НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

НН Інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

До захисту допущено:

Завідувач кафедри

_____ Оксана ТИМОЩУК

«___»_____2023 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системний аналіз і управління»

спеціальності 124 «Системний аналіз»

на тему: «Математичне моделювання динамічних процесів діяльності серця»

Виконав: студент IV курсу, групи КА-93
Соловей Данило Олегович _____

Керівник: доцент, к.ф.-м.н.
Шубенкова Ірина Анатоліївна _____

Консультант з нормоконтролю: доцент, к.ф-м.н.
Статкевич Віталій Михайлович _____

Консультант з економічного розділу: доцент, к.е.н.
Рощина Надія Василівна _____

Рецензент:

_____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
НН Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 124 «Системний аналіз»

Освітня програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Оксана ТИМОЩУК
«__» травня 2023 р.

ЗАВДАННЯ

на дипломну роботу студенту

Соловею Данилу Олеговичу

1. Тема роботи «Математичне моделювання динамічних процесів діяльності серця», керівник роботи доцент, к.ф.-м.н. Шубенкова Ірина Анатоліївна затверджені наказом по університету від «__» травня 2023 р. № _____

2. Термін подання студентом роботи __.06.2023.

3. Вихідні дані до роботи: датасет з даними людей з серцевими хворобами та без них

4. Зміст роботи: аналіз предметної області, огляд існуючих підходів, огляд та аналіз датасету, навчання моделей, створення програмного продукту, оцінка результатів.

5. Перелік ілюстративного матеріалу: = презентація для захисту дипломної роботи.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Рощина Н.В., доцент		

7. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Формування тематики дослідження	15.04.2023 - 21.04.2023	Виконано
2	Огляд існуючих підходів для вирішення задачі	15.04.2023 - 28.04.2023	Виконано
3	Збір інформації, пошук набору даних	21.04.2023 - 28.04.2023	Виконано
4	Написання програми	28.04.2023 - 05.05.2023	Виконано
5	Оформлення пояснювальної записки	05.05.2023 - 12.05.2023	Виконано
6	Створення презентації для захисту	05.05.2023 - 19.05.2023	Виконано
7	Попередній захист дипломної роботи	19.05.2023 - 26.05.2023	Виконано
8	Захист дипломної роботи	19.05.2023 - 26.05.2023	Виконано

Студент Данило СОЛОВЕЙ

Керівник Петро БІДЮК

РЕФЕРАТ

Дипломна робота: 152 ст., 13 табл., 46 рис., 2 додатки, 79 джерел.

ПРОГНОЗУВАННЯ, МОДЕЛЮВАННЯ, СЕРЦЕВІ ХВОРОБИ, МАШИННЕ НАВЧАННЯ.

Тема:

Математичне моделювання динамічних процесів діяльності серця.

У роботі розглянуто прогнозування наявності серцевої хвороби.

Об’єкт дослідження: є медичні показники (демографічні, симптоми, ЕКГ та результати обстежень) та їх значення для успішного діагностування захворювання.

Предметом дослідження: датасет з даними.

Мета роботи: розглянути різні методи прогнозування для діагностування захворювання, оцінити та обрати найкращий з них.

Методи дослідження: засоби для прогнозування.

Новизна цієї дипломної роботи складається в кількості параметрів для прогнозування наявності серцевої хвороби.

ABSTRACT

Thesis work: 152 p., 13 tables, 46 figs., 2 appendices, 79 sources.

FORECASTING, MODELING, HEART DISEASE, MACHINE LEARNING.

The theme: Mathematical modeling of dynamic processes of heart activity.

The work considers the prediction of the presence of heart disease.

Object of study: there are medical indicators (demographic, symptoms, ECG and examination results) and their importance for the successful diagnosis of the disease.

Subject of research: dataset with data.

Purpose: to consider various forecasting methods for diagnosing the disease, evaluate and choose the best of them.

Research methods: means for forecasting.

The novelty of this thesis consists in the number of parameters for predicting the presence of heart disease.

ЗМІСТ

ВСТУП 8

РОЗДІЛ 1 МЕТОДИ МОДЕЛЮВАННЯ В МЕДИЦИНІ 9

1.1 Математичне моделювання в медицині 10
1.2 Моделі системної динаміки 13
1.3 Моделі на основі агентів 16
1.4 Мережевий аналіз 19
1.5 Висновки до розділу 1 22

РОЗДІЛ 2 МАТЕМАТИЧНІ МОДЕЛІ ДЛЯ МОДЕЛЮВАННЯ ДИНАМІЧНИХ ПРОЦЕСІВ ДІЯЛЬНОСТІ СЕРЦЯ 23

2.1 Як функціонує серце 25
2.2 Моделювання серцебиття 27
2.3 Моделювання тиску 29
2.4 Моделювання системи кровотоку 31
2.5 Серцеві хвороби 33
2.5.1 Алгоритм K-Nearest Neighbors 34
2.5.2 Алгоритм Random Forest 36
2.5.3 Алгоритм Decision Tree 38
2.5.4 Алгоритм Support Vector Machine 39
2.5.5 Алгоритм Logistic Regression 40
2.5.6 Алгоритм Naïve Bayes 41
2.6 Висновки до розділу 2 42

РОЗДІЛ 3 ОЦІНКА ТА ПРОГНОЗУВАННЯ МЕТОДАМИ МАШИННОГО НАВЧАННЯ 46

3.1 Опис датасету 47
3.2 Аналіз даних 48
3.2.1 Кореляція даних 49
3.2.2 Розподіл за статтю 52
3.2.3 Розподіл за віком 54
3.2.4 Розподіл за болем у грудях 58
3.2.5 Розподіл за рівнем цукру 60
3.2.6 Розподіл за тиском 61
3.2.7 Розподіл за основними венами 63
3.2.8 Розподіл за ЕКГ 64
3.2.9 Розподіл за стенокардією при фізичному навантаженні (стенокардія напруження) 66

3.2.10 Розподіл за ST-схилом 67
3.2.11 Розподіл за таласемією 69
3.2.12 Розподіл за холестерином 69
3.2.13 Розподіл за серцебиттям 70
3.2.14 Розподіл за Oldpeak 70
3.3 Побудова моделей та прогнозування 71
3.3.1 Оцінювання прогнозування Logistic Regression 73
3.3.2 Оцінювання прогнозування для Random Forest 74
3.3.3 Оцінювання прогнозування для Naïve Bayes 75
3.3.4 Оцінювання прогнозування для KNN 76
3.3.5 Оцінювання прогнозування для Decision Tree 77
3.3.6 Оцінювання прогнозування для Support Vector Machine 78
3.4 Порівняння усіх методів 80
3.5 Висновки до розділу 3 80
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ 82
4.1 Постановка задачі проектування 82
4.2 Обґрунтування функцій програмного продукту 83
4.3 Обґрунтування системи параметрів програмного продукту 86
4.4 Аналіз експертного оцінювання параметрів 89
4.5 Аналіз рівня якості варіантів реалізації функцій 94
4.6 Економічний аналіз варіантів розробки ПП 95
4.7 Вибір кращого варіанту ПП техніко-економічного рівня 102
4.8 Висновки до розділу 4 103
ВИСНОВКИ 104
ПЕРЕЛІК ДЖЕРЕЛ 106
ДОДАТОК А ІЛЮСТРАТИВНИЙ МАТЕРІАЛ 115
ДОДАТОК Б ЛІСТИНГ ПРОГРАМИ 116

ВСТУП

Серце — це найважливіший орган нашого тіла, який забезпечує кровообіг та необхідне живлення всіх органів та тканин [1]. Це складний механізм, який працює у надзвичайно точному ритмі. Проте, багато людей стикається з різними серцево-судинними захворюваннями, які можуть призвести до смерті. Тому для того, щоб розробити ефективні методи лікування і запобігання цим захворюванням, необхідно вивчати процеси діяльності серця.

Моделювання серцево-судинної системи — це важлива галузь науки, яка дозволяє аналізувати різні фактори, що впливають на роботу серця, та розробляти методи лікування серцево-судинних захворювань [2, с.44-45]. Воно дозволяє досліджувати різні аспекти функціонування серця, такі як механізми скорочення серцевих м'язів, протікання крові через судини, та взаємодію серця з іншими системами організму.

Моделювання серцево-судинної системи можна розглядати з різних підходів, які залежать від рівня складності моделі. Наприклад, можна створювати прості моделі, які описують основні функції серцево-судинної системи, або складніші моделі, які враховують більш детальну структуру та функціонування серцево-судинної системи.

РОЗДІЛ 1 МЕТОДИ МОДЕЛЮВАННЯ В МЕДИЦИНІ

Застосування математики в медицині має давню історію. Ще в стародавньому світі вчені і філософи намагалися знайти гармонію в будові людського тіла. Наприклад, золотий переріз використовувався для опису взаємовідносин між різними частинами тіла [3, с.23]. Крім того, в системі загальноприйнятих на той час уявлень будь-яка хвороба розглядалася як відсутність рівноваги в основному організмі [4, с.45].

Моделювання є важливою складовою медичної практики і може бути застосоване в різних областях медицини, таких як діагностика, лікування та дослідження [3, с.67-68]. Математичні моделі мають великий потенціал щодо їх корисності в різних дисциплінах медицини та охорони здоров’я. Математичні моделі корисні в епідеміологічних дослідженнях [3, с.34-35], плануванні та оцінці профілактичних і контрольних програм [3, с.35], клінічних випробуваннях [4, с.66], вимірюванні стану здоров'я, аналізі витрат і вигод [3, с.37], діагностиці пацієнтів і в максимізації ефективності операцій, спрямованих на досягнення визначених цілей у межах наявних ресурсів [4, с.67].

Моделювання може бути показано за допомогою циклу. Кайзер та Блум сформулювали цикл моделювання, що дозволяє зрозуміти моделювання графічно (Рис. 1):

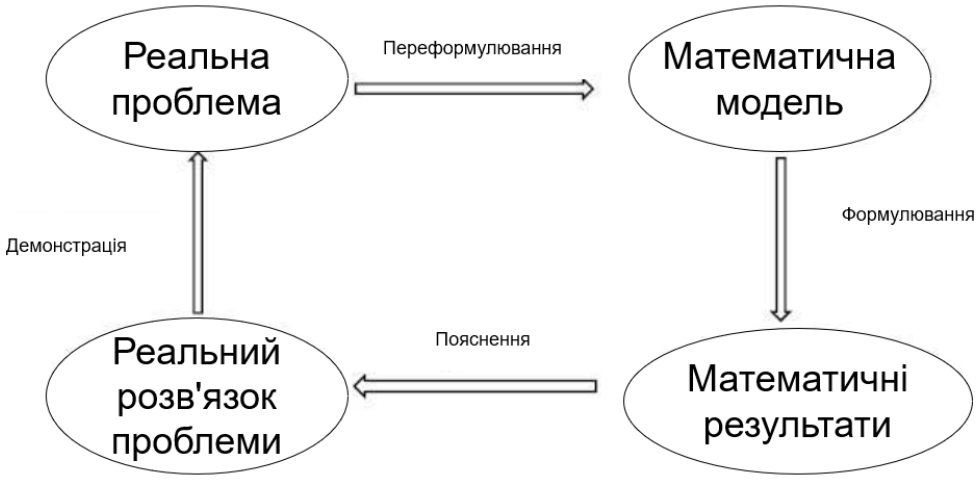


Рис. 1.1. — Цикл моделювання

З наведеного вище рисунка 1 можна зрозуміти, що реальність перетворюється на математичну модель за допомогою диференційних рівнянь, а потім, розв’язуючи рівняння за допомогою різних методів і прийомів, отримується рішення, після чого його інтерпретують в рішення реального світу.

1.1 Математичне моделювання в медицині

Сьогодні можна виділити два основних підходи до математичного опису явищ медицини.

Перший підхід полягає у виявленні закономірностей у кількісному аналізі лікарських даних. Зазвичай такі дослідження відносять до сфери біометрії [4, с.55]. Сучасні статистичні методи та комп'ютерні системи широко використовуються для обробки величезної кількості біометричних даних. У свою чергу, інтенсивний аналіз даних призводить до швидкого розвитку правильних математичних методів. Наприклад, термін «регресія» був введений у 19 столітті в результаті дослідження спадковості фізіологічних особливостей людини [6, с.24].

Другий підхід полягає в прогнозуванні поведінки системи за допомогою даних про механізми, що лежать в основі описаних процесів. Цей вид математичних моделей може мати узагальнений характер і описувати біологічні процеси будь-якого рівня складності [5, с.45]. Крім того, ці моделі можуть використовувати інформацію, отриману за допомогою першого підходу. Слід зазначити, що математичні моделі створюють компроміс між двома крайнощами.

Дуже просту модель можна легко побудувати та проаналізувати математично. Але вона може адекватно описати проблему лише в невеликому діапазоні умов. У свою чергу, дуже складна модель може врахувати набагато більше реальних процесів, але її важко налаштувати та працювати.

Один з найпоширеніших методів моделювання в медицині — комп'ютерна томографія та магнітно-резонансна томографія [5, с.145], які дозволяють отримувати детальні зображення внутрішніх органів та тканин. Ці зображення можуть бути використані для створення тривимірних моделей органів та тканин, що дозволяє лікарям більш точно діагностувати захворювання та планувати операції.

Інший поширений метод моделювання — це математичне моделювання [4, с.89], яке використовується для аналізу роботи різних систем організму, наприклад серцево-судинної системи. За допомогою математичних моделей можна аналізувати роботу органів в різних умовах, проводити віртуальні експерименти та встановлювати залежності між різними параметрами.

У медицині застосовуються модельовані людські тіла, які дозволяють досліджувати різні аспекти фізіології та поведінки людей в різних умовах. Ці моделі дозволяють вивчати ефективність ліків та діагностичних методів, а також використовувати їх для планування та віртуального тренування хірургічних операцій.

Нарешті, важливою складовою моделювання в медицині є використання комп'ютерних симуляцій та віртуальної реальності [5, с.123]. Ці технології дозволяють створювати віртуальні середовища.

Майже всі галузі сучасної фізіології використовують математичні та комп’ютерні методи певною мірою. Спостерігається тенденція до переходу від моделювання ізольованих систем і процесів до побудови складних моделей взаємопов'язаних систем, що контролюють регуляцію значної кількості життєвих функцій.

Системи здоров'я є складними адаптивними системами. Як такі, вони характеризуються надзвичайною складністю відносин між дуже різнорідними групами зацікавлених сторін і процесами, які вони створюють. Системні явища масової взаємозалежності, самоорганізації та емерджентної поведінки, нелінійності, часових затримок, петель зворотного зв’язку, залежності від шляху та переломних точок роблять поведінку системи охорони здоров’я важкою, а іноді й неможливою для прогнозування чи управління.

Системи охорони здоров’я охоплюють багаторівневу систему надання послуг місцевому, районному та національному населенню, від громадських центрів охорони здоров’я до лікарень третинного рівня. Спроба оцінити ефективність такої багатогранної організації представляє складне завдання. Таким чином, математичне моделювання, здатне моделювати поведінку складних систем, є життєво важливим дослідницьким інструментом, який допомагає зрозуміти функціонування та оптимізацію системи охорони здоров’я.

Моделі системної динаміки і Моделі на основі агентів — два найпопулярніші методи математичного моделювання для оцінки складних систем; у той час як Моделі системної динаміки використовується для вивчення поведінки системи на макрорівні, наприклад руху ресурсів або кількості в системі з часом, Моделі на основі агентів фіксують поведінку системи на

мікрорівні, наприклад, прийняття рішень людиною та неоднорідні взаємодії між людьми.

1.2 Моделі системної динаміки

Системно-динамічне моделювання (System dynamics model) — це методологія вивчення та управління складними системами зворотного зв'язку [8, с.12]. Ця методологія базується на припущенні, що складна поведінка системи (наприклад, поширеність інфекції в популяції) є результатом взаємодії циклів зворотного зв'язку, запасів і потоків, а також затримок. Системна динаміка зазвичай використовується, коли формальних аналітичних моделей не існує, але коли моделювання системи можна розробити шляхом зв'язування кількох механізмів зворотного зв'язку [9, с.15-16].

Цей метод спочатку виник у науці управління [9, с.14] через визнання необхідності явного моделювання нелінійних процесів, характерних для складних явищ, таких як опір політики, закон ненавмисних наслідків і часто неінтуїтивна поведінка соціальних систем [10, с.54]. Моделювання реалізовано у вигляді серії диференціальних рівнянь, які відстежують накопичення запасів (наприклад, людей, валюти, кількості захворювань тощо), які визначаються потоками (наприклад, швидкістю виникнення), циклами зворотного зв'язку (причинно-наслідкові цикли з балансуванням або підсилювальні ефекти) і затримки часу.

Цей тип системного моделювання, будучи нижчим у деталізації та вищим рівнем інтеграції, дозволяє експертам у галузі та місцевим зацікавленим сторонам досліджувати взаємозв'язок між різними технічними параметрами та загальною поведінкою системи та покращувати їхнє розуміння взаємодії та впливу між різними водними системами. Основна увага системної динаміки зосереджена на побудові моделей для представлення динамічної складності сукупних, часто високорівневих явищ, таких як прийняття нових продуктів в

організаціях або відносини хижак-жертва з часом. Результати моделювання дозволяють перевірити поведінку системи, яка може приймати різні моделі (наприклад, експоненціальне зростання, коливання, s-подібне зростання, колапс тощо [11, с.43]) і порівнювати з гіпотетичною чи очікуваною поведінкою системи (тобто, з еталонними моделями).

Моделі системної динаміки використовувалися для надання корисних ілюстративних моделей навіть за відсутності сильних емпіричних даних, щоб продемонструвати відносний вплив різних політик або стратегій втручання, особливо коли цикли зворотного зв'язку можуть використовуватися для пояснення моделей нелінійності або непередбачених наслідків (наприклад, модель профілактики хронічних захворювань у Homer & Hirsch [12, с.452–458]).

Порівняно з іншими типами моделей складних систем, моделі системної динаміки, як правило, мають ширші межі (тобто включають більшу кількість відповідних пояснювальних змінних) і більш піддатливі для включення змінних, для яких можуть бути недоступні сильні емпіричні дані [12, с.43]. У сфері системної динаміки великий акцент робиться на груповій побудові моделей [13, с.45] де моделі розробляються в процесі участі модельєра та практиків або кінцевих користувачів.

Методологія системно-динамічного моделювання добре підходить для вирішення динамічної складності, яка характеризує багато проблем громадського здоров'я [8, с.48]. Системно-динамічний підхід передбачає розробку комп'ютерних симуляційних моделей, які зображують процеси накопичення та зворотного зв'язку та які можна систематично перевіряти для пошуку ефективної політики для подолання опору політики.

Використовується ця методологія у багатьох випадках, як-от для запобігання хронічних хвороб. Моделювання системної динаміки профілактики хронічних захворювань має прагнути включити всі основні елементи сучасного екологічного підходу [4, с.98], включаючи результати захворювання, здоров'я

та ризиковану поведінку, фактори навколишнього середовища, а також пов’язані зі здоров’ям ресурси та системи доставки. Моделі системної динаміки є багатообіцяючими як засіб моделювання багатьох взаємодіючих захворювань і ризиків, взаємодії систем доставки та хворих груп населення, а також питань національної та державної політики.

Значення моделювання системної динаміки найкраще пояснити за допомогою ілюстрації. На рис. 1.2 представлено основну причинно-наслідкову структуру моделі та її вхідні дані:

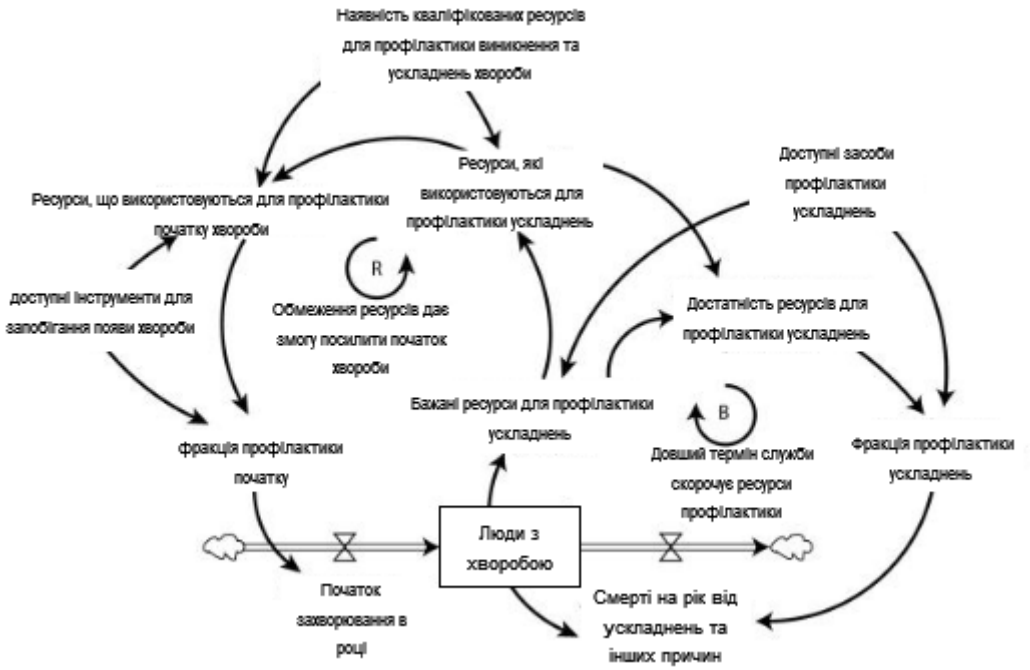


Рис. 1.2. — Причинно-наслідкова структура моделі системної динаміки та її вхідні дані

Єдиний запас людей із захворюваннями являє собою поступово мінливе чисте накопичення двох потоків: притоку початку захворювання та відтоку смертей. Передбачається, що кваліфіковані ресурси для профілактики, які складаються, можливо, з усіх постачальників первинної медичної допомоги в регіоні, де розташована популяція захворювання, є константою. Певні клінічні

інструменти (діагностичні та терапевтичні) доступні для цих постачальників для запобігання ускладненням, а інші інструменти доступні для запобігання їх початку [14, с.78-79].

Чим більше людей із захворюваннями та чим більше доступних інструментів для профілактики ускладнень, тим більше часу постачальників буде присвячено профілактиці ускладнень. Залишок часу тоді доступний для зусиль із запобігання початку захворювання серед пацієнтів, які не хворіють (наскільки це дозволяють доступні засоби запобігання початку захворювання), або поглинається іншими видами діяльності, не спрямованими на профілактику.

1.3 Моделі на основі агентів

Моделювання на основі агентів (Agent-based model) — це комп'ютерне моделювання, яке використовується для вивчення взаємодії між людьми, речами, місцями та часом. Це стохастичні моделі, побудовані знизу вгору, що означає, що окремим агентам (часто людям з епідеміології) присвоюють певні атрибути [15, с.23-24].

Агентне моделювання використовує комп'ютерне моделювання для вивчення складних систем з нуля, досліджуючи, як окремі елементи системи (агенти) поводяться як функція окремих властивостей, їх середовища та їх взаємодії один з одним [15, с.66-69]. Порівняно з системною динамікою, це призводить до форми децентралізованого моделювання, де немає формалізованого визначення поведінки глобальної системи (тобто немає диференціальних рівнянь, які керують процесами високого рівня системи) [15, с.34-39]. Моделювання на основі агентів є наймолодшим із цих трьох системних наукових методів, хоча його концептуальне коріння сягає важливих відкриттів 20-го століття в математиці, філософії та інформатиці,

включаючи винахід клітинних автоматів Фон Неймана і Гра в життя Джона Конвея [16, с.45].

Однією з перших впливових агентних моделей, яка чітко продемонструвала, як поведінку складних систем можна описати, використовуючи лише прості правила на рівні агенту, було моделювання Рейнольдсом зграї птахів [17, с.34-35]. Модель «boids» Рейнольдса використовувала лише три прості правила на рівні птахів:

1) відокремлення (не підходьте занадто близько до інших птахів);

2) вирівнювання (відповідати швидкості та напрямку птахів, що знаходяться поблизу);

3) згуртованість (голова до центру мас найближчих птахів).

Результатом симуляції з використанням цих правил став: «...витончений танець-подібний рух зграї, чий гіпнотичний ритм чітко структурований, але водночас дуже нелінійний» [18, с.143-146].

Моделювання на основі агентів використовувався у багатьох дисциплінах, але був особливо корисним для опису нових властивостей організаційних, соціальних і культурних систем в антропології, соціології, політології, бізнесі та економіці [18; 19]. У більш загальному плані було показано, що агентне моделювання є особливо корисним для моделювання нових явищ, таких як потоки зараження, ринки, організаційна поведінка та дифузія [19, с.45]; усі вони мають відношення до досліджень громадської охорони здоров’я.

Агенти запрограмовані на поведінку та взаємодію з іншими агентами та середовищем певним чином. Ці взаємодії спричиняють виникнення ефектів, які можуть відрізнятися від ефектів окремих агентів. Агентне моделювання відрізняється від традиційних методів на основі регресії тим, що, як і моделювання системної динаміки, воно дозволяє досліджувати складні системи [18, с.58], які демонструють незалежність індивідів і петлі зворотного зв’язку в причинно-наслідкових механізмах. Воно не обмежується даними

спостережень і може використовуватися для моделювання експериментів, які неможливо або неетично проводити в реальному світі.

Модель на основі агентів виглядає наступним чином (Рис. 1.3):

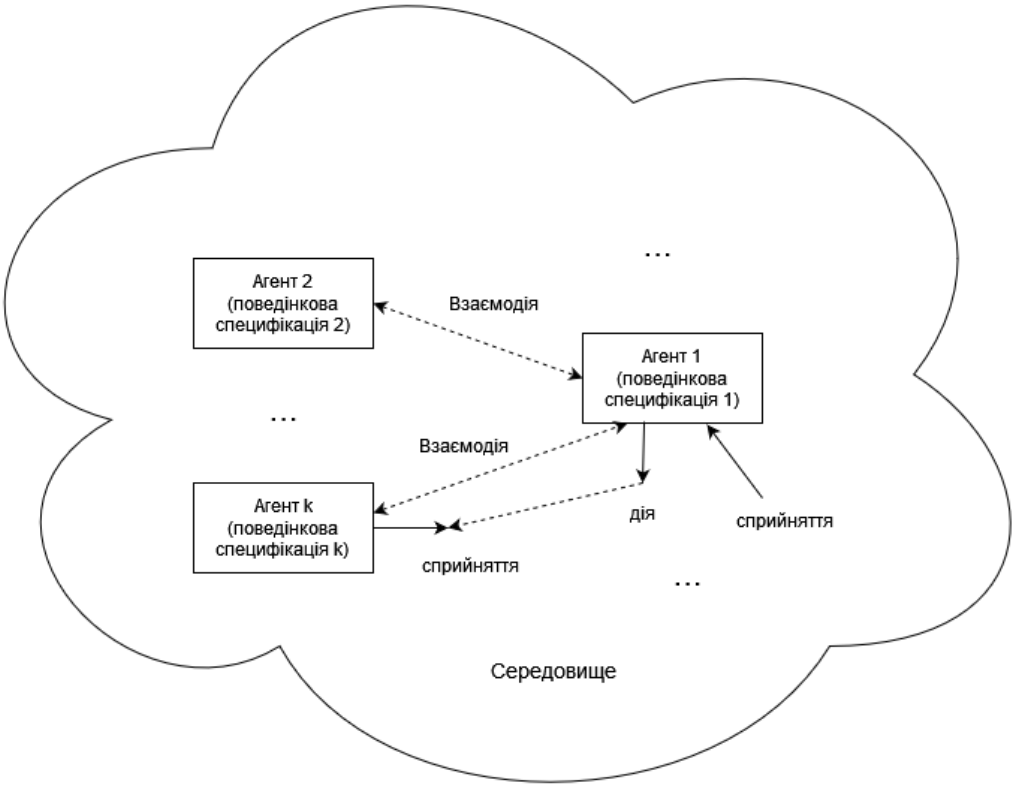


Рис. 1.3. — Модель на основі агентів

Однак моделювання на основі агентів не позбавлене обмежень. Параметри даних (наприклад, рівень репродуктивності при інфекційних захворюваннях) часто важко знайти в літературі [19, с.37-39]. Крім того, валідність моделі може бути важко оцінити, особливо при моделюванні неспостережуваних асоціацій. Загалом моделі на основі агентів надають додатковий інструмент для оцінки впливу на результати [18, с.44]. Це особливо корисно, коли відомо або підозрюється наявність взаємозв’язку, взаємності та петель зворотного зв’язку або коли експерименти в реальному світі неможливі.

На відміну від системно-динамічних моделей, Моделі на основі агентів є базовим представленням системи, що моделює мінливі стани окремих «агентів» у системі, а не широкі об’єкти чи сукупну поведінку, як в моделях системної динаміки [20, с.67-69]. Проте поведінку сукупної системи можна зробити висновком з моделей на основі агентів.

Використання моделей на основі агентів для моделювання поведінки системи було трансдисциплінарним, із застосуванням в економіці до екології, від соціальних наук до техніки [18; 19; 20]. Може бути кілька типів модельованих агентів, кожному з яких присвоєні власні характеристики та модель поведінки.

Агенти можуть вчитися на власному досвіді, приймати рішення та виконувати дії на основі встановлених правил (наприклад, евристики), інформуючи їх про взаємодію з іншими агентами, їхні власні призначені атрибути або на основі їхньої взаємодії з змодельованим середовищем [21, с.609-618]. Взаємодія між агентами може призвести до трьох рівнів спілкування між агентами; зв’язок «один-до-одного» між агентами, зв’язок «один-до-багатьох» між агентами та зв’язок «один-до-локації», де агент може впливати на інших агентів, що містяться в певному місці [21, с.405-409].

1.4 Мережевий аналіз

Мережевий аналіз (Network Analysis) — це метод дослідження та наукова парадигма, яка зосереджується на взаємозв’язках між групами акторів. Акторами можуть бути будь-які типи суб’єктів, які можуть мати відносини або зв’язки з іншими суб’єктами: люди, тварини, організації, країни, веб-сайти, документи та навіть гени.

З трьох методів, розглянутих у цьому розділі, мережевий аналіз має найдовшу історію — коріння мережевого аналізу можна простежити до низки

різних дисциплін, включаючи математику (особливо теорію графів і топологію), антропологію (системи спорідненості) і соціологію (соціальні зв'язки та структура) [22, с.34-39]. Проте те, що зараз визнається як сучасний мережевий аналіз, було створено на початку 1930-х років із винаходом Джейкоба Морено соціограми — графіка, який зображує структуру міжособистісних стосунків у групі [23, с.206-219] З наявністю ефективних комп'ютерних алгоритмів, розробкою спеціалізованого програмного забезпечення для аналізу мереж і «відкриттям» аналізу мереж сучасними фізиками та математиками [24] інтерес до аналізу мереж вибухнув.

Через довшу історію та здатності швидко аналізувати дані реального світу мережевий аналіз має більшу різноманітність застосувань і аналітичних підходів порівняно з моделями системної динаміки і моделями на основі агентів [25].

Незважаючи на різноманітність аналітичного аналізу, майже весь мережевий аналіз використовує один або декілька з трьох різних аналітичних режимів:

1) візуалізація мережі;

2) опис мережі;

3) статистичне моделювання мережі.

Однією з переваг мережевого аналізу є можливість візуально дослідити певну мережу, особливо якщо вона малого чи середнього розміру. Опис мережі становить основну частину аналізу мережі, і його можна гнучко використовувати для вирішення широкого спектру наукових питань. Рисунок 1.4 підкреслює це різноманіття:

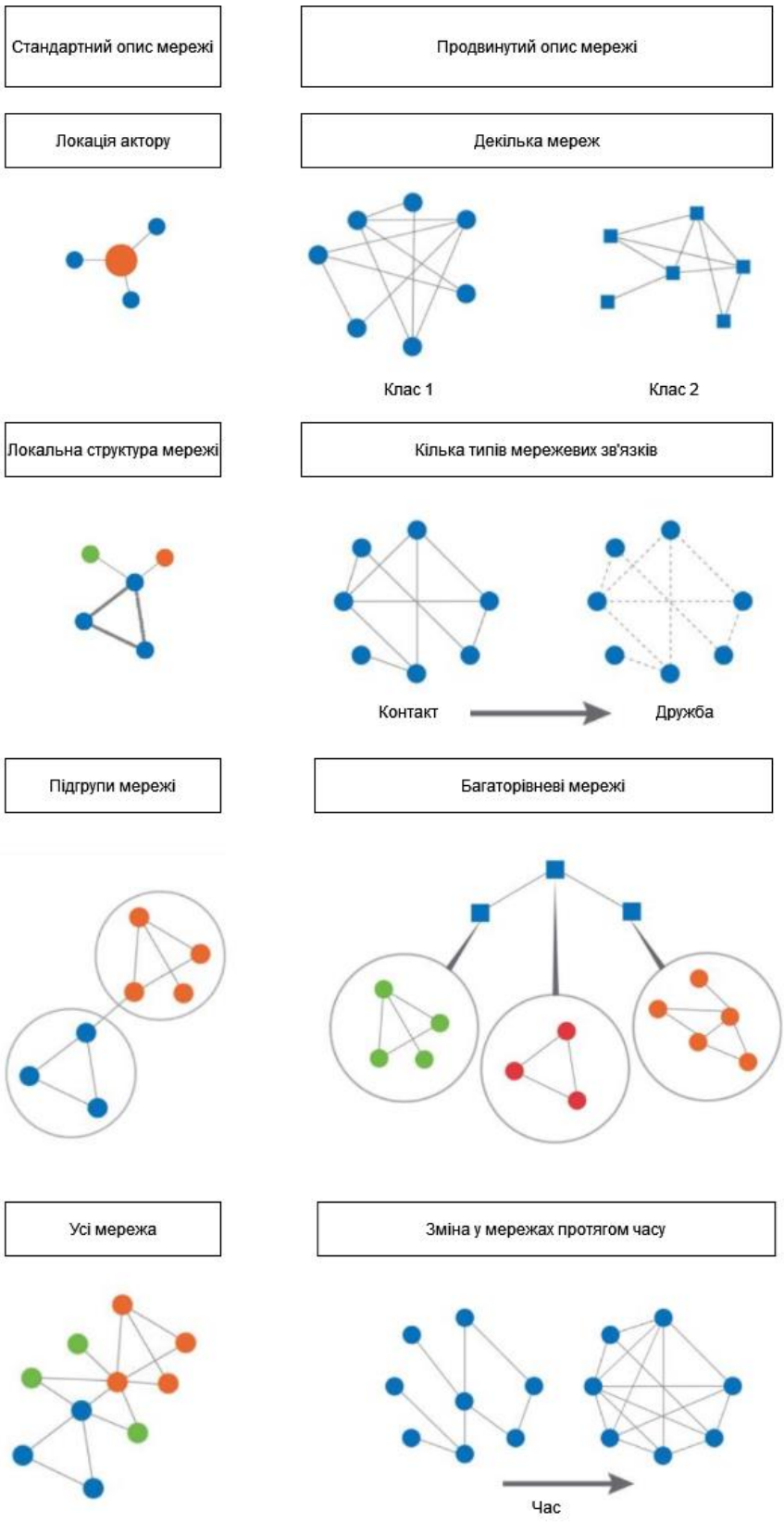


Рис. 1.4. — Підходи до аналізу для базового та розширеного аналізу мережі

1.5 Висновки до розділу 1

У першому розділі були розглянуті поняття, що пов’язані із математичним моделюванням у медицині. Математичне моделювання в медицині — це досить важливий інструмент, що дозволяє досліджувати характеристики та ефективність лікарських засобів, процеси лікування та попередження захворювань, а також допомагає приймати рішення щодо оптимальної стратегії лікування.

У розділі були розглянуті види моделювання в медицині: Системно-динамічні моделі, моделі на основі агентів та мережевий аналіз. Моделі системної динаміки можуть допомогти в передбаченні поведінки екосистем чи біологічних мереж, враховують взаємодію між різними компонентами системи та їхні впливи на динаміку системи в цілому.

До кожної моделі було наведено малюнок та опис.

Підбиваючи підсумки, біологічні моделі та математичне моделювання в медицині та біології є потужними інструментами для дослідження складних біологічних систем, виявлення нових лікарських цілей та розробки оптимальних стратегій лікування. Важливою умовою використання цих методів є їхній валідаційний аспект та можливість перевірки моделей на експериментальних даних.

РОЗДІЛ 2 МАТЕМАТИЧНІ МОДЕЛІ ДЛЯ МОДЕЛЮВАННЯ ДИНАМІЧНИХ ПРОЦЕСІВ ДІЯЛЬНОСТІ СЕРЦЯ

Хвороби серця є основною причиною смерті в усьому світі, особливо в розвинених країнах і країнах, що розвиваються. Лікування таких захворювань має різні форми, від найпростіших, таких як дотримання відповідних дієт, до дуже складних і небезпечних форм, таких як трансплантація серця. Через удосконалення технологій можна виявити нові можливості, наприклад, нові медичні допоміжні пристрої або прогностичні медичні інструменти, які допомагають знати про своєчасно виникнуть небезпечні ситуації для людського здоров'я.

Серцево-судинна система людини дуже складний механізм і включає багато механізмів управління. Були розроблені різні електричні моделі серця людини, часткові або повні, з лінійними та нелінійні. У літературі наявні фізичні аналогові моделі повністю штучного серця, барорецепторна модель, модель простору станів, електромеханічна бівентрикулярна модель серця та математична модель для штучної генерації сигналів електрокардіограми (ЕКГ).

Чисельне моделювання фізіології серця відіграє важливу роль для оцінки клінічних сценаріїв і результатів різних методів лікування перед експериментальним або клінічним застосуванням. Тим не менш, інформація, отримана з чисельного моделювання, залежить від змодельованих співвідношень у використаній моделі. Наприклад, моделі пружності, що змінюються в часі, описують співвідношення тиску й об’єму в камерах серця за допомогою функції пружності, яка змінюється з часом протягом серцевого циклу [26, с.1710-1726]. Моделі еластичності, що змінюються в часі, використовувалися для різних клінічних сценаріїв, таких як оцінка серцевої

недостатності [27, с.39-49], підтримка допоміжного пристрою лівого шлуночка [28, 733-745] або моделювання взаємодії між серцево-судинною та дихальною системами [29, с.73-79]. Незважаючи на те, що моделі пружності, що змінюються в часі, широко використовуються для моделювання серцевої функції, вони імітують лише артеріальний тиск і об'єм у камерах серця.

Більш детальні чисельні моделі, такі як моделі скорочення одного волокна, описують серцеву функцію шляхом імітації скорочення саркомера протягом серцевого циклу [30, с.467-499]. Моделі скорочення одного волокна дозволяють моделювати натяг волокна в камері серця, крім артеріального тиску та об'єму [31, с.593-603]. Моделі скорочення одного волокна також використовувалися для моделювання фізіологічних і клінічних сценаріїв, таких як вплив внутрішньоміокардіального тиску на швидкість коронарного артеріального кровотоку [32, с.1833-1835], моделювання варіабельності серцевого ритму плода [33, с.55-64] або оцінка роторного кровоносного насоса. підтримки [34, с.373-387] тощо. Хоча ці моделі керуються скороченням волокон і надають більше інформації про фізіологію серця, вони також залишаються недостатніми для моделювання різних механізмів, що відбуваються на кожному рівні серцевого скорочення, і не надають інформації про клінічні показники, такі як серце розміри камери для фізіологічних випадків.

Багатомасштабні моделі серцевої динаміки, що моделюють фізіологічні процеси на клітинному, білковому та органному рівнях, також були розроблені та використані для розуміння патофізіології серцевої недостатності [35, с.141-153]. Однак збільшення складності чисельної моделі може не покращити результат моделювання. Як показано в [36, с.273-277], відносно прості моделі, що описують серцеву функцію, симулюють фізіологічні та клінічні випадки, такі як серцева недостатність і механічна підтримка кровообігу; точніше щодо багатомасштабних моделей, оскільки невеликі розбіжності між реальною фізіологією та числовою моделлю в кожній масштабу викликають великі

відхилення на рівні органів. Крім того, моделювання зосереджених параметрів в основному використовувалося для моделювання серцево-судинної системи дорослих. Чисельне моделювання серцевої функції у дітей ще належить працювати, щоб оцінити різні фізіологічні сценарії, оскільки лише невелика кількість досліджень зосереджена на педіатричних випадках [37, с.87-88]. Детальний огляд моделей зосереджених параметрів наведено в [38, с.107-111].

2.1 Як функціонує серце

Серце складається з трьох шарів тканини:

1) Епікард;
2) Міокард;
3) Ендокард.

Ці шари оточені перикардом, тонкою зовнішньою оболонкою, яка захищає ваше серце. Серце складається з чотирьох камер: дві зліва і дві справа. Дві невеликі верхні камери є передсердями [39, с.189]. Дві більші нижні камери є шлуночками. Ці ліва і права сторони серця розділені м’язовою стінкою, яка називається перегородкою.

Серцево-судинна система. Серце весь час перекачує кров по всьому тілу — приблизно п'ять літрів, — і це називається кровообігом [40, с.122-123]. Серце, кров і кровоносні судини разом складають серцево-судинну систему. У правий бік серця надходить кров з низьким вмістом кисню, тому що більша частина була використана мозком і тілом. Він перекачує його до легенів, де отримує свіжий запас кисню. Потім кров повертається до лівої частини серця, готова бути викачана назад до мозку та решти тіла.

Кровоносні судини. Кров перекачується по тілу через мережу кровоносних судин:

1) артерії — вони несуть насичену киснем кров від серця до всіх частин тіла, зменшуючись у міру віддалення від серця;
2) капіляри — вони з'єднують найдрібніші артерії з найдрібнішими венами та допомагають обмінюватися водою, киснем, вуглекислим газом та іншими поживними речовинами та відходами між кров'ю та тканинами навколо них;
3) вени — вони несуть кров, якій не вистачає кисню, назад до серця, і стають більшими, коли вони наближаються до серця.

Кровоносні судини можуть розширюватися або звужуватися залежно від того, скільки крові потрібно кожній частині тіла [40, с.134]. Ця дія частково контролюється гормонами.

Клапани. Серце має чотири клапани. Вони діють як ворота, утримуючи кров рухатися в потрібному напрямку:

1) аортальний клапан — зліва;
2) мітральний клапан — зліва;
3) легеневий клапан — з правого боку;
4) трикуспідальний клапан — з правого боку.

Кров’яний тиск. Це вимірювання тиску в артеріях. Він відіграє важливу роль у тому, як серце доставляє свіжу кров до всіх кровоносних судин. Щоб кров рухалася по вашому тілу досить швидко, вона повинна бути під тиском. Це створюється зв’язком між трьома речами:

1) накачування серця;
2) розмір і розтяжність кровоносних судин;
3) густота самої крові.

Одне серцебиття — це єдиний цикл, під час якого серце скорочується та розслабляється, щоб перекачувати кров. У спокої нормальне серце скорочується приблизно від 60 до 100 разів щохвилини, і воно прискорюється під час фізичних вправ.

Щоб забезпечити адекватне кровопостачання тіла, чотири камери серця повинні регулярно перекачуватись у правильній послідовності. Існує дві фази циклу роботи серця:

1) систола — це коли серце скорочується, виштовхуючи кров із камер;
2) діастола — це період між скороченнями, коли м'яз серця (міокард) розслабляється, а камери наповнюються кров'ю

2.2 Моделювання серцебиття

Основна функція серця — перекачувати кров по всьому тілу [41, с.12-15]. Це можна розділити на два різні контури, перший — низького тиску і перекачує кров з нашого тіла в легені, щоб насититися киснем, перш ніж повернутися до серця. Другий — це контур високого тиску, який потім викачує насичену киснем кров з аорти до решти тіла.

Битися серце змушує синоатріальний вузол. Це природний кардіостимулятор серця, який змушує його скорочуватися в систолу за допомогою електрохімічної хвилі [42, с.45-48].

Сигнал електрокардіограми (ЕКГ) є одним з найбільш очевидних ефектів роботи серця людини [41, с.34-35]. Коливання між станами систоли і діастоли серця відображається на частоті серцевих скорочень. Поверхнева ЕКГ — це записана різниця потенціалів між двома електродами, розміщеними на поверхні шкіри в заздалегідь визначених точках [43, с.1283-1286]. Найбільшу амплітуду одного циклу нормальної ЕКГ називають зубцем R, що свідчить про процес деполяризації шлуночка. Час між послідовними R-хвилями широко використовується як міра функції серця, і це допомагає ідентифікувати пацієнтів із ризиком серцево-судинної події або смерті. Аналіз варіацій у цьому часовому ряді відомий як аналіз варіабельності серцевого ритму [44, с.678-690].

Розробка динамічної моделі стане корисним інструментом для аналізу впливу різних фізіологічних умов на ЕКГ [44, с.789-799]. Згенеровані моделлю сигнали ЕКГ з різними характеристиками також можуть бути використані як джерела сигналів для оцінки діагностичних пристроїв обробки сигналів ЕКГ. Динамічна реакція системи контролю серцево-судинної системи на фізіологічні зміни відображається на ВСР і артеріальному тиску

У 1972 роках Ерік Кристофер Зіман запропонував модель серцебиття з використанням нелінійного диференціального рівняння [45, с.60-87][46] (2.1) на основі рівняння Ван дер Поля-Лієнарда:

$$\begin{cases} \varepsilon \frac{dx}{dt} = -(x^3 + Tx + y) \\ \frac{dy}{dt} = x - x_d + (x_d - x_s)u \end{cases}, \quad (2.1)$$

де x — довжина м'язового волокна серця;

y — електрична керуюча змінна, яка запускає електрохімічну хвилю, що веде до скорочення серця (систоли);

x_d — середня довжина м'яза в діастолі;

$T > 0$ — напруга м'язів і пов'язана з артеріальним тиском (чим вище тиск, тим вище напруга м’язів).

Функція u визначається наступним чином:

— якщо $y_s \leq y \leq y_d$ та x такий, що $x^3 + Tx + y > 0$, то $u = 1$;

— якщо $y > y_d$, тоді $u = 1$;

— інакше $u = 0$.

2.3 Моделювання тиску

Артеріальний тиск, що складається з повторюваних систол і діастол, контролюється центральними механізмами для підтримки кровотоку.

Будь-яка модель системи кровообігу вибирає рівень деталізації, який є бажаним або практичним, залежно від того, що ми хочемо вивчати. Короткий фізіологічний огляд системи кровообігу показує, що кров тече від серця в артерії, потім потрапляє в капіляри, де відбувається газообмін, і, нарешті, транспортується назад до серця через вени. Судини, по яких збіднена киснем кров від правого шлуночка серця до легенів, утворюють легеневе коло кровообігу. Судини, по яких збагачена киснем кров від лівого шлуночка до всіх клітин тіла, називаються системним кровообігом.

При роботі з простими судинами системні капіляри об'єднуються в одну судину. Опірна судина має негнучкі стінки, тому об’єм крові, який вона може вмістити, є фіксованим. Це означає, що витрата в посудину повинна дорівнювати витраті з посудини (2.2):

$$Q = \frac{\Delta P}{R}, \qquad (2.2)$$

де Q — потік;

ΔP — перепад тиску;

R — протилежність до цього потоку.

Це аналог закону Ома, який стверджує, що струм прямо пропорційний падінню електричного потенціалу на дроті/резисторі. У еластичній посудині стінки гнучкі. Об'єм у посудині залежить від тиску, а податливість є мірою того, наскільки стінки можуть розтягнутися. Загальне припущення полягає в тому, що опір кровотоку незначний, а звідси випливає, що перепад тиску над

еластичною судиною дорівнює нулю. Оскільки стінки судин еластичні, заданий тиск P визначає об’єм крові V, який може вмістити судина (2.3):

$$V = CP, \tag{2.3}$$

де P — заданий тиск;

V — об’єм крові;

Знову маємо електричний аналог, який є конденсатором. Конденсатор утворений двома провідниками, розділеними ізолятором, наприклад двома паралельними металевими пластинами з повітрям між ними, як пояснюється в [47, с. 909-910]. Основна відмінність між цими двома системами полягає в тому, що в електричному корпусі не буде накопичення заряду немає різниці потенціалів на конденсаторі, але в посудині буде трохи крові, навіть якщо тиск дорівнює нулю. Щоб врахувати це доповнення, маємо (2.4):

$$V = V_d + CP, \tag{2.4}$$

де V_d — об’єм без тиску.

Потік через податливу посудину також можна отримати зі спрощення рівняння Стока, як описано Кінером і Снейдом [48, с.45-47]. Посудину вважають циліндром, а потік задано (2.5):

$$Q_\infty \frac{dP}{dx} A^2 \tag{2.5}$$

де A — площа поперечного перерізу.

Припускається, що площа поперечного перерізу A лінійно залежить від тиску. Тоді посудина довжиною L має перепад тиску та опір (2.6):

$$\mathrm{RQ} = \frac{1}{3\gamma}(1+\gamma \mathrm{P})^3 \Big|_{\mathrm{P}_1}^{\mathrm{P}_0} \tag{2.6}$$

$$\frac{\mathrm{V}}{\mathrm{V_d}} = \frac{3}{4}\left[\frac{(1+\gamma \mathrm{P})^4 \Big|_{\mathrm{P}_1}^{\mathrm{P}_0}}{(1+\gamma \mathrm{P})^3 \Big|_{\mathrm{P}_1}^{\mathrm{P}_0}}\right] \tag{2.7}$$

де $\gamma = c/A_d$;

c — це відповідність по довжині;

A_d — площа поперечного перерізу посудини при нульовому тиску.

Включаючи лише лінійні терміни, отримуємо лінійну модель для об'єму посудини податливості (2.9):

$$RQ = P_0 - P_1, \tag{2.8}$$

$$\frac{V}{V_d} = 1 + \frac{\gamma}{2}(P_0 + P_1) \tag{2.9}$$

2.4 Моделювання системи кровотоку

Моделі Віндкессель (з нім. — сосуд) відображають поведінку системи кровотоку за допомогою використання електричного кола для відображення фізичних параметрів.

Потік крові в організмі є складною системою, яка на мікроскопічному рівні потребує вичерпного аналізу внутрішньоклітинних взаємодій і нелінійних наближень для точного відображення поведінки потоку. Щоб уникнути складності моделі та швидко отримати важливу інформацію з ключових факторів, біомедичні інженери використовують зосереджені параметри, щоб

узагальнити поведінку системи та забезпечити швидкий аналіз основних компонентів, що контролюють кровотік.

Перша модель із зосередженими параметрами для відображення кровотоку була розроблена в 1899 році німецьким фізіологом Отто Франком [49], який відповідно назвав модель «Віндкессель». Ймовірно, Отто Франк розробив цю модель, спостерігаючи за рухом пари через двигун, як назва перекладається як «котловий вітер» [50]. Загалом, мета моделі пов’язати кровотік через артеріальну систему з точки зору взаємодії між об’ємом крові, що викачується з лівого шлуночка за один удар (тобто ударний об’єм), і здатністю аорти та великих еластичних артерій скорочуватися [51]. Основна концепція моделі Віндкессель проілюстрована на рис 2.1.

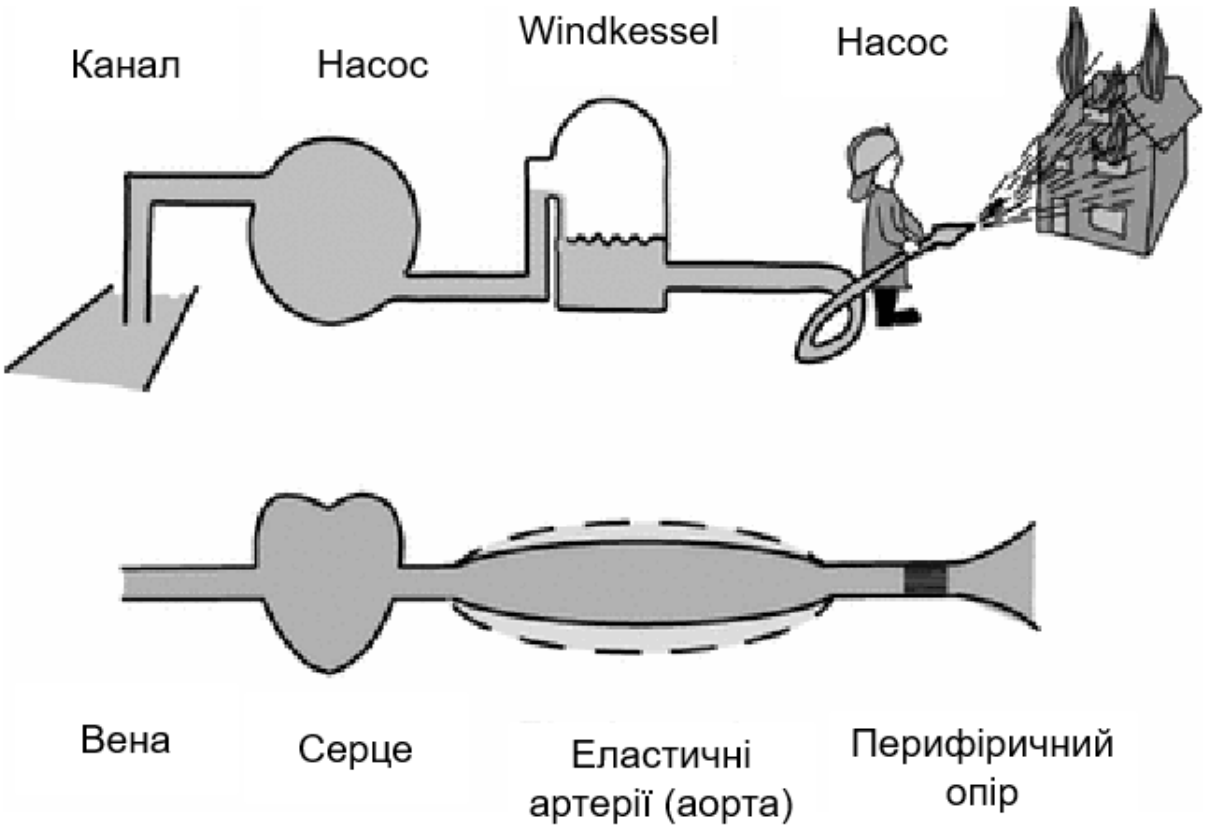


Рис. 2.1 — Концепція моделі Віндкессель

Канал аналогічний венам, насос — аналог серцю, Віндкессель — аналог великим еластичним артеріям (аорті) і носик — аналог периферичного опору. Було визначено, що коли податливість, аортальні клапани та периферичний опір об’єднуються, периферичний потік стає постійним.

У практичних застосуваннях поведінка моделі може передбачити зміни в серцево-судинній системі, коли один із параметрів (або вхідних даних) збурюється.

Модель Отто Франка в даний час відома як двоелементна модель Віндкессель, оскільки вона включає два елементи — опір і податливість, які відображають загальний периферичний опір кровотоку і загальну податливість артерій кров'яному тиску відповідно [52, с.66-79]. Ця двоелементна модель пояснювала поступове зниження тиску в аорті під час діастоли, але не систоли. Справді, подальші дослідження привели до трьохелементної моделі Віндкессель [53, с.66-77], яка включала імпеданс, а саме (аортальний або легеневий) опір змінному кровотоку. Однак ця модель створювала помилки на низьких частотах, пов'язані з елементом імпедансу та переоцінкою відповідності. Таким чином, була побудована чотириелементна модель з додаванням елемента інертності, який відображає загальну артеріальну інертність [54, с.78-88]. Хоча досить важко оцінити належну інерційність за допомогою цієї моделі, вона точно оцінює відповідність і опір добре реагують на всіх частотах.

2.5 Серцеві хвороби

Як вже було зазначено вище, однією з основних причин смертності сьогодні є захворювання серця. Отже, стає необхідним переконатися, що серцево-судинна система повинна залишатися здоровою.

За даними звіту, щороку від серцевих нападів помирає 7 000 000 людей [55]. Відповідно до за даними ВООЗ, у 2016 році близько 17,9 мільйона людей померли від серцевих хвороб [55], це 31% від усіх смертей.

Хвороба серця — це термін, що охоплює будь-які захворювання серця. На відміну від серцево-судинних захворювань, які описують проблеми з кровоносними судинами та системою кровообігу, а також із серцем. Серцева хвороба відноситься до проблем і деформацій самого серця. Серцево-судинні захворювання зазвичай стосуються станів, які включають звуження або блокування кровоносних судин, що може призвести до серцевого нападу, болю в грудях (стенокардії) або інсульту. Інші захворювання серця, наприклад ті, що впливають на серцевий м’яз, клапани або ритм, також вважаються формами серцевих захворювань.

Будь-яка технологія, яка може допомогти діагностувати серцеві захворювання до того, як буде завдано значної шкоди, виявиться корисною для збереження грошей і, що важливіше, життя людей. Методи аналізу даних можуть бути корисними для прогнозування захворювань серця. Прогностичні моделі можна створювати, знаходячи раніше невідомі закономірності та тренди в базах даних і використовуючи отриману інформацію. Інтелектуальний аналіз даних означає отримання знань із великої кількості даних. Машинне навчання — це технологія, яка може допомогти діагностувати хвороби серця ще до того, як буде задано шкоди.

Методів для передбачення серцевих хвороб є декілька.

2.5.1 Алгоритм K-Nearest Neighbors

Алгоритм k-найближчих сусідів (k-Nearest Neighbors, k-NN) — це тип алгоритму керованого навчання, який використовується як для регресії, так і для класифікації. k-NN намагається передбачити правильний клас для тестових

даних, обчислюючи відстань між тестовими даними та всіма точками навчання [56]. Потім обирається k кількість балів, яка є близькою до даних тесту. Алгоритм k-NN розраховує ймовірність того, що тестові дані належать до класів навчальних даних «k», і буде вибрано клас із найвищою ймовірністю. У випадку регресії значення є середнім із «k» вибраних точок навчання.

Продуктивність класифікатора k-найближчих сусідів сильно залежить від метрики відстані, яка використовується для ідентифікації k найближчих сусідів точок запиту. На практиці зазвичай використовується стандартна евклідова відстань. Коли точки даних є неперервними атрибутами, цей алгоритм використовує евклідову відстань (ED), задану рівнянням (2.10), або відстань Манхеттена (MD), визначену в рівнянні (2.11), для вимірювання відстані між точками даних.

$$ED = (\sum_{i=1}^{k}(x_i - y_i)^2)^{1/2} \qquad (2.10)$$

$$MD = \sum_{i=1}^{k}|(x_i - y_i)^2| \qquad (2.11)$$

Відстань гальмування (HD) визначається в рівнянні (2.12), яке можна використовувати, коли атрибути — це категорії:

$$HD = \sum_{i=0}^{k}|(x_i - y_i)^2| \qquad (2.12)$$

KNN розглядає всі особливості однакової ваги. Це метод вибору для задачі класифікації, коли набір даних не має численних змінних. Це стосується цього набору даних, використаного в цій роботі. Якщо функцій занадто багато, найкращий спосіб — зробити вибір функції.

Алгоритм k-NN широко використовується в медицині: він потребує величезний обсяг інформації, щоб можна було поставити діагноз на основі

історичних даних [57, с.56-59]. Він зосереджений на обчисленні ймовірності виникнення конкретного захворювання за допомогою унікального алгоритму. Цей алгоритм підвищує точність такої діагностики. Алгоритм можна використовувати для покращення автоматизованої діагностики, яка включає діагностику кількох захворювань із подібними симптомами.

Наприклад, алгоритм k-NN може бути використаний для класифікації нових пацієнтів на основі їх медичної історії та даних про симптоми. Крім того, він може бути використаний для прогнозування результатів лікування пацієнтів, зокрема, може допомогти визначити оптимальну дозу ліків для конкретного пацієнта на основі аналізу даних про попередні лікування та їх результати.

Алгоритм k-NN є особливо корисним в області медицини, оскільки він дозволяє використовувати існуючі дані для прийняття рішень з високою точністю і швидкістю. Крім того, він може бути застосований до різних видів даних, таких як клінічні дані, зображення, генетичні дані тощо [58].

2.5.2 Алгоритм Random Forest

Випадковий ліс, як випливає з назви, складається з великої кількості окремих дерев рішень, які працюють як ансамбль. Кожне окреме дерево у випадковому лісі видає прогнозований клас, і клас з найбільшою кількістю голосів стає прогнозом нашої моделі.

Випадковий ліс — це оцінювач, який підбирає кілька класифікаторів дерева рішень для різних підвибірок набору даних і використовує усереднення для підвищення точності прогнозування та контролю надмірного підгонки. У регресійній моделі прогноз ґрунтується на незалежній змінній. Випадковий ліс у регресії працює над створенням безлічі дерев рішень під час навчання та виведенням класу, який є режимом середнього прогнозування окремих дерев.

Цей ансамблевий класифікатор створює кілька дерев рішень і об'єднує їх, щоб отримати найкращий результат. Для навчання дерева він в основному застосовує початкове агрегування або пакетування. Для заданих даних $X=\{x_1, x_2, x_3, \ldots, x_n\}$, з відповідями $Y=\{x_1, x_2, x_3, \ldots, x_n\}$, що повторює пакетування від $b = 1$ до B . Невидимі зразки x′ створюються шляхом усереднення прогнозів $\sum B\ b=1\ fb(x')$ для кожного окремого дерева на x' (2.13):

$$j = \frac{1}{B}\sum_{b=1}^{B} fb(x') \qquad (2.13)$$

Невизначеність прогнозу на цьому дереві визначається його стандартним відхиленням (2.14):

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}\left(fb(x')-\hat{f}\right)^2}{B-1}} \qquad (2.14)$$

Random Forest є одним з найбільш популярних алгоритмів машинного навчання, що використовуються в медицині. Він використовується для класифікації, кластеризації та передбачення результатів лікування на основі масивів даних[59] [60].

Однією з найбільш поширених областей застосування Random Forest в медицині є діагностика хвороб. Наприклад, він може використовуватись для виявлення хвороб серця, раку, діабету та інших захворювань, для аналізу медичної діагностики раку молочної залози, при гострому інфаркті міокарда та для передбачення гострого ураження нирок Алгоритм Random Forest може обробляти велику кількість даних та здатний працювати з різними типами даних, що забезпечує високу точність та надійність результатів [61, с.65-69].

Крім того, Random Forest може використовуватись для прогнозування результатів лікування та визначення оптимальних методів лікування. Це

дозволяє медичним фахівцям більш точно передбачити ефективність різних методів лікування та вибрати оптимальний підхід до кожного конкретного випадку.

2.5.3 Алгоритм Decision Tree

Дерева рішень — це надійна та ефективна техніка прийняття рішень, яка забезпечує високу точність класифікації з простим представленням зібраних знань, і їх використовували в різних сферах прийняття медичних рішень.

Навчання на основі дерева рішень належить до категорії машинного навчання та є методом навчання під наглядом, подібним до дискримінантного аналізу. Зі статистичної точки зору, порівняно зі звичайним методом класифікації, який припускає, що джерело даних має фіксований розподіл ймовірностей, а потім виконує оцінку параметрів, дерево рішень є строгим «непараметричним» методом, який має кращу гнучкість для високих -розмірні атрибути та класифікаційні мітки вхідних даних і стійкість. Класифікація проблем за деревом рішень базується на логіці, а не на основі статистичних властивостей вибірок, як у традиційних моделях статистичної класифікації. Класифікація дерева рішень займає менше часу, менше комп'ютерних ресурсів і має високу ефективність [62, с.23-46].

Для навчальних вибірок даних D дерева будуються на основі високоентропійних вхідних даних. Ці дерева прості та швидкі, створені за допомогою рекурсивного підходу «розділяй і володарюй» зверху вниз. Обрізка дерев виконується для видалення невідповідних зразків на D (2.15):

$$Entropy = -\sum_{j=1}^{m} p_{ij} log_2 p_{ij} \quad (2.15)$$

Алгоритм Decision Tree є дуже важливим інструментом у медичній діагностиці та лікуванні, оскільки він може допомогти лікарям приймати рішення на основі даних, отриманих від пацієнта. У медицині Decision Tree використовується для багатьох завдань, включаючи діагностику, класифікацію хвороб, визначення прогнозування результатів лікування та багато інших. Наприклад, дерева рішень використовуються у онкології для класифікації типу раку, а також для визначення оптимального лікування для кожного пацієнта, з урахуванням його індивідуальних характеристик [63, с.53-54].

Окрім того, Decision Tree може використовуватися для виявлення ризиків розвитку захворювань та розробки стратегій профілактики. Наприклад, для визначення ризиків серцево-судинних захворювань, для аналізу факторів ризику, таких як вік, стать, куріння, діабет, гіпертонія та інші, і розробити план профілактики для кожного із пацієнтів.

2.5.4 Алгоритм Support Vector Machine

Машина опорних векторів або Support Vector Machine (SVM) є одним із найпопулярніших алгоритмів керованого навчання, який використовується для задач класифікації та регресії. Однак, в основному, він використовується для проблем класифікації в машинному навчанні [64, с.200-209].

Порівняно з новими алгоритмами, такими як нейронні мережі, вони мають дві основні переваги: вищу швидкість і кращу продуктивність з обмеженою кількістю вибірок (у тисячах). Це робить алгоритм дуже придатним для проблем класифікації тексту, де зазвичай є доступ до набору даних щонайбільше з кількох тисяч тегованих зразків.

Нехай навчальні вибірки мають набір даних $Data=\{y_i,x_i\};\ i=1,2,\ ...,\ n$, де $x \in R_n$ представляє i-й вектор, а $y_i \in R_n$ представляє цільовий елемент. Лінійний SVM знаходить оптимальну гіперплощину форми $f(x)=w^Tx+b$, де w —

вектор розмірних коефіцієнтів, а b — зсув. Це робиться шляхом вирішення подальшої задачі оптимізації (2.16):

$$Min_{w,\,b,\,\xi_i}\frac{1}{2}w^2 + C\sum_{I=1}^{N}\xi_i$$

$$\text{де } y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall_i \in \{1,2,\dots,m\} \qquad (2.16)$$

Алгоритм SVM є одним з найбільш популярних алгоритмів машинного навчання в медицині. Його використовують для різних завдань, таких як класифікація хвороб, діагностика, передбачення результатів лікування та інше. SVM працює шляхом побудови гіперплощини, яка максимально відокремлює класи даних в просторі ознак. Це дозволяє алгоритму ефективно класифікувати нові зразки на основі їх ознак. Застосування SVM дозволяє підвищити точність діагностики хвороб та передбачити ризики їх розвитку.

Наприклад, SVM можна використовувати для діагностики раку молочної залози, де на основі ознак, таких як вік, розмір пухлини, її форма, густота та інші, можна класифікувати нові зразки як злокачественні або доброякісні [65, с.345-345]. Також SVM може бути використаний для передбачення результатів лікування. Наприклад, для передбачення того, яка терапія буде ефективною для конкретного пацієнта з раком, на основі ознак, таких як вік, стадія хвороби, історія лікування та інші.

Отже, SVM — потужний алгоритм машинного навчання, який може знайти застосування в різних областях медицини та допомогти в покращенні діагностики та лікування хвороб [66, с. 1063–1076].

2.5.5 Алгоритм Logistic Regression

Логістична регресія — це алгоритм класифікації. Він використовується для прогнозування бінарного результату на основі набору незалежних змінних [64]. Логістична регресія — це правильний тип аналізу, який слід використовувати при роботі з двійковими даними. Ви знаєте, що маєте справу з двійковими даними, коли вихідна або залежна змінна має дихотомічний або категоріальний характер; Іншими словами, якщо він вписується в одну з двох категорій (наприклад, «так» або «ні», «здав» або «не здає» і так далі).

У медицині логістична регресія використовується для прогнозування ризиків виникнення певних захворювань, оцінки ефективності лікування та відбору найбільш ефективних методів профілактики. Наприклад, логістична регресія може використовуватись для прогнозування ризику виникнення серцево-судинних захворювань на основі даних про вік, стать, куріння, артеріальний тиск тощо [63]. Модель логістичної регресії зможе показати, які фактори є найбільш впливовими на ризик виникнення серцево-судинних захворювань та зробити прогноз для кожної окремої людини [66].

Отже, логістична регресія є корисним інструментом у медицині, який дозволяє зробити прогнозування ризиків виникнення певних захворювань та оцінити ефективність лікування.

2.5.6 Алгоритм Naïve Bayes

Naïve Bayes (Наївний Байес), також відомий як імовірнісний класифікатор, заснований на теоремі Байєса. Ця теорема, також відома як правило Байєса, дозволяє «інвертувати» умовні ймовірності. Нагадуємо, що умовні ймовірності представляють ймовірність події, даної при тому, що відбулася якась інша подія, яка представлена наступною формулою (2.17):

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B)}, \quad (2.17)$$

де P(A), P(B) — ймовірності подій A та B;

P(A|B) — умовна ймовірність, подія A за умови істинності події B;

P(B|A) — ймовірність події B та умови виконання події A.

Naïve Bayes класифікатори працюють по-різному, оскільки оперують парою ключових припущень, за що отримали звання «наївних». Цей алгоритм передбачає, що предиктори в наївній байєсівській моделі умовно незалежні або не пов'язані з будь-якою іншою особливістю моделі. Він також передбачає, що всі функції однаково сприяють результату [66].

Хоча ці припущення часто порушуються в реальних сценаріях (наприклад, наступне слово в електронному листі залежить від слова, яке йому передує), це спрощує задачу класифікації, роблячи її більш обчислювально придатною для обробки. Тобто для кожної змінної тепер буде потрібно тільки одна ймовірність, що, в свою чергу, полегшує обчислення моделі [67]. Незважаючи на це нереалістичне припущення про незалежність, алгоритм класифікації працює добре, особливо при невеликих розмірах вибірки.

Алгоритм Naive Bayes може бути застосований для діагностики різних хвороб, включаючи серцеві хвороби, рак, діабет, захворювання нирок тощо. Використовуючи медичні дані та симптоми пацієнтів, модель Naive Bayes може прогнозувати ймовірність наявності хвороби.

2.6 Висновки до розділу 2

Моделювання процесів серця — дуже важливий аспект серцевої фізіології та патології. Серце — це найскладніший орган людського тіла зі складною

структурою та функціями. Моделювання серця допомагає у розумінні, як працює серце та як різні фактори впливають на його роботу.

Моделювання тиску є важливим у вивченні гіпертонії, яка є загальним фактором ризику захворювань серця та інсульту. Моделювання артеріального тиску допомагає розумінню, як різні фактори впливають на тиск.

Крім того, дослідження серцевих хвороб є дуже важливою задачею у медицинській науці, оскільки це дає можливість прогнозувати ризик виникнення серцево-судинних захворювань та вибрати оптимальні методи лікування. У сфері медицини і біології, алгоритми машинного навчання займають все більш важливе місце. Кожен з алгоритмів, таких як k-NN, Decision Tree, Logistic Regression, Random Forest та SVM, має свої переваги та недоліки, що роблять їх відповідними для різних типів задач.

Алгоритм k-NN є простим та ефективним для вирішення задач класифікації та регресії. Він широко використовується для аналізу медичних даних, зокрема, для прогнозування захворювань, діагностики та класифікації медичних зображень.

Decision Tree є простим для зрозуміння та інтерпретації алгоритмом, що дозволяє експертам здійснювати інтерпретацію його рішень. Цей алгоритм використовується в медицині для класифікації пацієнтів на основі їх симптомів та медичної історії, а також для діагностики та прогнозування хвороб.

Random Forest є потужним алгоритмом для класифікації та регресії, який забезпечує високу точність прогнозування. Він використовується для аналізу медичних даних, зокрема, для виявлення ризикових факторів та визначення ефективності лікування.

SVM є потужним алгоритмом для класифікації та регресії, який добре працює з великими наборами даних. Він використовується в медицині для класифікації хвороб, виявлення ризикових факторів та відбору. Крім того, алгоритм SVM дозволяє побудувати границю рішень для класифікації даних,

яка може бути нелінійною. Це особливо корисно в медичних дослідженнях, де зазвичай збираються дані з багатьох різних джерел і вони можуть мати складну залежність між собою.

Logistic Regression може бути використана для визначення ефективності лікування, якщо врахувати фактори, такі як вік, стать, стан здоров'я пацієнта тощо. На основі отриманих результатів можна зробити висновок, яке лікування є найбільш ефективним для даного пацієнта

Алгоритм Naïve Bayes може бути використаний для аналізу медичних зображень, таких як рентгенограми, зображення з магнітно-резонансної томографії (МРТ), зображення з комп'ютерної томографії (КТ) тощо. Застосовуючи Naive Bayes до витягнутих ознак зображень, можна класифікувати аномалії та виявляти патології.

Усі ці алгоритми можуть бути використані для аналізу медичних даних, таких як зображення, біометричні дані, результати досліджень та інші клінічні дані. Вони дозволяють розпізнати певні закономірності в даних, що допомагає медичним фахівцям зробити більш точні діагнози та передбачити можливі наслідки захворювання.

Отже, використання алгоритмів k-NN, Decision Tree, Logistic Regression, Random Forest, Naïve Bayes та Support Vector Machine у медицині є дуже важливим, оскільки вони дозволяють аналізувати великі обсяги даних, знаходити залежності між ними та передбачати ризики та наслідки захворювань. Вони можуть бути використані в різних сферах медицини, включаючи діагностику, лікування, профілактику та наукові дослідження. Узагальнюючи, моделювання серцевих хвороб дозволяє на основі статистичного та машинного навчання прогнозувати ризик виникнення хвороби та вибирати оптимальні методи профілактики та лікування. Це є важливим інструментом для покращення здоров'я та зменшення ризику смертності від серцево-судинних захворювань.

РОЗДІЛ 3 ОЦІНКА ТА ПРОГНОЗУВАННЯ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

Захворювання серця вважаються одними з найнебезпечніших і небезпечних хронічних захворювань у всьому світі. При захворюваннях серця зазвичай серце не в змозі постачати достатню кількість крові до інших частин тіла для нормальної роботи [68, с.30]. Серцева недостатність виникає через закупорку і звуження коронарних артерій. Коронарні артерії відповідають за кровопостачання самого серця [69, с.2039-2048].

Найпоширеніші симптоми серцевих захворювань включають фізичну слабкість, задишку, набряки ніг і втому з супутніми ознаками тощо [70, с.255-260]. Ризик серцевих захворювань може бути збільшений способом життя людини, наприклад курінням, нездоровим харчуванням, високим рівнем холестерину, високим кров'яним тиском, недостатньою фізичною активністю та фітнесом тощо [71, с.1928-1952].

Хвороба серця має кілька типів, серед яких ішемічна хвороба серця (ІХС) є однією з найпоширеніших. Ця хвороба може призвести до болю в грудях, інсульту та інфаркту. Інші типи захворювань серця включають проблеми серцевого ритму, застійну серцеву недостатність, вроджені вади серця (вади серця під час народження) і серцево-судинні захворювання (ССЗ).

Спочатку для ідентифікації захворювань серця використовувалися традиційні методи дослідження, однак вони виявилися складними [72, с.7675-7680]. Через відсутність медичних діагностичних інструментів і медичних експертів, особливо в нерозвинених країнах, діагностика та лікування захворювань серця дуже складні [73, с.382-397]. Однак точна та відповідна діагностика серцевих захворювань є дуже важливою, щоб запобігти подальшому пошкодженню пацієнта [74].

Зазвичай використовуються традиційні інвазивні методи для діагностики захворювань серця, які ґрунтуються на історії хвороби пацієнта, результатах фізичних тестів і дослідженні лікарем пов’язаних симптомів [75]. Серед звичайних методів ангіографія вважається одним з найбільш точних методів виявлення проблем з серцем. Тим не менш, ангіографія має деякі недоліки, такі як висока вартість, різноманітні побічні ефекти та сильні технологічні знання [76]. Звичайні методи часто призводять до неточного діагнозу та займають більше часу через людські помилки. Крім того, це дуже дорогий і обчислювальний підхід для діагностики захворювання та потребує часу для оцінки [77].

3.1 Опис датасету

Датасет складається з чотирнадцяти параметрів:

1. **age** — вік;
2. **sex** — стать (1 = чоловіча, 0 = жіноча);
3. **cp (chest pain)** — характеристика білю у грудях (1 — типова стенокардія; 2 — атипова стенокардія; 3 — біль, який не є стенокардією; 4 — безсимптомна стенокардія);
4. **trestbps (The person's resting blood pressure)** (мм рт. ст. при надходженні в лікарню) — тиск у спокійному стані;
5. **chol** — Вимірювання холестерину людини в мг/дл;
6. **fbs** (The person's fasting blood sugar) — рівень цукру в крові натще (глюкоза) (> 120 mg/dl, 1 = true; 0 = false);
7. **restecg (Resting electrocardiographic measurement)** — Електрокардіографічне вимірювання у стану спокою (0 = normal; 1 = having ST-T wave abnormality; 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria);

8. thalach (The person's maximum heart rate achieved) — максимальний пульс;

9. exang (Exercise induced angina) — наявність стенокардії при фізичному навантаженні (1 = yes; 0 = no);

10. oldpeak (ST depression induced by exercise relative to rest) — ST депресія — це зниження сегмента ST на електрокардіограмі після фізичного навантаження порівняно з відпочинком

11. slope (the slope of the peak exercise ST segment) (1 — upsloping; 2 — flat; 3 — downsloping)

12. ca (The number of major vessels) — аналіз кількості основних вен (0-3), кольорованих флуоресцентною випромінюванням: чим більше рух крові, тим краще, тому люди з значенням "ca" рівним 0 мають більш високий рівень ймовірності наявності серцевої хвороби (0-3);

13. thal — хвороба крові таласемія, що є спадковою хворобою (3 = normal; 6 = fixed defect; 7 = reversable defect);

14. target — наявність серцевої хвороби (0 = no, 1 = yes)

Фактори ризику серцевих захворювань наступні: високий рівень холестерину, високий кров'яний тиск, діабет, вага, сімейна історія та куріння. Згідно з іншим джерелом, основними факторами, які неможливо змінити, є: збільшення віку, чоловічої статі та спадковості.

Основними факторами, які можуть бути змінені, є: куріння, високий рівень холестерину, високий кров'яний тиск, гіподинамія, надмірна вага та діабет. Інші фактори включають стрес, алкоголь і погане харчування.

3.2 Аналіз даних

Проведемо аналіз датасету.

3.2.1 Кореляція даних

Спочатку побудуємо матрицю кореляції, щоб проаналізувати, чи є у якихось даних залежності один між одним (рис. 3.1) та виявити тенденції :

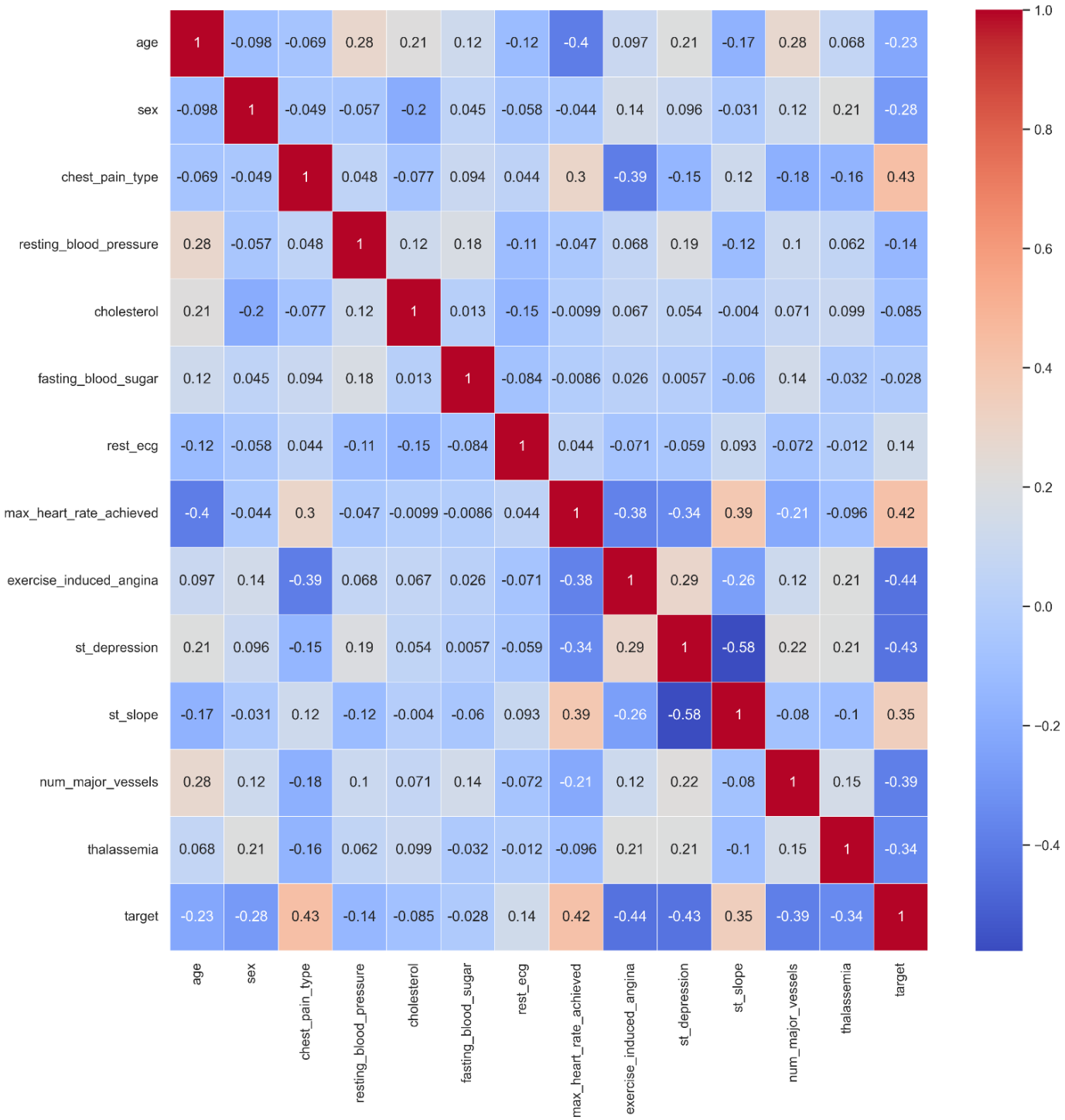


Рис. 3.1 — Матриця кореляції даних

Крім кореляції між функціями, можна ще перевірити кореляцію цільової змінної target (Рис. 3.2):

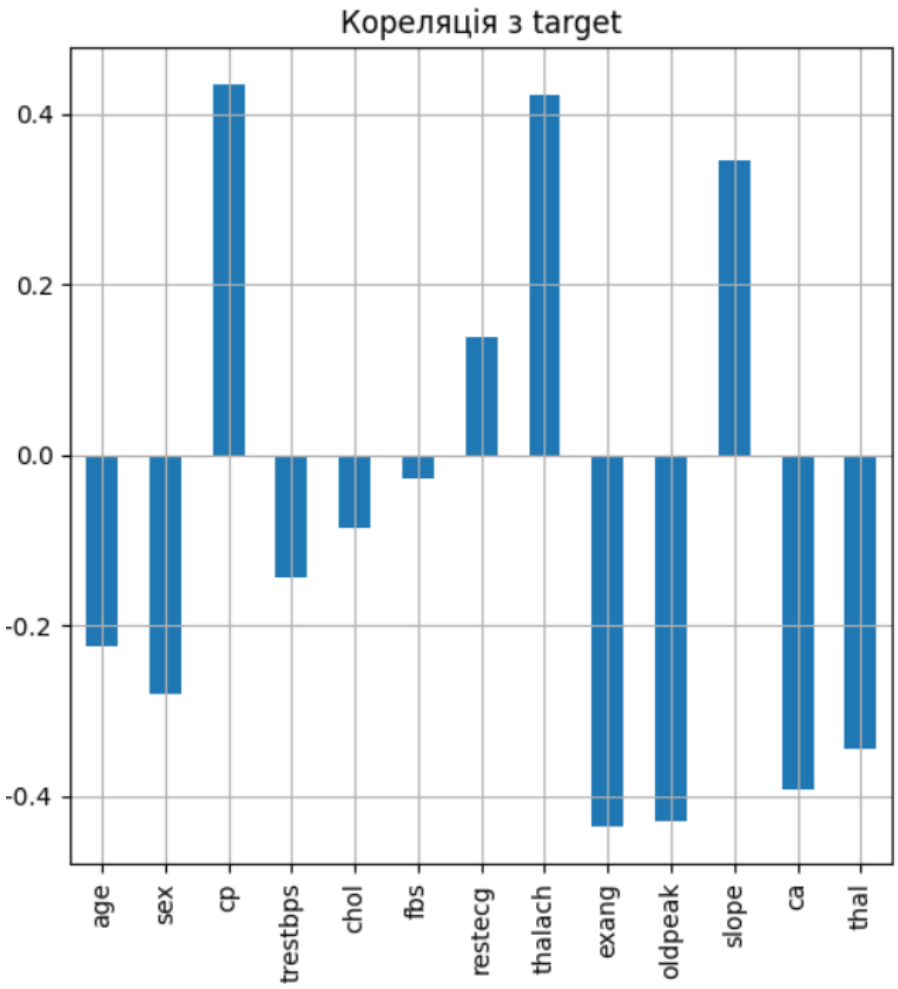


Рис. 3.2 — Кореляція target з іншими змінними

З рис. 3.1 та 3.2 можна побачити, що найбільша додатна кореляція у параметра target (наявність хвороби) з параметрами max_heart_rate_achieved, st_slope та chest_pain_type, а найбільша від'ємна кореляція з параметрами exercise_induced_angina, st_depression, thalassemia та num_major_vessels.

Треба також зазначити, що найсильніша (від'ємна) кореляція наявна між параметрами st_depression та st_slope.

Позитивна кореляція між болем у грудях (chest_pain_type) і target (мета, наш предиктор) має сенс, оскільки більший біль у грудях призводить до більшої ймовірності захворювання серця. chest_pain_type (біль у грудях) є порядковою ознакою з 4 значеннями: значення 1 — типова стенокардія, значення 2 —

атипова стенокардія, значення 3 — неангінозний біль, значення 4 — безсимптомний.

Крім того, ми бачимо негативну кореляцію між стенокардією, спричиненою фізичним навантаженням (exercise induced angina (exang)), і нашим предиктором (target). Це має логіку, оскільки під час фізичних вправ ваше серце потребує більше крові, але звужені артерії сповільнюють кровотік.

3.2.2 Розподіл за статтю

Подивимося на наші дані. На рис. 3.3 можна побачити, який у датасеті розподіл хворих і здорових людей:

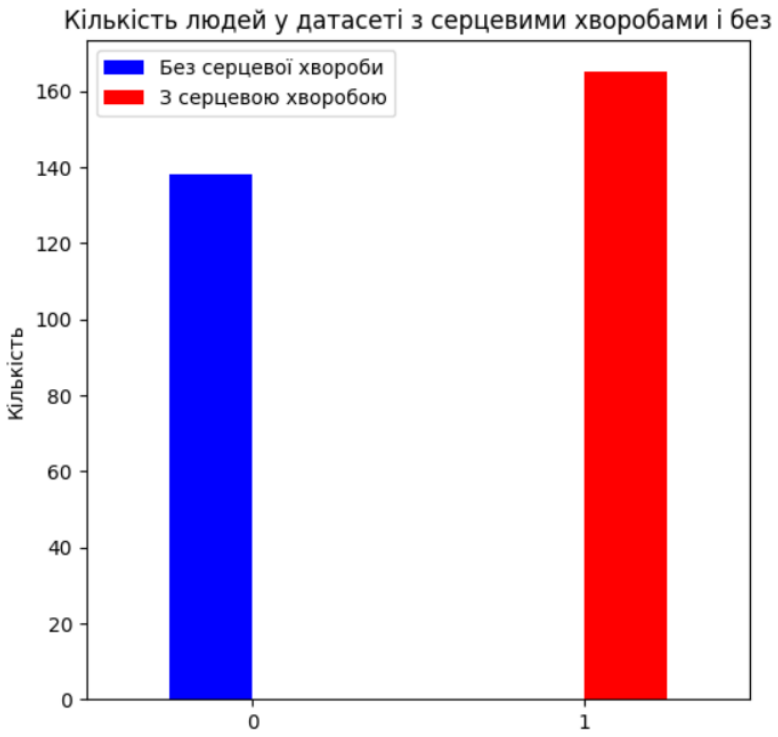


Рис. 3.3 — Розподіл кількості хворих і здорових людей

Стать може впливати на ризик розвитку деяких видів серцевих захворювань. Наприклад, у жінок після менопаузи, коли рівень естрогенів

знижується, ризик розвитку коронарної хвороби серця може збільшуватися. У чоловіків ризик розвитку серцевої недостатності може бути вищим, особливо у тих, у кого є історія куріння або вживання алкоголю. Однак в цілому стать не є єдиним фактором ризику серцевих захворювань, і інші фактори, такі як стиль життя, генетика та інші медичні проблеми, також можуть впливати на ризик розвитку серцевих захворювань

У нашому датасеті 165 людей (54.46%) з хворобами серця (target = 1) та 138 здорових людей (45.54%). На рис. 3.4 можна побачити розподіл кількості людей кожної статі:

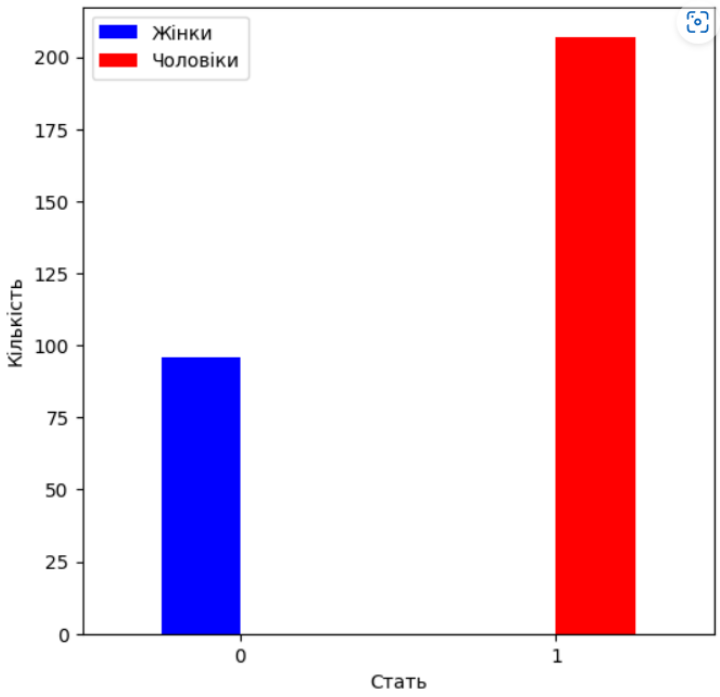


Рис. 3.4 — Розподіл за статтю

Маємо, що жінок у нашому датасеті 96 (31.68%), а чоловіків 207 (68.32%). Якщо дивитися в розрізі, скільки людей кожної статі хворі, а скільки здорові, то з рис. 3.5 можна побачити, що в датасеті більше хворих жінок, а здорових — чоловіків:

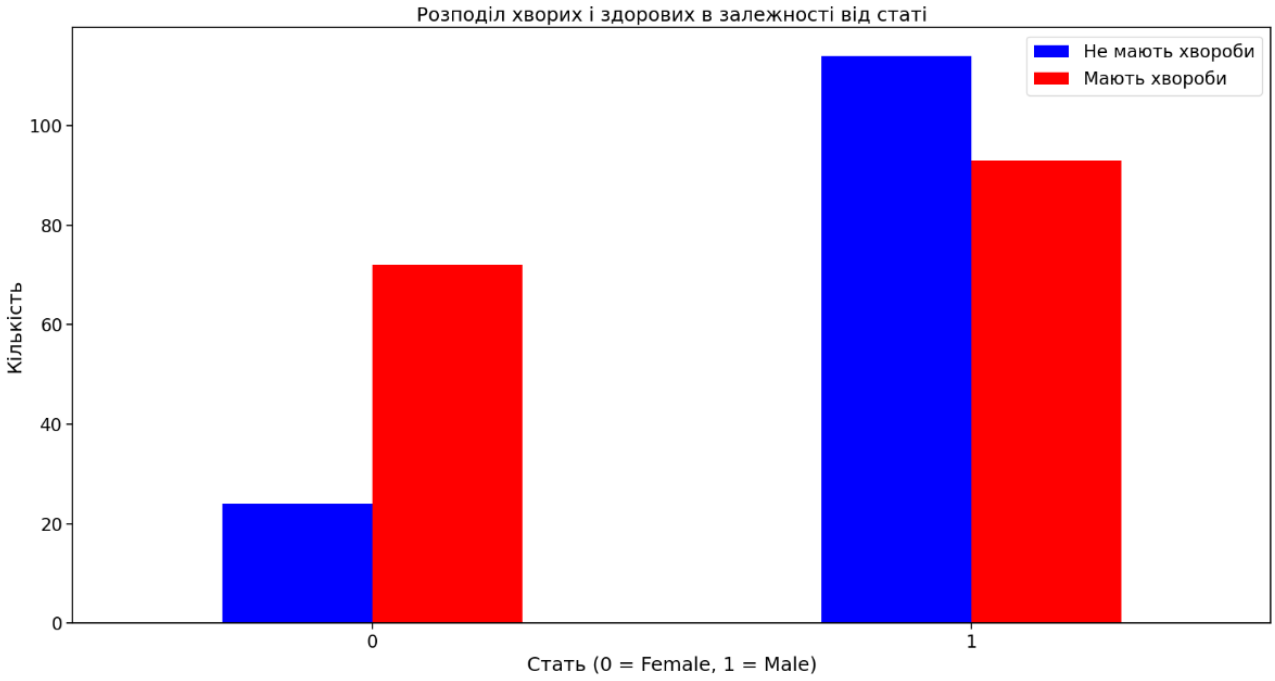


Рис. 3.5 — Розподіл хворих і здорових в залежності від статі

3.2.3 Розподіл за віком

Вік досить сильно впливає на ймовірність захворіти на серцеве захворювання. Зі старінням можуть збільшуватися товщина стінок серця, знижуватися гнучкість клапанів; збільшується ризик розвитку атеросклерозу.

Подивимося на розподіл за роками у датасеті на рис. 3.6:

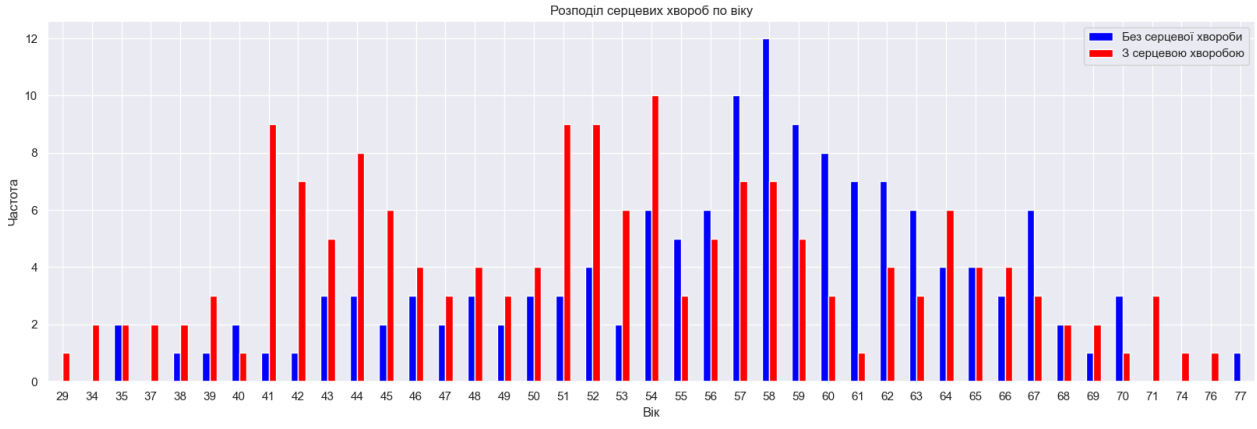


Рис. 3.6 — Розподіл датасету по віку

Як можна побачити, кількість хворих людей переважає над кількістю здорових людей до 54 років, а після вже спостерігається зворотна тенденція.

Подивимося детальніше на десятку за роками за найбільшою кількістю людей у датасеті на рис. 3.7:

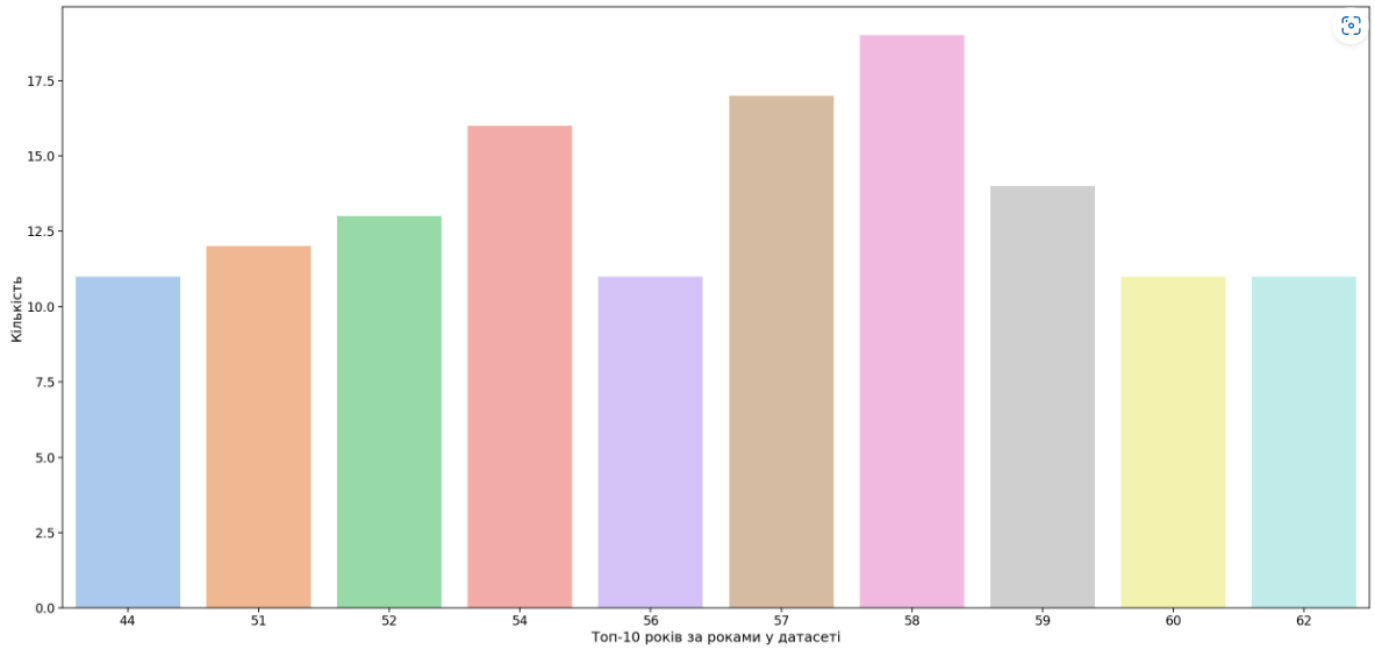


Рис. 3.7 — Топ-10 за віком

Як можна побачити з рис. 3.7, найбільша кількість людей у датасеті знаходиться у віці 58 років.

Якщо ж треба подивитися саме хворих людей, то звернемося до рис. 3.8:

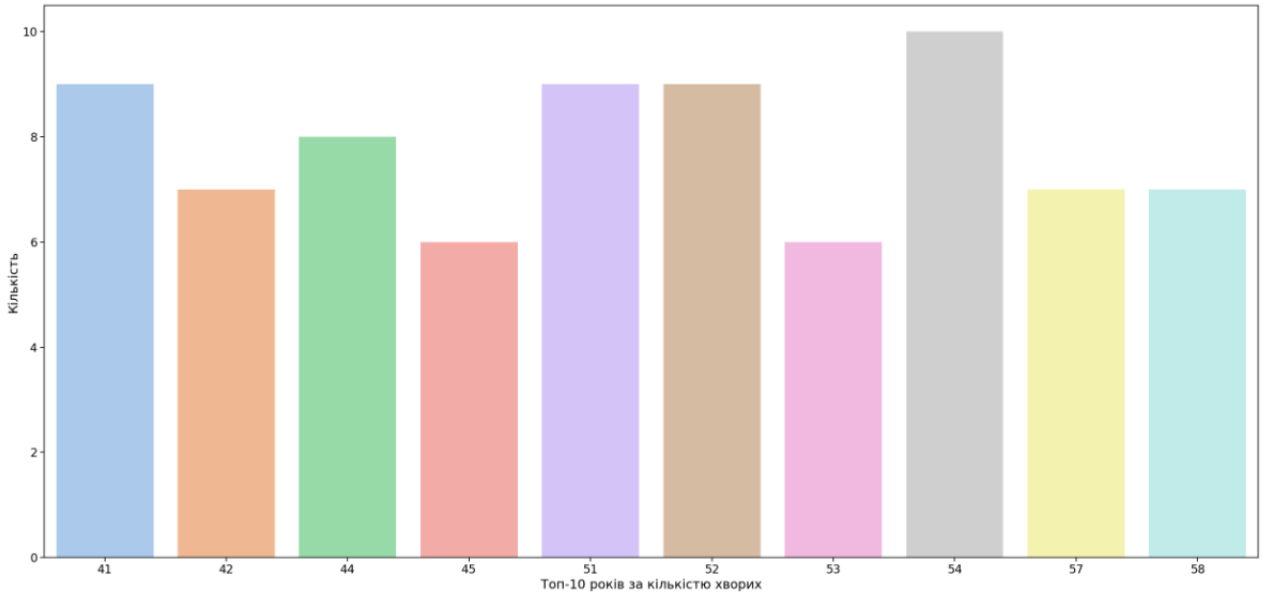


Рис. 3.8 — Топ-10 хворих за віком

Як видно з рис. 3.8, найбільша кількість хворих людей з датасету знаходиться у віці 54 роки. Мінімальний вік у датасеті складає 29 років; максимальний вік у датасеті складає 77 років; середній вік у датасеті це 54 роки. Можна розділити датасет на три категорії: молоді (від 29 до 40), середнього віку (від 40 до 55) та старшого віку (від 55), щоб визначити, яка з груп найбільш уразлива. На рис. 3.9 можна побачити, скільки людей за групами наявні у датасеті, а на рис. 3.10 можна побачити статистику стосовно тільки хворих людей:

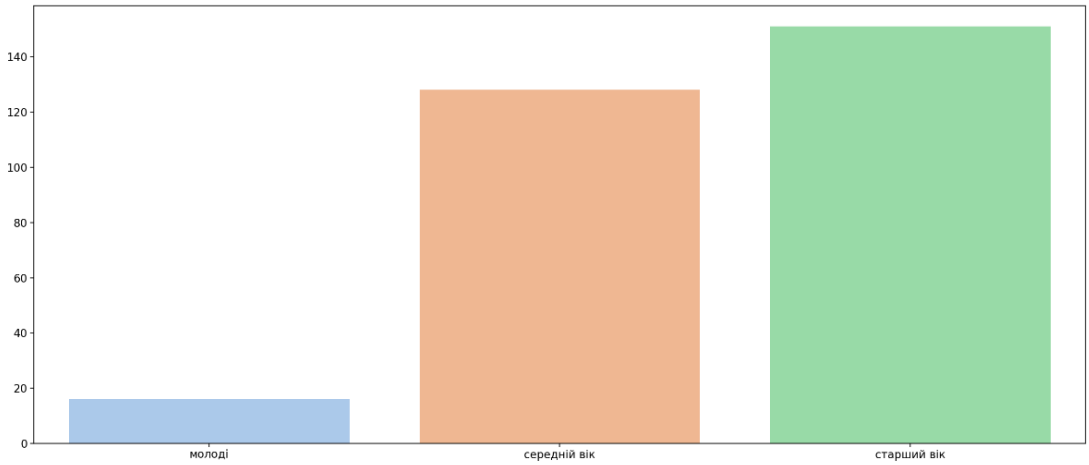


Рис. 3.9 — Розподіл за віковими групами людей у датасеті

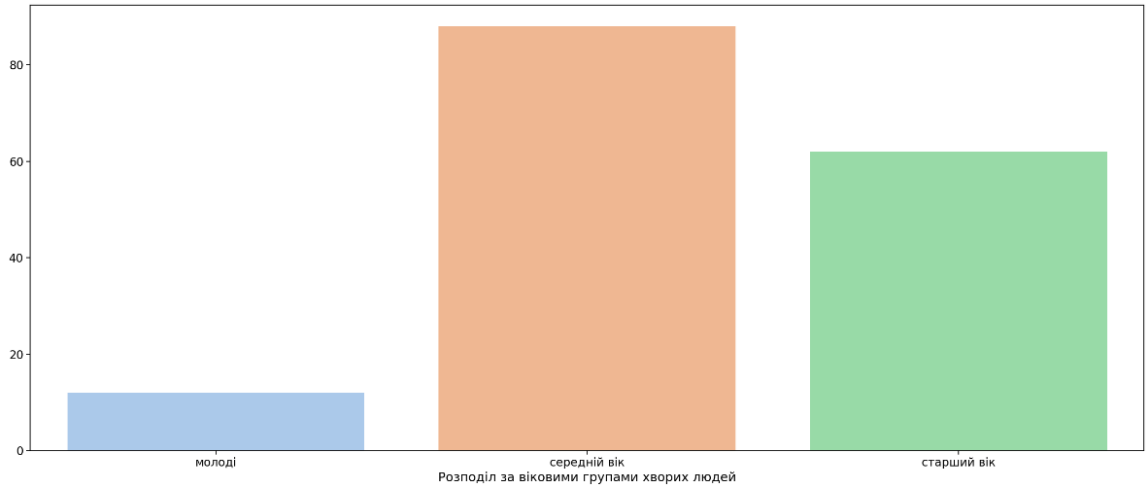


Рис. 3.10 — Розподіл за віковими групами хворих людей

З рис. 3.10 можна зробити висновки, що найбільша кількість людей з серцевими захворюваннями перебуває у середньому віці (від 40 до 55).

Існує зв'язок між віком та ризиком виникнення серцевих захворювань. За даними Всесвітньої організації охорони здоров'я, ризик виникнення серцевих захворювань збільшується з віком. Це пов'язано зі збільшенням віку, зростанням тривалості впливу факторів ризику на серце та змінами в структурі і функції серцево-судинної системи.

Наприклад, з віком може збільшуватися рівень артеріального тиску, що може призвести до розвитку гіпертонії. Гіпертонія є одним із головних факторів ризику серцевих захворювань. Крім того, з віком може збільшуватися відкладення холестерину в артеріях, що сприяє розвитку атеросклерозу. Атеросклероз також може призвести до серцевих захворювань, таких як ішемічна хвороба серця та інфаркт міокарда.

3.2.4 Розподіл за болем у грудях

Тепер розглянемо біль у грудях (chest_pain_type, cp). Як вже було зазначено в описі датасету, у нас є чотири типи болю у грудях. Angina — стенокардія — це тип болю в грудях, викликаний зменшенням припливу крові до серця. Стенокардія є симптомом ішемічної хвороби серця. Стенокардійний біль часто описується як здавлювання, тиск, тяжкість, стиснення або біль у грудях. Може здаватися, що лежить на грудях важка вага.

На рис. 3.11 можна побачити розподіл у датасеті:

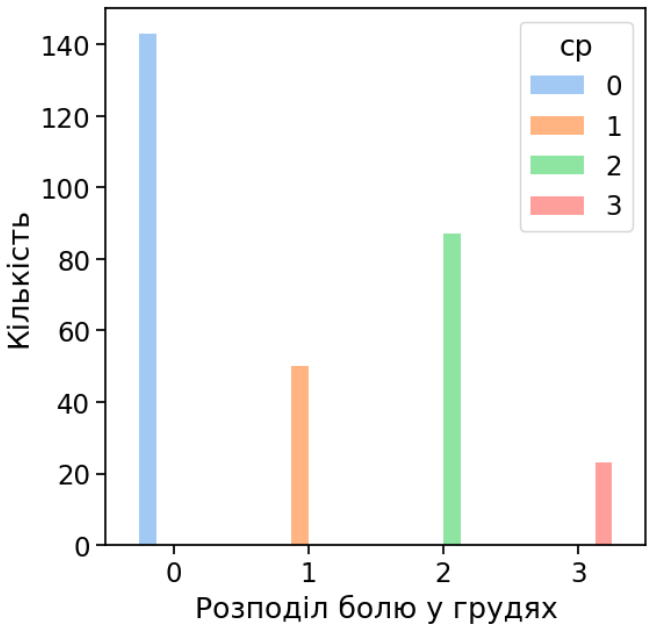


Рис. 3.11 — Розподіл за болем у грудях

Якщо ж дивитися на біль у грудях при розділенні на хворих та здорових людей, то треба звернути увагу на рис. 3.12:

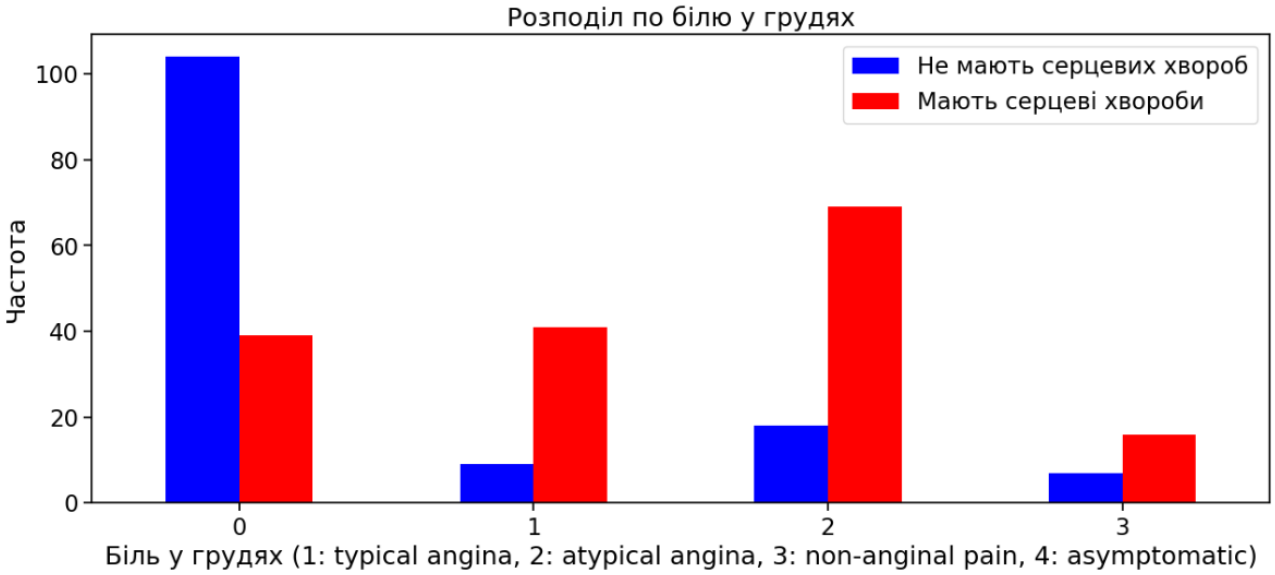


Рис. 3.12 — Розподіл білю у грудях хворих і здорових людей

З рис. 3.12 можна зробити наступні висновки: люди, які відчувають найменший біль у грудях, навряд чи мають серцеві захворювання (ср = 0); люди, які відчувають сильний біль у грудях, швидше за все, мають захворювання серця (ch = 1, 2, 3).

Біль у грудях може бути пов'язаний з різними станами і захворюваннями, однак він часто є одним з симптомів серцевих захворювань. Біль у грудях може виникати через недостатнє постачання крові та кисню до серця внаслідок звуження артерій (стенокардія), а також внаслідок зупинки кровообігу в серці (інфаркт міокарда). Однак, інші стани, такі як хвороба гір, пневмонія та хвороба перикарда також можуть викликати біль у грудях.

3.2.5 Розподіл за рівнем цукру

Аналіз рівня цукру в крові натщесерце відображає рівень цукру (глюкози) у крові. Високий рівень цукру у крові може бути ризиком розвитку захворювань серця. Чим вище рівень цукру у крові, тим більше ймовірність, що судини будуть ушкоджені, буде збільшений запальний процес в організмі, збільшений ризик утворення тромбів.

Все це відповідно може призвести до серцевого набору. Люди з діабетом, особливо ті, у кого не контролюється рівень цукру в крові, мають високий ризик розвитку серцевих захворювань. Крім того, підвищений рівень цукру в крові може сприяти розвитку інших факторів ризику, таких як високий кров'яний тиск та підвищений рівень холестерину, що також можуть сприяти розвитку серцевих захворювань.

На рис. 3.13 показано розподіл рівня цукру натщесерце (fasting_blood_sugar, FBS) у датасеті:

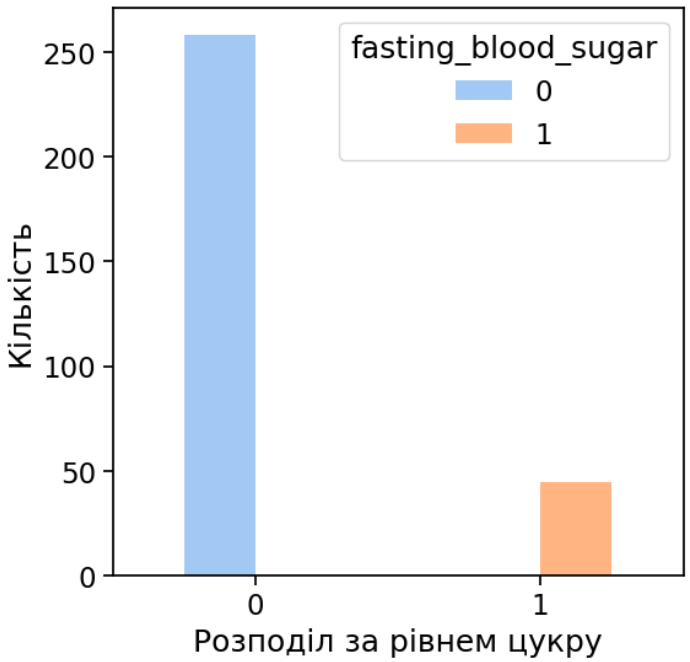


Рис. 3.13 — Розподіл за рівнем цукру

Якщо ж розділити ще на групи здорових і хворих людей, то буде наступна картина (рис. 3.14):

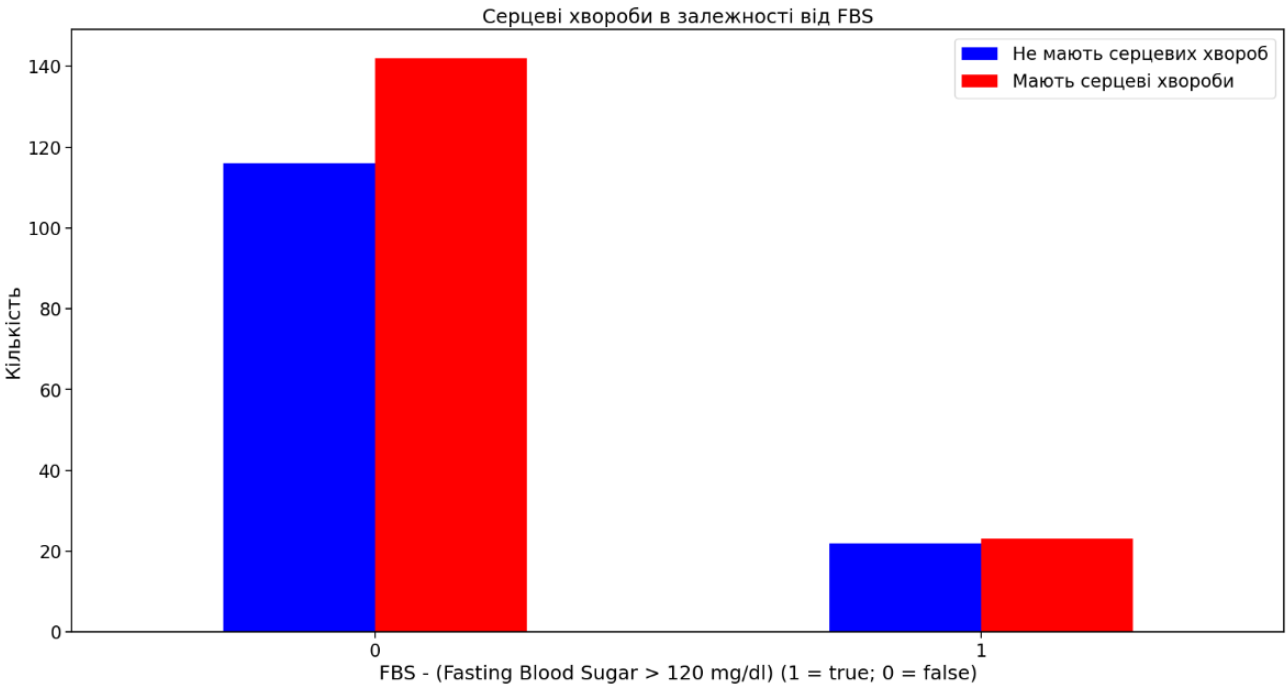


Рис. 3.14 — Розподіл FBS на хворих та здорових людей

Як можна помітити з рис. 3.14, більша кількість людей у датасеті мають рівень FBS < 120 mg/dl.

3.2.6 Розподіл за тиском

Високий артеріальний тиск (або гіпертонія; resting_blood_pressure, trestbps) вважається одним з факторів ризику серцево-судинних захворювань. Сама по собі гіпертонія не являється хворобою, проте вона може бути фактором ризику серцевих хвороб, таких як:

- серцева недостатність. Через надмірний тиск серце працює з надмірним навантаженням, що змушує його зношуватися, що, в свою чергу, призводить до збільшення серця і послаблення його функцій;

- серцевий інфаркт. Підвищений тиск може пошкодити артерії, і якщо якась з артерій буде заблокована тромбом, то це може призвести до інфаркту;
- інсульт. Високий артеріальний тиск може пошкодити судини в головному мозку і збільшити ризик розвитку інсульту;
- аритмія. Підвищений тиск може спричинити тахікардію або фібриляцію передсердь, що може призвести до аритмії.

У нашому датасеті розподіл з вимірами тиску виглядає наступним чином (рис. 3.15):

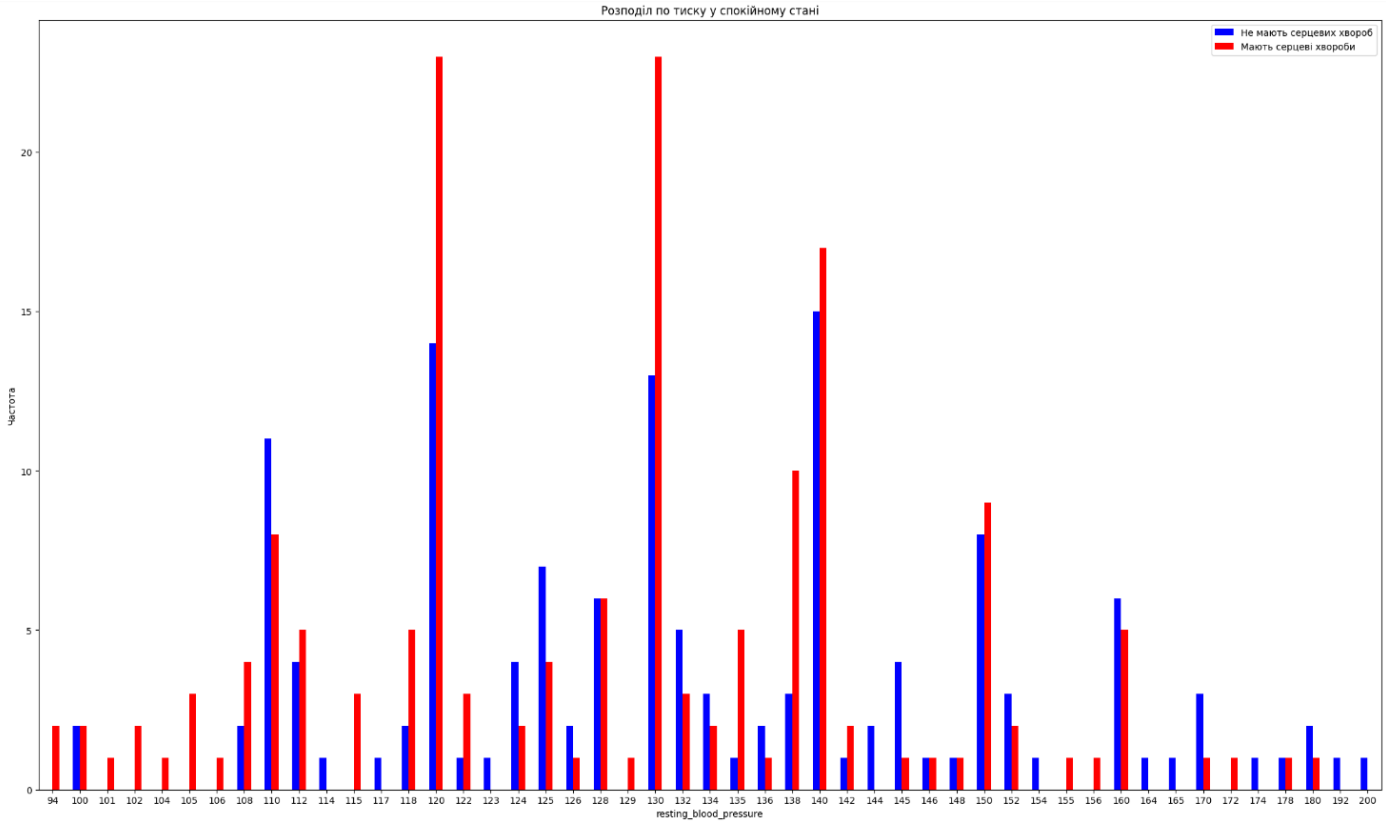


Рис. 3.15 — Розподіл тиску у спокійному стані

Крім того, варто зауважити, що для кожної людини оптимальний тиск може варіюватися в залежності від віку, статі, загального здоров'я та інших факторів. Деякі люди можуть мати природно нижчий або вищий тиск.

3.2.7 Розподіл за основними венами

Процедура просвічування вен флюоресцентним світлом, або венографія, в основному використовується для візуалізації вен та оцінки їх функції. Цей метод застосовується переважно для діагностики венозних захворювань, таких як варикозна хвороба, тромбози, порушення кровообігу в венах тощо.

Зазвичай, флюоресцентне просвічування вен не є безпосереднім методом для діагностики серцевих хвороб, проте в надзвичайних обставинах, коли виникає підозра на тромбоз легеневих артерій, може бути виконана процедура флюоресцентного просвічування вен для виявлення можливих тромбів, які можуть походити з вен нижніх кінцівок та виходити в легеневі артерії.

На рис. 3.16 можна побачити розподіл за венами (num_major_vessels, ca) у нашому датасеті:

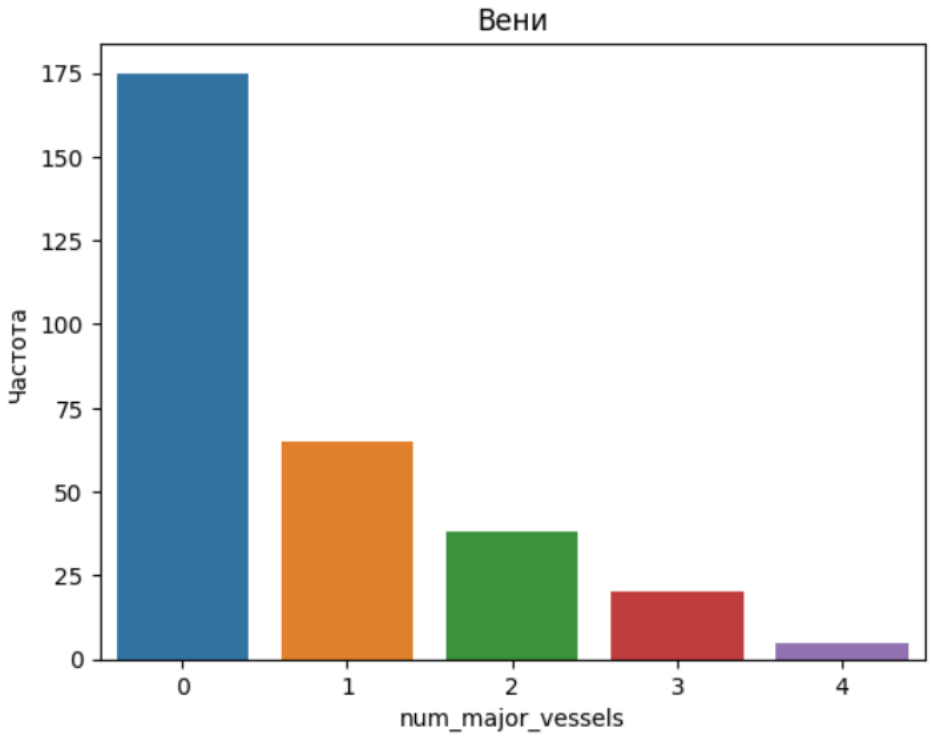


Рис. 3.16 — Розподіл за венами, що видні при флюоресценті

Чим менше видно вен, тим гірше рух крові у організмі і відповідно тим більше ймовірність наявності серцевої хвороби. Якщо ми подивимося на розподіл за хворими та здоровими людьми на рис. 3.17, то побачимо підтвердження даного твердження, бо найбільша кількість хворих має параметр num_major_vessels = 0:

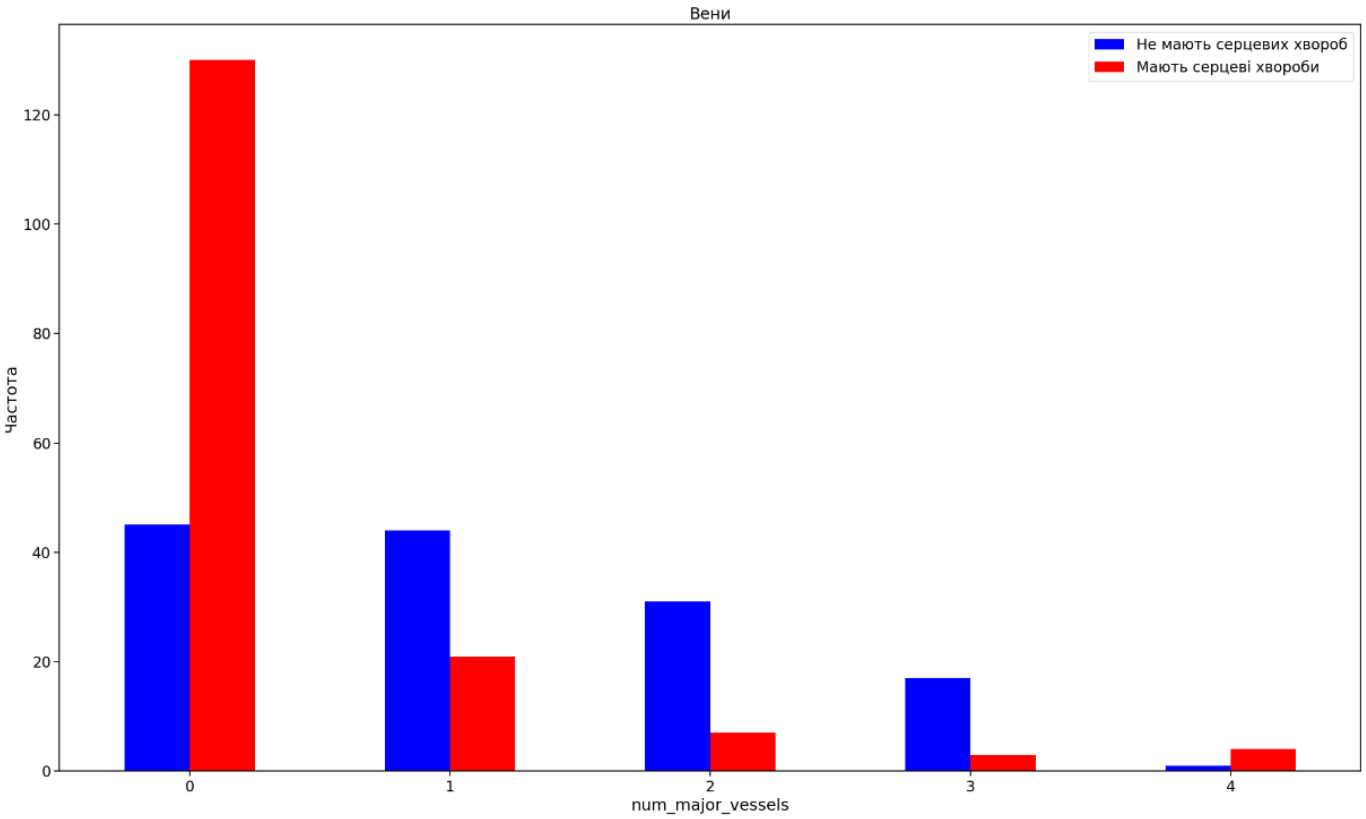


Рис. 3.17 — Розподіл за венами, що видні при флюоресценті, хворі та здорові люди

3.2.8 Розподіл за ЕКГ

ЕКГ (електрокардіограма) є одним з найпоширеніших та найкорисніших методів діагностики серцевих хвороб. ЕКГ дозволяє оцінити ритм серця,

виявити аномалії серцевої діяльності та виявити певні хвороби або ускладнення. ЕКГ може надати важливу інформацію про такі серцеві захворювання:

- аритмія. ЕКГ може виявити неправильні ритми серця, такі як фібриляція передсердь, тахікардія, брадикардія тощо;
- Ішемічна хвороба серця. ЕКГ може виявити зміни в електричній активності серця, які свідчать про недостатнє кровопостачання серця, що може бути спричинено звуженням артерій, що живлять серце;
- тощо.

У нашому датасеті ЕКГ (restecg) має три параметри: 0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria.

На рис. 3.18 зображений розподіл:

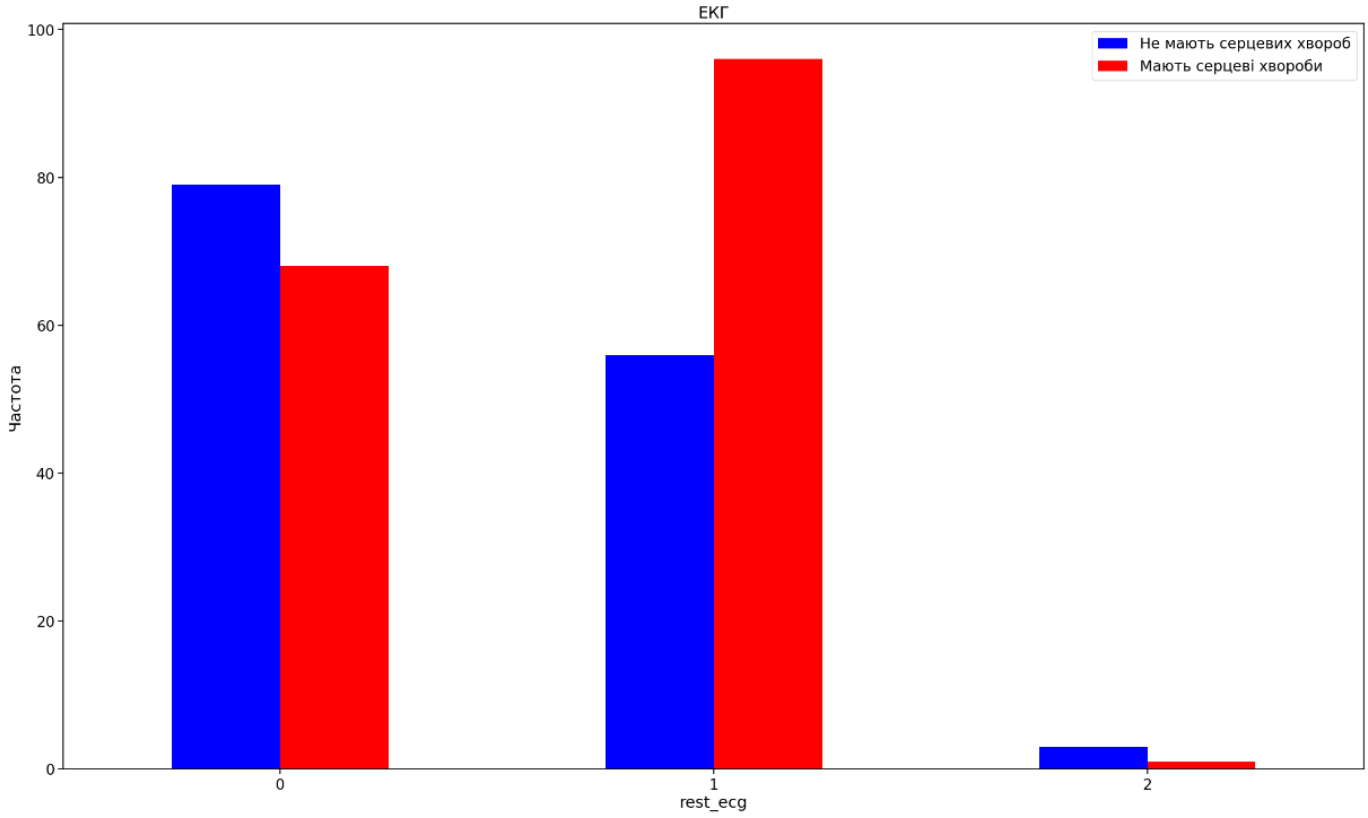


Рис. 3.18 — Розподіл за ЕКГ

Як можна побачити, найбільша кількість хворих людей мають rest_ecg = 1. Тим не менш, і при нормальних показниках ЕКГ все одно можуть бути серцеві хвороби.

3.2.9 Розподіл за стенокардією при фізичному навантаженні (стенокардія напруження)

Стенокардія є одним з проявів ішемічної хвороби серця. Це стан, при якому серцевий м'яз не отримує достатньо кровопостачання та кисню, особливо під час фізичного навантаження або емоційного стресу.

При фізичному навантаженні серце потребує більше кисню для праці. У випадку ішемічної хвороби серця, артерії, що живлять серце, можуть бути частково звужені атеросклеротичними бляшками або тромбами. Це може призводити до недостатнього кровопостачання до серцевого м'яза, що викликає біль або дискомфорт у грудях, відчуття стиснення або важкості. Це є характерним симптомом стенокардії напруження.

На рис. 3.19 можна побачити розподіл даних у датасеті за стенокардією (exang, exercise_induced_angina):

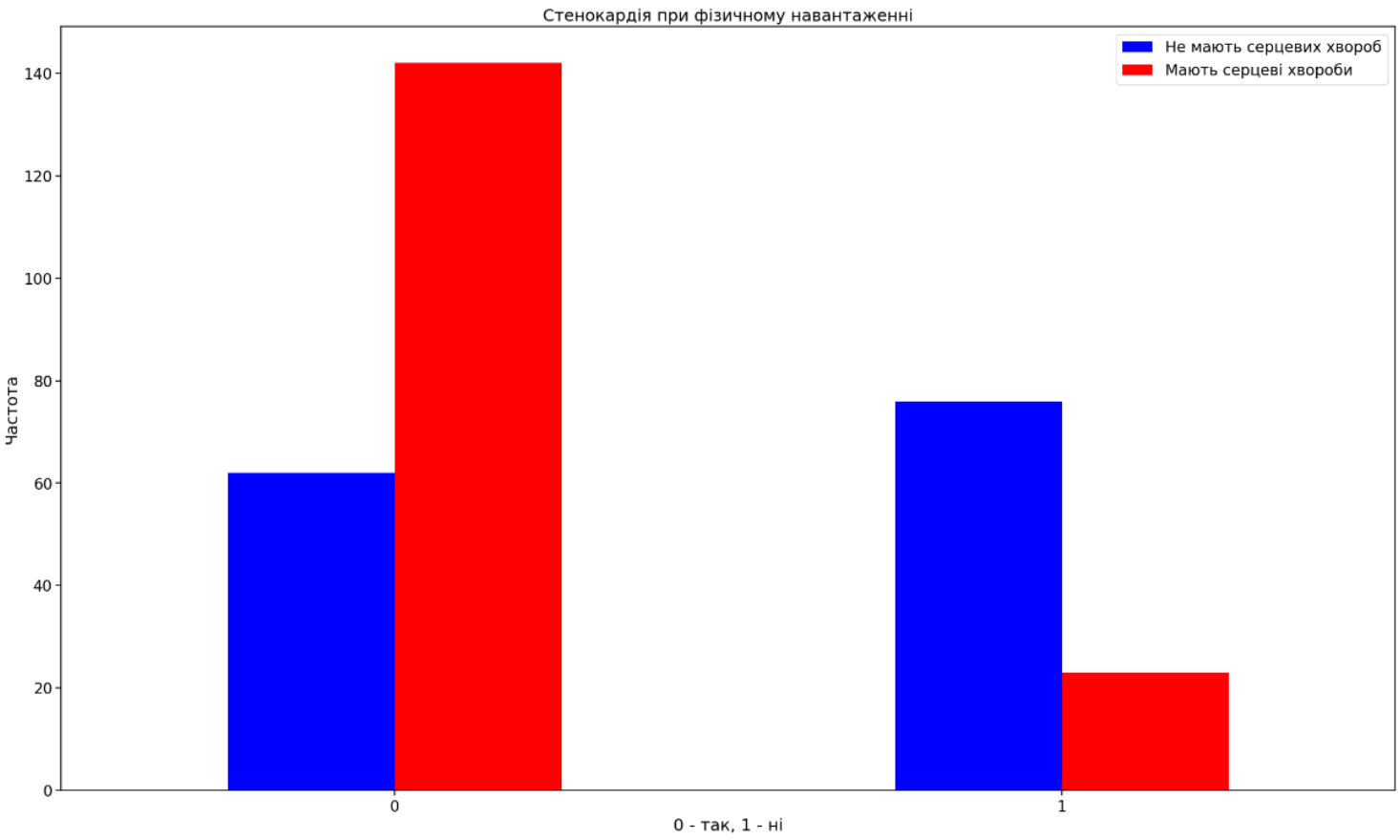


Рис. 3.19 — Розподіл за стенокардією при фізичному навантаженні

Як можна помітити, якщо стенокардія наявна, то з більшою ймовірністю людина буде мати серцеву хворобу.

3.2.10 Розподіл за ST-схилом

ST-схил (ST-slope) — це один з параметрів, що оцінюються на електрокардіограмі (ЕКГ). Він відображає нахил лінії ST-сегмента, який є частинкою ЕКГ зліва від зубця QRS і перед зубцем T. Зміни в ST-схилі можуть бути індикатором можливих серцевих хвороб або ішемічної хвороби серця.

Ось декілька можливих змін ST-схилу, на які поділяється датасет:

1. Підвищений ST-схил. Зміщення ST-схилу вгору може бути ознакою ішемії серця, тобто недостатнього кровопостачання до серцевого м'язу.

Це може бути пов'язано з стенокардією напруження або активною ішемією;

2. Знижений ST-схил. Зміщення ST-схилу вниз може вказувати на інфаркт міокарда або подібні ураження серцевого м'язу;
3. Горбкуватий або сплескоподібний ST-схил. Такі зміни можуть бути ознакою перикардиту, запалення оболонок серця.

На рис. 3.20 показано розподіл у нашому датасеті:

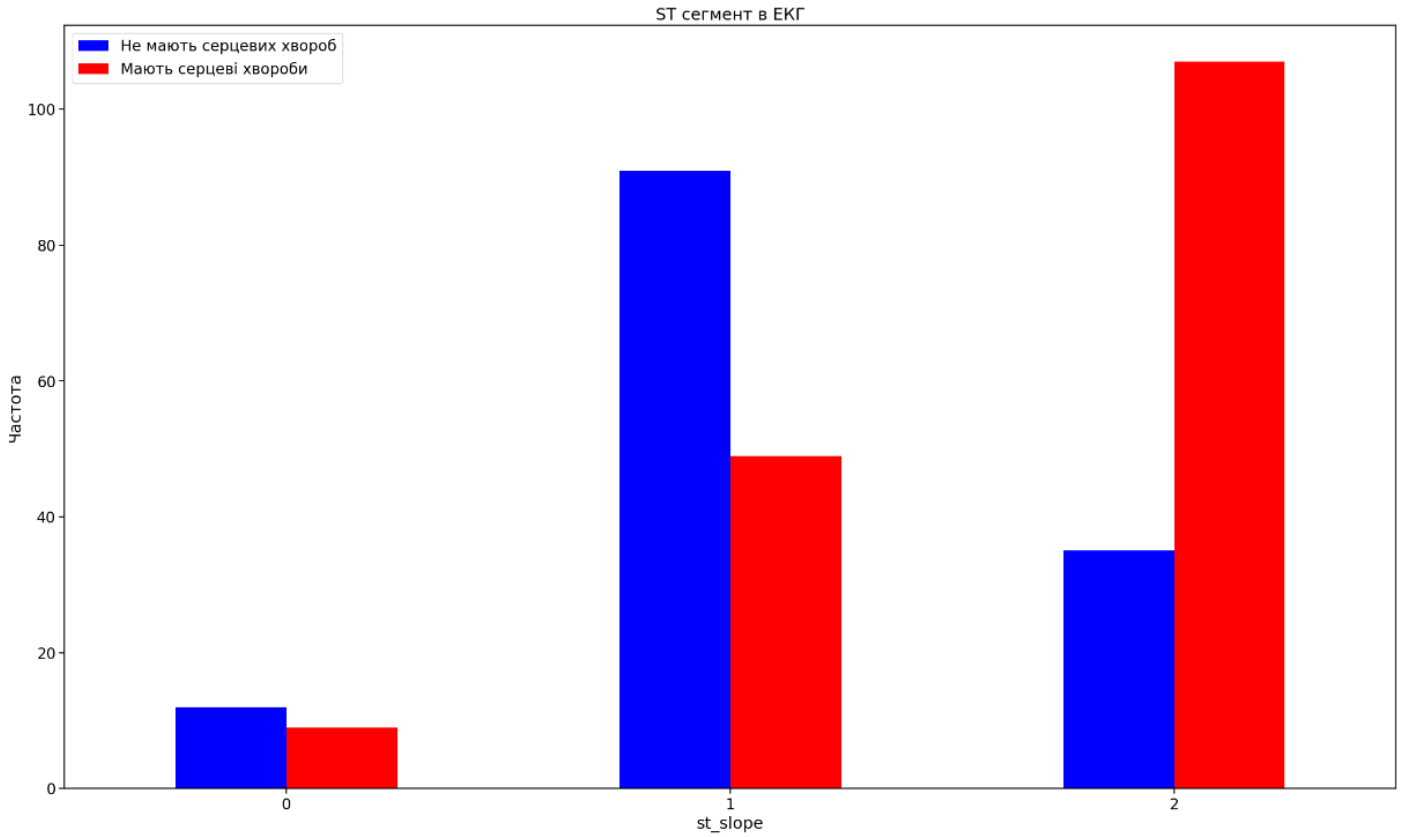


Рис. 3.20 — Розподіл за ST-схилом

З рис. 3.20 видно, що найбільша кількість хворих людей має горбкуватий ST-схил (st_slope = 3).

3.2.11 Розподіл за таласемією

Таласемія — це група генетичних захворювань, характеризуються порушенням синтезу гемоглобіну — білка, що несе кисень у червоних кров'яних клітинах. Це одна з найпоширеніших генетичних хвороб у світі.

Таласемія може призвести до анемії, перенавантаження серця чи навіть до змін структури серця, тому вона теж впливає на можливість захворіти на серцеву хворобу.

На рис. 3.21 можна побачити розподіл нашого датасета стосовно цього параметра:

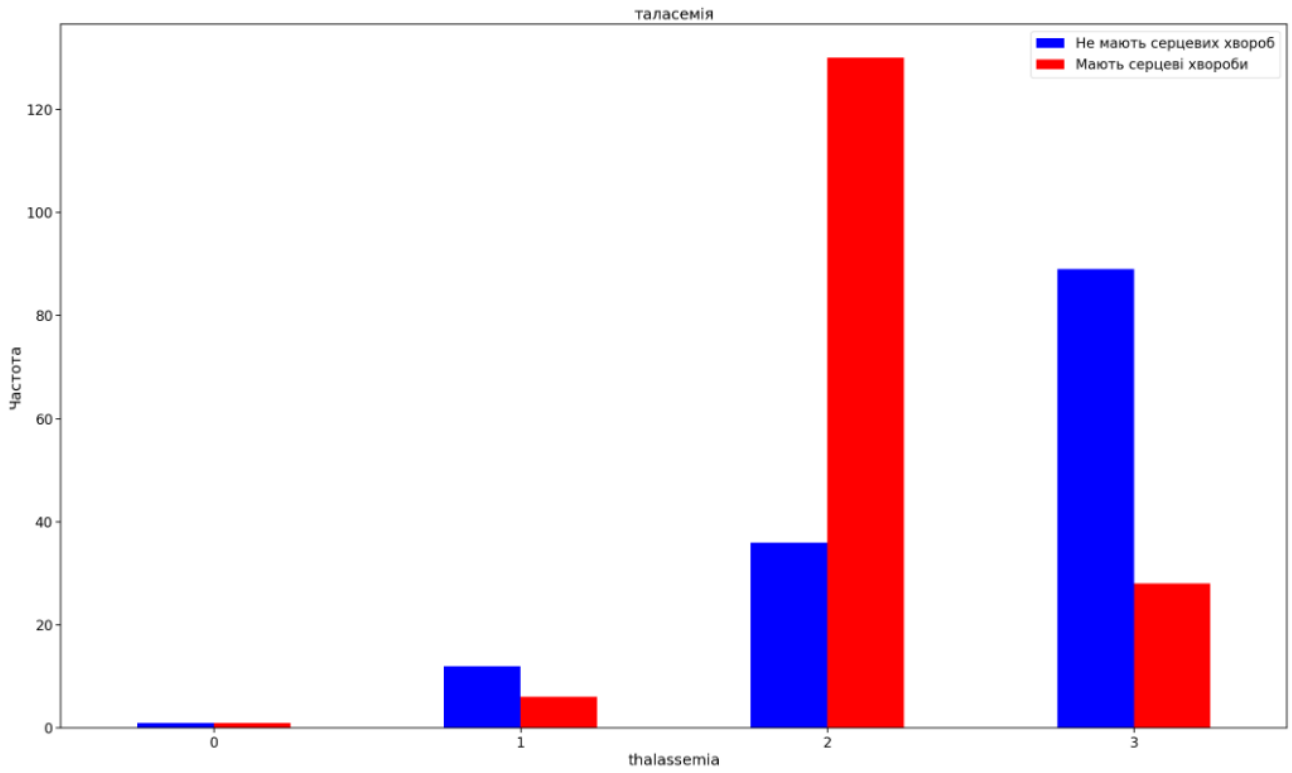


Рис. 3.21 — Розподіл датасету за таласемією

3.2.12 Розподіл за холестерином

Високий рівень холестерину (chol, cholesterol) впливає на ризик розвитку серцевих захворювань: при великій кількості холестерин починає «осідати» на стінках артерій у вигляді жирових нальотів, що призводить до звуження артерій. Це може закінчитися серцевим нападом тощо.

3.2.13 Розподіл за серцебиттям

Рівень максимального серцебиття відображає стан серцево-судинної системи: при низькому максимальному серцебитті можна сказати, що стан системи не дуже добрий.

На рис. 3.21 можна побачити розподіл за максимальним серцебиттям (thalach, max_heart_rate) у нашому датасеті:

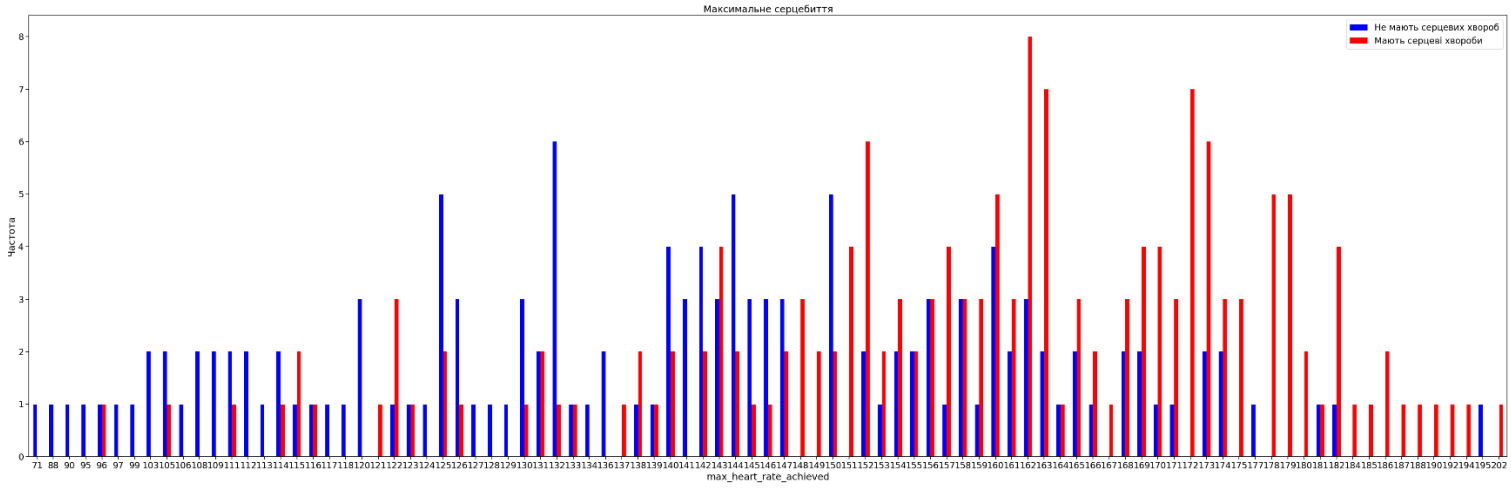


Рис. 3.21 — Розподіл датасету за серцебиттям

3.2.14 Розподіл за Oldpeak

У медицині термін "oldpeak" відноситься до ознаки, яка вимірює депресію сегменту ST на ЕКГ (електрокардіограмі) після виробленого піку навантаження. Ця ознака є важливою в оцінці серцевої функції та виявленні можливих

серцевих хвороб. Депресія сегменту ST вказує на недостатнє кровопостачання серця під час фізичного навантаження, що може бути показником присутності коронарної артеріальної хвороби або інших серцевих проблем.

У контексті серцевих хвороб, це важлива ознака, яка може використовуватися для діагностики, прогнозування та моніторингу стану серця у пацієнтів.

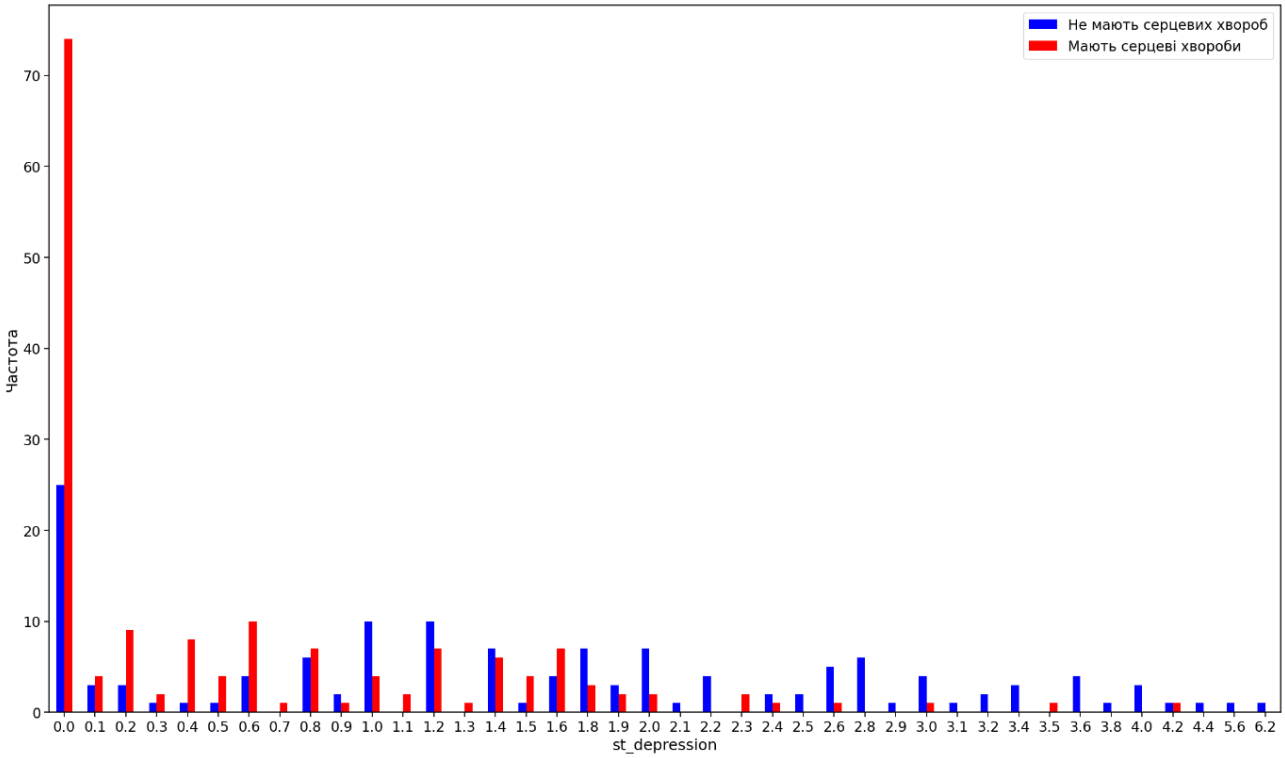


Рис. 3.22 — Розподіл датасету за oldpeak

3.3 Побудова моделей та прогнозування

Датасет було поділено на тестову та навчальну вибірки у співвідношенні 20% на 80%: У навчальній вибірці 242 записів та у тестовій вибірці 61 записів.

Щоб оцінити, наскільки точно модель передбачає правильні значення цільової змінної, треба оцінити продуктивність моделі. Для оцінювання точності моделі, що розроблена в рамках цієї роботи, будуть використовуватися

наступні метрики: Confusion matrix (матриця помилок), Accuracy score (показник точності), Precision (точність), Recall (повнота) та (F1-score) F1-показник.

Confusion Matrix виглядає наступним чином (рис. 3.23):

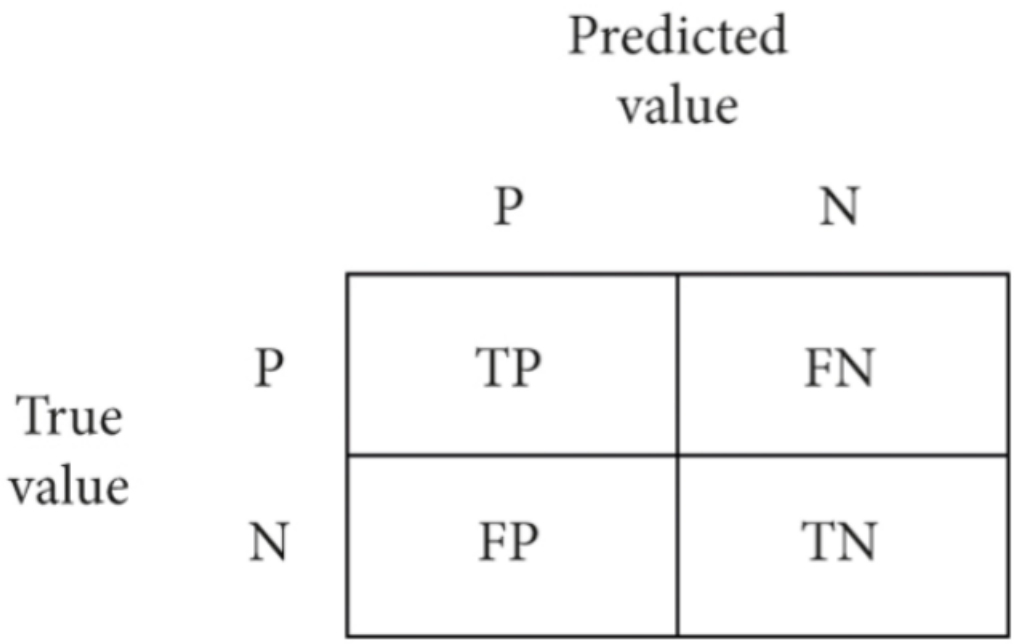


Рис. 3.23 — Загальний вигляд Confusion Matrix

Вона складається з наступних частин:

- TP — True Positive, значення ідентифікувалися правильно;
- FP — False Positive, значення ідентифікувалися як правильні, але насправді вони такими не являються;
- TN — True Negative, значення були негативними і ідентифікувалися як негативні;
- FN — False Negative, значення було правильним, але ідентифікувалося як негативне.

Confusion matrix допомагає виявити, наскільки добре модель розпізнає кожен клас, і визначити сильні та слабкі сторони моделі. Це допомагає прийняти рішення щодо подальшого вдосконалення моделі.

Accuracy будується за допомогою значень з Confusion Matrix і рахується за наступною формулою (3.1):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (3.1)$$

Recall обчислює відношення прогнозованих позитивних міток до загальної кількості позитивних міток (3.2):

$$Recall = \frac{TP}{TP+FN} \quad (3.2)$$

Precision рахується наступним чином (3.3):

$$Precision = \frac{TP}{TP+FP} \quad (3.3)$$

F1-Score залежить і від Recall, і від Precision (3.4):

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

3.3.1 Оцінювання прогнозування Logistic Regression

Матриця помилок для Logistic Regression має наступний вигляд (рис. 3.24):

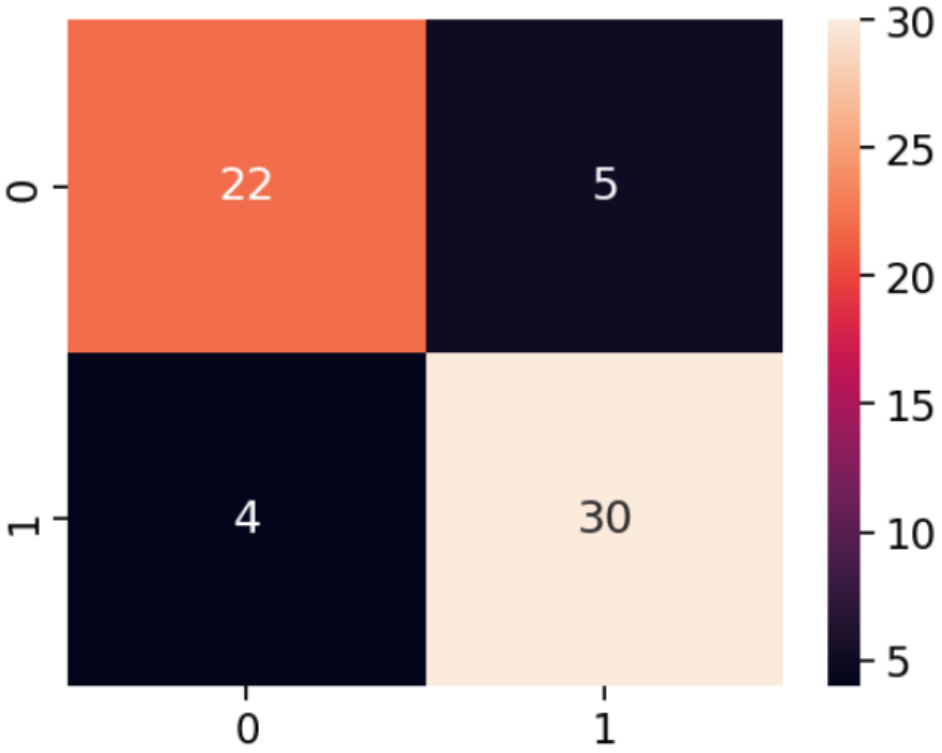


Рис. 3.23 — Confusion Matrix для Logistic Regression

У таблиці 3.1 можна побачити метрики оцінки для Logistic Regression:

Таблиця 3.1 — Оцінки для Logistic Regression

Accuracy	**Recall**	**Precision**	**F1-Score**
0.8525	0.8824	0.8571	0.8696

3.3.2 Оцінювання прогнозування для Random Forest

Confusion Matrix для Random Forest має наступний вигляд (Рис. 3.24):

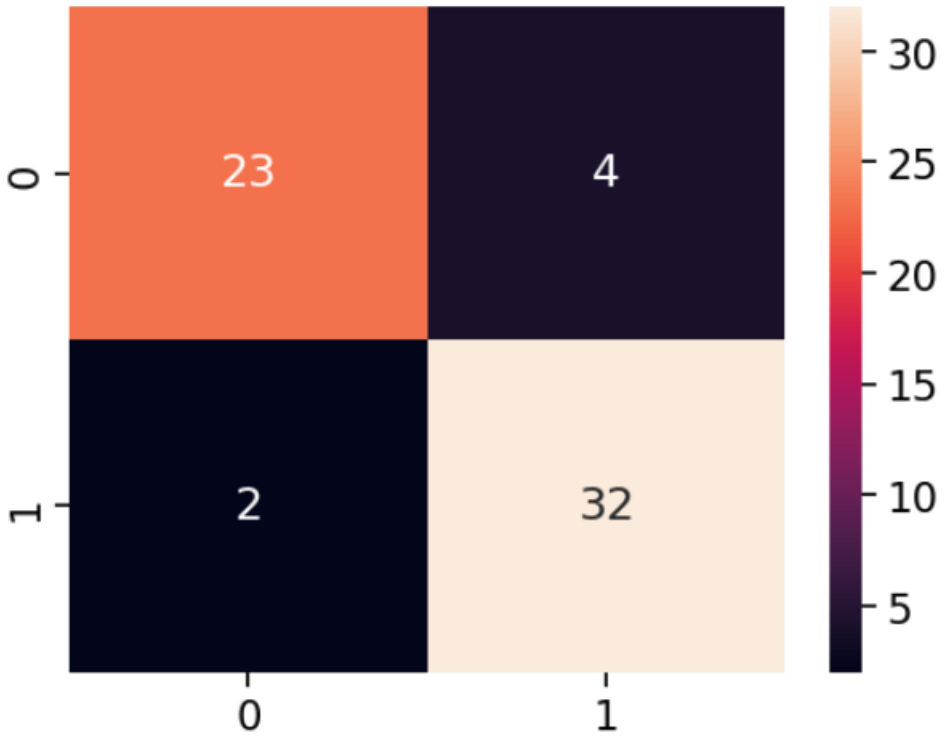


Рис. 3.24 — Confusion Matrix для Random Forest

У таблиці 3.2 можна побачити метрики оцінки для Random Forest:

Таблиця 3.2 — Оцінки для Logistic Regression

Accuracy	Recall	Precision	F1-Score
0.9016	0.9412	0.8889	0.9143

3.3.3 Оцінювання прогнозування для Naïve Bayes

Confusion Matrix для Naïve Bayes має наступний вигляд (Рис. 3.25):

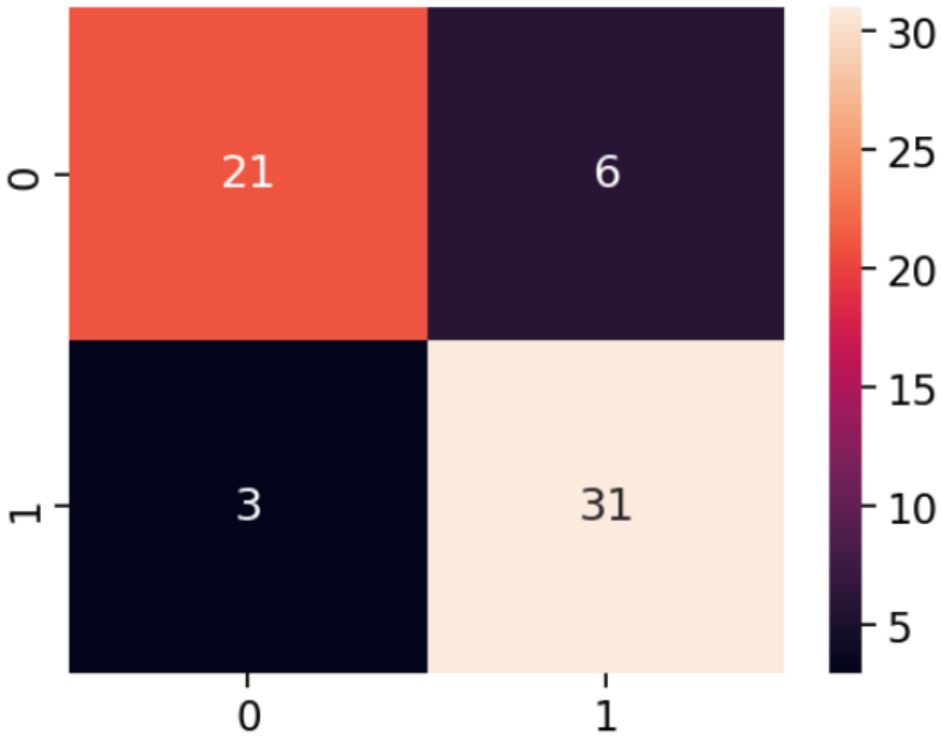


Рис. 3.25 — Confusion Matrix для Naïve Bayes

У таблиці 3.3 можна побачити метрики оцінки для Random Forest:

Таблиця 3.3 — Оцінки для Naïve Bayes

Accuracy	Recall	Precision	F1-Score
0.8525	0.9118	0.8378	0.8732

3.3.4 Оцінювання прогнозування для KNN

Confusion Matrix для KNN має наступний вигляд (Рис. 3.26):

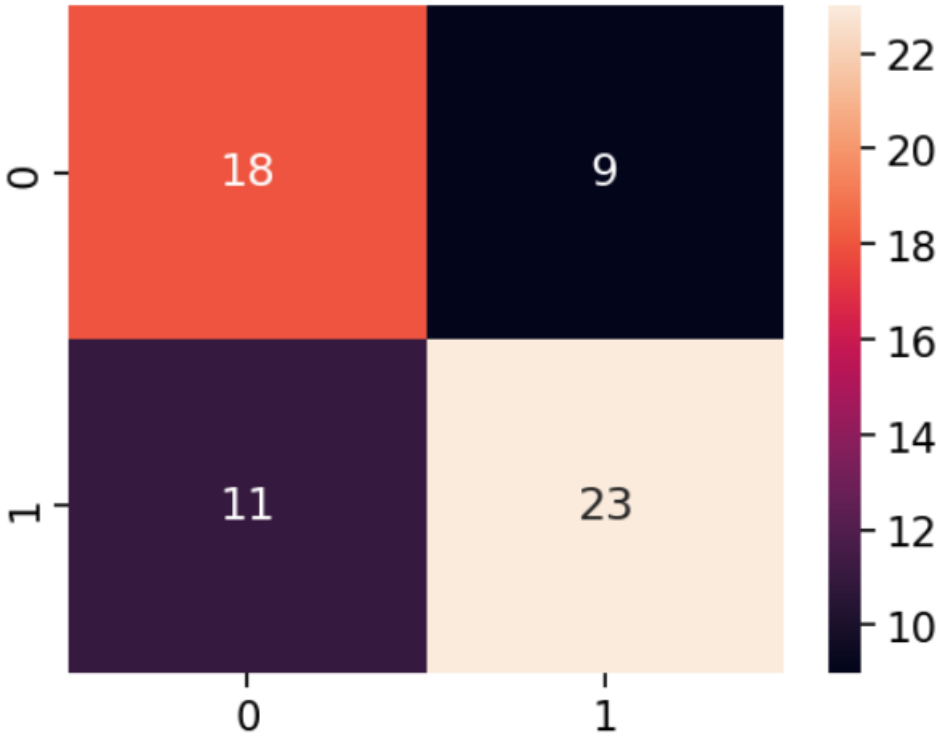


Рис. 3.26 — Confusion Matrix для KNN

У таблиці 3.4 можна побачити метрики оцінки для KNN:

Таблиця 3.4 — Оцінки для KNN

Accuracy	Recall	Precision	F1-Score
0.6393	0.5882	0.7143	0.6452

3.3.5 Оцінювання прогнозування для Decision Tree

Confusion Matrix для Decision Tree має наступний вигляд (Рис. 3.27):

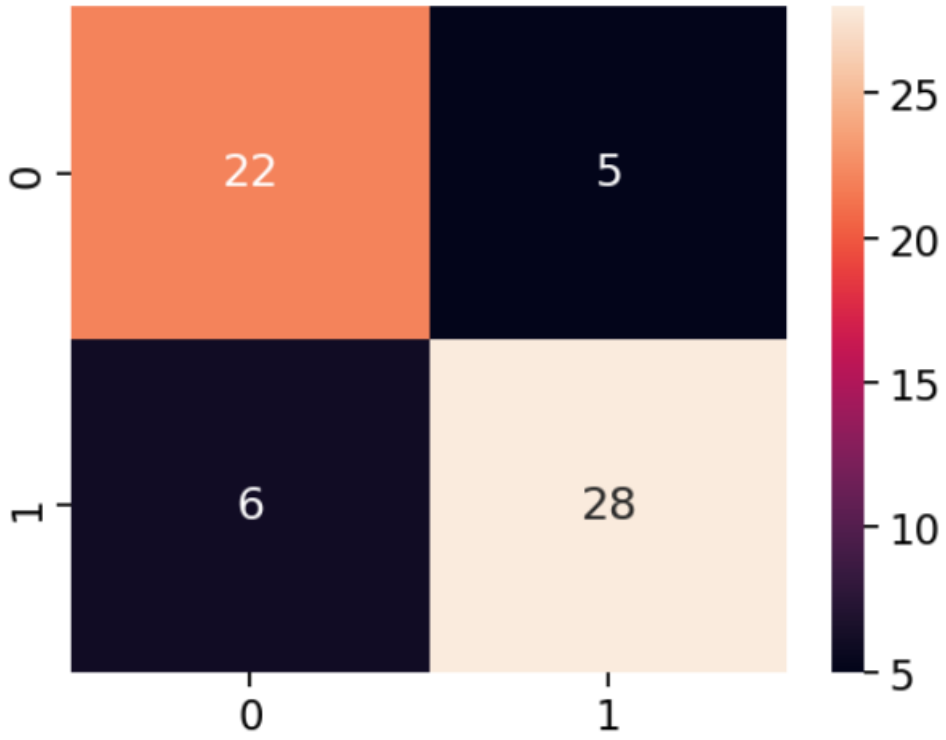


Рис. 3.27 — Confusion Matrix для Decision Tree

У таблиці 3.5 можна побачити метрики оцінки для Decision Tree:

Таблиця 3.5 — Оцінки для Decision Tree

Accuracy	Recall	Precision	F1-Score
0.8197	0.8235	0.8485	0.8358

3.3.6 Оцінювання прогнозування для Support Vector Machine

Confusion Matrix для Support Vector Machine має наступний вигляд (Рис. 3.28):

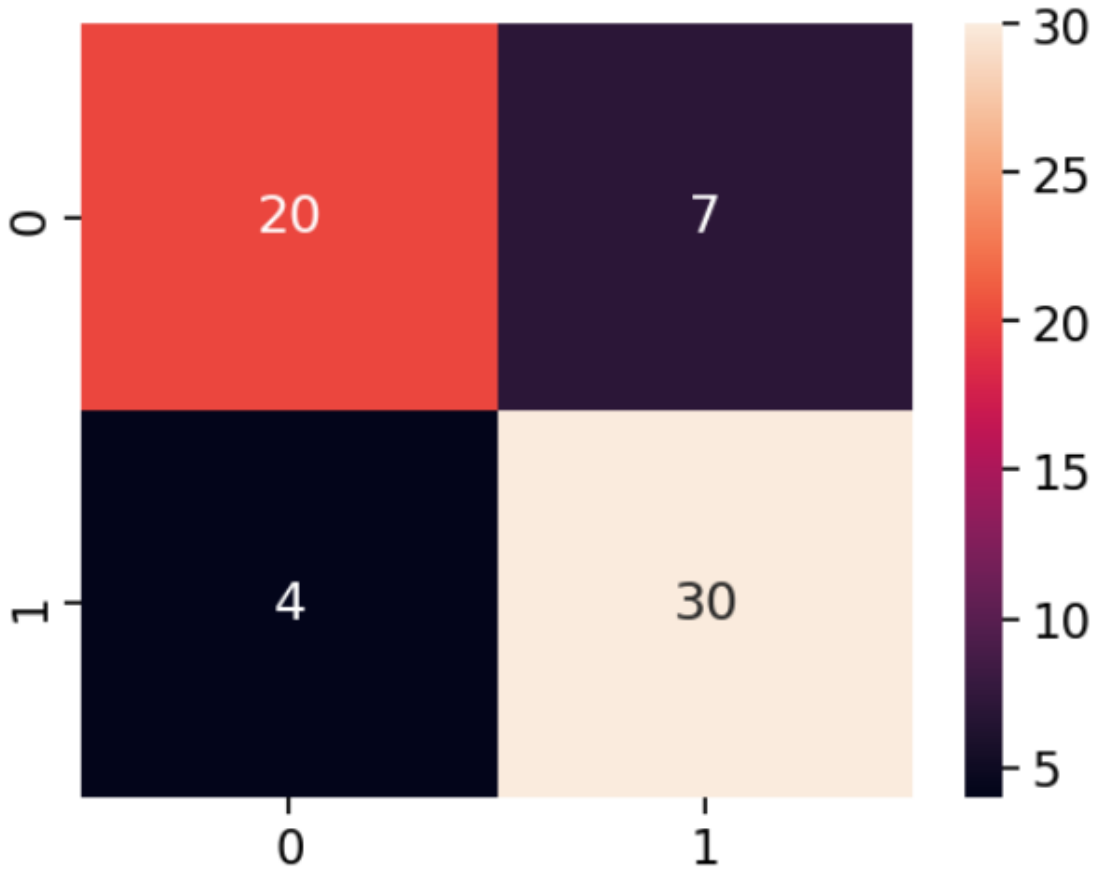


Рис. 3.28 — Confusion Matrix для Support Vector Machine

У таблиці 3.6 можна побачити метрики оцінки для Support Vector Machine:

Таблиця 3.6 — Оцінки для Support Vector Machine

Accuracy	**Recall**	**Precision**	**F1-Score**
0.8197	0.8824	0.8108	0.8451

3.4 Порівняння усіх методів

У таблиці 3.7 можна побачити загальне порівняння оцінок усіх методів, що були використані для моделювання:

Таблиця 3.7 — Порівняння алгоритмів

Метод	**Accuracy**	**Recall**	**Precision**	**F1-Score**
Logistic Regression	0.8525	0.8824	0.8571	0.8696
Random Forest	0.9016	0.9412	0.8889	0.9143
Naïve Bayes	0.8525	0.9118	0.8378	0.8732
KNN	0.6393	0.5882	0.7143	0.6452
Decision Tree	0.8197	0.8235	0.8485	0.8358
SVM	0.8197	0.8824	0.8108	0.8451

З таблиці можна побачити, що найкращу оцінку Accuracy має метод Random Forest; найкращу оцінку Recall має Naïve Bayes; найкраще Precision у алгоритму Random Forest; найкращу оцінку F1-Score має теж Random Forest.

Як видно з таблиці 3.7, алгоритм Random Forest має найвищі оцінки за трьома з чотирьох метрик, відповідно, він дає найкращу точність і результат для прогнозування.

3.5 Висновки до розділу 3

У третьому розділі було проведено аналіз датасету: було наведено кореляцію усіх параметрів та виокремлено ті, які мають найбільший вплив на цільову змінну; було проілюстровано наявність параметрів у датасеті та їхній розподіл по відношенню до хворих та здорових людей; було пояснено наявність

кожного з параметрів, його важливість та вплив на наявність у людини серцевого захворювання.

Крім того, у даному розділі було застосовано порівняно наступні алгоритми машинного навчання: Logistic Regression, Random Forest, Naïve Bayes, KNN, Decision Tree та SVM — для прогнозування наявності серцевої хвороби у людей. Для кожного алгоритму було наведено Confusion Matrix та наступні оцінки: Accuracy, Recall, Precision, F1-Score. Було проаналізовано ці метрики та зроблено висновок, що найкраще для прогнозування застосовувати алгоритм Random Forest, так як він має найкращі показники у трьох з чотирьох метрик.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

В заданому розділі буде проведено оцінювання основних характеристик для майбутнього програмного продукту, що спеціалізується на дослідженні демографічного стану.

Дана реалізація буде сприяти проведенню усіх необхідних досліджень, що дасть змогу якісно дослідити питання не лише в Україні, проте у всьому світі.

Також в даному дослідженні показано різні варіанти реалізації для забезпечення найбільш коректної та оптимальної стратегії вибору, що має вплив на економічні фактори та сумісність з майбутнім програмним продуктом. Для цього застосовувався апарат функціонально-вартісного аналізу.

Функціонально-вартісний аналіз (ФВА) передбачає собою технологію, що дозволяє оцінити реальну вартість продукту або послуги незалежно від організаційної структури компанії. ФВА проводиться з метою виявлення резервів зниження витрат за рахунок ефективніших варіантів виробництва, кращого співвідношення між споживчою вартістю виробу та витратами на його виготовлення. Для проведення аналізу використовується економічна, технічна та конструкторська інформація.

Алгоритм функціонально-вартісного аналізу включає в себе визначення послідовності етапів розробки продукту, визначення повних витрат (річних) та кількості робочих часів, визначення джерел витрат та кінцевий розрахунок вартості програмного продукту.

4.1 Постановка задачі проектування

У роботі застосовується метод ФВА для проведення техніко-економічного аналізу розробки системи прогнозу стійкості фінансових показників. Оскільки рішення стосовно проектування та реалізації компонентів,

що розробляється, впливають на всю систему, кожна окрема підсистема має її задовольняти. Тому фактичний аналіз представляє собою аналіз функцій програмного продукту, призначеного для збору, обробки та проведення аналізу даних по компанії.

Технічні вимоги до програмного продукту є наступні:

- функціонування на персональних комп'ютерах із стандартним набором компонентів;
- зручність та зрозумілість для користувача;
- швидкість обробки даних та доступ до інформації в реальному часі;
- можливість зручного масштабування та обслуговування;
- мінімальні витрати на впровадження програмного продукту.

4.2 Обґрунтування функцій програмного продукту

Головна функція F_0 — розробка програмного продукту, яка дозволяє аналізувати різні характеристики, що безпосередньо впливають на стійкість підприємства. Беручи за основу цю функцію, можна виділити наступні:

F_1 — вибір мови програмування.

F_2 — вибір середовища розробки.

F_3 — коректа обробка даних.

F_4 — графічне виведення.

Кожна з цих функцій має декілька варіантів реалізації:

Функція F_1:

а) Python

б) C++

Функція F_2:

а) Jupyter Notebook;

б) Visual Studio.

Функція F_3:

а) Вбудовані функції.

б) Написання власних.

Функція F_4:

а) Загальні функції середовища програмування.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рис. 4.1).

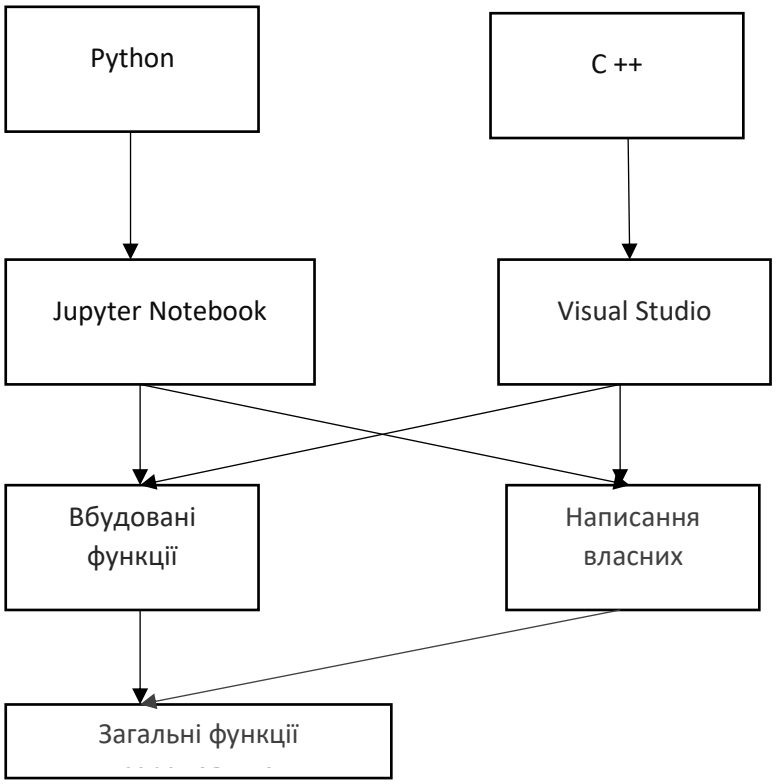


Рисунок 4.1 — Морфологічна карта

Морфологічна карта відображає множину всіх можливих варіантиів основних функцій. Позитивно-негативна матриця показана в таблиці 4.1.

Таблиця 4.1 — Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
F_1	*А*	Порівняно швидке ка програмне розроблення, доступність багатьох бібліотек	Повільна швидкість опрацювання, великий об'єм пам'яті для реалізації
	Б	Достатньо поширена мова програмування, чіткість представлення	Необхідність щоразу контролювати обсяг пам'яті, що збільшує код
F_2	*А*	Візуальна зрозумілість програми	Підтримується лише програмою Python
	Б	Надійність при реалізації	Необхідно окремо встановлювати програмне забезпечення та всі бібліотеки
F_3	*А*	Можна користуватися без попередніх напрацювань	Не завжди ідеально підходять до задачі та майбутнього результату
	Б	Можна створити універсальні функції для програми	Необхідно попередньо реалізувати не лише саму постановку, але й важливі методи, що досить затратно
F_4	*А*	Багато інструментів та можливостей	Час виконання програми

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому, що вони не відповідають поставленим перед програмним продуктом задачам. Ці варіанти відзначені у морфологічній карті.

Функція F_1:

Перевагу віддаємо швидкості вивчення, а також доступності при реалізації. Для спрощення роботи по написанню коду варіант Б має бути відкинутий.

Функція F_2:

Враховуючи той факт, що ми обрали в першій функції варіант А, для другої також обираємо даний варіант — варіант А.

Функція F_3:

Реалізація усіх варіантів є сприйнятливою для програми.

Обираємо єдиний можливий варіант (варіант А).

Таким чином, будемо розглядати такий варіанти реалізації ПП:

F_1а – F_2а – F_3а – F_4а

F_1а – F_2а – F_3б – F_4а

Для оцінювання якості розглянутих функцій обрана система параметрів, описана нижче.

4.3 Обґрунтування системи параметрів програмного продукту

На основі даних, розглянутих вище, визначаються основні параметри вибору, які будуть використані для розрахунку коефіцієнта технічного рівня.

Для того, щоб охарактеризувати програмний продукт, будемо використовувати наступні параметри:

- *X1* — швидкодія мови програмування;

- *X2* — об’єм пам’яті для обчислень та збереження даних;
- *X3* — час навчання даних;
- *X4* — потенційний об’єм програмного коду.

Гірші, середні і кращі значення параметрів вибираються на основі вимог замовника й умов, що характеризують експлуатацію програмного продукту, як показано у таблиці 4.2:

Таблиця 4.2 — Основні параметри програмного продукту

Назва Параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	X1	оп/мс	30	70	100
Об’єм пам’яті	X2	Мб	35	25	10
Час попередньої обробки даних	X3	мс	49	30	14
Потенційний об’єм програмного коду	X4	кількість рядків коду	500	350	290

За даними таблиці 4.3 будуються графічні характеристики параметрів (рис. 4.2 – рис. 4.5):

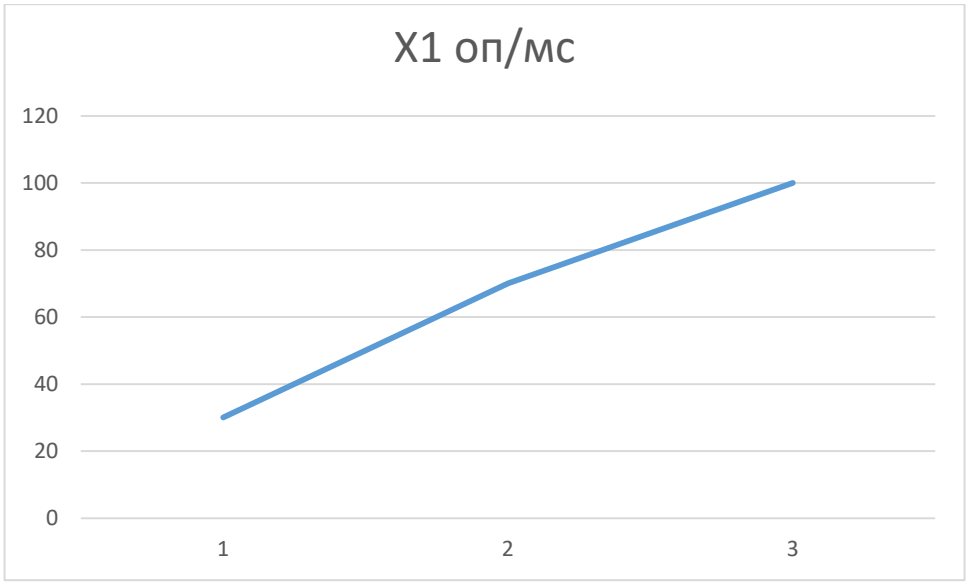


Рисунок 4.2 — X1, швидкодія мови програмування

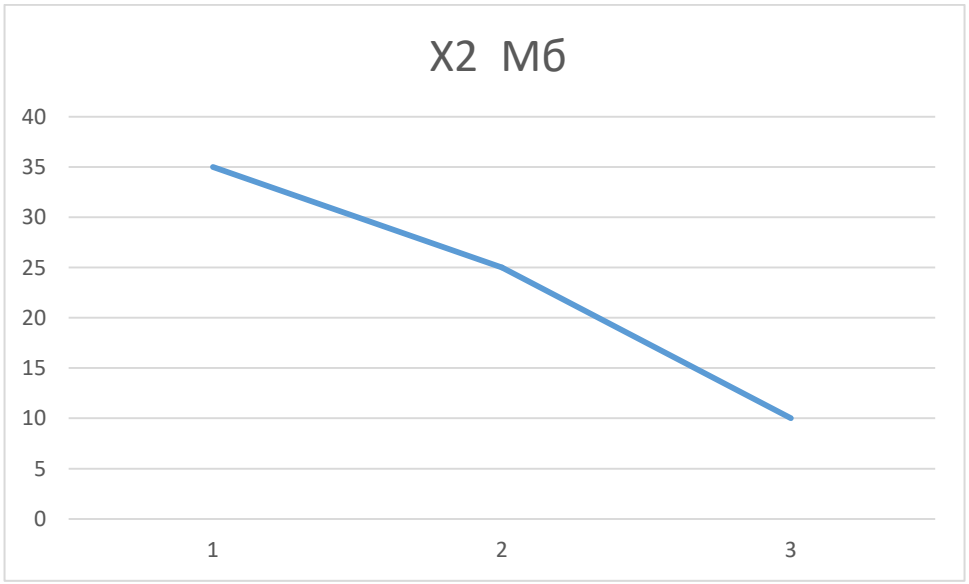


Рисунок 4.3 — X2, об’єм пам’яті

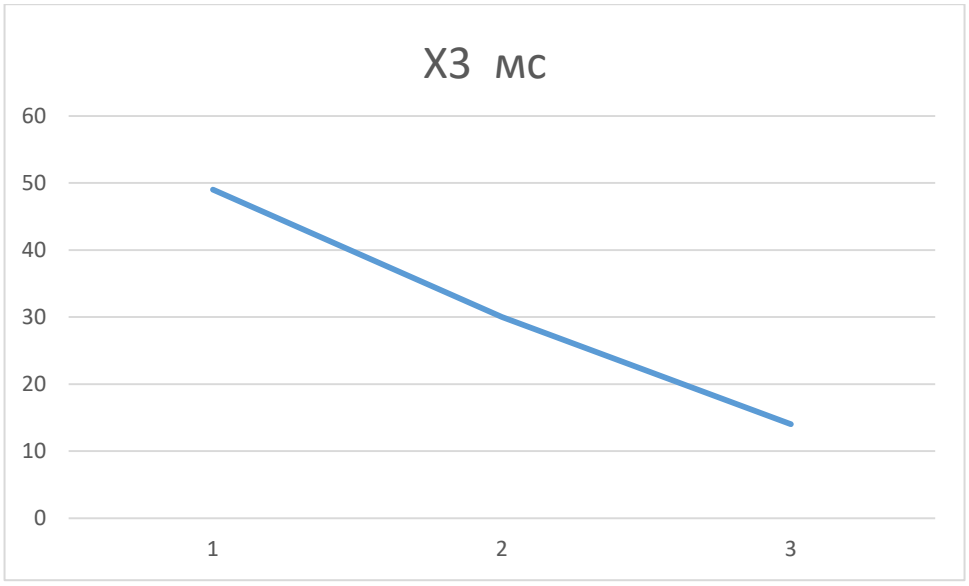


Рисунок 4.4 — X3, час попередньої обробки даних

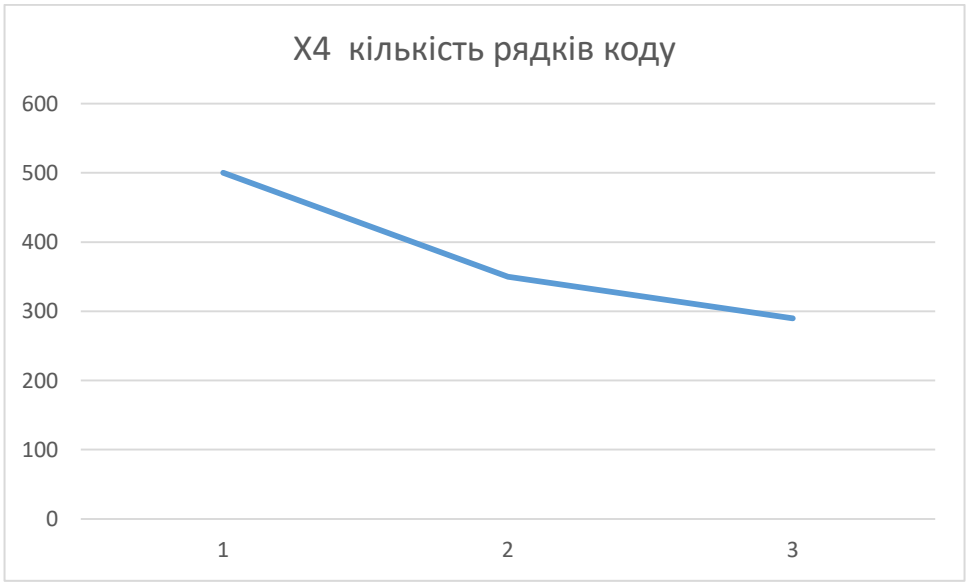


Рисунок 4.5 — X4, потенційний об’єм програмного коду

4.4 Аналіз експертного оцінювання параметрів

Після детального обговорення й аналізу кожний експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі — розробка

програмного продукту, який дає найбільш точні результати при знаходженні параметрів моделей адаптивного прогнозування і обчислення прогнозних значень.

Значимість кожного параметра визначається методом попарного порівняння. Оцінку проводить експертна комісія із 7 людей. Визначення коефіцієнтів значимості передбачає:

- визначення рівня значимості параметра шляхом присвоєння різних рангів;
- перевірку придатності експертних оцінок для подальшого використання;
- визначення оцінки попарного пріоритету параметрів;
- обробку результатів та визначення коефіцієнту значимості.

Результати експертного ранжування наведені у таблиці 4.3.

Таблиця 4.3 — Результати ранжування параметрів

Позначення параметра	Назва параметра	Одиниці виміру	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
			1	2	3	4	5	6	7			
X1	Швидкодія мови програмування	Оп/мс	1	2	1	1	2	2	1	10	-7,5	56,25
X2	Об’єм пам’яті	Mb	2	1	2	3	1	1	2	12	-5,5	30,25
X3	Час попередньої обробки	мс	3	4	4	2	3	4	4	24	6,5	42,25

	даних											
X4	Потенційний об’єм програмного коду	Кількість рядків коду	4	3	3	4	4	3	3	24	6,5	42,25
	Разом		10	10	10	10	10	10	10	70	0	171

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_{j=1}^{N} r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70, \#(4.1)$$

де N — число експертів,

n — кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17{,}5 \ \#(4.2)$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T. \#(4.3)$$

Сума відхилень по всім параметрам повинна дорівнювати 0;

г)загальна сума квадратів відхилення:

$$S = \sum_{i=1}^{N} \Delta_i^2 = 171. \ \#(4.4)$$

Порахуємо коефіцієнт узгодженості:

$$W = \frac{12S}{N^2(n^3 - n)} = \frac{12 \cdot 171}{7^2(4^3 - 4)} = 0{,}872 > W_k = 0{,}67. \#(4.5)$$

Ранжування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю 4.4.

Таблиця 4.4 — Попарне порівняння параметрів.

Параметри	Експерти							Кінцева оцінка	Числове значення
	1	2	3	4	5	6	7		
X1 і X2	<	>	<	<	>	>	<	<	0,5
X1 і X3	<	<	<	<	<	<	<	<	0,5
X1 і X4	<	<	<	<	<	<	<	<	0,5
X2 і X3	<	<	<	>	<	<	<	<	0,5
X2 і X4	<	<	<	<	<	<	<	<	0,5
X3 і X4	<	>	>	>	<	>	>	>	1,5

Числове значення, що визначає ступінь переваги *i*–го параметра над *j*–тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1.5 \text{ при } X_i > X_j \\ 1.0 \text{ при } X_i = X_j \\ 0.5 \text{ при } X_i < X_j \end{cases}. \#(4.6)$$

З отриманих числових оцінок переваги складемо матрицю A=$\| a_{ij} \|$.

Для кожного параметра зробимо розрахунок вагомості $K_{вi}$ за наступними формулами:

$$K_{\text{вi}} = \frac{b_i}{\sum_{i=1}^{n} b_i} \ \#(4.7)$$

$$b_i = \sum_{i=1}^{N} a_{ij} \ \#(4.8)$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятися від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$K_{\text{вi}} = \frac{b_i'}{\sum_{i=1}^{n} b_i'}, \ \#(4.9)$$

$$b_i' = \sum_{i=1}^{N} a_{ij} b_j \ \#(4.10)$$

Як видно з таблиці 4.5, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

Таблиця 4.5 — Розрахунок вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітер.		Друга ітер.		Третя ітер	
	X1	X2	X3	X4	b_i	$K_{\text{вi}}$	b_i^1	$K_{\text{вi}}^1$	b_i^2	$K_{\text{вi}}^2$
X1	1	0,5	0,5	0,5	2,5	0,16	9,25	0,16	34,125	0,17
X2	1,5	1	0,5	0,5	3,5	0,22	12,25	0,21	44,875	0,2
X3	1,5	1,5	1	1,5	5,5	0,34	21,25	0,36	77,875	0,36
X4	1,5	1,5	0,5	1	4,5	0,28	16,25	0,27	59,125	0,27
Всього:					16	1	59	1	216	1

4.5 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів *X2* (Об'єм пам'яті), *X3* (час попередньої обробки даних) та *X4* (потенційний об'єм програмного коду) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра *X1* (швидкість роботи мови програмування) обрано не найгіршим.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується так (таблиця 4.6):

$$K_K(j) = \sum_{i=1}^{n} K_{вi,j} B_{i,j} \text{ , } \#(4.11)$$

де n — кількість параметрів;

$K_{вi}$— коефіцієнт вагомості i–го параметра;

B_i — оцінка i–го параметра в балах.

Таблиця 4.6 — Розрахунок показників рівня якості варіантів реалізації основних функцій ПП

Основні функції	Варіант реалізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	A	X1	100	5	0,17	0,85
F2	A	X2	90	6	0,2	1,2

F3	А	X3	120	9	0,36	3,24
	Б	X2	180	8	0,2	1,6
F4	А	X4	350	7	0,27	1,89

За даними з таблиці 4.6 за формулою:

$$K_K = K_{\text{ТУ}}[F_{1k}] + K_{\text{ТУ}}[F_{2k}] + \ldots + K_{\text{ТУ}}[F_{zk}], \#(4.12)$$

визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 0{,}85 + 1{,}2 + 3{,}24 + 1{,}89 = 7{,}18,$$

$$K_{K2} = 0{,}85 + 1{,}2 + 1{,}6 + 1{,}89 = 5{,}54.$$

Як видно з розрахунків, кращим є перший варіант, для якого коефіцієнт технічного рівня має найбільше значення.

4.6 Економічний аналіз варіантів розробки ПП

Для визначення вартості розробки ПП спочатку проведемо розрахунок трудомісткості.

Всі варіанти включають в себе два окремих завдання:

1. Розробка проекту програмного продукту;

2. Розробка програмної оболонки;

Завдання 1 за ступенем новизни відноситься до групи А, завдання 2 — до групи Б. За складністю алгоритми, які використовуються в завданні 1 належать до групи 1; а в завданні 2 — до групи 3.

Для реалізації завдання 1 використовується довідкова інформація, а завдання 2 використовує інформацію у вигляді даних.

Проведемо розрахунок норм часу на розробку та програмування для кожного з завдань.

Загальна трудомісткість обчислюється як:

$$T_O = T_P \cdot K_П \cdot K_{СК} \cdot K_М \cdot K_{СТ} \cdot K_{СТ.М}, \#(4.13)$$

де T_P — трудомісткість розробки ПП;

$K_П$ — поправочний коефіцієнт;

$K_{СК}$ — коефіцієнт на складність вхідної інформації;

$K_М$ — коефіцієнт рівня мови програмування;

$K_{СТ}$ — коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ — коефіцієнт стандартного математичного забезпечення

Для першого завдання, виходячи із норм часу для завдань розрахункового характеру степеню новизни А та групи складності алгоритму 1, трудомісткість дорівнює: T_P =35 людино-днів. Поправочний коефіцієнт, який враховує вид нормативно-довідкової інформації для першого завдання: $K_П$ = 1.8. Поправочний коефіцієнт, який враховує складність контролю вхідної та вихідної інформації для всіх семи завдань рівний 1: $K_{СК}$ = 1. Оскільки при розробці першого завдання використовуються стандартні модулі, врахуємо це за допомогою коефіцієнта $K_{СТ}$ = 0.8. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 35 \cdot 1.8 \cdot 0.8 = 50,4 \text{ людино-днів.}$$

Проведемо аналогічні розрахунки для подальших завдань.

Для другого завдання (використовується алгоритм третьої групи складності, степінь новизни Б), тобто $T_Р$ =14 людино-днів, $К_П$ =0.8,$К_{СК}$ = 1,$К_{СТ}$ =0.7:

$$T_2 = 14 \cdot 0.8 \cdot 0.7 = 7.84 \text{ людино-днів.}$$

Складаємо трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їх трудомісткість:

$$T_I = (50.4 + 7.84 + 4.8 + 7.84) \cdot 8 = 567{,}04 \text{ людино-годин.}$$

$$T_{II} = (50.4 + 7.84 + 6.91 + 7.84) \cdot 8 = 583{,}92 \text{ людино-годин.}$$

Найбільш високу трудомісткість має варіант II.

В розробці беруть участь два програмісти з окладом 15000 грн., один аналітик в області даних з окладом 14000. Визначимо середню зарплату за годину за формулою:

$$С_Ч = \frac{М}{T_m \cdot t} \text{ грн., } \#(4.14)$$

де М — місячний оклад працівників;

T_m — кількість робочих днів тиждень;

t — кількість робочих годин в день.

$$С_Ч = \frac{15000 + 15000 + 14000}{4 \cdot 21 \cdot 8} = 65{,}48 \text{ грн.} \#(4.15)$$

Тоді, розрахуємо заробітну плату за формулою:

$$С_{ЗП} = С_{ч} \cdot Т_{i} \cdot К_{Д}, \#(4.16)$$

де $С_{ч}$ — величина погодинної оплати праці програміста;

$Т_{i}$ — трудомісткість відповідного завдання;

$К_{Д}$ — норматив, який враховує додаткову заробітну плату.

Зарплата розробників за варіантами становить:

I. $С_{ЗП} = 65.48 \cdot 567.04 \cdot 1.2 = 44555,74$ грн.

II. $С_{ЗП} = 65.48 \cdot 583.92 \cdot 1.2 = 45882,09$ грн.

Відрахування на єдиний соціальний внесок становить 22%:

I. $С_{ВІД} = С_{ЗП} \cdot 0.22 = 44555,74 \cdot 0.22 = 9802,27$ грн.

II. $С_{ВІД} = С_{ЗП} \cdot 0.22 = 45882,09 \cdot 0.22 = 10094,06$ грн.

Тепер визначимо витрати на оплату однієї машино-години. ($С_{М}$)

Так як одна ЕОМ обслуговує одного програміста з окладом 15000 грн., з коефіцієнтом зайнятості 0,2 то для однієї машини отримаємо:

$$С_{Г} = 12 \cdot М \cdot К_{З} = 12 \cdot 15000 \cdot 0,2 = 36000 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$С_{ЗП} = С_Г \cdot (1 + К_З) = 36000 \cdot (1 + 0.2) = 43200 \text{ грн.}$$

Відрахування на соціальний внесок:

$$С_{ВІД} = С_{ЗП} \cdot 0.22 = 43200 \cdot 0,22 = 9504 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при амортизації 25% та вартості ЕОМ – 14000 грн.

$$С_А = К_{ТМ} \cdot К_А \cdot Ц_{ПР} = 1.15 \cdot 0.25 \cdot 14000 = 4025 \text{ грн.},$$

де $К_{ТМ}$ — коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

$К_А$ — річна норма амортизації;

$Ц_{ПР}$ — договірна ціна приладу.

Витрати на ремонт та профілактику розраховуємо як:

$$С_Р = К_{ТМ} \cdot Ц_{ПР} \cdot К_Р = 1.15 \cdot 14000 \cdot 0.05 = 805 \text{ грн.},$$

де $К_Р$ — відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік розраховуємо за формулою:

$$Т_{ЕФ} = (Д_К - Д_В - Д_С - Д_Р) \cdot t_З \cdot К_В = (365 - 104 - 12 - 16) \cdot 8 \cdot 0.5 =$$

$$= 932 \text{ години},$$

де $Д_К$ — календарна кількість днів у році;

$Д_В$, $Д_С$ — відповідно кількість вихідних та святкових днів;

$Д_Р$ — кількість днів планових ремонтів устаткування;

t — кількість робочих годин в день;

$К_В$ — коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії розраховуємо за формулою:

$$С_{ЕЛ} = Т_{ЕФ} \cdot N_С \cdot К_З \cdot Ц_{ЕН} = 932 \cdot 0,25 \cdot 0,87 \cdot 3,52 = 713,54 \text{ грн.},$$

де $N_С$ — середньо-споживча потужність приладу;

$К_З$ — коефіцієнтом зайнятості приладу;

$Ц_{ЕН}$ — тариф за 1 КВт-годин електроенергії.

Накладні витрати розраховуємо за формулою:

$$С_Н = Ц_{ПР} \cdot 0.67 = 14000 \cdot 0,67 = 9380 \text{ грн.}$$

Тоді, річні експлуатаційні витрати будуть:

$$С_{ЕКС} = С_{ЗП} + С_{ВІД} + С_А + С_Р + С_{ЕЛ} + С_Н, \#(4.17)$$

$$С_{ЕКС} = 43200 + 9504 + 4025 + 805 + 713,54 + 9380 = 67627,54 \text{ грн.}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$С_{М\text{-}Г} = С_{ЕКС} / Т_{ЕФ} = 67627,54 / 932 = 72,55 \text{ грн/год.}$$

Оскільки в даному випадку всі роботи, які пов‘язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, складає:

$$C_М = C_{М-Г} \cdot Т, \#(4.18)$$

I. $C_М = 72,55 \cdot 567,04 = 41138,75$ грн.

II. $C_М = 72,55 \cdot 583,92 = 42363,4$ грн.

Накладні витрати складають 67% від заробітної плати:

$$C_Н = C_{ЗП} \cdot 0,67, \#(4.19)$$

I. $C_Н = 44555,74 \cdot 0,67 = 29852,35$ грн.

II. $C_Н = 45882,09 \cdot 0,67 = 30741$ грн.

Отже, вартість розробки ПП за варіантами становить:

$$C_{ПП} = C_{ЗП} + C_{ВІД} + C_М + C_Н, \#(4.20)$$

I. $C_{ПП} = 44555,74 + 9802,27 + 41138,75 + 29852,35 = 125349,11$ грн.

II. $C_{ПП} = 45882,09 + 10094,06 + 42363,4 + 30741 = 129080,55$ грн.

4.7 Вибір кращого варіанту ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$К_{ТЕР j} = К_{Кj} / С_{Фj}, \#(4.21)$$

$$К_{ТЕР1} = 7,18 / 125349,11 = 5,73 \cdot 10^{-5},$$

$$К_{ТЕР2} = 5,54 / 129080,55 = 4,29 \cdot 10^{-5}.$$

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $К_{ТЕР1} = 5,73 \cdot 10^{-5}$.

Після виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, можна зробити висновок, що з альтернатив, що залишились після першого відбору двох варіантів виконання програмного комплексу оптимальним є перший варіант реалізації програмного продукту. У нього виявився найкращий показник техніко-економічного рівня якості $К_{ТЕР} = 5,73 \cdot 10^{-5}$.

Цей варіант реалізації програмного продукту має такі параметри:

- мова програмування – Python;
- Використання моделей з значною кількістю характеристик;
- Реалізація заданих методів за допомогою вбудованих функцій;
- Використання стандартного інтерфейсу візуалізації, швидкість розробки.

Даний варіант виконання програмного комплексу дає користувачу зручний інтерфейс, швидку реалізацію програми та доступний функціонал для роботи.

4.8 Висновки до розділу 4

В даній частині було проведено повний функціонально-вартісний аналіз програмного продукту. Також було знайдено оцінку основних функцій програмного продукту. Роботу з дослідження можна уявно розподілити на дві частини.

В першій з них було проведено експертне дослідження усіх необхідних характеристик, що відповідають за реалізацію програмного продукту, а саме, визначено основні функції, експертні оцінки та важливі параметри. Також було знайдено коефіцієнт технічного рівня, що відповідає за найбільш якісний спосіб реалізації ПП.

В другій частині було проведено математичний аналіз, а також знаходження відповідних параметрів відповідно до попередньо заданих формул. Також саме дана частина відповідає за вибір найбільш економічно доцільних варіантів при реалізації програмного продукту.

В результаті виконання функціонально-вартісного аналізу програмного комплексу що розроблюється, було визначено та проведено оцінку основних функцій програмного продукту, а також знайдено параметри, які характеризують програмний продукт.

Проведено економічний аналіз варіантів розробки – трудомісткість, витрати на заробітну плату та інші витрати.

На основі аналізу вибрано варіант реалізації програмного продукту.

ВИСНОВКИ

У результаті виконання бакалаврської роботи було розглянуто поняття, пов'язані з математичним моделюванням в медицині. Математичне моделювання є важливим інструментом для дослідження характеристик та ефективності лікарських засобів, процесів лікування та попередження захворювань. Воно також допомагає приймати рішення щодо оптимальної стратегії лікування.

У другому розділі проведено аналіз датасету, включаючи кореляцію параметрів та розподіл їх по хворим та здоровим людям. Було розглянуто значення кожного параметра, його важливість та вплив на наявність серцевого захворювання.

У третьому розділі було застосовано алгоритми машинного навчання, такі як Logistic Regression, Random Forest, Naïve Bayes, KNN, Decision Tree та SVM, для прогнозування наявності серцевої хвороби. Для кожного алгоритму було наведено Confusion Matrix та оцінки Accuracy, Recall, Precision та F1-Score. Аналізуючи ці метрики, було встановлено, що найкращі результати показує алгоритм Random Forest.

Моделювання серцевих процесів є важливим аспектом фізіології та патології серця. Це допомагає розуміти роботу серця та вплив різних факторів на його функцію. Моделювання тиску також має велике значення вивченні гіпертонії та ризику серцевих захворювань.

Застосування алгоритмів машинного навчання у медицині має велике значення, оскільки вони дозволяють аналізувати великі обсяги даних, виявляти залежності та прогнозувати ризики та наслідки захворювань. Вони знаходять своє застосування в діагностиці, лікуванні, профілактиці та наукових дослідженнях. Моделювання серцевих хвороб допомагає прогнозувати ризик

захворювання та вибирати оптимальні методи профілактики та лікування, що сприяє покращенню здоров'я та зменшенню ризику смертності від серцево-судинних захворювань.

ПЕРЕЛІК ДЖЕРЕЛ

1. Внутрішня історія. Серце — найважливіший орган нашого організму. Йоханнес Хінріх фон Борстель. — Л.А., — 1991. — 343 с.
2. Modelling Congenital Heart Disease — Gianfranco Butera. — 5th ed. — Chicago, IL: Chaos Solitons Fract, — 2020. — 138 p.
3. Medical Modelling The Application of Advanced Design and Rapid Prototyping Techniques in Medicine Book, — Second Edition. — 2015. — 2323 p.
4. Mathematical Modelling in Medicine Ottesen, J.T.,Danielsen, — M. — 2019. — 244 p.
5. Farre, Albert; Rapley, Tim (2017-11-18). "The New Old (and Old New) Medical Model: Four Decades Navigating the Biomedical and Psychosocial Understandings of Health and Illness" — 2021. — 2939 p.
6. Atlas of Human Anatomy / Атлас анатомії людини Френк Неттер — 1987. — 288 p.
7. Mixture Modelling for Medical and Health Sciences Shu Kay Ng, Liming Xiang, Kelvin Kai Wing Yau
8. System Dynamics for Engineering Students. Elsevier; 2018. [Електронний ресурс] — Режим доступу: doi:https://doi.org/10.1016/C2011-0-05346-2.
9. Kitson A, Brook A, Harvey G, Jordan Z, Marshall R, O'Shea R, et al. Using Complexity and Network Concepts to Inform Healthcare Knowledge Translation. Int J Heal Policy Manag. 2017;7:231–43. [Електронний ресурс] — Режим доступу: — https://doi.org/10.15171/ijhpm.2017.79.
10. Forrester J. Industrial dynamics. Cambridge: — The MIT Press; — 1961. — 383 p.

11. Sterman J. Business dynamics : Systems thinking and modeling for a complex world. — The MIT Press; — 1991. — 390 p.
12. Homer JB, Hirsch GB. System dynamics modeling for public health: background and opportunities. — Am J Public Health. — 2006. — 4458 p.
13. Vennix J. Group model building: Facilitating team learning using system dynamics. — Chichester: John Wiley & Sons. — 1996. — 232 p.
14. Systems Science Methods in Public Health Douglas A. Luke1 and Katherine A. Stamatakis — Chichester: John Wiley & Sons. — 1996. — 2342 p.
15. Borshchev A, Filippov A. From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. Proc. — The 22nd International Conference of the System Dynamics Society; Oxford, England. — 2004. — 223 p.
16. Mitchell M. Complexity : a guided tour. — New York: Oxford University Press; — 2009. — 2332 p.
17. Reynolds CW. Flocks, herds and schools: A distributed behavioral model. — ACM SIGGRAPH Computer Graphics. — 1987. —2125 p.
18. Macy MW, Willer R. From factors to actors: Computational sociology and agent-based modeling. — Annual Review of Sociology. — 2002. — 2143 p.
19. Axelrod R. Agent-based modeling as a bridge between disciplines. In: Tesfatsion L, Judd KL, editors. Handbook of Computational Economics. — Vol. 2. — Elsevier. — 2006. — 2828 p.
20. Epstein JM. Agent-based computational models and generative social science. — Complexity. — 1999. —:4157 p.
21. Railsback SF, Lytinen SL, Jackson SK. Agent-based simulation platforms: Review and development recommendations. — Simulation. — 2006. — 8209 p.

22.Freeman LC. The development of social network analysis : a study in the sociology of science. — Vancouver, BC: Empirical Press; — 2004. — 383 p.
23.Moreno JL. Sociometry in relation to other social sciences. — Sociometry. — 1937. — 2069 p.
24.Buchanan M. Nexus : Small worlds and the groundbreaking science of networks. — New York: W.W. Norton. — 2002. — 3030 p.
25.Wasserman S, Faust K. Social network analysis : Methods and applications. — New York: Cambridge University Press; — 1994. — 833 p.
26.. Lankhaar J-W, Rövekamp FA, Steendijk P, Faes TJC, Westerhof BE, Kind T, et al. Modeling the instantaneous pressure-volume relation of the left ventricle: a comparison of six models. — Ann Biomed Eng. — 2009. — 1710 p.
27.Faes TJ, Kerkhof PL. The Volume Regulation Graph versus the Ejection Fraction as Metrics of Left Ventricular Performance in Heart Failure with and without a Preserved Ejection Fraction: A Mathematical Model Study. — Clin Med Insights Cardiol. — 2015. — 391 p.
28.Song Z, Gu K, Gao B, Wan F, Chang Y, Zeng Y. Hemodynamic effects of various support modes of continuous flow LVADs on the cardiovascular system: A numerical study. — Med Sci Monit Int Med J Exp Clin Res. — 2014. — 890 p.
29.Ellwein LM, Pope SR, Xie A, Batzel JJ, Kelley CT, Olufsen MS. Patient-specific modeling of cardiovascular and respiratory dynamics during hypercapnia. — Math Biosci. — 2013. — 338 p.
30.van de Vosse FN, Stergiopulos N. Pulse Wave Propagation in the Arterial Tree. — Annu Rev Fluid Mech. — 2011. — 4467 p.

31. Cox LGE, Loerakker S, Rutten MCM, de Mol BAJM, van de Vosse FN. A mathematical model to evaluate control strategies for mechanical circulatory support. — Artif Organs. — 2009. — 3393 p.
32. Bovendeerd PHM, Borsje P, Arts T, van De Vosse FN. Dependence of Intramyocardial Pressure and Coronary Flow on Ventricular Loading and Contractility: A Model Study. — Ann Biomed Eng. — 2006. —1845 p.
33. Jongen GJLM, van der Hout-van der Jagt MB, Oei SG, van de Vosse FN, Bovendeerd PHM. Simulation of fetal heart rate variability with a mathematical model. — Med Eng Phys. — 2017. — 564 p.
34. Bozkurt S, Bozkurt S. In-silico evaluation of left ventricular unloading under varying speed continuous flow left ventricular assist device support. —Biocybern Biomed Eng. — 2017. — 3737 p.
35. Bhattacharya-Ghosh B, Bozkurt S, Rutten MCM, van de Vosse FN, Díaz-Zuccarini V. An in silico case study of idiopathic dilated cardiomyopathy via a multi-scale model of the cardiovascular system. — Comput Biol Med. —2014. — 383 p..
36. Bozkurt S. In-silico modeling of left ventricle to simulate dilated cardiomyopathy and cf-lvad support. — J Mech Med Biol. — 2017. — 173 p.
37. Sheffer L, Santamore WP, Barnea O. Cardiovascular simulation toolbox. — Cardiovasc Eng Dordr Neth. — 2007. — 788 p.
38. Shimizu S, Une D, Kawada T, Hayama Y, Kamiya A, Shishido T, et al. Lumped parameter model for hemodynamic simulation of congenital heart diseases. — J Physiol Sci JPS. — 2018. — 111 p.
39. Heart Development and Regeneration Nadia Rosenthal and Richard P. — Harvey. — 2021. — 252 p.
40. Biology and Anatomy & Physiology Helps: The Heart Carolyn — Miller — . 2015. — 238 p

41. Cardiac and Vascular Biology Markus Hecker, Dirk J. — Duncker — 2021. — 3833 p.
42. Ukawa T, et al. Novel non-invasive method of measurement of endothelial function: Enclosed-zone flow-mediated dilatation (ezFMD) — Med Biol Eng Comput. — 2012. — 1239 p.
43. Chandrasekhar A, et al. Smartphone-based blood pressure monitoring via the oscillometric finger-pressing method.
44. Fuke S, Suzuki T, Nakayama K, Tanaka H, Minami S. Blood pressure estimation from pulse wave velocity measured on the chest. — Conf Proc IEEE Eng Med Biol Soc. — 2013. — 6110 p.
45. Malik, M., Hnatkova,K., Camm,A.J.:Practicality of postinfarction risk assessment based on time-domain measurement of heart rate variability. M. Malik, AJ. Camm, (Eds.), Heart Rate Variability.Armonk. — NY.F. — 2021. — 323 p.
46. E. C. Zeeman, “Differential Equations for the Heartbeat and Nerve Impulse,” Towards a Theoretical Biology, — Vol. 4, — 1972, — 434 p.
47. H.D. Young and R.A. Freedman. Sears and Zemansky’s University Physics: with Modern Physics. Pearson/Addison-Wesley, — 11th, international edition, — 2004. — 313 p.
48. J. Keener and J. Sneyd. Mathematical Physiology, volume System Physiology. —Springer. — second edition, — 2009. — 323 p.
49. Frank O Die Grundform des arteriellen — Pulses. Z Biol. — 2021. — 3783 p.
50. M . Hlaváč, J. Holčík, „WINDKESSEL M ODEL ANALYSIS IN M ATLAB“, in Proc 2004 Student Electrical Engineering, Information and Communication Technologies, Brno 2004.

[Електронне джерело]. — http://www.feec.vutbr.cz/EEICT/2004/sbornik/03-Doktorske_projekty/01-Elektronika/09lahvo.pdf

51. Peña Pérez, Nuria. "Windkessel modeling of the human arterial system." — Bachelor's thesis, — 2016. — 372 p.
52. Burattini, Roberto, and Paola Oriana Di Salvia. "Development of systemic arterial mechanical properties from infancy to adulthood interpreted by four-element windkessel models." Journal of Applied Physiology 103, no. 1 (2007): 66-79.
53. Stergiopulos, Nikos, Berend E. Westerhof, and Nico Westerhof. "Total arterial inertance as the fourth element of the windkessel model." American Journal of Physiology-Heart and Circulatory Physiology 276, no. 1 (1999): H81-H88
54. Taib, Ishkrizat. "Improvement of Haemodynamic Stent Strut Configuration for Patent Ductus Arteriosus Through Computational Modelling." PhD diss., Universiti Teknologi Malaysia, 2016.
55. Серцеві хвороби [Електронний ресурс] — Режим доступу: https://www.who.int/health-topics/cardiovascular-diseases
56. Серцеві хвороби [Електронний ресурс] — Режим доступу: https://atm.amegroups.com/article/view/10170/html
57. Perspective on the Application of Machine Learning Algorithms for Flow Parameter Estimation in Recycled Concrete Aggregate. — Dzięcioł J, Sas W.
58. M.W. Kenyhercz, N.V. Passalacqua, in Biological Distance Analysis, — 2016
59. Серцеві проблеми [Електронний ресурс] — Режим доступу: https://www-cem-org-

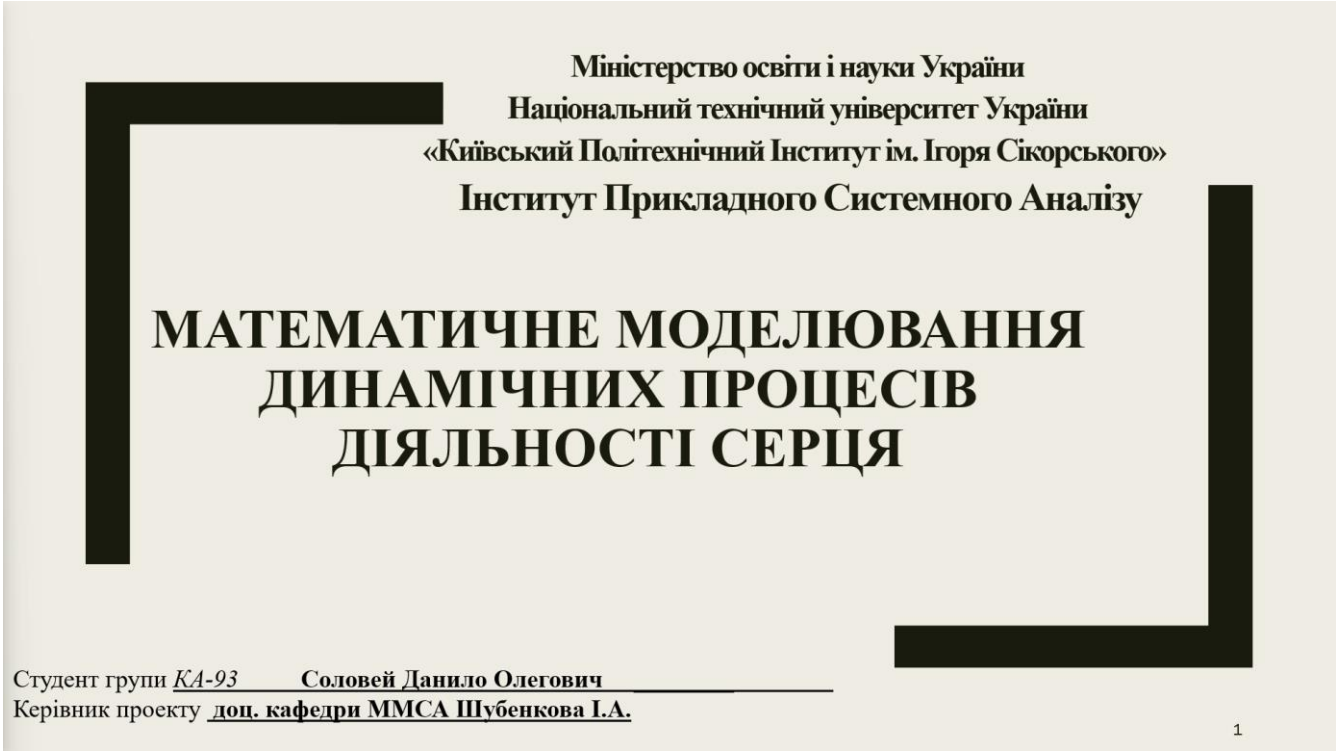
cn/rhtml/20210421/index.htm?_x_tr_sch=http&_x_tr_sl=zh-CN&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=sc

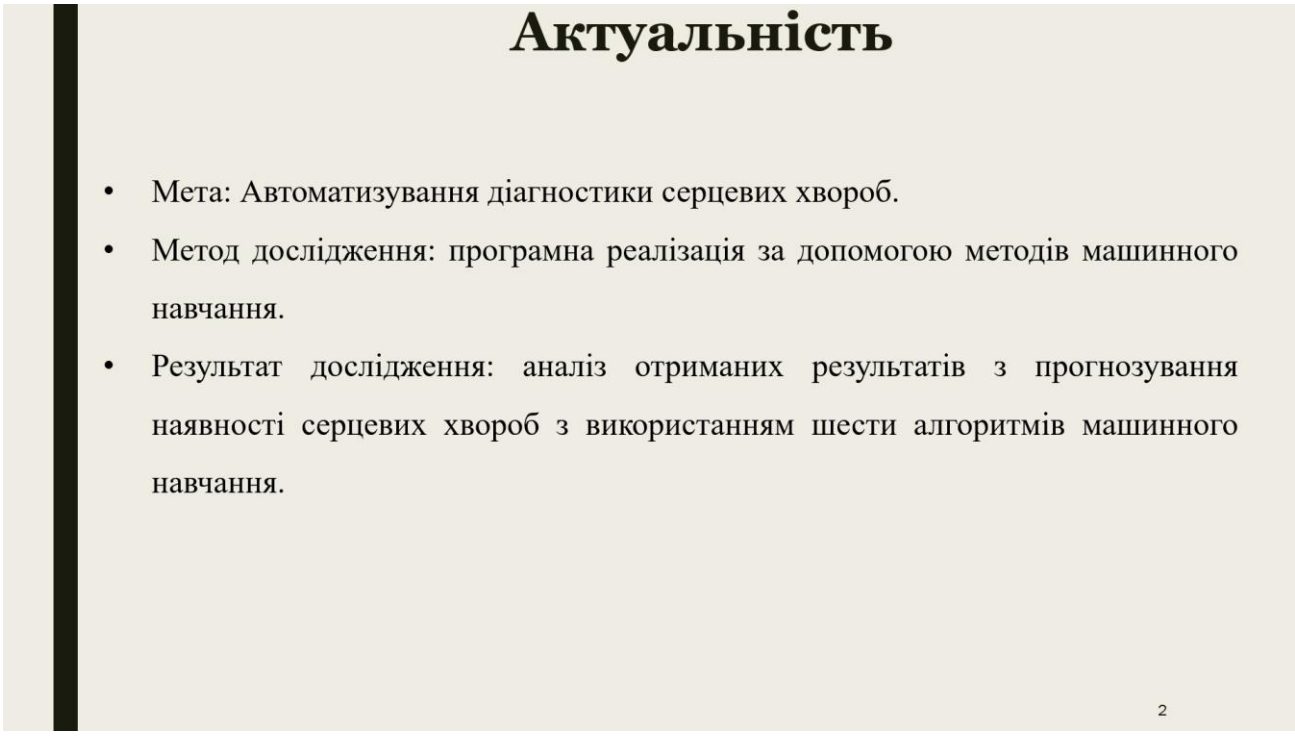
60. Серце [Електронний ресурс] — Режим доступу: http://www.cqvip.com/qk/96363x/201804/675908405.html
61. Haqqani HM, Chan KH, Kumar S, Denniss AR, Gregory AT. The contemporary era of sudden cardiac death and ventricular arrhythmias: basic concepts, recent developments and future directions. — Heart Lung Circ. 2019. — 281 p.
62. Breiman, L., Friedman, J. H., Olsen, R. A., and Stone, C. J., Classification and Regression Trees, — Wadsworth, USA, — 1984. — 383 p.
63. Murthy, K. V. S., On Growing Better Decision Trees from Data, PhD dissertation, — Johns Hopkins University, Baltimore, — MD, — 1997. — 322p.
64. Joachims, Thorsten. Transductive Inference for Text Classification using Support Vector Machines (PDF). Proceedings of the 1999 International — Conference on Machine Learning. —1999. —259.
65. Aizerman, Mark A.; Braverman, Emmanuel M. & Rozonoer, Lev I. (1964). "Theoretical foundations of the potential function method in pattern recognition learning". — Automation and Remote Control. — 2032. — 257 p.
66. Rosasco, Lorenzo; De Vito, Ernesto; Caponnetto, Andrea; Piana, Michele; Verri, Alessandro (2004-05-01). "Are Loss Functions All the Same?". Neural Computation. 16 (5): 1063–1076.
67. Mohan, S., Thirumalai, C., & Srivastava, G. (2019). Effective heart disease prediction using hybrid machine learning techniques. — IEEE Access, 7, — 2019. — 8154 p.
68. Bui, A. L., Horwich, T. B. & Fonarow, G. C. Epidemiology and risk profile of heart failure. — Nat. Rev. Cardiol. 8, — 2012. — 212 p.

69. Polat, K. & Güneş, S. Artificial immune recognition system with fuzzy resource allocation mechanism classifier, principal component analysis, and FFT method based new hybrid automated identification system for classification of EEG signals. — Expert Syst. Appl. — 34, 2039–2048 (2010).
70. Durairaj, M. & Ramasamy, N. A comparison of the perceptive approaches for preprocessing the data set for predicting fertility success rate. Int. J. Control — Theory Appl. —2016. — 292 p.
71. Das, R., Turkoglu, I. & Sengur, A. Effective diagnosis of heart disease through neural networks ensembles. — Expert Syst. Appl. — 2019. — 2322 p.
72. Allen, L. A. et al. Decision making in advanced heart failure: A scientific statement from the American Heart Association. — Circulation — 2015. — 1253 p.
73. Yang, H. & Garibaldi, J. M. A hybrid model for automatic identification of risk factors for heart disease. — J. Biomed. Inform. — 2019. — 583 p.
74. Alizadehsani, R., Hosseini, M. J., Sani, Z. A., Ghandeharioun, A. & Boghrati, R. In 2012 IEEE 12th International Conference on Data Mining Workshops. 9–16 (IEEE, New York).
75. Arabasadi, Z., Alizadehsani, R., Roshanzamir, M., Moosaei, H. & Yarifard, A. A. Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm. — Comput. Methods Programs Biomed. — 2021. — 3226 p.
76. Samuel, O. W., Asogbon, G. M., Sangaiah, A. K., Fang, P. & Li, G. An integrated decision support system based on ANN and Fuzzy_AHP for heart failure risk prediction. — Expert Syst. Appl. — 2017. — 3232 p.

77. Patil, S. B. & Kumaraswamy, Y. Intelligent and effective heart attack prediction system using data mining and artificial neural network. — Eur. J. Sci. Res. — 2018. — 3239 p.
78. Vanisree, K. & Singaraju, J. Decision support system for congenital heart disease diagnosis based on signs and symptoms using neural networks. — Int. J. Comput. Appl. — 2016. — 1923 p.
79. B. Edmonds. In Proceedings of AISB Symposium on Socially Inspired Computing — Hatfield. — 2005. — 292 p.

ДОДАТОК А ІЛЮСТРАТИВНИЙ МАТЕРІАЛ





Опис використаних метрик

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Дані

- Датасет для діагностування наявності серцевої хвороби.
- Розширений датасет містить дані о 303 пацієнтах, кожний з яких описується чотирнадцятьма атрибутами.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

- 80% даних було використано для навчання алгоритмів, а 20% для тестування.

Результати роботи

Метод	Accuracy	Recall	Precision	F1-Score
Logistic Regression	0.8525	0.8824	0.8571	0.8696
Random Forest	0.9016	0.9412	0.8889	0.9143
Naïve Bayes	0.8525	0.9118	0.8378	0.8732
KNN	0.6393	0.5882	0.7143	0.6452
Decision Tree	0.8197	0.8235	0.8485	0.8358
SVM	0.8197	0.8824	0.8108	0.8451

Висновки

- Математичне моделювання є важливим інструментом для дослідження характеристик та ефективності лікарських засобів, процесів лікування та попередження захворювань.
- було застосовано алгоритми машинного навчання, такі як Logistic Regression, Random Forest, Naïve Bayes, KNN, Decision Tree та SVM, для прогнозування наявності серцевої хвороби. Для кожного алгоритму було наведено Confusion Matrix та оцінки Accuracy, Recall, Precision та F1-Score. Аналізуючи ці метрики, було встановлено, що найкращі результати показує алгоритм Random Forest.
- Моделювання серцевих процесів є важливим аспектом фізіології та патології серця. Це допомагає розуміти роботу серця та вплив різних факторів на його функцію.

Дякую за увагу!

ДОДАТОК Б ЛІСТИНГ ПРОГРАМИ

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import sklearn

%matplotlib inline


import os


import warnings

warnings.filterwarnings('ignore')


data = pd.read_csv("heart.csv")


data.shape #щоб оцінити розмір наших тестових даних


data.head()  #виведемо перші п'ять


data.describe() #генерує описову статистику


count - кількість не нульових рядків у стовпці<br>

mean - середнє<br>

std - відхилення (дисперсія)<br>

min - мінімальне значення<br>

25% (1-й квартиль) відповідає значенню, яке ділить дані на 4 рівні частини, перша з яких містить 25% від всіх значень.<br>

50% (медіана) - це значення, яке розділяє дані на дві рівні частини, перша з яких містить 50% від всіх значень.<br>
```

75% (3-й квартиль) - це значення, яке розділяє дані на 4 рівні частини, перша з яких містить 75% від всіх значень.

max - максимальне значення

data.info() #що є в нашому датасеті

0. age - вік

1. sex - стать (1 = male, 0 = female)

angina - стенокардія - це тип болю в грудях, викликаний зменшенням припливу крові до серця. Стенокардія є симптомом ішемічної хвороби серця. Стенокардійний біль часто описується як здавлювання, тиск, тяжкість, стиснення або біль у грудях. Може здаватися, що лежить на грудях важка вага. Стенокардія може бути новим болем, який потребує перевірки лікарем, або повторюваним болем, який проходить під час лікування.

2. cp - The chest pain experienced (біль у грудях) (Value 1 - typical angina, Value 2 - atypical angina, Value 3 - non-anginal pain, Value 4 - asymptomatic)

3. trestbps - The person's resting blood pressure (mm Hg on admission to the hospital) (тиск у спокійному стані)

4. chol: The person's cholesterol measurement in mg/dl (виміри холестерину)

(Рівень цукру в крові натще (глюкоза))

5. fbs - The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)

(Електрокардіографічне вимірювання у стану спокою)

6. restecg - Resting electrocardiographic measurement (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)

7. thalach - The person's maximum heart rate achieved

Чи присутня стенокардія при фізичному навантаженні

8. exang - Exercise induced angina (1 = yes; 0 = no)

9. oldpeak - ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot. See more here)

ST депресія - це зниження сегмента ST на електрокардіограмі після фізичного навантаження порівняно з відпочинком

нахил: нахил вершини вправа ST відрізок (Значення 1: похилий, Значення 2: плоский, Значення 3: похилий)

10. slope - the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: flat, Value 3: downsloping)

ST сегмент в ЕКГ (електрокардіограма) - це відрізок на графіку, який відображає відносну ізоляцію між правим жовчним протоком та левою кімнатою серця. Вона відображає взаємодію між електролітами в крові та електродією в серцевих стінах. Величина ST сегменту може відображати інфаркт міокарду, наявність запалень у серцевих стінах та інші патології.

11. ca - The number of major vessels (0-3)

12. thal - A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)

13. target - Heart disease (0 = no, 1 = yes)

Фактори ризику серцевих захворювань наступні: високий рівень холестерину, високий кров'яний тиск, діабет, вага, сімейна історія та куріння.

Згідно з іншим джерелом, основними факторами, які неможливо змінити, є: збільшення віку, чоловічої статі та спадковості.

Зверніть увагу, що таласемія, одна зі змінних у цьому наборі даних, є спадковістю.

Основними факторами, які можуть бути змінені, є: куріння, високий рівень холестерину, високий кров'яний тиск, гіподинамія, надмірна вага та діабет.

Інші фактори включають стрес, алкоголь і погане харчування.

```
data.sample(5)
```

```
data.isnull().sum()
```

```
data.isnull().sum().sum()
```

пропущених значень немає

```
print(data.corr()["target"].abs().sort_values(ascending=False))
```

```
target_temp = data.target.value_counts()
```

Ця строка робить визначення кореляційної залежності між всіма признаками у датафреймі "data" та цільовою змінною "target". Кореляція визначається як абсолютне значення коефіцієнта кореляції Пірсона. Результати сортуються за зростанням їх величини.

Як можна побачити, найменше корелюють chol та fbs

```
# Exploratory Data Analysis (EDA)
```

```
y = data["target"]
```

В колонці target у нас показано, чи хвора ця людина, чи вона здорова. Тут ми подивимося, скільки у нас хворих та здорових людей

```
pd.crosstab(data.target, data.target).plot(kind="bar",figsize=(6,6), color=['blue', 'red'])

#pd.crosstab(data.target,data.target).plot(kind="bar",figsize=(6,6), color=['blue', 'red'])

#plt.figure(figsize=(6, 6))

#sns.countplot(x='target', data=data)

plt.legend(['Без серцевої хвороби', 'З серцевою хворобою'])

plt.title('Кількість людей у датасеті з серцевими хворобами і без')

plt.ylabel('Кількість')
```

```
plt.xlabel('')
plt.xticks(rotation=0, ha='center')

plt.show()



# Percentage of patient with or without heart problems in the given dataset

print("Відсоток пацієнтів з проблемами із серцем: "+ str(round(target_temp[0]*100/303, 2)))
print("Відсоток пацієнтів без проблем із серцем: "+ str(round(target_temp[1]*100/303, 2)))

#sns.barplot(data["sex"],y)
pd.crosstab(data.sex, data.sex).plot(kind="bar",figsize=(6,6), color=['blue', 'red'])
#sns.countplot(x='sex', data=data)
plt.xticks(rotation=0, ha='center')
plt.xlabel('Стать')
plt.ylabel('Кількість')
plt.legend(['Жінки', 'Чоловіки'])
plt.show()

###  0 - female and 1 - male



countFemale = len(data[data.sex == 0])
countMale = len(data[data.sex == 1])
print(countFemale)
print(countMale)
print("Відсоток пацієнток:{:.2f}%".format((countFemale)/(len(data.sex))*100))
print("Відсоток пацієнтів:{:.2f}%".format((countMale)/(len(data.sex))*100))
```

```
# Heart Disease Frequency for ages

pd.crosstab(data.age,data.target).plot(kind="bar",figsize=(20,6), color=['blue', 'red'])
plt.title('Розподіл серцевих хвороб по віку')
plt.xlabel('Вік')
plt.ylabel('Частота')
plt.xticks(rotation=0)
plt.legend(['Без серцевої хвороби', 'З серцевою хворобою'])
plt.savefig('heartDiseaseAndAges.png')
plt.show()

plt.figure(figsize=(25,12))
np.float = float
sns.set_context('notebook',font_scale = 1.5)
plt.xlabel('Топ-10 років за роками у датасеті')
plt.ylabel('Кількість')
sns.barplot(x=data.age.value_counts()[:10].index,y=data.age.value_counts()[:10].values)
plt.tight_layout()

data_with_disease = data[data['target'] == 1]

plt.figure(figsize=(25,12))
np.float = float
sns.set_context('notebook',font_scale = 1.5)
plt.xlabel('Топ-10 років за кількістю хворих')
plt.ylabel('Кількість')
sns.set_palette(sns.color_palette('pastel'))
sns.barplot(x=data_with_disease.age.value_counts()[:10].index,y=data_with_disease.age.value_counts()[:10].values)
plt.tight_layout()
```

```
plt.show()



minAge=min(data.age)
maxAge=max(data.age)
meanAge=data.age.mean()
print('Мінімальний вік :',minAge)
print('Максимальний вік :',maxAge)
print('Середній вік :',meanAge)

Young = data_with_disease[(data_with_disease.age>=29)&(data_with_disease.age<40)]
Middle = data_with_disease[(data_with_disease.age>=40)&(data_with_disease.age<55)]
Elder = data_with_disease[(data_with_disease.age>55)]

plt.figure(figsize=(23,10))
sns.set_context('notebook',font_scale = 1.5)
sns.barplot(x=['молоді','середній вік','старший вік'],y=[len(Young),len(Middle),len(Elder)])
plt.xlabel('Розподіл за віковими групами хворих людей')
plt.tight_layout()

Young = data[(data.age>=29)&(data.age<40)]
Middle = data[(data.age>=40)&(data.age<55)]
Elder = data[(data.age>55)]

plt.figure(figsize=(23,10))
sns.set_context('notebook',font_scale = 1.5)
sns.barplot(x=['молоді','середній вік','старший вік'],y=[len(Young),len(Middle),len(Elder)])
plt.tight_layout()
```

```
# Частота серцевих захворювань для статі (де 0 - жінка і 1 - чоловіки, а "червоний" - серцеві захворювання, а "синій" - не має серцевих захворювань)


pd.crosstab(data.sex,data.target).plot(kind="bar",figsize=(20,10),color=['blue','red'])

plt.title('Розподіл хворих і здорових в залежності від статі')

plt.xlabel('Стать (0 = Female, 1 = Male)')

plt.xticks(rotation=0)

plt.legend(["Не мають хвороби", "Мають хвороби"])

plt.ylabel('Кількість')


plt.show()


data_without_disease = data[data['target'] == 0]

pd.crosstab(data_without_disease.sex,data_without_disease.target).plot(kind="bar",figsize=(20,10),color=['blue'])

plt.title('Відсутність серцевих хвороб в залежності від статі')

plt.xlabel('Стать (0 = Female, 1 = Male)')

plt.xticks(rotation=0)

plt.legend(["Не мають серцеві хвороби"])

plt.ylabel('Частота')


# Додати підписи для кожної колонки

for i in range(len(data_without_disease.sex.unique())):

    count = data_without_disease[data_without_disease['sex'] == i]['target'].value_counts().values[0]

    plt.text(i, count, str(count), ha='center')


plt.show()



data_with_disease = data[data['target'] == 1]
```

```
pd.crosstab(data_with_disease.sex,data_with_disease.target).plot(kind="bar",figsize=(20,10),color=['red'])
plt.title('Наявність серцевих хвороб в залежності від статі')
plt.xlabel('Стать (0 = Female, 1 = Male)')
plt.xticks(rotation=0)
plt.legend(["Мають серцеві хвороби"])
plt.ylabel('Частота')

# Додати підписи для кожної колонки
for i in range(len(data_with_disease.sex.unique())):
    count = data_with_disease[data_with_disease['sex'] == i]['target'].value_counts().values[0]
    plt.text(i, count, str(count), ha='center')

plt.show()

run this line after prediction

data.columns = ['age', 'sex', 'chest_pain_type', 'resting_blood_pressure', 'cholesterol', 'fasting_blood_sugar',
'rest_ecg', 'max_heart_rate_achieved',
      'exercise_induced_angina', 'st_depression', 'st_slope', 'num_major_vessels', 'thalassemia', 'target']

# Серцеві хвороби в залежності від Fasting Blood sugar

pd.crosstab(data.fasting_blood_sugar,data.target).plot(kind="bar",figsize=(20,10),color=['blue','red'])
plt.title("Серцеві хвороби в залежності від FBS")
plt.xlabel('FBS - (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)')
plt.xticks(rotation=0)
plt.legend(["Не мають серцевих хвороб", "Мають серцеві хвороби"])
plt.ylabel('Кількість')
plt.show()
```

```
pd.crosstab(data.fasting_blood_sugar, data.fasting_blood_sugar).plot(kind="bar", figsize=(6,6))

plt.xticks(rotation=0)

plt.xlabel("Розподіл за рівнем цукру")

plt.ylabel('Кількість')


data_without_disease = data[data['target'] == 0]

pd.crosstab(data_without_disease.fasting_blood_sugar,data_without_disease.target).plot(kind="bar",figsize=(20,10),color=['blue'])

plt.title('Наявність серцевих хвороб в залежності від FBS')

plt.xlabel('FBS - (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)')

plt.xticks(rotation=0)

plt.legend(["Не мають серцеві хвороби"])

plt.ylabel('Частота')


# Додати підписи для кожної колонки

for i in range(len(data_without_disease.fasting_blood_sugar.unique())):

    count = data_without_disease[data_without_disease['fasting_blood_sugar'] == i]['target'].value_counts().values[0]

    plt.text(i, count, str(count), ha='center')


plt.show()


data_with_disease = data[data['target'] == 1]

pd.crosstab(data_with_disease.fasting_blood_sugar,data_with_disease.target).plot(kind="bar",figsize=(20,10),color=['red'])

plt.title('Наявність серцевих хвороб в залежності від FBS')

plt.xlabel('FBS - (Fasting Blood Sugar > 120 mg/dl) (0 = false; 1 = true)')

plt.xticks(rotation=0)

plt.legend(["мають серцеві хвороби"])

plt.ylabel('Частота')
```

```
# Додати підписи для кожної колонки
for i in range(len(data_with_disease.sex.unique())):
    count = data_with_disease[data_with_disease['fasting_blood_sugar'] ==
i]['target'].value_counts().values[0]
    plt.text(i, count, str(count), ha='center')

plt.show()

# Аналіз білю у грудях (4 типи білю у грудях)

#[Value 1: typical angina, Value 2: atypical angina, Value 3: non-anginal pain, Value 4: asymptomatic]

data["chest_pain_type"].unique()

pd.crosstab(data.chest_pain_type, data.chest_pain_type).plot(kind="bar", figsize=(6,6))
plt.xticks(rotation=0)
plt.xlabel("Розподіл болю у грудях")
plt.ylabel('Кількість')

pd.crosstab(data.chest_pain_type,data.target).plot(kind="bar",figsize=(15,6),color=['blue','red'])
#sns.barplot(x=data["chest_pain_type"], y=data["target"])
plt.xticks(rotation=0)
plt.title("Розподіл по білю у грудях")
plt.ylabel('Частота')
plt.legend(["Не мають серцевих хвороб", "Мають серцеві хвороби"])
plt.xlabel('Біль у грудях (1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic)')
plt.show()

# Аналіз тиску у спокійному стані (mm Hg on admission to the hospital)
```

```
data["resting_blood_pressure"].unique()


plt.figure(figsize=(26, 10))
sns.barplot(x=data["resting_blood_pressure"], y=data["target"], ci=None)
plt.title("Розподіл по тиску у спокійному стані")
plt.ylabel('Частота')
#
pd.crosstab(data.resting_blood_pressure,data.target).plot(kind="bar",figsize=(26,15),color=['blue','red'])
plt.title("Розподіл по тиску у спокійному стані")
plt.xticks(rotation=0)
plt.legend(["Не мають серцевих хвороб", "Мають серцеві хвороби"])
plt.ylabel('Частота')

# Аналіз електрокардіографічного вимірювання у стані спокою (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)


data["rest_ecg"].unique()


plt.figure(figsize=(26, 10))
sns.barplot(x=data["rest_ecg"], y=data["target"], ci=None)
plt.title("ЕКГ")
plt.ylabel('Частота')


pd.crosstab(data.rest_ecg,data.target).plot(kind="bar",figsize=(26,15),color=['blue','red'])
plt.title("ЕКГ")
plt.xticks(rotation=0)
plt.legend(["Не мають серцевих хвороб", "Мають серцеві хвороби"])
plt.ylabel('Частота')
```

Люди з restecg '1' та '0' більш схильні до серцевих хвороб, ніж з restecg '2'

#Analysing Exercise induced angina (1 = yes; 0 = no)

```
data["exercise_induced_angina"].unique()
```

```
plt.figure(figsize=(6, 6))
#sns.barplot(x=data["exercise_induced_angina"],y=data["target"], ci=None, palette=[ "blue", "red"])
pd.crosstab(data.exercise_induced_angina,data.target).plot(kind="bar",figsize=(26,15),color=['blue','red'])
plt.xticks(rotation=0)
plt.title("Стенокардія при фізичному навантаженні")
plt.legend(["Не мають серцевих хвороб", "Мають серцеві хвороби"])
plt.ylabel('Частота')
plt.xlabel('0 - так, 1 - ні')
```

###People with exercise_induced_angina=1 are much less likely to have heart problems

Аналіз схилу пікового ST сегменту при фізичному навантаженні (Значення 1: схильне вгору, Значення 2: пласке, Значення 3: схильне вниз)

```
data["st_slope"].unique()
```

ST сегмент в ЕКГ (електрокардіограма) - це відрізок на графіку, який відображає відносну ізоляцію між правим жовчним протоком та левою кімнатою серця. Вона відображає взаємодію між електролітами в крові та електродією в серцевих стінах. Величина ST сегменту може відображати інфаркт міокарду, наявність запалень у серцевих стінах та інші патології.

```
plt.figure(figsize=(25, 10))
sns.barplot(x=data["st_slope"],y=data["target"], ci=None)
```

```
pd.crosstab(data.st_slope,data.target).plot(kind="bar",figsize=(26,15), color=["blue", "red"])
```

```
plt.xticks(rotation=0)
plt.title("ST сегмент в ЕКГ")
plt.legend(["Не мають серцевих хвороб", "Мають серцеві хвороби"])
plt.ylabel('Частота')
```

Slope '2' causes heart pain much more than Slope '0' and '1'

```
# Analysing number of major vessels (0-3) colored by flourosopy
```

Аналіз кількості основних вен (0-3), кольорованих флуоресцентною випромінюванням: чим більше рух крові, тим краще, тому люди з значенням "ca" рівним 0 мають більш високий рівень ймовірності наявності серцевої хвороби.

```
data["num_major_vessels"].unique()
```

```
### count num_major vessels
```

```
sns.countplot(x=data["num_major_vessels"], data=data)
plt.title("Вени")
plt.ylabel('Частота')
```

```
### comparing with target
```

```
sns.barplot(x=data["num_major_vessels"],y=data['target'], ci=None)
pd.crosstab(data.num_major_vessels,data.target).plot(kind="bar",figsize=(26,15), color=["blue", "red"])
plt.xticks(rotation=0)
plt.title("Вени")
plt.legend(["Не мають серцевих хвороб", "Мають серцеві хвороби"])
plt.ylabel('Частота')
```

num_major_vessels=0 has astonishingly large number of heart patients

Analysing A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)

Аналіз захворювання крові зветься таласемія (3 = нормальне; 6 = фіксований дефект; 7 = повернення дефекту)

```
data["thalassemia"].unique()
```

plotting the thalassemia distribution (0,1,2,3)

```
sns.countplot(x=data["thalassemia"], data=data)
plt.title("таласемія")
plt.ylabel('Частота')
```

```
sns.barplot(x=data["st_depression"],y=data['target'], ci=None)
pd.crosstab(data.st_depression,data.target).plot(kind="bar",figsize=(26,15), color=["blue", "red"])
plt.xticks(rotation=0)
plt.legend(["Не мають серцевих хвороб", "Мають серцеві хвороби"])
plt.ylabel('Частота')
```

Correlation plot

Кореляційний аналіз - це метод статистичної оцінки, який використовується для вивчення сили взаємодії між двома номерно вимірюваними, неперервними змінними (наприклад, висота та вага).

```
cnames=['age','resting_blood_pressure','cholesterol','max_heart_rate_achieved','st_depression','num_major_vessels']
```

```
cinames = ['age', 'sex', 'chest_pain_type', 'resting_blood_pressure', 'cholesterol', 'fasting_blood_sugar', 'rest_ecg', 'max_heart_rate_achieved',
    'exercise_induced_angina', 'st_depression', 'st_slope', 'num_major_vessels', 'thalassemia', 'target']

import seaborn as sns
import numpy as np

f, ax = plt.subplots(figsize=(14, 14))

#Correlation plot
df_corr = data.loc[:,cinames]
#Generate correlation matrix
corr = df_corr.corr()

#Plot using seaborn library
sns.heatmap(corr, annot = True, cmap='coolwarm',linewidths=.1)

# Save the figure
plt.savefig('correlation_plot.png', dpi=300, bbox_inches='tight')
plt.show()

data.drop('target', axis = 1).corrwith(data.target).plot(kind = 'bar', grid = True, figsize = (6,6), title='Кореляція з target')

#Set the width and height of the plot
cnames=['age','resting_blood_pressure','cholesterol','max_heart_rate_achieved','st_depression','num_major_vessels']
f, ax = plt.subplots(figsize=(7, 5))

#Correlation plot
df_corr = data.loc[:,cnames]
```

```
#Generate correlation matrix

corr = df_corr.corr()


#Plot using seaborn library

sns.heatmap(corr, annot = True, cmap='coolwarm',linewidths=.1)

plt.show()


##Correlation analysis


df_corr = data.loc[:,cnames]

df_corr


# Поділ датасету на Train та Test вибірки


from sklearn.model_selection import train_test_split


predictors = data.drop("target",axis=1)

target = data["target"]


X_train,X_test,Y_train,Y_test = train_test_split(predictors,target,test_size=0.20,random_state=0)

print("У навчальній вибірці {0} записів та у тестовій вибірці {1} записів.".format(X_train.shape[0],
X_test.shape[0]))


X_train.shape


X_test.shape


Y_train.shape


Y_test.shape
```

importing Accuracy score

accuracy_score є метрикою для оцінки точності моделі машинного навчання. Вона рахує відношення кількості правильних прогнозів моделі до загальної кількості прогнозів. Отже, accuracy_score = (кількість правильних прогнозів) / (загальна кількість прогнозів). Результат від accuracy_score є дійсним числом від 0 до 1, де 1 означає, що всі прогнози моделі правильні, а 0 означає, що всі прогнози неправильні.

```
from sklearn.metrics import accuracy_score
```

```
# Logistic regression
```

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression().fit(X_train, Y_train)
print("Training set score: {:.3f}".format(logreg.score(X_train, Y_train)))
print("Test set score: {:.3f}".format(logreg.score(X_test, Y_test)))
```

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
lr.fit(X_train,Y_train)
Y_pred_lr = lr.predict(X_test)

score_lr = round(accuracy_score(Y_pred_lr,Y_test)*100,15)

print("Точність з використанням Logistic Regression: "+str(score_lr)+" %")
```

```
# Confusion Matrix for Logistic Regression
```

Confusion Matrix (Матриця плутанини) - це техніка оцінки якості класифікатора в машинному навчанні. Вона представляє собою таблицю, яка відображає взаємодію між реальними класами та

відповідними їм класами, визначеними класифікатором. Матриця допоможе визначити, скільки об'єктів класифікуються правильно та скільки помилок.

```
from sklearn.metrics import confusion_matrix

matrix = confusion_matrix(Y_test, Y_pred_lr)

sns.heatmap(matrix,annot = True, fmt = "d")
```

fmt = d is format = default

```
# precision Score

from sklearn.metrics import precision_score

precision = precision_score(Y_test, Y_pred_lr)

print("Точність у логістичній регресії: ", precision)
```

Recall

"Recall Score" це метрика, яка використовується для оцінки якість моделей машинного навчання в задачах класифікації. Вона показує, скільки з действительно позитивних класів було вірно визначено моделлю. Високий Recall Score означає, що модель відносно непомилково визначає позитивні класи, але існує можливість високої кількості false positive.

```
from sklearn.metrics import recall_score

recall = recall_score(Y_test, Y_pred_lr)

print("Параметр Recall: ",recall)
```

F-Score

F-Score, також відомий як F-мера, є метрикою оцінювання моделей машинного навчання. Вона використовує гармонічний середній з відношення точності (Precision) та Повторності (Recall). F-Score відрізняється від інших метрик, таких як Precision та Recall, тим, що вона ураховує обидва ці показники в одному відношенні. Вона може бути використана для оцінки якості моделей бінарної класифікації.

balance of precision and recall score

$F = \frac{2*precision*recall}{precision+recall}$

```
CM =pd.crosstab(Y_test, Y_pred_lr)
CM
```

```
print((2*precision*recall)/(precision+recall))
```

```
#let us save TP, TN, FP, FN
TN=CM.iloc[0,0]
FP=CM.iloc[0,1]
FN=CM.iloc[1,0]
TP=CM.iloc[1,1]
```

```
Accuracy_lr = (TP+TN)/(TP+TN+FN+FP)
Recall_lr = TP/(TP+FN)
```

```
Precision_lr = TP/(TP+FP)
F1score_lr = (2*Recall_lr*Precision_lr)/(Recall_lr+Precision_lr)

print("Accuracy_lr " + str(Accuracy_lr))
print("Recall_lr " + str(Recall_lr))
print("Precision_lr " + str(Precision_lr))
print("F1score_lr " + str(F1score_lr))

# Random Forest

#Random forest with 100 trees
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100, random_state=0)
rf.fit(X_train, Y_train)

print("Accuracy on training set: {:.3f}".format(rf.score(X_train, Y_train)))
print("Accuracy on test set: {:.3f}".format(rf.score(X_test, Y_test)))
```

Тепер обріжемо глибину дерев і перевіримо точність.

```
rf1 = RandomForestClassifier(max_depth=3, n_estimators=100, random_state=0)
rf1.fit(X_train, Y_train)

print("Accuracy on training set: {:.3f}".format(rf1.score(X_train, Y_train)))
print("Accuracy on test set: {:.3f}".format(rf1.score(X_test, Y_test)))

from sklearn.ensemble import RandomForestClassifier

max_accuracy = 0
```

```
for x in range(2000):
    rf = RandomForestClassifier(random_state=x)
    rf.fit(X_train,Y_train)
    Y_pred_rf = rf.predict(X_test)
    current_accuracy = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x

print(max_accuracy)
print(best_x)

rf = RandomForestClassifier(random_state=best_x)
rf.fit(X_train,Y_train)
Y_pred_rf = rf.predict(X_test)

Y_pred_rf.shape

score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,12)

print("The accuracy score achieved using Decision Tree is: " + str(score_rf) + " %")

## confusion matrix of Random Forest

from sklearn.metrics import confusion_matrix

matrix= confusion_matrix(Y_test, Y_pred_rf)

sns.heatmap(matrix,annot = True, fmt = "d")
```

```
# precision score of Random Forest

from sklearn.metrics import precision_score

precision = precision_score(Y_test, Y_pred_rf)

print("Precision: ", precision)

# recall of Random Forest

from sklearn.metrics import recall_score

recall = recall_score(Y_test, Y_pred_rf)

print("Recall is: ",recall)

# F-score of Random Forest

print((2*precision*recall)/(precision+recall))

### cm using bad style

CM =pd.crosstab(Y_test, Y_pred_rf)
CM

TN=CM.iloc[0,0]
FP=CM.iloc[0,1]
FN=CM.iloc[1,0]
TP=CM.iloc[1,1]
```

```
Accuracy_rf = (TP+TN)/(TP+TN+FN+FP)

Recall_rf = TP/(TP+FN)

Precision_rf = TP/(TP+FP)

F1score_rf = (2*Recall_rf*Precision_rf)/(Recall_rf+Precision_rf)


print("Accuracy_rf " + str(Accuracy_rf))

print("Recall_rf " + str(Recall_rf))

print("Precision_rf " + str(Precision_rf))

print("F1score_rf " + str(F1score_rf))


## False negative rate of the model of Random Forest


fnr=FN*100/(FN+TP)

fnr


# Naive Bayes


from sklearn.naive_bayes import GaussianNB


nb = GaussianNB()


nb.fit(X_train,Y_train)


Y_pred_nb = nb.predict(X_test)


Y_pred_nb.shape


score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)
```

```
print("The accuracy score achieved using Naive Bayes is: "+str(score_nb)+" %")

## confusion matrix of Naive Bayes

from sklearn.metrics import confusion_matrix

matrix= confusion_matrix(Y_test, Y_pred_nb)

sns.heatmap(matrix,annot = True, fmt = "d")

# precision score of Naive Bayes

from sklearn.metrics import precision_score

precision = precision_score(Y_test, Y_pred_nb)

print("Precision: ",precision)

# recall of Naive Bayes

from sklearn.metrics import recall_score

recall = recall_score(Y_test, Y_pred_nb)

print("Recall is: ",recall)

# F-score of Naive Bayes

print((2*precision*recall)/(precision+recall))
```

```
### bad cm style

CM = pd.crosstab(Y_test, Y_pred_nb)
CM

TN=CM.iloc[0,0]
FP=CM.iloc[0,1]
FN=CM.iloc[1,0]
TP=CM.iloc[1,1]

Accuracy_nb = (TP+TN)/(TP+TN+FN+FP)
Recall_nb = TP/(TP+FN)
Precision_nb = TP/(TP+FP)
F1score_nb = (2*Recall_nb*Precision_nb)/(Recall_nb+Precision_nb)

print("Accuracy_nb " + str(Accuracy_nb))
print("Recall_nb " + str(Recall_nb))
print("Precision_nb " + str(Precision_nb))
print("F1score_nb " + str(F1score_nb))

## false negative rate of the model

fnr = FN*100/(FN+TP)
fnr

# KNN (k-Nearest Neighbors)

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=7)
```

```
knn.fit(X_train,Y_train)
Y_pred_knn=knn.predict(X_test)

Y_pred_knn.shape

score_knn = round(accuracy_score(Y_pred_knn,Y_test)*100,2)

print("The accuracy score achieved using KNN is: "+str(score_knn)+" %")

## Confusion matrix of KNN

from sklearn.metrics import confusion_matrix

matrix= confusion_matrix(Y_test, Y_pred_knn)

sns.heatmap(matrix,annot = True, fmt = "d")

# precision score of KNN

from sklearn.metrics import precision_score

precision = precision_score(Y_test, Y_pred_knn)

print("Precision: ",precision)

# recall of KNN

from sklearn.metrics import recall_score

recall = recall_score(Y_test, Y_pred_knn)
```

```
print("Recall is: ",recall)

# F-score of KNN

print((2*precision*recall)/(precision+recall))

### bad cm

CM = pd.crosstab(Y_test, Y_pred_knn)
CM

TN=CM.iloc[0,0]
FP=CM.iloc[0,1]
FN=CM.iloc[1,0]
TP=CM.iloc[1,1]

## false negative rate of the model

fnr = FN*100/(FN+TP)
fnr

## for neighbors = 4

from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=4)
knn.fit(X_train,Y_train)
Y_pred_knn4=knn.predict(X_test)
```

```
Y_pred_knn4.shape

score_knn_4 = round(accuracy_score(Y_pred_knn,Y_test)*100,2)

print("The accuracy score achieved using KNN is: "+str(score_knn_4)+" %")

## false negative rate

CM = pd.crosstab(Y_test, Y_pred_knn4)
TN=CM.iloc[0,0]
FP=CM.iloc[0,1]
FN=CM.iloc[1,0]
TP=CM.iloc[1,1]
fnr = FN*100/(FN+TP)
fnr

Accuracy_knn = (TP+TN)/(TP+TN+FN+FP)
Recall_knn = TP/(TP+FN)
Precision_knn = TP/(TP+FP)
F1score_knn = (2*Recall_knn*Precision_knn)/(Recall_knn+Precision_knn)

print("Accuracy_knn " + str(Accuracy_knn))
print("Recall_knn " + str(Recall_knn))
print("Precision_knn " + str(Precision_knn))
print("F1score_knn " + str(F1score_knn))

# Decision Tree

from sklearn.tree import DecisionTreeClassifier
tree1 = DecisionTreeClassifier(random_state=0)
```

```
tree1.fit(X_train, Y_train)

print("Accuracy on training set: {:.3f}".format(tree1.score(X_train, Y_train)))

print("Accuracy on test set: {:.3f}".format(tree1.score(X_test, Y_test)))
```

Точність на навчальному наборі становить 100%, в той час як точність тестового набору набагато гірше. Це свідчить про те, що дерево переобладнується і погано узагальнюється до нових даних. Тому потрібно застосувати попередню обрізку до дерева. Встановлюємо max_depth=3, обмеження глибини залягання дерева зменшує перепідгон. Це призводить до зниження точності на тренувальному наборі, але поліпшення на тестовому наборі.

```
tree1 = DecisionTreeClassifier(max_depth=3, random_state=0)

tree1.fit(X_train, Y_train)

print("Accuracy on training set: {:.3f}".format(tree1.score(X_train, Y_train)))

print("Accuracy on test set: {:.3f}".format(tree1.score(X_test, Y_test)))


from sklearn.tree import DecisionTreeClassifier


max_accuracy = 0



for x in range(200):
    dt = DecisionTreeClassifier(random_state=x)
    dt.fit(X_train,Y_train)
    Y_pred_dt = dt.predict(X_test)
    current_accuracy = round(accuracy_score(Y_pred_dt,Y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x


#print(max_accuracy)

#print(best_x)
```

```
dt = DecisionTreeClassifier(random_state=best_x)
dt.fit(X_train,Y_train)
Y_pred_dt = dt.predict(X_test)

print(Y_pred_dt.shape)

# Discision Tree Visualization

df = pd.read_csv('heart.csv')

df.head()

from pandas import DataFrame, Series
from IPython.display import Image
from io import StringIO
import pydotplus
from sklearn import preprocessing

def plot_decision_tree(clf,feature_name,target_name):
    dot_data = StringIO()
    tree.export_graphviz(clf, out_file=dot_data,
                         feature_names=feature_name,
                         class_names=target_name,
                         filled=True, rounded=True,
                         special_characters=True)
    graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
    return Image(graph.create_png())
```

```
from sklearn import tree

clf = tree.DecisionTreeClassifier()

clf = clf.fit(X_train,Y_train)


plot_decision_tree(clf, X_train.columns,df.columns[1])


### accuracy


score_dt = round(accuracy_score(Y_pred_dt,Y_test)*100,2)


print("The accuracy score achieved using Decision Tree is: "+str(score_dt)+" %")


## Confusion Matrix of Decision Tree


from sklearn.metrics import confusion_matrix


matrix= confusion_matrix(Y_test, Y_pred_dt)


sns.heatmap(matrix,annot = True, fmt = "d")


# precision score of Decision Tree


from sklearn.metrics import precision_score


precision = precision_score(Y_test, Y_pred_dt)


print("Precision: ",precision)


# recall of Decision Tree
```

```
from sklearn.metrics import recall_score

recall = recall_score(Y_test, Y_pred_dt)

print("Recall is: ",recall)

# F-score of Decision Tree

print((2*precision*recall)/(precision+recall))

### bad cm style

CM = pd.crosstab(Y_test, Y_pred_dt)
CM

## false negative rate

TN=CM.iloc[0,0]
FP=CM.iloc[0,1]
FN=CM.iloc[1,0]
TP=CM.iloc[1,1]
fnr = FN*100/(FN+TP)
fnr

Accuracy_dt = (TP+TN)/(TP+TN+FN+FP)
Recall_dt = TP/(TP+FN)
Precision_dt = TP/(TP+FP)
F1score_dt = (2*Recall_dt*Precision_dt)/(Recall_dt+Precision_dt)

print("Accuracy_dt " + str(Accuracy_dt))
```

```
print("Recall_dt " + str(Recall_dt))
print("Precision_dt " + str(Precision_dt))
print("F1score_dt " + str(F1score_dt))

# Support Vector Machine

from sklearn.svm import SVC
svc_c = SVC(kernel = 'linear', random_state=0)
svc_c.fit(X_train, Y_train)
Y_pred_svm = svc_c.predict(X_test)
print("Accuracy on training set: {:.3f}".format(svc_c.score(X_train, Y_train)))
print("Accuracy on test set: {:.3f}".format(svc_c.score(X_test, Y_test)))

score_svm = round(accuracy_score(Y_pred_svm,Y_test)*100,2)

print("The accuracy score achieved using SVM is: "+str(score_svm)+" %")

CM = pd.crosstab(Y_test, Y_pred_svm)
CM
TN=CM.iloc[0,0]
FP=CM.iloc[0,1]
FN=CM.iloc[1,0]
TP=CM.iloc[1,1]

from sklearn.metrics import confusion_matrix
matrix= confusion_matrix(Y_test, Y_pred_svm)
sns.heatmap(matrix,annot = True, fmt = "d")

Accuracy_svm = (TP+TN)/(TP+TN+FN+FP)
Recall_svm = TP/(TP+FN)
```

```
Precision_svm = TP/(TP+FP)

F1score_svm = (2*Recall_svm*Precision_svm)/(Recall_svm+Precision_svm)


print("Accuracy_svm " + str(Accuracy_svm))

print("Recall_svm " + str(Recall_svm))

print("Precision_svm " + str(Precision_svm))

print("F1score_svm " + str(F1score_svm))


# FINAL SCORE


scores = [score_lr,score_nb,score_knn,score_dt,score_rf, score_svm]

algorithms = ["Logistic Regression","Naive Bayes","K-Nearest Neighbors","Decision Tree","Random Forest",
'Support Vector Machine']


for i in range(len(algorithms)):

    print("The accuracy score achieved using "+algorithms[i]+" is: "+str(scores[i])+" %")


sns.set(rc={'figure.figsize': (15, 8)})

plt.xlabel("Algorithms")

plt.ylabel("Accuracy score")


sns.barplot(x=algorithms, y=scores)


plt.show()



# In this project, We have used Machine Learning to predict whether a person is suffering from a heart
disease or not. After importing the data, we have analysed it using plots. Then, generated categorical
features and scaled other features. Then applied five Machine Learning algorithms: K Nearest Neighbors
Classifier, Naive Bayes, Logistic Regression, Decision Tree Classifier and Random Forest Classifier. In the end,
Random Forest achieved the highest score of 95.08%.
```