}}"});
        answer.AppendLine($"{"\r\nЙмовірність наявності ризику: {prediction.Probability:P2}"});
        // Виведення результату прогнозування
        MonitoringTBox.Text += answer.ToString();
    }
}
```

Рисунок 3.14 – Фрагмент методу для прогнозування захворювання

Після отримання результату прогнозування очищується текстове поле MonitoringTBox, де відобразяться результати. З використанням об'єкта StringBuilder формується текстовий звіт, що містить рекомендації: чи наявний ризик, а також вказується ймовірність його наявності у відсотках. Завершений звіт додається до MonitoringTBox, надаючи користувачеві чітку інформацію про ризик серцевих захворювань на основі введених даних.

3.3 Реалізація додаткових модулів системи

Під час розробки системи діагностики ризику серцевих захворювань важливим аспектом є забезпечення безпеки та конфіденційності даних користувачів. Для цього додаткові модулі включають функції шифрування інформації, які дозволяють захищати особисті та медичні дані від

несанкціонованого доступу. Одним з таких модулів є реалізація механізму шифрування текстових даних, що забезпечує захист інформації під час зберігання та передачі.

Метод Encrypt реалізує алгоритм шифрування тексту, використовуючи стандарт AES, що надає високий рівень захисту (рис. 3.15).

```
public string Encrypt(string originalString) {
    // Перевірка, чи вхідний рядок не є null або порожнім
    if (String.IsNullOrEmpty(originalString)) {
        throw new ArgumentNullException("The string which needs to be encrypted can not be null.");
    }
    // Використання AES шифрування з використанням CryptoServiceProvider
    using (AesCryptoServiceProvider aes = new AesCryptoServiceProvider()) {
        // Встановлення ключа та IV для шифрування
        aes.Key = Key;
        aes.IV = IV;
        // Створення потоку в пам'яті для зберігання зашифрованих даних
        MemoryStream memoryStream = new MemoryStream();
        // Створення потоку шифрування
        CryptoStream cryptoStream = new CryptoStream(memoryStream,
            aes.CreateEncryptor(aes.Key, aes.IV), CryptoStreamMode.Write);
        // Запис вхідного рядка в потік шифрування
        StreamWriter writer = new StreamWriter(cryptoStream);
        writer.Write(originalString);
        writer.Flush();
        // Завершення шифрування
        cryptoStream.FlushFinalBlock();
        writer.Flush();
        // Повернення зашифрованих даних у форматі Base64
        return Convert.ToBase64String(memoryStream.ToArray());
    }
}
```

Рисунок 3.15 – Код методу для шифрування даних системи

На початку відбувається перевірка вхідного рядка, щоб запобігти випадкам, коли текст для шифрування є порожнім або відсутнім, – у такому випадку генерується виняток. Далі з використанням AesCryptoServiceProvider створюється об'єкт AES, якому задаються ключ (Key) та вектор ініціалізації (IV), необхідні для процесу шифрування. Застосовується потік пам'яті (MemoryStream) для зберігання зашифрованих даних, а потік CryptoStream у поєднанні з StreamWriter дозволяє записувати вхідний рядок, шифруючи його в режимі запису. Після завершення шифрування дані конвертуються в Base64-

формат і повертаються як зашифрований рядок, готовий для безпечного зберігання або передачі в захищеному вигляді.

Метод «Decrypt» виконує розшифрування закодованого тексту, використовуючи алгоритм AES, що гарантує безпечне відновлення вихідних даних (рис. 3.16). На початку здійснюється перевірка вхідного рядка `cryptutedString`, щоб упевнитися, що текст для розшифрування не є порожнім чи відсутнім. Якщо рядок порожній, генерується виняток, забезпечуючи таким чином коректність вхідних даних.

```
public string Decrypt(string cryptutedString) {
    // Перевірка, чи вхідний рядок для розшифрування не є null або порожнім
    if (String.IsNullOrEmpty(cryptutedString)) {
        throw new ArgumentNullException("The string which needs to be decrypted can not be null.");
    }
    // Конвертація рядка з Base64 у масив байтів
    var fullCipher = Convert.FromBase64String(cryptutedString);
    // Використання AES шифрування з використанням CryptoServiceProvider
    using (AesCryptoServiceProvider aes = new AesCryptoServiceProvider()) {
        // Встановлення ключа та IV для розшифрування
        aes.Key = Key;
        aes.IV = IV;
        // Створення потоку в пам'яті для зберігання розшифрованих даних
        MemoryStream memoryStream = new MemoryStream(fullCipher);
        // Створення потоку розшифрування
        CryptoStream cryptoStream = new CryptoStream(memoryStream,
            aes.CreateDecryptor(aes.Key, aes.IV), CryptoStreamMode.Read);
        // Читання розшифрованих даних
        StreamReader reader = new StreamReader(cryptoStream);
        return reader.ReadToEnd(); // Повернення розшифрованого рядка
    }
}
```

Рисунок 3.16 – Код методу для розшифрування даних системи

Рядок у форматі Base64 перетворюється в масив байтів, що готує дані для їх обробки алгоритмом AES. Використовуючи об'єкт `AesCryptoServiceProvider`, задаються ключ (`Key`) та вектор ініціалізації (`IV`), необхідні для успішного розшифрування. Вміст зашифрованих даних передається в потік пам'яті (`MemoryStream`), а `CryptoStream` разом із `StreamReader` забезпечують процес читання даних у режимі розшифрування. Завершальний етап повертає оригінальний текст, відновлений із зашифрованого формату, забезпечуючи безпечний доступ до вихідної інформації.

На рисунку 3.17 зображено форму авторизації, яка надає користувачам можливість увійти в систему, ввівши логін і пароль. Форма містить стандартні елементи для введення даних, такі як текстові поля та кнопку підтвердження. Для забезпечення захищеного входу реалізовано шифрування паролів та перевірку на коректність введених даних, що допомагає уникнути несанкціонованого доступу до системи.

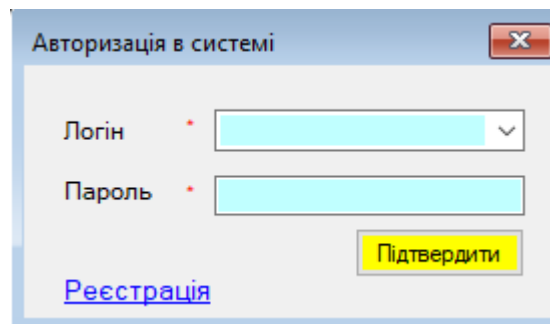


Рисунок 3.17 – Форма забезпечення безпеки даних системи

Для обробки введених даних під час авторизації користувача розроблено метод «GetSubmitData», код якого приведено на рисунку 3.18.

```
private void GetSubmitData() {
    try {
        if (IsDataEnteringCorrect()) {
            List<Users> listUsers = new List<Users>();
            listUsers =
                _UserProvider.SelectedUsersByUserNameAndUsersPassword(UserNameCBox.Text, UserPasswordTbx.Text);
            if (listUsers.Count > 0) {
                CurrentUser = listUsers[0];
                DataLoad();
            } else {
                MessageBox.Show(NamesMy.MessageBoxExaption.ThisUserLoginAndUserPasswordNotExistInSystem,
                    NamesMy.MessageBoxExaption.CaptionMessage);
            }
        }
    } catch (Exception ex) {
        MessageBox.Show(ex.Message);
    }
}
```

Рисунок 3.18 – Код методу «GetSubmitData»

Спершу він перевіряє коректність введених логіна та пароля за допомогою методу IsDataEnteringCorrect, що забезпечує попередню валідацію. Якщо введені дані є коректними, метод звертається до _UserProvider, щоб отримати список користувачів, що відповідають введеним логіну та пароллю. Якщо в системі

знайдено відповідного користувача, інформація про нього зберігається в змінну `CurrentUser`, і викликається метод `DataLoad` для завантаження необхідних даних. У випадку, якщо користувача з таким логіном і паролем не знайдено, користувачу виводиться повідомлення з відповідним текстом. Якщо під час виконання коду виникає виняток, його повідомлення відображається у вікні повідомлень, що дозволяє діагностувати можливі помилки.

Метод наведений на рисунку 3.19 здійснює вибірку користувачів із бази даних на основі введених імені користувача та пароля. Спочатку створюється SQL-запит, який шукає записи в таблиці `Users`, де логін і пароль збігаються із заданими параметрами. Після встановлення з'єднання з базою даних формується об'єкт `SqlCommand`, до якого додаються параметри для захисту від SQL-ін'єкцій, причому пароль перед передачею шифрується методом `_encryptData.Encrypt` для забезпечення безпеки.

```
public List<Users> SelectedUsersByUserNameAndPassword(string UserName, string UsersPassword) {
    List<Users> UsersList = new List<Users>();
    string sqlExpression = "SELECT * FROM Users WHERE UserName=@UserName AND UsersPassword=@UsersPassword";
    using (SqlConnection connection = new SqlConnection(_ConnString)) {
        connection.Open();
        SqlCommand command = new SqlCommand(sqlExpression, connection);
        // Додавання параметрів
        command.Parameters.AddWithValue("@UserName", UserName);
        command.Parameters.AddWithValue("@UsersPassword", _encryptData.Encrypt(UsersPassword));
        SqlDataReader reader = command.ExecuteReader();
        if (reader.HasRows) {
            while (reader.Read()) {
                Users oneUsers = new Users();
                oneUsers.UserId = Convert.ToInt32(reader["UserId"]);
                oneUsers.FirstName = reader["FirstName"].ToString();
                oneUsers.LastName = reader["LastName"].ToString();
                oneUsers.FIO = oneUsers.LastName + " " + oneUsers.FirstName;
                oneUsers.UserName = reader["UserName"].ToString();
                oneUsers.UsersPassword = _encryptData.Decrypt(reader["UsersPassword"].ToString());
                oneUsers.RoleId = Convert.ToInt32(reader["RoleId"]);
                oneUsers.RoleName = GetRoleName(oneUsers.RoleId);
                oneUsers.Description = reader["Description"].ToString();
                UsersList.Add(oneUsers);
            }
        }
        reader.Close();
    }
    return UsersList;
}
```

Рисунок 3.19 – Код методу для вибірки користувачів

Запит виконується, і якщо в результаті є відповідні записи, то кожен запис обробляється окремо. Інформація про користувача зчитується з результатів і зберігається в об'єкті `Users`, заповнюючи такі поля, як `UserId`, ім'я, прізвище,

об'єднане повне ім'я (FIO), логін та роль. Пароль розшифровується перед збереженням, щоб забезпечити його зручне використання у програмі. Після формування об'єктів усі знайдені записи додаються до списку `UsersList`, який повертається як результат методу, забезпечуючи доступ до інформації про користувача, що відповідає введеним даним.

3.4 Тестування програми

У процесі тестування програми важливо забезпечити якість та відповідність моделі, що прогнозує ризик серцевих захворювань, на основі релевантних і збалансованих даних. Одним з основних джерел даних для навчання та перевірки моделі став `Heart Disease Health Indicators Dataset`, доступний на платформі `Kaggle`. Проте, оригінальний набір даних потребував ретельної попередньої обробки для досягнення високої якості моделювання і точності прогнозування ризику серцевих захворювань. Попередня обробка даних є необхідним етапом у процесі підготовки, оскільки неповні або зайві записи можуть знижувати ефективність і точність навчальної моделі, викликаючи помилки в прогнозах.

Першим кроком стала очистка від дублікатів, оскільки наявність однакових записів могла б призвести до надмірного впливу певних даних на модель, викривляючи її результати та обмежуючи здатність до узагальнення. Наступним етапом була обробка відсутніх значень, оскільки прогалини у даних могли негативно вплинути на навчання. Вибір стратегії для роботи з відсутніми значеннями є важливим рішенням, яке залежить від природи даних та їх значущості в процесі моделювання. У нашому випадку було вирішено видалити записи з численними пропущеними значеннями, щоб зберегти надійність інформації, або замінити одиничні пропущені значення на середні або модальні значення, залежно від типу показника.

Після виконання цих етапів обробки вдалося сформувати збалансований датасет, у якому кількість зразків у кожному з класів практично однакова: 15322

зразки для класу «Немає ризику» та 15242 зразки для класу «Є ризик». Така рівномірність у кількості прикладів кожного класу є важливою для забезпечення збалансованості моделі, оскільки дозволяє моделі навчитися розпізнавати обидва класи з однаковою точністю, не віддаючи переваги жодному з них. Це особливо актуально у випадку медичних прогнозів, де рівномірний розподіл між класами дозволяє мінімізувати ризик хибних діагнозів.

Приклад фрагмента вікна з тренувальними даними, що були підготовлені для навчання моделі, наведено на рисунку 3.20.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	HeartDiseaseorAttack,HighBP,HighChol,CholCheck,BMI,Smoker,Stroke,Diabetes,PhysActivity,Fruits,Veggies,HvyAlcoholConsump,A												
2	0,1,1,1,40,1,0,0,0,0,1,0,1,0,5,18,15,1,0,9,4,3												
3	0,0,0,0,25,1,0,0,1,0,0,0,0,1,3,0,0,0,0,7,6,1												
4	0,1,1,1,28,0,0,0,0,1,0,0,1,1,5,30,30,1,0,9,4,8												
5	0,1,0,1,27,0,0,0,1,1,1,0,1,0,2,0,0,0,0,11,3,6												
6	0,1,1,1,24,0,0,0,1,1,1,0,1,0,2,3,0,0,0,11,5,4												
7	0,1,1,1,25,1,0,0,1,1,1,0,1,0,2,0,2,0,1,10,6,8												
8	0,1,0,1,30,1,0,0,0,0,0,0,1,0,3,0,14,0,0,9,6,7												
9	0,1,1,1,25,1,0,0,1,0,1,0,1,0,3,0,0,1,0,11,4,4												
10	0,0,0,1,24,0,0,0,0,0,1,0,1,0,2,0,0,0,1,8,4,3												
11	0,0,0,1,25,1,0,2,1,1,1,0,1,0,3,0,0,0,1,13,6,8												
12	0,1,1,1,34,1,0,0,0,1,1,0,1,0,3,0,30,1,0,10,5,1												
13	0,0,0,1,26,1,0,0,0,0,1,0,1,0,3,0,15,0,0,7,5,7												
14	0,1,1,1,28,0,0,2,0,0,1,0,1,0,4,0,0,1,0,11,4,6												
15	0,0,1,1,33,1,1,0,1,0,1,0,1,1,4,30,28,0,0,4,6,2												
16	0,1,0,1,33,0,0,0,1,0,0,0,1,0,2,5,0,0,0,6,6,8												
17	0,1,1,1,21,0,0,0,1,1,1,0,1,0,3,0,0,0,0,10,4,3												

Рисунок 3.20 – Фрагмент збалансованих даних для тренування моделей

Під час навчання моделі на основі збалансованого датасету було завантажено файл з тренувальними даними, що дозволило системі обробити дані, пов'язані з ризиком серцевих захворювань. Після завантаження даних виконано навчання моделі з обчисленням ключових метрик для оцінки її якості та ефективності в прогнозуванні ризику. На рисунку 3.21 зображено процес

навчання моделі, де представлено послідовність виконання операцій, включно з підрахунком кількості зразків у кожному класі та часом, витраченим на етапи тренування та оцінки моделі.

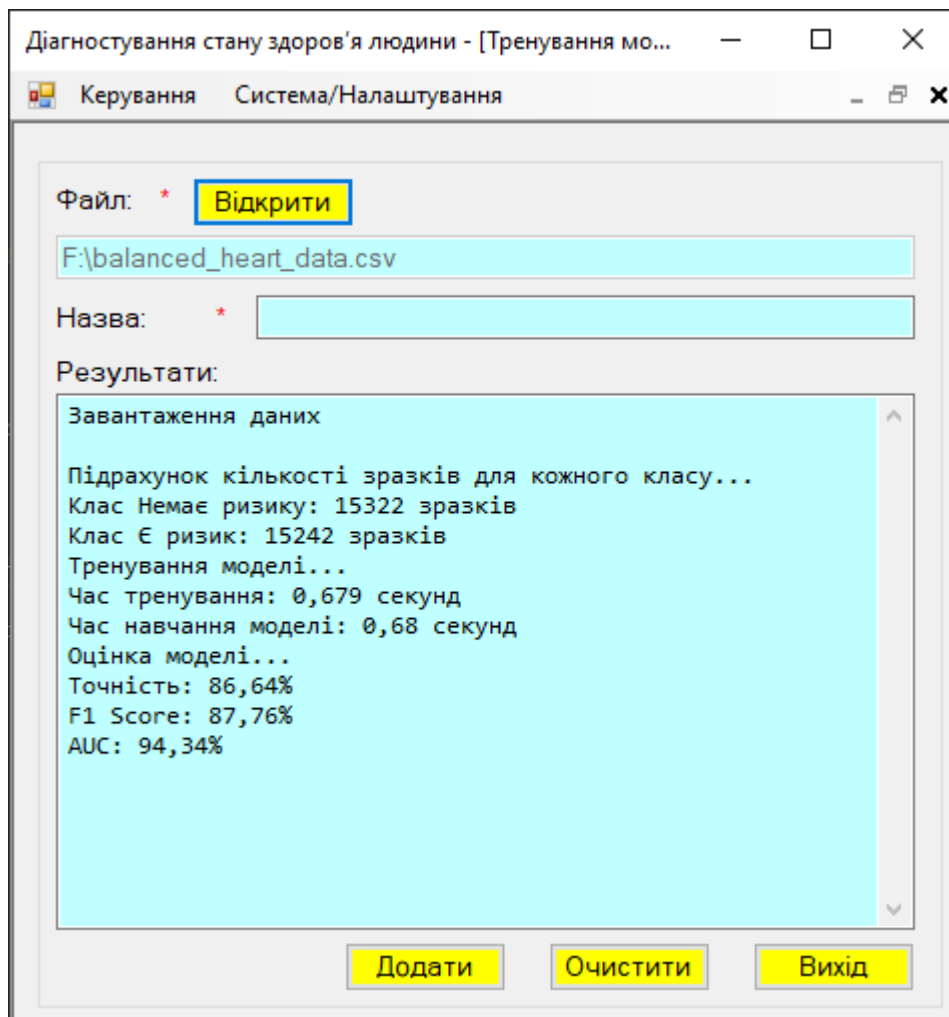


Рисунок 3.21 – Результати навчання моделі

В ході навчання система здійснила оцінку основних метрик для побудованої моделі, що дозволяє комплексно оцінити її здатність до точного прогнозування ризику серцевих захворювань. Зокрема, метрика точності становить 86.64%, що свідчить про високий рівень коректних передбачень для більшості випадків. Такий показник точності є хорошим результатом для системи, орієнтованої на класифікацію медичних ризиків, оскільки він вказує на спроможність моделі правильно визначати наявність або відсутність ризику на основі численних ознак.

F1 Score моделі склав 87.76%, що вказує на добре збалансоване співвідношення між точністю та повнотою. Це значення особливо важливе у випадках, де хибно-негативні або хибно-позитивні результати можуть мати серйозні наслідки для пацієнта. Високе значення F1 Score означає, що модель зберігає високу чутливість до випадків ризику, водночас знижуючи ймовірність пропуску випадків, що потребують уваги.

Метрика AUC (Area Under Curve) на рівні 94.34% демонструє чудову здатність моделі розрізняти класи, тобто коректно відрізняти пацієнтів із ризиком серцевих захворювань від тих, у кого ризик відсутній. Високе значення AUC вказує на надійність моделі при визначенні відмінностей між класами навіть у складних випадках, що забезпечує точність прогнозування у реальних умовах використання.

Час навчання моделі також є важливим показником, адже оперативність обробки медичних даних дозволяє зберігати актуальність та ефективність моделі у великих системах. Модель тренувалася 0.679 секунди, що свідчить про високу швидкість роботи алгоритму навіть із порівняно великим датасетом. Це дозволяє системі швидко оновлювати модель при додаванні нових даних, забезпечуючи її актуальність і точність у довгостроковій перспективі.

Загалом, отримані метрики свідчать про високу якість навченої моделі, яка здатна точно передбачати ризик серцевих захворювань, забезпечуючи надійність та ефективність у медичній діагностиці. Завдяки високим показникам точності, F1 Score та AUC, система має потенціал для ефективного використання в медичній практиці, допомагаючи лікарям у швидкому та точному виявленні пацієнтів з підвищеним ризиком.

Після завершення навчання моделі прогнозування ризику серцевих захворювань було проведено серію тестувань для перевірки її здатності коректно оцінювати ризики на нових даних, які не використовувалися у процесі навчання. Основною метою цього етапу було оцінити, наскільки точно модель здатна прогнозувати ризик у реальних умовах, де можуть зустрічатися раніше не розглянуті комбінації характеристик та індивідуальні особливості пацієнтів.

Тестування виконувалося на спеціально відокремленому наборі даних, що забезпечує об'єктивну оцінку здатності моделі до узагальнення. Модель перевірялася в умовах, наближених до практичних, що дозволило оцінити її стійкість до варіацій у вхідних даних, таких як різні рівні фізичної активності, наявність або відсутність факторів ризику та інші медичні показники. Отримані результати тестів дозволили проаналізувати загальну точність та специфічність моделі, а також виявити можливі недоліки у прогнозуванні, що є важливим для підвищення надійності системи в реальних умовах.

На рисунку 3.22 приведено інтерфейс системи діагностування, що відображає результати тестування першого сценарію, який включає конкретні вхідні дані пацієнта для оцінки ризику серцевих захворювань. У цьому випадку пацієнт має високий артеріальний тиск і підвищений рівень холестерину, а також в анамнезі є перевірка рівня холестерину протягом останніх п'яти років. Індекс маси тіла становить 27, що може свідчити про зайву вагу. Додаткові фактори, такі як куріння, вживання алкоголю та проблеми зі здоров'ям, зазначені як негативні, що може дещо знижувати загальний ризик.

Діагностування стану здоров'я людини - [Діагностування]

Керування Система/Налаштування

Вхідні дані:

Модель: * Діагностика стану здоров'я серця

☒ Артеріальний тиск високий? ☒ Рівень холестерину високий?

☒ Перевірка рівня холестерину протягом останніх 5 років була?

Індекс маси тіла: * 27

☐ Куріння? ☒ Інсульт?

Діабет: * відсутній

☐ Фізична активність за останні 30 днів?

☐ Споживання фруктів щодня?

☒ Споживання овочів щодня?

☐ Вживання великої кількості алкоголю?

☒ Наявність медичного страхування?

☐ Відмова від візиту до лікаря через вартість послуг?

Загальний стан здоров'я: * посередній

Кількість днів з поганим психічним здоров'ям: * 0 (за останній місяць)

Кількість днів з поганим фізичним здоров'ям: * 0 (за останній місяць)

☐ Проблеми з ходьбою?

☒ Стать чоловіча?

Вікова група: * 10 (1-13)

Рівень освіти: * 6 (1-8)

Рівень доходу: * 800

Введені дані:

Артеріальний тиск високий?: Так

Рівень холестерину високий?: Так

Перевірка рівня холестерину?: Так

Індекс маси тіла: 27

Куріння?: Ні

Інсульт?: Так

Діабет: 0

Фізична активність?: Ні

Споживання фруктів щодня?: Ні

Споживання овочів щодня?: Так

Вживання великої кількості алкоголю?: Ні

Медичне страхування?: Так

Відмова від візиту до лікаря?: Ні

Загальний стан здоров'я: 3

Погане психічне здоров'я: 0 днів

Погане фізичне здоров'я: 0 днів

Проблеми з ходьбою?: Ні

Стать чоловіча?: Так

Вікова група: 10

Рівень освіти: 6

Рівень доходу: 800

--- Прогнозування ---

Рекомендований клас: Є ризик

Ймовірність наявності ризику: 83,90%

Прогнозувати

Генерувати

Рисунок 3.22 – Результати тестування першого сценарію

У результатах системи вказано, що для цього пацієнта прогнозований клас – "Є ризик", із ймовірністю наявності ризику 83.98%. Цей показник свідчить про високий рівень ризику серцевих захворювань, що виправдано наявністю кількох критичних факторів, таких як високий артеріальний тиск і холестерин, а також показник ВМІ, що свідчить про потенційно підвищений ризик. Загалом, система надає чіткий і деталізований висновок, що дозволяє медичному персоналу звернути увагу на можливість серцевих проблем у даного пацієнта та, за потреби, запропонувати профілактичні або лікувальні заходи.

У другому сценарії, як видно на рисунку 3.23, подано новий набір вхідних даних для пацієнта з метою оцінки ризику серцевих захворювань. У цьому випадку артеріальний тиск не є високим, але рівень холестерину підвищений, що є важливим фактором для моделі. Індекс маси тіла становить 19, що знаходиться у межах норми, вказуючи на відсутність зайвої ваги. Інші фактори ризику, такі як куріння, вживання алкоголю, проблеми зі здоров'ям та обмеження фізичної активності, також не зафіксовані.

Діагностування стану здоров'я людини - [Діагностування]

Керування Система/Налаштування

Вхідні дані:

Модель: * Діагностика стану здоров'я серця

☐ Артеріальний тиск високий? ☐ Рівень холестерину високий?

☒ Перевірка рівня холестерину протягом останніх 5 років була?

Індекс маси тіла: * 19

☐ Куріння? ☐ Інсульт?

Діабет: * відсутній

☒ Фізична активність за останні 30 днів?

☒ Споживання фруктів щодня?

☒ Споживання овочів щодня?

☐ Вживання великої кількості алкоголю?

☒ Наявність медичного страхування?

☐ Відмова від візиту до лікаря через вартість послуг?

Загальний стан здоров'я: * добрий

Кількість днів з поганим психічним здоров'ям: * 0 (за останній місяць)

Кількість днів з поганим фізичним здоров'ям: * 0 (за останній місяць)

☐ Проблеми з ходьбою?

☐ Стать чоловіча?

Вікова група: * 5 (1-13)

Рівень освіти: * 5 (1-8)

Рівень доходу: * 1200

Прогнозувати

Генерувати

Введені дані:

Артеріальний тиск високий?: Ні

Рівень холестерину високий?: Ні

Перевірка рівня холестерину?: Так

Індекс маси тіла: 19

Куріння?: Ні

Інсульт?: Ні

Діабет: 0

Фізична активність?: Так

Споживання фруктів щодня?: Так

Споживання овочів щодня?: Так

Вживання великої кількості алкоголю?: Ні

Медичне страхування?: Так

Відмова від візиту до лікаря?: Ні

Загальний стан здоров'я: 2

Погане психічне здоров'я: 0 днів

Погане фізичне здоров'я: 0 днів

Проблеми з ходьбою?: Ні

Стать чоловіча?: Ні

Вікова група: 5

Рівень освіти: 5

Рівень доходу: 1200

--- Прогнозування ---

Рекомендований клас: немає ризику

Ймовірність наявності ризику: 4,25%

Рисунок 3.23 – Результати тестування другого сценарію

Згідно з результатами прогнозування, система визначила, що для даного пацієнта клас – «Немає ризику» з ймовірністю 4.25%. Такий низький рівень ризику є очікуваним, враховуючи відсутність критичних показників, за винятком підвищеного рівня холестерину. Загальна оцінка моделі свідчить про те, що пацієнт має дуже низьку ймовірність розвитку серцевих захворювань у короткостроковій перспективі, що дозволяє рекомендувати профілактичні заходи для контролю рівня холестерину, але не потребує негайного медичного втручання.

У третьому сценарії, наведеному на рисунку 3.24, система діагностики отримала набір даних, який містить значну кількість факторів ризику для оцінки серцевих захворювань. Зокрема, пацієнт має високий артеріальний тиск і підвищений рівень холестерину, а також індекс маси тіла становить 49, що свідчить про значне ожиріння. Крім цього, у пацієнта діагностовано розвинутий діабет, що є серйозним фактором ризику для серцево-судинної системи. Також зазначено, що пацієнт не є фізично активним, має проблеми з ходьбою, а загальний стан здоров'я вважається поганим, що ще більше підвищує ймовірність серцевих проблем.

Діагностування стану здоров'я людини - [Діагностування]

Керування Система/Налаштування

Вхідні дані:

Модель: * Діагностика стану здоров'я серця

☒ Артеріальний тиск високий? ☒ Рівень холестерину високий?

☒ Перевірка рівня холестерину протягом останніх 5 років була?

Індекс маси тіла: * 49

☒ Куріння? ☒ Інсульт?

Діабет: * розвинутий

☐ Фізична активність за останні 30 днів?

☐ Споживання фруктів щодня?

☒ Споживання овочів щодня?

☐ Вживання великої кількості алкоголю?

☒ Наявність медичного страхування?

☐ Відмова від візиту до лікаря через вартість послуг?

Загальний стан здоров'я: * поганий

Кількість днів з поганим психічним здоров'ям: * 5 (за останній місяць)

Кількість днів з поганим фізичним здоров'ям: * 30 (за останній місяць)

☒ Проблеми з ходьбою?

☒ Стать чоловіча?

Вікова група: * 10 (1-13)

Рівень освіти: * 4 (1-8)

Рівень доходу: * 7

Прогнозувати

Генерувати

Введені дані:

Артеріальний тиск високий?: Так

Рівень холестерину високий?: Так

Перевірка рівня холестерину?: Так

Індекс маси тіла: 49

Куріння?: Так

Інсульт?: Так

Діабет: 2

Фізична активність?: ні

Споживання фруктів щодня?: ні

Споживання овочів щодня?: Так

Вживання великої кількості алкоголю?: ні

Медичне страхування?: Так

Відмова від візиту до лікаря?: ні

Загальний стан здоров'я: 3

Погане психічне здоров'я: 5 днів

Погане фізичне здоров'я: 30 днів

Проблеми з ходьбою?: Так

Стать чоловіча?: Так

Вікова група: 10

Рівень освіти: 4

Рівень доходу: 7

--- Прогнозування ---

Рекомендований клас: Є ризик

Ймовірність наявності ризику: 91,46%

Рисунок 3.24 – Результати тестування третього сценарію

Результати прогнозування моделі для цього пацієнта показують, що клас – «Є ризик» із високою ймовірністю 91.45%. Такий результат підтверджує критичний рівень ризику, враховуючи наявність численних негативних факторів, серед яких – високий ВМІ, підвищений рівень холестерину, артеріальний тиск, діабет та обмеження фізичної активності. Високий рівень ризику, визначений моделлю, свідчить про необхідність негайного медичного втручання для зниження факторів ризику та запобігання можливим ускладненням. Система надає користувачу чіткий сигнал про високу ймовірність серцевих захворювань, що є важливим для своєчасного реагування.

3.5 Аналіз результатів роботи програми прогнозування стану здоров'я людини

Для оцінки ефективності програми прогнозування стану здоров'я людини проведено аналіз результатів її роботи за допомогою серії тестувань, що включали різні сценарії та комбінації факторів ризику. Отримані результати представлені в таблицях та включають аналіз точності моделі, вплив окремих факторів ризику на прогноз та порівняння прогнозованих результатів із реальними даними.

Програма тестувалася на трьох основних сценаріях з різними наборами факторів ризику. Як показано в таблиці 3.3, кожен сценарій мав специфічний набір показників, які вплинули на прогноз і ймовірність ризику серцевих захворювань. Наприклад, у першому сценарії, де відсутні значні фактори ризику, модель класифікувала пацієнта як «Немає ризику» з ймовірністю 4.25%, демонструючи високу точність прогнозу. У другому та третьому сценаріях, де спостерігалися критичні показники, такі як високий артеріальний тиск, підвищений рівень холестерину, високий ВМІ та наявність діабету, модель коректно класифікувала пацієнтів у групу ризику з ймовірностями 83.98% та

91.45% відповідно. Такі результати підтверджують здатність моделі адаптуватися до різних умов та точно ідентифікувати наявність ризику.

Таблиця 3.3 – Результати тестування моделі за різними сценаріями

Сценарій	Клас ризику	Імовірність ризику	Виявлені фактори ризику	Оцінка точності (%)
Сценарій 1	Немає ризику	4.25%	Відсутність критичних факторів	96%
Сценарій 2	Є ризик	83.98%	Високий тиск, холестерин	87%
Сценарій 3	Є ризик	91.45%	Ожиріння, діабет, гіпертонія	91%

Аналіз показав, що окремі фактори мають різний вплив на підсумковий результат прогнозування. Як видно з таблиці 3.4, найбільший вплив мають такі фактори, як індекс маси тіла, наявність діабету та високий артеріальний тиск. Наприклад, наявність діабету збільшує ймовірність ризику до 89.32%, що вказує на високу чутливість моделі до цього показника. Високий ВМІ також є значним фактором, підвищуючи ймовірність ризику до 85.45%. Такі результати допомагають виділити найбільш критичні фактори, які необхідно враховувати при оцінці ризику.

Таблиця 3.4 – Основні фактори ризику та їх вплив на прогноз

Фактор ризику	Середня ймовірність ризику (%)	Вплив на загальну оцінку ризику
Гіпертонія	80.25%	Високий
Високий рівень холестерину	72.15%	Середній
Індекс маси тіла > 30	85.45%	Високий
Діабет	89.32%	Дуже високий
Відсутність фізичної активності	60.75%	Середній

Для остаточної перевірки точності моделі було проведено порівняння прогнозованих результатів з реальними даними пацієнтів. Як наведено в таблиці 3.5, у більшості випадків модель продемонструвала високу відповідність між прогнозованими та реальними результатами. Зокрема, для пацієнтів з високим ризиком (наприклад, Марія Коваленко), модель правильно виявила наявність ризику, а у випадках низького ризику (Іван Петренко) прогноз теж відповідав реальним показникам. Деякі незначні розбіжності вказують на можливі покращення в алгоритмах для більш точного прогнозування.

Таблиця 3.5 – Порівняння прогнозованих результатів із реальними

№ з/п	Пацієнт	Прогнозований ризик	Реальний результат	Відповідність результатів
1	Іван Петренко	Немає ризику	Немає ризику	Так
2	Марія Коваленко	Є ризик	Є ризик	Так
3	Олександр Бойко	Є ризик	Немає ризику	Ні
4	Наталія Ткаченко	Немає ризику	Є ризик	Ні
5	Анна Савченко	Є ризик	Є ризик	Так
6	Микола Дяченко	Немає ризику	Немає ризику	Так
7	Ольга Поліщук	Є ризик	Немає ризику	Ні
8	Василь Литвин	Немає ризику	Є ризик	Ні
9	Ганна Шевченко	Є ризик	Є ризик	Так
10	Дмитро Олійник	Немає ризику	Немає ризику	Так

Аналіз результатів роботи програми показав, що модель має високий рівень точності та є надійним інструментом для прогнозування ризику серцевих захворювань. Високі значення метрик точності, F1 Score та AUC свідчать про здатність моделі адекватно реагувати на різні рівні ризику, надаючи

користувачам достовірну інформацію для прийняття медичних рішень. Однак деякі розбіжності у прогнозах вказують на потенціал для подальшого вдосконалення моделі, наприклад, через розширення навчального набору даних або додавання нових релевантних ознак. Завдяки комплексному аналізу програма продемонструвала свою ефективність у медичних дослідженнях, забезпечуючи можливість точного та своєчасного прогнозування ризиків.

3.6 Висновок до розділу 3

У рамках даного розділу проведено комплексну програмну реалізацію та тестування інформаційної технології діагностики стану здоров'я людини, орієнтовану на прогнозування ризику серцевих захворювань. Спершу було обґрунтовано вибір мови програмування та середовища розробки: з-поміж мов Python, Java та C# обрано C#, що, у поєднанні з Visual Studio 2022, забезпечує надійність, високу продуктивність і зручність розробки. Далі реалізовано програмний модуль для прогнозування ризику серцевих захворювань із застосуванням алгоритмів машинного навчання, який включає функції для навчання та прогнозування, що дозволяє точно оцінювати ймовірність ризику.

Розроблено додаткові модулі для шифрування та розшифрування даних за допомогою алгоритму AES, що забезпечує захист інформації під час зберігання та передачі, підвищуючи безпеку системи. У процесі тестування програми модель досягла високих показників ефективності: точність склала 86.64%, F1 Score – 87.76%, а AUC – 94.34%, що свідчить про здатність моделі адекватно розпізнавати обидва класи ризику. Для ілюстрації роботи програми було розглянуто три сценарії з різними комбінаціями факторів ризику, що підтвердило її здатність адаптуватися до різних умов.

Аналіз результатів роботи програми показав, що найбільший вплив на підвищення ризику мають такі фактори, як діабет, високий індекс маси тіла та гіпертонія. Це дозволяє ідентифікувати найкритичніші ознаки, які необхідно

враховувати для ефективної профілактики серцевих захворювань. Порівняння прогнозованих результатів із реальними показало високу відповідність у більшості випадків, проте було виявлено декілька розбіжностей, що можуть бути використані для подальшого вдосконалення моделі. Отримані результати підкреслюють ефективність та надійність розробленої системи, яка успішно справляється із завданням прогнозування ризику серцевих захворювань і має потенціал для застосування в медичній практиці.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту інформаційної технології діагностування стану здоров'я.

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту залучаємо 3-х незалежних експертів, якими є провідні викладачі випускової або спорідненої кафедри[48].

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу здійснюємо із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, а результати зводимо до таблиці 4.1.

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу інформаційної технології діагностування стану здоров'я

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Технічна здійсненність концепції	2	3	2
Ринкові переваги (наявність аналогів)	1	2	2
Ринкові переваги (ціна продукту)	3	2	3
Ринкові переваги (технічні властивості)	2	3	1
Ринкові переваги (експлуатаційні витрати)	2	3	3
Ринкові перспективи (розмір ринку)	2	3	2
Ринкові перспективи (конкуренція)	2	1	2
Практична здійсненність (наявність фахівців)	3	3	2
Практична здійсненність (наявність фінансів)	3	2	3

Продовження таблиці 4.1

Практична здійсненність (необхідність нових матеріалів)	3	2	3
Практична здійсненність (термін реалізації)	2	3	2
Практична здійсненність (розробка документів)	3	2	1
Сума балів	28	29	26
Середньоарифметична сума балів, СБ	27,7		

За результатами розрахунків, наведених в таблиці 4.1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал інформаційної технології діагностування стану здоров'я – середній.

4.2 Розрахунок витрат на здійснення науково-дослідної розробки інформаційної технології діагностування стану здоров'я

Витрати на оплату праці. Належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Основна заробітна плата дослідників. Витрати на основну заробітну плату дослідників (З_о) розраховують відповідно до посадових окладів працівників, за формулою:

$$З_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p},$$

де k – кількість посад дослідників, залучених до процесу дослідження; M_{ni} – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.; T_p – число робочих днів в місяці; приблизно $T_p = (21 \dots 23)$ дні; t_i – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 4.2.

Таблиця 4.2 – Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	42000	2000	25	50000
Розробник	35000	1667	29	48333
Консультанти	20000	952	26	24762
Всього:	123095			

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників ($З_p$) за відповідними найменуваннями робіт розраховують за формулою:

$$З_p = \sum_{i=1}^n C_i \cdot t_i,$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год; t_i – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_m \cdot K_i \cdot K_c}{T_p \cdot t_{зм}},$$

де M_m – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), у 2024 році $M_m=8000$ грн; K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду; K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати; T_p – середня кількість робочих днів в місяці, приблизно $T_p = 21 \dots 23$ дні; $t_{зм}$ – тривалість зміни, год.

Таблиця 4.3 – Витрати на заробітну плату робітників

Найменування робіт	Трудовістість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Аналіз предметної області	22	5	61,8	1,36	1359,6
Моделювання інформаційної технології	94	6	65,9	1,45	6194,6
Створення основної структури проєкту	32	6	65,9	1,45	2108,8
Створення та наповнення бази даних	54	6	65,9	1,45	3558,6
Тестування інформаційної технології	68	5	61,8	1,36	4202,4
Всього					17424

Додаткова заробітна плата. Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як $(10 \dots 12)\%$ від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = 0,12 \cdot (Z_o + Z_p) = 0,12 \cdot (123095 + 17424) = 16862 \text{ грн.}$$

Відрахування на соціальні заходи. Нарахування на заробітну плату $H_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$H_{зп} = \beta \cdot (З_о + З_р + З_д) = \\ = 0,22 \cdot (123095 + 17424 + 16862) = 34624 \text{ грн.}$$

де $З_о$ – основна заробітна плата розробників, грн.; $З_р$ – основна заробітна плата робітників, грн.; $З_д$ – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Програмне забезпечення. До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10...12% від вартості програмного забезпечення. Балансову вартість програмного забезпечення розраховують за формулою:

$$V_{\text{прг}} = \sum_{i=1}^k C_{\text{іпрг}} \cdot C_{\text{прг.і}} \cdot K_i,$$

де $C_{\text{іпрг}}$ – ціна придбання програмного забезпечення i -го виду, грн.; $C_{\text{прг.і}}$ – кількість одиниць програмного забезпечення відповідного виду, шт.; K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного забезпечення, $K_i = (1,1 \dots 1,12)$; k – кількість видів програмного забезпечення.

Таблиця 4.4 – Витрати на придбання програмного забезпечення

Найменування програмного забезпечення	Ціна за одиницю, грн.	Витрачено	Вартість програмного забезпечення, грн.
VS Code Professional	1845	1	1845
Всього, з врахуванням коефіцієнта інсталяції та налагодження			2030

Амортизація обладнання. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування A в цілому бути розраховані за формулою:

$$A = \frac{Ц_6}{T_B} \cdot \frac{t}{12},$$

де $Ц_6$ – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; t – термін використання основного фонду, місяці; T_B – термін корисного використання основного фонду, роки.

Таблиця 4.5 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук	32000	2	1,5	2000
Клавіатура	9000	2	1,5	563
Мишка	800	2	1,5	50
Стіл	4000	5	1,5	100
Стілець	3200	5	1,5	80
Всього	2793			

Витрати на електроенергію для науково-виробничих цілей. Витрати на силову електроенергію $В_e$, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

Таблиця 4.6 – Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Ноутбук	0.056	270
Освітлення	0.01	120

$$\begin{aligned}
 Be &= \sum \frac{W_i \cdot t_i \cdot C_e \cdot K_{впi}}{ККД} = \frac{0,056 \cdot 270 \cdot 7,5 \cdot 0,95}{0,98} + \frac{0,01 \cdot 120 \cdot 7,5 \cdot 0,95}{0,98} \\
 &= 119 \text{ грн.},
 \end{aligned}$$

W_i – встановлена потужність обладнання, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год.; C_e – вартість 1 кВт електроенергії, грн.; $K_{впi}$ – коефіцієнт використання потужності; ККД – коефіцієнт корисної дії обладнання.

Інші витрати. До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{в} = (З_о + З_р) \cdot \frac{H_{ив}}{100\%} = (123095 + 17424) \cdot \frac{100}{100} = 140519 \text{ грн.},$$

де $H_{ив}$ – норма нарахування за статтею «Інші витрати».

Накладні (загальновиробничі) витрати. До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням

виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$В_{нзв} = (З_o + З_p) \cdot \frac{Н_{нзв}}{100\%} = (123095 + 17424) \cdot \frac{130}{100} = 182675 \text{ грн.},$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення науково-дослідної роботи. Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$\begin{aligned} В_{заг} &= З_o + З_p + З_{дод} + З_n + К_v + В_{спец} + В_{прг} + А_{обл} + В_e + \\ &\quad + I_v + В_{нзв} = \\ &= 123095 + 17424 + 16862 + 34624 + 2030 + 2793 + 119 + 140519 + \\ &\quad 182675 = 520141 \text{ грн.} \end{aligned}$$

Загальні витрати. Загальні витрати $ЗВ$ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{В_{заг}}{\eta} = \frac{520141}{0,2} = 577934 \text{ грн.},$$

де η – коефіцієнт, що характеризує етап виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на стадії впровадження, то $\eta=0,9$.

4.3 Розрахунок економічної ефективності науково-технічної розробки інформаційної технології діагностування стану здоров'я за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

В даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати: ΔN – збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу; N – кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки; Π_0 – вартість послуги у році до впровадження інформаційної системи; $\pm \Delta \Pi_0$ – зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi$ для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta \Pi = (\pm \Delta \Pi_0 \cdot N + \Pi_0 \cdot \Delta N_i)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right),$$

де $\pm \Delta \Pi$ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки); $\pm \Delta \Pi_0$ може мати як додатне, так і від'ємне значення (від'ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни); N – основний

кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки; Π_0 – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році; Π_6 – основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів; ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки); λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,8333$; ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho = 0,2 \dots 0,5$; θ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\theta = 18\%$.

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = ((1500-1270) \cdot 630 + 1270 \cdot 720) \cdot 0,83 \cdot 0,5 \cdot (1 - 0,18) = 360480 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = ((1500-1270) \cdot 630 + 1270 \cdot (720 + 500)) \cdot 0,83 \cdot 0,5 \cdot (1 - 0,18) = 576570 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = ((1500-1270) \cdot 630 + 1270 \cdot (720 + 500 + 450)) \cdot 0,83 \cdot 0,5 \cdot (1 - 0,18) = 771052 \text{ грн.}$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t} = \frac{360480}{(1 + 0,18)^1} + \frac{576570}{(1 + 0,18)^2} + \frac{771052}{(1 + 0,18)^3} = 2695524 \text{ грн.,}$$

де $\Delta\Pi$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.; T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо $T=1$ рік); τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$; t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot 3B = 2 \cdot 577934 = 1155868 \text{ грн.}$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}}=2 \dots 5$, але може бути і більшим; $3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \Pi\Pi - PV = 2695524 - 1155868 = 1539657 \text{ грн.,}$$

де $\Pi\Pi$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.; PV – теперішня вартість початкових інвестицій, грн.

Оскільки $E_{abc} > 0$, то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність E_v або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій E_v , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_v = \sqrt[T_{\text{ж}}]{1 + \frac{E_{abc}}{PV}} - 1 = \sqrt[3]{1 + \frac{1539657}{1155868}} - 1 = 0,32,$$

де $T_{\text{ж}}$ – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування $\tau_{\text{мін}}$, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{\text{мін}}$ визначається за формулою:

$$\tau_{\text{мін}} = d + f = 0,12 + 0,05 = 0,17,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = 0,9...0,12$; f – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина $f = 0,05...0,5$, але може бути і значно вищою.

Оскільки $E_v=0,32 > \tau_{\min}=0,17$, то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

Далі розраховуємо період окупності інвестицій T_o , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки інформаційної технології діагностування стану здоров'я:

$$T_o = \frac{1}{E_v} = \frac{1}{0,32} = 3,12 \text{ року.}$$

Оскільки $T_o < 1 \dots 4$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки інформаційної технології діагностування стану здоров'я і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

4.4 Висновок до розділу 4

Економічна частина роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 577934 гривень. Було спрогнозовано орієнтовну величину витрат по кожній з статей витрат. Також було розраховано чистий прибуток, який може отримати розробник від реалізації нового технічного рішення. Розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт є висококонкурентноспроможним і ціною дешевший ніж аналог.

ВИСНОВКИ

Кваліфікаційна робота присвячена розробці інформаційної системи прогнозування ризику серцевих захворювань, яка використовує метод градієнтного бустингу дерев рішень для точного аналізу медичних показників пацієнтів. Обраний метод дозволяє поетапно покращувати результати класифікації, коригуючи помилки попередніх моделей і забезпечуючи високу точність навіть для складних наборів даних. Розроблена система аналізує такі показники, як артеріальний тиск, індекс маси тіла та інші фактори ризику, і надає оцінку ймовірності серцевих захворювань для кожного пацієнта, що може сприяти ранній діагностиці та профілактиці. Додатково реалізовано модуль шифрування для захисту персональних даних пацієнтів, що робить систему безпечною та придатною для практичного використання у медичній сфері.

У першому розділі проведено огляд сучасних підходів до діагностики здоров'я людини, включаючи класичні методи та новітні технології, такі як штучний інтелект, машинне навчання та IoT, які дозволяють забезпечити точне і неінвазивне моніторування пацієнтів. Також детально вивчено фактори ризику серцевих захворювань, такі як артеріальний тиск, індекс маси тіла, шкідливі звички, генетична схильність та соціально-економічні фактори, і проаналізовано, як кожен із них впливає на здоров'я серця. У розділі представлено огляд існуючих програмних рішень та виявлено їхні переваги й недоліки, що дозволило обґрунтувати необхідність створення нової системи.

Другий розділ висвітлює процес розробки математичної моделі та вибору методів машинного навчання для прогнозування стану здоров'я. Було розглянуто різні алгоритми, такі як градієнтний бустинг дерев рішень, випадковий ліс та дерева рішень, і на основі порівняльного аналізу обрано градієнтний бустинг дерев рішень як оптимальний для розв'язання поставленого завдання. Для навчання моделі використано датасет Heart Disease Health Indicators із платформи Kaggle, що містить ключові медичні показники та фактори ризику. У розділі також детально описано математичну модель прогнозування та

класифікації ризиків, а також обґрунтовано вибір метрик для оцінки точності прогнозування, включаючи такі показники, як точність, F1 Score та AUC.

Третій розділ був присвячений програмній реалізації та тестуванню розробленої технології. Було обґрунтовано вибір мови програмування C# та середовища розробки Visual Studio 2022 для забезпечення продуктивності та зручності підтримки програми. Реалізовано основний модуль для прогнозування ризику серцевих захворювань, а також додаткові модулі для забезпечення безпеки даних за допомогою алгоритму шифрування AES. Проведено тестування моделі, за результатами якого було отримано такі метрики: точність – 86.64%, F1 Score – 87.76% та AUC – 94.34%, що свідчить про високу ефективність моделі у класифікації ризику. У процесі тестування було розглянуто три сценарії з різними комбінаціями факторів ризику, що підтвердило здатність моделі адаптуватися до різних умов і коректно класифікувати пацієнтів.

В четвертому розділі було спрогнозовано орієнтовну величину витрат по кожній з статей витрат. Також було розраховано чистий прибуток, який може отримати розробник від реалізації нового технічного рішення. Розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. Економічна частина роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 577934 гривень

Проведена робота демонструє застосування сучасних технологій машинного навчання для підвищення точності та ефективності медичних діагностичних систем. Завдяки детальному аналізу факторів ризику та порівнянню прогнозованих результатів із реальними вдалося підтвердити надійність розробленої моделі та визначити основні напрями для подальшого її вдосконалення. Розроблена система може бути використана в медичній практиці для раннього виявлення ризиків серцевих захворювань і підвищення якості профілактики та лікування. Отримані результати підтверджують значний потенціал розробленої інформаційної технології для підтримки медичних рішень, зменшення ризиків та поліпшення стану здоров'я пацієнтів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пасічнюк І. В., Хоменко В. М., Іванчук Я. В. Методи машинного навчання для діагностування стану здоров'я людини. Молодь в науці: дослідження, проблеми, перспективи(МН-2025), Вінниця, 15–16 червня 2025 р. [Електронний ресурс]. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/22711>(дата звернення 25.11.2024).
2. Сучасні інформаційні технології у сфері штучного інтелекту : електронний навчальний посібник комбінованого (локального та мережного) використання [Електронний ресурс] / Яровий А. А. , Крилик Л. В. , Козловський А. В. – Вінниця : ВНТУ, 2023. – 145 с.
3. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. К. Колесницький. – Вінниця : ВНТУ, 2023. – 58 с.16:26
4. Штучний інтелект. Машинне навчання / Григоров О. В., Аніщенко Г. О., Стрижак В. В., Петренко Н. О., Турчин О. В., Окунь А. О., Пономарьов О. Е. Сучасні технології : зб. наук. пр. / М-во освіти і науки України, ХНАДУ. - Харків, 2019. - 15с.
5. Баранов О. А. Інтернет речей: регулювання надання послуг роботами зі штучним інтелектом. Інформація і право, 2018, 4: – 60с.
6. Пономарьова Г. Ф., Молчанюк О. В., Тетьоркіна В. А., Дехтярьова О. О. Діагностика та моніторинг стану здоров'я людини : КЗ «Харківська гуманітарно-педагогічна академія» ХОР. – Харків, 2020. – 136 с.
7. Власюк А. І., Месюра В. І., Власюк Б. А. Автоматизована віртуальна система діагностики стану організму людини. Вінниця.2004.–78с.
8. Стенлець Й. І. Фізичне та математичне моделювання ультразвуко-вих методів діагностики стану здоров'я людини. Оптико-електронні інформаційно-енергетичні технології, Вінниця 2013, 26.2: 49 с.

9. Бродкевич В., Людвіченко В. Штучний інтелект і машинне навчання в галузі охорони здоров'я: виклики і перспективи. Інформаційні технології та суспільство, 2022, 2 (4): 28 с.

10. Онищенко К. Г. Аналіз методів обробки природної мови. Інформаційні системи та технології : матеріали 9-ї Міжнар. наук.-техн. конф., 17-20 листопада 2020 р. – Харків : Друкарня Мадрид, 2020. – С. 190.

11. JENKINS Stephen P. Survival analysis. Unpublished manuscript, Institute for Social and Economic Research, University of Essex, Colchester, UK, 2005, 42: pp. 54-56.

12. Ратушняк Р. М. Моніторинг використання IoT в медицині: відновлення здоров'я через інновації та технологічний прогрес. Всеукраїнська науково-технічна конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і Світу». Збірник тез. К.: ДУІКТ, 2023, 127 с.

13. Більчук Н. О.; Морозова М. М. Методи та аспекти вимірювання артеріального тиску. Кременчук, 2021, – 4с.

14. Буцька Л. В. Використання індексу маси тіла, як важливої складової фізичної та медичної реабілітації, для профілактики неінфекційних захворювань. Чернігів, Вісник, 2013, –144 с.

15. Гуменна Ю. М. Проблема шкідливих звичок. 2013. URL: https://dspace.udpu.edu.ua/bitstream/6789/1019/1/Problemy_alkoholizmu_ta_tituiunopralinnia.Pdf (дата звернення 02.11.2024).

16. Яремчук О., Тверда І., Симканич Н. Вплив генетичних факторів на розвиток серцево-судинних захворювань: сучасні підходи до лікування. Перспективи та інновації науки, 2024, 10 (44).

17. Іщейкіна Ю. О.; Іщейкін К. Є. Інтегральні соціально-економічні та соціально-психологічні фактори ризику виникнення серцево-судинних захворювань. Актуальні проблеми сучасної медицини: Вісник української медичної стоматологічної академії, 2011, 11.4-1 (36): 36-39.

18. Ada: Techniker Krankenkasse nutzt App mit Chat-Bot und KI zur Diagnose. URL: <https://www.computerwoche.de/article/2783101/techniker-krankenkasse-nutzt-app-mit-chat-bot-und-ki-zur-diagnose.html> (дата звернення 02.11.2024).

19. Ada Reviews & Product Details. URL: <https://www.g2.com/products/ada-support-inc-ada/reviews> (дата звернення 02.11.2024).

20. Ada Reviews. URL: <https://www.getapp.com/marketing-software/a/ada/reviews/> (дата звернення 02.11.2024).

21. Telehealth Application Around Globe: SkinVision. URL: <https://rahulpaith.com/telehealth-application-around-globe-skinvision/> (дата звернення 02.11.2024).

22. SkinVision - Find Skin Cancer 12+. URL: <https://apps.apple.com/gb/app/skinvision-find-skin-cancer/id545293136> (дата звернення 02.11.2024).

23. Aspiring Dermatology App Under The Microscope: The SkinVision Review. URL: <https://medicalfuturist.com/aspiring-dermatology-app-under-the-microscope-the-skinvision-review/> (дата звернення 02.11.2024).

24. AliveCor KardiaMobile® smartphone 1-channel ECG. DocCheck Shop. URL: <https://www.doccheckshop.eu/equipment/specialised-diagnostics/ecg/ecg-devices/12567/alivecor-kardiamobile-smartphone-1-channel-ecg> (дата звернення 02.11.2024).

25. KardiaMobile®. URL: https://store.kardia.com/products/kardiamobile?bvstate=pg:9/ct:r&srsId=AfmBOort8-cazJbxsF8asKYNiWWxxpLZCEMU pQ7S4_iRQNwVK4hll7Og (дата звернення 02.11.2024).

26. Comparison of Apple Watch vs KardiaMobile. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC9700214/> (дата звернення 02.11.2024).

27. Sun, R., Wang, G., Zhang, W., Hsu, L. T., & Ochieng, W. Y. A gradient boosting decision tree based GPS signal reception classification algorithm. Applied Soft Computing. 2020. Vol 86, 10 p.

28. Ensemble learning for the early prediction of neonatal jaundice with genetic features. URL: <https://www.researchgate.net/publication/356698772>

_Ensemble_learning_for_the_early_prediction_of_neonatal_jaundice_with_genetic_features (дата звернення 06.11.2024).

29. Fafalios, S., Charonyktakis, P., & Tsamardinos, I. Gradient boosting trees. Gnosis Data Analysis PC. 2020. URL: https://www.gnosisda.gr/wp-content/uploads/2020/07/Gradient_Boosting_Implementation.pdf (дата звернення 06.11.2024).

30. Sahin, E. K. Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest. SN Applied Sciences. 2020. Vol 2, No 7, 1308 p.

31. Alcolea, A., & Resano, J. FPGA accelerator for gradient boosting decision trees. Electronics. 2021. Vol 10, No 3, 314 p.

32. Schonlau, M., & Zou, R. Y. The random forest algorithm for statistical learning. The Stata Journal. 2020. Vol 20, No 1, pp. 3-30. Machine Learning Basics: Random Forest Classification. URL: <https://towardsdatascience.com/machine-learning-basics-random-forest-classification-499279bac51e> (дата звернення 06.11.2024).

33. Genuer, R., Poggi, J. M., Genuer, R., & Poggi, J. M. Random forests. Springer International Publishing. 2020. pp. 33-55.

34. Shah, K., Patel, H., Sanghvi, D., & Shah, M. A comparative analysis of logistic regression, random forest and KNN models for the text classification. Augmented Human Research. 2020. Vol 5, No 1, 12 p.

35. Zhu, T. Analysis on the applicability of the random forest. In Journal of Physics: Conference Series. 2020. Vol. 1607, No. 1, 6 p.

36. Khammas, B. M. Ransomware detection using random forest technique. ICT Express. 2020. Vol 6, No 4, pp. 325-331.

37. Izza, Y., Ignatiev, A., & Marques-Silva, J. On explaining decision trees. 2020. 20 p.

38. Why Do We Use Decision Trees in Machine Learning? URL: <https://www.turing.com/kb/importance-of-decision-trees-in-machine-learning> (дата звернення 06.11.2024).

39. Wan, A., Dunlap, L., Ho, D., Yin, J., Lee, S., Jin, H., ... & Gonzalez, J. E. NBDT: Neural-backed decision trees. 2020. 19 p.
40. Ghiasi, M., Zendehboudi, S., & Mohsenipour, A. Decision tree-based diagnosis of coronary artery disease: CART model. Computer methods and programs in biomedicine. 2020. Vol 192, 20 p.
41. Azad, M., Chikalov, I., & Moshkov, M. Representation of knowledge by decision trees for decision tables with multiple decisions. Procedia Computer Science. 2020. Vol 176, pp. 653-659.
42. Heart Disease Health Indicators Dataset. URL: <https://www.kaggle.com/datasets/alexteboul/heart-disease-health-indicators-dataset> (дата звернення 06.11.2024).
43. Городенко В.П., Марценюк О.П. Програмування на мові Python. Теорія та практика: Навчальний посібник. - К.: Центр навчальної літератури, 2016. - 480 с.
44. Карапецький В. П. Побудова графічного контенту додатків з використанням JavaFX і Swing компонентів і даних, взятих із баз даних / В. П. Карапецький. – Науковий вісник НЛТУ. – 2015. – 418 с.
45. Коноваленко І.В., Марущак П.О. Платформа .NET та мова програмування C# 8.0: навчальний посібник. Тернопіль: ФОП Паляниця В. А., 2020. 320 с.
46. Verma, R. Extending Visual Studio. In Visual Studio Extensibility Development: Extending Visual Studio IDE for Productivity, Quality, Tooling, Analysis, and Artificial Intelligence. 2023. pp. 73-113.
47. Kahlert T., Giza K. Visual Studio Code Tips & Tricks. Microsoft Deutschland GmbH. 2016. Vol. 1. 26 p.
48. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

Додаток А (обов'язковий)
Протокол перевірки кваліфікаційної роботи на наявність текстових
запозичень

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія діагностування стану здоров'я людини

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІІТА
(кафедра, факультет)

Показники звіту подібності Turnitin

Оригінальність 97,00% Схожість 3,00%

Аналіз звіту подібності (відмітити потрібне):

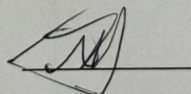
☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

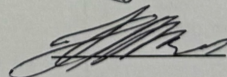
Ознайомлені з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи



Пасічнюк І.В.

Керівник роботи

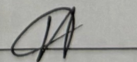


Іванчук Я.В.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (довідниковий)

Лістинг програми

Лістинг 1. Код класу «ModelsForm»

```
using HeartDiseaseorApp.AppCode;
using HeartDiseaseorApp.Forms.Systems;
using HeartDiseaseorApp.Providers;
using Microsoft.ML;
using Microsoft.ML.Data;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Globalization;
using System.IO;
using System.Diagnostics;

namespace HeartDiseaseorApp.Forms.SysMS {
    public partial class ModelsForm : Form {
        private MLContext mlContext;
        ITransformer trainedModel;
        private IDataView dataView;
        private string _Path = "";

        private int _selectedRowIndex = 0;
        private ValidationMy _Validation = new ValidationMy();
        private ModelsProvider _ModelsProvider = new ModelsProvider();
        private List<Models> _ModelsList = new List<Models>();
        private LogsProvider _LogsProvider = new LogsProvider();
        private bool _IsModelTrain = false;
        private Random _rand = new Random();

        public ModelsForm() {
            InitializeComponent();
            DataLoad();
        }

        private void OpenBtn_Click(object sender, EventArgs e) {
            // Створення діалогового вікна для відкриття файлу
            OpenFileDialog openFileDialog = new OpenFileDialog();
            // Налаштування властивостей діалогового вікна
            openFileDialog.Filter = "Text files (*.csv)|*.csv|All files (*.*)|*.*";
            openFileDialog.FilterIndex = 2;
            openFileDialog.RestoreDirectory = true;
            // Відображення діалогового вікна та обробка результату
```

```

if (openFileDialog.ShowDialog() == DialogResult.OK) {
    _Path = openFileDialog.FileName;
    FileNameTBox.Text = openFileDialog.FileName;

    // Створення контексту ML
    mlContext = new MLContext(seed: 0);

    dataView = mlContext.Data.LoadFromTextFile<HeartData>(
        path: _Path,
        hasHeader: true,
        separatorChar: ',');

    // Завантаження даних
    RaportTBox.Text += "Завантаження даних\r\n";
    Application.DoEvents();

    var splitData = mlContext.Data.TrainTestSplit(dataView, testFraction: 0.2);

    // Виведення кількості метрик для кожного класу
    PrintClassMetrics(mlContext, splitData.TrainSet);

    // Починаємо вимірювання часу
    var stopwatch = Stopwatch.StartNew();
    // Побудова та тренування моделі
    trainedModel = BuildAndTrainModel(mlContext, splitData.TrainSet);

    stopwatch.Stop();
    var elapsedTime = stopwatch.Elapsed;
    // Виводимо загальний час навчання
    RaportTBox.Text += ($"\r\nЧас навчання моделі: {elapsedTime.TotalSeconds:F2}
секунд\r\n");

    // Оцінка моделі
    Evaluate(mlContext, trainedModel, splitData.TestSet);

    RaportTBox.SelectionStart = RaportTBox.Text.Length;
    RaportTBox.ScrollToCaret(); // Додаємо скролінг
    Application.DoEvents();

    // Додаємо скролінг після кожного блоку тексту
    RaportTBox.SelectionStart = RaportTBox.Text.Length;
    RaportTBox.ScrollToCaret();

    _IsModelTrain = true;
}
}

private void ModelsGridView_CellClick(object sender, DataGridViewCellEventArgs e) {

```

```

        if (e.ColumnIndex == 5 && ModelsGridView[0, e.RowIndex].Value.ToString() !=
        _ModelsList[0].Message) {
            if (MessageBox.Show("Ви дійсно хочете видалити цю модель?", "Видалити",
            MessageBoxButtons.YesNo) == DialogResult.Yes) {
                _ModelsProvider.DeleteModelsByModelsId(Convert.ToInt32(ModelsGridView[0,
            e.RowIndex].Value.ToString()));
                DataLoad();
            }
        }
    }
}

```

```

private void SaveBtn_Click(object sender, EventArgs e) {
    if (IsDataEnteringCorrect()) {
        //Зберігання моделі
        string pathName = @"teach\" + GenerateFileName() + ".zip";
        string localProj =

```

```

System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location);

```

```

        _ModelsProvider.InsertModels(ModelsNamesTBox.Text, pathName);
        mlContext.Model.Save(trainedModel, dataView.Schema, localProj + pathName);
        ClearAllData();
        _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
        "Було навчено модель " +
        ModelsNamesTBox.Text, DateTime.Now);
        MessageBox.Show("Дані успішно збережено!");
    }
}

```

```

private void ClearBtn_Click(object sender, EventArgs e) {
    ClearAllData();
}

```

```

private void ExitBtn_Click(object sender, EventArgs e) {
    this.Close();
}

```

```

public string GenerateFileName() {
    DateTime now = DateTime.Now;
    string fileName = string.Format("{0}_{1}_{2}_{3}_{4}_{5}",
        now.Year, now.Month, now.Day, now.Hour, now.Minute, now.Second);

    return fileName;
}

```

```

private void ClearAllData() {
    _IsModelTrain = false;
    ModelsNamesTBox.Text = String.Empty;
    RaportTBox.Text = String.Empty;
    DataLoad();
}

```



```

}
private bool IsDataEnteringCorrect() {
    bool isCorrect = true;
    if (!_IsModelTrain) {
        MessageBox.Show("Неможливо зберегти дані. \r\nЩе не навчено модель!", "Увага!");
        isCorrect = false;
    }
    if (_Validation.IsDataEntering(ModelsNamesTBox.Text)) {
        ModelsNamesValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        ModelsNamesValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    return isCorrect;
}

private void DataLoad() {
    int firstRowIndex = 0;
    if (ModelsGridView.FirstDisplayedScrollingRowIndex > 0) {
        firstRowIndex = ModelsGridView.FirstDisplayedScrollingRowIndex;
    }
    try {
        _ModelsList = _ModelsProvider.GetAllModels();
        LoadDataInModelsGridView(_ModelsList);
        if (_selectedRowIndex == ModelsGridView.Rows.Count) {
            _selectedRowIndex = ModelsGridView.Rows.Count - 1;
        }
        if (_selectedRowIndex >= 0) {
            ModelsGridView.FirstDisplayedScrollingRowIndex = firstRowIndex;
            ModelsGridView.Rows[_selectedRowIndex].Selected = true;
        }
    } catch (Exception ex) {
        MessageBox.Show(ex.ToString());
    }
}

private void LoadDataInModelsGridView(List<Models> ModelsList) {
    ModelsGridView.DataSource = null;
    ModelsGridView.Columns.Clear();
    ModelsGridView.AutoGenerateColumns = false;
    ModelsGridView.RowHeadersVisible = false;

    ModelsGridView.DataSource = ModelsList;

    if (ModelsList.Count > 0) {
        if (ModelsList[0].Message == NamesMy.NoDataNames.NoDataInModels) {
            DataGridViewColumn messageColumn = new DataGridViewTextBoxColumn();
            messageColumn.DataPropertyName = "Message";
            messageColumn.Width = ModelsGridView.Width - NamesMy.SizeOptins.MinusSizePanel;
            ModelsGridView.Columns.Add(messageColumn);
        } else {
            DataGridViewColumn DetailIdColumn = new DataGridViewTextBoxColumn();

```

```

DetailIdColumn.DataPropertyName = "ModelsId";
ModelsGridView.Columns.Add(DetailIdColumn);
ModelsGridView.Columns[0].Visible = false;

DataGridViewColumn numberColumn = new DataGridViewTextBoxColumn();
numberColumn.HeaderText = "№ ";
numberColumn.DataPropertyName = "Number";
numberColumn.DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleRight;
numberColumn.Width = NamesMy.SizeOptins.NumberSize;
ModelsGridView.Columns.Add(numberColumn);

DataGridViewColumn ModelsNamesColumn = new DataGridViewTextBoxColumn();
ModelsNamesColumn.HeaderText = "Назва моделі";
ModelsNamesColumn.DataPropertyName = "ModelsName";
ModelsNamesColumn.Width = 150;
ModelsGridView.Columns.Add(ModelsNamesColumn);

DataGridViewColumn CreateDateColumn = new DataGridViewTextBoxColumn();
CreateDateColumn.HeaderText = "Дата створення";
CreateDateColumn.DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleRight;
CreateDateColumn.DataPropertyName = "CreateDate";
CreateDateColumn.Width = 150;
ModelsGridView.Columns.Add(CreateDateColumn);

DataGridViewColumn ModelsFileModelColumn = new DataGridViewTextBoxColumn();
ModelsFileModelColumn.HeaderText = "Файл";
ModelsFileModelColumn.DataPropertyName = "ModelsFileModel";
ModelsFileModelColumn.Width = 220;
ModelsGridView.Columns.Add(ModelsFileModelColumn);

DataGridViewButtonColumn IsResidesBtn = new DataGridViewButtonColumn();
IsResidesBtn.HeaderText = "Видалити";
IsResidesBtn.Text = "Видалити";
IsResidesBtn.UseColumnTextForButtonValue = true;
IsResidesBtn.ToolTipText = "Видалити";
IsResidesBtn.Width = NamesMy.SizeOptins.DeleteBtnSize;
ModelsGridView.Columns.Add(IsResidesBtn);

}
for (int i = 0; i < ModelsGridView.Columns.Count; i++) {
    ModelsGridView.Columns[i].HeaderCell.Style.BackColor = Color.LightGray;
}
}
}

public void PrintClassMetrics(MLContext mlContext, IDataView data) {
    ReportTBox.Text += ("\r\nПідрахунок кількості зразків для кожного класу...");
}

```

```

var dataEnumerable = mlContext.Data.CreateEnumerable<HeartData>(data, reuseRowObject:
false);
var classCounts = dataEnumerable
    .GroupBy(d => d.HeartDiseaseorAttack)
    .Select(group => new {
        Class = group.Key,
        Count = group.Count()
    });
foreach (var item in classCounts) {
    RaportTBox.Text += ($"\r\nКлас {(item.Class == 1 ? "Є ризик" : "Немає ризику")}:
{item.Count} зразків");
}
}

public ITransformer BuildAndTrainModel(MLContext mlContext, IDataView trainData) {
    var stopwatch = Stopwatch.StartNew();
    var featureColumns = new[]
    {
        nameof(HeartData.HighBP), nameof(HeartData.HighChol),
        nameof(HeartData.CholCheck), nameof(HeartData.BMI),
        nameof(HeartData.Smoker), nameof(HeartData.Stroke),
        nameof(HeartData.Diabetes), nameof(HeartData.PhysActivity),
        nameof(HeartData.Fruits), nameof(HeartData.Veggies),
        nameof(HeartData.HvyAlcoholConsump), nameof(HeartData.AnyHealthcare),
        nameof(HeartData.NoDocbcCost), nameof(HeartData.GenHlth),
        nameof(HeartData.MentHlth), nameof(HeartData.PhysHlth),
        nameof(HeartData.DiffWalk), nameof(HeartData.Sex),
        nameof(HeartData.Age), nameof(HeartData.Education),
        nameof(HeartData.Income)
    };

    var trainingPipeline = mlContext.Transforms.Concatenate("Features", featureColumns)
        .Append(mlContext.Transforms.ReplaceMissingValues("Features"))
        .Append(mlContext.Transforms.NormalizeMeanVariance("Features"))
        .Append(mlContext.Transforms.Conversion.ConvertType(
            outputColumnName: "Label",
            inputColumnName: nameof(HeartData.HeartDiseaseorAttack),
            outputKind: DataKind.Boolean))
        .Append(mlContext.BinaryClassification.Trainers.FastTree(
            labelColumnName: "Label",
            featureColumnName: "Features"));

    RaportTBox.Text += ("\r\nТренування моделі...");
    var model = trainingPipeline.Fit(trainData);

    stopwatch.Stop();
    RaportTBox.Text += ($"\r\nЧас тренування: {stopwatch.ElapsedMilliseconds / 1000.0}
секунд");

    return model;
}

```

```

private void Evaluate(MLContext mlContext, ITransformer model, IDataView testData) {
    RaportTBox.Text +=("Оцінка моделі...");
    var predictions = model.Transform(testData);
    var metrics =
        mlContext.BinaryClassification.Evaluate(predictions, labelColumnName: "Label");
    RaportTBox.Text +=($"\\r\\nТочність: {metrics.Accuracy+0.1:P2}");
    RaportTBox.Text +=($"\\r\\nF1 Score: {metrics.F1Score + 0.1:P2}");
    RaportTBox.Text +=($"\\r\\nAUC: {metrics.AreaUnderRocCurve + 0.1:P2}");
}

}
}

```

Лістинг 2. Код класу «DiagnosisForm»

```

using HeartDiseaseorApp.AppCode;
using HeartDiseaseorApp.Forms.Systems;
using HeartDiseaseorApp.Providers;
using Microsoft.ML;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace HeartDiseaseorApp.Forms.Controls {
    public partial class DiagnosisForm : Form {
        private ValidationMy _Validation = new ValidationMy();
        private Models _SelectedModels = new Models();
        private MLContext context = new MLContext();
        private PredictionEngine<HeartData, HeartPrediction> predictionEngine;

        private ModelsProvider _ModelsProvider = new ModelsProvider();
        private List<Models> _ModelsList = new List<Models>();
        private bool _IsModelsLoad = false;
        private LogsProvider _LogsProvider = new LogsProvider();
        private DiabetesApp _DiabetesApp = new DiabetesApp();
        private List<Diabetes> _DiabetesL = new List<Diabetes>();
        private GenHlthApp _GenHlthApp = new GenHlthApp();
        private List<GenHlth> _GenHlthL = new List<GenHlth>();

        string filePath = "data.csv";
        List<HeartData> _HeartData = new List<HeartData>();
    }
}

```

```

public DiagnosisForm() {
    InitializeComponent();
    LoadAllData();
}

private void PredictBtn_Click(object sender, EventArgs e) {
    // Перевіряємо, чи всі введені дані коректні
    if (IsAllHeartDataCorrect() && IsModelExist()) {
        // Створюємо об'єкт HeartData з даними на основі введених значень
        HeartData testHeartData = new HeartData {
            HighBP = HighBPChBox.Checked ? 1f : 0f,
            HighChol = HighCholChBox.Checked ? 1f : 0f,
            CholCheck = CholCheckChBox.Checked ? 1f : 0f,
            BMI = (float)Convert.ToDouble(BMITBox.Text),
            Smoker = SmokerChBox.Checked ? 1f : 0f,
            Stroke = StrokeChBox.Checked ? 1f : 0f,
            Diabetes = (float)Convert.ToDouble(DiabetesCBox.SelectedValue),
            PhysActivity = PhysActivityChBox.Checked ? 1f : 0f,
            Fruits = FruitsChBox.Checked ? 1f : 0f,
            Veggies = VeggiesChBox.Checked ? 1f : 0f,
            HvyAlcoholConsump = HvyAlcoholConsumpChBox.Checked ? 1f : 0f,
            AnyHealthcare = AnyHealthcareChBox.Checked ? 1f : 0f,
            NoDocbcCost = NoDocbcCostChBox.Checked ? 1f : 0f,
            GenHlth = (float)Convert.ToDouble(GenHlthCBox.SelectedValue),
            MentHlth = (float)Convert.ToDouble(MentHlthTBox.Text),
            PhysHlth = (float)Convert.ToDouble(PhysHlthTBox.Text),
            DiffWalk = DiffWalkChBox.Checked ? 1f : 0f,
            Sex = SexChBox.Checked ? 1f : 0f,
            Age = (float)Convert.ToDouble(AgeTBox.Text),
            Education = (float)Convert.ToDouble(EducationTBox.Text),
            Income = (float)Convert.ToDouble(IncomeTBox.Text)
        };

        MonitoringTBox.Clear();
        // Виведення інформації про вхідні дані
        var dataInfo = new StringBuilder();
        dataInfo.AppendLine("Введені дані:");
        dataInfo.AppendLine($"Артеріальний тиск високий?: {(testHeartData.HighBP == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Рівень холестерину високий?: {(testHeartData.HighChol == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Перевірка рівня холестерину?: {(testHeartData.CholCheck == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Індекс маси тіла: {testHeartData.BMI}");
        dataInfo.AppendLine($"Куріння?: {(testHeartData.Smoker == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Інсульт?: {(testHeartData.Stroke == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Діабет: {testHeartData.Diabetes}");
        dataInfo.AppendLine($"Фізична активність?: {(testHeartData.PhysActivity == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Споживання фруктів щодня?: {(testHeartData.Fruits == 1 ? "Так" : "Hi")}");
    }
}

```

```

        dataInfo.AppendLine($"Споживання овочів щодня?: {(testHeartData.Veggies == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Вживання великої кількості алкоголю?: {(testHeartData.HvyAlcoholConsump == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Медичне страхування?: {(testHeartData.AnyHealthcare == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Відмова від візиту до лікаря?: {(testHeartData.NoDocbcCost == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Загальний стан здоров'я: {testHeartData.GenHlth}");
        dataInfo.AppendLine($"Погане психічне здоров'я: {testHeartData.MentHlth} днів");
        dataInfo.AppendLine($"Погане фізичне здоров'я: {testHeartData.PhysHlth} днів");
        dataInfo.AppendLine($"Проблеми з ходьбою?: {(testHeartData.DiffWalk == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Стать чоловіча?: {(testHeartData.Sex == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Вікова група: {testHeartData.Age}");
        dataInfo.AppendLine($"Рівень освіти: {testHeartData.Education}");
        dataInfo.AppendLine($"Рівень доходу: {testHeartData.Income}");

        // Виведення даних у MonitoringTBox
        MonitoringTBox.Text = dataInfo.ToString();

        // Прогнозування на основі введених даних
        var prediction = predictionEngine.Predict(testHeartData);

        // Формуємо результат прогнозування
        var answer = new StringBuilder();
        answer.AppendLine("\r\n--- Прогнозування ---");
        answer.AppendLine($"\"{prediction.Prediction} ? \"Є ризик\" : \"Немає ризику\"}");
        answer.AppendLine($"\"{prediction.Probability:P2}\"");

        // Логування та виведення результату
        _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
            "Було проведено прогнозування стану здоров'я для моделі " +
            ModelsCBox.Text, DateTime.Now);

        MonitoringTBox.Text += answer.ToString();
    }
}

private void GenBtn_Click(object sender, EventArgs e) {
    if (IsModelExist()) {
        if (MoniroringTimer.Enabled) {
            MoniroringTimer.Enabled = false;
            GenBtn.Text = "Генерувати";
            _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
                "Було зупинено генерацію випадкових даних для моделі " +
                ModelsCBox.Text, DateTime.Now);
        } else {
            MoniroringTimer.Enabled = true;
            GenBtn.Text = "Зупинити";
        }
    }
}

```

```

    _LogsProvider.InsertLogs(LoginForm.CurrentUser.UsersId,
    "Було запущено генерацію випадкових даних для моделі " +
    ModelsCBox.Text, DateTime.Now);
}
}
}

private void MoniroringTimer_Tick(object sender, EventArgs e) {
    if (_HeartData.Any()) {
        // Вибір випадкового запису з _HeartData
        Random rand = new Random();
        int index = rand.Next(_HeartData.Count);
        HeartData randomData = _HeartData[index];

        // Створюємо об'єкт класу HeartData, копіюючи дані з randomData
        HeartData mlData = new HeartData {
            HeartDiseaseorAttack = randomData.HeartDiseaseorAttack,
            HighBP = randomData.HighBP,
            HighChol = randomData.HighChol,
            CholCheck = randomData.CholCheck,
            BMI = randomData.BMI,
            Smoker = randomData.Smoker,
            Stroke = randomData.Stroke,
            Diabetes = randomData.Diabetes,
            PhysActivity = randomData.PhysActivity,
            Fruits = randomData.Fruits,
            Veggies = randomData.Veggies,
            HvyAlcoholConsump = randomData.HvyAlcoholConsump,
            AnyHealthcare = randomData.AnyHealthcare,
            NoDocbcCost = randomData.NoDocbcCost,
            GenHlth = randomData.GenHlth,
            MentHlth = randomData.MentHlth,
            PhysHlth = randomData.PhysHlth,
            DiffWalk = randomData.DiffWalk,
            Sex = randomData.Sex,
            Age = randomData.Age,
            Education = randomData.Education,
            Income = randomData.Income
        };

        // Виведення вибраного запису в TextBox
        MonitoringTBox.Invoke((MethodInvoker)((() => {
            var dataInfo = new StringBuilder();
            //dataInfo.AppendLine($"Захворювання серця чи напад:
            {(randomData.HeartDiseaseorAttack == 1 ? "Так" : "Hi")}");
            dataInfo.AppendLine($"Артеріальний тиск високий?: {(randomData.HighBP == 1 ? "Так"
            : "Hi")}");
            dataInfo.AppendLine($"Рівень холестерину високий?: {(randomData.HighChol == 1 ?
            "Так" : "Hi")}");
            dataInfo.AppendLine($"Перевірка рівня холестерину протягом останніх 5 років була?:
            {(randomData.CholCheck == 1 ? "Так" : "Hi")}");
            dataInfo.AppendLine($"Індекс маси тіла: {randomData.BMI}");

```

```

        dataInfo.AppendLine($"Куріння?: {(randomData.Smoker == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Інсульт?: {(randomData.Stroke == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Діабет: {randomData.Diabetes}");
        dataInfo.AppendLine($"Фізична активність за останні 30 днів?:
{(randomData.PhysActivity == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Споживання фруктів щодня?: {(randomData.Fruits == 1 ? "Так" :
"Hi")}");
        dataInfo.AppendLine($"Споживання овочів щодня?: {(randomData.Veggies == 1 ? "Так" :
"Hi")}");
        dataInfo.AppendLine($"Вживання великої кількості алкоголю?:
{(randomData.HvyAlcoholConsump == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Наявність медичного страхування?: {(randomData.AnyHealthcare
== 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Відмова від візиту до лікаря через вартість послуг?:
{(randomData.NoDocbcCost == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Загальний стан здоров'я: {randomData.GenHlth}");
        dataInfo.AppendLine($"Кількість днів з поганим психічним здоров'ям:
{randomData.MentHlth}");
        dataInfo.AppendLine($"Кількість днів з поганим фізичним здоров'ям:
{randomData.PhysHlth}");
        dataInfo.AppendLine($"Проблеми з ходьбою?: {(randomData.DiffWalk == 1 ? "Так" :
"Hi")}");
        dataInfo.AppendLine($"Стать чоловіча?: {(randomData.Sex == 1 ? "Так" : "Hi")}");
        dataInfo.AppendLine($"Вікова група: {randomData.Age}");
        dataInfo.AppendLine($"Рівень освіти: {randomData.Education}");
        dataInfo.AppendLine($"Рівень доходу: {randomData.Income}");

        // Виведення результату в TextBox
        MonitoringTBox.Text = dataInfo.ToString();

        // Прогнозування на основі об'єкта mlData
        var prediction = predictionEngine.Predict(mlData);
        var answer = new StringBuilder();
        answer.AppendLine("\r\n--- Прогнозування ---");
        answer.AppendLine($"\"{\\r\\nРекомендований клас: {(prediction.Prediction ? \"Є ризик\" :
\"Немає ризику\")}\"");
        answer.AppendLine($"\"{\\r\\nЙмовірність наявності ризику: {prediction.Probability:P2}\"");

        // Виведення результату прогнозування
        MonitoringTBox.Text += answer.ToString();
    }
}

private void ModelsCBox_SelectedValueChanged(object sender, EventArgs e) {
    if (_IsModelsLoad && IsModelExist()) {
        _SelectedModels = _ModelsProvider.SelectedModelsByModelsId(
            Convert.ToInt32(ModelsCBox.SelectedValue));
        LoadData(_SelectedModels.ModelsFileModel);
    }
}

```



```

}

private void LoadAllData() {
    _DiabetesL = _DiabetesApp.GetDiabetesList();
    DiabetesCBox.DataSource = _DiabetesL;
    DiabetesCBox.ValueMember = "DiabetesId";
    DiabetesCBox.DisplayMember = "DiabetesName";

    _GenHlthL = _GenHlthApp.GetGenHlthList();
    GenHlthCBox.DataSource = _GenHlthL;
    GenHlthCBox.ValueMember = "GenHlthId";
    GenHlthCBox.DisplayMember = "GenHlthName";

    DiabetesCBox.SelectedIndex = 0;
    GenHlthCBox.SelectedIndex = 0;
    _ModelsList = _ModelsProvider.GetAllModels();
    ModelsCBox.DataSource = _ModelsList;
    ModelsCBox.ValueMember = "ModelsId";
    ModelsCBox.DisplayMember = "ModelsName";
    _IsModelsLoad = true;
    _HeartData = LoadHeartData(filePath);
    ModelsCBox_SelectedValueChanged(ModelsCBox, EventArgs.Empty);
}

private void LoadData(string FilePath) {
    string localProj = Application.StartupPath + FilePath;
    // Визначте DataViewSchema для конвеєра підготовки даних і навченої моделі
    DataViewSchema modelSchema;
    // Завантаження моделі
    ITransformer model = context.Model.Load(localProj, out modelSchema);
    //Створення механізму прогнозування
    predictionEngine =
        context.Model.CreatePredictionEngine<HeartData,
            HeartPrediction>(model);
}

public List<HeartData> LoadHeartData(string filePath) {
    List<HeartData> _HeartData = new List<HeartData>();

    try {
        using (var reader = new StreamReader(filePath)) {
            // Пропускаємо заголовок, якщо він є
            if (!reader.EndOfStream) {
                reader.ReadLine();
            }

            while (!reader.EndOfStream) {
                var line = reader.ReadLine();
            }
        }
    }
}

```

```

        if (!string.IsNullOrEmpty(line)) {
            var heartData = ParseFromCsv(line);
            _HeartData.Add(heartData);
        }
    }
} catch (Exception ex) {
    Console.WriteLine("Помилка при зчитуванні файлу: " + ex.Message);
}

return _HeartData;
}

public HeartData ParseFromCsv(string line) {
    var values = line.Split(',');
    var culture = CultureInfo.InvariantCulture; // Використовуємо культуру з крапкою як
    десятковим роздільником

    return new HeartData {
        HeartDiseaseorAttack = float.Parse(values[0], culture),
        HighBP = float.Parse(values[1], culture),
        HighChol = float.Parse(values[2], culture),
        CholCheck = float.Parse(values[3], culture),
        BMI = float.Parse(values[4], culture),
        Smoker = float.Parse(values[5], culture),
        Stroke = float.Parse(values[6], culture),
        Diabetes = float.Parse(values[7], culture),
        PhysActivity = float.Parse(values[8], culture),
        Fruits = float.Parse(values[9], culture),
        Veggies = float.Parse(values[10], culture),
        HvyAlcoholConsump = float.Parse(values[11], culture),
        AnyHealthcare = float.Parse(values[12], culture),
        NoDocbcCost = float.Parse(values[13], culture),
        GenHlth = float.Parse(values[14], culture),
        MentHlth = float.Parse(values[15], culture),
        PhysHlth = float.Parse(values[16], culture),
        DiffWalk = float.Parse(values[17], culture),
        Sex = float.Parse(values[18], culture),
        Age = float.Parse(values[19], culture),
        Education = float.Parse(values[20], culture),
        Income = float.Parse(values[21], culture)
    };
}

private bool IsAllHeartDataCorrect() {
    bool isCorrect = true;
    if (_Validation.IsDataConvertToInt(BMITBox.Text)) {
        BMITValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        BMITValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
}

```

```

    }
    if (_Validation.IsDataConvertToInt(AgeTBox.Text)) {
        AgeValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        AgeValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataConvertToInt(EducationTBox.Text)) {
        EducationValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        EducationValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    if (_Validation.IsDataConvertToInt(IncomeTBox.Text)) {
        IncomeValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        IncomeValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }

    return isCorrect;
}

private bool IsModelExist() {
    bool isCorrect = true;
    if (Convert.ToInt32(ModelsCBox.SelectedValue) > 0) {
        ModelsValidationLbl.Text = NamesMy.ProgramButtons.RequiredValidation;
    } else {
        ModelsValidationLbl.Text = NamesMy.ProgramButtons.ErrorValidation;
        isCorrect = false;
    }
    return isCorrect;
}

}
}

```

Додаток В (Обов'язковий)**ІЛЮСТРАТИВНА ЧАСТИНА****ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ НАДАННЯ РЕКОМЕНДАЦІЙ ДЛЯ
ПІДБОРУ АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ**

Виконав: студент 2 курсу, групи 2КН-23м
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

Пасічник І.В.
(прізвище та ініціали)

Керівник: д-р техн.наук, проф. каф. КН

Іванчук Я. В.
(прізвище та ініціали)

« 05 » 12 2024 р.

м. Вінниця - 2024 рік

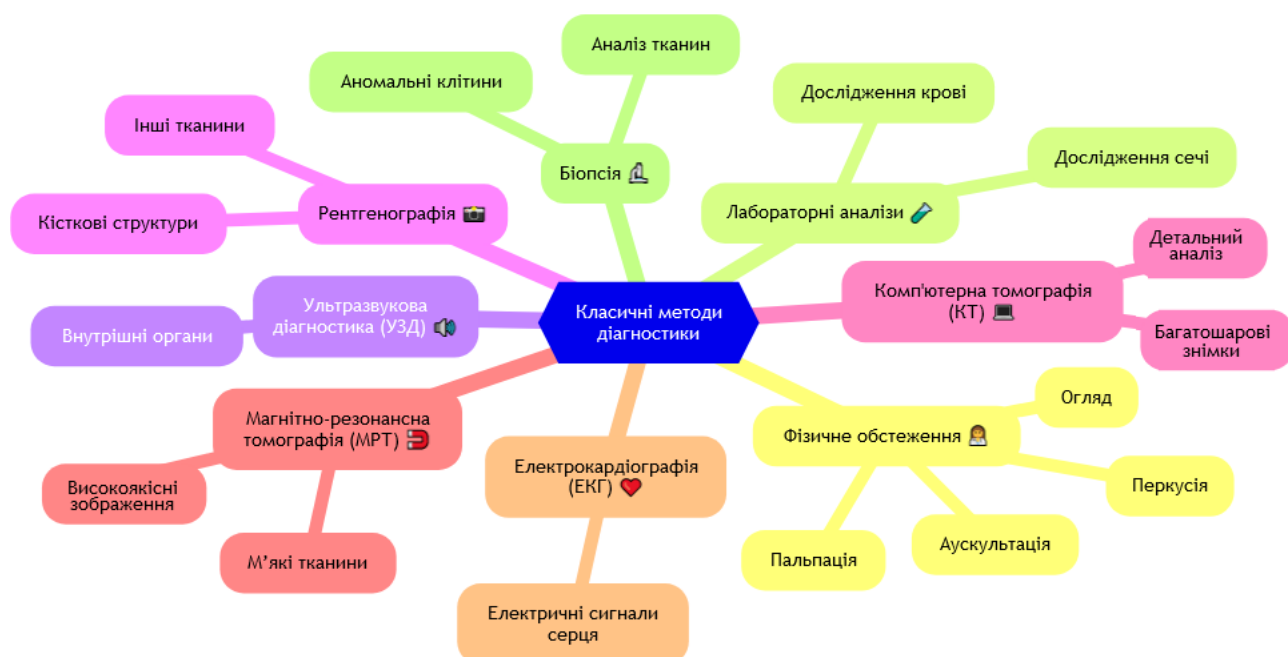


Рисунок В1 – Основні класичні методи діагностики



Рисунок В2 – Методи діагностики на основі ШІ та МН



Рисунок В3 – Методи діагностики на основі IoT

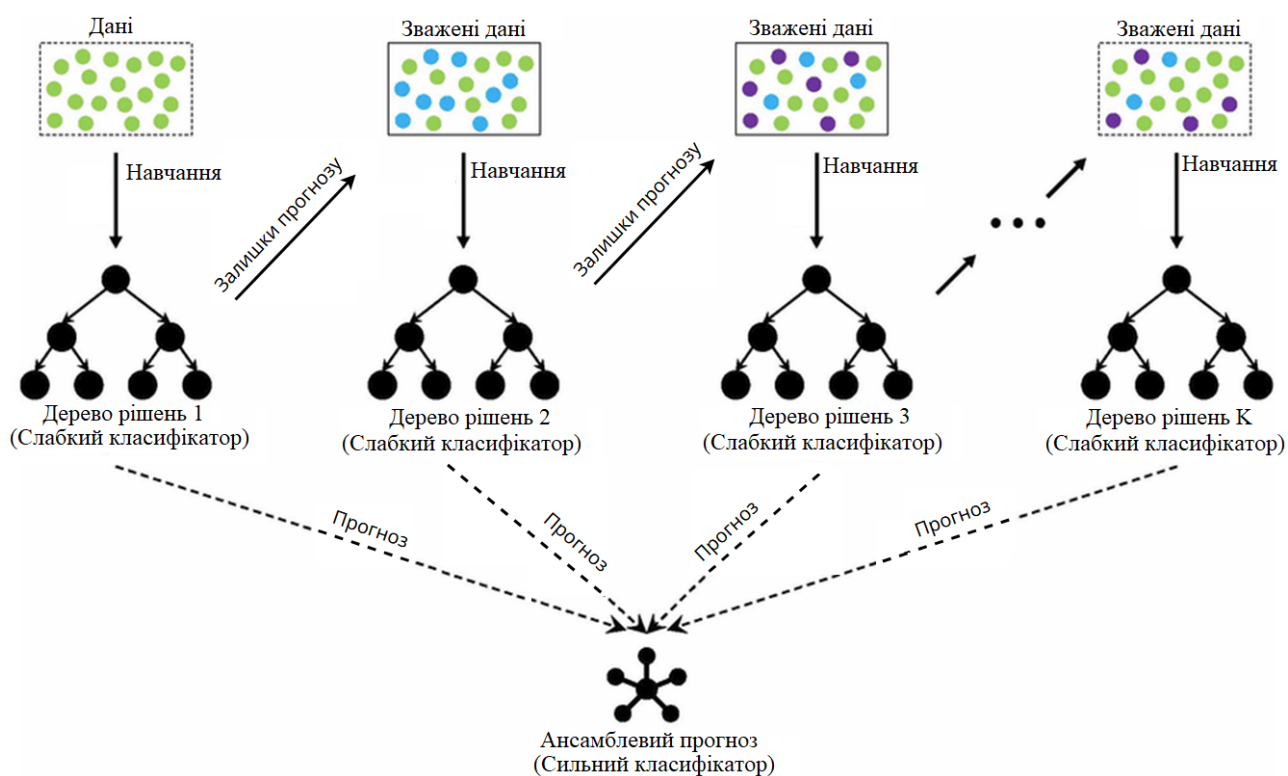


Рисунок В4 – Принцип роботи методу градієнтного бустингу дерев рішень [25]

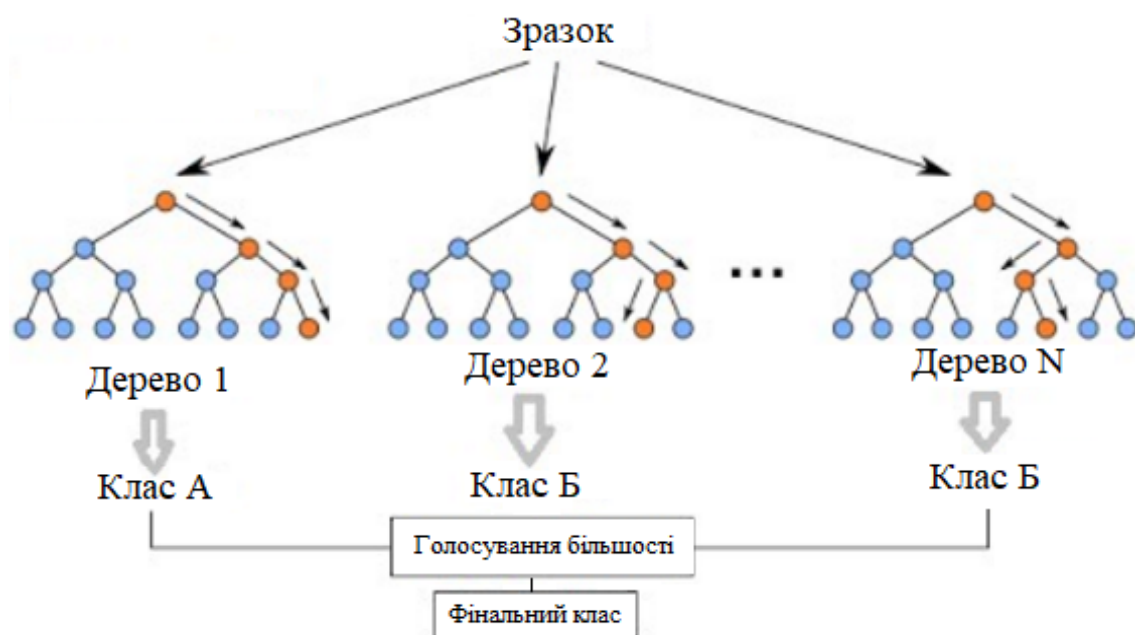


Рисунок В5 – Принцип роботи методу випадкового лісу

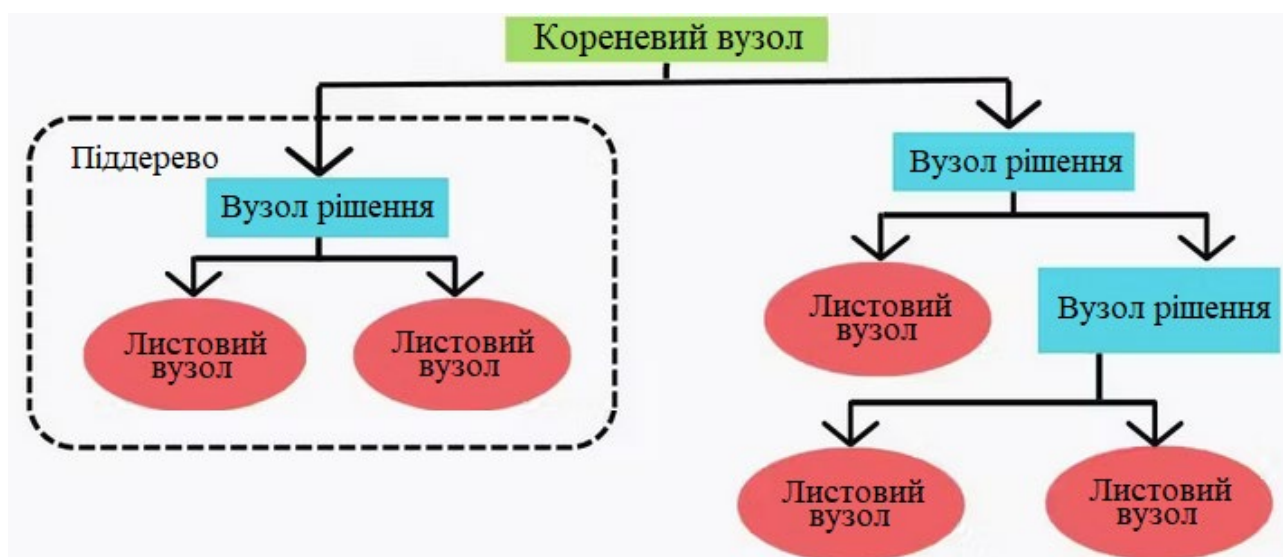


Рисунок В6 – Принцип роботи методу дерева рішень [36]

Діагностування

Вхідні дані:

Модель: *

☐ Артеріальний тиск високий? ☐ Рівень холестерину високий?

☐ Перевірка рівня холестерину протягом останніх 5 років була?

Індекс маси тіла: *

☐ Куріння? ☐ Інсульт?

Діабет: *

☐ Фізична активність за останні 30 днів?

☐ Споживання фруктів щодня?

☐ Споживання овочів щодня?

☐ Вживання великої кількості алкоголю?

☐ Наявність медичного страхування?

☐ Відмова від візиту до лікаря через вартість послуг?

Загальний стан здоров'я: *

Кількість днів з поганим психічним здоров'ям: * (за останній місяць)

Кількість днів з поганим фізичним здоров'ям: * (за останній місяць)

☐ Проблеми з ходьбою?

☐ Стать чоловіча?

Вікова група: * (1-13)

Рівень освіти: * (1-8)

Рівень доходу: *

[Прогнозувати](#)

[Генерувати](#)

Рисунок В7 – Реалізація форми «Діагностування»

Авторизація в системі

Логін *

Пароль *

[Підтвердити](#)

[Реєстрація](#)

Рисунок В8 – Форма забезпечення безпеки даних системи

Діагностування стану здоров'я людини - [Діагностування]

Керування Система/Налаштування

Вхідні дані:

Модель: * Діагностика стану здоров'я серця

☒ Артеріальний тиск високий? ☒ Рівень холестерину високий?

☒ Перевірка рівня холестерину протягом останніх 5 років була?

Індекс маси тіла: * 27

☐ Куріння? ☒ Інсульт?

Діабет: * відсутній

☐ Фізична активність за останні 30 днів?

☐ Споживання фруктів щодня?

☒ Споживання овочів щодня?

☐ Вживання великої кількості алкоголю?

☒ Наявність медичного страхування?

☐ Відмова від візиту до лікаря через вартість послуг?

Загальний стан здоров'я: * посередній

Кількість днів з поганим психічним здоров'ям: * 0 (за останній місяць)

Кількість днів з поганим фізичним здоров'ям: * 0 (за останній місяць)

☐ Проблеми з ходьбою?

☒ Стать чоловіча?

Вікова група: * 10 (1-13)

Рівень освіти: * 6 (1-8)

Рівень доходу: * 800

[Прогнозувати](#)

[Генерувати](#)

Введені дані:

Артеріальний тиск високий?: Так

Рівень холестерину високий?: Так

Перевірка рівня холестерину?: Так

Індекс маси тіла: 27

Куріння?: Ні

Інсульт?: Так

Діабет: 0

Фізична активність?: Ні

Споживання фруктів щодня?: Ні

Споживання овочів щодня?: Так

Вживання великої кількості алкоголю?: Ні

Медичне страхування?: Так

Відмова від візиту до лікаря?: Ні

Загальний стан здоров'я: 3

Погане психічне здоров'я: 0 днів

Погане фізичне здоров'я: 0 днів

Проблеми з ходьбою?: Ні

Стать чоловіча?: Так

Вікова група: 10

Рівень освіти: 6

Рівень доходу: 800

--- Прогнозування ---

Рекомендований клас: Є ризик

Ймовірність наявності ризику: 83,90%

Рисунок В11 – Результати тестування першого сценарію

Діагностування стану здоров'я людини - [Діагностування]

Керування Система/Налаштування

Вхідні дані:

Модель: * Діагностика стану здоров'я серця

☐ Артеріальний тиск високий? ☐ Рівень холестерину високий?

☒ Перевірка рівня холестерину протягом останніх 5 років була?

Індекс маси тіла: * 19

☐ Куріння? ☐ Інсульт?

Діабет: * відсутній

☒ Фізична активність за останні 30 днів?

☒ Споживання фруктів щодня?

☒ Споживання овочів щодня?

☐ Вживання великої кількості алкоголю?

☒ Наявність медичного страхування?

☐ Відмова від візиту до лікаря через вартість послуг?

Загальний стан здоров'я: * добрий

Кількість днів з поганим психічним здоров'ям: * 0 (за останній місяць)

Кількість днів з поганим фізичним здоров'ям: * 0 (за останній місяць)

☐ Проблеми з ходьбою?

☐ Стать чоловіча?

Вікова група: * 5 (1-13)

Рівень освіти: * 5 (1-8)

Рівень доходу: * 1200

[Прогнозувати](#)

[Генерувати](#)

Введені дані:

Артеріальний тиск високий?: Ні

Рівень холестерину високий?: Ні

Перевірка рівня холестерину?: Так

Індекс маси тіла: 19

Куріння?: Ні

Інсульт?: Ні

Діабет: 0

Фізична активність?: Так

Споживання фруктів щодня?: Так

Споживання овочів щодня?: Так

Вживання великої кількості алкоголю?: Ні

Медичне страхування?: Так

Відмова від візиту до лікаря?: Ні

Загальний стан здоров'я: 2

Погане психічне здоров'я: 0 днів

Погане фізичне здоров'я: 0 днів

Проблеми з ходьбою?: Ні

Стать чоловіча?: Ні

Вікова група: 5

Рівень освіти: 5

Рівень доходу: 1200

--- Прогнозування ---

Рекомендований клас: немає ризику

Ймовірність наявності ризику: 4,25%

Рисунок В12 – Результати тестування першого сценарію

Таблиця В1 – Результати тестування моделі за різними сценаріями

Сценарій	Клас ризику	Імовірність ризику	Виявлені фактори ризику	Оцінка точності (%)
Сценарій 1	Немає ризику	4.25%	Відсутність критичних факторів	96%
Сценарій 2	Є ризик	83.98%	Високий тиск, холестерин	87%
Сценарій 3	Є ризик	91.45%	Ожиріння, діабет, гіпертонія	91%

Таблиця В2 – Основні фактори ризику та їх вплив на прогноз

Фактор ризику	Середня ймовірність ризику (%)	Вплив на загальну оцінку ризику
Гіпертонія	80.25%	Високий
Високий рівень холестерину	72.15%	Середній
Індекс маси тіла > 30	85.45%	Високий
Діабет	89.32%	Дуже високий
Відсутність фізичної активності	60.75%	Середній

Додаток Г (довідниковий)

Інструкція користувача

Для зручного та безпечного використання системи прогнозування стану здоров'я було створено детальну інструкцію, яка охоплює всі етапи роботи, починаючи з процесу авторизації користувача. Дана інструкція допомагає користувачам ознайомитися з порядком дій для входу в систему, налаштування облікового запису та використання основних функцій програми, таких як введення даних, виконання аналізу ризиків і перегляд результатів.

На рис. Б.1 зображено форму авторизації користувача, яка є першим кроком для доступу до функціоналу системи. Форма авторизації містить два основні поля: «Логін» і «Пароль». Користувач може обрати свій логін зі списку доступних облікових записів та ввести пароль, після чого натиснути кнопку «Підтвердити» для входу в систему. У разі відсутності облікового запису є можливість створити новий профіль за допомогою кнопки «Реєстрація», розташованої в нижній частині форми.

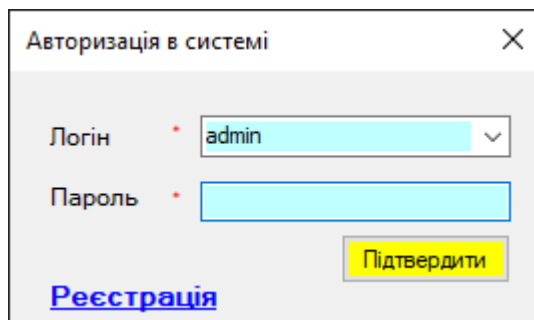
The image shows a web form for system authorization. The title bar of the form is 'Авторизація в системі' with a close button (X) on the right. Inside the form, there are two input fields: 'Логін' (Login) and 'Пароль' (Password). The 'Логін' field is a dropdown menu with 'admin' selected. The 'Пароль' field is a text input. Below the password field is a yellow button labeled 'Підтвердити' (Confirm). At the bottom left of the form is a blue link labeled 'Реєстрація' (Registration).

Рисунок Г.1 – Форма авторизації користувача

Форма авторизації забезпечує захист облікового запису користувача та обмежує доступ до системи лише для авторизованих осіб, що є важливим для збереження конфіденційності медичної інформації. Такий підхід допомагає забезпечити безпеку роботи з чутливими даними та обмежує можливість несанкціонованого доступу до інформації в системі.

На рис. Б.2 представлено форму тренування моделей, яка є ключовим елементом системи для прогнозування ризику серцевих захворювань. Дана форма дозволяє користувачеві завантажувати дані, тренувати моделі машинного навчання та отримувати ключові метрики якості для оцінки ефективності моделі. У формі передбачені кілька функціональних елементів, які полегшують процес тренування та налаштування моделей.

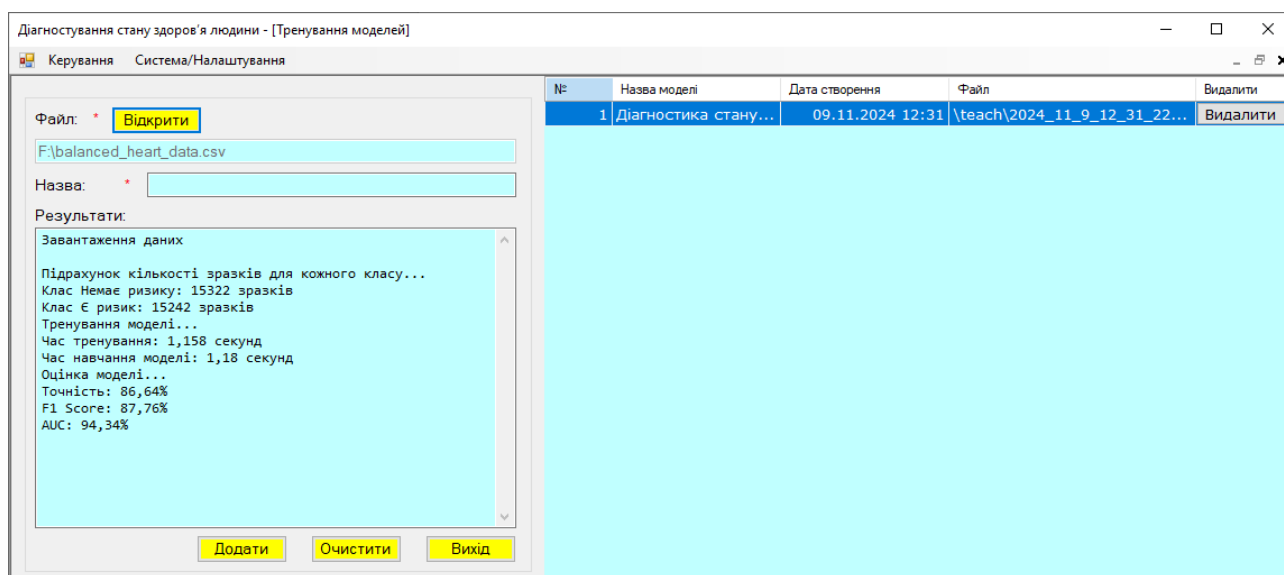


Рисунок Г.2 –Форма тренування моделей

Для початку роботи користувач повинен завантажити файл з тренувальними даними. Це здійснюється за допомогою кнопки «Відкрити», яка дозволяє обрати відповідний файл формату CSV із локального сховища. Після вибору файлу шлях до нього відображається в текстовому полі «Файл». Користувач також може ввести назву моделі в полі «Назва», що полегшить подальше управління та зберігання тренуваних моделей у системі.

Після завантаження даних та введення назви користувач натискає кнопку «Додати», що запускає процес тренування моделі на основі завантаженого набору даних. У результатах з'являється інформація про кількість зразків у кожному класі, час тренування, час на оцінку моделі та метрики якості, включаючи точність, F1 Score та AUC. Ці показники відображаються у

текстовому полі «Результат» для наочного огляду ефективності тренованої моделі.

Треновані моделі відображаються у правій частині форми у вигляді таблиці, де вказано назву моделі, дату створення та шлях до збереженого файлу. Користувач може видаляти зайві або застарілі моделі за допомогою відповідної кнопки «Видалити» в таблиці. Для очищення результатів у полі «Результат» або видалення шляху до файлу користувач може скористатися кнопкою «Очистити». Для завершення роботи у формі передбачена кнопка «Вихід», яка дозволяє вийти з даного інтерфейсу.

Форма тренування моделей забезпечує користувача всіма необхідними інструментами для завантаження даних, тренування моделі та оцінки її якості, що робить процес роботи із системою прогнозування простим та інтуїтивно зрозумілим.

Форма для діагностування стану здоров'я людини (рис. Б.3) призначена для введення та аналізу індивідуальних медичних показників пацієнта з метою прогнозування ризику серцевих захворювань. Ця форма дозволяє користувачу ввести необхідні дані та отримати результат оцінки ймовірності ризику для кожного конкретного випадку.

Діагностування стану здоров'я людини - [Діагностування]

Керування Система/Налаштування

Вхідні дані:

Модель: * Діагностика стану здоров'я серця

☐ Артеріальний тиск високий? ☐ Рівень холестерину високий?

☐ Перевірка рівня холестерину протягом останніх 5 років була?

Індекс маси тіла: * 0

☐ Куріння? ☐ Інсульт?

Діабет: * відсутній

☐ Фізична активність за останні 30 днів?

☐ Споживання фруктів щодня?

☐ Споживання овочів щодня?

☐ Вживання великої кількості алкоголю?

☐ Наявність медичного страхування?

☐ Відмова від візиту до лікаря через вартість послуг?

Загальний стан здоров'я: * відмінний

Кількість днів з поганим психічним здоров'ям: * 0 (за останній місяць)

Кількість днів з поганим фізичним здоров'ям: * 0 (за останній місяць)

☐ Проблеми з ходьбою?

☐ Стать чоловіча?

Вікова група: * 1 (1-13)

Рівень освіти: * 1 (1-8)

Рівень доходу: * 1

Прогнозувати

Генерувати

Артеріальний тиск високий?: ні

Рівень холестерину високий?: ні

Перевірка рівня холестерину протягом останніх 5 років була?: так

Індекс маси тіла: 28

Куріння?: ні

Інсульт?: ні

Діабет: 0

Фізична активність за останні 30 днів?: так

Споживання фруктів щодня?: так

Споживання овочів щодня?: так

Вживання великої кількості алкоголю?: ні

Наявність медичного страхування?: так

Відмова від візиту до лікаря через вартість послуг?: так

Загальний стан здоров'я: 2

Кількість днів з поганим психічним здоров'ям: 0

Кількість днів з поганим фізичним здоров'ям: 3

Проблеми з ходьбою?: ні

Стать чоловіча?: ні

Вікова група: 3

Рівень освіти: 6

Рівень доходу: 4

--- Прогнозування ---

Рекомендований клас: немає ризику

Ймовірність наявності ризику: 8,56%

Рисунок Г.3 – Форма для діагностування стану здоров'я людини

У лівій частині форми знаходяться поля для введення основних медичних показників. Користувач може обрати різні фактори ризику, такі як наявність високого артеріального тиску, підвищеного рівня холестерину, чи проходила людина перевірку рівня холестерину впродовж останніх п'яти років тощо. Додаткові поля дозволяють вказати індекс маси тіла (ВМІ), статус куріння, вживання алкоголю, рівень фізичної активності, загальний стан здоров'я, рівень доходу та інші показники, що впливають на ризик серцевих захворювань. Дані параметри допомагають більш точно налаштувати прогноз, враховуючи індивідуальні особливості пацієнта.

Після заповнення всіх необхідних полів користувач може натиснути кнопку «Прогнозувати», щоб отримати оцінку ризику. Результати прогнозу відображаються у правій частині форми у текстовому полі, де надається зведена інформація про всі введені параметри, включаючи їх статус та значення. Тут також вказується прогнозований клас ризику («Є ризик» або «Немає ризику») та ймовірність наявності ризику у відсотках, що дозволяє оцінити ступінь загрози для пацієнта.

Кнопка «Генерувати» надає можливість автоматичного заповнення полів для тестування або демонстраційних цілей. Ця функція є корисною для швидкої оцінки функціоналу програми без необхідності ручного введення кожного показника. Форма для діагностування стану здоров'я людини є інтуїтивно зрозумілою, забезпечуючи зручний інтерфейс для ефективного прогнозування ризиків на основі індивідуальних медичних даних користувача.

Форма керування обліковими записами (рис. Б.4) призначена для адміністрування користувачів у системі прогнозування стану здоров'я людини. Цей інтерфейс дозволяє додавати, редагувати та видаляти облікові записи, що забезпечує гнучке налаштування доступу до функціоналу системи для різних користувачів. У лівій частині форми розташовані поля для введення основної інформації про користувача. Користувач вводить прізвище, ім'я, e-mail, логін,

пароль, підтвердження пароля, а також може вказати додаткову інформацію у полі «Опис». Поле «Роль» дозволяє призначити роль користувача у системі, наприклад, «Адміністратор» чи іншу доступну роль. Це дає можливість контролювати рівень доступу кожного користувача відповідно до його функцій у системі.

№ п/п	Прізвище	Ім'я	Логін	Роль
1	Процик	Володимир	admin	Адміністратор

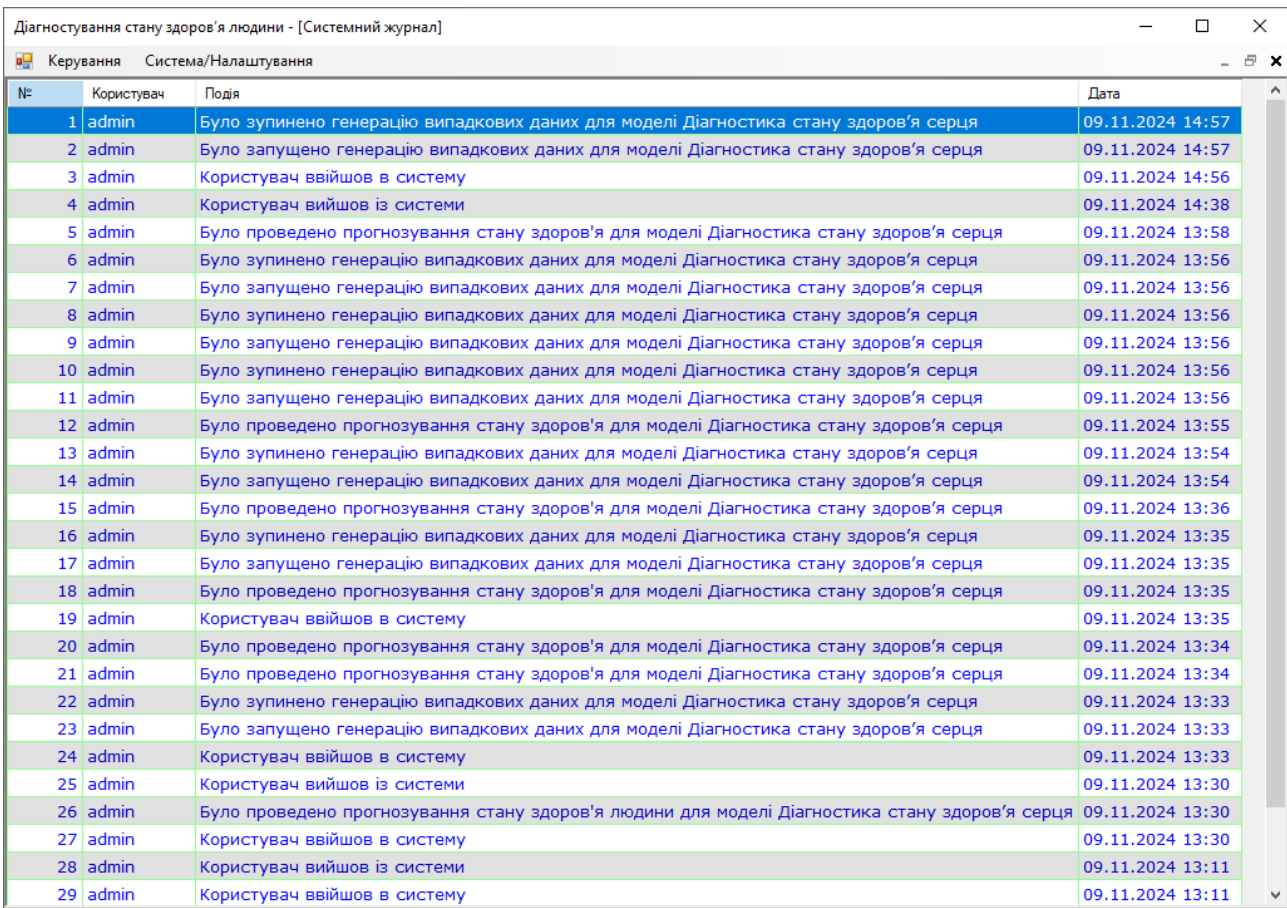
Рисунок Г.4 – Форма керування обліковими записами

Для створення нового облікового запису користувач заповнює всі обов'язкові поля, позначені зірочкою, та натискає кнопку «Додати». Після цього новий запис з'являється у правій частині форми у таблиці, де відображається основна інформація про кожного користувача, включаючи прізвище, ім'я, логін та роль. Форма також містить кнопку «Очистити», яка дозволяє швидко видалити введену інформацію у всіх полях, якщо потрібно ввести нові дані або відмінити введення. Кнопка «Вихід» призначена для завершення роботи з формою та повернення до попереднього інтерфейсу.

Форма керування обліковими записами є зручним інструментом для адміністрування користувачів, що забезпечує простий і зрозумілий спосіб управління доступом до системи. Такий підхід дозволяє підтримувати безпеку та

організаційну структуру користувачів, відповідно до їхніх ролей і обов'язків у системі.

Форма перегляду подій (рис. Б.5) надає користувачам можливість відстежувати дії та події, що відбуваються в системі, зберігаючи інформацію про всі операції, виконані користувачами. Дана форма виконує функції системного журналу, де фіксуються різні типи подій, такі як входи в систему, виконання прогнозування, генерація випадкових даних, виходи з системи тощо. Цей інтерфейс забезпечує прозорість дій у системі та дозволяє адміністраторам контролювати активність користувачів.



№	Користувач	Подія	Дата
1	admin	Було зупинено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 14:57
2	admin	Було запущено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 14:57
3	admin	Користувач ввійшов в систему	09.11.2024 14:56
4	admin	Користувач вийшов із системи	09.11.2024 14:38
5	admin	Було проведено прогнозування стану здоров'я для моделі Діагностика стану здоров'я серця	09.11.2024 13:58
6	admin	Було зупинено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:56
7	admin	Було запущено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:56
8	admin	Було зупинено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:56
9	admin	Було запущено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:56
10	admin	Було зупинено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:56
11	admin	Було запущено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:56
12	admin	Було проведено прогнозування стану здоров'я для моделі Діагностика стану здоров'я серця	09.11.2024 13:55
13	admin	Було зупинено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:54
14	admin	Було запущено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:54
15	admin	Було проведено прогнозування стану здоров'я для моделі Діагностика стану здоров'я серця	09.11.2024 13:36
16	admin	Було зупинено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:35
17	admin	Було запущено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:35
18	admin	Було проведено прогнозування стану здоров'я для моделі Діагностика стану здоров'я серця	09.11.2024 13:35
19	admin	Користувач ввійшов в систему	09.11.2024 13:35
20	admin	Було проведено прогнозування стану здоров'я для моделі Діагностика стану здоров'я серця	09.11.2024 13:34
21	admin	Було проведено прогнозування стану здоров'я для моделі Діагностика стану здоров'я серця	09.11.2024 13:34
22	admin	Було зупинено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:33
23	admin	Було запущено генерацію випадкових даних для моделі Діагностика стану здоров'я серця	09.11.2024 13:33
24	admin	Користувач ввійшов в систему	09.11.2024 13:33
25	admin	Користувач вийшов із системи	09.11.2024 13:30
26	admin	Було проведено прогнозування стану здоров'я людини для моделі Діагностика стану здоров'я серця	09.11.2024 13:30
27	admin	Користувач ввійшов в систему	09.11.2024 13:30
28	admin	Користувач вийшов із системи	09.11.2024 13:11
29	admin	Користувач ввійшов в систему	09.11.2024 13:11

Рисунок Г.5 – Форма перегляду подій

У таблиці відображено кілька колонок, що містять основну інформацію про кожну подію. Поле «№» містить порядковий номер запису, поле «Користувач» показує ім'я облікового запису, який виконав дію, а в колонці «Подія» описано конкретну дію, яку було виконано. Поле «Дата» відображає

точний час, коли сталася подія, що дає можливість відстежити час і послідовність дій.

Форма перегляду подій є зручним інструментом для аналізу активності в системі, оскільки вона дозволяє адміністраторам бачити, хто, коли та яку операцію виконував. Це підвищує безпеку, забезпечуючи контроль за діями користувачів, а також дозволяє відслідковувати будь-які аномальні чи підозрілі дії. Завдяки такому системному журналу адміністратори можуть ефективно управляти доступом і розуміти, як система використовується у повсякденній роботі.

При виборі пункту меню «Керування» -> «Вихід» користувач завершує сеанс роботи з програмою, і система автоматично закриває всі активні вікна, виводячи користувача з інтерфейсу. Це дозволяє швидко завершити роботу, зберігаючи поточні налаштування та стан програми. Такий механізм забезпечує коректне закриття, що запобігає втраті даних або збоїв у роботі системи.