

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Інформаційна система оцінювання функціонального стану професійних спортсменів»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології
проекткування»

Виконавець роботи: студент групи ІТ.м-81 Шестак Максим Олександрович

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

« » грудня 2019 р.

Науковий керівник

(підпис)

к.т.н., доц., Шендрик В. В.

Голова комісії

(підпис)

Шифрін Д.М.

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

Студент _____

(підпис)

Суми-2019

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
«__» _____ 2019 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентові

Шестак Максим Олександрович

(прізвище, ім'я, по батькові)

1 Тема проекту Інформаційна система оцінювання функціонального стану професійних спортсменів

затверджена наказом по університету від «19» листопада 2019 р. №2305-III

2 Термін здачі студентом закінченого проекту « 10 » грудня 2019 р.

3 Вхідні дані до проекту _____

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) _____

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1	Дослідження актуальності проблеми	02.07.2019	
2	Огляд аналогів	16.07.2019	
3	Вибір інструментів реалізації	13.08.2019	
4	Проектування БД	27.08.2019	
5	Робота з даними	10.09.2019	
6	Реалізація моделей та порівняння	24.09.2019	
7	Створення API для моделі та інтеграція в ІС	08.10.2019	
8	Тестування	22.10.2019	

Магістрант _____

Шестак М.О.

Керівник роботи _____

к.т.н., доц. Шендрик В.В.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Інформаційна система оцінювання функціонального стану професійних спортсменів».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 31 найменування, додатків. Загальний обсяг роботи – 72 сторінка, у тому числі 58 сторінок основного тексту, 3 сторінки списку використаних джерел, 11 сторінок додатків.

Кваліфікаційну роботу магістра присвячено розробці інформаційної системи для оцінювання функціонального стану професійних спортсменів. В роботі проведено аналіз предметної області, огляд аналогів. У роботі виконано планування та моделювання інформаційної системи, робота з даними, створення та порівняння моделей, реалізація інтеграції моделей в веб-застосунок. Результатом проведеної роботи є розроблена інформаційна система для оцінювання функціонального стану професійних спортсменів.

Обсяг реферату – одна повна сторінка

Ключові слова: інформаційна система, веб-застосунок, база даних, машинне навчання.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області.....	9
1.1 Дослідження актуальності проблеми.....	9
1.2 Огляд інформаційних систем обліку показників стану здоров'я.....	10
2 Постановка задачі та методи дослідження.....	15
2.1 Мета та задачі дослідження.....	15
2.2 Методи дослідження.....	16
2.2.1 Інструменти для роботи з даними та візуалізації.....	16
2.2.2 Інструменти для створення моделей.....	19
2.2.3 Інструменти для реалізації БД.....	21
2.2.4 Інструменти для створення API.....	23
2.2.5 Інструменти для реалізації веб-застосунку.....	23
2.2.6 Методи машинного навчання.....	24
3 Моделювання інформаційної системи.....	31
3.1 Структура інформаційної системи.....	31
3.2 Моделювання інформаційної системи.....	32
3.3 Проектування бази даних.....	34
3.4 Збір даних.....	40
3.5 Робота з даними. Розвідувальний аналіз даних.....	42
3.6 Реалізація моделей та їх порівняння.....	50
3.7 Створення API для обраної моделі та інтеграція в ІС.....	52
4 Реалізація інформаційної системи оцінювання функціонального стану професійних спортсменів.....	53

Висновки	57
Список використаних джерел	59
Додаток А.....	62

ВСТУП

Оцінка функціонального стану спортсменів тісно пов'язана з медициною, а ефективне використання інформаційних систем є однією з передумов забезпечення її якості.

Використання інформаційних систем в медицині надає можливості для аналізу великих об'ємів даних. Обробка інформації великого потоку пацієнтів також неможлива без використання інформаційних систем. [29]

Оскільки в спортивній сфері досить важливим є спортивний відбір, адже постійно зростає конкуренція і необхідно вчасно виявляти найперспективніших кандидатів. [23]

Застосування сучасних технологічних інструментів для допомоги лікарям у підвищенні точності діагностування хвороб стає однією з найактивніших тем досліджень, особливо алгоритмів машинного навчання та штучного інтелекту.

Машинне навчання – це технологія аналізу даних, яка надає можливість комп'ютерам навчатись. Цей спосіб використовує обчислювальні методи для отримання інформації безпосередньо з даних. Ефективність машинного навчання покращується відповідно до якості даних, а разом з цим також вдосконалюється процес прогнозування захворювань.

Методи машинного навчання не можуть повністю замінити спеціалістів, але натомість можуть легко давати відповіді на більш прості запитання. Їх можна використовувати для створення автоматизованих систем діагностики, для розпізнавання неструктурованих даних, для аналізу та передбачення, а також для автоматичної класифікації та перевірки інформації.

Об'єктом дослідження є процес використання методів машинного навчання в медицині та спорті.

Предметом дослідження є використання методів машинного навчання для аналізу функціонального стану.

Основна мета цього дослідження – створення інформаційної системи з використанням методик машинного навчання для оцінювання функціонального стану професійних спортсменів.

Серед задач можна виділити наступні:

- формування списку вимог;
- вибір методів дослідження;
- підготовка та обробка даних;
- реалізація моделей машинного навчання;
- тестування та вибір найоптимальнішої моделі;
- інтеграція із ІС.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження актуальності проблеми

У наш час все частіше виникають завдання, зібрані назвою Big Data. Розвиток діяльності людини веде до накопичення великої кількості даних. Аналіз цих даних може бути проведений вручну, а може бути виконаним за допомогою машинного навчання, тобто за допомогою алгоритмів автоматичного знаходження закономірностей в емпіричних даних. Одна з областей застосування засобів машинного навчання – медицина.

Іноді термін штучний інтелект (ШІ) згадують в контексті кінцевої мети навчання машин. Кінцева мета – навчити їх думати самостійно. Це явище відоме, як «загальний штучний інтелект», але до цього ще далеко. Однак одна форма штучного інтелекту, а саме, машинне навчання, вже починає проявляти себе в області медицини найкращим чином, і в один прекрасний день зможе виконувати всі функції від читання рентгенівських знімків до виставлення рахунків за медичні послуги.

В рамках машинного навчання відбувається підбір алгоритмів шляхом підстановки величезної кількості змінних в пошуках комбінацій для точного прогнозу. Здатність машинного навчання обробляти величезні масиви даних з метою підготовки прогнозів – це вже суттєвий прогрес.

Наприклад, машинне навчання може підвищити точність медичної візуалізації та діагностики патологій. Добре відомо, що проведення оцінки двома експертами буде більш точним в порівнянні з висновками одного фахівця. А за допомогою функції машинного навчання комп'ютер може виконувати контроль якості і виявляти найбільш часто пропущені особливості з метою поліпшення якості та результатів. [29]

Машинне навчання може також застосовуватися для виявлення прихованих симптомів захворювання і підбору відповідних методів лікування,

що особливо важливо для лікування захворювань, які вкрай бажано діагностувати на ранніх стадіях, наприклад, в рамках скринінгу новонародженого, а також для вибору оптимальних методів лікування.

Однак яким би вражаючим не видався досвід застосування машинного навчання, воно не зможе замінити лікарів та інших медичних працівників. За допомогою машинного навчання можна підтвердити діагноз і навіть спрогнозувати результат хвороби, але неможливо пояснити причини того, що відбувається. Для цього, як і раніше, необхідні кваліфікація і знання досвідченого лікаря. [21]

1.2 Огляд інформаційних систем обліку показників стану здоров'я

Було проаналізовано декілька наявних медичних систем. Вони відрізняються ціною та наявністю різних функціональних можливостей. Але все ж будь-яка медична система дає можливість зберігати та обробляти основну медичну інформацію, формувати деякі звіти та включає контроль доступу.

MEDODS

MEDODS є медичною системою нового покоління. Застосовується як в приватних клініках так і в державних установах.

Не потребує тривалого навчання по роботі з системою завдяки дружньому і простому інтерфейсу.

Володіє як основними поширеними функціями як: запис на прийом і виставлення рахунків, так і потужні маркетингові модулі.

Основним плюсом за який її найчастіше вибирають - це вартість MEDODS, вона дуже демократична і швидко окупається.

MEDODS включає в себе:

- хмарне рішення;
- SMS і email сповіщення пацієнтів;
- онлайн запис на прийом до лікаря;
- шаблони різних протоколів;
- електронна медична карта пацієнта;
- можливість ведення складського обліку;
- створення звітів;
- інтеграція з онлайн касами;
- інтеграція з лабораторіями;
- інтеграція з букінг-порталами;
- інтеграція з телефонією.

Medesk

Управління приватною клінікою за допомогою медичної системи Medesk це поєднання сучасних технологій і медицини.

Систему відрізняє простий і зрозумілий інтерфейс, що дозволяє без особливих зусиль освоїти роботу з системою вашими фахівцями.

Medesk включає в себе:

- хмарне рішення;
- SMS оповіщення пацієнтів;
- онлайн запис на прийом до лікаря;
- шаблони різних протоколів;
- інтеграція з онлайн касами;
- дистанційна робота по консультуванню пацієнтів.

MedElement

Основною метою розробки MedElement ставлять підвищення якості медичних послуг для населення.

Основними плюсами системи можна виділити:

- хмарне зберігання даних;
- система електронного кабінету лікаря;
- велика база медичних довідників;
- мобільний додаток можна скачати безкоштовно;
- окремий модуль для автоматизації блоку громадського харчування клініки;
- часті оновлення медичної системи.

Clinic365

Дана медична інформаційна система має дружній інтерфейс для всіх основних функцій медичної інформаційної системи

Дане рішення включає в себе такі модулі як:

- картотека пацієнтів;
- розклад;
- медичний документообіг;
- фінанси для контролю оплат.

Основною особливістю МІС Clinic365 є можливість побудувати алгоритм роботи з пацієнтом. У картку включається така інформація як, перевагу клієнта, історія контактів з пацієнтом, і звичайно медична інформація. Є потужна технічна підтримка продукту.

Інфоклініка

Медична інформаційна система «ІНФОКЛІНІКА» допоможе побудувати структуроване інформаційний простір медичної клініки. Дане рішення буде зручно як пацієнтам, так і персоналу клініки.

МІС "Інформатика" володіє всіма необхідними інструментами, що дозволяє з легкістю налаштувати інтеграцію з іншими системами вашої клініки, а так-же дозволяє налаштувати гнучку комунікацію з пацієнтами.

У складі системи багато заготовлених шаблонів звітів, протоколів, історій хвороби.

IDENT

IDENT це медична інформаційна система для стоматологій, яка дозволить уникнути паперової тяганини з пацієнтами. Відмінною особливістю є те, що вона створювалася під керівництвом стоматологів і власне для стоматологів.

Є кілька версій МІС, відмінність у кількості включених модулів.

Навіть з базовим функціоналом інформаційна система дозволяє:

- швидко справлятися з розкладом і записом на прийом;
- вести бухгалтерію;
- швидко роздруковувати документи для пацієнта;
- проводити статистичні звіти.

ClinicIQ

Даний продукт є онлайн сервісом для стоматологій і медичних центрів.

Має широкий функціонал починаючи від бази пацієнтів, закінчуючи, виставленням рахунків та бухгалтерією.

Є інтеграція з порталом DocDoc і месенджером Viber. Основним плюсом даного сервісу є дуже продумана система запису на прийом і обліку пацієнтів.

Проаналізувавши дані медичні системи було створено порівняльну таблицю (табл. 1.1)

Таблиця 1.1 – Порівняння медичних інформаційних систем

Назва	Картогека пацієнтів	Конструктор аналізів	Контроль доступу	Особистий кабінет	Прогнозування показників
MEDODS	+	-	+		-
Medesk	+	-	+	+	-
MedElement	+	-	+	+	-
Clinic365	+	+	+	-	-
Інфоклініка	+	-	+	+	-
IDENT	+	+	+	-	-
ClinicIQ	+	-	+	-	-

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі дослідження

Метою даної роботи є створення інформаційної системи для оцінювання функціонального стану професійних спортсменів. Дану систему зможуть використовувати як медичні працівники, так і звичайні користувачі.

Серед основних необхідних медичних досліджень можна виділити наступні:

- антропометрія;
- клінічний аналіз крові;
- клінічний аналіз сечі;
- спірографія;
- аналіз крові біохімічний;
- ехокардіографія;
- гоніометрія.

Отримані результати досліджень можна буде використовувати для оцінки та прогнозування функціонального стану спортсмена.

Основні задачі дослідження:

- обрати методи дослідження на виконати планування;
- виконати моделювання інформаційної системи;
- провести розвідувальний аналіз даних та виконати їх обробку;
- реалізувати обрані моделі машинного навчання, провести навчання та виконати порівняння;
- виконати інтеграцію із існуючою інформаційною системою.

2.2 Методи дослідження

Перед початком роботи над створенням інформаційної системи необхідно обрати методи і технології. Їх можна розподілити за категоріями призначення, а саме:

- для роботи з даними;
- для створення моделей машинного навчання;
- для реалізації БД;
- для створення API взаємодії моделі та веб-застосунку.

2.2.1 Інструменти для роботи з даними та візуалізації

Pandas – це бібліотека, яка надає дуже зручні з точки зору використання інструменти для зберігання даних і роботи з ними.

Особливість pandas полягає в тому, що ця бібліотека дуже швидка, гнучка і виразна. Це важливо, тому що вона використовується з мовою Python, яка не відрізняється високою продуктивністю. Pandas прекрасно підходить для роботи з одновимірними і двовимірними таблицями даних, добре інтегрована з іншими зовнішніми системами – є можливість працювати з файлами CSV, таблицями Excel, а також існує можливість використання разом мовою R.

Pandas добре підходять для різних типів даних:

- табличні дані (схожі на таблиці SQL або Excel);
- дані про дату або час;
- довільні матричні дані (введені однорідно або неоднорідно) з мітками рядків і стовпців;
- будь-яка інша форма спостережних або статистичних наборів даних.

Існує дві основні структури даних Pandas – це Series (одновимірні) та DataFrame (двовимірні), обробляють переважну більшість типових випадків використання у фінансах, статистиці, суспільствознавстві та багатьох галузях інженерії. Бібліотека Pandas побудована на основі бібліотеки NumPy і може добре інтегруватися в наукове обчислювальне середовище з багатьма іншими бібліотеками сторонніх організацій.

Основні переваги Pandas:

- проста робота з відсутніми даними (представленими як NaN), з цілими числами та з плаваючою точкою;
- легко модифікувати: стовпці можна вставляти та видаляти з DataFrame
- автоматичне та явне вирівнювання даних: об'єкти можуть бути явно вирівняні використовуючи набір міток;
- потужна та гнучка функція groupBy дозволяє виконувати операції комбінування і поділу на наборах даних для агрегування або перетворення даних;
- легко конвертувати необроблені та по-різному індексовані дані в інших структурах даних Python та NumPy в об'єкти DataFrame;
- інтелектуальний поділ даних на основі міток;
- інтуїтивне об'єднання та приєднання наборів даних;
- гнучка перестановка та поворот наборів даних;
- надійні інструменти вводу-виводу для завантаження даних із файлів CSV, файлів Excel, баз даних та збереження і завантаження даних у надшвидкому форматі HDF5;
- функціонал, орієнтований на роботу з часом: генерація діапазону дат і перетворення частоти, зміщення дати тощо.

Seaborn – це бібліотека для створення статистичної графіки в Python. Вона побудована поверх бібліотеки matplotlib і тісно інтегрована з структурами даних Pandas.

Ось деякі з функцій, які пропонує бібліотека Seaborn:

- API, орієнтований на набір даних, для вивчення зв'язків між декількома змінними;
- спеціалізована підтримка використання категоричних змінних;
- варіанти візуалізації уніваріантних або біваріантних розподілів та порівняння їх між підмножинами даних;
- автоматичне оцінювання та побудова лінійних моделей регресії для різних змінних;
- зручні графіки, що дають уявлення про загальну структуру складних наборів даних;
- абстракції високого рівня для структурування сіток з декількома ділянками, які дозволяють легко будувати складні візуалізації;
- інструменти для вибору кольорової палітри.

Seaborn дозволяє зробити візуалізацію центральною частиною вивчення та розуміння даних. Функції, орієнтовані на набір графіків, можуть працювати з фреймами даних та масивами, що містять цілі набори даних, внутрішньо виконують необхідне семантичне відображення та статистичну агрегацію для отримання інформативних графіків.

2.2.2 Інструменти для створення моделей

Бібліотека Scikit-learn – найпопулярніший вибір для вирішення завдань класичного машинного навчання. Вона надає широкий набір алгоритмів навчання з учителем і без вчителя. Навчання з учителем передбачає наявність розміченого датасета, в якому відомо значення цільового показника. У той час як навчання без вчителі не передбачає наявності розмітки в датасета – потрібно навчитися отримувати корисну інформацію з довільних даних. Одна з основних переваг бібліотеки полягає в тому, що вона працює на основі декількох поширених математичних бібліотек, і легко інтегрує їх одну з іншою. Ще однією перевагою є широка спільнота і докладна документація. Scikit-learn широко використовується для промислових систем, в яких застосовуються алгоритми класичного машинного навчання, для досліджень, а так само для новачків, які тільки почали робити перші кроки в області машинного навчання.

Для своєї роботи, scikit-learn використовує такі популярні бібліотеки:

- NumPy: математичні операції і операції над тензорами;
- SciPy: науково-технічні обчислення;
- Matplotlib: візуалізація даних;
- IPython: інтерактивна консоль для Python;
- SymPy: символічна математика;
- Pandas: обробка, маніпуляція і аналіз даних.

Що містить Scikit-learn

До завдань бібліотеки не входить завантаження, обробка, маніпуляція даними і їх візуалізація. З цими завданнями відмінно справляються бібліотеки Pandas і NumPy. Scikit-learn спеціалізується на алгоритмах машинного навчання для вирішення завдань навчання з учителем: класифікації (прогноз ознаки, множина допустимих значень якої обмежена) і регресії (прогноз ознаки з дійсними значеннями), а також для задач навчання без учителя:

кластеризації (розбиття даних по класах, які модель визначить сама), зниження розмірності (подання даних в просторі меншої розмірності з мінімальними втратами корисної інформації) і детектування аномалій.

Бібліотека реалізує наступні основні методи:

- лінійні – моделі, завдання яких побудувати розділяючу (для класифікації) або апроксимуючу (для регресії) гіперплощину;
- метричні – моделі, які обчислюють відстань по одній з метрик між об'єктами вибірки, і приймають рішення в залежності від цієї відстані (метод k-найближчих сусідів);
- дерева рішень – навчання моделей, що базуються на безлічі умов, оптимально обраних для вирішення завдання;
- ансамблеві методи – методи, засновані на деревах рішень, які комбінують переваги безлічі дерев, і таким чином підвищують їх якість роботи, а також дозволяють проводити відбір ознак (бустінг, беггінг, випадковий ліс, мажоритарне голосування);
- нейронні мережі – комплексний нелінійний метод для задач регресії і класифікації;
- SVM – нелінійний метод, який навчається визначати межі прийняття рішень;
- наївний баєсів класифікатор – прямий ймовірнісний розподіл для задачі класифікації;
- PCA – лінійний метод зниження розмірності і відбору ознак;
- t-SNE – нелінійний метод зниження розмірності;
- метод K-середніх – найпоширеніший метод для кластеризації, що приймає на вхід число кластерів, за якими повинні бути розподілені дані;

- крос-валідація – метод, при якому для навчання використовується весь датасет (на відміну від розбиття на вибірки train/test), проте навчання відбувається багаторазово, і в якості валідаційної вибірки на кожному кроці виступають різні частини датасету. Підсумковий результат являє собою усереднення отриманих результатів;
- Grid Search – метод для знаходження оптимальних гіперпараметрів моделі шляхом побудови сітки із значень гіперпараметрів і послідовного навчання моделей з усіма можливими комбінаціями гіперпараметрів з сітки.

Це – лише базовий список. Крім цього, Scikit-learn містить функції для розрахунку значень метрик, вибору моделей, препроцесінга даних та інші.

2.2.3 Інструменти для реалізації БД

Базу даних було реалізовано за допомогою NoSQL бази даних MongoDB.

Нереляційна база даних - це база даних, в якій на відміну від більшості традиційних систем баз даних не використовується таблична схема рядків і стовпців. У цих базах даних застосовується модель зберігання, оптимізована під конкретні вимоги типу збережених даних. Наприклад, дані можуть зберігатися як прості пари "ключ - значення", документи JSON або граф, що складається з ребер і вершин.

Всі ці сховища даних не використовують реляційну модель. Крім того, вони, як правило, підтримують певні типи даних. Процес запиту даних також специфічний. Наприклад, сховища даних часових рядів розраховані на запити до послідовностей даних, упорядкованих за часом, а сховища даних графів - на аналіз зважених зв'язків між сутностями. Жоден з форматів не підходить в повній мірі при виконанні завдань управління даними про транзакції.

Термін NoSQL застосовується до сховищ даних, які не використовують мову запитів SQL, а запитують дані за допомогою інших мов і конструкцій. На практиці NoSQL означає "нереляційних база даних", навіть незважаючи на те, що багато хто з цих баз даних під витримують запити, сумісні з SQL. Однак базова стратегія виконання запитів SQL зазвичай значно відрізняється від застосовуваної в системі управління реляційної базою даних (реляційна СУБД).

MongoDB (від humongous) – крос-платформова документо-орієнтована система управління базами даних. Класифікована як база даних NoSQL, MongoDB відходить від традиційних основ реляційної структури бази даних на користь JSON-подібних документів з динамічними схемами (MongoDB називає цей формат BSON), що робить інтеграцію даних більш простою і швидкою.

У MongoDB застосовується безсхемна організація неструктурованих або слабоструктурованих даних, тобто організація, яка не потребує опису схеми бази даних. Цей спосіб організації даних не призначений для того, щоб пов'язувати документи з даними однієї колекції з документами, що містять дані, іншій колекції через ключові поля. У MongoDB немає власних коштів створення відносин між колекціями і документами.

MongoDB призначена для зберігання даних в документах колекцій в форматі JSON (як звичайний текст), які не мають чіткої структури та призначені для зберігання будь-якого типу даних. Такий формат і організація зберігання даних в СУБД MongoDB забезпечує оперативну обробку даних, їх запис і виведення результатів. Таким чином, MongoDB відмінно підходить для роботи з нереляційними даними.

Звідси випливає, що MongoDB добре працює з наборами даних (колекціями і документами), які не пов'язані між собою, і її можна

використовувати для створення Web додатків, що забезпечують високу продуктивність і необмежену горизонтальне масштабування колекцій даних.

2.2.4 Інструменти для створення API

Flask - фреймворк для створення веб-додатків на мові програмування Python. Відноситься до категорії так званих мікрофреймворків – мінімалістичний каркасів веб-додатків, свідомо надають лише базові функції.

Flask був розроблений так, щоб бути простим у використанні та розширенні. Ідея Flask – створити міцну основу для веб-додатків різної складності. Flask дозволяє підключити будь-які розширення, які вважаються за потрібне. Крім того, існує можливість створення власних модулів. Flask чудово підходить для всіх видів проєктів, а в особливості для прототипування. Flask залежить від двох зовнішніх бібліотек: шаблонізатора Jinja2 та набору інструментів Werkzeug WSGI.

2.2.5 Інструменти для реалізації веб-застосунку

Angular - це frontend фреймворк з відкритим кодом, розроблений компанією Google для створення динамічних та сучасних веб-додатків. Вперше представлений в 2009 році та набув за останні роки великої популярності завдяки усуненню непотрібного коду та забезпеченню створення більш легких та швидших програм.

Angular допомагає створювати інтерактивні та динамічні додатки на одній сторінці (SPA) завдяки використанню шаблонування, двостороннього прив'язування, модульності, роботи з RESTful API, реалізації ін'єкції залежностей. Однією з найбільших переваг Angular є те, що він підтримується Google, що забезпечує довгострокову підтримку (LTS).

Angular додатки побудовані за допомогою мови TypeScript, розширення для JavaScript, що забезпечує більш високу безпеку, оскільки підтримує типи

(примітиви, інтерфейси тощо). Це допомагає обробляти та усувати помилки на початку написання коду або виконання завдань з технічного обслуговування.

Які плюси можна виділити:

- підтримка Google, Microsoft;
- інструменти розробника (CLI);
- єдина структура проекту;
- TypeScript з "коробки" (можна писати строго типізований код);
- реактивне програмування з RxJS;
- єдиний фреймворк з Dependency Injection з "коробки";
- шаблони, засновані на розширенні HTML;
- кросбраузерний Shadow DOM з коробки (або його емуляція);
- кросбраузерна підтримка HTTP, WebSockets, Service Workers;
- не потрібно нічого додатково налаштовувати. Більше ніяких обгорток;
- більш сучасний фреймворк, ніж AngularJS (на рівні React, Vue);
- велике ком'юніті.

2.2.6 Методи машинного навчання

Метод k найближчих сусідів (англ.: k -nearest neighbors method, k -NN) - один з методів вирішення задачі класифікації. Передбачається, що вже є якась кількість об'єктів з точною класифікацією (тобто для кожного них точно відомо, якого класу він належить). Необхідно виробити правило, що дозволяє віднести новий об'єкт до одного з можливих класів (тобто. Самі класи відомі заздалегідь). В основі k -NN лежить наступне правило: об'єкт вважається належним того класу, до якого належить більшість його найближчих сусідів. Під «сусідами» тут розуміються об'єкти, близькі до досліджуваного в тому чи іншому сенсі.

Алгоритм КНН

1. Завантаження даних
2. Ініціалізація значення k
3. Для кожного прикладу в даних
 - 3.1 Виконати обчислення відстані між прикладом запиту та поточним прикладом даних.
 - 3.2 Додайте відстань та індекс прикладу до упорядкованої колекції
4. Відсортувати колекцію відстаней та показники від найменшої до найбільшої (у порядку зростання) за відстанями
5. Виберіть перші K записів із відсортованої колекції
6. Якщо регресія, повернути середнє значення ознак K
7. Якщо класифікація, повернути найпопулярніший клас серед K записів

Приклад (рис. 2.1). Якщо обрати $k=3$, то результатом прогнозування буде клас В, якщо обрати $k=6$, то клас А.

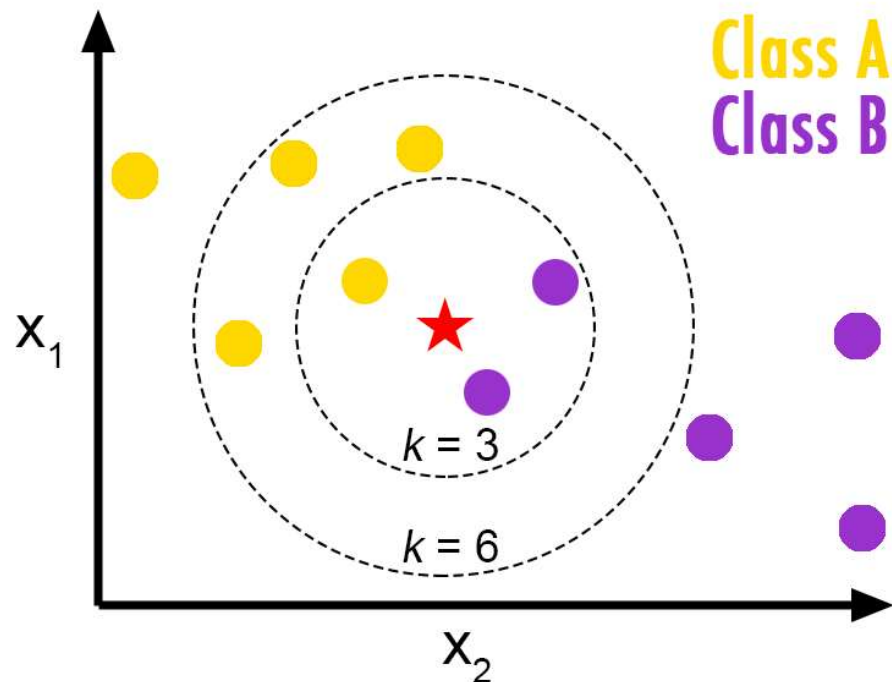


Рисунок 2.1 – Приклад використання алгоритму при $k=3$ і $k=6$

KNN на практиці

Основним недоліком KNN є те, що він стає значно повільнішим, обсяг даних збільшується, тому це робить його непрактичним вибором у середовищах, де прогнози потрібно робити швидко. Крім того, існують більш швидкі алгоритми, які дозволяють отримати більш точні результати класифікації та регресії.

Однак, якщо є достатня кількість обчислювальних ресурсів для швидкого оброблення даних, які використовуються для прогнозування, KNN все ще може бути корисним у вирішенні завдань. Прикладом цього є використання алгоритму KNN в системах рекомендацій, застосування KNN-пошуку.

Перерахуємо позитивні особливості.

- даний алгоритм вважається стійким до аномальних викидів, тому що ймовірність попадання такого запису в число k -найближчих сусідів невелика. Якщо ж це сталося, то вплив на голосування (особливо зважене) (при $k > 2$) також, швидше за все, буде незначним, і, отже, малим буде і вплив на підсумок класифікації;
- програмну реалізацію алгоритму можна вважати відносно простою;
- результат роботи алгоритму легко інтерпретувати;
- також є можливість зміни алгоритму, шляхом використання найбільш придатних функцій поєднання і метрик дозволяє підлаштувати алгоритм для конкретної задачі.

Алгоритм KNN володіє і рядом недоліків. По-перше, набір даних, який використовується для алгоритму, повинен бути репрезентативним. По-друге, модель не можна розглядати окремо від даних: щоб класифікувати новий

приклад потрібно використовувати всі приклади. Даний недолік сильно обмежує використання алгоритму

Дерева рішень (рис. 2.2).

Аналіз дерева рішень - це загальний інструмент прогнозування моделювання, який застосовується у ряді різних областей. Взагалі дерева рішень будуються за допомогою алгоритмічного підходу, який визначає способи розділення набору даних на основі різних умов. Це один з найбільш широко використовуваних і практичних методів для контролю за навчанням. Дерева рішень - це непараметричний метод навчання з учителем, який використовується як для завдань класифікації, так і для регресії. Мета - створити модель, яка прогнозує значення цільової змінної, вивчаючи прості правила прийняття рішень, виведені з особливостей даних.

Як працює алгоритм дерева рішень?

Основна ідея будь-якого алгоритму дерева рішень полягає в наступному:

- Виберіть найкращий атрибут за допомогою засобів вибору атрибутів для розділення записів.
- Зробіть цей атрибут вузлом рішення і після чого розбити набір даних на менші підмножини.
- Почати побудову дерева, повторюючи цей процес рекурсивно для кожної вершини, поки одна з умов не буде відповідати:
 - Усі кортежі належать до одного і того ж значення атрибута.
 - Більше немає атрибутів.
 - Більше випадків немає.

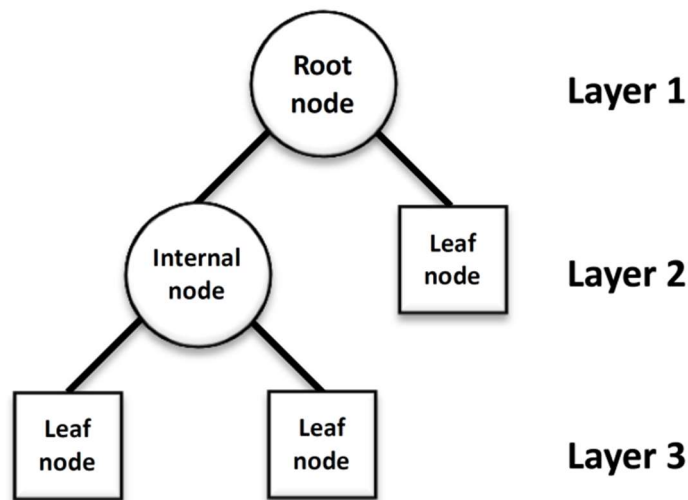


Рисунок 2.2 – Приклад дерева

Support Vector Machines

Метод опорних векторів (support vector), званий раніше алгоритмом "узагальненого портрета", був розроблений радянськими математиками В. Н. Вапніка і А. Я. Червоненкіса (1974) і з тих пір набув широкої популярності.

Основна ідея класифікатора на опорних векторах полягає в тому, щоб будувати розділяє поверхню з використанням тільки невеликого підмножини точок, що лежать в зоні, критичної для поділу, тоді як інші вірно класифікуються спостереження навчальної вибірки поза цією зоною ігноруються (точніше, є "резервуаром" для оптимізаційного алгоритму).

Якщо є два класи спостережень і передбачається лінійна форма кордону між класами, то можливі два випадки. Перший з них пов'язаний з можливістю ідеального поділу даних за допомогою деякої гіперплощини $z_k(x) = \sum_{i=1}^p \beta_i x_i + \beta_0$ (двовимірний варіант представлений зліва на рис. 2.3). Оскільки таких гіперплощин може бути безліч, то оптимальною є розділяє поверхню, яка максимально віддалена від навчальних точок, тобто має максимальний зазор M (margin).

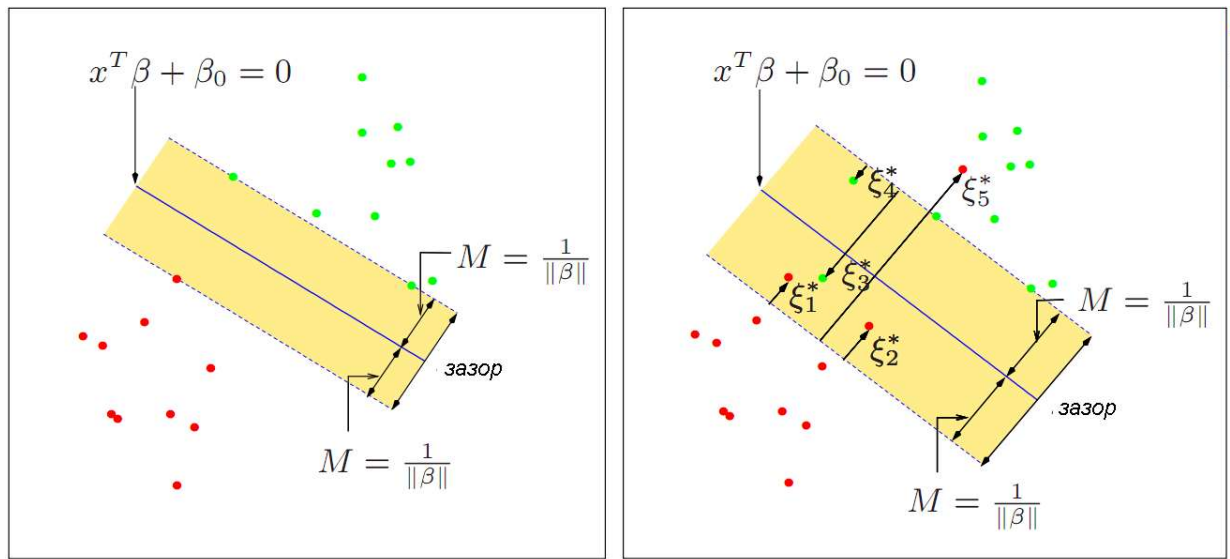


Рисунок 2.3 – Класифікатори з мінімальним зазором (зліва) і на опорних векторах (праворуч)

На рис. 2.3 праворуч показаний інший випадок, коли хмара точок перекривається і обидва класи лінійно нероздільні. Власне опорними векторами називаються спостереження, що лежать безпосередньо на кордоні розділяє смуги, або на неправильній для свого класу стороні щодо кордонів зазору (такі точки відзначені $\xi^* J$ на рис. 6.2). Для граничних і всіх інших точок приймається $\xi^* J = 0$.

Оптимальну розділяє гіперплощищу такого класифікатора $z_k(x) = \sum_{i=1}^p \beta_i x_i + \beta_0$ також знаходять з умови максимізації ширини зазору M , але при цьому дозволяється невірно класифікувати деяку невелику групу спостережень, що відносяться до опорних векторах. Для цього задаються додатковою умовою оптимізації $\sum_j \xi_j^* \leq C$, де C - допустима кількість порушень кордону зазору і їх вираженість, яке зазвичай вибирається з використанням перехресної перевірки. Математично пошук рішення зводиться до зручної задачі квадратичної оптимізації з лінійними обмеженнями, яка гарантовано сходиться до одного глобального мінімуму.

Оскільки на розташування гіперплощини впливають тільки ті спостереження, які лежать на кордонах зазору або порушують його, то вирішальне правило такого класифікатора досить стійко до викидів більшості точок, розташованих поза "критичної зони" поділу. Це властивість відрізняє його від властивостей інших класифікаторів. Наприклад, що розділяє площину LDA на рис. 6.2 проводиться перпендикулярно дискримінаційної функції z і залежить від середніх і коваріаційної матриці, що обчислюються за всіма наявними спостереженнями.

3 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Структура інформаційної системи

Перед початком реалізації інформаційної системи необхідно створити її базову архітектуру. Оскільки в наявності є вже створена інформаційна система, то необхідно додати архітектурні об'єкти, що будуть відповідати за прогнозування і зв'язок з базою даних та веб застосунком. Базова архітектура зображена на рисунку 3.1, та включає в себе 2 додаткових модулі – модуль прогнозування та модуль API, який відповідатиме за зв'язок між веб застосунком на модулем прогнозування.

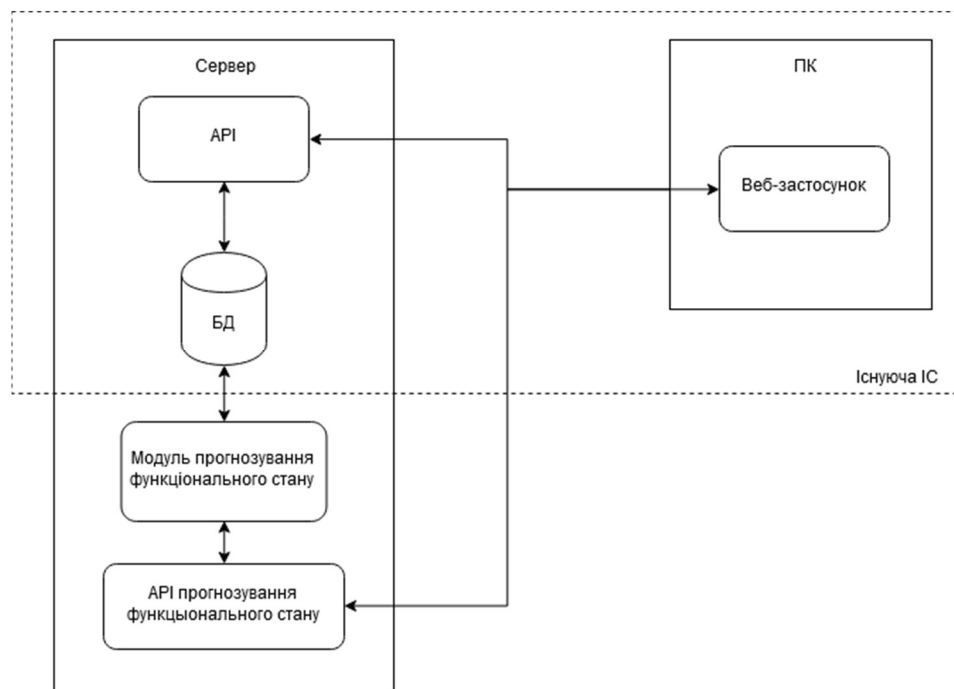


Рисунок 3.1 – Доповнена архітектура системи

3.2 Моделювання інформаційної системи

Важливим кроком є моделювання роботи інформаційної системи, адже саме це дає змогу відслідкувати усі процеси, що будуть протікати всередині. Для цього потрібно побудувати діаграму IDEF0 (рис. 3.2), а після чого провести її декомпозицію (рис. 3.3).

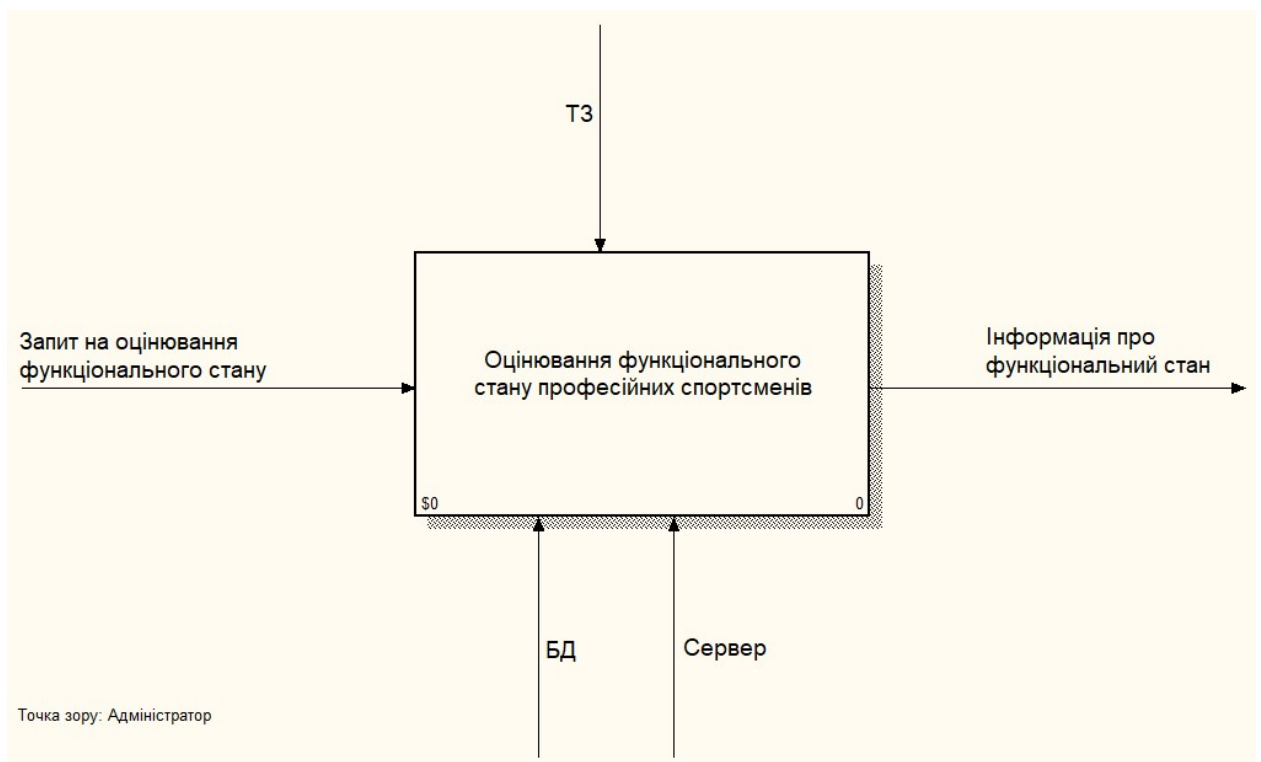


Рисунок 3.2 – Контекстна діаграма інформаційної систем

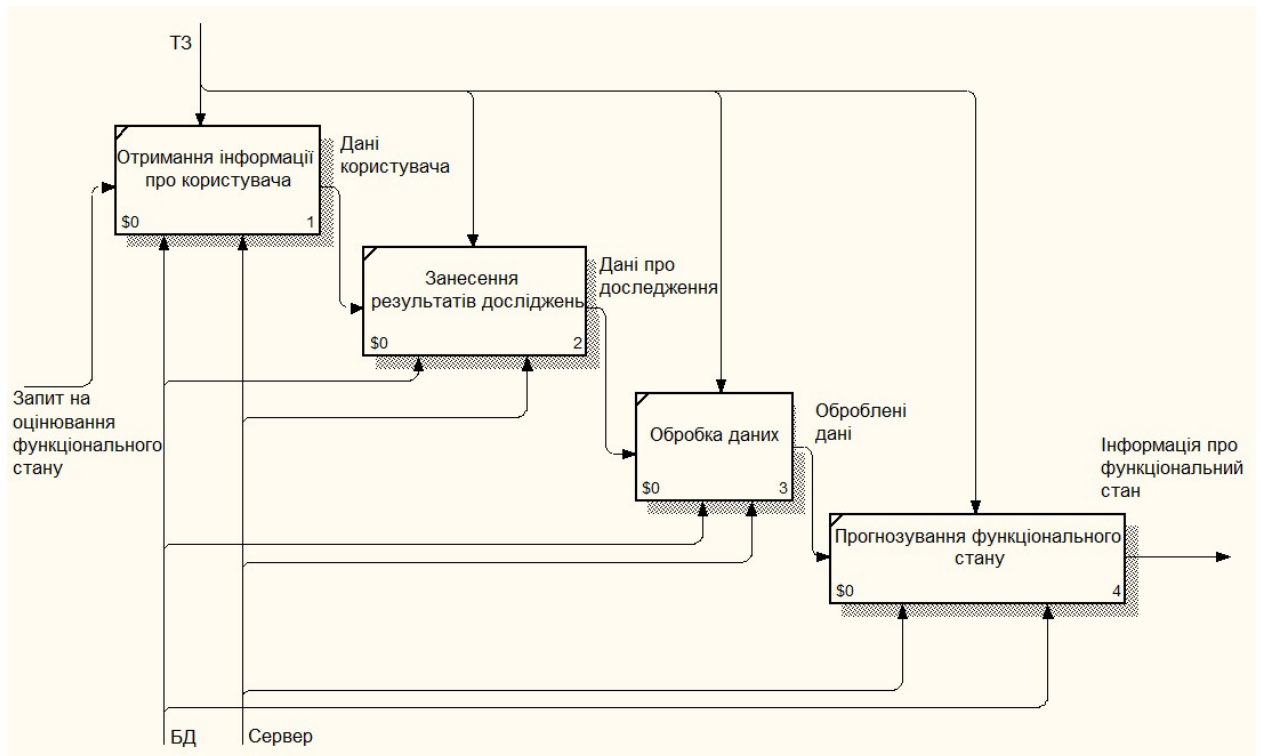


Рисунок 3.3 – Процеси інформаційної системи

Також важливим етапом є створення діаграми використання (рис. 3.4).

Інформаційна система матиме 6 акторів – база даних, модуль прогнозування, адміністратор, користувач, гість та редактор. Кожен актор матиме певний спектр дозволених дій з інформаційною системою.

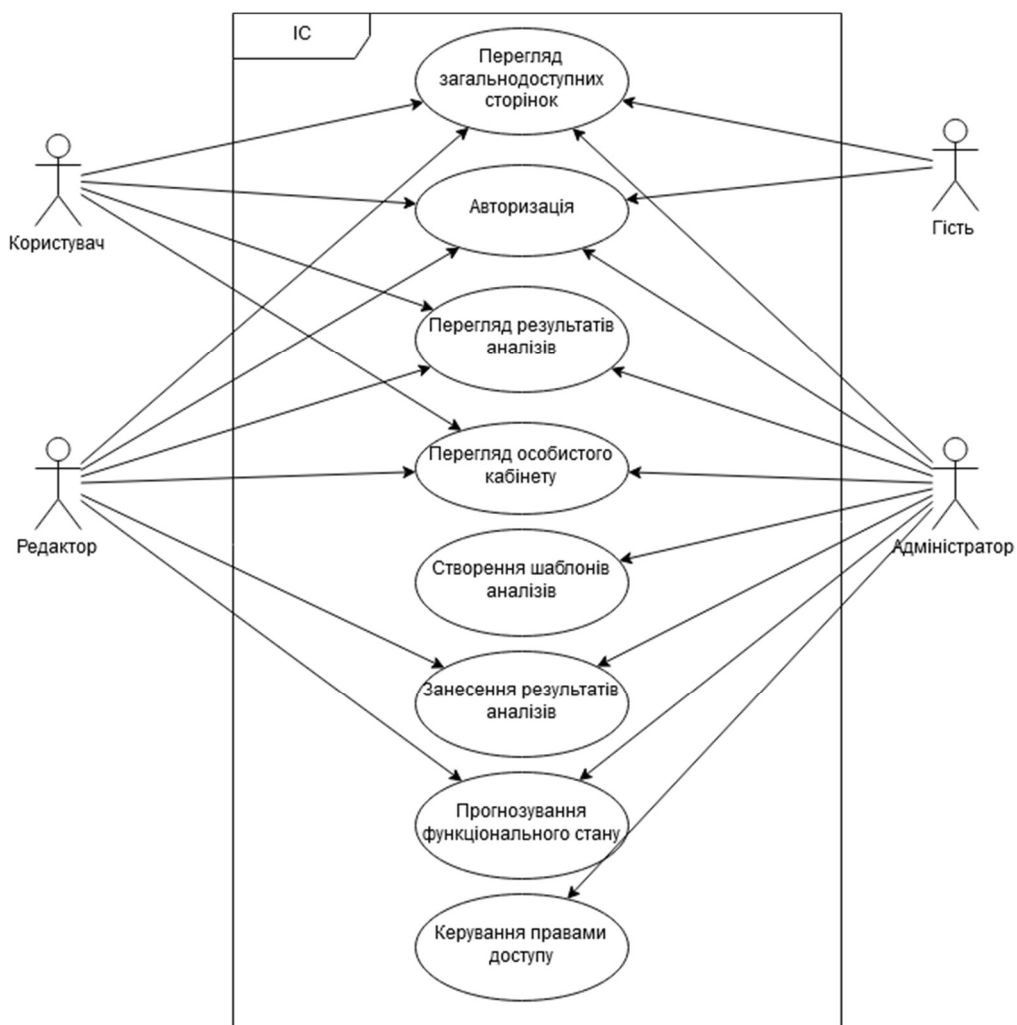


Рисунок 3.4 – Діаграма варіантів використання

3.3 Проектування бази даних

Для проектування бази даних було використано ER (Entity Relationship) діаграму. Для початку було виділено основні сутності:

- user
- analysisdefinition
- analysis
- field

Після чого для кожної сутності було створено їх атрибути

Наступним кроком було виділення зв'язків між сутностями. Результатом даного моделювання є діаграма зображена на рисунку 3.5.

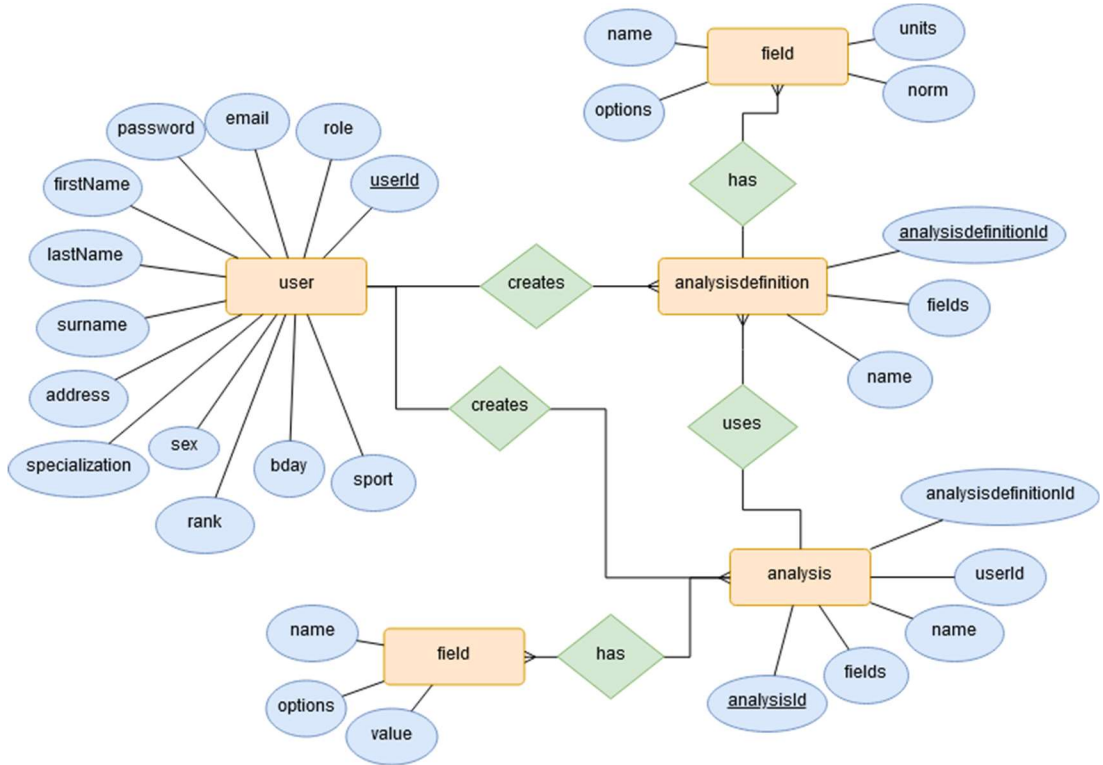


Рисунок 3.5 – ER діаграма

Для того, щоб зберігати об'єкти в базі даних було створено їх моделі (табл. 3.1-3.5).

Таблиця 3.1 – Модель User

Назва	Опис
email	Адреса електронної пошти
firstName	Ім'я
lastName	Прізвище
password	Пароль
role	Права доступу

Продовження таблиці 3.1

address	Адреса проживання
bday	Дата народження
rank	Розряд
sex	Стать
specialization	Спеціалізація
sport	Вид спорту
surname	По-батькові

Таблиця 3.2 – Модель AnalysisDefinition

Назва	Опис
name	Назва
fields	Масив показників
name	Назва
options	Перелік якісних варіантів
norm	Норма
units	Одиниці вимірювання

Таблиця 3.3 – Модель Analysis

Назва	Опис
name	Ім'я
created	Дата створення
_userId	Ід користувача
fields	Масив показників
name	Назва
value	Значення

Продовження таблиці 3.3

options	Перелік якісних варіантів
norm	Норма
units	Одиниці вимірювання

Таблиця 3.4 – Модель ProfileRequest

Назва	Опис
address	Адреса проживання
bday	Дата народження
firstName	Ім'я
lastName	Прізвище
rank	Розряд
sex	Стать
specialization	Спеціалізація
sport	Вид спорту
surname	По-батькові

Таблиця 3.5 – Інформація про дослідження функціонального стану

Назва	Опис
age	Вік
gender	Стать
height	Зріст
weight	Вага
systolic	Систолічний артеріальний тиск
diastolic	Діастолічний артеріальний тиск

Продовження таблиці 3.5

cholesterol	Рівень холестеролу (нормальний, вище норми, значно вище норми)
glucose	Рівень глюкози (нормальний, вище норми, значно вище норми)
smoke	Курить чи ні
alcohol	Вживає алкоголь чи ні
active	Фізично активний чи ні

Далі було створено моделі використовуючи бібліотеку mongoose, які забезпечують зв'язок між базою даних та методами API (рис. 3.6).

```

1  var mongoose = require('mongoose');
2  var CardioRecordSchema = new mongoose.Schema({
3    age_years: Number,
4    gender: Number,
5    height: Number,
6    weight: Number,
7    systolic: Number,
8    diastolic: Number,
9    cholesterol: Number,
10   glucose: Number,
11   smoke: Number,
12   alcohol: Number,
13   active: Number
14 });
15 mongoose.model('CardioRecord', CardioRecordSchema);
16
17 const CardioRecord = module.exports = mongoose.model('CardioRecord');
18
19 module.exports.findAll = (callback) => {
20   |   CardioRecord.find({}, callback);
21 }

```

Рисунок 3.6 – Приклад коду для створення mongoose схеми

Після чого було використано отримані схеми для створення відповідних контролерів які мають свої унікальні маршрути (рис. 3.7) та реалізують різні HTTP методи (рис. 3.8).

```

16 var TestController = require('./controllers/TestController');
17 var AuthController = require('./controllers/AuthController');
18 var UserController = require('./controllers/UserController');
19 var AnalysisDefinitionController = require('./controllers/AnalysisDefinitionController');
20 var AnalysisController = require('./controllers/AnalysisController');
21 var ProfileRequestController = require('./controllers/ProfileRequestController');
22 | var CardioController = require('./controllers/CardioController');
23
24 app.use('/test', TestController);
25 app.use('/api/auth', AuthController);
26 app.use('/api/users', UserController);
27 app.use('/api/analysisDefinitions', AnalysisDefinitionController);
28 app.use('/api/analyzes', AnalysisController);
29 app.use('/api/profileRequests', ProfileRequestController);
30 | app.use('/api/cardio', CardioController);

```

Рисунок 3.7 – Приклад коду для визначення маршрутів контролерів

```

11 | router.post('/', VerifyToken, (req, res) => {
17 | });
18
19 | router.get('/', VerifyToken, (req, res) => {
24 | });
25
26 | router.get('/:id', VerifyToken, (req, res) => {
31 | });
32
33 | router.delete('/:id', VerifyToken, (req, res) => {
39 | });
40
41 | router.put('/:id', VerifyToken, (req, res) => {
46 | });
47 |

```

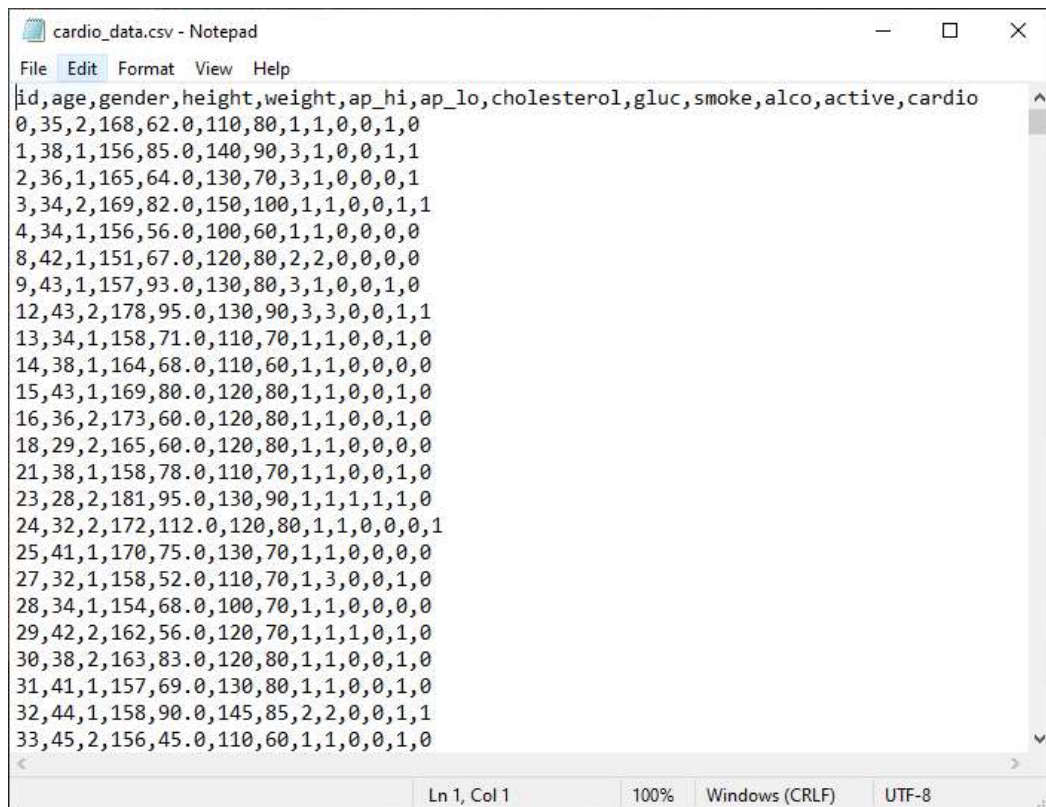
Рисунок 3.8 – Приклад реалізації маршрутів

3.4 Збір даних

Дані для прогнозування були взяті з інформаційної системи обліку показників стану здоров'я професійних спортсменів та відкритих джерел. Дані включають наступні показники:

- Вік;
- стать;
- зріст;
- вага;
- систолічний артеріальний тиск;
- діастолічний артеріальний тиск;
- холестерол (нормальний, вище норми, значно вище норми);
- рівень глюкози (нормальний, вище норми, значно вище норми);
- курить чи ні;
- вживає алкоголь чи ні;
- фізично активний чи ні;
- наявність або відсутність серцево-судинних захворювань – цільова змінна.

Усі дані були зібрані у файл з розширенням CSV. Він зображений на рисунку 3.9.



```
cardio_data.csv - Notepad
File Edit Format View Help
id,age,gender,height,weight,ap_hi,ap_lo,cholesterol,gluc,smoke,alco,active,cardio
0,35,2,168,62.0,110,80,1,1,0,0,1,0
1,38,1,156,85.0,140,90,3,1,0,0,1,1
2,36,1,165,64.0,130,70,3,1,0,0,0,1
3,34,2,169,82.0,150,100,1,1,0,0,1,1
4,34,1,156,56.0,100,60,1,1,0,0,0,0
8,42,1,151,67.0,120,80,2,2,0,0,0,0
9,43,1,157,93.0,130,80,3,1,0,0,1,0
12,43,2,178,95.0,130,90,3,3,0,0,1,1
13,34,1,158,71.0,110,70,1,1,0,0,1,0
14,38,1,164,68.0,110,60,1,1,0,0,0,0
15,43,1,169,80.0,120,80,1,1,0,0,1,0
16,36,2,173,60.0,120,80,1,1,0,0,1,0
18,29,2,165,60.0,120,80,1,1,0,0,0,0
21,38,1,158,78.0,110,70,1,1,0,0,1,0
23,28,2,181,95.0,130,90,1,1,1,1,1,0
24,32,2,172,112.0,120,80,1,1,0,0,0,1
25,41,1,170,75.0,130,70,1,1,0,0,0,0
27,32,1,158,52.0,110,70,1,3,0,0,1,0
28,34,1,154,68.0,100,70,1,1,0,0,0,0
29,42,2,162,56.0,120,70,1,1,1,0,1,0
30,38,2,163,83.0,120,80,1,1,0,0,1,0
31,41,1,157,69.0,130,80,1,1,0,0,1,0
32,44,1,158,90.0,145,85,2,2,0,0,1,1
33,45,2,156,45.0,110,60,1,1,0,0,1,0
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Рисунок 3.9 – Приклад збереження інформації у файлі CSV

3.5 Робота з даними. Розвідувальний аналіз даних

У статистиці розвідувальний аналіз даних (EDA) - це підхід до аналізу наборів даних для узагальнення їх основних характеристик, часто за допомогою візуальних методів. Статистична модель може використовуватися чи ні, але в першу чергу EDA призначена для того, щоб побачити, що дані можуть сказати нам, крім формальної задачі моделювання або перевірки гіпотез. Джон Тьюкі просував дослідний аналіз даних, щоб спонукати статистиків досліджувати дані і, можливо, сформулювати гіпотези, які могли б привести до збору нових даних і експериментів. EDA відрізняється від аналізу початкових даних (IDA), який вужче фокусується на перевірці припущень, необхідних для підбору моделі і перевірки гіпотез, а також обробки пропущених значень і перетворення змінних у міру необхідності. EDA включає в себе IDA.

Термін «розвідувальний аналіз» був вперше введений математиком з Принстонського університету Дж.Тьюкі. Він також сформулював основні цілі даного аналізу:

- Максимальна «проникнення» в дані.
- Виявлення основних структур.
- Вибір найбільш важливих змінних.
- Виявлення відхилень та аномалій.
- Перевірка основних гіпотез (припущень).
- Розробка початкових моделей.

До основних методів розвідувального аналізу відноситься процедура аналізу розподілів змінних, кореляційний аналіз с метою пошуку коефіцієнтів,

що перевершують по величині певні порогові значення, факторний аналіз, дискримінантний аналіз, багатовимірне шкалювання, візуальний аналіз гістограм і т.д.

Проведемо розвідувальний аналіз даних для зібраного датасету. Для початку виведемо загальну інформацію про датасет (рис. 3.10).

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70000 entries, 0 to 69999
Data columns (total 13 columns):
id                70000 non-null int64
age               70000 non-null int64
gender            70000 non-null int64
height            70000 non-null int64
weight            70000 non-null float64
ap_hi             70000 non-null int64
ap_lo             70000 non-null int64
cholesterol       70000 non-null int64
gluc              70000 non-null int64
smoke             70000 non-null int64
alco              70000 non-null int64
active            70000 non-null int64
cardio            70000 non-null int64
dtypes: float64(1), int64(12)
memory usage: 6.9 MB
```

Рисунок 3.10 – Загальна інформація про датасет

Виведемо загальну інформацію про показники (рис. 3.11).

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
id	70000.0	49972.419900	28851.302323	0.0	25006.75	50001.5	74889.25	99999.0
age	70000.0	37.318229	4.772436	21.0	34.00	38.0	41.00	46.0
gender	70000.0	1.349571	0.476838	1.0	1.00	1.0	2.00	2.0
height	70000.0	164.359229	8.210126	55.0	159.00	165.0	170.00	250.0
weight	70000.0	74.205690	14.395757	10.0	65.00	72.0	82.00	200.0
ap_hi	70000.0	128.817286	154.011419	-150.0	120.00	120.0	140.00	16020.0
ap_lo	70000.0	96.630414	188.472530	-70.0	80.00	80.0	90.00	11000.0
cholesterol	70000.0	1.366871	0.680250	1.0	1.00	1.0	2.00	3.0
gluc	70000.0	1.226457	0.572270	1.0	1.00	1.0	1.00	3.0
smoke	70000.0	0.088129	0.283484	0.0	0.00	0.0	0.00	1.0
alco	70000.0	0.053771	0.225568	0.0	0.00	0.0	0.00	1.0
active	70000.0	0.803729	0.397179	0.0	1.00	1.0	1.00	1.0
cardio	70000.0	0.499700	0.500003	0.0	0.00	0.0	1.00	1.0

Рисунок 3.11 – Інформація про показники

Підрахуємо кількість даних для різних класів цільової змінної та виведемо їх співвідношення (рис. 3.12).

```
df['cardio'].value_counts()
```

```
0    35021
1    34979
Name: cardio, dtype: int64
```

```
df['cardio'].value_counts(normalize=True)
```

```
0    0.5003
1    0.4997
Name: cardio, dtype: float64
```

Рисунок 3.12 – Співвідношення значень цільової змінної

Бачимо, що класи майже збалансовані, тому нормалізація непотрібна.

Проведемо аналіз на наявність викидів. Побудуємо діаграму розмаху («ящик з вусами») для наступних показників: зріст (рис. 3.13), вага(рис. 3.14), систолічний тиск (рис. 3.15), діастолічний тиск (рис. 3.16).

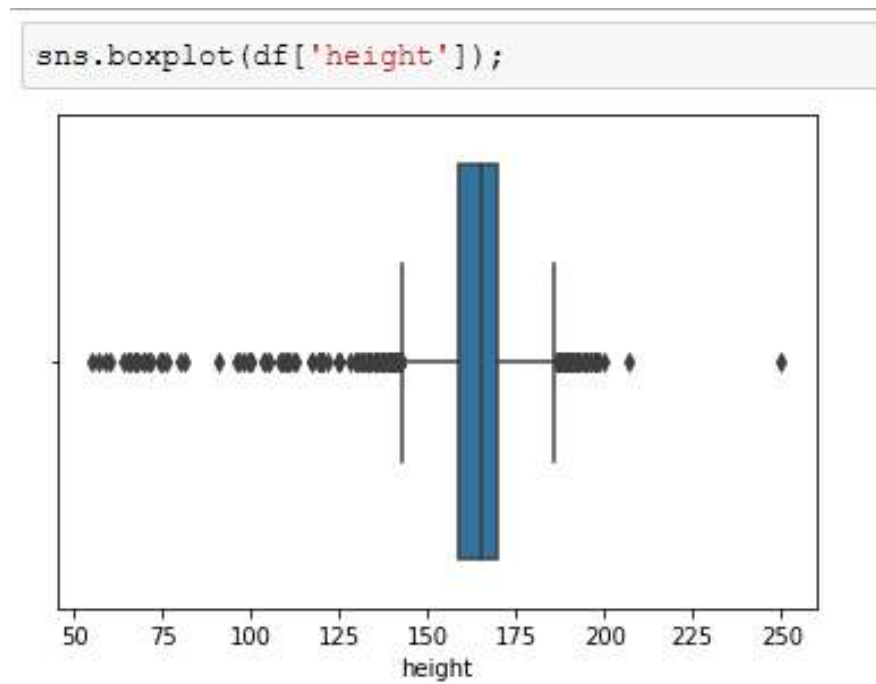


Рисунок 3.13 – Діаграма розмаху для показника зросту

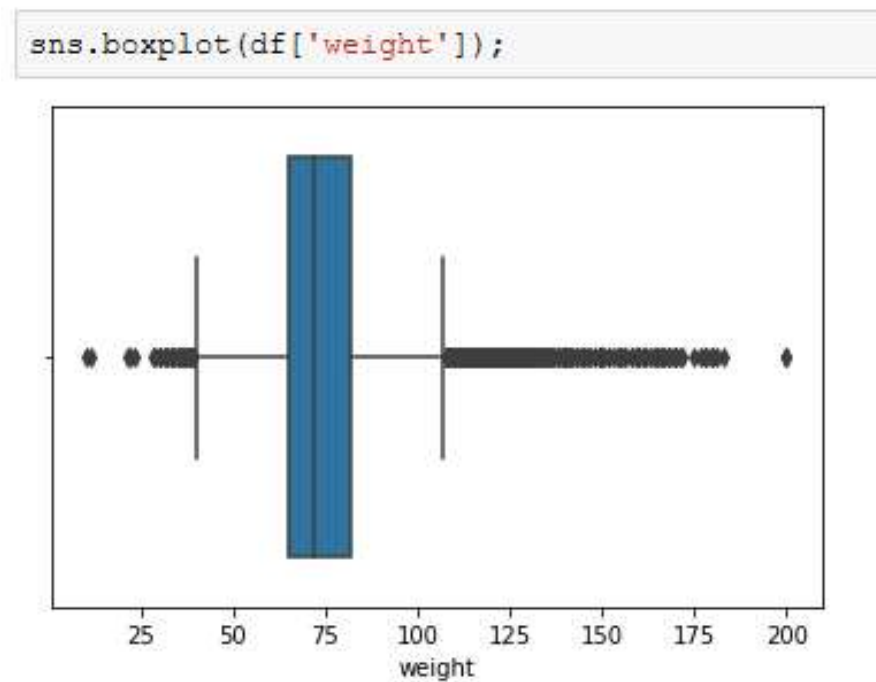


Рисунок 3.14 – Діаграма розмаху для показника ваги

```
sns.boxplot(df['ap_hi']);
```

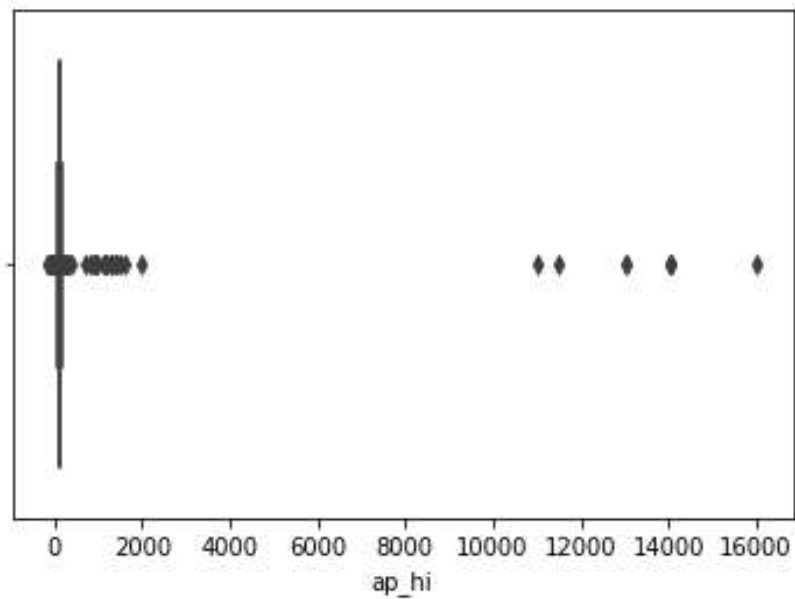


Рисунок 3.15 – Діаграма розмаху для показника систолічного тиску

```
sns.boxplot(df['ap_lo']);
```

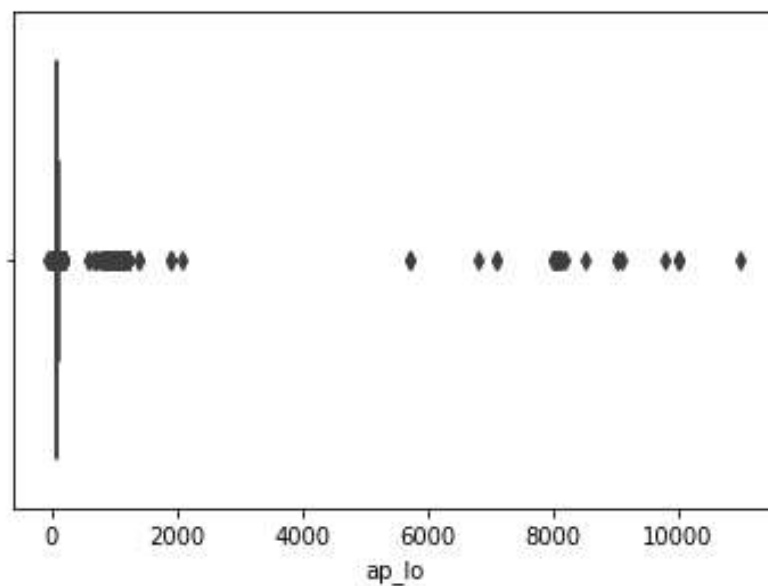


Рисунок 3.16 – Діаграма розмаху для показника діастолічного

Судячи з графіків видно, що в датасеті існують викиди. Для того, щоб їх позбавитись можна використати правило трьох сигм, або видалити записи

$$[\bar{x} - 3\sigma; \bar{x} + 3\sigma]$$

вручну. Правило трьох сигм говорить, що майже всі нормально розподілені дані випадкової величини лежать в інтервалі з ймовірністю не менше 99,7 %.

Після видалення записів, використовуючи правило трьох сигм, розмір датасету зменшився на 2.8%. Побудуємо знову діаграми розмаху для перевірки (рис. 3.17-3.18).

```
sns.boxplot(df_wo['systolic']);
```

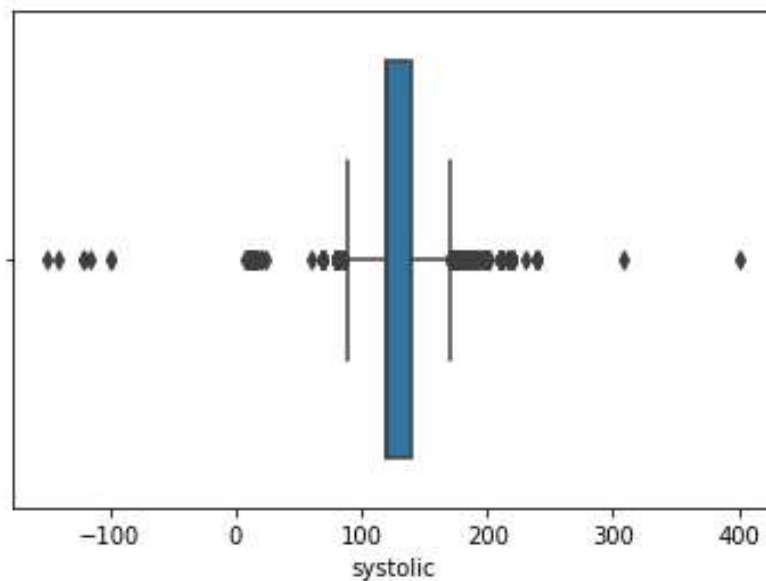


Рисунок 3.17 – Діаграма розмаху для показника систолічного тиску після видалення викидів використовуючи метод трьох сигм

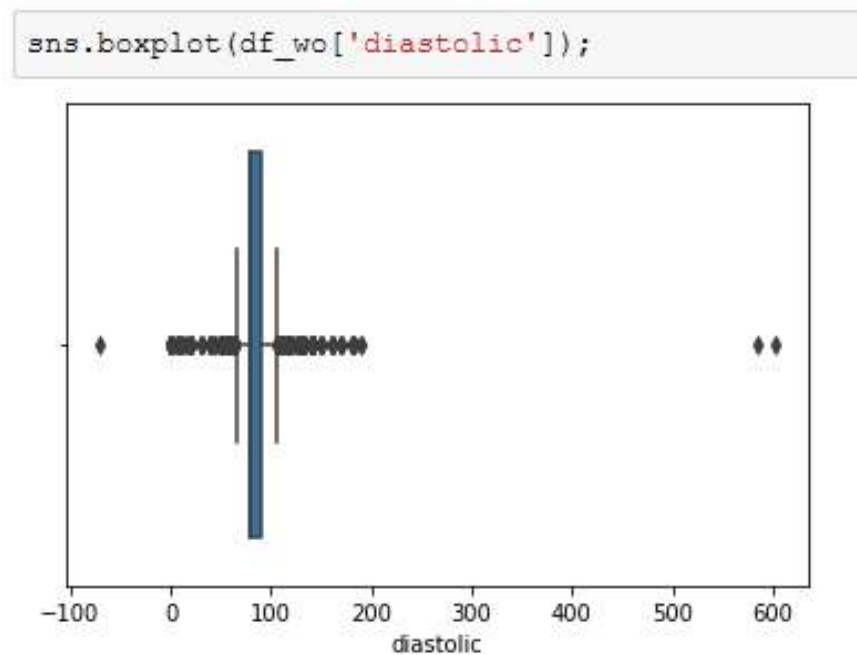


Рисунок 3.18 – Діаграма розмаху для показника діастолічного тиску після видалення викидів використовуючи метод трьох сигм

Як бачимо з рисунків викиди все ще залишились. Використаємо функцію `drop` для видалення викидів (рис. 3.19).

```
df_wo2 = df_wo.drop(df_wo[(df_wo['systolic'] < 0) | (df_wo['systolic'] > 300)].index)
df_wo2 = df_wo2.drop(df_wo2[(df_wo2['diastolic'] < 0) | (df_wo2['diastolic'] > 300)].index)
df_wo2.shape[0] / df.shape[0]
0.9722714285714286
```

Рисунок 3.19 – приклад використання функції `drop`

Після видалення викидів розмір датасету зменшився на 2.8% відносно оригінального розміру.

Наступним етапом було створення нових ознак на основі існуючих. Було створено наступні нові ознаки:

`high_SBP` – приймає значення `true` якщо систолічний тиск вище 140

`high_DBP` – приймає значення `true` якщо діастолічний тиск вище 90

`bmi` – індекс маси тіла, що обчислюється за формулою $\frac{\text{вага}}{\text{зріст}^2}$

map – середній артеріальний тиск $\left(\frac{\text{систоличний тиск} + 2 * \text{діастолічний тиск}}{3}\right)$

Останнім етапом роботи з даними є вибір найважливіших показників. Для цього можна використати один із методів оцінки важливості показників – SelectKBest (рис. 3.20).

```
# separate independent & dependent variables
X = df[num_features] #independent columns
y = df[target]      #target column i.e price range

# apply SelectKBest class to extract top 10 best features
bestfeatures = SelectKBest(score_func=chi2, k=13)
fit = bestfeatures.fit(X,y)

dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
#concat two dataframes for better visualization
featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Specs','Score'] #naming the dataframe columns
print(featureScores.nlargest(8,'Score')) #print 8 best features
```

	Specs	Score
2	systolic	26995.858307
12	map	14390.159246
3	diastolic	8732.201102
1	weight	5161.861288
11	waist_cm	4127.945178
9	age_years	3415.309481
10	bmi	2162.590800
4	cholesterol	1119.939055

Рисунок 3.20 – Приклад оцінки ознак

Список найважливіших ознак наведено в таблиці 3.6

Таблиця 3.6 – Список найважливіших ознак

Ознака	Кількість набраних балів
systolic	26995.858307
map	14390.159246
diastolic	8732.201102
weight	5161.861288
waist_cm	4127.945178
age_years	3415.309481
bmi	2162.590800
cholesterol	1119.939055

3.6 Реалізація моделей та їх порівняння

Для реалізації моделей машинного навчання було використано наступні класи Scikit-learn:

- SVC – для методу опорних векторів;
- GridSearchCV і KNeighborsClassifier – для методу k найближчих сусідів;
- DecisionTreeClassifier – для методу дерева рішень.

Для того, щоб обрати найкращу модель необхідно використати методи оцінки моделей машинного навчання. Одним із таких методів є метод перехресної перевірки.

В його основі лежить поділ вихідного безлічі даних на k приблизно рівних блоків, наприклад $k = 5$. Потім на $k-1$, тобто на 4-х блоках, проводиться навчання моделі, а 5-й блок використовується для тестування. Процедура повторюється k разів, при цьому на кожному проході для перевірки вибирається новий блок, а навчання проводиться на що залишилися.

Перехресна перевірка має дві основні переваги перед застосуванням одного безлічі для навчання і одного для тестування моделі:

- Розподіл класів виявляється більш рівномірним, що покращує якість навчання.
- Якщо при кожному проході оцінити вихідну помилку моделі і усереднити її за всіма проходах, то отримана оцінка буде більш достовірною.

На практиці найчастіше вибирається $k = 10$ (10-ти прохідна перехресна перевірка), коли модель навчається на 9/10 даних і тестується на 1/10.

Дослідження показали, що в цьому випадку виходить найбільш достовірною оцінка вихідних помилок моделі.

Для того, щоб розділити датасет на дві частини – одна для навчання, а інша для перевірки, було використано метод із бібліотеки Scikit-learn – `train_test_split`, який дозволяє розділити датасет на тренувальну і тестову вибірки у заданому співвідношенні та у випадковому порядку.

Отримані вибірки було використано для навчання та перевірки моделей за допомогою методу перехресної перевірки. Отримано наступні результати (табл. 3.7)

Таблиця 3.7 – Точність навчених моделей

Модель	Результат
SVC	~75%
KNeighborsClassifier	~65.5%
DecisionTreeClassifier	~64.5%

3.7 Створення API для обраної моделі та інтеграція в ІС

Для того, щоб можна було використовувати створену модель в комбінації з інформаційною системою необхідно реалізувати API для взаємодії.

Першим кроком є збереження створеної моделі у вигляді окремого файлу. Це дасть змогу повторно використовувати модель без необхідності її навчання. Для цього було використано функцію `joblib.dump` із бібліотеки `Scikit-learn`. Для завантаження файлу моделі використано функцію `joblib.load`.

Далі було використано бібліотеку `Flask` для реалізації методів API для модуля прогнозування. Приклад роботи зображено на рисунку 3.21.

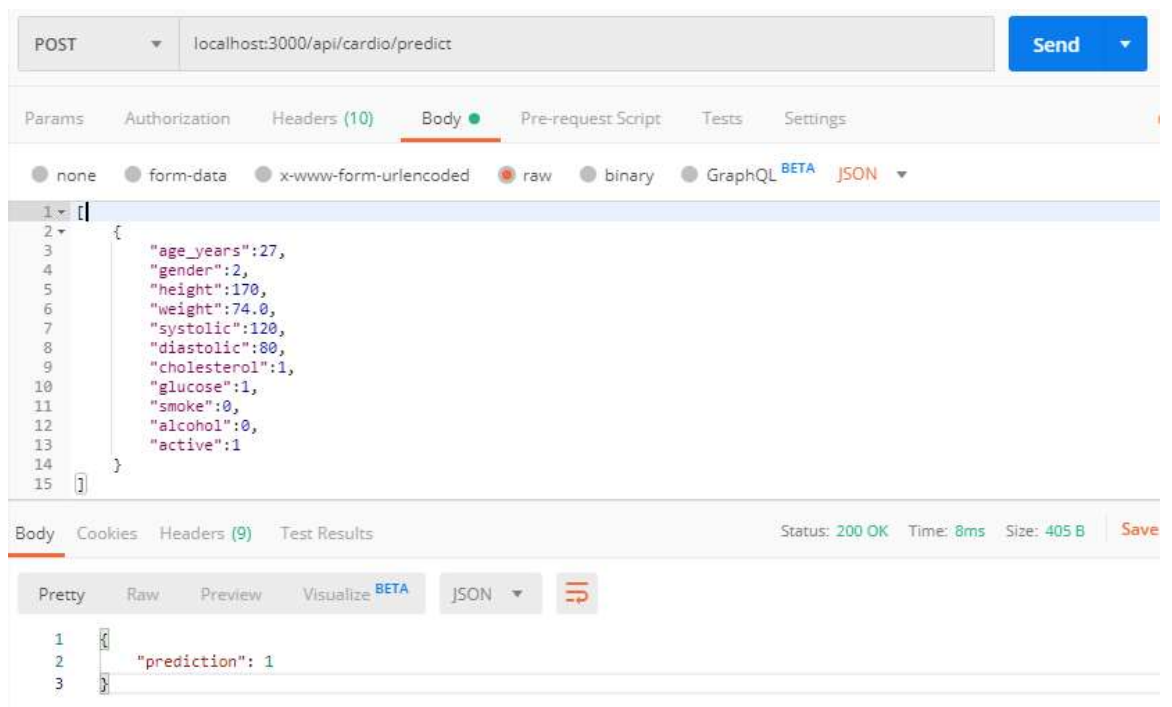


Рисунок 3.21 – Приклад виконання запиту

4 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОЦІНЮВАННЯ ФУНКЦІОНАЛЬНОГО СТАНУ ПРОФЕСІЙНИХ СПОРТСМЕНІВ

Результатом практичного застосування описаних моделей та методів є інформаційна система для оцінювання функціонального стану спортсменів. Для демонстрації роботи розробленого модуля прогнозування використаємо програму Postman. Приклади запитів наведені на рисунках 4.1-4.3.

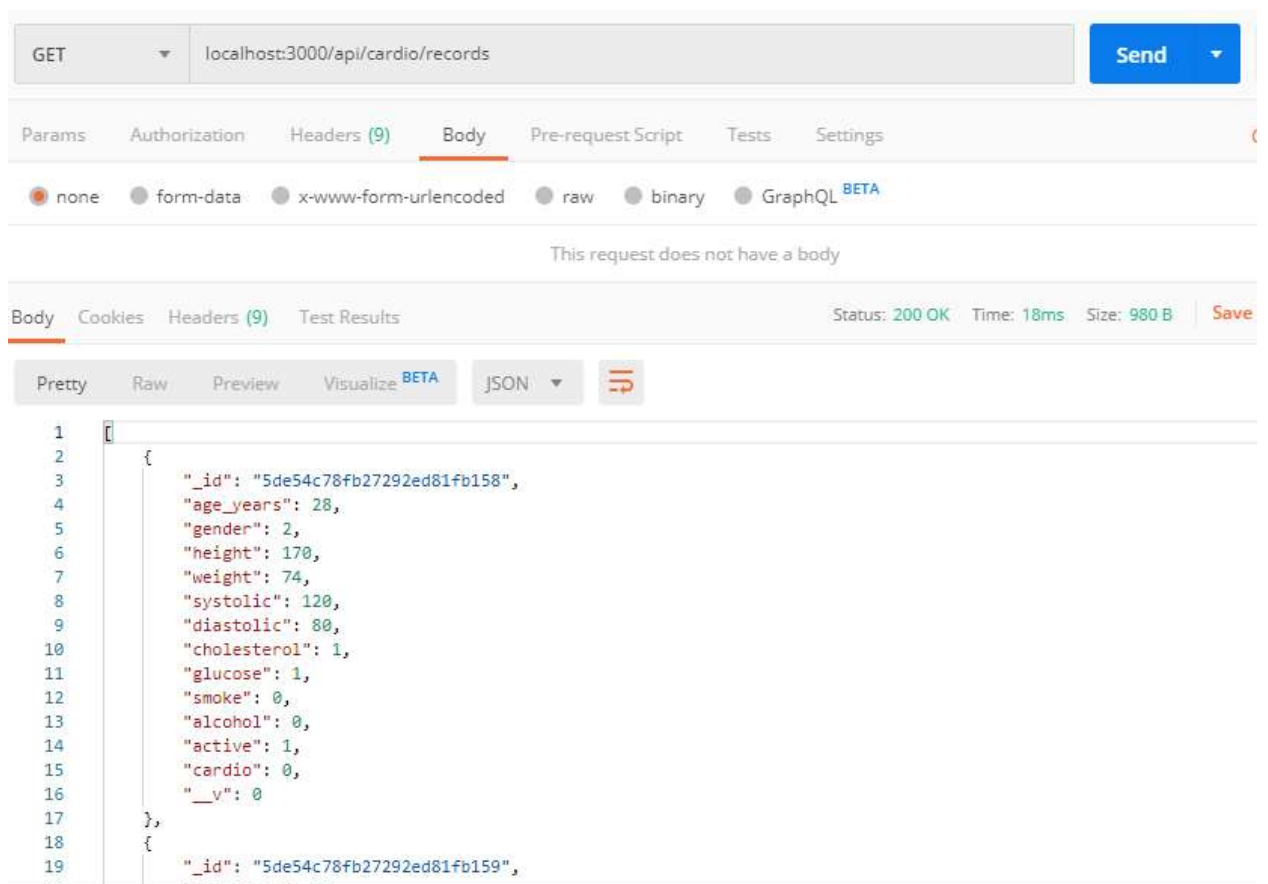


Рисунок 4.1 – Приклад запиту для отримання усіх записів

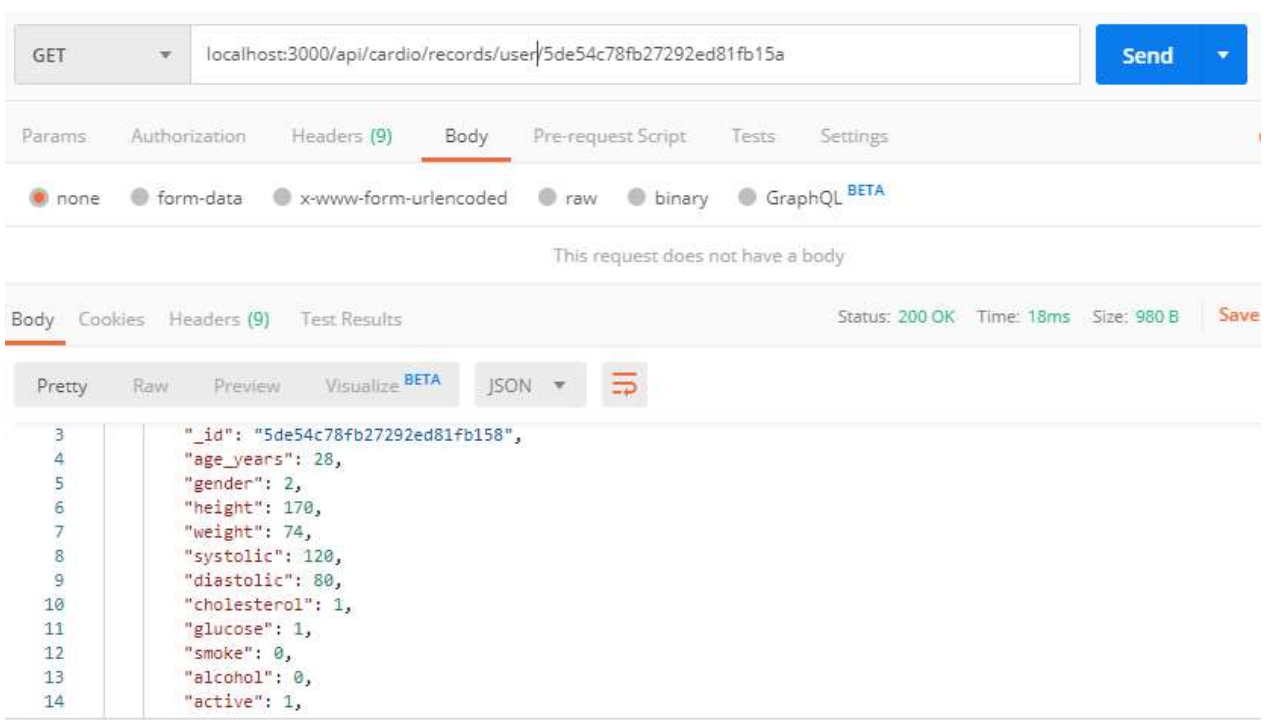


Рисунок 4.2 – Приклад запиту для отримання записів окремого користувача

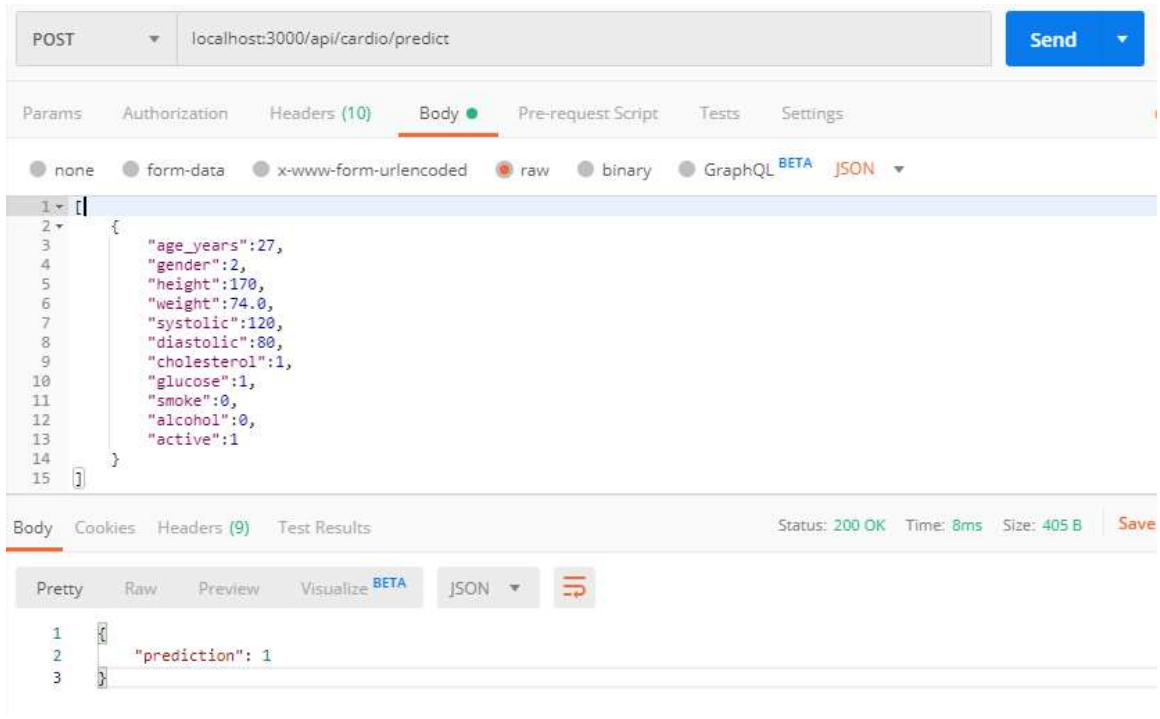


Рисунок 4.3 – Приклад запиту для отримання прогнозу функціонального стану

Приклади веб-застосунку наведені на рисунках 4.4-4.7 .

age_years	gender	height	weight	systolic	diastolic	cholesterol	glucose	smoke	alcohol	active	
28	2	170	74	120	80	1	1	0	0	1	View
35	2	165	86	130	80	2	2	0	0	1	View
26	1	158	54	100	60	3	3	0	0	1	View
31	2	176	71	125	82	1	1	0	0	1	View
35	2	165	86	130	80	2	2	0	0	1	View
26	1	158	54	100	60	3	3	0	0	1	View

Рисунок 4.4 – Список усіх записів для прогнозу

parameter	value
age_years	28
gender	2
height	170
weight	74
systolic	120
diastolic	80
cholesterol	1
glucose	1
smoke	0
alcohol	0
active	1

[Predict](#)

Виявлена ймовірність виникнення серцевих захворювань

Рисунок 4.5 – Перегляд запису та результат прогнозування

Sport-id Home ML CardioRecords Конструктор Типи аналізів Користувачі Запити Профіль admin

Вік
25

Стать
1

Зріст
173

Вага
72

Систолічний тиск
120

Діастолічний тиск
80

Холестирол (категорія)
1

Глюкоза (категорія)
1

Курить

Вживає алкоголь

Фізично активний

Add

Рисунок 4.6 – Приклад занесення даних про дослідження в інформаційну систему

Sport-id Home ML CardioRecords Конструктор Типи аналізів Користувачі Запити Профіль admin

Пошук + Додати

Прізвище	Ім'я	По-батькові	Додати аналіз	Профіль користувача
admin	admin	admin	+ Додати	Переглянути
Лочман	Анісія	В'ячеславівна	+ Додати	Переглянути
Шкурат	Оксана	Юрївна	+ Додати	Переглянути
Сеник	Станіслав	Анатолійович	+ Додати	Переглянути
Палажченко	Євген	Володимирович	+ Додати	Переглянути
Супрун	Володимир	Олегович	+ Додати	Переглянути
Сторож	Юрій	В'ячеславович	+ Додати	Переглянути
Бутрим	Віталій	Миколайович	+ Додати	Переглянути
Кримаренко	Юрій	Олександрович	+ Додати	Переглянути

Рисунок 4.7 – Перегляд списку користувачів системи

ВИСНОВКИ

Ефективне використання інформаційних систем стає запорукою забезпечення якості медицини. Все частіше різні медичні заклади користуються перевагами інформаційних систем, адже вони надають можливість обробки та аналізу великих обсягів даних.

Однією з найпопулярніших і актуальних тем різних досліджень є використання систем машинного навчання. Тому, машинне навчання все частіше зустрічається у повсякденному житті. Можна зробити висновок, що дані методи і технології будуть розвиватися і далі, кожного разу все більше інтегруючись у повсякденне життя.

Після аналізу існуючих програмних продуктів та з'ясовано, що жодна з них не задовольняє заданим вимогам. Тому на підставі цього було вирішено створити власну інформаційну систему. Для оцінки функціонального стану професійних спортсменів було вирішено використати методи машинного навчання.

Після цього було сформульовано основні методи та задачі дослідження. Також було розглянуто низку інструментів для реалізації поставленої задачі.

Наступним етапом було моделювання інформаційної системи. Для початку було створено загальну модель інформаційної системи та бази даних. Оскільки моделі машинного навчання можуть бути чутливими до даних, було проведено аналіз даних та виявлено і видалено викиди. Також важливим кроком була нормалізація даних.

Після закінчення розвідувального аналізу даних було створено декілька моделей машинного навчання, виконано їх навчання, та за допомогою різних метрик було обрано найкращу модель – метод опорних векторів, що показав точність прогнозу близько 75%.

Отриману модель було використано для створення незалежного модуля прогнозування функціонального стану, для цього вона була збережена в окремий файл з можливістю повторного використання в будь-якій програмі без необхідності повторного навчання.

За допомогою фреймворку Flask було реалізовано основний API для взаємодії модулю прогнозування з веб-застосунком.

Оскільки точність методів машинного навчання в більшості випадків залежить від якості вхідного набору даних, то можна зробити висновок, що поточної інформації недостатньо для отримання точних прогнозів, тому необхідно розширити набір показників для аналізу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Aggarwal C.C., Charu C. Data Classification Algorithms and Applications. 2015: Chapman & Hall /CRC.
2. Aggarwal C.C., Charu C. Outlier Analysis. Springer, 2013.
3. Berry, Michael J. A. “Data mining techniques: for marketing, sales, and customer relationship management “/ Michael J.A. Berry, Gordon Linoff. – 2nd ed.
4. Bootstrap documentation [Електронний ресурс] Режим доступу до ресурсу: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
5. Guido S. Introduction to Machine Learning with Python / S. Guido, A. Müller. – Sebastopol, United States: O'Reilly Media, Inc, USA, 2016. – 392 с.
6. HTML самоучитель [Електронний ресурс] – Режим доступу до ресурсу: <http://htmlbook.ru/blog/samouchitel-html-1>.
7. Larose, Daniel T. “Discovering knowledge in data: an introduction to data mining” / Daniel T. Larose
8. Typescript documentation [Електронний ресурс] Режим доступу до ресурсу: <https://www.typescriptlang.org/docs/home.html>
9. Болгов М.Ю. Автоматизация медицинских учреждений: Руководство пользователя TherDep5 / М.Ю. Болгов. – К.: Куприянова, 2006. – 464 с.
10. Воронцов К.В. Лекции по методу опорных векторов [Електронний ресурс] URL: <http://www.ccas.ru/voron/download/SVM.pdf>
11. Гаман П.І. Інформаційні технології в системі охорони здоров'я. Державне регулювання процесів економічного і соціального розвитку, 2008
12. Гультяев, А.К. MS office Project Professional 2013 Управление проектами: Практическое пособие / А.К. Гультяев. - СПб.: Корона Принт, 2016. - 480 с

13. Гультьев, А.К. MS Project 2010 Professional. Управление проектами: Самоучитель / А.К. Гультьев. - СПб.: Корона Принт, 2009. - 512 с.
14. Джалота, П. Управление проектами в области информационных технологий / П. Джалота. - М.: Лори, 2014. - 224 с.
15. Довідник CSS [Електронний ресурс] – Режим доступу до ресурсу: <http://htmlbook.ru/css>.
16. Довідник HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://webref.ru/html>.
17. Дудзяний І. М. Об'єктно-орієнтоване моделювання програмних систем / І. М. Дудзяний. – Львів, 2007. – 107 с. – (Видавничий центр ЛНУ імені Івана Франка).
18. Карпова, Т. С. Базы данных: Модели, разработка, реализация: Учеб. пособие / Т. С. Карпова – СПб. и др.: Питер, 2002. – 304 с.
19. Коваленко, С.П. Управление проектами: Практическое пособие / С.П. Коваленко. - Мн.: Тетралит, 2013. - 192 с.
20. Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 1 : навчальний посібник / Кветний Р. Н., Богач І. В., Бойко О. Р., Софина О. Ю., Шушура О.М.; за заг. ред. Р.Н. Кветного. – Вінниця: ВНТУ, 2012. – 193 с.
21. Куліш Н.М. Сучасні проблеми впровадження інформаційних технологій в медичну практику. Буковинський державний медичний університет, 2012
22. Луїс Педро Коельо, Віллі Річард, Построение систем машинного обучения на языке Python, 2016. – 302 с.
23. Московко М.В., Козоріз О.С. Інформаційна система для відбору спортсменів до змагань. Вінницький національний технічний університет

24. Назаренко Г.И. Медицинские информационные системы: теория и практика / Г.И. Назаренко, Я.И. Гулиев, Д.Е. Ермаков. – М.: Физматлит, 2005. – 320 с.
25. Пасічник В.В. Організація баз даних та знань: підручник для ВНЗ/ В.В. Пасічник, В.А. Резніченко.– К.: Видавнича група ВНУ, 2006.-384с.
26. Підручник JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <http://javascript.ru/tutorial>.
27. Спирина А.А. Общая теория статистики: Учебник под ред. / Спирина А.А., Башиной О.Э. – 5-е изд., перераб. и доп. М.: Финансы и статистика, 1999, 281 с.
28. Сучасний підручник JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.javascript.ru/>.
29. Чабан О., Бойко О. Огляд світової практики щодо впровадження медичних інформаційних систем та проблеми створення єдиного медико-інформаційного простору. Львівський національний медичний університет імені Данила Галицького, 2013
30. Чурпій І.К. Сучасний стан інформатизації в медицині/ І.К. Чурпій, Н.В. Чурпій, В.Д. Скрипко / Буковинський медичний вісник.-2011. -Т. 15, N 1.
31. Шаховська Н. Б., Болюбаш Ю. Я. Модель великих даних “сутність-характеристика”. Національний університет “Львівська політехніка”, кафедра інформаційних систем та мереж, 2015

ДОДАТОК А

1 ПЛАНУВАННЯ РОБІТ

1.1 Деталізація мети проекту методом SMART.

Продуктом дипломного проекту є інформаційна система оцінювання функціонального стану професійних спортсменів. Деталізація мети методом SMART наведена у табл. А.1.

Таблиця А.1 – Деталізація мети методом SMART

Specific (конкретна)	Прогнозування функціонального стану
Measurable (вимірювана)	Покращення процесу оцінювання за рахунок експертної оцінки
Achievable (досяжна)	Отримання прогнозів функціонального стану за рахунок використання методів машинного навчання
Relevant (реалістична)	Для досягнення поставленої мети є всі програмні, технічні та емпіричні знання, тому ціль є реалістичною.
Time-framed (обмежена у часі)	Ціль обмежена у часі, тому що результатів потрібно досягти в найближчі пів року, так як пізніше така система може бути вже створена.

1.2 Планування змісту структури робіт ІТ-проекту (WBS)

З метою визначення обсягу і змісту проекту, в управлінні проектними роботами використовується такий інструмент як ієрархічна структура робіт (WBS - work breakdown structure). Вона дає цінні дані для оцінки вартості, складання графіка та оцінки прогресу проекту.

Структура декомпозиції робіт (WBS) є ієрархічною декомпозицією цілей проекту на орієнтовані на результат завдання, що виконуються

проектною групою для досягнення спільних цілей проекту. WBS утворює основу всієї діяльності з планування проекту. WBS ділить обсяг проектних робіт на більш дрібні, керовані пакети робіт для збереження кращого контролю над операціями проекту.

Зміст структури робіт показано на рисунку А.1.1.

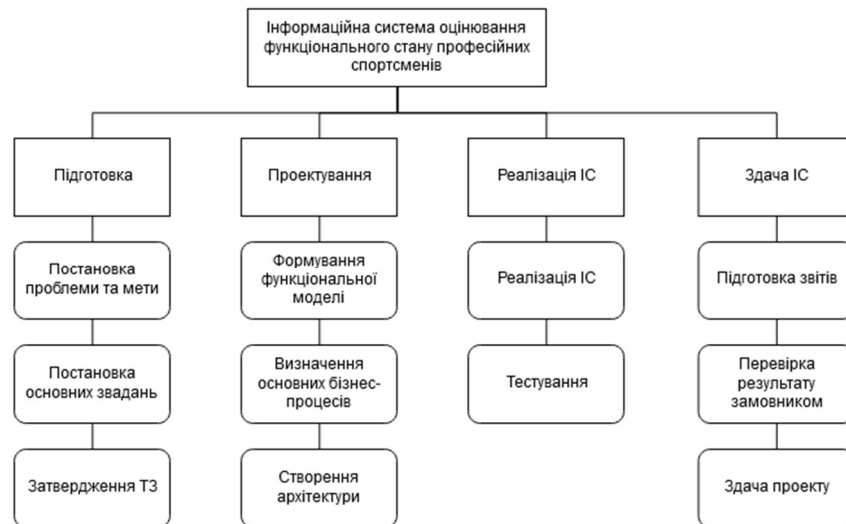


Рисунок А.1.1 – WBS діаграма

1.3 Планування структури організації, для впровадження готового проекту (OBS)

Наступним кроком розробки структури проекту є визначення організаційної структури (OBS) проекту.

Організаційна структура проекту (OBS) – є графічним відображенням учасників проекту (фізичних та юридичних осіб) та їхніх відповідальних осіб, залучених до реалізації проекту.

Елементами OBS можуть бути:

- окремі виконавці (керівники, фахівці, службовці);

- організації, структурні підрозділи і служби, у яких зайнята та або інша кількість фахівців, що виконують певні функціональні обов'язки;
- зовнішні постачальники обладнання, послуг;
- інші організації.

Будується OBS аналогічно робочій структурі, а саме:

- на першому рівні відображається організаційна структура як єдиний елемент;
- на другому і нижчих рівнях ділиться нижча частина структури на основні організаційні елементи.

Цей процес повторюється до найбільш низького рівня - базових робочих груп (змішаних цільових або функціональних), а при реалізації малих проектів - до окремих виконавців.

Обсяг робіт для цих найнижчих організаційних рівнів являє собою найбільш низькі елементи WBS, кожен з яких можна планувати і контролювати як окремі одиниці. Саме таке правило діє для створення OBS. Кількість рівнів залежить від розміру проекту.

1.4 Побудова календарного графіка виконання ІТ-проекту

Календарний графік виконання ІТ-проекту представимо у вигляді діаграми Ганта, яка складається із відрізків, які розміщені на горизонтальній шкалі часу. Кожен відрізок представляє собою певне завдання чи підзавдання. Початок, кінець і довжина відрізка відповідає початку, завершенню та тривалості завдання.

Завдання можуть виконуватися як паралельно так і послідовно. Якщо завдання виконуються послідовно, то існує зв'язок між нею і попередньою задачею відповідно. Наступна задача буде виконуватися тільки після завершення попередньої.

Паралельні завдання в проекті потрібно починати якнайшвидше, що дає змогу зекономити час і тривалість виконання проекту.

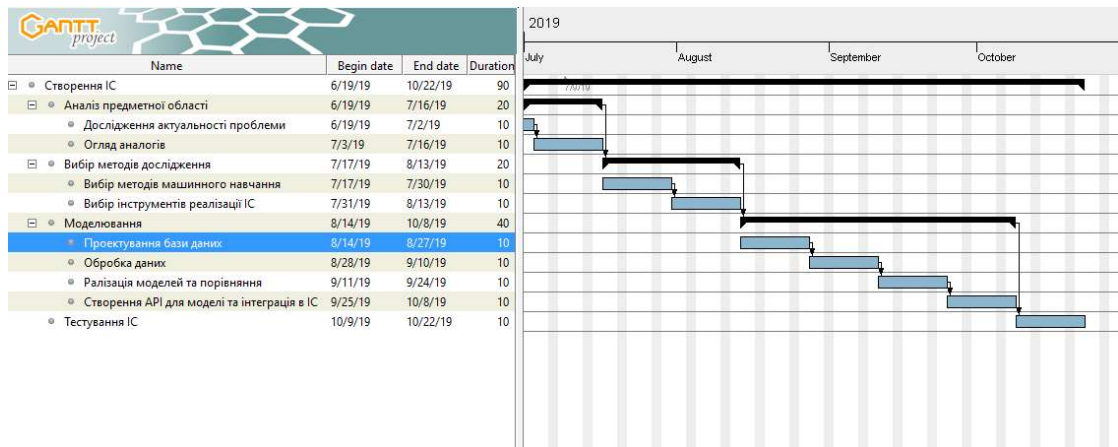


Рисунок А.1.2 – Діаграма Ганта

1.5 Аналіз ризиків проекту

Виконаємо якісну і кількісну оцінку ризиків роботи. При якісній оцінці визначимо ризики, що потребують швидкого реагування. Така оцінка визначить ступінь важливості ризику і дозволить вибрати спосіб реагування. Кількісна оцінка ризиків буде виконана для більш повної ідентифікації ризиків та ступеня їхнього впливу на виконання проекту. Кількісна і якісна оцінка ризиків можуть використовуватися окремо або разом, залежно від наявного часу і бюджету, необхідності в кількісній або якісній оцінці ризиків. У табл. 1.4 знаходиться класифікація ризиків за показниками ймовірності виникнення ризику та величині втрат.

Далі виконаємо планування реагування на ризики — це розробка методів і технологій зниження негативного впливу ризиків на проект. Визначимо ефективність розробки реагування на проект, визначимо чи будуть наслідки впливу ризику на проект позитивними або негативним. Оцінюємо ризики за показниками, що знаходяться в табл. А.2. На основі оцінки будемо

матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рис. А.1.3.

Таблиця А.2 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

Ймовірність виникнення	3	RS_2	RS_3,	RS_5, RS_9	
	2	RS_1, RS_13	RS_4, RS_6	RS_7, RS_14	
	1	RS_12	RS_8, RS_11	RS_10, RS_15	
			1	2	3
			Вплив ризику		

Рисунок А1.3 – Матриця ймовірності виникнення ризиків та впливу ризику

- зелений колір – прийнятні ризики;
- жовтий колір – виправданні ризики;
- червоний колір – недопустимі ризики.

На підставі отриманого значення індексу ризику класифікують: за рівнем ризику, що знаходиться в табл. А.3.

Таблиця А.3 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$1 \leq R \leq 2$	1,8,11,12,13
2	Виправдані	$3 \leq R \leq 4$	2,4,6,10,15
3	Недопустимі	$6 \leq R \leq 9$	3,5,7,9,14

Таблиця А.4 – Оцінка ймовірності виникнення, величини витрат та індексу ризику

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Непорозуміння між розробником та замовником	Низька	Середній	3	<ol style="list-style-type: none"> 1. Налагодити гарні відносини між розробником та керівником. 2. Дотримуватися ділового етикету спілкування. 3. Створити комфортні умови для співпраці 	Попередження	При виявленні непорозуміння потрібно вияснити, що саме стало причиною непорозуміння обговорити її та створити здорову атмосферу в колективі.
RS_2	Відкритий	Поява альтернативного продукту	Низька	Високий	3	<ol style="list-style-type: none"> 1. Провести попереднє дослідження альтернативних продуктів. 2. Вибрати унікальну стратегію створення проекту. 	Прийняття	

Продовження таблиці А.4

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_3	Відкритий	Нечітке завдання на розробку	Середня	Високий	6	<ol style="list-style-type: none"> 1. Ясно і однозначно обговорити із замовником усі види вимог. 2. Скласти глосарій для запобігання розбіжностей у розумінні слів та термінів. 3. Періодичний контроль замовником етапів роботи. 	Попередження	При виявленні невідповідностей деяких характеристик продукту заявленим вимовам потрібно уважно та чітко окреслити те, що було виконано невірно та зробити правки.
RS_4	Відкритий	Низька кваліфікація розробників проекту	Середня	Середній	4	<ol style="list-style-type: none"> 1. Підвищити кваліфікацію персоналу. 2. Переглянути онлайн-ресурси для підвищення рівня знань. 	Пом'якшення	Врахувати час на підготовку працівників. Видати літературу, переглянути онлайн-уроки.
RS_5	Відкритий	Неоптимальний розподіл часу	Висока	Високий	9	Провести аналіз актуальності найважливіших процесів та робіт. Звернути особливу увагу на правильність розподілу часу. Правильно визначити пріоритети виконання робіт. Чітко дотримуватися календарного плану.	Пом'якшення	Змінити порядок пріоритетів робіт. Знайти способи оптимізації роботи із вже існуючою розстановкою. Обговорити варіанти внесення правок до термінів реалізації із замовником.

Продовження таблиці А.4

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_6	Відкритий	Часте внесення змін у ТЗ	Середня	Середній	4	<ol style="list-style-type: none"> 1. Виділити всі необхідні параметри проекту. 2. Чітко описати вимоги до проекту. 3. Обговорити всі технічні засоби виконання проекту та умови реалізації. 	Перенос	Узгодити всі положення з замовником, у разі потреби внести необхідні зміни та поправки.
RS_7	Відкритий	Вибір неефективної технології розробки	Середня	Високий	6	<ol style="list-style-type: none"> 1. Проаналізувати методи та засоби, для виконання проекту. 2. Обрати зрозумілу та легку в використанні технологію розробки. 	Пом'якшення	Виділити час та ресурси на пошуки покращення обраної технології. Застосувати допоміжні ресурси.
RS_8	Відкритий	Не вірна оцінка масштабів проекту	Низька	Середній	2	<p>Провести детальний аналіз проекту. Визначити основні етапи проекту, розподілити час на їх виконання.</p> <p>Проаналізувати масштаби проекту на основі додаткових джерел.</p>	Пом'якшення	Переоцінка масштабів проекту. Перебудова стратегії реалізації проекту.

Продовження таблиці А.4

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_9	Відкритий	Помилки проектування	Висока	Високий	9	На етапі проектування тісно співпрацювати із замовником та на певних етапах демонструвати поточні результати.	Пом'якшення	Здійснювати проміжний контроль результатів в ході виконання проекту.
RS_10	Відкритий	Збої в роботі програмного забезпечення	Низька	Високий	3	1. Підготувати резерв програмних засобів. 2. Залучити спеціаліста для усунення збоїв.	Попередження	Замінити програмне забезпечення.
RS_11	Відкритий	Відсутність резервних копій даних	Низька	Середній	2	1. Налаштувати автоматичне збереження даних. 2. Зберігати дані на різних носіях інформації.	Попередження	Робити копію даних після кожного виконаного етапу.
RS_12	Відкритий	Реалізація непотрібної функціональності	Низька	Низький	1	Попередити замовника про можливість додаткового функціоналу.	Використання	Обговорити вигоди і збитки від можливих змін проекту.
RS_13	Відкритий	Невиконання моніторингу проекту	Середня	Низький	2	Здійснювати проміжний контроль результатів в ході виконання проекту. Здійснювати моніторинг проекту працівниками.	Перенос	Здійснювати моніторинг проекту замовником. Надання проміжних результатів виконання проекту після кожного етапу.

Продовження таблиці А.4

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_14	Відкритий	Виникнення проблем із програмним забезпеченням користувачів	Середня	Високий	6	Розробка проекту з врахуванням вимог до програмного забезпечення користувачів проекту. Модифікація проекту з врахуванням різних версій програмного забезпечення, яке буде застосовуватися.	Прийняття	
RS_15	Відкритий	Зміна вимог замовника в процесі розробки проекту	Низька	Високий	3	Узгодити всі питання на початкових етапах, щоб мінімізувати кількість змін під час розробки.	Пом'якшення	Переоцінка проекту, кожного разу, коли вимоги змінюються.