

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ОБЛІКУ ТА АНАЛІЗУ МЕДИЧНИХ
ДАНИХ»**

Рівень «бакалавр»

Галузь знань 12 «Інформаційні технології»,

спеціальність 122 «Комп'ютерні науки»

Виконав здобувач 4 курсу

Єноктаєв Олександр Олександрович

Керівник: к.пед.н., доцент

Логінова Наталія Іванівна

Рецензент: к.пед.н., доцент

Лобода Юлія Геннадіївна

Одеса – 2024

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
Факультет кібербезпеки та інформаційних технологій
Кафедра інформаційних технологій
Галузь знань: 12 Інформаційні технології
Спеціальність: 122 Комп'ютерні науки
Назва освітньої програми: Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційних технологій
доцент Н.І. Логінова

_____ «____» _____ 20__ року

З А В Д А Н Н Я

**на кваліфікаційну (бакалаврську) роботу
здобувачу Єноктаєву Олександр Олександровичу**

- Тема роботи: «Інформаційна система для обліку та аналізу медичних даних»
Керівник роботи: к.пед.н, доцент Наталія Іванівна Логінова
затверджені наказом НУ ОЮА від «02» квітня 2024 року № 809-06
- Строк подання здобувачем роботи «20» травня 2024 рік
- Дата видачі завдання «15» вересня 2023 рік

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	02.10.2023	
2	Проектування веб-застосунку	04.12.2023	
3	Вибір програмних засобів для реалізації поставлених задач	28.01.2024	
4	Розробка клієнтської частини	14.02.2024	
5	Розробка серверної частини	15.03.2024	
6	Тестування функціональності системи	26.04.2024	
7	Оформлення звітної частини	17.05.2024	

Здобувач _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Єноктаєв О. О. «Інформаційна система для обліку та аналізу медичних даних». – Кваліфікаційна робота бакалавра за спеціальністю 122 «Комп'ютерні науки», освітньою програмою «Комп'ютерні науки».

Кваліфікаційна робота містить 3 розділи, 45 рисунків, 5 таблиць, 24 літературних джерела за переліком посилань. Робота виконана на 66 сторінках загального тексту і 55 сторінках основного тексту.

Метою роботи є використання та розробка інформаційної системи для обліку та аналізу медичних даних.

Об'єктом дослідження є інформаційна система для обліку та аналізу медичних даних.

Предметом дослідження виступають процеси розробки інформаційної системи для обліку та аналізу медичних даних.

У кваліфікаційній роботі розроблено інформаційну систему для обліку та аналізу медичних даних. Досліджено специфіку розробки інформаційних систем та розробку функціоналу й інтерфейсу. Проведено порівняльний аналіз використання інформаційних систем у медицині. Побудовано модель логістичної регресії, яка використовується для бінарної класифікації і передбачення ймовірності настання серцевої недостатності на основі клінічних параметрів. Для оцінки якості створеної моделі проведено ROC-аналіз. Інформаційна система може бути впроваджена та використана в реальному медичному середовищі.

Ключові слова: інформаційна система, медичні дані, аналіз даних, логістична регресія, R, Shiny, MongoDB.

ABSTRACT

Yenoktaev O.O. "Information system for accounting and analysis of medical data."
– Bachelor's qualification work in specialty 122 "Computer Science", educational program "Computer Science".

The qualification work contains 3 chapters, 45 figures, 5 tables, 24 literary sources according to the list of references. The work was completed on 66 pages of the general text and 55 pages of the main text.

The purpose of the work is the use and development of an information system for accounting and analysis of medical data.

The object of the study is an information system for recording and analyzing medical data.

The subject of the study is the process of developing an information system for recording and analyzing medical data.

In the qualification work, an information system for accounting and analysis of medical data was developed. The specifics of the development of information systems and the development of functionality and interface have been studied. A comparative analysis of the use of information systems in medicine was conducted. A logistic regression model was built, which is used for binary classification and prediction of the probability of heart failure based on clinical parameters. ROC analysis was performed to assess the quality of the created model. The information system can be implemented and used in a real medical environment.

Keywords: information system, medical data, data analysis, logistic regression, R, Shiny, MongoDB.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ	6
ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Специфіка розробки інформаційних систем для медичної галузі	9
1.2 Аналіз наявних аналогів	12
1.3 Аналіз функціональних вимог	18
1.4 Висновки до розділу 1	19
2 ОПИС АЛГОРИТМІВ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	20
2.1 Огляд використовуваних технологій та інструментів.....	20
2.2 Математична модель аналізу медичних даних	25
2.3 Опис структури та принципів роботи розроблених алгоритмів	28
2.3.1 Аналіз медичних даних за допомогою моделі логістичної регресії.....	29
2.3.2 Діаграма послідовності.....	31
2.3.3 MindMap інформаційної системи.....	32
2.4 Висновки до розділу 2	34
3 РОЗРОБКА ТА АНАЛІЗ ОТРИМАНИХ ДАНИХ.....	35
3.1 Розробка бази даних.....	35
3.2 Розробка клієнтської частини	38
3.3 Розробка серверної частини	41
3.4 Функціональне тестування системи.....	45
3.5 Висновки до розділу 3	53
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТКИ.....	58
Додаток А. Результати тестування.....	58
Додаток Б. Програмний код	60

ПЕРЕЛІК СКОРОЧЕНЬ

ЕСОЗ – електронна система охорони здоров'я

МІС – медична інформаційна система

ВООЗ – Всесвітня організація охорони здоров'я

DAPT – Dual Antiplatelet Therapy, подвійна антитромбоцитарна терапія

SCORE – Systemic Coronary Risk Estimation, серцево-судинний ризик

IDE – інтегроване середовище розробки

ML – Machine Learning, машинне навчання

AI – Artificial Intelligence, штучний інтелект

UI – User Interface

HTML – HyperText Markup Language, мова гіпертекстової розмітки

CSS – Cascading Style Sheets, каскадні таблиці стилів

JSON – JavaScript Object Notation, текстовий формат обміну даними

R – мова програмування

ROC – Receiver Operating Characteristic

AUC – Area Under the Curve

ВСТУП

Актуальність теми дослідження обумовлена тим, що в сучасній медичній галузі спостерігається стрімке зростання обсягів медичних даних, які потребують ефективного зберігання, обробки та аналізу. Традиційні методи обробки медичної інформації часто виявляються недостатньо ефективними для вирішення складних завдань прогнозування та діагностики. Використання інформаційних систем, спеціально розроблених для обліку та аналізу медичних даних, може значно покращити якість медичних послуг, сприяти ранньому виявленню захворювань і підвищити ефективність лікування.

Зростання кількості пацієнтів із хронічними захворюваннями, ставить перед медичними установами нові виклики щодо моніторингу стану здоров'я та своєчасного реагування на зміни в стані пацієнтів. Інформаційні системи, які використовують моделі машинного навчання для аналізу медичних даних, здатні автоматизувати процеси обробки великих обсягів даних, забезпечуючи точні прогнози та рекомендації щодо подальшого лікування.

Метою роботи є використання та розробка інформаційної системи для обліку та аналізу медичних даних.

Об'єктом дослідження є інформаційна система для обліку та аналізу медичних даних.

Предметом дослідження виступають процеси розробки інформаційної системи для обліку та аналізу медичних даних.

Для досягнення поставленої мети потрібно вирішити такі *завдання*:

Відповідно до мети завданнями кваліфікаційної роботи є:

- аналіз предметної області та конкурентів;
- формування функціональних вимог;
- проєктування інформаційної системи;
- розробка бази даних інформаційної системи, клієнтської та серверної частини.

У роботі використовувалися такі загальнонаукові та спеціальні методи наукового дослідження: аналіз, синтез, порівняння, метод аналогій, статистичний аналіз, класифікація, логістична регресія, ROC-аналіз тощо.

Практична значущість одержаних результатів полягає у розробці ефективної інформаційної системи для обліку та аналізу медичних даних, що сприяє підвищенню ефективності медичних послуг, забезпечує можливість раннього виявлення пацієнтів з високим ризиком та своєчасного втручання, дозволяє здійснювати точний, швидкий аналіз медичних даних, що сприяє більш обґрунтованому прийняттю рішень медичним персоналом.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Специфіка розробки інформаційних систем для медичної галузі

Завдяки високому рівню розвитку інформаційних технологій нині накопичуються великі обсяги медичних даних, які є основою для розроблення інформаційних систем обліку та аналізу даних. У медицину активно впроваджуються технології моніторингу за станом пацієнтів, на основі яких розробляються динамічні методи автоматизованого формування рекомендацій лікування. Інтеграція IT-рішень у медичну практику є одним із найефективніших і найперспективніших методів запобігання лікарським помилкам. Основне завдання яких полягає в тому, щоб запобігти, шляхом проведеного всебічного аналізу, таким діям, котрі потенційно загрожують здоров'ю та життю пацієнта.

В Україні запроваджена електронна система охорони здоров'я (ЕСОЗ) – це національна інформаційна система, яка зберігає медичні дані про здоров'я пацієнтів у єдиному місці. Вона складається з центральної бази даних та медичних інформаційних систем (МІС) [1].

Згідно, з Міністерством охорони здоров'я, під медичною інформаційною системою розуміють інформаційно-комунікаційні системи, які дозволяють лікарням, лабораторіям та аптекам автоматизувати свою роботу та взаємодіяти з центральною базою даних ЕСОЗ [1]. Зокрема, існуючи МІС спрямовані на задоволення потреб працівників у сфері охорони здоров'я та медичних установ. Активність у цьому напрямку дуже велика, так понад 35 МІС які були розроблені українськими IT-компаніями приєдналися до ЕСОЗ. Кожна з цих систем проходить тестування на відповідність технічним вимогам, включаючи питання безпеки, що встановлюють та перевіряють державні органи [2].

Однак, розробники МІС не обмежуються лише інтеграцією з ЕСОЗ, а розробляють додаткові сервіси, що відповідають потребам медичних установ. Наприклад, це можуть бути аналітичні модулі, модулі обліку ліків, бухгалтерські модулі або модуль запису на приймання. Таким чином, розробка і впровадження

нових інформаційних систем для медичної сфери спрямовані на поліпшення ефективності та доступності медичних послуг для громадян, а також на забезпечення відповідності стандартам та вимогам держави у сфері збереження та обробки медичних даних.

Медичні дані – це будь-яка інформація, пов'язана зі здоров'ям людини, яка використовується для прийняття рішень щодо діагностики, лікування або догляду за пацієнтом. Ці дані можуть бути представлені в різних форматах, включаючи:

- демографічну інформацію: ім'я, вік, стать, адреса;
- історію хвороби: хронічні захворювання, алергії, операції, госпіталізації;
- результати лабораторних досліджень: аналізи крові, сечі, зображення;
- медичні записи: записи лікарів, медсестр, інших медичних працівників;
- фармацевтичну інформацію: рецепти, безрецептурні ліки, добавки;
- дані моніторингу здоров'я: артеріальний тиск, частота серцевих скорочень, рівень глюкози в крові тощо [2, 3].

Інформаційна система для обліку та аналізу медичних даних – це комплекс програмних та апаратних засобів, що забезпечують збір, зберігання, обробку та аналіз медичних даних з метою підтримки прийняття рішень у медичній практиці.

Основні принципи створення таких систем включають: гнучкість – можливість адаптації до змінюваних умов; надійність – забезпечення точності та повноти даних; доступність – можливість доступу до даних в реальному часі; конфіденційність – захист медичної інформації від несанкціонованого доступу.

Специфіка розробки інформаційних систем для обліку та аналізу медичних даних містить кілька ключових аспектів:

1. Конфіденційність та безпека даних. Оскільки медичні дані є дуже чутливими та конфіденційними, система повинна забезпечувати високий рівень захисту, включаючи шифрування, контроль доступу та ідентифікацію користувачів. Так, Закон України «Про захист персональних даних» [4] та Постанова Кабінету Міністрів України «Деякі питання електронної системи охорони здоров'я» [5] встановлюють правила збору, зберігання та обробки

медичних даних та інших персональних даних пацієнтів у медичних установах. Вони включають основні аспекти законодавства України щодо захисту медичних даних, а саме :

- заборона безпідставного розголошення медичної інформації;
- зобов'язання медичним установам забезпечити адекватний захист медичних даних від несанкціонованого доступу та випадкової втрати чи пошкодження;
- гарантія пацієнтам право на доступ до своїх медичних записів та іншої медичної інформації, яка стосується їхнього здоров'я;
- у разі порушення вимог законодавства щодо захисту медичних даних передбачені штрафи та інші санкції для винних осіб та організацій.

2. Вимоги до робочого процесу медичних установ. Система повинна відповідати специфічним вимогам та потребам робочого процесу медичних установ, забезпечуючи швидкий доступ до інформації та зручний інтерфейс для користувачів.

3. Специфіка медичної термінології. Розробники повинні мати глибоке розуміння медичної термінології та процесів, що використовуються в клінічній практиці, для ефективної інтеграції цієї термінології в систему.

4. Інтеграція з наявними системами. Багато медичних установ вже використовують певні інформаційні системи. Тому важливо, щоб нова система могла легко інтегруватися з наявними системами, забезпечуючи безперервність у роботі.

5. Аналіз та обробка великих обсягів даних. Зважаючи на те, що медичні дані можуть бути об'ємними та різноманітними, система повинна мати потужні засоби для аналізу та обробки цих даних, включаючи інструменти машинного навчання та штучного інтелекту.

6. Дотримання стандартів та правових актів. Розробники повинні бути свідомими до стандартів та правових актів, які стосуються медичних даних і впроваджувати відповідні заходи для дотримання цих вимог.

Враховуючи специфіку розробки інформаційних систем та специфіку медичних даних, включаючи їх об'єм, різноманітність та конфіденційність, створення інформаційної системи для обліку та аналізу цих даних стає необхідністю. Основним інструментом для аналізу даних в таких системах є моделі машинного навчання. Вони дозволяють визначити взаємозв'язки між різними змінними, наприклад, між однією залежною бінарною змінною та однією або декількома незалежними змінними. У контексті медичних даних це може означати прогнозування ймовірності певного медичного стану на основі різних факторів. Такі фактори можуть включати вік пацієнта, стать, історію хвороби, результати лабораторних досліджень тощо. Таким чином, інформаційні системи для обліку та аналізу медичних даних допомагають покращити якість медичного обслуговування та забезпечити більш точне прогнозування та діагностику.

1.2 Аналіз наявних аналогів

У сучасному світі існує значна кількість інформаційних систем для обліку та аналізу медичних даних, що використовуються у різних галузях медицини для прогнозування, діагностики та лікування різноманітних захворювань. Проте, згідно з даними Всесвітньої організації охорони здоров'я (ВООЗ), серцево-судинні захворювання є найпоширенішими у світі. Саме тому ми акцентуємо увагу на системах, які використовуються для прогнозування серцево-судинних захворювань. Для створення власної інформаційної системи важливим етапом є пошук та аналіз наявних аналогів з метою подальшого формування функціональних вимог. Тому розглянемо переваги та недоліки обраних систем: Framingham Risk Score, Шкала DAPT, Шкала SCORE, Шкала HAS-BLED за допомогою SWOT-аналізу.

SWOT-аналіз є попереднім інструментом прийняття рішень, одним з найпотужніших та найпоширеніших стратегічних інструментів у всьому світі. Результати SWOT-аналізу розділені на чотири категорії: S (Strengths) – сильні

сторони, W (Weaknesses) – слабкі сторони, O (Opportunities) – можливості, T (Threats) – загрози [6].

1. Framingham Risk Score for Hard Coronary Heart Disease – фрамінгемська шкала ризику тяжкої ішемічної хвороби серця (рис. 1.1) , ця система оцінює 10-річний ризик серцевого нападу. Вона призначена для використання у недіабетичних пацієнтів віком від 30 до 79 років [7].

Framingham Risk Score for Hard Coronary Heart Disease

Estimates 10-year risk of heart attack.

INSTRUCTIONS
 There are several distinct Framingham risk models. MDCalc uses the 'Hard' coronary Framingham outcomes model, which is intended for use in **non-diabetic** patients age 30-79 years with no prior history of coronary heart disease or intermittent claudication, as it is the most widely applicable to patients without previous cardiac events. See the [official Framingham website](#) for additional Framingham risk models.

When to Use ▾
 Pearls/Pitfalls ▾

Age		years
Sex	Female Male	
Smoker	No Yes	
Total cholesterol	Norm: 150 - 200	mg/dL ↔
HDL cholesterol	Norm: 0 - 60	mg/dL ↔
Systolic BP	Norm: 100 - 120	mm Hg
Blood pressure being treated with medicines	No Yes	

Result:
 Please fill out required fields.

Рисунок 1.2 – Фрамінгемська шкала ризику тяжкої ішемічної хвороби серця

Таблиця 1.1 – SWOT-аналіз Framingham Risk Score

Сильні сторони	Слабкі сторони
- велика база даних та довга історія досліджень; - використання визнаних факторів ризику; - добре вивчена та валідована модель прогнозування.	- менша точність для індивідуальних прогнозів; - обмежена адаптивність до нових факторів ризику; - складність для розуміння
Можливості	Загрози
- постійне оновлення та вдосконалення моделі; - інтеграція з іншими клінічними системами	- конкуренція з новими, більш точними моделями; - зміни в клінічних підходах та рекомендаціях.

2. Шкала DAPT – шкала оцінки ризику, що використовується для визначення оптимальної тривалості подвійної антитромбоцитарної терапії (рис.1.2) [8].

DAPT

Вік

- ☒ більше 75
- ☐ 65-75
- ☐ менше 65

Куріння

- ☒ Ні ☐ Так

Діабет

- ☒ Ні ☐ Так

Інфаркт при госпіталізації

- ☒ Ні ☐ Так

Попереднє ЧКВ або інфаркт

- ☒ Ні ☐ Так

Стент вкритий паклітакселом

- ☒ Ні ☐ Так

Діаметр стенту менше 3 мм

- ☒ Ні ☐ Так

Застійна серцева недостатність або фракція викиду < 30 %

- ☒ Ні ☐ Так

Венозне стентування

- ☒ Ні ☐ Так

Результат:

Рекомендація:

Рисунок 1.2 – Шкала DAPT

Таблиця 1.2 – SWOT-аналіз Шкала DAPT

Сильні сторони	Слабкі сторони
- допомагає в прийнятті рішень щодо тривалості та доцільності антиагрегантної терапії після перенесеного серцевого нападу або стентування; - використання в клінічній практиці та підтверджена ефективність.	- складність визначення оптимальної тривалості терапії; - може вимагати додаткових ресурсів та часу для оцінки та розрахунку ризику
Можливості	Загрози
- вдосконалення та розширення шкали для врахування нових факторів ризику; - розробки алгоритмів прийняття рішень для автоматизації клінічної практики	- поява конкурентних моделей або методів для прийняття рішень; - зміни в клінічних рекомендаціях та протоколах.

3. Шкала SCORE (Systematic COronary Risk Evaluation) – використовується для оцінки ризику смерті людини від серцево-судинних захворювань протягом найближчих 10 років (рис 1.3) [9].

SCORE

Шкала SCORE () дозволяє оцінити ризик смерті людини від серцево-судинних захворювань протягом найближчих 10 років. Рекомендується використовувати шкалу SCORE у людей у віці 40 років і старше.

*Виберіть всі пункти для оцінки

Стать

Жінка

Чоловік

Показники 1

Систолічний АТ, мм.рт.ст

Холестерин, ммоль / л

Показники 2

Вік

☐ Палить

Шкала ризику

<1%

1%

2%

3-4%

5%

10-14%

>15%

European Guidelines on CVD Prevention in Clinical Practice 2016 Eur J Prev Cardiol. 2016 Jul; 23(11):NP1-NP96. doi: 10.1177/2047487316653709

Скинути
Розрахувати

Рисунок 1.3 – Шкала SCORE

Таблиця 1.3 – SWOT-аналіз Шкала SCORE

Сильні сторони	Слабкі сторони
- використовується для оцінки ризику серцево-судинних подій у пацієнтів на основі різних факторів ризику; - простий у використанні та інтерпретації.	- менша точність для індивідуальних прогнозів; - не враховує деякі нові фактори ризику
Можливості	Загрози
- постійне оновлення та адаптація для врахування нових даних та клінічних відкриттів; - інтеграція з іншими клінічними системами для автоматизації оцінки ризику.	- конкуренція з іншими моделями оцінки ризику, які можуть бути більш точними або універсальними; - зміни в клінічних підходах, які можуть вплинути на використання SCORE в практиці.

4. Шкала HAS-BLED дозволяє оцінити ризик великої кровотечі при миготливій аритмії та для оцінки ризику великої кровотечі у пацієнтів, які отримують антикоагулянти (рис. 1.4) [10].

HAS-BLED

Шкала HAS-BLED дозволяє оцінити ризик великої кровотечі при миготливій аритмії. Під великою кровотечею мається на увазі: будь який ІНТРАКРАНІАЛЬНОГО крововилив, кровотеча, що вимагає госпіталізації, що супроводжується зниженням гемоглобіну, або вимагає гемотрансфузії.

*Виберіть всі пункти для оцінки

Клінічні симптоми

☐ Інсульт +1

☐ Вік старше 65 років +1

☐ Зловживання алкоголем +1

☐ Артеріальна гіпертензія (систолический артеріальний тиск вище 160 мм рт. Ст.) +1

☐ Прийом препаратів, що підвищують ризик кровотечі (антиагреганти, НПЗЗ) +1

☐ Кровотеча в анамнезі і / або схильність до кровотеч (в т.ч. анемія) +1

☐ Лабільний МНО (нестабільне / висока або в терапевтичному діапазоні менше 60% часу) +1

☐ Порушена функція нирок (діаліз, трансплантація нирки або креатинін сироватки вище 200 мкмоль / л) +1

☐ Порушена функція печінки (важке хронічне захворювання або підвищення білірубину в 2 рази і більше вище верхньої межі норми в поєднанні з підвищенням активності АсТ / АЛТ в 3 рази і більше вище верхньої межі норми). +1

Ip GY. Implications of the CHA(2)DS(2)-VASc and HAS-BLED Scores for thromboprophylaxis in atrial fibrillation. Am J Med. 2011 Feb; 124(2):111-4. PubMed ID: 20887966

Скинути
Розрахувати

Рисунок 1.4 – Шкала HAS-BLED

Таблиця 1.4 – SWOT-аналіз Шкала HAS-BLED

Сильні сторони	Слабкі сторони
- допомога в клінічному прийнятті рішень щодо ризику кровотеч та прийому антикоагулянтів; - використовується в клінічній практиці та має підтверджену ефективність.	- складність використання та оцінки, оскільки вимагає досить детального аналізу даних пацієнта; - не враховує всі можливі фактори ризику кровотеч та антикоагулянтної терапії
Можливості	Загрози
- розробка додаткових алгоритмів для автоматизованого розрахунку ризику та прийняття рішень на основі HAS-BLED SCORE; - розширення використання в клінічних сценаріях та популяціях.	- конкуренція з іншими моделями або методиками для оцінки ризику кровотеч та вибору антикоагулянтної терапії; - зміни в клінічних протоколах та рекомендаціях, що можуть вплинути на використання HAS-BLED SCORE в практиці.

Для кращого зрозуміння потенційних переваг та обмежень кожної системи була зроблена візуалізація результатів SWOT-аналізу для конкретної системи (рис.1.5). Кожна діаграма відображає сильні та слабкі сторони, можливості та загрози для відповідної системи.

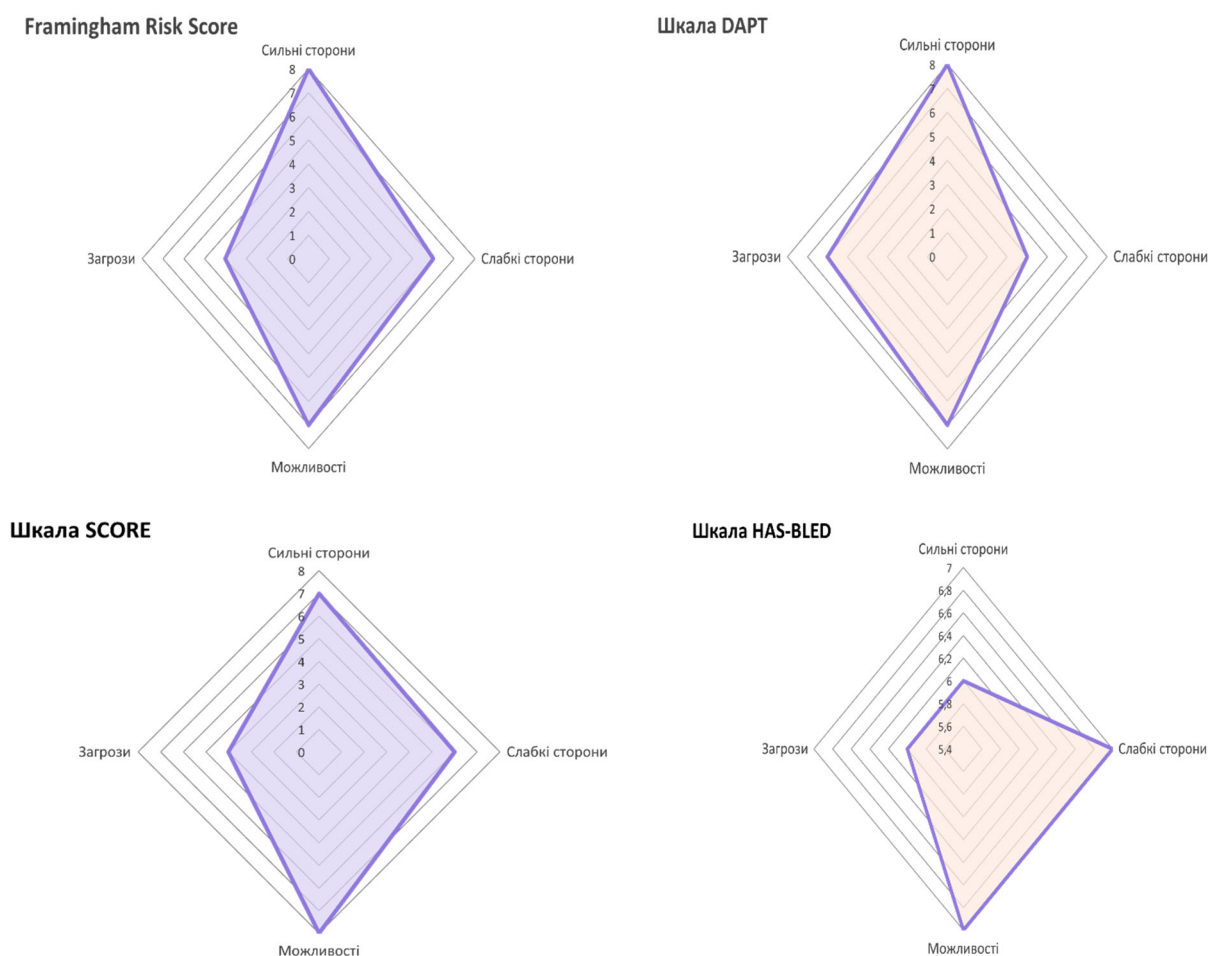


Рисунок 1.5 - Візуалізації результатів SWOT-аналізу для конкретної системи

Після проведення SWOT-аналізу систем Framingham Risk Score, Шкала DAPT, Шкала SCORE та Шкала HAS-BLED, можливо зробити висновок. Перш за все, всі ці системи мають свої сильні сторони, які роблять їх ефективними в певних областях. Однак, вони також мають слабкі сторони, які можуть обмежувати їхню ефективність або застосування. Щодо можливостей, кожна система має потенціал для подальшого розвитку та вдосконалення. Наприклад, з використанням новітніх технологій, таких як штучний інтелект та машинне навчання, можливо значно покращити точність прогнозування та діагностики. Однак, існують також загрози, які потрібно враховувати. Це можуть бути технічні виклики, проблеми з конфіденційністю даних, а також зміни в медичних стандартах та регулюванні. Враховуючи всі ці фактори, можливо сформулювати вимоги щодо того, які аспекти потребують уваги при розробці інформаційної системи для обліку та аналізу медичних даних.

1.3 Аналіз функціональних вимог

Для розробки інформаційної системи для обліку та аналізу медичних даних, а саме оцінки ризику серцевої недостатності, треба визначити вимоги які повинні охоплювати ключові функції та особливості, які система повинна забезпечувати. Існують функціональні та нефункціональні вимоги для розробки інформаційних систем [11].

Функціональні вимоги визначають, які функції інформаційної системи повинні бути реалізовані:

- система повинна мати можливість обробки медичних даних пацієнтів та визначення ризику серцевої недостатності на основі введених параметрів (наприклад, вік, стать, аналізи крові тощо);
- забезпечувати зберігання історії медичних записів пацієнтів, результатів обстежень, лікування та інших важливих даних;
- дозволяти генерувати звіти для лікарів, дослідників та адміністраторів, щоб аналізувати дані та приймати рішення.

Нефункціональні вимоги визначають якості, які система повинна відповідати, а саме:

- забезпечувати захист конфіденційної інформації пацієнтів, використовуючи шифрування, контроль доступу та інші методи;
- система повинна бути доступною для користувачів 24/7, забезпечуючи надійність та швидкий доступ до даних;
- працювати ефективно, обробляючи запити користувачів швидко та без затримок.
- мати зручний та інтуїтивно зрозумілий інтерфейс для лікарів, медсестр, пацієнтів та інших фахівців.

Урахування функціональних та нефункціональних вимоги допоможуть розробити інформаційну систему, яка буде ефективною для збору, аналізу та візуалізації даних для оцінки ризику серцевої недостатності.

1.4 Висновки до розділу 1

У розділі проаналізовано специфіку розробки інформаційних систем для медичної галузі. Дано поняття медичних даних, інформаційної системи для обліку та аналізу медичних даних. Розглянуто специфіка розробки інформаційних систем для обліку та аналізу медичних даних, доведено що основним інструментом для аналізу даних в таких системах є моделі машинного навчання.

Проведено пошук та аналіз наявних аналогів з метою подальшого формування функціональних вимог. За допомогою SWOT-аналізу розглянуто переваги та недоліки обраних систем. Зазначено, що всі ці системи мають свої сильні сторони, які роблять їх ефективними в певних областях. Однак, вони також мають слабкі сторони, які можуть обмежувати їхню ефективність або застосування. Враховуючи всі фактори сформульовані функціональні вимоги, які потребують уваги при розробці інформаційних систем для обліку та аналізу медичних даних.

РОЗДІЛ 2 ОПИС АЛГОРИТМІВ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Огляд використовуваних технологій та інструментів

Для розробки даного дослідження було вибрано декілька ключових технологій та інструментів, які найкраще відповідають цілям та вимогам проекту.

1. R – мова програмування.

Основною мовою програмування було обрано R через його сильні статистичні можливості та широкий спектр пакетів, які спеціально розроблені для аналізу даних [12]. Аналітики даних використовують R, оскільки він пропонує статистичні пакети для кількісних застосувань. Вони включають нейронні мережі, філогенетику, розширені графіки та моделі машинного навчання.

В галузі охорони здоров'я R знаходить застосування в клінічних дослідженнях, епідеміології, біостатистиці та геноміці. R також є мовою з відкритим кодом, призначеною для внесення змін. R – потужна мова для аналізу даних і розширеного статистичного моделювання, працює зі складними наборами даних та має велику колекцію статистичних пакетів з можливістю візуалізації [13].

У мови R є понад 5 мільйонів репозиторіїв на Github, його використовують Facebook, Twitter, Google, Uber, Airbnb. Завдяки зростанню R-спільноти та поповненню безкоштовних бібліотек і пакетів, доступних у CRAN, наприклад, e1071, CARET, H2O тощо, R краще оснащений для вирішення проблем, які виникають через AI/ML та глибоке навчання. Крім того, мова R широко використовується для розробки прототипів, особливо після появи інтегрованих середовищ розробки (IDE), таких як RStudio.

Згідно з дослідженнями інституту досліджень людських даних IQVIA, провідного світового постачальника передової аналітики, технологічних рішень та послуг з клінічних досліджень для галузі медико-біологічних наук у таблиці 1.5 наведено спрощений набір критеріїв, за допомогою яких провідний фахівець міг би визначити, яке програмне забезпечення відповідає рішенням його завдання [14].

Таблиця 1.5 – Порівняння відомих переваг програмного рішення R та Python

Критерій	R	Python
Вартість	безкоштовно	безкоштовно
Навчання	Помірний	Помірний
Маніпулювання даними	Високий	Високий
Аналітичне моделювання	Високий	Середній
Графічні можливості	Високий	Середній
Опрацювання тексту	Середній	Високий
Підтримка глибокого навчання	Середній	Високий
Загальні вживання	Високий	Середній
Підтримка клієнтів	Середній	Середній
Удосконалення в інструменті	Високий	Середній

Таким чином, з метою врахування вимог до збору, аналізу та обліку медичних даних, а також можливостей візуалізації, для розробки інформаційної системи була обрана мова програмування R.

2. RStudio – це інтегроване середовище розробки для R [15]. Він містить консоль, редактор підсвічування синтаксису, який підтримує пряме виконання коду, а також інструменти для креслення, історії, налагодження та керування робочим простором (рис. 2.1). RStudio доступний у версіях з відкритим вихідним кодом і комерційних версіях і працює на робочому столі Windows, Mac і Linux.

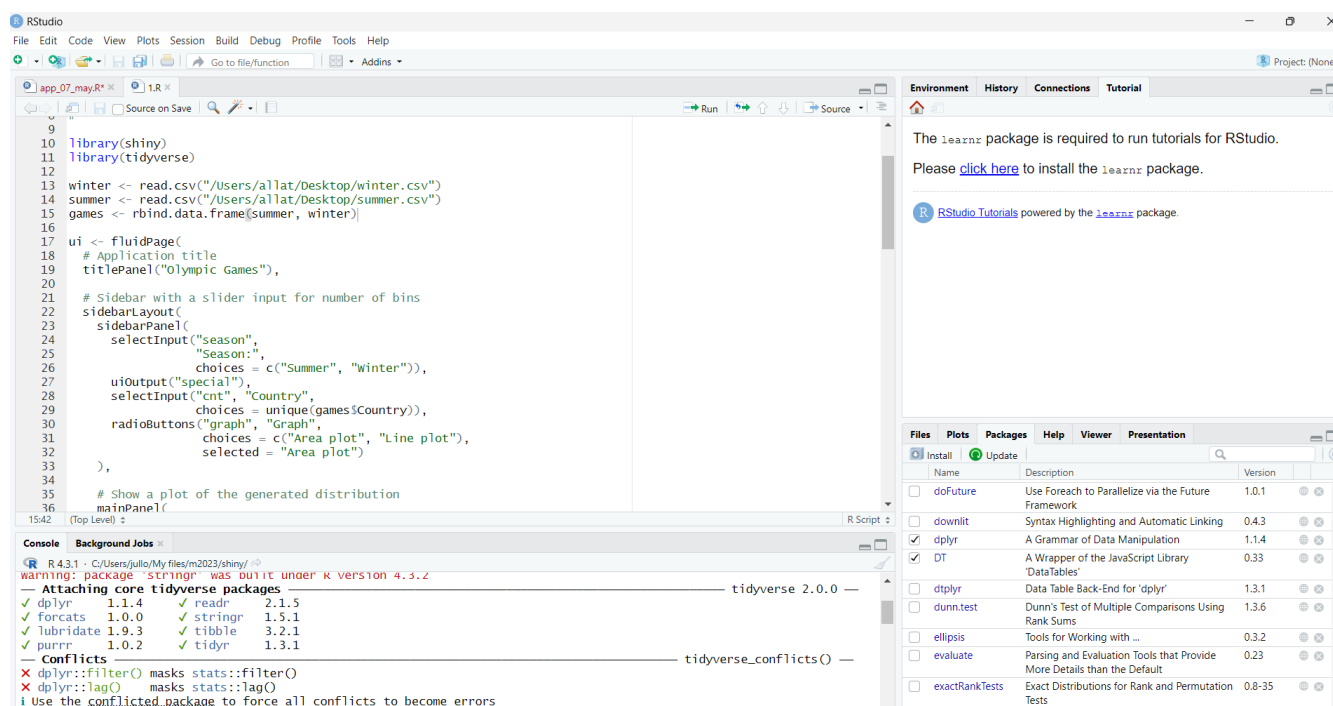


Рисунок 2.1 – Інтегроване середовище розробки RStudio

3. Shiny – фреймворк R для створення інтерактивних вебзастосунків, було вибрано для розробки фронтенду – користувацького інтерфейсу. Shiny дозволяє легко інтегрувати аналітичні та візуалізаційні можливості R з інтерактивним вебінтерфейсом [16]. Основні компоненти Shiny (рис. 2.2) включають клієнтську частину, серверну частину і реактивне програмування.

```
library(shiny)
library(bslib)

# Визначити User Interface ----
ui <- page_sidebar(
)

# Визначити server logic ----
server <- function(input, output) {
}

# Run the app ----
shinyApp(ui = ui, server = server)
```

Рисунок 2.2 – Структура застосунку Shiny

Основні компоненти Shiny:

1) Клієнтська частина (UI, User Interface) – визначає зовнішній вигляд вебзастосунку. Використовує функції з пакету shiny для створення HTML-елементів, таких як текстові поля, кнопки, графіки, таблиці тощо. Код для створення клієнтської частини зазвичай знаходиться у функції ui.

2) Серверна частина (Server) – виконує обчислення та обробляє логіку вебзастосунку. Взаємодіє з клієнтською частиною для оновлення вмісту на основі введення користувача. Код для створення серверної частини знаходиться у функції server.

3) Реактивне програмування – ключова концепція Shiny, яка дозволяє автоматично оновлювати вміст вебзастосунку при зміні вхідних даних. Використовує реактивні вирази (reactive), реактивні значення (reactiveValues) та спостерігачів (observe, observeEvent).

Плюси Shiny:

- дозволяє створювати складні вебзастосунки;

- інтуїтивно зрозумілий синтаксис для користувачів R;
- забезпечує створення інтерактивних графіків, таблиць та інших елементів інтерфейсу, які реагують на дії користувача в режимі реального часу;
- дозволяє швидко створювати прототипи застосунків для візуалізації та аналізу даних;
- підтримує широкий спектр бібліотек для статистичного аналізу, машинного навчання, графічних візуалізацій тощо;
- можливість додавання користувацьких JavaScript, HTML та CSS для створення більш складних інтерфейсів та інтеграцій.

4. MongoDB – це нереляційна база даних, яка пропонує високу продуктивність, високу доступність та легкість масштабування [17]. Вона дозволяє працювати з даними більш гнучким та ефективним способом, ніж традиційні реляційні бази даних.

Ось декілька ключових аспектів використання MongoDB :

1) Зберігання даних. MongoDB використовується для зберігання введених користувачем даних та результатів прогнозування. Це дозволяє зберігати історію введених даних та відповідних прогнозів для подальшого аналізу.

2) Обробка даних. MongoDB використовується для ефективної обробки великих обсягів даних. Це особливо корисно при роботі з великими наборами даних, які використовуються для тренування моделі.

3) Гнучкість структури даних. Однією з ключових переваг MongoDB є її схема без даних, яка дозволяє зберігати дані в гнучкому, JSON-подібному форматі, що включає в себе вкладені документи та масиви. Це робить MongoDB ідеально підходящим для проектів, які можуть вимагати змін у структурі даних.

4) Масштабованість. MongoDB легко масштабується горизонтально, додаючи сервери та забезпечуючи високу доступність та балансування навантаження.

5. Пакети R – це організовані набори методів та класів для виконання вузького набору задач під час програмування на R. Для вирішення певних класів завдань в даній роботі було використано такі пакети R [18].

Shiny – пакет для створення інтерактивних вебдодатків на мові програмування R [16];

Tidyverse – набір пакетів для обробки та аналізу даних. Включає такі пакети, як dplyr, ggplot2, readr, tidyr та інші;

Dplyr – пакет надає функції для маніпулювання даними, такі як фільтрація, сортування, групування та об'єднання;

ggplot2 – пакет для створення графіків та візуалізації даних. Він базується на граматиці графіків та дозволяє легко створювати складні графіки;

ROCR – пакет призначений для аналізу характеристик робочої характеристики ROC та побудови кривих ROC;

Cutpointr – пакет для визначення оптимальних значень порогів в різних задачах, таких як класифікація та діагностика;

Shinythemes – пакет, який надає додаткові теми для веб-додатків, створених з використанням Shiny;

DT – пакет дозволяє створювати інтерактивні таблиці для відображення даних;

Mongolite – пакет призначений для роботи з MongoDB в R;

Bslib – пакет, який дозволяє налаштовувати теми для вебзастосунків, створених з використанням Shiny;

Thematic – пакет надає додаткові теми для графіків, створених з використанням ggplot2;

GGally – розширення для ggplot2, яке надає додаткові можливості для візуалізації даних, такі як матриці розсіювання, гістограми та інші.

Слід відзначити, що використання численних пакетів R для різних завдань забезпечує ефективну роботу з даними та можливості їх аналізу та візуалізації, що є ключовими для успішної розробки.

Таким чином, у процесі розробки були використані різноманітні технології та інструменти (рис. 2.3), які забезпечують ефективний аналіз та обробку даних, а також створення інтерактивного вебзастосунку.



Рисунок 2.3 – Логотипи обраних засобів розробки

2.2 Математична модель аналізу медичних даних

В медичній сфері для аналізу медичних даних використовуються різноманітні моделі, які допомагають вирішувати різні завдання. Одні з них можуть бути спрямовані на класифікацію, інші на прогнозування, а ще інші на виявлення аномалій. Модель логістичної регресії є однією з найпоширеніших у медичних дослідженнях. Вона використовується для бінарної класифікації (наприклад, визначення наявності захворювання на підставі деяких ознак). Логістична регресія дозволяє оцінити ймовірність належності об'єкта до певного класу, її прогностична сила допомагає медичним працівникам виявляти пацієнтів із високим ризиком, забезпечуючи раннє втручання та стратегії лікування.

Логістична регресія є популярним методом для бінарної класифікації, тобто для передбачення одного з двох можливих результатів. У контексті розробки інформаційної системи прогнозу серцевої недостатності, логістична регресія може бути використана для передбачення ймовірності настання серцевої недостатності (позитивний результат) або її відсутності (негативний результат) на основі різних клінічних параметрів.

1. Формалізація задачі [19] .

Нехай y – бінарна змінна, яка приймає значення 1, якщо у пацієнта настала серцева недостатність, і 0, якщо не настала. $x = (x_1, x_2, \dots, x_n)$ – набір незалежних

змінних (факторів ризику), які можуть впливати на ймовірність серцевої недостатності. Це можуть бути такі параметри, як вік, стать, артеріальний тиск, рівень холестерину, частота серцевих скорочень тощо.

2. Логістична функція.

Логістична регресія моделює ймовірність настання події (серцевої недостатності) за допомогою логістичної функції:

$$P(y = 1|x) = \frac{1}{1+e^{-(\beta_0+\beta_1x_1+\beta_2x_2+\dots+\beta_nx_n)}} \quad (2.1)$$

де $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ – коефіцієнти моделі, які необхідно оцінити.

3. Лінійна частина моделі.

Лінійна комбінація незалежних змінних називається лінійним предиктором:

$$z = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n \quad (2.2)$$

4. Ймовірність та логіт-функція.

Ймовірність настання серцевої недостатності можна записати як:

$$P(y = 1|x) = \sigma(z) = \frac{1}{1+e^{-z}} \quad (2.3)$$

де $\sigma(z)$ – логістична функція.

Логіт-функція, яка є оберненою до логістичної, має вигляд:

$$\log \text{it}(P) = \log\left(\frac{P}{1-P}\right) = z = \beta_0 + \beta_1x_1 + \beta_2x_2 + \dots + \beta_nx_n \quad (2.4)$$

5. Оцінювання коефіцієнтів.

Коефіцієнти $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ оцінюються методом максимальної ймовірності. Цей метод знаходить такі значення коефіцієнтів, які максимізують логарифм функції правдоподібності:

$$L(\beta_0, \beta_1, \dots, \beta_n) = \prod_{i=1}^m P(y_i | x_i)^{y_i} (1 - P(y_i | x_i))^{1-y_i} \quad (2.5)$$

6. Прийняття рішення.

Після оцінки коефіцієнтів моделі, для нового набору даних x_{new} можна обчислити ймовірність серцевої недостатності:

$$P(y = 1|x_{\text{new}}) = \frac{1}{1+e^{-(\beta_0+\beta_1x_{1,\text{new}}+\beta_2x_{2,\text{new}}+\dots+\beta_nx_{n,\text{new}})}} \quad (2.6)$$

Рішення приймається на основі порогового значення α . Наприклад, якщо $P(y = 1|x_{new}) \geq \alpha$, то прогнозується серцева недостатність (тобто $y=1$), інакше $y=0$.

Для оцінки якості класифікаційної моделі логістичної регресії використовуються ROC – крива та AUC – міра площі під ROC-кривою. Вони допомагають визначити, наскільки добре модель може відрізнити класи позитивних і негативних результатів [20].

На рис. 2.4. зображена ROC-крива, де по осі ординат позначена частота істинно позитивних результатів (чутливість), по осі абсцис – частота хибно позитивних результатів (1 - специфічність) по всьому діапазону точок поділу. Значення осей відповідають ймовірностям від 0 до 1 (тобто від 0 до 100%).

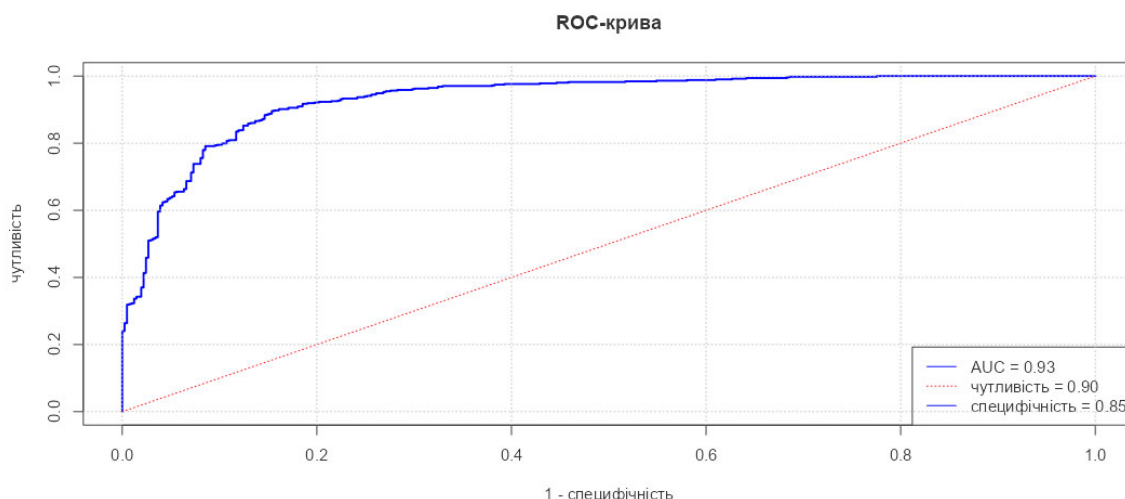


Рисунок 2.4 – ROC-крива

ROC-крива показує співвідношення між True Positive Rate (чутливістю) та False Positive Rate (1 - специфічністю) при різних порогах класифікації. Існує чотири порога класифікації: істинно позитивний результат (true-positive, TP); хибно позитивний результат (false-positive, FP); істинно негативний результат (true-negative, TN); хибно негативний результат (false-negative, FN).

Чутливість тесту:

$$TPR = \frac{TP}{TP+FN} \quad (2.7)$$

Специфічність тесту

$$FPR = \frac{FP}{TN+FP} \quad (2.8)$$

Точність тесту – частка всіх правильних результатів:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.9)$$

Окрім графіка ROC-кривої, якість моделі також оцінюється за допомогою характеристики площі під кривою ROC (AUC - Area Under the ROC Curve). З певними обмеженнями, можна припустити, що чим вищий показник AUC, тим сильнішою є прогностична здатність моделі.

AUC – дає одне числове значення для оцінки якості моделі:

- AUC = 1 означає ідеальну модель, яка правильно класифікує всі приклади;
- AUC = 0.5 означає модель, яка класифікує випадково;
- AUC < 0.5 означає, що модель гірша за випадкову класифікацію.

Таким чином, ROC-крива показує здатність моделі розрізняти позитивні та негативні класи при різних порогових значеннях. AUC дає змогу узагальнити продуктивність моделі в одному числі: чим ближче AUC до 1, тим краща модель.

Модель логістичної регресії може бути використана для оцінки впливу незалежних змін (предикторів) на залежну змінну, а також для прогнозування класифікації нових спостережень. Предиктори можуть бути числовими, інтервальними або категоріальними змінними. Вибір незалежних змін для включення в модель може бути здійснений на основі рівня статистичної значимості менше 0,1, визначеного за результатами ROC-аналізу.

2.3 Опис структури та принципів роботи розроблених алгоритмів

Структура, принципи роботи розроблених алгоритмів є важливим інструментом, який може допомогти покращити процес розробки інформаційної системи, зробити його більш ефективним, надійним та керованим.

2.3.1 Аналіз медичних даних за допомогою моделі логістичної регресії

Аналіз медичних даних за допомогою моделі логістичної регресії, складається з декількох етапів, які можна подати загальною схемою, наведеною на рис. 2.5. Це допоможе зрозуміти процес побудови, тренування та використання моделі для прийняття медичних рішень.

Складові побудови моделі логістичної регресії [21]:

- вибір проблеми дослідження, в нашому випадку, це прогнозування ймовірності ризику серцевої недостатності та типу даних, які є для її вирішення;
- вибір методу, для аналізу медичних даних та прогнозування ризику серцевої недостатності, будемо використовувати метод логістичної регресії;
- збір даних для побудови моделі;
- підготовка даних до аналізу, яка включає обробку відсутніх значень, викидів, очищення тощо;
- вибір функції, вибір змінних які будуть використовуватись у моделі. Залежні змінні – це змінні, які ми намагаємось передбачити. Незалежні змінні – це змінні, які ви будете використовуватись для прогнозування залежних змінних, тобто проведення дослідницького аналізу даних, отримання уявлень та визначення будь-яких зв'язків між незалежними змінними та результатом;
- розділення даних на навчальний набір та тестовий набір. Навчальний набір (70%) використовується для побудови моделі, а тестовий набір (30%) використовується для оцінки продуктивності моделі;
- розробка та тренування моделі, це процес підбору параметрів моделі таким чином, щоб вона якомога краще відповідала даним навчального набору;
- налаштування моделі – це процес вибору значущих змінних, які контролюють процес навчання моделі;
- прогноз моделі навчальної та тестової вибірки для передбачення залежних змінних для нових даних;
- оцінка продуктивності моделі за допомогою ROC-аналізу, матрицею плутанини, AUC;

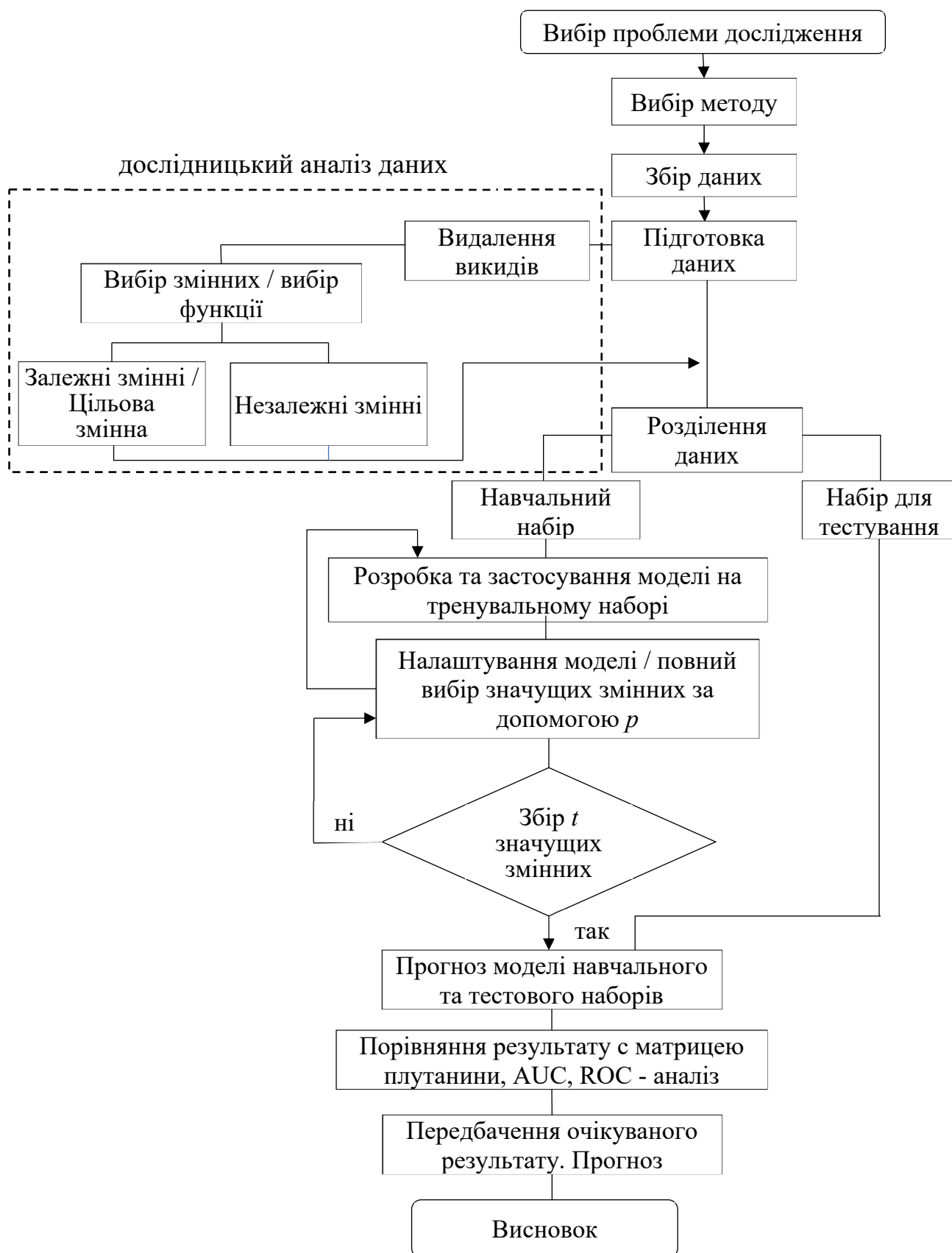


Рисунок 2.5 – Схема аналізу медичних даних з використанням моделі логістичної регресії

- інтерпретація результатів моделі, оцінка продуктивності моделі, процес визначення того, наскільки добре модель відповідає даним тестового набору;
- процес визначення того, чи є модель корисною для вирішення проблеми дослідження
- створення підсумкового висновка.

2.3.2 Діаграма послідовності

Діаграма послідовності (Sequence Diagram) – це один з типів діаграм у мові моделювання UML (Unified Modeling Language) використовується для детального проєктування функцій системи, аналізу та специфікації вимог, відображення процесів і сценаріїв, опису взаємодій між об'єктами в певному часовому контексті. Діаграми послідовності допомагають візуалізувати, як об'єкти обмінюються повідомленнями для виконання функції або процесу [23].

На рисунку 2.6. показано діаграма послідовності.

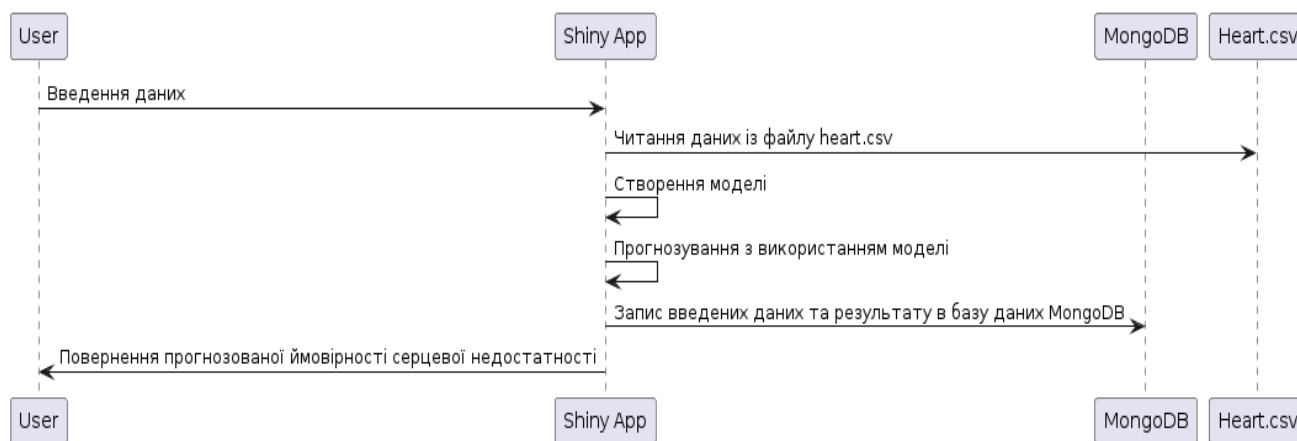


Рисунок 2.6. – Діаграма послідовності

Основні компоненти діаграми послідовності є:

Об'єкти – елементи системи, які беруть участь у процесі.

Повідомлення – горизонтальні стрілки, які відображають обмін інформацією між об'єктами. Вони можуть бути синхронними (запит з очікуванням відповіді) або асинхронними.

Життєві лінії – вертикальні пунктирні лінії, що показують існування об'єкта протягом часу.

Аналізуючи діаграму послідовності (рис.2.6) опишемо взаємодії між об'єктами в певному часовому контексті.

Користувач вводить дані про серцево-судинні захворювання у інформаційній системі Shiny App.

Shiny App читає дані з файлу heart.csv та створює модель логістичної регресії для прогнозування ймовірності розвитку серцевої недостатності:

Shiny App записує введені користувачем дані та прогнозовану ймовірність серцевої недостатності в базу даних MongoDB.

Shiny App повертає користувачеві інформацію про прогнозовану ймовірність серцевої недостатності.

Діаграма послідовності, яка дає візуальне уявлення про те, як різні компоненти системи взаємодіють між собою для прогнозування ризику серцевої недостатності. Вона показує, які повідомлення надсилаються між компонентами та в якому порядку.

2.3.3 MindMap інформаційної системи

MindMap (ментальні карти) – це метод, який допомагає візуально розвивати та систематизувати ідеї та інформацію, зрозуміти концепції, розбиваючи їх на складові. Цей метод використовується для розробки нових ідей або для аналізу та кращого розуміння існуючої інформації [23].

На відміну від інших візуальних діаграм, ментальна карта побудована довкола однієї центральної теми. Вся інформація на карті «прив'язана» до однієї і тієї ж початкової точки.

На рисунку 2.7. розглянемо MindMap структурну схему інформаційної системи для обліку та аналізу медичних даних. Центральна тема – це інформаційна система обліку та аналізу медичних даних. Кожен елемент відображає окремий компонент інформаційної системи. Кожен вузол або гілка, що виходить від центральної теми, представляє окремий процес або функцію в системі. Ці взаємозв'язки допомагають зрозуміти, як різні частини системи взаємодіють і впливають одна на одну. Кожен крок на ментальній карті представляє окремий етап процесу аналізу даних. Наприклад, “Крок 1” включає збір даних, такі як лабораторні результати. “Крок 3” включає обробку цих даних за допомогою метода логістичної регресії для виявлення прогнозу ймовірності серцевої недостатності.

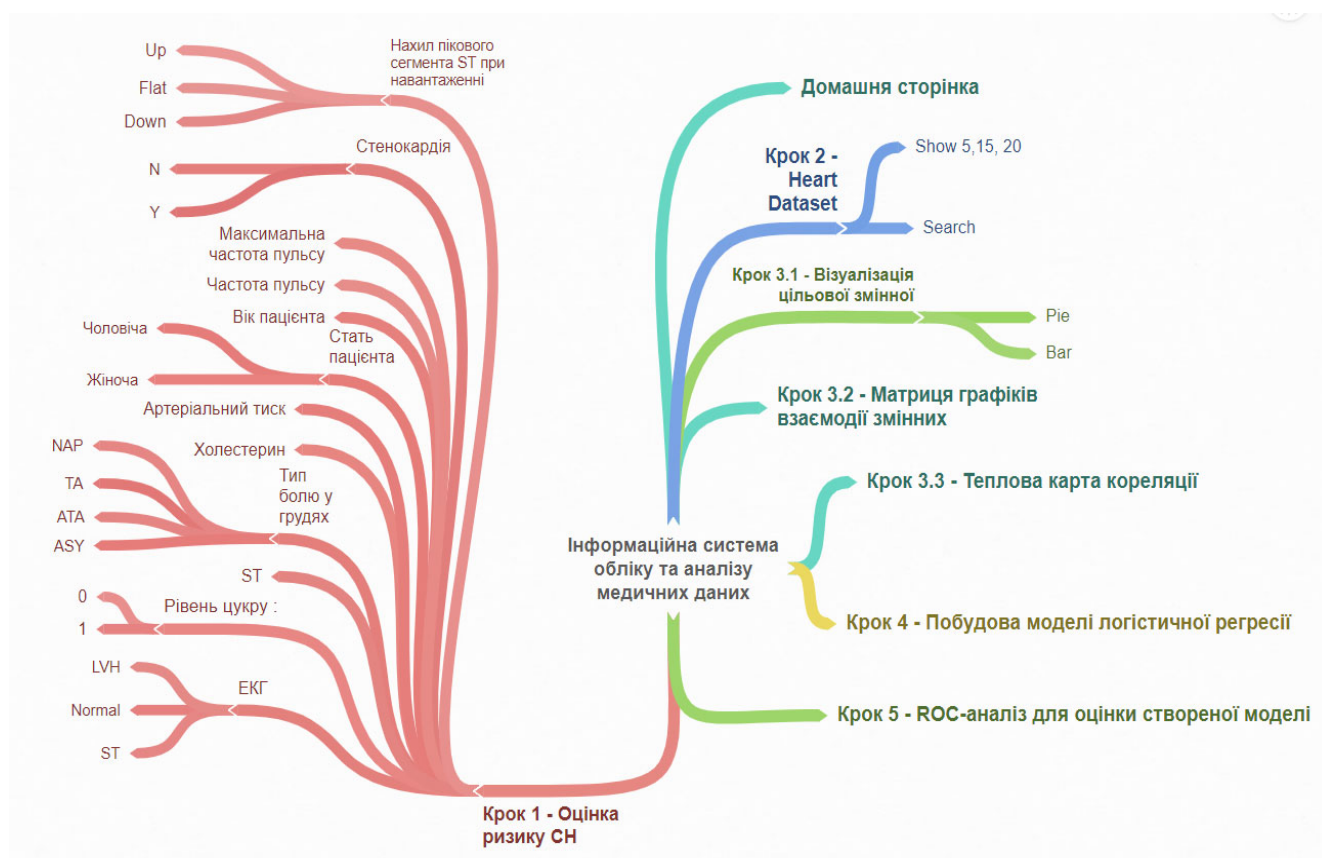


Рисунок 2.7 – Структурна схема інформаційної системи аналізу медичних даних

MindMap використовується для візуального представлення складного процесу аналізу даних, що дає можливість оптимізувати та покращити інформаційну систему.

2.4 Висновки до розділу 2

У другому розділі проведено огляд технологій та інструментів, які використовувалися для розробки інтелектуальної системи. Для створення фронтенду використовувались сучасних технологій HTML, CSS, Bootstrap, а також клієнтська частина User Interface фреймворку Shiny. Для створення бекенду використовувалась серверна частина Server фреймворку Shiny, а також реактивне програмування, яке дозволяє автоматично оновлювати вміст інформаційної системи при зміні вхідних даних. Для обліку, обробки та зберігання введених користувачем даних та результатів прогнозування ймовірності серцевої недостатності використовувалась нереляційна база даних MongoDB

Було розглянута математична модель логістичної регресії аналізу медичних даних, яка використовується для бінарної класифікації передбачення ймовірності настання серцевої недостатності або її відсутності на основі різних клінічних параметрів.

Для розуміння побудови, тренування та використання моделі була представлена схема аналізу медичних даних з використання моделі логістичної регресії.

Створено діаграму послідовності (Sequence Diagram), яка дає візуальне уявлення, як компоненти системи взаємодіють між собою та ментальну карту MindMap для кращого розуміння структури інформаційної системи. Це допомогло забезпечити реалізацію інформаційної системи для обліку та аналізу медичних даних, а саме ймовірності прогнозування серцевої недостатності.

РОЗДІЛ 3 РОЗРОБКА ТА АНАЛІЗ ОТРИМАНИХ ДАНИХ

3.1 Розробка бази даних

Використання баз даних є важливим елементом створення інформаційних систем для обліку та аналізу медичних даних. При розробці системи було використано MongoDB, яка забезпечує високу гнучкість і масштабованість, що є важливим для медичних додатків, які повинні обробляти великі обсяги даних, що швидко змінюються.

Переваги використання MongoDB при розробці даної інформаційної системи:

- підтримує реплікацію даних, що забезпечує високу доступність та захист від втрати даних. Це критично для медичних додатків, де надійність зберігання даних є першочерговою;
- дозволяє зберігати структуровані та неструктуровані дані, включаючи текст, зображення, JSON-документи, що робить її універсальним рішенням для різних типів медичних даних;
- підтримує ACID транзакції, що дозволяє забезпечити цілісність даних при виконанні складних операцій;
- легко інтегрується з Shiny, дозволяючи зберігати і аналізувати дані в реальному часі, що важливо для інтерактивних додатків;
- легко інтегрується з хмарними рішеннями та контейнерами (наприклад, Docker), що спрощує розгортання та масштабування системи.

Для створення бази даних та колекції в MongoDB, можна використовувати інструменти командного рядка MongoDB або MongoDB Compass, графічний інтерфейс користувача для роботи з MongoDB. Для створення бази даних використовувався MongoDB Compass. Розглянемо порядок дій.

1. Запуск MongoDB Compass.
2. Підключення до MongoDB. Для того, щоб підключитися до MongoDB серверу, треба ввести URL підключення (рис.3.1): `mongodb://localhost:27017`.

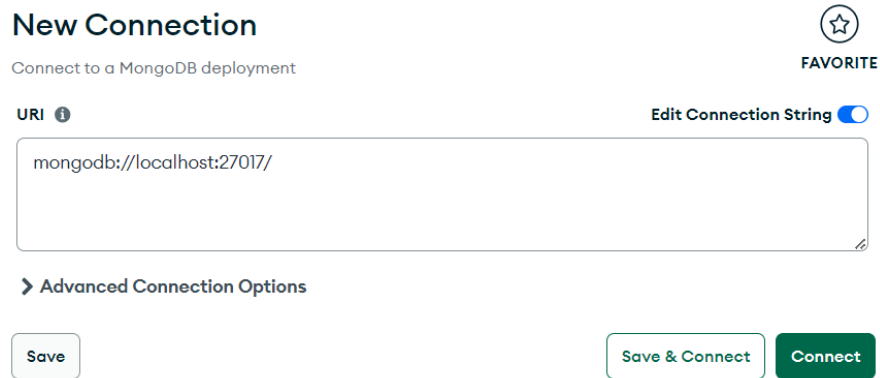


Рисунок 3.1 – Підключення до MongoDB серверу

3. Створення бази даних, для цього у MongoDB Compass треба натиснути "Create Database", ввести HeartDB як ім'я бази даних, а prediction_results як ім'я початкової колекції, та натисніть "Create Database". Ось такий вигляд має створена бази даних HeartDB на програмному рівні (рис. 3.2).

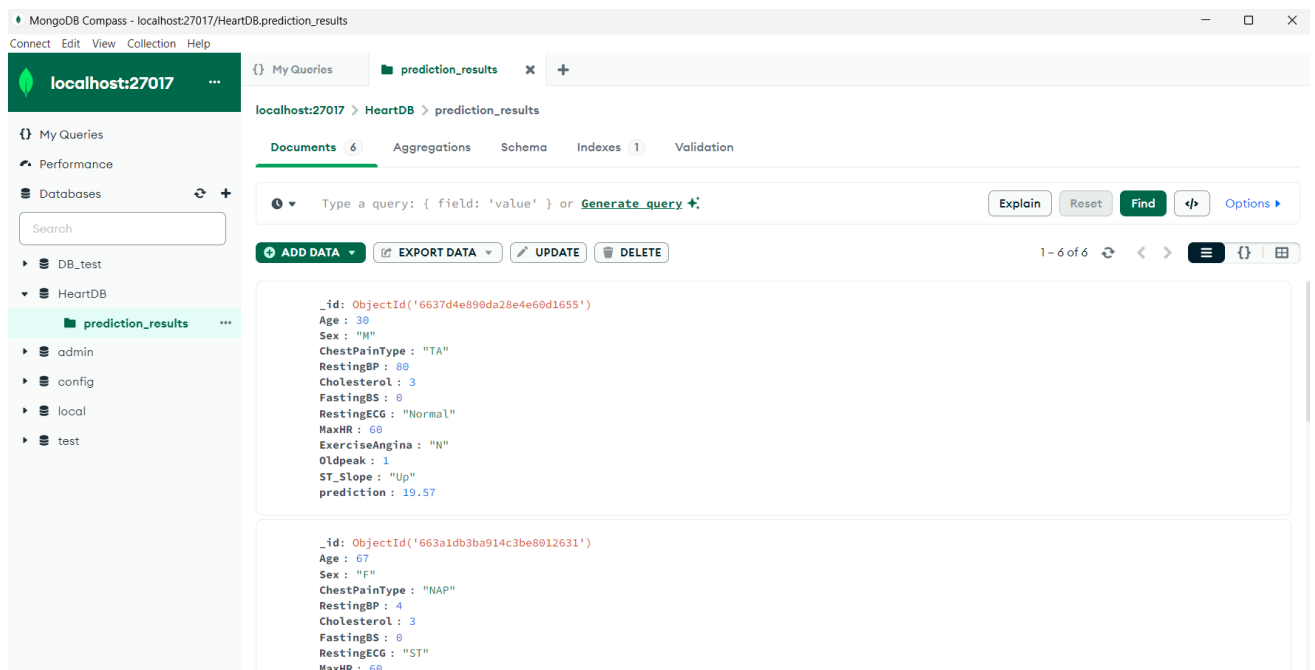


Рисунок 3.2 – Вигляд бази даних HeartDB

4. Після створення бази даних HeartDB та колекції prediction_results, можна використовувати R для збереження даних у MongoDB.

Для взаємодії з MongoDB будемо використовувати бібліотеку mongolite. Якщо пакет не встановлено, встановимо пакет командою

`install.packages("mongolite")` та завантажмо пакет командою `library(mongolite)`.

Налаштуємо параметри підключення до MongoDB (рис.3.3):

Задаємо адресу хосту MongoDB сервера, ім'я користувача та пароль для автентифікації.

```
options(mongodb = list(
  "host" = "mongodb://localhost:27017",
  "username" = "myuser",
  "password" = "mypassword"
))
```

Рисунок 3.3 – Параметри підключення до MongoDB

Визначаємо ім'я бази даних HeartDB та ім'я колекції prediction_results, в яку будемо зберігати дані (рис 3.4).

```
databaseName <- "HeartDB"
collectionName <- "prediction_results"
```

Рисунок 3.4 – Визначення ім'я бази даних та ім'я колекції

Визначаємо функцію для збереження даних у MongoDB (рис. 3.5).

```
saveData <- function(data) {
  # Підключення до бази даних
  db <- mongo(collection = collectionName, db = databaseName, url = options()$mongodb$host)

  # Вставка даних у колекцію MongoDB у вигляді data.frame
  result <- db$insert(data)
  print(result)
}
```

Рисунок 3.5 – Визначення функції для збереження даних

Функція saveData відповідає за збереження даних у MongoDB, функція mongo створює з'єднання з базою даних, функція insert зберігає дані у колекцію. Результат операції вставки зберігається у змінну result.

3.2 Розробка клієнтської частини

Для створення клієнтської частини використовувався фреймворк Shiny в R. Клієнтська частина складається з компонентів інтерфейсу користувача ui (user interface). Розглянемо детальний опис компонентів інтерфейсу користувача.

1. Встановлюємо заголовок вебзастосунку (рис.3.6.)

```
ui <- fluidPage(
  titlePanel("Інформаційна система обліку та аналізу медичних даних"),
```

Рисунок 3.6 – Код заголовка вебзастосунку

2. Задаємо вкладки вебзастосунку – tabsetPanel.
- головна вкладка – містить загальну інформацію про систему (рис. 3.7).

```
tabPanel(icon("home"),|
  mainPanel(
    tags$h3(class = "title", "Система призначена для збору та аналізу даних оцінки ризику серцевої недостатності"),
    h4("Step 1 - Шкала оцінки ризику серцевої недостатності"),
    tags$h4(class = "stp", "Обґрунтування застосування шкали для оцінки ризику серцевої недостатності"),
    h4("Step 2 - Heart Dataset для аналізу даних"),
    h4("Step 3 - Візуалізація даних"),
    h4("Step 4 - Побудова моделі логістичної регресії для прогнозування ймовірності розвитку серцевої недостатності"),
    h4("Step 5 - ROC-аналіз для оцінки прогностичної сили створеної моделі"),
  ),
```

Рисунок 3.7 – Код головної вкладки

- Step 1: Оцінка ризику серцевої недостатності (рис.3.8), включає введення даних пацієнта (вік, стать, тип болю в грудях тощо) для оцінки ризику серцевої недостатності.

```
tabPanel("Step 1",
  tags$h3(class = "title", "Оцінка ризику серцевої недостатності"),
  sidebarLayout(
    sidebarPanel(
      numericInput("Age", "Вік пацієнта", value = NULL, min = 1, max = 100),
      selectInput("Sex", "Стать пацієнта", c("", "M", "F")),
      selectInput("ChestPainType", "Тип болю в грудях", c("", "TA", "ATA", "NAP", "ASY")),
      numericInput("RestingBP", "Артеріальний тиск у стані спокою, мм рт.ст.", value = NULL, min = 0, max = 200),
      numericInput("Cholesterol", "Холестерин сировотки, мм/дл", value = NULL, min = 0, max = 700),
      selectInput("FastingBS", "Рівень цукру в крові натщесерце, 1: якщо > 120 мг/дл, 0: у противному випадку", c("", 0, 1)),
      selectInput("RestingECG", "ЕКГ в стані спокою", c("", "Normal", "LVH", "ST")),
      numericInput("MaxHR", "Максимальна частота пульсу", value = NULL, min = 60, max = 210),
      selectInput("ExerciseAngina", "стенокардія, яка спричинена фізичним навантаженням", c("", "N", "Y")),
      numericInput("Oldpeak", "ST - значення, яке виміряно при депресії", value = NULL, min = -3.0, max = 6.5),
      selectInput("ST_Slope", "Нахил пікового сегмента ST при навантаженні", c("", "Up", "Flat", "Down")),
      actionButton("predictBtn", "Отримати результат", class = "btn-lg")
    ),
    mainPanel(
      textOutput('result')
    )
  ),
```

Рисунок 3.8 – Код step 1: Оцінка ризику серцевої недостатності

– Step 2: Heart Dataset, відображає дані, які використовуються для оцінки ризику серцевої недостатності (рис.3.9).

```
tabPanel("Step 2 - Heart Dataset",
  tags$style(HTML("
    .title {
      color: blue;
      font-weight: bold;
      text-align: center;
    }
    .stp {
      #color: blue;
      font-weight: bold;
      text-align: center;
    }
    .description {
      #color: green;
      #font-style: italic;
      text-align: center;
    }
    .output-text {
      font-size: 20px;
      text-align: center;
    }
  ")),
  tags$h3(class = "title", "Heart Dataset"),
  tags$p(class = "description", "Для оцінки ризику серцевої недостатності на основі відкритого медичного датасету була розроблена модель класифікації з використанням методу логістичної регресії. Медичний датасет був сформований шляхом об'єднання декількох незалежних датасетів, які раніше не були об'єднані. | Людям із серцево-судинними захворюваннями потрібно раннє виявлення та лікування, у цьому модель машинного навчання може надати велику допомогу."),
  fluidRow(
    column(width = 3, choices = c(10, 15, 20), selected = 10)
  ),
  fluidRow(
    column(width = 12, DTOutput("myTable"))
  ),
  DTOutput("myTable")),
```

Рисунок 3.9 – Код step 2: Heart Dataset

– Step 3.1: Візуалізація цільової змінної, відображає діаграми (стовпчикова або кругова) для візуалізації даних (рис. 3.10).

```
tabPanel("Step 3.1",
  tags$h3(class = "title", "Візуалізація цільової змінної"),
  theme = bs_theme(version = 5, bootswatch = "minty"),
  selectInput("plotType", "Виберіть тип діаграми:", choices = c("Bar", "Pie")),
  plotOutput("plot")),
```

Рисунок 3.10 – Код step 3.1: Візуалізація цільової змінної

– Step 3.2: Матриця графіків взаємодії змінних, відображає матрицю графіків для аналізу взаємозв'язків між змінними (рис.3.11).

```
tabPanel("Step 3.2",
  tags$h3(class = "title", "Матриця графіків взаємодії змінних"),
  plotOutput("dataplot")),
```

Рисунок 3.10 – Код step 3.2: Матриця графіків взаємодії змінних

– Step 3.3: Теплова карта кореляції, відображає теплову карту кореляції між змінними.

```
tabPanel("Step 3.3",
  tags$h3(class = "title", "Теплова карта кореляції"),
  plotOutput("correlationPlot")),
```

Рисунок 3.11 – Код step 3.3: Теплова карта кореляції

– Step 4: Data Analysis, включає результати аналізу даних та побудову моделі логістичної регресії.

```
tabPanel("Step 4 - Data Analysis",
  tags$h3(class = "title", "Побудова моделі логістичної регресії для прогнозування ймовірності розвитку серцевої недостатності"),
  verbatimTextOutput("modelSummary")),
```

Рисунок 3.12 – Код step 4: Data Analysis

– Step 5: ROC-аналіз, відображає ROC-криву для оцінки прогностичної сили моделі.

```
tabPanel("Step 5",
  tags$h3(class = "title", "ROC-аналіз для оцінки створеної моделі"),
  plotOutput("dataPlot"))
```

Рисунок 3.13 – Код step 5: ROC-аналіз

Цей код створює інтерактивний вебзастосунок з використанням Shiny, який дозволяє вводити дані пацієнтів, аналізувати їх та візуалізувати результати. Можливо розширювати функціональність сервера, додаючи логіку для обробки даних, побудови моделей та інших операцій, які потрібні для системи.

3.3 Розробка серверної частини

Для створення серверної частини використовувався фреймворк Shiny в R. Серверна частина складається з функції сервера (server), яка визначає логіку обробки даних, взаємодії з базою даних, виконання прогнозувань та відображення результатів. Вона включає в себе реактивні об'єкти та вирази для автоматичного оновлення відповідей на зміни вхідних даних, а також використовує модель прогнозування для генерації прогнозів на основі введених користувачем даних.

Ось ключові складові серверної частини:

1. Оголошення логіки сервера. Це початок визначення серверної частини вебзастосунку. Створюється функцію server, яка визначає весь функціонал, пов'язаний з обробкою введених даних, аналізом та відображенням результатів. Вона приймає два аргументи: input і output. input містить значення вхідних елементів інтерфейсу користувача, а output використовується для визначення вихідних елементів, які відображаються користувачу.

```
server <- function(input, output) {...}
```

Ця функція є основним елементом серверної частини.

2. Створення моделі логістичної регресії. Визначається модель на основі даних heart, за допомогою функції логістичної регресії glm() (рис.3.14), яка буде використовуватися для прогнозування ймовірності розвитку серцевої недостатності на основі введених даних користувача.

```
model <- glm(formula=HeartDisease~Age+factor(Sex)+
  factor(ChestPainType)+RestingBP+Cholesterol+FastingBS+
  factor(RestingECG)+MaxHR+factor(ExerciseAngina)+Oldpeak+
  factor(ST_Slope), data=heart, family=binomial)
```

Рисунок 3.14 – Модель логістичної регресії

3. Обробка події, цей фрагмент коду виконується, коли користувач натискає кнопку "Отримати результат".

```
observeEvent(input$predictBtn, {...})
```

3.1. Перевірка заповнення всіх полів (рис. 3.15), цей код перевіряє, чи всі поля введення заповнені. Якщо якесь поле не заповнено, відображається повідомлення з проханням заповнити всі поля. Якщо ні, виводиться повідомлення про помилку.

```
if(any(sapply(input, function(x) is.null(x) || is.na(x)))) {  
  output$result <- renderText({  
    "Будь ласка, заповніть усі поля"  
  })  
}
```

Рисунок 3.15 – Код перевірки заповнення всіх полів

3.2. Інакше створюється новий набір даних на основі введених користувачем даних (рис. 3.16).

```
else {  
  newdata <- data.frame(  
    Age = as.numeric(input$Age),  
    Sex = as.factor(input$Sex),  
    ChestPainType = as.factor(input$ChestPainType),  
    RestingBP = as.numeric(input$RestingBP),  
    Cholesterol = as.numeric(input$Cholesterol),  
    FastingBS = as.numeric(input$FastingBS),  
    RestingECG = as.factor(input$RestingECG),  
    MaxHR = as.numeric(input$MaxHR),  
    ExerciseAngina = as.factor(input$ExerciseAngina),  
    Oldpeak = as.numeric(input$Oldpeak),  
    ST_Slope = as.factor(input$ST_Slope)  
  )  
}
```

Рисунок 3.16 – Код створення нового набору даних

3.3. Прогнозування за допомогою моделі, цей код використовує модель для прогнозування ймовірності серцевої недостатності:

```
pred <- predict(model, newdata, type = "response")
```

3.4. Додавання прогнозу до даних та збереження в MongoDB, код (рис. 3.17) додає прогноз до нового набору даних і зберігає його в базі даних MongoDB.

```
newdata$prediction <- round(pred*100, 2)  
saveData(newdata)
```

Рисунок 3.17 – Код додавання прогнозу до даних та збереження в MongoDB

3.5. Відображення результату, цей фрагмент коду відображає прогнозовану ймовірність серцевої недостатності.

```
output$result <- renderText({
  paste('Ймовірність серцевої недостатності: ', round(pred*100, 2))
})
```

Рисунок 3.17 – Код відображення результату прогнозу

Для обґрунтування прогностичності моделі використовуються кроки 4-6.

4. Відображення таблиці Heart. На рис. 3.18 фрагмент коду, який створює інтерактивну таблицю для відображення даних heart.

```
output$myTable <- renderDT({
  datatable(
    heart,
    filter = 'top',
    options = list(
      autowidth = TRUE,
      scrollX = TRUE,
      scrollY = "400px",
      scrollCollapse = TRUE,
      paging = TRUE,
      lengthMenu = list(c(5, 15, 20, -1), c('5', '15', '20', ' '))
    )
  ), server = FALSE)
```

Рисунок 3.18 – Код створення інтерактивної таблиці Heart

5. Візуалізація.

5.1. Візуалізація розповсюдження серцевих захворювань, в цьому фрагменту коду (рис. 3.19) створюється стовпчаста або кругова діаграму залежно від вибору користувача.

5.2. Візуалізація взаємодії змінних. На рисунку 3.20 фрагмент коду матриці графіків для візуалізації взаємодії між змінними.

5.3. Теплова карта кореляції. На рисунку 3.21 представлено фрагмент коду створення теплової карти кореляцій між числовими змінними.

```

output$plot <- renderPlot({
  if (input$plotType == "Bar") {
    ggplot(data = as.data.frame(types), aes(x = Var1, y = Freq)) +
      geom_bar(stat = "identity", fill = c("lightgreen", "gray")) +
      labs(x = "Хвороба серця", y = "Частота", title = "Розповсюдження серцевих захворювань") +
      theme_minimal() +
      coord_flip() +
      guides(fill=guide_legend(title="Хвороба серця"))
  } else if (input$plotType == "Pie") {
    ggplot(data = as.data.frame(types), aes(x = "", y = Freq, fill = factor(Var1))) +
      geom_col(width = 1) +
      coord_polar("y", start=0) +
      theme_void() +
      labs(title = "Розповсюдження серцевих захворювань") +
      geom_text(aes(label = paste0(round(Freq/sum(Freq)*100, 1), "%"),
                    position = position_stack(vjust = 0.5)) +
      theme(legend.position = "right") +
      guides(fill=guide_legend(title="Хвороба серця"))
  }
})

```

Рисунок 3.19 – Код візуалізація розповсюдження серцевих захворювань

```

output$dataplot <- renderPlot({
  heart$HeartDisease <- as.factor(heart$HeartDisease)
  df_subset <- heart[, c("Age", "Sex", "Cholesterol", "MaxHR", "HeartDisease")]
  p <- ggpairs(df_subset, aes(color = HeartDisease),
              lower = list(continuous = wrap("points", alpha = 0.3)),
              diag = list(continuous = wrap("densityDiag")))
  p <- p + theme(text = element_text(size = 15))
  print(p)
})

```

Рисунок 3.20 – Код матриці графіків для візуалізації взаємодії між змінними

```

output$correlationPlot <- renderPlot({
  num_columns <- sapply(heart, is.numeric)
  numeric_data <- heart[, num_columns]
  corr_matrix <- cor(numeric_data)
  corrplot::corrplot(corr_matrix, method="color", type="upper",
                    addCoef.col = "black",
                    tl.col="black", tl.srt=45,
                    diag=FALSE)
})

```

Рисунок 3.21 – Код створення теплової карта кореляції

6. Аналіз моделі та ROC-крива

6.1. Відображення підсумкової інформації про модель (рис. 3.22).

```

output$modelSummary <- renderPrint({
  model_summary <- capture.output(summary(model))
  cat(model_summary, sep = "\n")
  cat("\n")
})

```

Рисунок 3.22 – Код відображення підсумкової інформації про модель

6.2. ROC-крива для оцінки моделі. На рисунку 3.23 показано фрагмент коду для створення та відображення ROC-кривої для оцінки ефективності моделі.

```
output$dataPlot <- renderPlot({
  heart$predict <- predict(model, newdata = heart, type = "response")
  pred <- prediction(heart$predict, heart$HeartDisease)
  perf <- performance(pred, "tpr", "fpr")
  plot(perf, col = "blue", lwd = 2, xlab = "1 - специфічність", ylab = "чутливість", main = "ROC-кривая")
  lines(c(0,1),c(0,1), lty = 3, lwd = 1, col = "red")
  legend("bottomright", legend = c("AUC = 0.93", "чутливість = 0.90", "специфічність = 0.85"),
        col = c("blue", "red"), lty = c(1, 3), text.font = 1)
  grid(col = "gray", lty = "dotted")

  cp = cutpointr(heart$predict, heart$HeartDisease, pos_class = 1, neg_class = 0)
  print(cp)
})
```

Рисунок 3.23 – Код створення та відображення ROC-кривої

Серверна частина виконує ключові функції для обробки введених користувачем даних, побудови та використання моделі для прогнозування, а також створення різних візуалізацій для аналізу даних. Кожен з цих кроків є важливою частиною загальної логіки роботи серверної частини вебзастосунку. Вона також може бути розширена шляхом додавання нових функцій і вдосконалення існуючих, таких як: використання інших алгоритмів машинного навчання для підвищення точності прогнозів, додавання нових візуалізацій і статистичних методів для більш детального аналізу, оптимізація процесу введення даних і відображення результатів для підвищення зручності використання.

3.4 Функціональне тестування системи

Функціональне тестування є одним з ключових етапів розробки програмного забезпечення. Цей процес дозволяє переконатися, що всі функції системи працюють відповідно до встановлених вимог і очікуваних сценаріїв. Основна мета функціонального тестування полягає в виявленні та усуненні можливих помилок, що забезпечує надійність та правильну роботу всієї розробленої системи.

При завантаженні системи користувач бачить домашню сторінку (рис. 3.24).

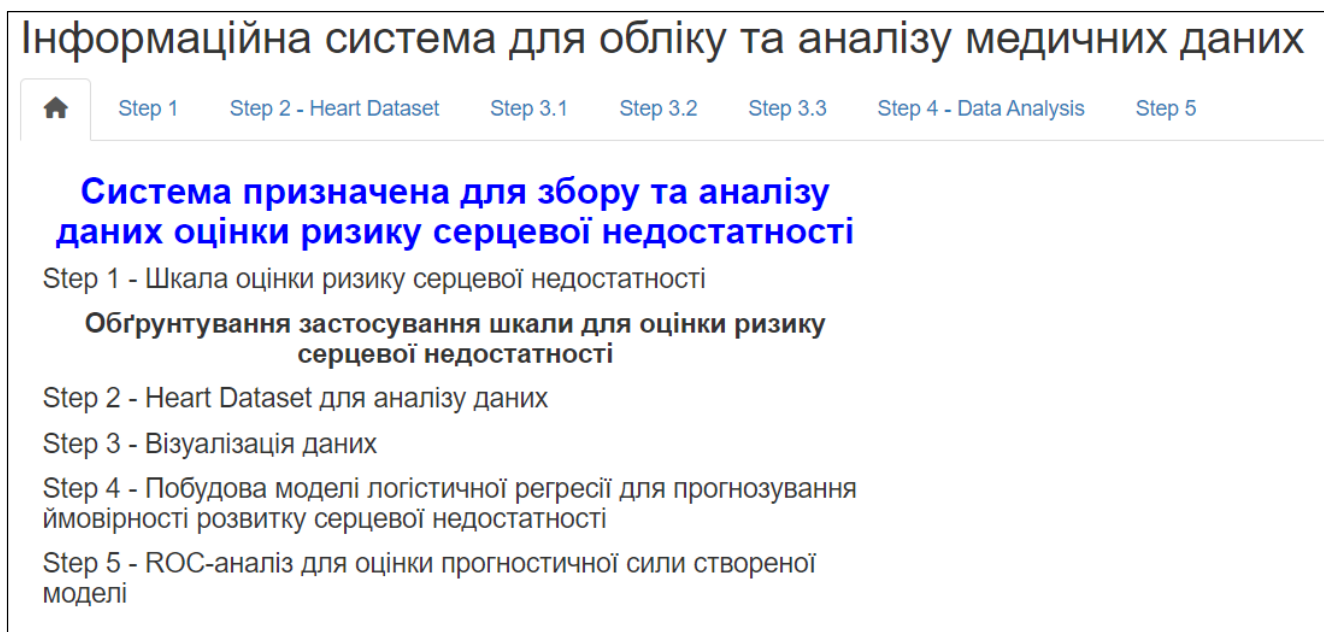


Рисунок 3.23 – Домашня сторінка

Сторінка містить послідовність кроків, які користувач повинен виконати. Кожен крок має свій номер та опис, який пояснює, що відбувається на цьому етапі. Кроки включають оцінку ризику серцевої недостатності, обґрунтування вибору шкали для оцінки ризику, аналіз даних, побудову моделі та ROC-аналіз для оцінки прогностичної сили створеної моделі оцінки ризику серцевої недостатності.

Сторінка містить послідовність кроків, які користувач повинен виконати. Кожен крок має свій номер та опис, який пояснює, що відбувається на цьому етапі. Кроки включають оцінку ризику серцевої недостатності, обґрунтування вибору шкали для оцінки ризику, аналіз даних, побудову моделі та ROC-аналіз для оцінки прогностичної сили створеної моделі.

Перевіримо виконання тестових сценаріїв.

1. Перевірка створення нового прогнозу включає:
 - перевірку коректності введення даних (рис 3.24);
 - заповнення всіх необхідних полів, таких як вік, стать, тип болю в грудях тощо;
 - натискання кнопки “Отримати результат”;
 - перевірка коректності та правильного відображення прогнозу ймовірності серцевої недостатності (Додаток А);

Вік пацієнта

Стать пацієнта

Тип болю в грудях

TA
ATA
NAP
ASY

Рисунок 3.24 – Перевірка коректності введення даних

- у випадку не заповнення одного із полів перевірка відображення повідомлення про помилку (Додаток А).
- 2. Перевірка збереження даних.
 - перевірка збереження введених даних та результату прогнозу в базі даних (рис. 3.25).

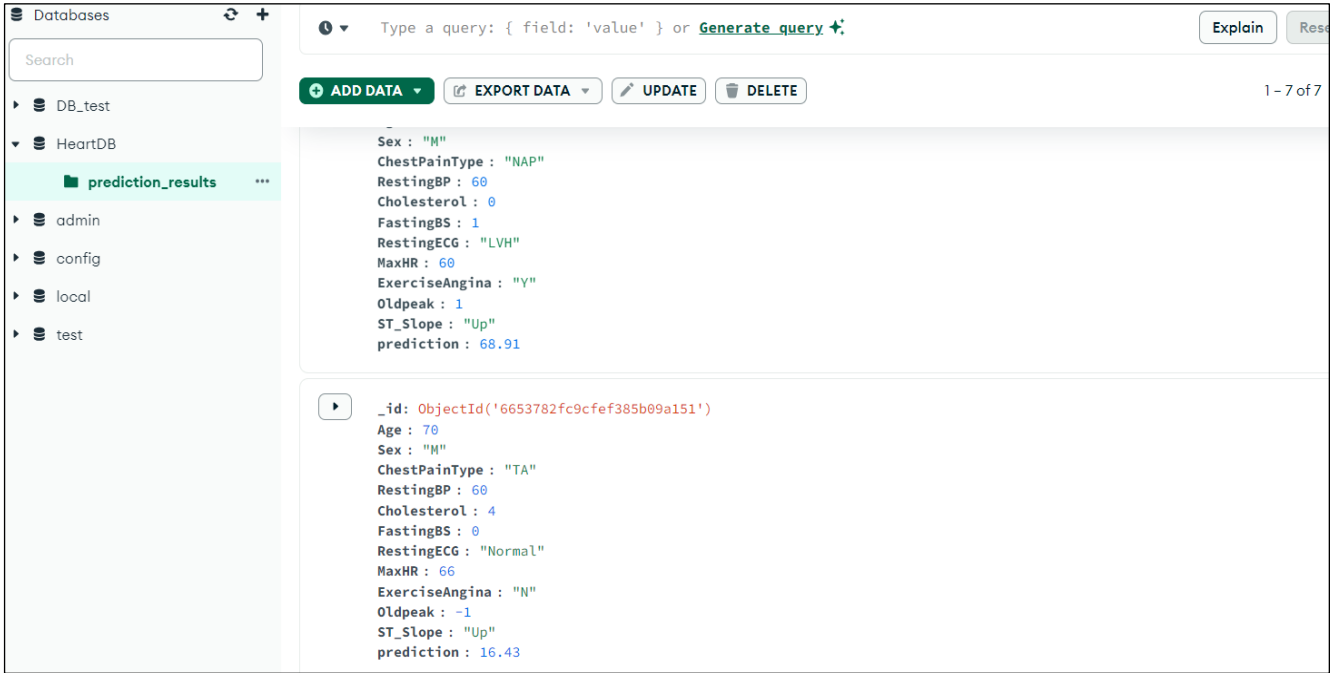


Рисунок 3.24 – Перевірка збереження даних в базі даних

3. Перевірка відображення коректних даних у таблиці, яка використовується для створення моделі (рис 3.25).

Heart Dataset

Для оцінки ризику серцевої недостатності на основі відкритого медичного датасету була розроблена модель класифікації з використання методу логістичної регресії. Медичний датасет був сформований шляхом об'єднання декількох незалежних датасетів, які раніше не були об'єднані. Людям із серцево-судинними захворюваннями потрібно раннє виявлення та лікування, у цьому модель машинного навчання може надати велику допомогу.

Show entries Search:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak
1	40	M	ATA	140	289	0	Normal	172	N	0
2	49	F	NAP	160	180	0	Normal	156	N	1
3	37	M	ATA	130	283	0	ST	98	N	0
4	48	F	ASY	138	214	0	Normal	108	Y	1.5
5	54	M	NAP	150	195	0	Normal	122	N	0

Showing 1 to 5 of 918 entries Previous 2 3 4 5 ... 184 Next

Рисунок 3.25 – Перевірка відображення коректних даних у таблиці

Для створення моделі логістичної регресії були використані реальні дані пацієнтів. Було проаналізовано набір даних Kaggle [24], який був створений шляхом об'єднання різних наборів даних, вже доступних незалежно, але які раніше не об'єднувалися. У цьому наборі даних 5 наборів даних про серце об'єднані за 11 спільними ознаками, що робить його найбільшим набором даних про серцеві захворювання, доступним для дослідницьких цілей.

У ньому використовуються такі п'ять наборів даних:

Клівленд: 303 спостереження, Угорський: 294 зауваження, Швейцарія: 123 спостереження, Лонг-Біч, Вірджинія: 200 спостережень, Набір даних Stalog (серце): 270 спостережень. Усього: 1190 спостережень. У результаті дослідницького аналізу було виявлено дубльовані 272 спостереження. Фінальний датасет – 918 спостережень.

Дані та опис атрибутів :

– Вік: вік пацієнта [років]

- Стать: стать пацієнта [М: Чоловік, Ж: Жінка].
 - ChestPainType: тип болю в грудях [ТА: типова стенокардія, АТА: атипова стенокардія, NAR: неангінальний біль, ASY: безсимптомний].
 - RestingBP: артеріальний тиск у стані спокою [мм рт. ст.].
 - Холестерин: холестерин сироватки [мм/дл]
 - FastingBS: рівень цукру в крові натщесерце [1: якщо FastingBS > 120 мг/дл, 0: в іншому випадку]
 - ЕКГ спокою: результати електрокардіограми спокою [Нормальний: нормальний, ST: наявність аномалій ST-T (інверсія зубця Т та/або підйом чи депресія ST > 0,05 мВ), ГЛШ: вірогідна чи визначена гіпертрофія лівого шлуночка за критеріями Естеса].
 - MaxHR: досягнута максимальна частота пульсу [числове значення від 60 до 202].
 - Стенокардія при фізичному навантаженні: стенокардія, викликана фізичним навантаженням [Д: Так, Н: Ні]
 - Oldpeak: oldpeak = ST [Числове значення, виміряне під час депресії]
 - ST_Slope: нахил пікового сегмента ST при навантаженні [Вгору: вгору, Плоский: плоский, Вниз: вниз]
 - HeartDisease: вихідний клас [1: захворювання серця, 0: нормальне]
4. Перевірка створення діаграм. Кроки 3.1, 3.2, 3.3.

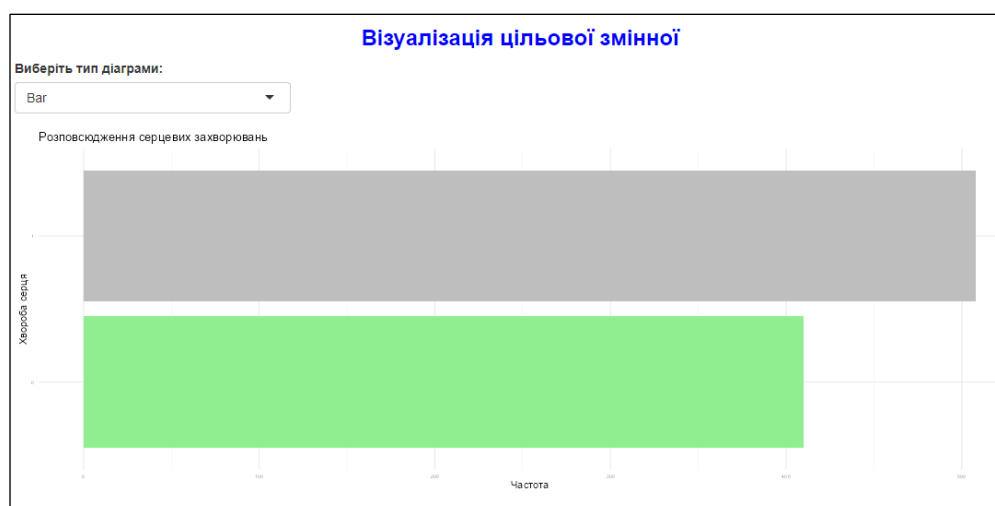


Рисунок 3.26 – Відображення цільової змінної в вигляді стовпчастої діаграми

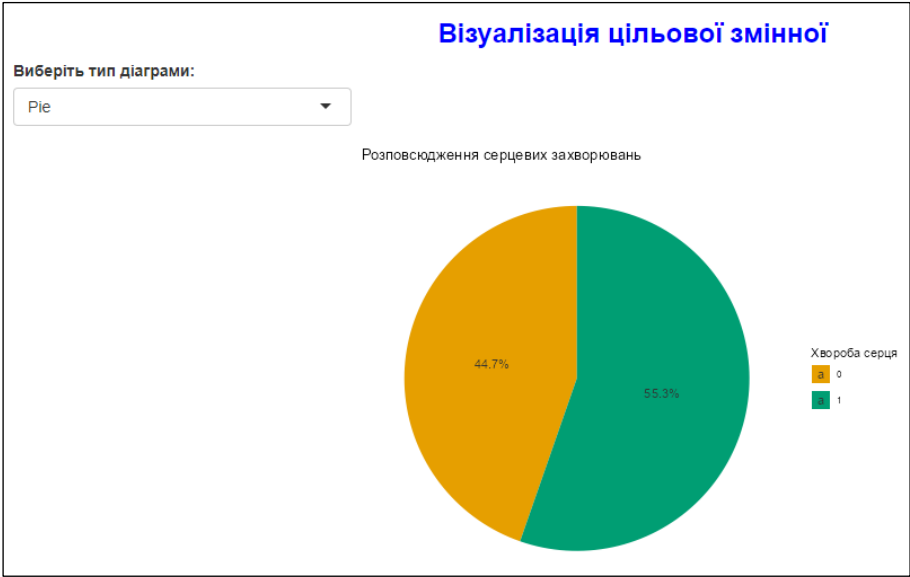


Рисунок 3.27 – Відображення цільової змінної в вигляді кругової діаграми

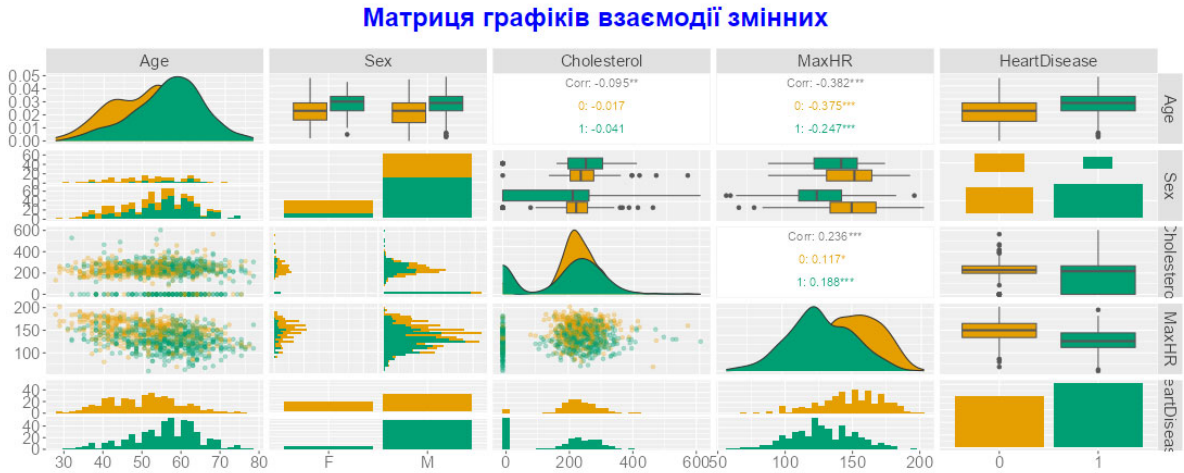


Рисунок 3.28 – Відображення матриці взаємодії змінних



Рисунок 3.29 – Відображення теплової карти кореляції

5. Перевірка створення моделі логістичної регресії.

Coefficients:				
	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-1.163656	1.416003	-0.822	0.411197
Age	0.016550	0.013197	1.254	0.209803
factor(Sex)M	1.466477	0.279834	5.241	1.60e-07 ***
factor(ChestPainType)ATA	-1.830289	0.326293	-5.609	2.03e-08 ***
factor(ChestPainType)NAP	-1.685682	0.266001	-6.337	2.34e-10 ***
factor(ChestPainType)TA	-1.488392	0.432572	-3.441	0.000580 ***
RestingBP	0.004194	0.006010	0.698	0.485296
Cholesterol	-0.004115	0.001087	-3.785	0.000154 ***
FastingBS	1.136482	0.274999	4.133	3.59e-05 ***
factor(RestingECG)Normal	-0.177033	0.271925	-0.651	0.515022
factor(RestingECG)ST	-0.268546	0.350020	-0.767	0.442945
MaxHR	-0.004288	0.005023	-0.854	0.393249
factor(ExerciseAngina)Y	0.900292	0.244513	3.682	0.000231 ***
Oldpeak	0.380643	0.118466	3.213	0.001313 **
factor(ST_Slope)Flat	1.453902	0.429086	3.388	0.000703 ***
factor(ST_Slope)Up	-0.994101	0.450196	-2.208	0.027234 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1				
(Dispersion parameter for binomial family taken to be 1)				
Null deviance: 1262.14 on 917 degrees of freedom				
Residual deviance: 594.19 on 902 degrees of freedom				
AIC: 626.19				

Рисунок 3.30 – Результат розрахунку моделі логістичної регресії

	Estimate	2.5 %	97.5 %
(Intercept)	0.3123421	0.01924353	5.0080915
Age	1.0166878	0.99080373	1.0435108
factor(Sex)M	4.3339419	2.52420787	7.5724521
factor(ChestPainType)ATA	0.1603672	0.08317101	0.2998814
factor(ChestPainType)NAP	0.1853181	0.10912842	0.3101553
factor(ChestPainType)TA	0.2257354	0.09556024	0.5234977
RestingBP	1.0042023	0.99236643	1.0161206
Cholesterol	0.9958937	0.99373862	0.9979908
FastingBS	3.1157887	1.83334268	5.3976580
factor(RestingECG)Normal	0.8377518	0.49054068	1.4267101
factor(RestingECG)ST	0.7644903	0.38438351	1.5196299
MaxHR	0.9957213	0.98595680	1.0055913
factor(ExerciseAngina)Y	2.4603214	1.52397001	3.9803497
Oldpeak	1.4632257	1.16323952	1.8520518
factor(ST_Slope)Flat	4.2797800	1.80966730	9.7944350
factor(ST_Slope)Up	0.3700562	0.14989787	0.8801516

Рисунок 3.31 – Розрахунок відношення шансів

З розрахунку моделі логістичної регресії та відношення шансів можна зробити висновок, що факторами, які впливають на розвиток серцевої недостатності є: стать пацієнта, стенокардія, холестерин, рівень цукру в крові, депресія.

Для визначення оптимального порогу класифікації пацієнтів, які мають ризик серцевої недостатності, розраховано коефіцієнт *optimal cutpoint* (рис. 3.32), який дорівнює 0,517598.

```

direction optimal_cutpoint method      sum_sens_spec      acc sensitivity specificity      AUC
<chr>      <dbl> <chr>                  <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
1 >=      0.517598 maximize_metric      1.74201 0.873638      0.895669      0.846341 0.932672

```

Рисунок 3.32 – Розрахунок коефіцієнта *optimal cutpoint*

Якщо передбачена ймовірність серцевої недостатності перевищує цей поріг, пацієнт класифікується як ризик серцевої недостатності.

Точність моделі становить приблизно 0,874, що означає, що приблизно 87,4% передбачень моделі вірні. Чутливість (або істинно позитивний коефіцієнт) становить приблизно 0,896, що означає, що модель правильно ідентифікує приблизно 89,6% пацієнтів із серцевою недостатністю. Специфічність (або істинно-негативний коефіцієнт) становить приблизно 0,846, що означає, що модель правильно ідентифікує приблизно 84,6% пацієнтів без серцевої недостатності.

6. Перевірка правильності побудови ROC-кривої. Для оцінки прогностичної сили створеної моделі побудуємо ROC-кривою. Площа під ROC-кривою є міркою якості моделі бінарної класифікації. Оцінку якості отриманої регресійної моделі за допомогою ROC-аналізу наведено на рисунку 3.33.

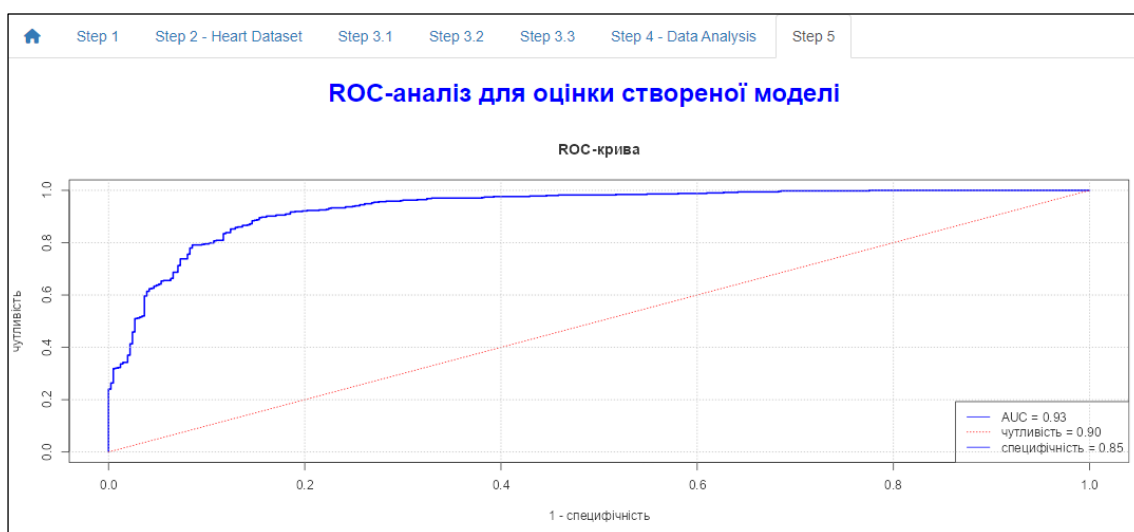


Рисунок 3.33 – Перевірка правильності побудови ROC-кривої

Площа під ROC-кривою (AUC) становить приблизно 0,93. Значення AUC, близьке до 1, це вказує на відмінну якість моделі, тобто модель демонструє хорошу продуктивність.

Таким чином, функціональне тестування підтвердило, що розроблена система відповідає всім вимогам і успішно виконує свої функції. Вона надає точні та надійні результати, що є критично важливим для медичних інформаційних систем. Успішне функціональне тестування також сприяє підвищенню довіри користувачів до системи та забезпечує основу для її подальшого впровадження та використання в медичній практиці.

3.5 Висновки до розділу 3

У третьому розділі розглядається розробка інформаційної системи для обліку та аналізу медичних даних. Основні компоненти, розглянуті у цьому розділі, включають розробку бази даних для обліку, зберігання та аналізу медичних даних, розробку клієнтської частини та розробку серверної частини.

Для розробки клієнтської частини використовувався фреймворк Shiny, і було надано детальний опис компонентів інтерфейсу користувача. Розробка серверної частини виконувалася з використанням фреймворку Shiny, і були розглянуті основні складові частини серверної частини.

Було проведено функціональне тестування, яке є критично важливим етапом у процесі розробки програмного забезпечення, зокрема для інформаційної системи для обліку та аналізу медичних даних. Проведення цього тестування дозволило переконатися, що всі функції системи працюють коректно та відповідають встановленим вимогам та очікуванням користувачів.

ВИСНОВКИ

Кваліфікаційна робота присвячена розробці інформаційної системи для обліку та аналізу медичних даних. В процесі виконання роботи зроблено аналіз предметної області та конкурентів, визначено функціональні вимоги, спроектована інформаційна система, розроблена база даних для обліку медичних даних, клієнтська та серверна частини.

У першому розділі було проаналізовано специфіку розробки інформаційних систем для медичної галузі. Визначено, що основним інструментом для аналізу даних у таких системах є моделі машинного навчання. Проведено детальний аналіз існуючих аналогів за допомогою SWOT-аналізу, який виявив як переваги, так і недоліки обраних систем. На основі цього аналізу було сформульовано функціональні вимоги до розробки ефективної інформаційної системи для обліку та аналізу медичних даних.

У другому розділі було детально розглянуто технології та інструменти, використані для розробки інтелектуальної системи. Для створення фронтенду використовувались сучасні технології HTML, CSS, Bootstrap, а також клієнтська частина User Interface фреймворку Shiny. Для бекенду застосовувалась серверна частина Server фреймворку Shiny та реактивне програмування, що дозволяє автоматично оновлювати вміст інформаційної системи при зміні вхідних даних. Для обліку, обробки та зберігання даних використовувалась нереляційна база даних MongoDB.

Було розглянуто математичну модель логістичної регресії, яка використовується для бінарної класифікації і передбачення ймовірності настання серцевої недостатності на основі клінічних параметрів. Представлено схему аналізу медичних даних із використанням логістичної регресії, діаграму послідовності (Sequence Diagram) та ментальну карту (MindMap) для кращого розуміння структури інформаційної системи.

У третьому розділі було детально розглянуто розробку інформаційної системи, включаючи розробку бази даних, клієнтської та серверної частин з

використанням фреймворку Shiny. Було проведено функціональне тестування системи, яке підтвердило коректність роботи всіх функцій та відповідність встановленим вимогам і очікуванням користувачів.

У результаті проведеного дослідження було розроблено ефективну інформаційну систему для обліку та аналізу медичних даних з використанням сучасних технологій та інструментів. Система дозволяє здійснювати точне прогнозування серцевої недостатності, що сприяє ранньому виявленню ризиків та покращенню медичного обслуговування пацієнтів. Проведене функціональне тестування підтвердило надійність та правильність роботи розробленої системи. Таким чином, створена інформаційна система є значним внеском у сферу медичних інформаційних технологій та має потенціал для подальшого розвитку та впровадження у медичній практиці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Міністерства охорони здоров'я України. Що таке медичні інформаційні системи та які послуги вони надають. URL: <https://moz.gov.ua/article/news/scho-take-medichni-informacijni-sistemi-ta-jaki-poslugi-voni-nadajut>
2. Електронна система охорони здоров'я в Україні. Цифрова трансформація системи охорони здоров'я. URL: <https://ehealth.gov.ua/>
3. Державний заклад "Центр медичної статистики Міністерства охорони здоров'я": офіційний сайт. URL: <http://medstat.gov.ua>
4. Про захист персональних даних: Закон України від 01.06.2010 № 2297-VI. Дата оновлення 27.04.2024 р. № 3585-IX. URL: <https://zakon.rada.gov.ua/laws/show/2297-17>
5. Деякі питання електронної системи охорони здоров'я: Постанова Кабінету Міністрів України; Порядок від 25.04.2018 № 411. Дата оновлення від 26.03.2024 р. №315-2024-п. URL: <https://zakon.rada.gov.ua/laws/show/411-2018>
6. Gretzky W. Strategic planning and SWOT analysis. *Essentials of strategic planning in healthcare*. 1(12), 2010. С. 91-108. URL: <http://surl.li/ttmfv>
7. Framingham Risk Score for Hard Coronary Heart Disease. URL: <https://www.mdcalc.com/calc/38/framingham-risk-score-hard-coronary-heart-disease#evidence>
8. Шкала PRECISE-DAPT, Шкала DAPT. URL: <https://clincasequest.academy/precise-dapt-scale-dapt-scale/>
9. Шкала SCORE. URL: <https://cardioprostir.com.ua/applications-and-calculators/calculators/score>
10. Шкала HAS-BLED. URL: <https://cardioprostir.com.ua/applications-and-calculators/calculators/has-bled>
11. Functional vs Non-functional Requirements. URL: <http://surl.li/ttfzl>
12. The R Project for Statistical Computing. URL: <https://www.r-project.org/>

13. From an expert statistician's tool kit: R vs Python programming language. URL: <http://surl.li/ttiai>
14. IQVIA. From an expert statistician's tool kit: R vs Python programming language. URL: <http://surl.li/ttiai>
15. RStudio Desktop. URL: <https://posit.co/download/rstudio-desktop>
16. Shiny. URL: <https://shiny.posit.co/>
17. MongoDB. URL: <https://www.mongodb.com/>
18. Available CRAN Packages By Name URL https://cran.r-project.org/web/packages/available_packages_by_name.html
19. Topic 10 Logistic Regression. URL: <http://surl.li/tvmly>
20. Logistic Regression in Machine Learning. URL: <http://surl.li/tvmnb>
21. Wu, TingTing et al. "Logistic regression technique is comparable to complex machine learning algorithms in predicting cognitive impairment related to post intensive care syndrome." Scientific reports vol. 13,1 2485. 11 Feb. 2023, doi:10.1038/s41598-023-28421-6
22. Моделювання програмного забезпечення [Електронний ресурс]: навч.-метод. посібник / укл.: Манаков С.Ю., Трофименко О.Г., Лобода Ю.Г., Дика А.І. Одеса: Фенікс, 2023. 145 с. URL: <http://surl.li/tvspr>
23. What is Mind Mapping? What Are Its Uses? URL: <https://www.mindmaps.com/what-is-mind-mapping/>
24. Kaggle. Heart Failure Prediction Dataset. URL: <https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction>

ДОДАТКИ

Додаток А. Результати тестування

Оцінка ризику серцевої недостатності

Ймовірність серцевої недостатності: 16.43

Вік пацієнта

70

Стать пацієнта

M

Тип болю в грудях

ТА

Артеріальний тиск у стані спокою, мм рт.ст.

60

Холестерин сировотки, мм/дл

4

Рівень цукру в крові натщесерце, 1: якщо > 120 мг/дл, 0: у противному випадку

0

ЕКГ в стані спокою

Normal

Максимальна частота пульсу

66

стенокардія, яка спричинена фізичним навантаженням

N

ST - значення, яке виміряно при депресії

-1

Нахил пікового сегмента ST при навантаженні

Up

Отримати результат

Рисунок А.1 – Відображення правильного прогнозу

Оцінка ризику серцевої недостатності

Будь ласка, заповніть усі поля

Вік пацієнта

49

Стать пацієнта

F

Тип болю в грудях

NAP

Артеріальний тиск у стані спокою, мм рт.ст.

Холестерин сировотки, мм/дл

2

Рівень цукру в крові натщесерце, 1: якщо > 120 мг/дл, 0: у противному випадку

1

ЕКГ в стані спокою

Максимальна частота пульсу

стенокардія, яка спричинена фізичним навантаженням

N

ST - значення, яке виміряно при депресії

1

Нахил пікового сегмента ST при навантаженні

Up

Отримати результат

Рисунок А.2 – Відображення повідомлення про помилку

Додаток Б. Програмний код

```

library(shiny)
library(ggplot2)
library(tidyverse)
library(readxl)
library(reshape2)
library(dplyr)
library(ROCR)
library(cutpointtr)
library(shinythemes)
library(DT)
library(mongolite)
library(bslib)
library(thematic)
library(GGally)
options(mongodb = list(
  "host" = "mongodb://localhost:27017"
  "username" = "myuser",
  "password" = "mypassword"
))
databaseName <- "HeartDB"
collectionName <- "prediction_results"
saveData <- function(data) {
  db <- mongo(collection = collectionName,
    db = databaseName, url = "mongodb://localhost:27017")
  print(data)
  result<-db$insert(data)
  print(result)
}
heart <- read.csv("heart.csv", header = TRUE)

server <- function(input, output) {
  model <- glm(formula=HeartDisease~Age+factor(Sex)+

```

```

factor(ChestPainType)+RestingBP+Cholesterol+FastingBS+
  factor(RestingECG)+MaxHR+factor(ExerciseAngina)+Oldpeak+
  factor(ST_Slope), data=heart, family=binomial)
observeEvent(input$predictBtn, {
  if(any(sapply(input, function(x) is.null(x) || is.na(x)))) {
    output$result <- renderText({
      "Будь ласка, заповніть усі поля"
    })
  } else {
    newdata <- data.frame(
      Age = as.numeric(input$Age),
      Sex = as.factor(input$Sex),
      ChestPainType = as.factor(input$ChestPainType),
      RestingBP = as.numeric(input$RestingBP),
      Cholesterol = as.numeric(input$Cholesterol),
      FastingBS = as.numeric(input$FastingBS),
      RestingECG = as.factor(input$RestingECG),
      MaxHR = as.numeric(input$MaxHR),
      ExerciseAngina = as.factor(input$ExerciseAngina),
      Oldpeak = as.numeric(input$Oldpeak),
      ST_Slope = as.factor(input$ST_Slope)
    )

    pred <- predict(model, newdata, type = "response")

    newdata$prediction <- round(pred*100, 2)

    saveData(newdata)

    output$result <- renderText({
      paste('Ймовірність серцевої недостатності: ',
round(pred*100, 2))
    })
  }
})

```

```

output$myTable <- renderDT({
  datatable(
    heart,
    filter = 'top',
    options = list(
      autoWidth = TRUE,
      scrollX = TRUE,
      scrollY = "400px",
      scrollCollapse = TRUE,
      paging = TRUE,
      lengthMenu = list(c(5, 15, 20, -1),
c('5', '15', '20', ' '))
    )
  ), server = FALSE)
types <- table(heart$HeartDisease)
output$plot <- renderPlot({
  if (input$plotType == "Bar") {
    ggplot(data = as.data.frame(types),
aes(x = Var1, y = Freq)) +
    geom_bar(stat = "identity", fill = c("lightgreen", "gray"))+
    labs(x = "Хвороба серця", y = "Частота",
title = "Розповсюдження серцевих захворювань") +
    theme_minimal() +
    coord_flip() +
    guides(fill=guide_legend(title="Хвороба серця"))
  }
  else if (input$plotType == "Pie") {
    ggplot(data = as.data.frame(types),
aes(x = "", y = Freq, fill = factor(Var1))) +
    geom_col(width = 1) +
    coord_polar("y", start=0) +
    theme_void() +
    labs(title = "Розповсюдження серцевих захворювань") +
    geom_text(aes(label = paste0(round(Freq/sum(Freq)*100,

```

```

1), "%")},

      position = position_stack(vjust = 0.5)) +
theme(legend.position = "right") +
guides(fill=guide_legend(title="Характера черта"))
}
})

output$dataplot <- renderPlot({
  heart$HeartDisease <- as.factor(heart$HeartDisease)
  df_subset <- heart[, c("Age", "Sex", "Cholesterol", "MaxHR",
"HeartDisease")]

  p <- ggpairs(df_subset, aes(color = HeartDisease),
    lower = list(continuous = wrap("points", alpha = 0.3)),
    diag = list(continuous = wrap("densityDiag")))
  p <- p + theme(text = element_text(size = 15))
  print(p)
})

output$correlationPlot <- renderPlot({
  num_columns <- sapply(heart, is.numeric)
  numeric_data <- heart[, num_columns]
  corr_matrix <- cor(numeric_data)
  corrrplot::corrrplot(corr_matrix,
    method="color", type="upper",
    addCoef.col = "black",
    tl.col="black", tl.srt=45,
    diag=FALSE)
})

output$modelSummary <- renderPrint({
  model_summary <- capture.output(summary(model))
  cat(model_summary, sep = "\n")
  cat("\n")

  output$modelSummary <- renderPrint({
    model_summary <- capture.output(summary(model))
    cat(model_summary, sep = "\n")
    cat("\n")

    coef_summary <- coef(model)

```

```

    conf_int <- confint(model)
    exp_coefs <- exp(cbind(coef_summary, conf_int))
    colnames(exp_coefs) <- c("Estimate", "2.5 %", "97.5 %")
    exp_coefs_df <- as.data.frame(exp_coefs)
    print(exp_coefs_df)
  })
  output$dataPlot <- renderPlot({
    heart$predict <- predict(model, newdata = heart,
type = "response")
    pred <- prediction(heart$predict, heart$HeartDisease)
    perf <- performance(pred, "tpr", "fpr")
    plot(perf, col = "blue", lwd = 2, xlab = "1 - специфічність",
ylab = "чутливість", main = "ROC-крива")
    lines(c(0,1),c(0,1), lty = 3, lwd = 1, col = "red")
    legend("bottomright", legend = c("AUC = 0.93",
"чутливість = 0.90", "специфічність = 0.85"),
col = c("blue", "red"), lty = c(1, 3), text.font = 1)
    grid(col = "gray", lty = "dotted")
    cp = cutpointr(heart$predict, heart$HeartDisease, pos_class = 1,
neg_class = 0)
    print(cp)
  })
})
}

ui <- fluidPage(
  titlePanel("Інформаційна система для обліку та аналізу медичних
даних"),
  tabsetPanel(
    tabPanel(icon("home"),
      mainPanel(
tags$h3(class = "title", "Система призначена для збору та
аналізу даних оцінки ризику серцевої недостатності"),
h4("Step 1 - Шкала оцінки ризику серцевої недостатності"),
tags$h4(class = "str", "Обґрунтування застосування шкали
для оцінки ризику серцевої недостатності"),

```

```

h4("Step 2 - Heart Dataset для аналізу даних"),
h4("Step 3 - Візуалізація даних"),
h4("Step 4 - Побудова моделі логістичної регресії для
прогнозування ймовірності розвитку серцевої недостатності"),
h4("Step 5 - ROC-аналіз для оцінки прогностичної сили
створеної моделі"),
    )),
    tabPanel("Step 1",
      tags$h3(class = "title", "Оцінка ризику серцевої
недостатності"),
      sidebarLayout(
        sidebarPanel(
          numericInput("Age", "Вік пацієнта", value = NULL, min = 1, max = 100),
          selectInput("Sex", "Стать пацієнта", c("", "M", "F")),
          selectInput("ChestPainType", "Тип Болю в грудях", c("", "TA", "ATA",
"NAAP", "ASY")),
          numericInput("RestingBP", "Артеріальний тиск у стані спокою, мм
рт.ст.", value = NULL, min = 0, max = 200),
          numericInput("Cholesterol", "Холестерин сировотки, мм/дл", value =
NULL, min = 0, max = 700),
          selectInput("FastingBS", "Рівень цукру в крові натщесерце, 1: якщо >
120 мг/дл, 0: у протилежному випадку", c("", 0, 1)),
          selectInput("RestingECG", "ЕКГ в стані спокою", c("", "Normal", "LVH",
"ST")),
          numericInput("MaxHR", "Максимальна частота пульсу", value = NULL, min
= 60, max = 210),
          selectInput("ExerciseAngina", "стенокардія, яка спричинена фізичним
навантаженням", c("", "N", "Y")),
          numericInput("Oldpeak", "ST - значення, яке виміряно при депресії",
value = NULL, min = -3.0, max = 6.5),
          selectInput("ST_Slope", "Нахил пікового сегмента ST при навантаженні",
c("", "Up", "Flat", "Down")),
          actionButton("predictBtn", "Отримати результат", class = "btn-lg")
        ),
        mainPanel(

```

```

column(width = 3, choices = c(10, 15, 20), selected = 10)
),
fluidRow(
  column(width = 12, DTOutput("myTable"))
),
DTOutput("myTable")),
tabPanel("Step 3.1",
  tags$h3(class = "title",
    "Візуалізація цільової змінної"),
  theme = bs_theme(version = 5, bootswatch = "minty"),
  selectInput("plotType", "Виберіть тип діаграми:",
    choices = c("Bar", "Pie")),
  plotOutput("plot")),
tabPanel("Step 3.2",
  tags$h3(class = "title",
    "Матриця графіків взаємодії змінних"),
  plotOutput("dataplot")),
tabPanel("Step 3.3",
  tags$h3(class = "title", "Теплова карта кореляції"),
  plotOutput("correlationPlot")),
tabPanel("Step 4 - Data Analysis",
  tags$h3(class = "title",
    "Побудова моделі логістичної регресії для прогнозування ймовірності
    розвитку серцевої недостатності"),
  verbatimTextOutput("modelSummary")),
tabPanel("Step 5",
  tags$h3(class = "title",
    "ROC-аналіз для оцінки створеної моделі"),
  plotOutput("dataPlot"))
)
)
thematic_shiny()
shinyApp(ui = ui, server = server)

```