

Національний лісотехнічний університет України
(повне найменування вищого навчального закладу)
Навчально-науковий інститут комп'ютерних наук
та інформаційних технологій
(повне найменування інституту, назва факультету (відділення))
Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

Магістерська кваліфікаційна робота

другий (магістерський)
(рівень вищої освіти)

на тему:

**Інформаційна система персоналізованих медичних
рекомендацій на основі методів машинного навчання**

Виконав: студент VI курсу, групи КН-63м
спеціальності

122 – “Комп'ютерні науки”

(шифр і назва напрямку підготовки, спеціальності)

Паньків А.О.

(прізвище та ініціали)

Керівник Пірко І.Б.

(прізвище та ініціали)

Рецензент

Сторожук О.А.

(прізвище та ініціали)

Львів – 2025 р.

Національний лісотехнічний університет України
(повне найменування вищого навчального закладу)


ННІ комп'ютерних наук та інформаційних технологій

Кафедра комп'ютерних наук

Рівень вищої освіти другий (магістерський)

Спеціальність 122 "Комп'ютерні науки"
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри КН

 Борецька І. Б.
"10" травня 2025 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Паньків Андрій Олегович
(прізвище, ім'я, по батькові)

1. Тема роботи **Інформаційна система персоналізованих медичних
рекомендацій на основі методів машинного навчання**

керівник роботи Пірко І. Б., канд. фіз.-мат. наук, доцент.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 29.04. 2025 р.№ С-288

2. Термін подання студентом роботи 10.12. 2025 р.

3. Вихідні дані до роботи:

- вивчити предметну область, проаналізувати існуючі фактори та методи моделювання, а також відповідні програмні продукти;
- розглянути і використати алгоритми, які лежать в основі математичної моделі персоналізованих медичних рекомендацій;
- спроектувати інформаційну систему з допомогою мови програмування Python та відповідних бібліотек;
- представити результати роботи інформаційної системи.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Розділ 1. Стан проблемної області

Розділ 2. Інформаційне забезпечення

Розділ 3. Математичне забезпечення

Розділ 4. Програмне забезпечення

Розділ 5. Розроблення стартап-проєкту

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

додаток А, додаток Б

6. Дата видачі завдання 1 травня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літературних даних та інших джерел згідно досліджуваної теми	01.05-30.05.2025	виконано
2	Аналіз досліджуваної теми та вибір відповідних варіантів її розробки	01.06-30.06. 2025	виконано
3	Постановка задачі та її формалізація	01.07-30.07. 2025	виконано
4	Вибір та обґрунтування методів і засобів проведення дослідження	01.08-30.08. 2025	виконано
5	Розроблення концептуальної схеми реалізації завдання	01.09-15.09. 2025	виконано
6	Програмна реалізація завдання	16.09-30.10. 2025	виконано
7	Тестування програмного продукту та отриманих результатів	01.11-15.11. 2025	виконано
8	Розробка пояснювальної записки магістерської роботи	16.11-30.11. 2025	виконано
9	Корегування пояснювальної записки згідно вимог, розроблення презентації	01.12-09.12. 2025	виконано

Студент

Керівник роботи


(підпис)


(підпис)

Паньків А.О.
(прізвище та ініціали)

Пірко І.Б.
(прізвище та ініціали)

АНОТАЦІЯ

Магістерська робота містить 86 сторінок пояснювальної записки, 9 рисунків, 4 таблиці, 15 джерел, 2 додатки.

У роботі розроблено інформаційну систему персоналізованих медичних рекомендацій на основі методів машинного навчання. Проведено аналіз сучасних підходів у сфері персоналізованої медицини та зібрано вимоги до функціоналу системи. Створено математичну модель та алгоритм, що дозволяють формувати рекомендації на основі індивідуальних медичних показників користувача. Реалізовано програмне забезпечення зі зручним графічним інтерфейсом для збору даних та виведення рекомендацій.

Ключові слова: *медичні рекомендації, машинне навчання, персоналізована медицина, медичні дані, класифікація, Python.*

ABSTRACT

The thesis contains 86 pages of explanatory note, 9 figures, 4 tables, 15 used literary sources, 2 appendix.

The work developed an information system of personalized medical recommendations based on machine learning methods. An analysis of modern approaches in the field of personalized medicine was conducted and requirements for the system's functionality were collected. A mathematical model and algorithm were created that allow generating recommendations based on the user's individual medical indicators. Software with a convenient graphical interface for data collection and output of recommendations was implemented.

Keywords: *medical recommendations, machine learning, personalized medicine, medical data, classification, Python.*

ТЕХНІЧНЕ ЗАВДАННЯ

В дипломній роботі потрібно розробити інформаційну систему персоналізованих медичних рекомендацій, для цього потрібно вирішити такі завдання.

1. Провести аналіз сучасних систем персоналізованих медичних рекомендацій, визначити їх переваги та недоліки.

2. Розробити математичну модель системи, що базується на методах машинного навчання для формування медичних рекомендацій.

3. Створити алгоритм обробки індивідуальних медичних даних користувачів з врахуванням особливостей захворювань та персональних параметрів.

4. Реалізувати програмне забезпечення з графічним інтерфейсом для збору медичних даних та видачі персоналізованих рекомендацій.

5. Провести тестування розробленої інформаційної системи, оцінити її ефективність та можливості подальшого розширення функціоналу та інтеграції у телемедичні платформи.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ	8
ВСТУП	9
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ	11
1.1. Огляд рекомендацій щодо персоналізованої медицини	11
1.2. Рекомендації щодо ліків на основі кількох захворювань	14
1.3. Рекомендації щодо ліків з додатковими знаннями	16
Висновки до розділу	19
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	20
2.1. Загальна характеристика інформаційного забезпечення	20
2.2. Структура вхідних та вихідних даних	21
2.3. Опис наборів даних інформаційної системи	22
Висновки до розділу	28
РОЗДІЛ 3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	29
3.1. Формалізація задачі	29
3.2. Модель машинного навчання	30
3.3. Робота з медичними даними	32
Висновки до розділу	35
РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	36
4.1. Розроблення інформаційної системи персоналізованих медичних рекомендацій	36
4.2. Результати роботи інформаційної системи	47
4.3. Розроблення графічного інтерфейсу інформаційної системи	49
Висновки до розділу	53
РОЗДІЛ 5. РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ	54
5.1. Структура проєкту інформаційної системи персоналізованих медичних рекомендацій	54
5.2. Цільова аудиторія	55
5.3. Бізнес-модель стартапу	57

Висновки до розділу	60
ВИСНОВКИ	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
ДОДАТКИ	64
ДОДАТОК А	64
ДОДАТОК Б	71

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

AI (Artificial Intelligence) – штучний інтелект;

ANN (Artificial Neural Network) – штучна нейронна мережа;

AUC (Area Under the Curve) – метрика, що вимірює площу під кривою ROC і показує здатність моделі розрізняти класи;

BoW – метод мішок слів;

CNN (Convolutional Neural Network) – згорткова нейронна мережа;

IDE – інтегроване середовище розробки;

GAN () – генеративно-змагальні мережі;

ML (Machine Learning) – машинне навчання;

NLP – обробка природної мови;

ROC (Receiver Operating Characteristic) – графік, що відображає залежність між чутливістю та частотою хибних спрацьовувань для різних порогів класифікації;

SHAP (SHapley Additive exPlanations) – аналіз внеску кожної ознаки у результат;

Word2Vec – слово у вектор;

ЕСОЗ – електронна система охорони здоров'я;

ЕМК – електронні медичні карти;

ЕМЗ – електронний медичний запис;

МІС – медична інформаційна система;

ШНМ – штучна нейронна мережа.

ВСТУП

Актуальність дипломної роботи

Зростаюча поширеність хронічних захворювань та інших медичних проблем робить необхідним створення ефективних інструментів для персоналізованих рекомендацій. Високий рівень навантаження на медичну систему збільшує потребу у автоматизованих рішеннях для швидкої діагностики та рекомендацій. Використання методів машинного навчання дозволяє аналізувати великі об'єми медичних даних та формувати більш точно індивідуальні рекомендації. Розробка такої системи сприяє підвищенню доступності медичних консультацій для населення. Персоналізовані рекомендації допомагають зменшити кількість діагностичних помилок і покращити результати лікування. Впровадження інтелектуальних систем у медицину є актуальним завдяки розвитку телемедицини та мобільних додатків. Аналіз сучасних досліджень демонструє високий потенціал застосування машинного навчання для оптимізації медичних процесів. Задоволення потреб користувачів у швидких, точних та персоналізованих рекомендаціях є важливим фактором у сучасному суспільстві. Враховуючи глобальні тенденції у медичній галузі розробка такої системи є необхідною для підвищення ефективності та якості надання медичних послуг.

Предмет дослідження – розробка інформаційної системи для персоналізованих медичних рекомендацій на основі методів машинного навчання.

Об'єкт дослідження – набір медичних даних і зображень, що використовуються для формування рекомендацій у медицині.

Мета роботи – розробка та реалізація інформаційної системи для формування медичних рекомендацій на основі аналізу індивідуальних даних користувача за допомогою методів машинного навчання.

Завдання.

В дипломній роботі потрібно розробити інформаційну систему персоналізованих медичних рекомендацій, для цього потрібно вирішити такі завдання.

1. Провести аналіз сучасних систем персоналізованих медичних рекомендацій, визначити їх переваги та недоліки.
2. Розробити математичну модель системи, що базується на методах машинного навчання для формування медичних рекомендацій.
3. Створити алгоритм обробки індивідуальних медичних даних користувачів з урахуванням особливостей захворювань та персональних параметрів.
4. Реалізувати програмне забезпечення з графічним інтерфейсом для збору медичних даних та видачі персоналізованих рекомендацій.
5. Провести тестування розробленої інформаційної системи, оцінити її ефективність та можливості подальшого розширення функціоналу та інтеграції у телемедичні платформи.

Наукова новизна одержаних результатів

Наукова новизна полягає у створенні нових підходів до застосування машинного навчання для персоналізованого медичного консультування, що дозволяє більш точно аналізувати індивідуальні дані пацієнтів. Впроваджено розробку математичної моделі системи, яка враховує специфіку медичних параметрів та їх взаємовпливи. Також запропоновано новий алгоритм формування рекомендацій, що забезпечує їхню високу точність. Впроваджено архітектуру системи, яка може бути адаптована під різні захворювання та розширювана новими функціями. Ефективність системи підтверджена результатами експериментів, що демонструють її застосовність у практичних медичних умовах.

Практичне значення одержаних результатів

Розроблена система може бути впроваджена у телемедичних платформах та мобільних додатках, що дозволяє зручно використовувати її для широкого кола користувачів. Впровадження системи сприяє більш точному прогнозуванню та діагностиці захворювань на основі індивідуальних медичних даних пацієнтів. Вона здатна допомогти медичним працівникам швидше та точніше приймати рішення, зменшуючи навантаження на лікарів. Результати роботи сприятимуть покращенню якості медичного обслуговування та підвищенню рівня здоров'я населення.

РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1. Огляд рекомендацій щодо персоналізованої медицини

Отримання цінних медичних знань з масивних медичних даних для підтримки прийняття клінічних рішень став важливою галуззю досліджень. Персоналізовані рекомендації щодо медицини є важливим напрямком досліджень у цій галузі, метою якого є рекомендування найбільш придатних ліків для кожного пацієнта відповідно до стану здоров'я пацієнта. Персоналізовані рекомендації щодо медицини можуть допомогти лікарям приймати правильні рішення та уникати виникнення медичних відхилень. Виходячи з цього, в роботі [1] зроблено огляд персоналізованих рекомендацій щодо медицини. Визначено проблему персоналізованої рекомендації щодо медицини. Потім, починаючи з ключових теорій та технологій, алгоритми персоналізованої рекомендації щодо медицини, запропоновані в останні роки, систематично класифікуються (рекомендації щодо множинних захворювань, рекомендації щодо медицини з комбінаційним шаблоном, рекомендації щодо медицини з додатковими знаннями та рекомендації щодо медицини на основі зворотного зв'язку) та поглиблено аналізуються. Представлено, як оцінювати алгоритми персоналізованої рекомендації щодо медицини та деякі загальні показники оцінки, висуваються проблеми персоналізованої рекомендації щодо медицини та прогнозуються майбутні напрямки досліджень та тенденції розвитку.

Клінічна система підтримки рішень [2] є важливим засобом покращення якості медичної допомоги, метою якого є оцінка та покращення якості медичної допомоги, зменшення кількості медичних помилок та контроль медичних витрат. Рекомендація ліків є важливою проблемою, яка може допомогти приймати найбільш доцільні рішення щодо ліків шляхом поглибленого аналізу медичних записів [2]. Вибір ліків стандартним та науковим способом відіграє активну роль у сприянні відновленню здоров'я. Тому проблема рекомендації ліків отримала значну увагу з боку багатьох дослідників, було запропоновано деякі моделі рекомендації ліків [3]. Системи рекомендацій традиційної медицини розроблені на основі правил, визначених досвідченими лікарями відповідно до клінічних рекомендацій. Крім того ці

традиційні методи, що базуються на правилах, важко впровадити. По-перше, через обмеження особистого пізнання, покладатися на лікарів у створенні правил недостатньо. По-друге, для створення та підтримки правил потрібно багато часу та досвіду.

З появою та популяризацією електронних медичних карток (ЕМК) отримання та збір медичних даних стали зручнішими, що призвело до накопичення великої кількості клінічних даних, таких як життєво важливі показники, діагноз захворювань, рецептурні ліки, медичні витрати. Технологія глибокого навчання забезпечує нові технічні засоби для вилучення та використання медичних даних. Це дозволяє персоналізувати рекомендації щодо ліків, щоб компенсувати недоліки традиційних підходів. Мета персоналізованої медицини полягає у використанні електронних медичних карток для прогнозування найбільш придатних ліків для кожного пацієнта, які можна використовувати як орієнтир для лікарів для призначення. Дослідження охоплює отримання закономірностей змін стану здоров'я пацієнтів, що неявно містяться в ЕМК, пом'якшення побічних ефектів, спричинених лікарською взаємодією [3]. Персоналізована модель рекомендацій щодо ліків може забезпечити кращі персоналізовані рекомендації щодо лікування лікарям та пацієнтам для покращення прогнозу пацієнтів та ефективнішого використання медичних ресурсів.

Виходячи з вищезазначеного персоналізована рекомендація ліків є важливим дослідницьким питанням. З'явилася велика кількість пов'язаних досліджень щодо рекомендації персоналізованої медицини [4-7]. Оскільки рекомендація персоналізованої медицини є міждисциплінарною, тобто вимагає знань у машинному навчанні, медицині та інших аспектах, новим дослідникам у цій галузі важко зрозуміти новітні розробки. Щоб вирішити цю проблему, запропоновано огляд [8-10] дослідницької роботи з рекомендації персоналізованої медицини, сподіваючись розібратися та узагальнити методи рекомендації персоналізованої медицини з нової перспективи. Зокрема прокласифіковано існуючі методи рекомендації персоналізованої медицини з різних точок зору.

Розглянемо визначення проблеми персоналізованої рекомендації щодо ліків [11]. Більшість моделей персоналізованої рекомендації щодо ліків приймають рішення на основі часткових даних електронної медичної картки.

Усі медичні записи пацієнта записуються в електронний медичний запис (ЕМЗ), тобто ЕМЗ зберігає діагнози, ліки та іншу інформацію про пацієнта у вигляді медичних кодів. Використовують часову послідовність госпіталізацій для представлення медичних даних пацієнта в ЕМЗ. Кожна госпіталізація є кортежем, що складається з усіх кодів захворювань та кодів ліків при госпіталізації. Якщо також враховувати демографічну інформацію та лабораторні показники пацієнта, вираз кожної госпіталізації слід розширити від бінарної групи до триплетної групи.

Оскільки постановка проблеми рекомендації персоналізованої медицини різниться в деталях за різними сценаріями, рис. 1.1 є узагальненим схематичним зображенням проблеми рекомендації ліків. Тут наведено просте визначення персоналізованої рекомендації ліків: враховуючи дані про попередні госпіталізації пацієнта, тобто історичну послідовність госпіталізацій, та інформацію про захворювання пацієнта на момент госпіталізації. Персоналізована рекомендація ліків має на меті створити набір ліків, які є безпечними та ефективними для пацієнта на основі цієї інформації.

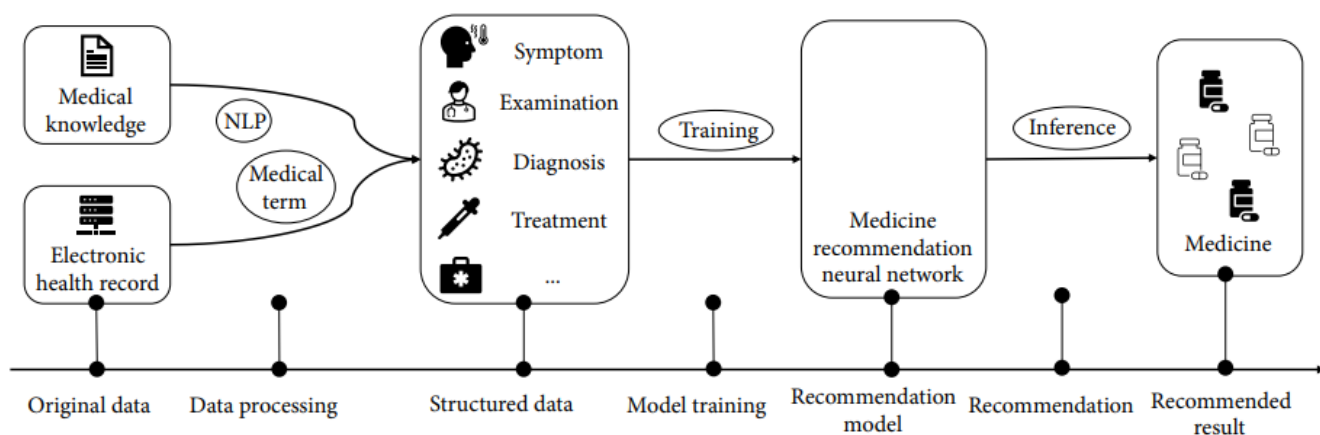


Рисунок 1.1 – Узагальнене схематичне зображення проблеми рекомендації ліків

Рекомендації щодо ліків спираються на електронну медичну карту та пов'язані медичні знання, витягуючи структуровану інформацію про симптоми, обстеження, діагнози та лікування пацієнтів за допомогою таких методів, як обробка природної мови (NLP). Потім ця інформація використовується для навчання моделі нейронної

мережі для рекомендації ліків. Отже, для нових візитів навчена нейронна мережа для рекомендації ліків може бути використана для висновків щодо отримання відповідних ліків.

Загалом персоналізована рекомендація щодо ліків стосується рекомендації найбільш підходящих ліків для кожного пацієнта на основі стану здоров'я пацієнта. Але вибір способу характеристики стану пацієнта відрізняється за різних обставин. У деяких випадках немає інформації про попередні візити пацієнта, рекомендація базується лише на поточному захворюванні пацієнта. У деяких випадках демографічна інформація та лабораторні показники можуть бути враховані на додаток до попередніх візитів пацієнта та захворювання пацієнта.

1.2. Рекомендації щодо ліків на основі кількох захворювань

Хоча під час лікування конкретного захворювання кожен пацієнт відрізняється, лікарі повинні враховувати багато факторів одночасно, є багато ліків на вибір. Як результат деякі моделі рекомендацій надають персоналізовані рекомендації щодо ліків для конкретних захворювань, щоб допомогти лікарям аналізувати стан пацієнтів та приймати кращі рішення на основі складної інформації. Оскільки діабет є хронічним та прогресуючим захворюванням, існує багато різних ліків, доступних для лікування, багато сучасних персоналізованих моделей рекомендацій щодо ліків спрямовані на діабет. В роботі [12] розроблено систему підтримки рішень, яка використовує підхід, що базується на багатокритеріальному прийнятті рішень та онтології предметної області. Згодом в роботах [13-15] запропонували систему рекомендацій щодо ліків від діабету, засновану на технології нечіткого міркування та онтології. Було використано підхід, заснований на подібності пацієнтів, запропоновано медичні знання для отримання рекомендованих ліків. Окрім діабету, також були запропоновані методи персоналізованої рекомендації ліків, орієнтовані на інші поширені захворювання, такі як рак, епілепсія, гіпертензія, трансплантація кісткового мозку та туберкульоз. Загалом ці моделі можуть мати кращий рекомендаційний ефект для конкретних захворювань. Однак, з одного боку, розробити модель рекомендацій для кожного захворювання занадто складно, з

іншого боку пацієнти часто страждають не лише від одного захворювання. У цьому випадку модель рекомендацій щодо ліків для конкретних захворювань не працює.

Для пацієнтів, які страждають від кількох захворювань одночасно, навіть досвідченим лікарям важко дати відповідні рекомендації. З огляду на цю проблему деякі дослідження запропонували загальні персоналізовані моделі рекомендацій щодо ліків, які можуть вирішити проблему супутньої наявності кількох захворювань. LEArn to Prescribe (LEAP) [15] розбиває рекомендацію щодо ліків на багатоетапний процес, рекомендуючи один препарат за раз і автоматично визначаючи відповідний кінець рекомендації. Персоналізований рецепт на супутні захворювання агрегує дані з багатьох різних джерел, щоб вивчити три рівні представленості пацієнта, захворювання та ліків. Об'єднуючи різні рівні представництва, можна досягти персоналізованого призначення супутніх захворювань, кореляції між ліками слід розглядати неупорядковано та запропоновано комбіновану мережу прогнозування ліків.

В роботі [14] вперше ввели графову структуру в задачу персоналізованої рекомендації ліків шляхом інтеграції графа знань про взаємодію ліків як модуль пам'яті графової згорткової мережі та моделювання записів пацієнтів як запитів. Використано можливості генеративно-змагальних мереж (GAN) для вивчення представлень з медичних записів конкретного пацієнта та надання точних рекомендацій щодо ліків пацієнту. Також запропоновано модель рекомендації ліків, яка зосереджена на взаємодії ліків під назвою SafeDrug. Ліки мають тенденцію повторюватися протягом багатьох візитів пацієнта, відмінності між ліками є більш значущими. Було запропоновано нову модель MICRON для вирішення цієї проблеми за допомогою рекурентної мережі залишків для фіксації змін ліків. У всіх вищезазначених роботах формулюється рекомендація ліків як завдання класифікації з кількома мітками для прогнозування ліків. Було згенеровано ліки за допомогою нового механізму реплікації або прогнозування та розроблено мережу умовної генерації.

Ранні моделі персоналізованих рекомендацій щодо ліків рекомендували ліки, що підходять пацієнтам, на основі виключно їхнього зв'язку з кожним препаратом.

Було розроблено [12] хмарно-допоміжну рекомендацію щодо ліків CADRE, нову хмарно-допоміжну модель рекомендацій щодо ліків, яка може рекомендувати пацієнту ліки на основі симптомів. В роботі [13] змодельовано рекомендацію щодо ліків як проблему прогнозування зв'язків і реалізовано зменшення розмірності шляхом навчання представлення трьох типів даних, а саме хвороби, ліків та пацієнта, щоб отримати їх представлення в одному низьковимірному просторі. Однак потенційні взаємодії між ліками, такі як синергія або антагонізми, значною мірою ігноруються і можуть призвести до того, що рекомендовані ліки будуть неоптимальними або навіть шкідливими.

Нещодавно персоналізовані моделі рекомендацій щодо ліків почали впроваджувати рекомендації щодо ліків на основі комбінованої моделі, тобто розглядають остаточний рекомендований препарат як повну комбінацію, а не кожен окремо.

Деякі персоналізовані моделі рекомендацій щодо ліків намагаються реалізувати динамічну рекомендацію комбінацій ліків, розділяючи госпіталізацію пацієнта на кілька часових вікон та рекомендуючи ліки для кожного часового вікна. Для моделювання взаємозв'язків між кількома ліками, захворюваннями та пацієнтами, було застосовано фреймворк актор-критик політики. Використовуючи модель навчання з контрольованим підкріпленням, засновану на рекурентних нейронних мережах, пацієнтам щодня рекомендується набір ліків. В роботі запроновано [14] перший фреймворк глибокого навчання з підкріпленням для оцінки оптимальних динамічних варіантів лікування на основі медичних даних спостережень.

1.3. Рекомендації щодо ліків з додатковими знаннями

Моделі ранніх рекомендацій щодо ліків базуються на правилах, які розробляються лікарями або спеціалістами на основі медичних знань та досвіду. Однак розробка та підтримка правил вимагають багато робочої сили, моделі на основі правил важко персоналізувати. З широким застосуванням електронних медичних карт, які забезпечують достатньо даних для досліджень, пов'язаних з підтримкою клінічних рішень, все більше досліджень починають впроваджувати

персоналізовані рекомендації щодо ліків за допомогою підходу, заснованого на даних.

Існує два типи моделей, що базуються на даних та електронних медичних картах (ЕМК). Моделі, що базуються на екземплярах, надають рекомендації лише відповідно до захворювань та процедур поточної госпіталізації пацієнта, ігноруючи історію хвороби пацієнта. Як результат модель, що базується на екземплярах, не може враховувати історичний перебіг захворювання. Щоб вирішити цю проблему, поздовжні моделі розроблені для використання поздовжніх історій пацієнтів та отримання часових кореляцій. Поздовжня модель враховує історію хвороби пацієнта, інтегрує дані про попередню госпіталізацію пацієнта в представлення пацієнта та вводить опис функції ліків як дані медичних знань.

Деякі персоналізовані моделі рекомендацій щодо ліків намагаються ввести медичні знання як допоміжну інформацію на основі набору даних. CADRE - це персоналізована модель рекомендацій щодо ліків, розроблена на основі спільної фільтрації користувачів та вводить дані медичних знань, а саме функціональний опис ліків. В роботі [11] змодельовано рекомендації щодо ліків як проблему прогнозування зв'язків та об'єднали електронні медичні карти та медичні знання для побудови медичного гетерогенного графа. Виконуючи навчання представлення на різних типах вузлів у гетерогенних графах, можна отримати вбудовування різних типів медичних сутностей в один низьковимірний простір.

Хоча більшість моделей персоналізованих рекомендацій щодо ліків базуються на даних електронних медичних карт, все ще існує кілька алгоритмів, які рекомендують ліки через відгуки пацієнтів або взаємодію пацієнтів з іншими суб'єктами. На відміну від інших персоналізованих моделей рекомендацій ліків, які призначені для допомоги лікарям у прийнятті рішень, моделі рекомендацій ліків у цій категорії є орієнтованими на користувача та призначені для допомоги пацієнтам у прийнятті рішень, тобто безпосередньо рекомендують ліки, які будуть придбані пацієнтам. Тому такі моделі можуть рекомендувати лише безрецептурні ліки. Ці моделі в основному допомагають пацієнтам, які страждають від поширених захворювань (таких як застуда та запалення шлунково-кишкового тракту) або

хронічних захворювань, вибирати серед ліків з подібною ефективністю. Вони також можуть допомогти людям без захворювань вибирати продукти для здоров'я. Порівняно з іншими персоналізованими алгоритмами рекомендацій ліків, рекомендації ліків, засновані на взаємодії або зворотному зв'язку, ближчі до рекомендацій товарів.

Оцінки пацієнтів щодо ліків, які мають великий вплив на рішення інших пацієнтів, зможуть спонукати інших пацієнтів приймати більш обгрунтовані рішення. Однак більшість попередніх досліджень з оцінки зосереджувалися на прогнозуванні рейтингів та рекомендаціях у сфері електронної комерції, існує мало досліджень у медичній галузі. Крім того, географічні та часові фактори також мають певний вплив на рекомендації ліків. Розроблено модель iDrug для вирішення цієї практичної проблеми. Зокрема, iDrug спочатку визначає шестикомпонентну мобільну мережу соціальних рейтингів для представлення чотирьох сутностей: пацієнт, ліки, час і місце, а також взаємодії між ними. Усі пацієнти тут страждають від одного й того ж захворювання або мають однакові симптоми. Потім на основі взаємозв'язків витягується ключова географічна та часова інформація для реалізації прогнозування рейтингу та рекомендації безрецептурних ліків.

Коли кількість пацієнтів з певним захворюванням швидко зростає за короткий проміжок часу, що призводить до масового навантаження на систему охорони здоров'я, пацієнти не можуть отримати рекомендації щодо ліків від лікарів і можуть приймати ліки лише самостійно, що може призвести до погіршення стану здоров'я. Персоналізована система рекомендацій щодо ліків може вирішити цю проблему.

Зокрема, для векторизації вихідних даних використовуються метод мішок слів (BoW), слово у вектор (Word2Vec) та частота термінів – обернена частота документів, а потім для виконання завдання класифікації використовуються наївний баєсівський метод, логістична регресія та класифікатор лінійних опорних векторів, щоб реалізувати персоналізовані рекомендації щодо ліків. Коментарі пацієнтів є формою вираження емоцій пацієнта та можуть бути використані як інформація для рекомендації найкращих ліків від певного захворювання.

ВИСНОВКИ ДО РОЗДІЛУ

В даному розділі встановлено, що персоналізовані медичні рекомендації є важливим напрямом у сучасній цифровій медицині. Існуючі системи рекомендацій активно використовують дані електронних медичних записів для прогнозування оптимального лікування. Традиційні експертні системи, що базуються на заданих правилах, мають обмежену гнучкість та складність масштабування. Методи машинного навчання дозволяють обробляти великі об'єми медичних даних та виявляти приховані закономірності. Сучасні підходи включають використання нейронних мереж, графових моделей, алгоритмів класифікації та навчання з підкріпленням. Особливу увагу дослідники приділяють рекомендаціям при множинних захворюваннях, де важливо враховувати взаємодію ліків. У літературі розглянуто як моделі, орієнтовані на підтримку лікарів, так і моделі, орієнтовані безпосередньо на пацієнтів. Ефективність таких систем значною мірою залежить від повноти медичних даних та коректної інтерпретації симптомів. Актуальним напрямом є інтеграція медичних знань з результатами обробки природної мови та онтологіями. Деякі моделі також враховують географічні, часові фактори та відгуки користувачів для підвищення точності рекомендацій. Проаналізовані джерела підтверджують високий потенціал персоналізованих рекомендацій у покращенні якості лікування.

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Загальна характеристика інформаційного забезпечення

Інформаційне забезпечення є важливою складовою функціонування будь-якої інформаційної системи. Воно визначає структуру, об'єм, джерела та способи організації даних, необхідних для прийняття рішень. У контексті медичних інформаційних систем особливу увагу приділено точності, повноті, конфіденційності та актуальності даних, що використовуються.

Для функціонування системи персоналізованих медичних рекомендацій необхідно обробляти значний об'єм різномірної інформації, яка може містити як структуровані числові дані, так і неструктуровані текстові записи, зображення, історії хвороб. Тому інформаційне забезпечення охоплює такі компоненти, як джерела даних, структура вхідних та вихідних даних, правила їх зберігання, доступу та обробки.

Основними джерелами даних для системи є:

- електронні медичні картки пацієнтів;
- анкети пацієнта, які заповнюються при первинному прийомі;
- результати лабораторних аналізів та інструментальних досліджень;
- скарги та симптоми, введені вручну або через чат-інтерфейс;
- історія хронічних захворювань та супутні діагнози;
- прийом лікарських препаратів, алергії;
- дані про спосіб життя: рівень фізичної активності, харчування, шкідливі звички.



Рисунок 2.1 – Типи медичних даних

2.2. Структура вхідних та вихідних даних

Для ефективного функціонування системи дані повинні бути структуровані у формат придатний до аналізу засобами машинного навчання.

Таблиця 2.1 – Структура вхідних даних

Назва параметра	Тип	Приклад значення
Вік	Числовий	45
Стать	Категоріальний	Жіноча
Індекс маси тіла (ІМТ)	Обчислюваний	27,3
Глюкоза	Числовий	6,4
Гіпертонія	Бінарний	1
Алергії	Категоріальний	Пеніцилін
Симптоми	Текстовий	Сонливість, слабкість
Хронічні хвороби	Категоріальний	Діабет 2 типу
Прийом ліків	Категоріальний	Метформін

Усі дані проходять попередню обробку, включаючи нормалізацію числових значень, кодування категоріальних даних, видалення пропусків та перевірку на узгодженість.

Вихідними даними системи є персоналізовані рекомендації, що формуються за результатами аналізу медичного профілю користувача. Ці рекомендації мають таку структуру:



Рисунок 2.2 – Структура рекомендацій для здоров'я

Рекомендації можуть бути подані у вигляді тексту або таблиці з можливістю виводу у PDF-звіт або інтеграції через API для медичних закладів. Розглянуто основні аспекти інформаційного забезпечення системи персоналізованих медичних рекомендацій. Визначено структуру вхідних і вихідних даних, джерела отримання медичної інформації, вимоги до її якості та способи зберігання. Це створює необхідну основу для реалізації системи, здатної працювати з великим об'ємом інформації у сфері охорони здоров'я.

2.3. Опис наборів даних інформаційної системи

Файл `description.csv` використовується в інформаційній системі персоналізованих медичних рекомендацій як вхідний набір даних, що містить медичні параметри пацієнтів, а також передбачувані або реальні діагнози. Він служить джерелом для тренування та тестування моделей машинного навчання, а також формування рекомендацій на основі вхідних ознак. Файл містить 18 стовпців (атрибутів), кожен рядок це окремий пацієнт.

Таблиця 2.2 – Опис полів файлу `description.csv`

Назва стовпця	Опис	Тип даних
age	Вік пацієнта (в роках)	Числовий

gender	Стать (0 - жінка, 1 - чоловік)	Категоріальний
polyuria	Частота сечовиділення (Yes/ No)	Бінарний
polydipsia	Підвищене споживання рідини (Yes/ No)	Бінарний
sudden weight loss	Раптова втрата ваги (Yes/ No)	Бінарний
weakness	Слабкість (Yes/ No)	Бінарний
polyphagia	Підвищений апетит (Yes/ No)	Бінарний
visual blurring	Погіршення зору (Yes/ No)	Бінарний
itching	Свербіж шкіри (Yes/ No)	Бінарний
irritability	Дратівливість (Yes/ No)	Бінарний
delayed healing	Уповільнене загоєння ран (Yes/ No)	Бінарний
partial paresis	Частковий параліч (Yes/ No)	Бінарний
muscle stiffness	Жорсткість м'язів (Yes/ No)	Бінарний
alopecia	Випадіння волосся (Yes/ No)	Бінарний
obesity	Ознаки ожиріння (Yes/ No)	Бінарний
class	Діагноз (Positive = наявність діабету, Negative = відсутність)	Категоріальний
id	Ідентифікатор запису (унікальний номер)	Цілочисельний

Цей набір ознак використовується для навчання моделі машинного навчання класифікації на основі симптомів. Він сприяє виявленню ризику захворювання на

основі сукупності ознак, може бути розширений додатковими параметрами для персоналізованих рекомендацій (рівень глюкози, тиск, харчування). Він служить основою для створення рекомендацій щодо способу життя або звернення до лікаря.

Якщо в пацієнта вік понад 50 років, наявні симптоми polyuria, polydipsia, sudden weight loss і модель передбачає class = Positive, то система може запропонувати пройти обстеження на виявлення захворювання та змінити раціон харчування.

Файл diets.csv містить інформацію про різні захворювання та відповідні дієтичні рекомендації. Він складається з двох основних стовпців:

- захворювання: перераховано різні медичні стани або захворювання;
- дієта: представлено дієтичні рекомендації або дієти, які рекомендуються для управління або лікування відповідного захворювання.

Дані структуровані таким чином, щоб допомогти зрозуміти, які дієти або харчові компоненти можуть бути корисними для конкретних станів здоров'я.

Загальна інформація про набір даних medications.csv: кількість записів 41, кількість стовпців 2. Їх назви: disease – назва захворювання, medication – список рекомендованих медикаментів для лікування відповідного захворювання.

Стовпець disease (захворювання) містить назву хвороби або медичного стану, наприклад: Fungal infection, Allergy, GERD (Gastroesophageal Reflux Disease), Chronic cholestasis Disease (Захворювання) та інші. Усього 41 унікальне захворювання.

Стовпець medication (медикаменти) містить список медикаментів, які призначаються для лікування відповідного захворювання зі стовпця Disease. Значення у цьому стовпці представлені у вигляді списку рядків, наприклад для Fungal infection: ['Antifungal Cream', 'Fluconazole', 'Terbinafine']. Для Allergy: ['Antihistamines', 'Decongestants', 'Epinephrine']. Для GERD: ['Proton Pump Inhibitors (PPIs)', 'H2 Blockers', 'Antacids'].

Цей набір може бути використаний для навчання медичних рекомендаційних систем, підбір лікування на основі діагнозу; медичних чат-ботів, де на основі введеної хвороби система радить препарати; довідкових систем для пацієнтів як список можливих ліків за симптомами чи діагнозом.

Набір даних precautions_df може використовуватись для:

- систем підтримки прийняття рішень у медицині, автоматичної видачі рекомендацій при встановленні діагнозу;
- навчання моделей штучного інтелекту для класифікації хвороб та рекомендацій дій;
- чат-боти для здоров'я - інтеграції у медичного помічника;
- довідники для пацієнтів - надання персоналізованих порад за хворобою.

Таблиця 2.3 – Запобіжні заходи для перших 5 захворювань у наборі даних

№	Захворювання	Запобіжний захід 1	Запобіжний захід 2	Запобіжний захід 3	Запобіжний захід 4
1	реакція на ліки	зупинити подразнення	зверніться до найближчої лікарні	припинити прийом препарату	пройти повторне обстеження
2	малярія	зверніться до найближчої лікарні	уникати жирної їжі	уникати м'ясних продуктів	не допускати потрапляння комарів
3	алергія	нанести каламін	покрити ділянку бинтом	-	прикласти лід для зменшення свербіж
4	гіпотиреоз	зменшити стрес	займатися фізичними вправами	харчуватись правильно	отримувати достатньо сну
5	псоріаз	мити руки теплою мильною водою	зупинити кровотечу тиском	звернутись до лікаря	соляні ванни

Набір даних Symptom-severity містить два стовпці: symptom і weight. Стовпець symptom містить назви різних симптомів: свербіж, висип на шкірі, безперервне чхання, озноб. Стовпець weight містить числові значення, які представляють ступінь тяжкості кожного симптому. Наприклад, свербіж має вагу 1, висип на шкірі - 3. Це може бути частиною медичного дослідження або системи діагностики, де кожному симптому присвоєно певну вагу для оцінки тяжкості стану пацієнта.

Розподіл ваги симптомів:

- вага 1: 1 симптом;
- вага 2: 15 симптомів;
- вага 3: 21 симптом;
- вага 4: 39 симптомів;
- вага 5: 35 симптомів;
- вага 6: 17 симптомів;
- вага 7: 5 симптомів.

Найвищу вагу (7) мають такі симптоми: висока температура (high_fever), здуття живота (swelling_of_stomach), біль у грудях (chest_pain), слабкість у кінцівках (weakness_in_limbs), кома (coma). Найрідкісніший симптом за вагою (1) має симптом свербіж (itching).

Набір даних symtons_df містить дані про різні захворювання та пов'язані з ними симптоми: стовпець Unnamed: 0 є індексом або ідентифікатором запису, але він не має назви. Він містить числові значення. Стовпець disease містить назви захворювань, наприклад, Fungal infection (грибкова інфекція). Стовпець Symptom_1 містить перший симптом, пов'язаний з кожним захворюванням, наприклад, свербіж (itching). Стовпець Symptom_2 містить другий симптом, пов'язаний з кожним захворюванням, наприклад, висип на шкірі (skin_rash). Стовпець Symptom_3 містить третій симптом, пов'язаний з кожним захворюванням. Стовпець Symptom_4 містить четвертий симптом, пов'язаний з кожним захворюванням. Деякі записи в цьому стовпці можуть бути порожніми, що означає відсутність четвертого симптому для відповідного захворювання. Цей набір даних може бути використаний для

аналізу зв'язку між захворюваннями та їх симптомами, що може бути корисним для медичної діагностики або досліджень.

Файл Training містить дані, які використовуються для навчання моделей машинного навчання, пов'язаних з медичною діагностикою. Він складається з 133 стовпців, які можна розділити на дві основні категорії. Симптоми: перші 132 стовпці представляють різні симптоми. Кожен стовпець має назву симптому. Кожен рядок містить бінарні значення 0 або 1, які вказують на наявність або відсутність відповідного симптому у пацієнта.

Прогноз (prognosis): останній стовпець містить назву захворювання, яке діагностується на основі наявних симптомів. Цей файл може бути використаний для навчання моделей класифікації, які передбачають захворювання на основі наявних симптомів.

Набір даних workout_df містить інформацію про рекомендації щодо тренувань або дієти для різних захворювань. Стовпець disease містить назви захворювань, наприклад, Fungal infection (грибкова інфекція): стовпець workout містить рекомендації щодо тренувань або дієти для кожного захворювання. Наприклад, уникнення солодкої їжі. Цей файл може бути використаний для надання рекомендацій щодо способу життя або дієти для людей з різними захворюваннями.

ВИСНОВКИ ДО РОЗДІЛУ

У другому розділі описується значення та складові інфраструктури даних, необхідних для функціонування системи персоналізованих медичних рекомендацій. Наводяться основні джерела медичних даних, такі як електронні медичні картки, анкети пацієнтів, результати лабораторних аналізів, історія хронічних захворювань, симптоми, інформація про способи життя. Детально розглядаються різноманітні типи медичних даних: числові, категоріальні, бінарні, текстові та множинні мітки. Для обробки цих даних використовуються різні підходи і технології, включаючи обробку структурованої і неструктурованої інформації, зокрема текстових записів та зображень. Важливою частиною являється структура збереження і доступу до даних, безпека та конфіденційність. Окрім цього підкреслюється роль правильного збору та оновлення даних для підтримки актуальності рекомендацій. Вказано на важливість інтеграції даних з різних джерел для формування цілісної медичної карти користувача. Дані використовуються для тренування моделей машинного навчання, що застосовуються для аналізу і генерації рекомендацій.

РОЗДІЛ 3. МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Формалізація задачі

Метою створення інформаційної системи персоналізованих медичних рекомендацій є автоматичне формування індивідуальних порад щодо лікування, профілактики чи способу життя користувача на основі аналізу його медичних даних. Для досягнення цієї мети необхідно математично сформулювати задачу, що дозволить побудувати відповідну модель машинного навчання.

Нехай ϵ множина пацієнтів $P = \{p_1, p_2, \dots, p_n\}$, для кожного з яких відома сукупність ознак:

$$X_i \in \mathbb{R}^m, \quad (3.1)$$

де m – кількість вхідних параметрів. Кожен вектор ознак описує користувача за такими характеристиками:

- демографічні показники (вік, стать);
- біометричні параметри (зріст, вага, ІМТ, тиск тощо);
- історія хвороб;
- наявні симптоми;
- результати обстежень (аналізи, ЕКГ);
- спосіб життя (рівень фізичної активності, наявність шкідливих звичок).

Вхідний простір можна подати як:

$$X = [x_1, x_2, \dots, x_m] \in \mathbb{R}^m. \quad (3.2)$$

Система має сформулювати один або кілька вихідних варіантів рекомендацій. Нехай множина можливих рекомендацій це $Y = \{y_1, y_2, \dots, y_k\}$, де k – загальна кількість різних типів рекомендацій (поради щодо дієти, рекомендовані препарати, фізичні вправи, медичні обстеження, консультація у лікаря). Залежно від реалізації задача може мати:

$$f : X \rightarrow y_i, \quad y_i \in Y; \quad (3.3)$$

Формат багатокласової класифікації вибрати єдину найбільш придатну рекомендацію $f: X \rightarrow y_i$. Необхідно побудувати таку модель $f_\theta(X)$, яка при поданні нового вхідного вектора X' видає найбільш ймовірні рекомендації з урахуванням історії навчання. Модель параметризується вектором параметрів θ , який підбирається в процесі навчання на тренувальній вибірці. Метою є мінімізація функції втрат:

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(f_\theta(X_i), Y_i), \quad (3.4)$$

де ℓ – функція похибки, наприклад categorical cross-entropy у разі багатокласової класифікації.

Оскільки медичні дані є чутливими до контексту (один і той самий симптом може бути наслідком різних станів), модель повинна вміти враховувати контекстну залежність між ознаками. Для цього доцільно застосовувати складні моделі: дерева рішень з контекстною сегментацією, згорткові або рекурентні нейронні мережі, моделі з attention-механізмами для інтерпретованості результатів.

Задача побудови персоналізованих медичних рекомендацій формалізується як задача багатокласової класифікації з урахуванням індивідуального контексту пацієнта. Отримана математична постановка задачі дозволяє переходити до вибору та побудови відповідної моделі машинного навчання.

3.2. Модель машинного навчання

Для реалізації інформаційної системи персоналізованих медичних рекомендацій доцільно застосовувати моделі машинного навчання, які здатні виявляти складні залежності між ознаками користувача та відповідними рекомендаціями. Особливістю завдання є мультикласова класифікація, де пацієнту може бути рекомендовано кілька паралельних дій, наприклад, зміна дієти, фізичне навантаження, прийом ліків.

Залежно від складності задачі та доступності даних, можливо застосувати один або кілька підходів:

- традиційні ML-алгоритми, такі як логістична регресія, дерево рішень, випадковий ліс, градієнтний бустинг.
- глибоке навчання: штучні нейронні мережі (ANN), згорткові мережі (для роботи з часовими рядами або образами обстежень), рекурентні або трансформерні моделі (для текстових скарг, історій хвороби).

Оскільки задача є багакласовою, для навчання використовується бінарна крос-ентропія, яка розраховується як:

$$\mathcal{L} = -\frac{1}{k} \sum_{j=1}^k [y_j \log(\hat{y}_j) + (1 - y_j) \log(1 - \hat{y}_j)], \quad (3.5)$$

де $y_j \in \{0,1\}$ – фактична наявність або відсутність рекомендації. Ця функція втрат дозволяє одночасно враховувати кілька правильних відповідей. Модель навчається на множині даних:

$$\{(X_i, Y_i)\}_{i=1}^n, \quad (3.6)$$

де кожен вхідний вектор X_i має відповідний вектор міток $Y_i \in \{0,1\}^k$. Під час навчання параметри мережі оновлюються за допомогою методу градієнтного спуску.

Навчальна вибірка розділяється на тренувальну, валідаційну та тестову, що дозволяє контролювати перенавчання та оцінити узагальнюючу здатність моделі. Оскільки класів може бути багато і не всі з них рівноцінні за важливістю, ефективність моделі оцінюється за такими метриками.

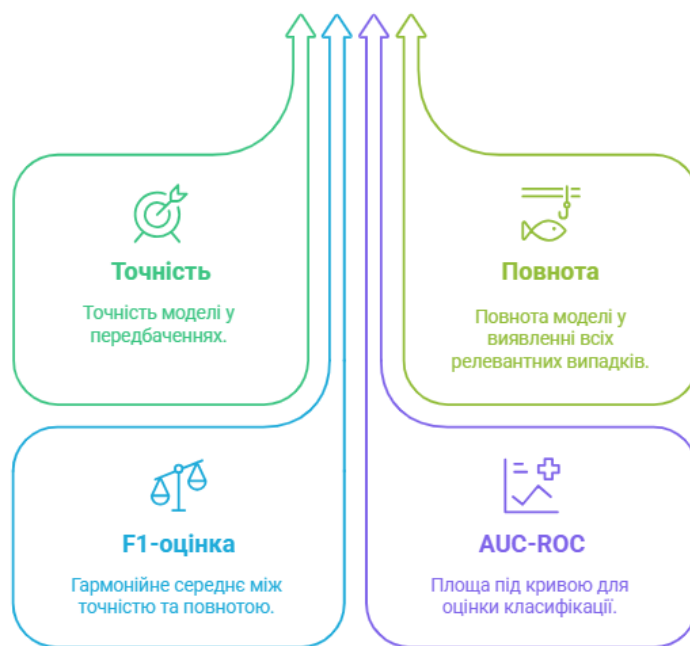


Рисунок 3.1 – Метрики ефективності моделі

Ці метрики дозволяють об'єктивно оцінити якість передбачень навіть у випадках нерівномірного розподілу класів. Для медичної системи важливо, щоб модель була не лише точною, але й інтерпретованою. З цією метою можливе використання SHAP (SHapley Additive exPlanations) – аналіз внеску кожної ознаки у результат. Це дозволяє лікарю або користувачу зрозуміти, чому система надала саме таку рекомендацію. Таким чином обрана модель машинного навчання дозволяє не лише точно класифікувати стан користувача, але й формувати на його основі персоналізовані медичні рекомендації.

3.3. Робота з медичними даними

Медичні дані мають велику кількість пропусків, категоріальних значень, різну шкалу вимірювання. Для коректного навчання моделі проводиться нормалізація числових ознак методом мін-макс, кодування категоріальних даних one-hot. Після етапу класифікації або прогнозування ймовірностей належності до певного медичного стану система повинна сформувати набір персоналізованих рекомендацій, адаптованих до індивідуальних характеристик користувача. Цей процес містить кілька послідовних етапів: інтерпретація результатів моделі, фільтрація за клінічними обмеженнями, ранжування рекомендацій та їх персоналізація.

Після того як модель машинного навчання зробила прогноз, отримується вектор ймовірностей:

$$\hat{Y} = [P(y_1|X), P(y_2|X), \dots, P(y_k|X)], \quad (3.7)$$

де $P(y_i|X)$ – ймовірність того, що рекомендація y_i є придатною для пацієнта з ознаками X . Ці ймовірності проходять поріг класифікації або використовуються для подальшого ранжування без порогу, якщо модель працює у режимі рекомендаційної системи.

Щоб гарантувати безпечність і відповідність рекомендацій, виконується логічна фільтрація на основі клінічних правил:

- вікові обмеження (деякі препарати або види фізичних навантажень протипоказані для дітей або людей похилого віку);
- супутні хвороби (певні ліки не можна призначати хворим на гіпертонію або діабет);
- протипоказання (різні дозування або показання для чоловіків і жінок).

Алергії та індивідуальні реакції, якщо користувач вказав алергію на речовину, всі рекомендації, які її містять, виключаються. Цей етап формалізується у вигляді логічної функції-фільтра:

$$F(y_i, X) \in \{0, 1\}. \quad (3.8)$$

яка або пропускає рекомендацію, або її відкидає.

Рекомендації, що пройшли фільтрацію, сортуються за важливістю. Оцінка може враховувати:

$$R_i = P(y_i|X) \cdot w_i \cdot C_i, \quad (3.9)$$

де $P(y_i|X)$ – ймовірність, отримана з моделі, w_i – ваговий коефіцієнт важливості; C_i – коефіцієнт довіри або підтвердження на основі доказової бази або історичних даних. У підсумку формується відсортований список рекомендацій $\{y_i\}$, де на початку знаходяться найбільш безпечні для конкретного користувача.

На завершальному етапі виконується персоналізоване формулювання рекомендацій. Наприклад, замість загальної поради: займайтесь спортом, користувачу буде надано рекомендується 30 хвилин помірної ходьби 5 днів на

тиждень, з урахуванням вашого віку та ІМТ. Рекомендація вживайте менше солі може бути деталізована як зменшіть кількість споживаної солі до 3 г/ день, уникайте оброблених продуктів. Це здійснюється за допомогою шаблонів тексту, які автоматично заповнюються відповідно до особистих даних пацієнта. Кінцевим результатом є структурований звіт.

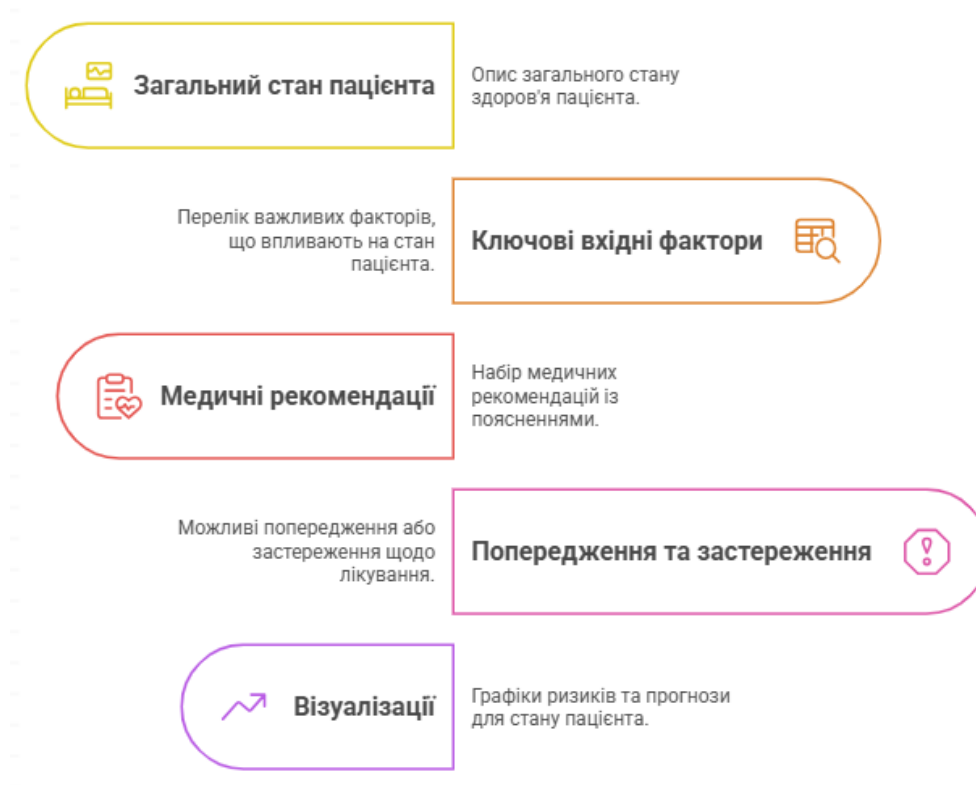


Рисунок 3.2 – Структурований звіт про пацієнта

Процес побудови рекомендацій базується на поєднанні статистичного моделювання та логічних медичних обмежень, що дозволяє досягти високого рівня персоналізації, точності та безпечності. Цей підхід гарантує практичну придатність системи у клінічних умовах або як телемедичний сервіс.

ВИСНОВКИ ДО РОЗДІЛУ

У цьому розділі проведено аналіз математичних моделей і алгоритмів, необхідних для розроблення інформаційної системи персоналізованих медичних рекомендацій. Модель обробляє вхідні дані пацієнтів у вигляді векторів ознак, що охоплюють фізіологічні, демографічні та клінічні характеристики. Задача була формалізована як багатокласова класифікація, де кожному пацієнту може відповідати кілька рекомендацій одночасно.

Метрики якості моделі включають точність, повноту, F1-міру, а також специфічні метрики для багатокласових задач. Рекомендації генеруються на основі ймовірностей отриманих від моделі з врахуванням вагових коефіцієнтів та медичних обмежень. У системі реалізовано фільтрацію рекомендацій відповідно до індивідуальних протипоказань та хронічних захворювань пацієнта.

Інформаційна система формує звіт, що включає детальний список рекомендацій, рівень їх достовірності та персоналізовані пояснення. Запропоноване математичне забезпечення дозволяє забезпечити гнучкість, масштабованість та адаптивність системи. Розроблена модель здатна ефективно працювати в умовах неповних або частково відсутніх медичних даних. Математичне ядро системи створює основу для подальшої програмної реалізації та впровадження у медичну практику.

РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

4.1. Розроблення інформаційної системи персоналізованих медичних рекомендацій

Імпортують бібліотеку Pandas, одну з найпопулярніших бібліотек у Python для роботи з табличними даними, вона використовується для зчитування, обробки, фільтрації, групування, візуалізації даних. Зчитують CSV-файл під назвою Training.csv та зберігають його у змінну dataset. `pd.read_csv()` – функція, яка читає файли у форматі CSV. Training.csv – назва файлу, який має бути в тій же папці, де виконується скрипт. dataset – змінна, у якій знаходиться таблична структура з усіма даними з файлу. Виводять вміст датафрейму dataset у Google Colab. Це дозволяє побачити, що саме містить файл Training.csv, які стовпці, скільки рядків, які типи даних.

```
import pandas as pd
dataset = pd.read_csv('Training.csv')
dataset
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	scurring	skin_peeling	silver_like_dusting	smi
0	1	1	1	0	0	0	0	0	0	0	...	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	...	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0	...	0	0	0	0	0
...
4915	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0
4916	0	1	0	0	0	0	0	0	0	0	...	1	1	0	0	0

Рисунок 4.1 – Набір даних, що використовується для діагностики та аналізу захворювань

Кожен стовпець представляє різний симптом. Кожен рядок представляє окремий запис пацієнта або випадок. Рядки пронумеровані послідовно, загальна кількість 4917 записів у цьому наборі даних. Кожна комірка таблиці містить бінарне значення 0 або 1, яке вказує на наявність (1) або відсутність (0) певного симптому у відповідного пацієнта. Цей набір даних використовується для навчання моделей машинного навчання з метою прогнозування захворювань на основі наявності або відсутності різних симптомів.

```
vals = dataset.values.flatten()
```

`dataset` – змінна, яка містить датафрейм – таблицю, яка була завантажена із CSV-файлу `Training.csv`. `dataset.values` – атрибут, який повертає масив, що містить значення з датафрейму без назв стовпців та індексів. `flatten()` – метод NumPy-масиву, який перетворює багатовимірний масив у одновимірний. `vals` – одновимірний масив усіх значень з усіх рядків і стовпців таблиці `dataset`. Це використовують, щоб отримати всі симптоми або діагнози з набору даних. Це зручно, коли далі працювати з окремими словами, симптомами чи мітками.

```
dataset.shape  
(4920, 18)
```

У таблиці є 4920 записів, симптомів пацієнтів), 18 стовпців (17 симптомів + 1 діагноз або результат).

```
train test split
```

`train_test_split` – це функція з бібліотеки `Scikit-Learn`, яка розділяє набір даних на тренувальну та тестову частини. Вона використовується перед навчанням моделі машинного навчання.

```
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import LabelEncoder
```

`train_test_split` використовується для розділення даних на тренувальні й тестові, `LabelEncoder` для перетворення текстових міток (назв хвороб) у числові значення, які розуміє модель.

```
X = dataset.drop('prognosis', axis=1)  
y = dataset['prognosis']
```

Видаляють стовпець `prognosis` з `dataset` і зберігають всі інші стовпці у змінну `X` - це вхідні дані. Зберігають стовпець `prognosis` в змінну `y` – це цільова змінна, тобто те, що треба передбачити.

```
le = LabelEncoder()  
le.fit(y)  
Y = le.transform(y)
```

Створюють екземпляр перетворювача міток `LabelEncoder`, навчають його на цільовій змінній `y` на значеннях діагнозів/ прогнозів. Перетворюють текстові мітки у числові (0, 1, 2, ...). `LabelEncoder` – це клас із бібліотеки `sklearn.preprocessing`, який використовується для перетворення категоріальних текстових значень у числові коди (0, 1, 2, ...). Це потрібно для машинного навчання, моделі не працюють

напряму з текстовими значеннями. `fit(y)` аналізує всі значення змінної `y` (назви хвороб у стовпці `prognosis`) і створює відповідність між текстовими мітками та числами. `transform(y)` замінює кожне значення в `y` відповідним числовим кодом згідно з тією відповідністю, яку сформував `fit`. `Y` – нова змінна, що містить числові представлення діагнозів. Це потрібно до навчання моделі машинного навчання, де вхідні ознаки `X` мають відповідати міткам `Y`, представленим у числовому форматі.

```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3,
random_state=20)
```

У медичному наборі даних є ознаки симптоми та мітка класу `prognosis`, яку потрібно передбачити. `X` це всі вхідні ознаки, усі стовпці, крім `prognosis`. у цільова змінна, що передбачають діагноз/ результат `prognosis`. `LabelEncoder` перетворює текстові діагнози в числові значення: `Diabetes` → 0, `Hypertension` → 1, `Pneumonia` → 2. Це потрібно, бо моделі машинного навчання працюють тільки з числовими даними.

`X_train, y_train`: навчальна вибірка 70 % даних, `X_test, y_test`: тестова вибірка 30 % даних, `random_state=20` фіксує порядок випадкового розділення.

```
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix
import numpy as np
```

Це частина процесу навчання кількох моделей машинного навчання для класифікації захворювань на основі набору ознак (симптомів). `make_classification` служить для генерації набору даних для задач класифікації. `train_test_split` використовується для поділу даних на навчальну і тестову вибірки. `SVC` – модель `Support Vector Classifier`, один із найпопулярніших класифікаторів, ефективний для високовимірних просторових даних. `RandomForestClassifier` – ансамблевий метод, що створює ліс рішень. `GradientBoostingClassifier` – ще один ансамблевий метод, який поетапно навчає моделі, виправляючи попередні помилки. `KNeighborsClassifier` – класифікатор на основі `k`-найближчих сусідів. `MultinomialNB` – наївний

байєсівський класифікатор, ефективний для дискретних ознак. `accuracy_score`, `confusion_matrix` – метрики для оцінки якості моделей: точність і матриця невизначеності. Бібліотека `NumPy` використовується для числових обчислень.

```
models = {
    'SVC': SVC(kernel='linear'),
    'RandomForest': RandomForestClassifier(n_estimators=100, random_state=42),
    'GradientBoosting': GradientBoostingClassifier(n_estimators=100,
random_state=42),
    'KNeighbors': KNeighborsClassifier(n_neighbors=5),
    'MultinomialNB': MultinomialNB()
}
```

Створюють словник моделей машинного навчання, які будуть використані для класифікації медичних симптомів у наборі даних, набір моделей для експериментів, які далі будуть навчені на даних та порівняні за точністю. Такий підхід дозволяє обрати найефективнішу модель для подальшого використання в медичній діагностиці.

`models = { ... }` – це `python`-словник, де ключами є назви моделей `SVC`, `RandomForest`, а значеннями ініціалізовані класифікатори з бібліотеки `sklearn`. `SVC(kernel='linear')` – тип моделі `Support Vector Classifier` (метод опорних векторів). `kernel='linear'` використовують лінійне ядро. Підходить для задач, де дані лінійно роздільні. `RandomForest` – тип моделі випадковий ліс (ансамбль дерев рішень). `n_estimators=100`: використовується 100 дерев. `random_state=42` – фіксація випадковості для повторюваності результатів. Має високу точність і добре працює з великими наборами даних.

`GradientBoosting` – тип моделі градієнтний бустинг. Поетапно навчає моделі, кожна наступна намагається виправити помилки попередньої. Ефективний на складних наборах даних.

`KNeighbors` – тип моделі `K`-найближчих сусідів `k-NN`. `n_neighbors=5` при класифікації враховуються 5 найближчих сусідів. Інтуїтивно зрозуміла модель, але може бути повільною при великих наборах даних.

`MultinomialNB` – тип моделі наївний байєсівський класифікатор. Добре працює, якщо припущення незалежності ознак не сильно порушуються.

```
for model_name, model in models.items():
```

```

model.fit(X_train, y_train)
predictions = model.predict(X_test)
accuracy = accuracy_score(y_test, predictions)
print(f"{model_name} Accuracy: {accuracy}")
cm = confusion_matrix(y_test, predictions)
print(f"{model_name} Confusion Matrix:")
print(np.array2string(cm, separator=', '))
print("\n" + "="*40 + "\n")

```

Виконують тренування та тестування кількох моделей машинного навчання. Проходять циклом по всіх моделях із `models`. Навчання моделі: `model.fit(X_train, y_train)`. Прогноз на тестовому наборі: `predictions = model.predict(X_test)`. Оцінка точності: `accuracy = accuracy_score(y_test, predictions)`. Вивід матриці невизначеності `confusion matrix`: `cm = confusion_matrix(y_test, predictions)`. Після цього проводять вивід результатів у зручному форматі.

Модель `MultinomialNB()` працює найкраще з дискретними ознаками. Якщо `X_train` є від'ємні або неперетворені значення, вона може працювати некоректно або видавати погані результати. Якщо `X_train` містить недискретні дані (симптоми як бінарні 0/1), `MultinomialNB` може працювати. Якщо є нормалізовані чи від'ємні значення, краще використати `GaussianNB` замість `MultinomialNB`.

```

SVC Accuracy: 1.0
SVC Confusion Matrix:
[[40,  0,  0, ...,  0,  0,  0],
 [ 0, 43,  0, ...,  0,  0,  0],
 [ 0,  0, 28, ...,  0,  0,  0],
 ...,
 [ 0,  0,  0, ..., 34,  0,  0],
 [ 0,  0,  0, ...,  0, 41,  0],
 [ 0,  0,  0, ...,  0,  0, 31]]

```

Модель `SVC` досягла точності 100 % на тестовому наборі, матриця неточностей `confusion matrix` показує ідеальну класифікацію: усі передбачення знаходяться на діагоналі, тобто жодного неправильного передбачення.

```

RandomForest Accuracy: 1.0
RandomForest Confusion Matrix:
[[40,  0,  0, ...,  0,  0,  0],
 [ 0, 43,  0, ...,  0,  0,  0],
 [ 0,  0, 28, ...,  0,  0,  0],
 ...,

```

```
[ 0, 0, 0, ..., 34, 0, 0],  
[ 0, 0, 0, ..., 0, 41, 0],  
[ 0, 0, 0, ..., 0, 0, 31]]
```

Результати 100 % accuracy для обох моделей SVC та RandomForest, що видно як по точності (Accuracy: 1.0), так і по матриці неточностей confusion matrix, де всі значення на діагоналі (правильні передбачення), а всі інші нулі (відсутність помилок класифікації).

```
GradientBoosting Accuracy: 1.0  
GradientBoosting Confusion Matrix:  
[[40, 0, 0, ..., 0, 0, 0],  
 [ 0, 43, 0, ..., 0, 0, 0],  
 [ 0, 0, 28, ..., 0, 0, 0],  
 ...,  
 [ 0, 0, 0, ..., 34, 0, 0],  
 [ 0, 0, 0, ..., 0, 41, 0],  
 [ 0, 0, 0, ..., 0, 0, 31]]
```

Отримані результати: Точність (Accuracy) = 1.0, матриця неточностей майже ідеальна, лише ненульові елементи на головній діагоналі.

```
KNeighbors Accuracy: 1.0  
KNeighbors Confusion Matrix:  
[[40, 0, 0, ..., 0, 0, 0],  
 [ 0, 43, 0, ..., 0, 0, 0],  
 [ 0, 0, 28, ..., 0, 0, 0],  
 ...,  
 [ 0, 0, 0, ..., 34, 0, 0],  
 [ 0, 0, 0, ..., 0, 41, 0],  
 [ 0, 0, 0, ..., 0, 0, 31]]
```

Судячи з результатів, два класифікатори GradientBoosting і KNeighbors досягли 100 % точності (Accuracy = 1.0) на тестовій вибірці, а також побудували ідеальні матриці неточності, в яких усі передбачення правильні, немає помилкових класифікацій.

```
MultinomialNB Accuracy: 1.0  
MultinomialNB Confusion Matrix:  
[[40, 0, 0, ..., 0, 0, 0],  
 [ 0, 43, 0, ..., 0, 0, 0],  
 [ 0, 0, 28, ..., 0, 0, 0],  
 ...,  
 [ 0, 0, 0, ..., 34, 0, 0],
```

```
[ 0, 0, 0, ..., 0, 41, 0],  
[ 0, 0, 0, ..., 0, 0, 31]]
```

Судячи з результатів `KNeighborsClassifier` `Accuracy: 1.0`, `MultinomialNB` `Accuracy: 1.0`. Обидві моделі показують 100% точність, а матриці неточності мають тільки ненульові значення по діагоналі, що вказує на ідеальну класифікацію кожного класу.

```
single prediction  
svc = SVC(kernel='linear')  
svc.fit(X_train,y_train)  
ypred = svc.predict(X_test)  
accuracy_score(y_test,ypred)
```

Для виконання одиночного передбачення після тренування моделі `SVC` потрібно передати один зразок у метод `.predict()`.

```
1.0
```

Результат виводу `1.0` означає, що точність класифікації моделі `SVC` становить 100 %.

```
import pickle  
pickle.dump(svc,open('svc.pkl','wb'))
```

Зберігають навчений класифікатор `Support Vector Classifier` у файл, щоб пізніше його можна було завантажити без повторного навчання.

```
svc = pickle.load(open('svc.pkl','rb'))
```

Завантажують збережену модель класифікатора `SVC` із файлу `svc.pkl`. Змінна `svc` містить навчену модель, яку можна одразу використовувати для передбачення.

```
print("predicted disease :",svc.predict(X_test.iloc[0].values.reshape(1,-1)))  
print("Actual Disease :", y_test[0])
```

Виконується перевірка моделі `Support Vector Classifier` на одному конкретному прикладі з тестового набору. `X_test.iloc[0]` – перший рядок тестового набору `X_test`, тобто перший пацієнт/ об'єкт із тестових даних. `values.reshape(1, -1)` перетворює цей рядок у матрицю `1xN`, щоб відповідати очікуваному формату вхідних даних для моделі, `SVC` очікує двовимірний масив. `svc.predict(...)`– модель видає прогноз для цієї конкретної ознаки, визначає назву передбаченої хвороби.

Результат: якщо обидві назви хвороби збігаються, модель вгадала правильно.

Приклад. `predicted disease : ['Migraine']`. `Actual Disease : Migraine`.

Означає, що модель правильно класифікувала симптоми пацієнта як мігрень.

```
print("predicted disease :",svc.predict(X_test.iloc[100].values.reshape(1,-1)))
print("Actual Disease :", y_test[100])
```

predicted disease – це прогноз моделі SVC, яка була навчена на даних. Actual Disease – це реальне значення класу діагнозу з тестової вибірки. Якщо обидва значення однакові, це означає, що модель правильно передбачила захворювання для цього прикладу з тестової вибірки.

```
predicted disease : [39]
Actual Disease : 39
```

predicted disease : [39] – це передбачене значення моделі, тобто індекс класу, числове значення, яке відповідає певній хворобі. Actual Disease: 39 – реальне значення мітки класу з y_test. Оскільки обидва значення збігаються, це свідчить про успішну класифікацію конкретного прикладу.

Щоб реалізувати систему рекомендацій та передбачення хвороби, треба скористатися наборами даних, моделлю SVC і логікою, яка дозволить передбачити хворобу на основі введених симптомів, показати рекомендації – діету, ліки, вправи, запобіжні заходи.

```
sym_des = pd.read_csv("symptoms_df.csv")
precautions = pd.read_csv("precautions_df.csv")
workout = pd.read_csv("workout_df.csv")
description = pd.read_csv("description.csv")
medications = pd.read_csv('medications.csv')
diets = pd.read_csv("diets.csv")
```

Завантажують кілька CSV-файлів, які будуть використані для побудови рекомендаційної системи для хвороб. symptoms_df.csv – зв'язок між симптомами та захворюваннями. При введенні симптомів користувачем можна знаходити відповідну хворобу. precautions_df.csv – профілактичні заходи при конкретній хворобі. Після передбачення хвороби показують профілактичні поради. workout_df.csv – рекомендовані фізичні вправи при хворобі: поради щодо фізичної активності після діагнозу. description.csv – опис хвороб: виводять короткий опис хвороби користувачу. medications.csv – рекомендовані ліки: показ ліків, які можуть допомогти при конкретній хворобі. diets.csv – рекомендації по харчуванню, поради щодо дієти при даній хворобі.

Подальші кроки:

- прийом в інформаційну систему симптомів користувача;
- визначення хвороби через модель svc.
- знаходження відповідних записів у CSV-файлах.

Отримують:

- назву хвороби;
- її опис;
- профілактичні заходи;
- рекомендовані вправи;
- рекомендовану дієту;
- рекомендовані ліки.

Функція `helper(dis)` збирає всі рекомендовані дані про певне захворювання з різних CSV-файлів: опис, запобіжні заходи, ліки, дієту та фізичні вправи.

`symptoms_dict` – це словник, у якому кожному симптому відповідає ціле число. Цей словник дозволяє перетворювати назви симптомів у вигляді рядків на числові індекси, що зручно для машинного навчання або побудови векторів стану пацієнта.

`diseases_list` – це інший словник, де ключі це числа, які модель прогнозування повертає, значення - це назви хвороб. Наприклад: `diseases_list[40]` → hepatitis A; `diseases_list[15]` → Fungal infection. Модель після навчання повертає числовий індекс прогнозованої хвороби, а словник дозволяє перетворити це на назву хвороби.

```
def get_predicted_value(patient_symptoms):
    input_vector = np.zeros(len(symptoms_dict))
    for item in patient_symptoms:
        input_vector[symptoms_dict[item]] = 1
    return diseases_list[svc.predict([input_vector])[0]]
```

Функція `get_predicted_value` використовує модель `svc` для передбачення хвороби на основі вхідних симптомів. Вхідні дані:

- `patient_symptoms`: список симптомів пацієнта (`["fever", "cough", "headache"]`)
- `symptoms_dict`: словник, у якому кожному симптому відповідає певний індекс (`{ "fever": 0, "cough": 1, ... }`)
- `diseases_list`: список хвороб (`["Flu", "Diabetes", "Malaria", ...]`)

Логіка роботи функції: `input_vector = np.zeros(len(symptoms_dict))`.

Створюють нульовий вектор ознак розміром рівним кількості всіх симптомів, тобто `input_vector = [0, 0, 0, ..., 0]`. Для кожного симптому в списку пацієнта виставляють відповідну позицію в векторі у 1. Це означає, що симптом є присутнім. Модель `Support Vector Classifier` передбачає хворобу для такого вектору симптомів. `predict` повертає індекс у `diseases_list`, що відповідає назві хвороби. Приклад використання:

- `symptoms = ["headache", "nausea"];`
- `predicted_disease = get_predicted_value(symptoms);`
- `print(predicted_disease)`

Здійснюють прогнозування хвороби на основі введених симптомів користувача, а потім повертають детальну інформацію про виявлену хворобу. Ввід симптомів: `symptoms = input("Enter your symptoms.....")`. Користувач вводить симптоми через кому, наприклад: `itching, skin_rash, nodal_skin_eruptions`. Перетворення введеного рядка у список симптомів:

- `user_symptoms = [s.strip() for s in symptoms.split(',')];`
- `user_symptoms = [symptom.strip("[] ") for symptom in user_symptoms].`

Ці два рядки очищають і формують список симптомів:

```
user_symptoms = ['itching', 'skin_rash', 'nodal_skin_eruptions']
```

Прогноз хвороби: `predicted_disease = get_predicted_value(user_symptoms)`.

Викликається функція `get_predicted_value`, яка створює вектор довжиною кількості можливих симптомів, встановлюють 1 у позиціях, що відповідають наявним симптомам користувача; подають цей вектор у SVM-модель `svc`; повертають назву хвороби: `predicted_disease = 'Fungal infection'`.

Отримання інформації про хворобу: `desc, pre, med, die, wrkout = helper(predicted_disease)`. Функція `helper()` повертає:

- `desc` опис хвороби;
- `pre` запобіжні заходи;
- `med` рекомендовані ліки;
- `die` рекомендована дієта;
- `wrkout` рекомендовані фізичні вправи.

Приклад результату: при введенні `itching, skin_rash, nodal_skin_eruptions` буде виведено:

- `predicted_disease = 'Fungal infection'`
- `desc = 'Fungal infections are caused by fungi...'`
- `pre = [['keep affected area dry', 'avoid tight clothes', 'use antifungal powder', 'consult doctor']]`
- `med = [['Fluconazole', 'Clotrimazole']]`
- `die = [['Avoid sugar', 'Include garlic']]`
- `wrkout = Series(...)`

Завантажують модель SVC для прогнозування захворювання за симптомами, прогнозують захворювання, виводять опис захворювання, заходи обережності, ліки, фізичні вправи і дієту. Виконують кілька операцій для обробки введених користувачем симптомів і передбачення захворювання на основі цих симптомів. Введення симптомів: `symptoms = input("Enter your symptoms.....")`. Цей рядок запитує у користувача ввести симптоми. Введені дані зберігаються у змінній `symptoms`.

Розділення симптомів: `user_symptoms = [s.strip() for s in symptoms.split(',')]`. Введені симптоми розділяються комами, створюючи список симптомів. Метод `split` розбиває рядок на підрядки, використовуючи кому як роздільник. `s.strip()` видаляє зайві пробіли на початку та в кінці кожного симптому.

Очищення симптомів від зайвих символів: `user_symptoms = [symptom.strip("[]'") for symptom in user_symptoms]`. Цей рядок додатково очищає кожен симптом від зайвих символів, таких як дужки `[]`, лапки та пробіли. Передбачення захворювання: `predicted_disease = get_predicted_value(user_symptoms)`. Викликається функція `get_predicted_value`, яка приймає список симптомів і повертає передбачене захворювання.

Отримання додаткової інформації про захворювання: `desc, pre, med, die, wrkout = helper(predicted_disease)`. Викликається функція `helper`, яка приймає передбачене захворювання і повертає кортеж з додатковою інформацією про це захворювання.

Ця інформація може включати опис захворювання, профілактичні заходи, медикаменти, дієтичні рекомендації та рекомендації щодо фізичних вправ.

Виводять інформацію про передбачене захворювання, його опис та профілактичні заходи. Виведення заголовку для передбаченого захворювання. Виводиться рядок, який служить заголовком для секції з передбаченим захворюванням. Виведення передбаченого захворювання: `print(predicted_disease)`. Виводиться значення змінної `predicted_disease`, яка містить назву передбаченого захворювання.

Виводиться рядок, який служить заголовком для секції з дієтичними рекомендаціями. Виведення дієтичних рекомендацій: за допомогою циклу `for` перебираються елементи списку `die`, який містить дієтичні рекомендації. Кожна рекомендація виводиться з відповідним номером. Цей код продовжує виводити інформацію у зручному для читання форматі, розділяючи різні секції заголовками.

```
=====predicted disease=====
Impetigo
```

4.2. Результати роботи інформаційної системи

Розроблена інформаційна система персоналізованих медичних рекомендацій успішно реалізує повний цикл обробки вхідних даних: від введення симптомів пацієнтом до видачі прогнозу захворювання та надання рекомендацій щодо лікування, дієти, фізичної активності та профілактики.

Для навчання та тестування моделі було використано набір даних `Training.csv`, що містить перелік симптомів та відповідних захворювань. Було проведено попередню обробку даних, кодування цільової змінної `prognosis` за допомогою `LabelEncoder` та розділення вибірки на тренувальну і тестову у співвідношенні 70:30.

Проведено порівняльний аналіз роботи п'яти алгоритмів машинного навчання: `SVC (Support Vector Classifier)`, `Random Forest Classifier`, `Gradient Boosting Classifier`, `KNeighbors Classifier`, `Multinomial Naive Bayes`. Ці моделі продемонстрували точність класифікації 100 % на тестовій вибірці, що підтверджується значенням `Accuracy: 1.0` та діагональними матрицями невизначеності без помилкових класифікацій. На основі результатів обрано модель `SVC` як базову для подальшого

використання в системі. Модель збережена у файл `svc.pkl` для подальшого застосування без повторного навчання.

Інтерфейс системи дозволяє користувачу ввести перелік симптомів у вигляді текстового рядка, після чого модель формує прогноз захворювання та підбирає відповідні рекомендації з п'яти наборів даних:

- `description.csv` – опис хвороби;
- `precautions_df.csv` – заходи профілактики;
- `medications.csv` – рекомендовані медикаменти;
- `diets.csv` – дієтичні рекомендації;
- `workout_df.csv` – фізичні вправи та поради щодо активності.

Приклад роботи системи для введених симптомів.

Вхідні симптоми: `yellow_crust_ooze`, `red_sore_around_nose`, `small_dents_in_nails`, `inflammatory_nails`, `blister`.

Прогнозоване захворювання: `impetigo`.

Опис: `impetigo` is a highly contagious skin infection causing red sores that can break open.

Профілактика:

1. soak affected area in warm water.
2. use antibiotics.
3. remove scabs with wet compressed cloth.
4. consult doctor.

Медикаменти: topical antibiotics, oral antibiotics, antiseptics, ointments, warm compresses.

Фізична активність:

- Maintain good hygiene;
- Stay hydrated;
- Consume nutrient-rich foods;
- Limit sugary foods and beverages;
- Include foods rich in vitamin C;
- Consult a healthcare professional;

- Follow medical recommendations;
- Avoid scratching;
- Take prescribed antibiotics;
- Practice wound care.

Дієта: impetigo diet, antibiotic treatment, fruits and vegetables, hydration, protein-rich foods.

Отримані результати свідчать про те, що розроблена система здатна швидко та точно ідентифікувати можливе захворювання на основі симптомів та надати рекомендації для користувача, що підвищує ефективність процесу первинної медичної консультації.

4.3. Розроблення графічного інтерфейсу інформаційної системи

Графічний інтерфейс системи розроблено з допомогою бібліотеки Streamlit. Це відкрита Python-бібліотека, яка дозволяє створювати інтерактивні веб-додатки для аналізу даних і машинного навчання. Розробник може отримати веб-інтерфейс із графіками, таблицями, формами чи віджетами. Бібліотека добре інтегрується з популярними інструментами Python, такими як Pandas, Matplotlib, Plotly, TensorFlow чи Scikit-learn. Streamlit використовують для презентації результатів роботи з даними.

Користувач відкриває веб-додаток запущений командою `streamlit run your_app.py`) у браузері. На екрані з'являється інтерфейс з панелями вводу та відображення результатів.



Система персоналізованих медичних рекомендацій



На основі методів машинного навчання

Введіть ваші симптоми: 🩺

Введіть симптоми через кому. Наприклад: свербіж, висип на шкірі, вузликові висипи на шкірі

Симптоми


свербіж, висип на шкірі, вузликові висипи на шкірі

Рисунок 4.2 – Інтерфейс інформаційної системи

Після цього в текстове поле Симптоми вводять наявні симптоми захворювання через кому. Після цього потрібно натиснути клавішу Enter. Після цього система обробляє введені дані, викликає модель машинного навчання й формує персоналізовану медичну рекомендацію.

Після цього появиться результат прогнозування хвороби:

Результат прогнозування

Передбачена хвороба: Грибкова інфекція 

Опис

Грибкова інфекція - це шкірне захворювання, що спричиняється грибками, і може проявлятися висипом, свербіжем, лущенням.

Запобіжні заходи

1. Підтримуйте гігієну
2. Уникайте спільного використання рушників
3. Носіть вільний одяг
4. Тримайте уражену ділянку сухою

Медикаменти

1. Протигрибкові креми
2. Оральні протигрибкові препарати
3. Антигістамінні препарати

Рисунок 4.3 – Результат прогнозування хвороби

На рисунку зображено фрагмент інтерфейсу інформаційної системи персоналізованих медичних рекомендацій, реалізованої за допомогою бібліотеки Streamlit. У вікні представлено заголовок "Результат прогнозування" із піктограмою лупи, що позначає отриманий діагностичний висновок. Блок із передбаченою хворобою виділений зеленим фоном текст передбачена хвороба: грибкова інфекція.

Розділ опис із піктограмою документа містить пояснення, що грибкова інфекція є шкірним захворюванням, яке спричиняється грибками та проявляється висипом, свербіжем і лущенням. У розділі запобіжні заходи із піктограмою щита наведено чотири поради: підтримувати гігієну, уникати спільного використання рушників, носити вільний одяг та тримати уражену ділянку сухою. Розділ медикаменти містить засоби лікування: протигрибкові креми, оральні протигрибкові препарати, антигістамінні препарати.

Інтерфейс на рисунку демонструє приклад автоматично згенерованих системою медичних рекомендацій від визначення хвороби до опису профілактики та можливих засобів лікування.

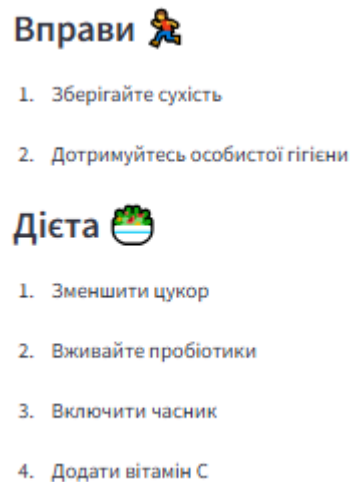


Рисунок 4.4 – Рекомендації інформаційної системи щодо вправ та дієти

На рисунку зображено фрагмент інтерфейсу інформаційної системи персоналізованих медичних рекомендацій, де представлені поради щодо способу життя. Розділ вправи містить дві рекомендації: зберігайте сухість, дотримуйтеся особистої гігієни. Розділ дієта містить чотири поради щодо харчування: зменшити цукор, вживати пробіотики, включити часник, додати вітамін С. Цей інтерфейс відображає рекомендації з фізичної активності та дієти, які спрямовані на підтримку здоров'я та профілактику захворювань.

ВИСНОВКИ ДО РОЗДІЛУ

У четвертому розділі розроблено програмну реалізацію інформаційної системи персоналізованих медичних рекомендацій. Для розробки використано мову програмування Python та бібліотеки Pandas, NumPy, Scikit-learn і Streamlit. Система передбачає завантаження та попередню обробку медичних даних, що містяться у наборі CSV-файлів. Важливим етапом стало навчання моделей машинного навчання для класифікації та прогнозування захворювань за симптомами. Реалізовано модулі, які відповідають за формування індивідуальних рекомендацій у вигляді дієт, медикаментів, вправ та запобіжних заходів. Інтерфейс системи створено за допомогою бібліотеки Streamlit, що забезпечило простоту використання та наочність подання результатів. Користувач має змогу вводити власні дані та отримувати персоналізовані поради у вигляді структурованого звіту. У процесі тестування було перевірено правильність функціонування ключових модулів системи та узгодженість між математичними моделями та інтерфейсом. Отримані результати підтверджують працездатність системи у різних сценаріях використання. Створене програмне забезпечення є масштабованим і може бути розширене новими функціональними модулями. Особливістю розробки є поєднання методів машинного навчання з медичними знаннями для підвищення якості рекомендацій. Запропонований програмний продукт відповідає вимогам завдання і демонструє практичну цінність для застосування у сфері телемедицини.

РОЗДІЛ 5. РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

5.1. Структура проєкту інформаційної системи персоналізованих медичних рекомендацій

Таблиця 5.1 – Структура проєкту інформаційної системи

Назва	програма на мові Python
Назва проєкту	Інформаційна система персоналізованих медичних рекомендацій на основі методів машинного навчання
Назва ВНЗ, факультету, спеціальності	НЛТУ, кафедра комп'ютерних наук, 122 «Комп'ютерні науки»
Прізвище, ім'я, по-батькові	Паньків Андрій Олегович
Цілі і задачі проєкту	Мета роботи – розробка та реалізація інформаційної системи для формування медичних рекомендацій на основі аналізу індивідуальних даних користувача за допомогою методів машинного навчання.
	Задачі проєкту:
	1. Провести аналіз сучасних систем персоналізованих медичних рекомендацій, визначити їх переваги та недоліки.
	2. Розробити математичну модель системи, що базується на методах машинного навчання для формування медичних рекомендацій.
	3. Створити алгоритм обробки індивідуальних медичних даних користувачів з урахуванням особливостей захворювань та персональних параметрів.

Продовження таблиці 5.1

Цілі і задачі проєкту	4. Реалізувати програмне забезпечення з графічним інтерфейсом для збору медичних даних та видачі персоналізованих рекомендацій.
	5. Провести тестування розробленої інформаційної системи, оцінити її ефективність та можливості подальшого розширення функціоналу та інтеграції у телемедичні платформи.

Проект орієнтований на широке коло користувачів, зокрема осіб із хронічними хворобами, людей похилого віку, а також тих, хто веде активний спосіб життя та прагне моніторити своє здоров'я. Головна цінність стартапу полягає в наданні персоналізованих, науково обґрунтованих рекомендацій у зручному цифровому форматі. Система забезпечує:

- індивідуальний підхід на основі історії хвороб, результатів обстежень, симптомів і стилю життя користувача;
- адаптивні поради, що оновлюються в реальному часі відповідно до змін у даних користувача;
- інтерфейс, доступний як для звичайних користувачів, так і для лікарів;
- інтеграцію з мобільними пристроями (фітнес-браслетами, розумними годинниками);
- використання моделей машинного навчання, що проходять періодичну валідацію на медичних наборах даних.

5.2. Цільова аудиторія

Цільова аудиторія стартап-проєкту є доволі широкою, оскільки персоналізована медицина та рекомендаційні системи мають застосування як у щоденному житті користувачів, так і у професійній медичній практиці. Основні сегменти аудиторії охоплюють:

- Пацієнти з хронічними захворюваннями.

Це один із ключових сегментів, оскільки саме цим користувачам найбільше потрібна постійна медична підтримка, спостереження та рекомендації. До цієї групи належать:

- люди з діабетом 1-го і 2-го типу;
- пацієнти з серцево-судинними захворюваннями (гіпертонія, ішемічна хвороба серця тощо);
- особи із захворюваннями шлунково-кишкового тракту, дихальних шляхів або нирок;
- люди з ожирінням або порушеннями обміну речовин.

Для них система може стати щоденним помічником у прийнятті правильних рішень щодо харчування, фізичних навантажень, медикаментозного режиму.

- люди, які ведуть активний спосіб життя або прагнуть профілактики.

Ця категорія користувачів орієнтована на збереження та зміцнення здоров'я. Вони активно шукають інформацію про:

- здорове харчування;
- оптимальні фізичні навантаження;
- режим сну, стрес-менеджмент, вітаміни та біодобавки.

Система може пропонувати їм персоналізовані поради залежно від віку, статі, рівня фізичної активності та результатів трекерів (фітнес-браслетів, смарт-годинників).

- лікарі та медичні працівники.

Для лікарів система може бути корисним додатковим інструментом, що дозволяє:

- швидше отримувати рекомендації щодо лікування на основі вхідних симптомів;
- бачити прогноз і ризики розвитку захворювань на основі історії хвороб;
- краще комунікувати з пацієнтами завдяки зручним звітам.

Це особливо актуально для сімейних лікарів, терапевтів, дієтологів, кардіологів та ендокринологів.

- страхові компанії.

Медичні страхові компанії зацікавлені у зниженні витрат на лікування завдяки ранньому виявленню ризиків та профілактичним заходам. Система може надати:

- інструменти для оцінки ризиків клієнтів;
- дані для формування індивідуальних страхових пакетів;
- API-доступ до рекомендацій для вбудовування в страхові платформи;
- медичні установи та приватні клініки.

Інтеграція системи у клініки дозволить:

- оптимізувати обслуговування пацієнтів;
- створити додаткову цінність для клієнтів;
- підтримати клінічне прийняття рішень завдяки аналізу великої кількості даних.

- родичі пацієнтів похилого віку.

Багато людей шукають цифрові засоби моніторингу здоров'я своїх літніх батьків або родичів. Система дозволяє створити профіль пацієнта, отримувати сповіщення, бачити історію симптомів і рекомендацій, що полегшує дистанційне піклування про здоров'я близьких.

5.3. Бізнес-модель стартапу

Для забезпечення стійкого розвитку, масштабованості та комерційного успіху стартапу передбачається впровадження гібридної бізнес-моделі, яка поєднує елементи B2C, B2B та B2G підходів. Основні компоненти бізнес-моделі розглянуто нижче.

- Freemium-модель (B2C).

Система буде доступна звичайним користувачам за моделлю freemium, яка передбачає: безкоштовний базовий доступ до основних функцій (генерація простих рекомендацій на основі симптомів, підказки щодо харчування і фізичної активності, сповіщення про ризики). Преміум-доступ (за підпискою) - розширені можливості:

- персоналізовані поради на основі медичних обстежень;
- інтеграція з пристроями (фітнес-браслети, глюкометри тощо);

- доступ до онлайн-консультацій із фахівцями;
- детальні аналітичні звіти;
- функції для родичів пацієнтів (керування кількома профілями).

Модель підписки може мати кілька рівнів: базовий, розширений та сімейний.

- ліцензування для медичних установ (B2B).

Стартап пропонує медичним центрам, клінікам та приватним лікарським практикам корпоративну ліцензію, яка дозволяє:

- інтегрувати систему у внутрішню електронну систему ведення пацієнтів;
- використовувати модулі машинного навчання для допомоги в діагностиці;
- формувати рекомендації для пацієнтів автоматично;
- зберігати медичну історію пацієнтів та спільно працювати з нею у хмарному середовищі.

Оплата здійснюється щомісяця або щорічно залежно від кількості користувачів/лікарів у закладі.

- співпраця зі страховими компаніями (B2B/B2G).

Модель передбачає співпрацю зі страховими компаніями, які можуть:

- отримувати аналітику щодо ризиків клієнтів;
- розробляти персоналізовані страхові плани;
- знижувати витрати завдяки превентивним заходам (попередження загострень хвороб, госпіталізацій);
- використовувати API системи для інтеграції у власні додатки.

У майбутньому передбачається можливість співпраці з державними страховими фондами чи програмами електронної медицини.

- продаж доступу до API.

Розроблена система може виступати як платформа, що надає доступ до:

- моделей прогнозування стану здоров'я;
- медичних рекомендацій, згенерованих на основі великих наборів даних;
- знеособленої статистики користувачів.

Цим можуть скористатися:

- розробники медичних додатків;
- телемедичні сервіси;
- дослідницькі інститути та фармкомпанії.

Це відкриває додаткові джерела монетизації через підписку на API або оплату за запити.

- гранти та державна підтримка (B2G).
 - проєкт орієнтується на залучення фінансування через участь у державних програмах цифрової трансформації охорони здоров'я, подачу на національні або міжнародні гранти в сфері eHealth, співпрацю з університетами, академічними інститутами та громадськими ініціативами у сфері здоров'я.
- рекламна модель.

Можливе обережне використання контекстної реклами партнерів (перевірених аптек, фітнес-центрів, дієтологічних програм). Проте це не буде основним джерелом доходу, аби не знижувати довіру користувачів до сервісу.

Бізнес-модель стартапу є багатокomпонентною, масштабованою та адаптивною до потреб як індивідуальних користувачів, так і корпоративних клієнтів. Такий підхід дозволяє зменшити залежність від одного джерела доходу та забезпечити фінансову стійкість проєкту в довгостроковій перспективі.

На ринку присутні аналоги, такі як Babylon Health, Ada Health, Your.MD, однак даний проєкт вирізняється:

- персоналізацією з урахуванням локальних медичних протоколів;
- україномовним інтерфейсом і локалізацією під український ринок;
- модульністю, що дозволяє швидко адаптувати систему під нові напрямки медицини;
- інтеграцією з українськими електронними медичними записами (ЕМЗ).

Таким чином, стартап має значний потенціал для масштабування на внутрішньому та міжнародному ринках.

ВИСНОВКИ ДО РОЗДІЛУ

Стартап-проект спрямований на розробку інформаційної системи персоналізованих медичних рекомендацій на основі методів машинного навчання. Основна ідея проекту полягає у створенні зручного цифрового сервісу, який забезпечує користувачів персональними порадами щодо здоров'я. Система аналізує індивідуальні медичні показники користувача, історію захворювань та поточні симптоми. В основі роботи стартапу лежить застосування алгоритмів машинного навчання для виявлення ризиків та формування точних рекомендацій. Важливою перевагою є можливість інтеграції з пристроями моніторингу здоров'я, такими як фітнес-браслети або глюкометри. Розроблена система буде корисною не лише для звичайних користувачів, а й для лікарів, страхових компаній і медичних закладів. Бізнес-модель передбачає freemium-доступ, ліцензування, API для партнерів та грантове фінансування. Вагомою рисою проекту є адаптація під українське законодавство та локальні медичні протоколи. Цільова аудиторія стартапу охоплює людей із хронічними захворюваннями, активних користувачів, медиків та родичів пацієнтів. Очікується, що система дозволить зменшити навантаження на лікарів та підвищить якість профілактики і лікування. Стратегія розвитку стартапу передбачає поступове розширення функціоналу та вихід на міжнародний ринок. Таким чином, стартап має високий потенціал для успішного впровадження в галузі цифрової медицини.

ВИСНОВКИ

Розроблена інформаційна система успішно реалізує повний цикл обробки медичних даних від введення симптомів до формування персоналізованих рекомендацій. Використання методів машинного навчання дозволяє підвищити точність діагностики та персоналізацію медичних рекомендацій на основі індивідуальних даних пацієнта. Для розв'язання задачі багатокласової класифікації захворювань було використано кілька алгоритмів, серед яких Random Forest та SVM показали високу ефективність. Попередня обробка даних, включаючи нормалізацію, кодування категоріальних змінних та видалення пропусків, є важливим етапом для забезпечення якості моделі. Система базується на структурованих медичних даних, що включають симптоми, анамнез, супутні захворювання та інші фактори здоров'я.

Використання бібліотек Scikit-Learn, Pandas, Streamlit дозволило розробити та протестувати систему з інтуїтивним графічним інтерфейсом. Розроблена система забезпечує не лише діагноз, а й комплексні рекомендації щодо лікування, дієти, фізичної активності та профілактики. Впровадження персоналізованих рекомендацій сприяє зменшенню кількості діагностичних помилок та покращує результати лікування. Інтеграція медичних знань з моделями машинного навчання забезпечує більш безпечні та клінічно обґрунтовані рекомендації.

Графічний інтерфейс, реалізований за допомогою Streamlit, є зручним для користувачів і дозволяє легко вводити симптоми та отримувати результати. Система може бути інтегрована в телемедичні платформи та мобільні додатки для широкого застосування серед населення. Використання метрик якості, таких як accuracy, AUC та confusion matrix дозволило об'єктивно оцінити продуктивність моделі.

Розроблена система має потенціал для підтримки лікарів у прийнятті клінічних рішень, особливо в умовах великого навантаження. Особливістю системи є врахування множинних захворювань одночасно, що важливо для пацієнтів з хронічними станами. Система може бути корисною не лише для пацієнтів, а й для лікарів, дієтологів, страхових компаній та інших учасників медичної сфери.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tawadrous D., Shariff S., Iansavichus A., Jain A. Use of clinical decision support systems for kidney-related drug prescribing: A systematic review. *American journal of kidney diseases*, vol. 58, №6, 2011. pp 903-914.
2. Palleria C., Paolo A., Giofre C., Caglioti C. Pharmacokinetic drug-drug interaction and their implication in clinical management. *Journal of research in medical sciences*, vol. 18, №7, 2013. pp. 601-608.
3. Chen R., Chiu J., Bati C. The recommendation of medicines based on multiple criteria decision making and domain ontology - an example of anti-diabetic medicines. *International conference on machine learning and cybernetics*. Guilin, China, 2011, pp. 27-32.
4. Chen S., Huang Y., Chen R. A recommendation system for anti-diabetic drugs selection based on fuzzy reasoning and ontology techniques. *International journal of pattern recognition and artificial intelligence*, vol. 27, №4, 2013. p.1359001-1359008.
5. Mahmoud N., Elbeh H. IRS-T2D: Individualize recommendation system for type diabetes medication based on ontology and SWRL. *International conference on informatics and systems*, Giza, Egypt, 2016, pp. 203-209.
6. Liu H., Xie G., Mei J. An efficacy driven approach for medication recommendation in type diabetes treatment using data mining techniques. *Studies in health technologies and informatics*, vol. 192, 2013. p.1071-1075.
7. Wedagu M., Chen D., Hussain M. Medicine recommendation system for diabetes using prior medical knowledge. *International conference on vision, image and signal processing*, Bangkok, Thailand, 2020. pp. 1-5.
8. Su R., Huang Y., Zhang D., Xiao G. SRDFM: Siamese response deep factorization machine to improve anti-cancer drug recommendation. *Briefings in bioinformatics*, vol. 23, №2, 2022. p. 534-539.
9. Chen D., Jin D., Goh T., Li N. Context-awareness based personalized recommendation of anti-hypertension drugs. *Journal of medical systems*, vol. 40, №9, 2016. pp.1-10.

10. Liu Y., Logan B., Liu N., Xu Z. Deep reinforcement learning for dynamic treatment regimes on medical registry data. International conference on healthcare informatics (ICHI), Park City, UT, USA, 2017. pp.380-385.
11. Zhang Y., Chen R., Tang J., Stewart W. LEAP: Learning to prescribe effective and safe treatment combinations for multimorbidity. International conference on knowledge discovery and data mining, Halifax, Canada, 2017. pp.1315-1324.
12. Мороз В. М., Ковальчук Л. Я. Медична інформатика. Київ: Медицина, 2018. 320 с.
13. Додонов О. Г., Трофимчук О. М. Інформаційні технології в медицині. Київ: Наукова думка, 2016. 280 с.
14. Баранов А. В., Ткаченко В. В. Цифрова медицина: перспективи та виклики. Харків: ХНУРЕ, 2021. 210 с.
15. Котенко О.Є., Шафроненко А.Ю., Сенніков І.А., Гаврилов А. Ю. Огляд методів підтримки персоналізованих медичних даних. Збірник наукових праць конференції «Розвиток наукової думки постіндустріального суспільства: сучасний дискурс». Вінниця: ТОВ «УКРЛОГОС Груп», 2024, стор. 455-457.

ДОДАТКИ

ДОДАТОК А

1.ipynb

```
import pandas as pd
dataset = pd.read_csv('Training.csv')
dataset

vals = dataset.values.flatten()

dataset.shape

train test split

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

X = dataset.drop('prognosis', axis=1)
y = dataset['prognosis']

le = LabelEncoder()
le.fit(y)
Y = le.transform(y)

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3,
random_state=20)

from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, confusion_matrix
import numpy as np

models = {
    'SVC': SVC(kernel='linear'),
    'RandomForest': RandomForestClassifier(n_estimators=100, random_state=42),
    'GradientBoosting': GradientBoostingClassifier(n_estimators=100,
random_state=42),
```

```

    'KNeighbors': KNeighborsClassifier(n_neighbors=5),
    'MultinomialNB': MultinomialNB()
}

```

```

for model_name, model in models.items():
    model.fit(X_train, y_train)
    predictions = model.predict(X_test)
    accuracy = accuracy_score(y_test, predictions)
    print(f"{model_name} Accuracy: {accuracy}")
    cm = confusion_matrix(y_test, predictions)
    print(f"{model_name} Confusion Matrix:")
    print(np.array2string(cm, separator=', '))
    print("\n" + "="*40 + "\n")

```

SVC Accuracy: 1.0

SVC Confusion Matrix:

```

[[40,  0,  0, ...,  0,  0,  0],
 [ 0, 43,  0, ...,  0,  0,  0],
 [ 0,  0, 28, ...,  0,  0,  0],
 ...,
 [ 0,  0,  0, ..., 34,  0,  0],
 [ 0,  0,  0, ...,  0, 41,  0],
 [ 0,  0,  0, ...,  0,  0, 31]]

```

RandomForest Accuracy: 1.0

RandomForest Confusion Matrix:

```

[[40,  0,  0, ...,  0,  0,  0],
 [ 0, 43,  0, ...,  0,  0,  0],
 [ 0,  0, 28, ...,  0,  0,  0],
 ...,
 [ 0,  0,  0, ..., 34,  0,  0],
 [ 0,  0,  0, ...,  0, 41,  0],
 [ 0,  0,  0, ...,  0,  0, 31]]

```

GradientBoosting Accuracy: 1.0

GradientBoosting Confusion Matrix:

```

[[40,  0,  0, ...,  0,  0,  0],
 [ 0, 43,  0, ...,  0,  0,  0],
 [ 0,  0, 28, ...,  0,  0,  0],
 ...,
 [ 0,  0,  0, ..., 34,  0,  0],
 [ 0,  0,  0, ...,  0, 41,  0],
 [ 0,  0,  0, ...,  0,  0, 31]]

```

```

KNeighbors Accuracy: 1.0
KNeighbors Confusion Matrix:
[[40,  0,  0, ...,  0,  0,  0],
 [ 0, 43,  0, ...,  0,  0,  0],
 [ 0,  0, 28, ...,  0,  0,  0],
 ...,
 [ 0,  0,  0, ..., 34,  0,  0],
 [ 0,  0,  0, ...,  0, 41,  0],
 [ 0,  0,  0, ...,  0,  0, 31]]

```

```

MultinomialNB Accuracy: 1.0
MultinomialNB Confusion Matrix:
[[40,  0,  0, ...,  0,  0,  0],
 [ 0, 43,  0, ...,  0,  0,  0],
 [ 0,  0, 28, ...,  0,  0,  0],
 ...,
 [ 0,  0,  0, ..., 34,  0,  0],
 [ 0,  0,  0, ...,  0, 41,  0],
 [ 0,  0,  0, ...,  0,  0, 31]]

```

```

svc = SVC(kernel='linear')
svc.fit(X_train,y_train)
ypred = svc.predict(X_test)
accuracy_score(y_test,ypred)
1.0

```

```

import pickle
pickle.dump(svc,open('svc.pkl','wb'))

```

```

svc = pickle.load(open('svc.pkl','rb'))

```

```

print("predicted disease :",svc.predict(X_test.iloc[0].values.reshape(1,-1)))
print("Actual Disease :", y_test[0])

```

```

print("predicted disease :",svc.predict(X_test.iloc[100].values.reshape(1,-1)))
print("Actual Disease :", y_test[100])

```

```

predicted disease : [39]
Actual Disease : 39

```

```

sym_des = pd.read_csv("syptoms_df.csv")
precautions = pd.read_csv("precautions_df.csv")
workout = pd.read_csv("workout_df.csv")

```

```

description = pd.read_csv("description.csv")
medications = pd.read_csv('medications.csv')
diets = pd.read_csv("diets.csv")

def helper(dis):
    desc = description[description['Disease'] == predicted_disease]['Description']
    desc = " ".join([w for w in desc])
    pre = precautions[precautions['Disease'] == dis][['Precaution_1', 'Precaution_2',
    'Precaution_3', 'Precaution_4']]
    pre = [col for col in pre.values]
    med = medications[medications['Disease'] == dis]['Medication']
    med = [med for med in med.values]

    die = diets[diets['Disease'] == dis]['Diet']
    die = [die for die in die.values]

    wrkout = workout[workout['disease'] == dis] ['workout']
    return desc,pre,med,die,wrkout

symptoms_dict = {'itching': 0, 'skin_rash': 1, 'nodal_skin_eruptions': 2,
'continuous_sneezing': 3, 'shivering': 4, 'chills': 5, 'joint_pain': 6,
'stomach_pain': 7, 'acidity': 8, 'ulcers_on_tongue': 9, 'muscle_wasting': 10,
'vomiting': 11, 'burning_micturition': 12, 'spotting_urination': 13, 'fatigue': 14,
'weight_gain': 15, 'anxiety': 16, 'cold_hands_and_feets': 17, 'mood_swings': 18,
'weight_loss': 19, 'restlessness': 20, 'lethargy': 21, 'patches_in_throat': 22,
'irregular_sugar_level': 23, 'cough': 24, 'high_fever': 25, 'sunken_eyes': 26,
'breathlessness': 27, 'sweating': 28, 'dehydration': 29, 'indigestion': 30,
'headache': 31, 'yellowish_skin': 32, 'dark_urine': 33, 'nausea': 34,
'loss_of_appetite': 35, 'pain_behind_the_eyes': 36, 'back_pain': 37, 'constipation':
38, 'abdominal_pain': 39, 'diarrhoea': 40, 'mild_fever': 41, 'yellow_urine': 42,
'yellowing_of_eyes': 43, 'acute_liver_failure': 44, 'fluid_overload': 45,
'swelling_of_stomach': 46, 'swelled_lymph_nodes': 47, 'malaise': 48,
'blurred_and_distorted_vision': 49, 'phlegm': 50, 'throat_irritation': 51,
'redness_of_eyes': 52, 'sinus_pressure': 53, 'runny_nose': 54, 'congestion': 55,
'chest_pain': 56, 'weakness_in_limbs': 57, 'fast_heart_rate': 58,
'pain_during_bowel_movements': 59, 'pain_in_anal_region': 60, 'bloody_stool': 61,
'irritation_in_anus': 62, 'neck_pain': 63, 'dizziness': 64, 'cramps': 65, 'bruising':
66, 'obesity': 67, 'swollen_legs': 68, 'swollen_blood_vessels': 69,
'puffy_face_and_eyes': 70, 'enlarged_thyroid': 71, 'brittle_nails': 72,
'swollen_extremities': 73, 'excessive_hunger': 74, 'extra_marital_contacts': 75,
'drying_and_tingling_lips': 76, 'slurred_speech': 77, 'knee_pain': 78,
'hip_joint_pain': 79, 'muscle_weakness': 80, 'stiff_neck': 81, 'swelling_joints': 82,
'movement_stiffness': 83, 'spinning_movements': 84, 'loss_of_balance': 85,

```

```
'unsteadiness': 86, 'weakness_of_one_body_side': 87, 'loss_of_smell': 88,
'bladder_discomfort': 89, 'foul_smell_of_urine': 90, 'continuous_feel_of_urine': 91,
'passage_of_gases': 92, 'internal_itching': 93, 'toxic_look_(typhos)': 94,
'depression': 95, 'irritability': 96, 'muscle_pain': 97, 'altered_sensorium': 98,
'red_spots_over_body': 99, 'belly_pain': 100, 'abnormal_menstruation': 101,
'dischromic_patches': 102, 'watering_from_eyes': 103, 'increased_appetite': 104,
'polyuria': 105, 'family_history': 106, 'mucoid_sputum': 107, 'rusty_sputum': 108,
'lack_of_concentration': 109, 'visual_disturbances': 110,
'receiving_blood_transfusion': 111, 'receiving_unsterile_injections': 112, 'coma':
113, 'stomach_bleeding': 114, 'distention_of_abdomen': 115,
'history_of_alcohol_consumption': 116, 'fluid_overload.1': 117, 'blood_in_sputum':
118, 'prominent_veins_on_calf': 119, 'palpitations': 120, 'painful_walking': 121,
'pus_filled_pimples': 122, 'blackheads': 123, 'scurring': 124, 'skin_peeling': 125,
'silver_like_dusting': 126, 'small_dents_in_nails': 127, 'inflammatory_nails': 128,
'blister': 129, 'red_sore_around_nose': 130, 'yellow_crust_ooze': 131}
```

```
diseases_list = {15: 'Fungal infection', 4: 'Allergy', 16: 'GERD', 9: 'Chronic
cholestasis', 14: 'Drug Reaction', 33: 'Peptic ulcer disease', 1: 'AIDS', 12:
'Diabetes ', 17: 'Gastroenteritis', 6: 'Bronchial Asthma', 23: 'Hypertension ', 30:
'Migraine', 7: 'Cervical spondylosis', 32: 'Paralysis (brain hemorrhage)', 28:
'Jaundice', 29: 'Malaria', 8: 'Chicken pox', 11: 'Dengue', 37: 'Typhoid', 40:
'hepatitis A', 19: 'Hepatitis B', 20: 'Hepatitis C', 21: 'Hepatitis D', 22:
'Hepatitis E', 3: 'Alcoholic hepatitis', 36: 'Tuberculosis', 10: 'Common Cold', 34:
'Pneumonia', 13: 'Dimorphic hemorrhoids(piles)', 18: 'Heart attack', 39: 'Varicose
veins', 26: 'Hypothyroidism', 24: 'Hyperthyroidism', 25: 'Hypoglycemia', 31:
'Osteoarthritis', 5: 'Arthritis', 0: '(vertigo) Parosymal Positional Vertigo', 2:
'Acne', 38: 'Urinary tract infection', 35: 'Psoriasis', 27: 'Impetigo'}
```

```
def get_predicted_value(patient_symptoms):
    input_vector = np.zeros(len(symptoms_dict))
    for item in patient_symptoms:
        input_vector[symptoms_dict[item]] = 1
    return diseases_list[svc.predict([input_vector])[0]]
```

```
symptoms = input("Enter your symptoms.....")
user_symptoms = [s.strip() for s in symptoms.split(',')]
user_symptoms = [symptom.strip("[]' ") for symptom in user_symptoms]
predicted_disease = get_predicted_value(user_symptoms)
```

```
desc, pre, med, die, wrkout = helper(predicted_disease)
```

```
print("=====predicted disease=====")
print(predicted_disease)
```

```

print("====description====")
print(desc)
print("====precautions====")
i = 1
for p_i in pre[0]:
    print(i, ": ", p_i)
    i += 1

print("====medications====")
for m_i in med:
    print(i, ": ", m_i)
    i += 1

print("====workout====")
for w_i in wrkout:
    print(i, ": ", w_i)
    i += 1

print("====diets====")
for d_i in die:
    print(i, ": ", d_i)
    i += 1

symptoms = input("Enter your symptoms.....")
user_symptoms = [s.strip() for s in symptoms.split(',')]
user_symptoms = [symptom.strip("[]' ") for symptom in user_symptoms]
predicted_disease = get_predicted_value(user_symptoms)
desc, pre, med, die, wrkout = helper(predicted_disease)

yellow_crust_ooze,red_sore_around_nose,small_dents_in_nails,inflammatory_nails,
blister
symptoms = input("Enter your symptoms.....")
user_symptoms = [s.strip() for s in symptoms.split(',')]
user_symptoms = [symptom.strip("[]' ") for symptom in user_symptoms]
predicted_disease = get_predicted_value(user_symptoms)

Enter your
symptoms.....yellow_crust_ooze,red_sore_around_nose,small_dents_in_nails,inflammat
ory_nails,blister
====predicted disease====
Impetigo
====description====
Impetigo is a highly contagious skin infection causing red sores that can break open.

```

=====precautions=====

- 1 : soak affected area in warm water
- 2 : use antibiotics
- 3 : remove scabs with wet compressed cloth
- 4 : consult doctor

=====medications=====

- 5 : ['Topical antibiotics', 'Oral antibiotics', 'Antiseptics', 'Ointments', 'Warm compresses']

=====workout=====

- 6 : Maintain good hygiene
- 7 : Stay hydrated
- 8 : Consume nutrient-rich foods
- 9 : Limit sugary foods and beverages
- 10 : Include foods rich in vitamin C
- 11 : Consult a healthcare professional
- 12 : Follow medical recommendations
- 13 : Avoid scratching
- 14 : Take prescribed antibiotics
- 15 : Practice wound care

=====diets=====

- 16 : ['Impetigo Diet', 'Antibiotic treatment', 'Fruits and vegetables', 'Hydration', 'Protein-rich foods']

ДОДАТОК Б

2.py

```
import streamlit as st
import pandas as pd
import numpy as np
import pickle
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.svm import SVC
import os

dataset = pd.read_csv('Training.csv')
X = dataset.drop('prognosis', axis=1)
y = dataset['prognosis']
le = LabelEncoder()
le.fit(y)
Y = le.transform(y)
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3,
random_state=20)
except FileNotFoundError:
st.error("Файл 'Training.csv' не знайдено. Будь ласка, переконайтеся, що він знаходиться у тому ж каталозі, що й цей скрипт.")

model_filename = 'svc.pkl'
if os.path.exists(model_filename):
    try:
        svc = pickle.load(open(model_filename, 'rb'))
    except Exception as e:
        st.error(f"Помилка завантаження моделі '{model_filename}': {e}")
        st.stop()
else:
    st.warning(f"Файл моделі '{model_filename}' не знайдено. Модель буде навчено з нуля.")
    svc = SVC(kernel='linear')
    svc.fit(X_train, y_train)
    with open(model_filename, 'wb') as f:
        pickle.dump(svc, f)
    st.info("Модель SVC навчено та збережено як 'svc.pkl'.")

try:
```

```

precautions = pd.read_csv("precautions_df.csv")
description = pd.read_csv("description.csv")
medications = pd.read_csv('medications.csv')
diets = pd.read_csv("diets.csv")
workout = pd.read_csv("workout_df.csv")
except FileNotFoundError as e:
    st.error(f"Не знайдено один з файлів з даними: {e}. Будь ласка, переконайтеся, що
всі файли знаходяться у тому ж каталозі.")
    st.stop()

symptoms_dict = {'itching': 0, 'skin_rash': 1, 'nodal_skin_eruptions': 2,
'continuous_sneezing': 3, 'shivering': 4, 'chills': 5, 'joint_pain': 6,
'stomach_pain': 7, 'acidity': 8, 'ulcers_on_tongue': 9, 'muscle_wasting': 10,
'vomiting': 11, 'burning_micturition': 12, 'spotting_urination': 13, 'fatigue': 14,
'weight_gain': 15, 'anxiety': 16, 'cold_hands_and_feets': 17, 'mood_swings': 18,
'weight_loss': 19, 'restlessness': 20, 'lethargy': 21, 'patches_in_throat': 22,
'irregular_sugar_level': 23, 'cough': 24, 'high_fever': 25, 'sunken_eyes': 26,
'breathlessness': 27, 'sweating': 28, 'dehydration': 29, 'indigestion': 30,
'headache': 31, 'yellowish_skin': 32, 'dark_urine': 33, 'nausea': 34,
'loss_of_appetite': 35, 'pain_behind_the_eyes': 36, 'back_pain': 37, 'constipation':
38, 'abdominal_pain': 39, 'diarrhoea': 40, 'mild_fever': 41, 'yellow_urine': 42,
'yellowing_of_eyes': 43, 'acute_liver_failure': 44, 'fluid_overload': 45,
'swelling_of_stomach': 46, 'swelled_lymph_nodes': 47, 'malaise': 48,
'blurred_and_distorted_vision': 49, 'phlegm': 50, 'throat_irritation': 51,
'redness_of_eyes': 52, 'sinus_pressure': 53, 'runny_nose': 54, 'congestion': 55,
'chest_pain': 56, 'weakness_in_limbs': 57, 'fast_heart_rate': 58,
'pain_during_bowel_movements': 59, 'pain_in_anal_region': 60, 'bloody_stool': 61,
'irritation_in_anus': 62, 'neck_pain': 63, 'dizziness': 64, 'cramps': 65, 'bruising':
66, 'obesity': 67, 'swollen_legs': 68, 'swollen_blood_vessels': 69,
'puffy_face_and_eyes': 70, 'enlarged_thyroid': 71, 'brittle_nails': 72,
'swollen_extremities': 73, 'excessive_hunger': 74, 'extra_marital_contacts': 75,
'drying_and_tingling_lips': 76, 'slurred_speech': 77, 'knee_pain': 78,
'hip_joint_pain': 79, 'muscle_weakness': 80, 'stiff_neck': 81, 'swelling_joints': 82,
'movement_stiffness': 83, 'spinning_movements': 84, 'loss_of_balance': 85,
'unsteadiness': 86, 'weakness_of_one_body_side': 87, 'loss_of_smell': 88,
'bladder_discomfort': 89, 'foul_smell_of_urine': 90, 'continuous_feel_of_urine': 91,
'passage_of_gases': 92, 'internal_itching': 93, 'toxic_look_(typhos)': 94,
'depression': 95, 'irritability': 96, 'muscle_pain': 97, 'altered_sensorium': 98,
'red_spots_over_body': 99, 'belly_pain': 100, 'abnormal_menstruation': 101,
'dischromic_patches': 102, 'watering_from_eyes': 103, 'increased_appetite': 104,
'polyuria': 105, 'family_history': 106, 'mucoid_sputum': 107, 'rusty_sputum': 108,
'lack_of_concentration': 109, 'visual_disturbances': 110,
'receiving_blood_transfusion': 111, 'receiving_unsterile_injections': 112, 'coma':

```

113, 'stomach_bleeding': 114, 'distention_of_abdomen': 115,
'history_of_alcohol_consumption': 116, 'fluid_overload.1': 117, 'blood_in_sputum':
118, 'prominent_veins_on_calf': 119, 'palpitations': 120, 'painful_walking': 121,
'pus_filled_pimples': 122, 'blackheads': 123, 'scurring': 124, 'skin_peeling': 125,
'silver_like_dusting': 126, 'small_dents_in_nails': 127, 'inflammatory_nails': 128,
'blister': 129, 'red_sore_around_nose': 130, 'yellow_crust_ooze': 131}

uk_to_en_symptoms_map = {

'свербіж': 'itching', 'висип на шкірі': 'skin_rash', 'вузликові висипи на шкірі':
'nodal_skin_eruptions', 'постійне чхання': 'continuous_sneezing', 'тремтіння':
'shivering', 'озноб': 'chills', 'біль у суглобах': 'joint_pain', 'біль у шлунку':
'stomach_pain', 'печія': 'acidity', 'виразки на язичку': 'ulcers_on_tongue',
'виснаження м\`язів': 'muscle_wasting', 'блювота': 'vomiting', 'печіння при
сечовипусканні': 'burning_micturition', 'плямисті виділення при сечовипусканні':
'spotting_urination', 'втома': 'fatigue', 'збільшення ваги': 'weight_gain',
'тривога': 'anxiety', 'холодні руки і ноги': 'cold_hands_and_feets', 'перепади
настрою': 'mood_swings', 'втрата ваги': 'weight_loss', 'занепокоєння':
'restlessness', 'млявість': 'lethargy', 'плями в горлі': 'patches_in_throat',
'нерегулярний рівень цукру': 'irregular_sugar_level', 'кашель': 'cough', 'висока
температура': 'high_fever', 'запали очі': 'sunken_eyes', 'задишка': 'breathlessness',
'пітливість': 'sweating', 'зневоднення': 'dehydration', 'нетравлення': 'indigestion',
'головний біль': 'headache', 'жовтувата шкіра': 'yellowish_skin', 'темна сеча':
'dark_urine', 'нудота': 'nausea', 'втрата апетиту': 'loss_of_appetite', 'біль за
очима': 'pain_behind_the_eyes', 'біль у спині': 'back_pain', 'запор': 'constipation',
'біль у животі': 'abdominal_pain', 'діарея': 'diarrhoea', 'помірне підвищення
температури': 'mild_fever', 'жовта сеча': 'yellow_urine', 'пожовтіння очей':
'yellowing_of_eyes', 'гостра печінкова недостатність': 'acute_liver_failure',
'перевантаження рідиною': 'fluid_overload', 'набряк живота': 'swelling_of_stomach',
'збільшення лімфовузлів': 'swelled_lymph_nodes', 'нездужання': 'malaise',
'затуманений та спотворений зір': 'blurred_and_distorted_vision', 'мокрота':
'phlegm', 'першіння в горлі': 'throat_irritation', 'почервоніння очей':
'redness_of_eyes', 'тиск у пазухах': 'sinus_pressure', 'нежить': 'runny_nose',
'закладеність': 'congestion', 'біль у грудях': 'chest_pain', 'слабкість в кінцівках':
'weakness_in_limbs', 'прискорене серцебиття': 'fast_heart_rate', 'біль під час
дефекації': 'pain_during_bowel_movements', 'біль в анальній області':
'pain_in_anal_region', 'кров у стільці': 'bloody_stool', , 'біль у шиї': 'neck_pain',
'запаморочення': 'dizziness', 'судоми': 'cramps', 'синці': 'bruising', 'ожиріння':
'obesity', 'набряклі ноги': 'swollen_legs', 'набряклі кровоносні судини':
'swollen_blood_vessels', 'одутлість обличчя та очей': 'puffy_face_and_eyes',
'збільшена щитовидка': 'enlarged_thyroid', 'ламкі нігті': 'brittle_nails', 'набряклі
кінцівки': 'swollen_extremities', 'надмірний голод': 'excessive_hunger', 'сухість і
поколювання губ': 'drying_and_tingling_lips', 'невиразна мова': 'slurred_speech',
'біль у коліні': 'knee_pain', 'біль у тазостегновому суглобі': 'hip_joint_pain',

```

'м\язова слабкість': 'muscle_weakness', 'скутість шиї': 'stiff_neck', 'набряк
суглобів': 'swelling_joints', 'скутість рухів': 'movement_stiffness', 'обертальні
рухи': 'spinning_movements', 'втрата рівноваги': 'loss_of_balance', 'нестійкість':
'unsteadiness', 'слабкість однієї сторони тіла': 'weakness_of_one_body_side', 'втрата
нюху': 'loss_of_smell', 'дискомфорт у сечовому міхурі': 'bladder_discomfort',
'неприємний запах сечі': 'foul_smell_of_urine', 'постійне відчуття сечі':
'continuous_feel_of_urine', 'виділення газів': 'passage_of_gases', 'внутрішній
свербіж': 'internal_itching', 'токсичний вигляд (тифос)': 'toxic_look_(typhos)',
'депресія': 'depression', 'дратівливість': 'irritability', 'м\язовий біль':
'muscle_pain', 'змінений стан свідомості': 'altered_sensorium', 'червоні плями по
тілу': 'red_spots_over_body', 'біль у животі (живіт)': 'belly_pain', дисхромічні
плями': 'dischromic_patches', 'сльозотеча': 'watering_from_eyes', 'підвищений
апетит': 'increased_appetite', 'поліурія': 'polyuria', 'сімейна історія':
'family_history', 'слизова мокрота': 'mucoid_sputum', 'іржава мокрота':
'rusty_sputum', 'неуважність': 'lack_of_concentration', 'порушення зору':
'visual_disturbances', 'отримання переливання крові': 'receiving_blood_transfusion',
'отримання нестерильних ін\єкцій': 'receiving_unsterile_injections', 'кома': 'coma',
'шлункова кровотеча': 'stomach_bleeding', 'здуття живота': 'distention_of_abdomen',
'історія вживання алкоголю': 'history_of_alcohol_consumption', 'перевантаження
рідиною.1': 'fluid_overload.1', 'кров у мокроті': 'blood_in_sputum', 'видатні вени на
ікрі': 'prominent_veins_on_calf', 'прискорене серцебиття': 'palpitations', 'болюча
ходьба': 'painful_walking', 'прищі з гноєм': 'pus_filled_pimples', 'чорні точки':
'blackheads', 'рубцювання': 'scarring', 'лущення шкіри': 'skin_peeling', 'сріблястий
наліт': 'silver_like_dusting', 'маленькі вм\ятини на нігтях':
'small_dents_in_nails', 'запалені нігті': 'inflammatory_nails', 'пухир': 'blister',
'червона ранка навколо носа': 'red_sore_around_nose', 'жовті кірки':
'yellow_crust_ooze'
}

```

```

translated_diseases_list = {
    15: 'Грибкова інфекція', 4: 'Алергія', 16: 'ГЕРХ', 9: 'Хронічний холестаз',
    14: 'Медикаментозна реакція', 33: 'Виразкова хвороба', 1: 'СНІД',
    12: 'Цукровий діабет', 17: 'Гастроентерит', 6: 'Бронхіальна астма',
    23: 'Гіпертонія', 30: 'Мігрень', 7: 'Шийний спондиліоз',
    32: 'Параліч (крововилив у мозок)', 28: 'Жовтяниця', 29: 'Малярія',
    8: 'Вітряна віспа', 11: 'Лихоманка денге', 37: 'Черевний тиф',
    40: 'Гепатит А', 19: 'Гепатит В', 20: 'Гепатит С', 21: 'Гепатит D',
    22: 'Гепатит Е', 3: 'Алкогольний гепатит', 36: 'Туберкульоз',
    10: 'Звичайна застуда', 34: 'Пневмонія', 13: 'Диморфологічний геморой',
    18: 'Інфаркт міокарда', 39: 'Варикозне розширення вен', 26: 'Гіпотиреоз',
    24: 'Гіпертиреоз', 25: 'Гіпоглікемія', 31: 'Остеоартрит', 5: 'Артрит',
    0: '(вертиго) Пароксизмальне позиційне запаморочення', 2: 'Акне',
    38: 'Інфекція сечовивідних шляхів', 35: 'Псоріаз', 27: 'Імпетиго'
}

```

```
}
```

```
original_diseases_list = {  
    15: 'Fungal infection', 4: 'Allergy', 16: 'GERD', 9: 'Chronic cholestasis',  
    14: 'Drug Reaction', 33: 'Peptic ulcer disease', 1: 'AIDS',  
    12: 'Diabetes ', 17: 'Gastroenteritis', 6: 'Bronchial Asthma',  
    23: 'Hypertension ', 30: 'Migraine', 7: 'Cervical spondylosis',  
    32: 'Paralysis (brain hemorrhage)', 28: 'Jaundice', 29: 'Malaria',  
    8: 'Chicken pox', 11: 'Dengue', 37: 'Typhoid',  
    40: 'hepatitis A', 19: 'Hepatitis B', 20: 'Hepatitis C', 21: 'Hepatitis D',  
    22: 'Hepatitis E', 3: 'Alcoholic hepatitis', 36: 'Tuberculosis',  
    10: 'Common Cold', 34: 'Pneumonia', 13: 'Dimorphic hemmorhoids(piles)',  
    18: 'Heart attack', 39: 'Varicose veins', 26: 'Hypothyroidism',  
    24: 'Hyperthyroidism', 25: 'Hypoglycemia', 31: 'Osteoarthritis',  
    5: 'Arthritis', 0: '(vertigo) Paroymsal Positional Vertigo', 2: 'Acne',  
    38: 'Urinary tract infection', 35: 'Psoriasis', 27: 'Impetigo'  
}
```

```
def get_predicted_value(patient_symptoms):  
    input_vector = np.zeros(len(symptoms_dict))  
    for item in patient_symptoms:  
        if item in symptoms_dict:  
            input_vector[symptoms_dict[item]] = 1  
    prediction_index = svc.predict([input_vector])[0]  
    translated_name = translated_diseases_list.get(prediction_index, "Невідома  
хвороба")  
    original_name = original_diseases_list.get(prediction_index)  
    return translated_name, original_name
```

```
translation_maps = {  
    'Fungal infection': {  
        "description": "Грибкова інфекція - це шкірне захворювання, що спричиняється  
грибками, і може проявлятися висипом, свербіжем, лущенням.",  
        "precautions": ["Підтримуйте гігієну", "Уникайте спільного використання  
рушників", "Носіть вільний одяг", "Тримайте уражену ділянку сухою"],  
        "medications": ["Протигрибкові креми", "Оральні протигрибкові препарати",  
"Антигістамінні препарати"],  
        "workout": ["Зберігайте сухість", "Дотримуйтеся особистої гігієни"],  
        "diets": ["Зменшити цукор", "Вживайте пробіотики", "Включити часник", "Додати  
вітамін С"]  
    },  
    'Allergy': {
```

```

    "description": "Алергія - це імунна реакція організму на сторонні речовини, яка може викликати свербіж, висип, чхання.",
    "precautions": ["Уникайте алергенів", "Приймайте антигістамінні препарати", "Носіть маску в запилених місцях", "Підтримуйте чистоту вдома"],
    "medications": ["Антигістамінні", "Стероїди", "Крем з кортизоном"],
    "workout": ["Виконувати дихальні вправи", "Прогулянки на свіжому повітрі"],
    "diets": ["Уникайте оброблених продуктів", "Протизапальні продукти", "Додати фрукти та овочі", "Пити багато води"]
  },
  'GERD': {
    "description": "ГЕРХ (гастроєзофагеальна рефлюксна хвороба) - це стан, коли шлунковий сік потрапляє назад в стравохід, викликаючи печію.",
    "precautions": ["Уникайте гострої їжі", "Не лягайте після їжі", "Підійміть узголів'я ліжка", "Зменште вживання алкоголю"],
    "medications": ["Антациди", "Інгібітори протонної помпи", "Блокатори H2-рецепторів"],
    "workout": ["Легкі прогулянки", "Не напружуйте м'язи живота"],
    "diets": ["Зменшити жирну їжу", "Їжте маленькими порціями", "Додати імбир", "Уникайте цитрусових"]
  },
  'Chronic cholestasis': {
    "description": "Хронічний холестаза - це стан, при якому порушується відтік жовчі, що може призвести до жовтяниці та свербіжів.",
    "precautions": ["Звернутися до гастроентеролога", "Уникати алкоголю", "Стежити за дієтою", "Приймати призначені ліки"],
    "medications": ["Урсодезоксихолева кислота", "Холестирамін"],
    "workout": ["Регулярні помірні вправи"],
    "diets": ["Низькожирова дієта", "Додати овочі та фрукти", "Пити багато води"]
  },
  'Drug Reaction': {
    "description": "Реакція на ліки - це небажана реакція організму на медикамент, яка може проявлятися висипом, свербіжем, анафілаксією.",
    "precautions": ["Негайно припинити прийом ліків", "Звернутися до лікаря", "Повідомити про алергію", "Уникати аналогічних препаратів"],
    "medications": ["Антигістамінні", "Стероїди", "Адреналін"],
    "workout": ["Відпочинок", "Відновлення"],
    "diets": ["Легка їжа", "Пити багато рідини"]
  },
  'Peptic ulcer disease': {
    "description": "Виразкова хвороба - це утворення виразок на слизовій оболонці шлунка або дванадцятипалої кишки.",
    "precautions": ["Уникати гострої їжі", "Уникати стресу", "Не палити", "Обмежити вживання алкоголю"],

```

```

    "medications": ["Антациди", "Інгібітори протонної помпи", "Антибіотики"],
    "workout": ["Йога", "Медитація", "Легкі прогулянки"],
    "diets": ["Зменшити кислотність", "Дрібне харчування", "Відварні страви",
"Каші"]
  },
  'AIDS': {
    "description": "СНІД (синдром набутого імунного дефіциту) - це хронічне захворювання, викликане ВІЛ, яке руйнує імунну систему.",
    "precautions": ["Дотримуватися антиретровірусної терапії", "Проходити регулярні обстеження", "Уникати інфекцій", "Дотримуватися здорового способу життя"],
    "medications": ["АРВ-препарати"],
    "workout": ["Регулярні фізичні вправи", "Дотримуватися розпорядку дня"],
    "diets": ["Багатий білком", "Фрукти та овочі", "Пити багато води"]
  },
  'Diabetes ': {
    "description": "Цукровий діабет - це хронічне захворювання, при якому організм не може ефективно використовувати цукор (глюкозу) для отримання енергії.",
    "precautions": ["Контролювати рівень цукру в крові", "Дотримуватися дієти", "Регулярні фізичні навантаження", "Приймати ліки за призначенням"],
    "medications": ["Метформін", "Інсулін", "Глібенкламід"],
    "workout": ["Ходьба", "Плавання", "Їзда на велосипеді"],
    "diets": ["Збалансоване харчування", "Обмежити вуглеводи", "Більше клітковини"]
  },
  'Gastroenteritis': {
    "description": "Гастроентерит - це запалення шлунка та кишечника, яке часто викликається вірусом і проявляється блювотою та діареєю.",
    "precautions": ["Пити багато рідини", "Їсти легку їжу", "Відпочивати", "Дотримуватися гігієни рук"],
    "medications": ["Регідратаційні розчини", "Протидіарейні препарати"],
    "workout": ["Відпочинок", "Уникати навантажень"],
    "diets": ["BRAT-дієта (банани, рис, яблучне пюре, тости)"]
  },
  'Bronchial Asthma': {
    "description": "Бронхіальна астма - це хронічне запальне захворювання дихальних шляхів, що викликає напади задишки та кашель.",
    "precautions": ["Уникати алергенів", "Не палити", "Регулярно приймати інгалятор", "Стежити за вологістю повітря"],
    "medications": ["Інгалятори", "Бронходилататори", "Стероїди"],
    "workout": ["Дихальні вправи", "Плавання", "Ходьба"],
    "diets": ["Протизапальні продукти", "Вітамін D", "Уникати сульфатів"]
  },
  'Hypertension ': {

```

```

    "description": "Гіпертонія (високий кров'яний тиск) - це хронічне підвищення артеріального тиску, що збільшує ризик інфаркту та інсульту.",
    "precautions": ["Контролювати тиск", "Зменшити сіль у раціоні", "Регулярні фізичні навантаження", "Уникати стресу"],
    "medications": ["Діуретики", "Інгібітори АПФ", "Бета-блокатори"],
    "workout": ["Кардіо", "Швидка ходьба", "Плавання"],
    "diets": ["DASH-дієта", "Обмежити сіль", "Калій"]
  },
  'Migraine': {
    "description": "Мігрень - це тип головного болю, який часто супроводжується нудотою, блювотою та чутливістю до світла й звуку.",
    "precautions": ["Уникати тригерів", "Відпочивати в темній кімнаті", "Звернутися до невролога", "Регулярний сон"],
    "medications": ["Ібупрофен", "Триптани", "Бета-блокатори"],
    "workout": ["М'яка йога", "Ходьба", "Релаксація"],
    "diets": ["Уникати кофеїну", "Не пропускати прийоми їжі", "Пити багато води", "Зменшити нітрати"]
  },
  'Cervical spondylosis': {
    "description": "Шийний спондильоз - це вікове зношення шийних хребців, що може спричинити біль у шиї та оніміння.",
    "precautions": ["Регулярні вправи для шиї", "Правильна постава", "Використовувати ортопедичну подушку", "Фізіотерапія"],
    "medications": ["Знеболювальні", "М'язові релаксанти"],
    "workout": ["Вправи для шиї", "Йога", "Плавання"],
    "diets": ["Додати кальцій", "Вітамін D", "Протизапальні продукти"]
  },
  'Paralysis (brain hemorrhage)': {
    "description": "Параліч (крововилив у мозок) - це втрата м'язової функції, спричинена пошкодженням мозку внаслідок крововиливу.",
    "precautions": ["Реабілітація", "Фізіотерапія", "Контроль тиску", "Дотримуватися всіх медичних рекомендацій"],
    "medications": ["Ліки для контролю тиску", "Антикоагулянти"],
    "workout": ["Фізична терапія", "Масаж"],
    "diets": ["Здорова, збалансована", "Зменшити сіль та жири"]
  },
  'Jaundice': {
    "description": "Жовтяниця - це пожовтіння шкіри та очей, спричинене надлишком білірубіну в крові, що вказує на проблеми з печінкою.",
    "precautions": ["Відпочинок", "Уникати алкоголю", "Дотримуватися дієти", "Звернутися до лікаря"],
    "medications": ["Гепатопротектори", "Вітаміни"],
    "workout": ["Легкі прогулянки"],

```

```

    "diets": ["Легка дієта", "Багато рідини", "Уникати жирної їжі"]
  },
  'Malaria': {
    "description": "Малярія - це інфекційне захворювання, що передається через укуси малярійного комара і супроводжується лихоманкою, ознобом.",
    "precautions": ["Приймати антималярійні препарати", "Використовувати репеленти", "Спати під сіткою", "Звернутися до лікаря"],
    "medications": ["Хлорохін", "Артемізинін"],
    "workout": ["Відпочинок"],
    "diets": ["Висококалорійна", "Багато рідини", "Фрукти та овочі"]
  },
  'Chicken pox': {
    "description": "Вітряна віспа - це дуже заразне вірусне захворювання, що викликає висип та свербіж по всьому тілу.",
    "precautions": ["Уникати контакту з іншими", "Не розчісувати висип", "Відпочинок", "Звернутися до лікаря"],
    "medications": ["Ацикловір", "Антигістамінні"],
    "workout": ["Відпочинок"],
    "diets": ["М'яка їжа", "Багато рідини", "Уникати гострого та кислого"]
  },
  'Dengue': {
    "description": "Лихоманка денге - це вірусна інфекція, що передається комарами, і може викликати сильну лихоманку та м'язовий біль.",
    "precautions": ["Пити багато рідини", "Відпочинок", "Уникати укусів комарів", "Звернутися до лікаря"],
    "medications": ["Парацетамол", "Не використовувати аспірин"],
    "workout": ["Відпочинок"],
    "diets": ["Багато рідини", "Легка їжа", "Фрукти"]
  },
  'Typhoid': {
    "description": "Черевний тиф - це бактеріальна інфекція, що передається через забруднену їжу та воду, і викликає високу температуру.",
    "precautions": ["Приймати антибіотики", "Пити кип'ячену воду", "Дотримуватися гігієни", "Відпочинок"],
    "medications": ["Антибіотики (напр., ципрофлоксацин)"],
    "workout": ["Відпочинок"],
    "diets": ["Легка, рідка їжа", "Протерта їжа"]
  },
  'hepatitis A': {
    "description": "Гепатит А - це вірусне захворювання печінки, яке передається через забруднену їжу або воду.",
    "precautions": ["Відпочинок", "Уникати алкоголю", "Дотримуватися гігієни рук", "Дотримуватися дієти"],

```

```

    "medications": ["Підтримуюча терапія"],
    "workout": ["Відпочинок"],
    "diets": ["Легка дієта", "Нежирні продукти"]
  },
  'Hepatitis B': {
    "description": "Гепатит В - це серйозна вірусна інфекція печінки, яка може стати хронічною і викликати цироз або рак печінки.",
    "precautions": ["Регулярні обстеження", "Вакцинація", "Уникати алкоголю", "Дотримуватися рекомендацій лікаря"],
    "medications": ["Противірусні препарати"],
    "workout": ["Легкі фізичні вправи"],
    "diets": ["Здорова дієта", "Уникати жирного", "Обмежити цукор"]
  },
  'Hepatitis C': {
    "description": "Гепатит С - це вірусна інфекція, яка викликає запалення печінки і може призвести до серйозних ускладнень.",
    "precautions": ["Уникати алкоголю", "Дотримуватися рекомендацій лікаря", "Регулярні обстеження", "Противірусна терапія"],
    "medications": ["Противірусні препарати"],
    "workout": ["Помірні фізичні навантаження"],
    "diets": ["Збалансоване харчування", "Уникати жирних продуктів"]
  },
  'Hepatitis D': {
    "description": "Гепатит D - це вірусна інфекція, яка виникає тільки у людей з гепатитом В.",
    "precautions": ["Дотримуватися лікування гепатиту В", "Уникати алкоголю", "Регулярні обстеження"],
    "medications": ["Противірусні препарати"],
    "workout": ["Відпочинок"],
    "diets": ["Дієта для печінки"]
  },
  'Hepatitis E': {
    "description": "Гепатит Е - це вірусна інфекція печінки, що передається через забруднену воду.",
    "precautions": ["Дотримуватися гігієни", "Пити тільки чисту воду", "Відпочинок"],
    "medications": ["Підтримуюча терапія"],
    "workout": ["Відпочинок"],
    "diets": ["Легка дієта"]
  },
  'Alcoholic hepatitis': {
    "description": "Алкогольний гепатит - це запалення печінки, спричинене надмірним вживанням алкоголю.",

```

```

      "precautions": ["Припинити вживання алкоголю", "Звернутися до лікаря",
"Дотримуватися дієти", "Приймати призначені ліки"],
      "medications": ["Стероїди", "Вітаміни"],
      "workout": ["Відпочинок"],
      "diets": ["Без алкоголю", "Протеїнова дієта", "Багато рідини"]
    },
    'Tuberculosis': {
      "description": "Туберкульоз - це бактеріальна інфекція, яка зазвичай вражає
легені і передається повітряно-крапельним шляхом.",
      "precautions": ["Приймати антибіотики", "Дотримуватися гігієни",
"Ізолюватися", "Дотримуватися рекомендацій лікаря"],
      "medications": ["Ізоніазид", "Рифампіцин"],
      "workout": ["Помірні вправи", "Свіже повітря"],
      "diets": ["Здорова, збалансована", "Багато білка", "Вітаміни"]
    },
    'Common Cold': {
      "description": "Звичайна застуда - це вірусна інфекція верхніх дихальних
шляхів, яка викликає чхання, кашель, нежить.",
      "precautions": ["Відпочинок", "Пити багато рідини", "Полоскати горло",
"Приймати ліки від симптомів"],
      "medications": ["Парацетамол", "Ібупрофен", "Спреї для носа"],
      "workout": ["Легкі прогулянки"],
      "diets": ["Курячий бульйон", "Вітамін С", "Імбирний чай"]
    },
    'Pneumonia': {
      "description": "Пневмонія - це запалення легень, яке може бути спричинене
бактеріями, вірусами чи грибами.",
      "precautions": ["Приймати антибіотики", "Відпочинок", "Пити багато рідини",
"Дотримуватися рекомендацій лікаря"],
      "medications": ["Антибіотики", "Протикашльові препарати"],
      "workout": ["Відновлення", "Дихальні вправи"],
      "diets": ["Дієта з високим вмістом білка", "Вітаміни"]
    },
    'Dimorphic hemorrhoids (piles)': {
      "description": "Геморой - це запалення і розширення вен у прямій кишці та
анальному отворі.",
      "precautions": ["Збільшити клітковину в раціоні", "Пити багато води",
"Уникати напруження при дефекації", "Використовувати мазі"],
      "medications": ["Креми для геморою", "Знеболжувальні"],
      "workout": ["Ходьба", "Вправи Кегеля"],
      "diets": ["Багато клітковини", "Фрукти, овочі"]
    },
    'Heart attack': {

```

```

    "description": "Інфаркт міокарда - це пошкодження серцевого м'яза через блокування кровотоку до серця.",
    "precautions": ["Негайно викликати швидку допомогу", "Розжувати аспірин", "Відпочинок", "Дотримуватися медичних рекомендацій"],
    "medications": ["Аспірин", "Нітрогліцерин"],
    "workout": ["Кардіо вправи (за рекомендацією лікаря)"],
    "diets": ["Низькожирова", "Зменшити сіль", "Збільшити клітковину"]
  },
  'Varicose veins': {
    "description": "Варикозне розширення вен - це розширені, скручені вени, які можуть бути болючими, найчастіше на ногах.",
    "precautions": ["Носити компресійні панчохи", "Піднімати ноги", "Уникати довгого стояння", "Регулярні фізичні вправи"],
    "medications": ["Венотоніки"],
    "workout": ["Ходьба", "Плавання", "Їзда на велосипеді"],
    "diets": ["Зменшити сіль", "Багато клітковини"]
  },
  'Hypothyroidism': {
    "description": "Гіпотиреоз - це стан, при якому щитовидна залоза не виробляє достатньо гормонів.",
    "precautions": ["Приймати гормональну терапію", "Регулярні аналізи", "Збалансоване харчування", "Відпочинок"],
    "medications": ["Левотироксин"],
    "workout": ["Йога", "Прогулянки"],
    "diets": ["Зменшити оброблену їжу", "Багато йоду"]
  },
  'Hyperthyroidism': {
    "description": "Гіпертиреоз - це стан, коли щитовидна залоза виробляє занадто багато гормонів, що прискорює метаболізм.",
    "precautions": ["Приймати ліки", "Звернутися до ендокринолога", "Уникати стресу", "Регулярні обстеження"],
    "medications": ["Тиреостатики", "Бета-блокатори"],
    "workout": ["Легкі вправи"],
    "diets": ["Без йоду", "Протизапальні продукти"]
  },
  'Hypoglycemia': {
    "description": "Гіпоглікемія (низький рівень цукру в крові) - це небезпечний стан, який може спричинити слабкість, тремтіння, сплутаність свідомості.",
    "precautions": ["Завжди мати при собі цукор", "Регулярно харчуватися", "Приймати ліки за призначенням", "Контроль цукру"],
    "medications": ["Глюкоза"],
    "workout": ["Помірні вправи"],
    "diets": ["Регулярні прийоми їжі", "Складні вуглеводи"]
  }

```

```

},
'Osteoarthritis': {
    "description": "Остеоартрит - це дегенеративне захворювання суглобів, яке спричиняє біль та скутість.",
    "precautions": ["Помірні фізичні навантаження", "Контроль ваги", "Фізіотерапія", "Використання ортезів"],
    "medications": ["НПЗП", "Хондропротектори"],
    "workout": ["Плавання", "Ходьба", "Йога"],
    "diets": ["Протизапальна", "Зменшити цукор"]
},
'Arthritis': {
    "description": "Артрит - це запалення одного або декількох суглобів, що викликає біль та набряк.",
    "precautions": ["Фізіотерапія", "Вправи", "Тепло/холод", "Протизапальні ліки"],
    "medications": ["НПЗП", "Глюкокортикоїди"],
    "workout": ["Легкі вправи", "Плавання"],
    "diets": ["Протизапальна", "Омега-3", "Вітамін D"]
},
'(vertigo) Paroxysmal Positional Vertigo': {
    "description": "Пароксизмальне позиційне запаморочення - це відчуття обертання, що виникає при зміні положення голови.",
    "precautions": ["Виконувати вправи Еплі", "Уникати швидких рухів головою", "Звернутися до лікаря"],
    "medications": ["Протизапаморочні засоби"],
    "workout": ["Вестибулярна гімнастика"],
    "diets": ["Без кофеїну", "Зменшити сіль"]
},
'Acne': {
    "description": "Акне - це хронічне запальне захворювання шкіри, що проявляється висипами та прищами.",
    "precautions": ["Не чіпати обличчя руками", "Дотримуватися гігієни", "Збалансоване харчування", "Звернутися до дерматолога"],
    "medications": ["Креми з бензоїлпероксидом", "Саліцилова кислота"],
    "workout": ["Підтримуйте гігієну"],
    "diets": ["Зменшити цукор", "Безмолочна дієта"]
},
'Urinary tract infection': {
    "description": "Інфекція сечовивідних шляхів - це інфекція в будь-якій частині сечової системи.",
    "precautions": ["Пити багато рідини", "Дотримуватися гігієни", "Не затримувати сечовипускання", "Приймати антибіотики"],
    "medications": ["Антибіотики", "Знеболювальні"],

```

```

        "workout": ["Легкі вправи"],
        "diets": ["Журавлинний сік", "Багато води"]
    },
    'Psoriasis': {
        "description": "Псоріаз - це хронічне аутоімунне захворювання шкіри, що характеризується червоними, лускатими плямами.",
        "precautions": ["Зволожувати шкіру", "Фототерапія", "Зменшити стрес", "Дотримуватися дієти"],
        "medications": ["Місцеві мазі", "Імуносупресанти"],
        "workout": ["Плавання", "Йога"],
        "diets": ["Без глютену", "Протизапальна", "Омега-3"]
    },
    'Impetigo': {
        "description": "Імпетіго - це дуже заразне шкірне захворювання, що спричиняє появу червоних виразок, які можуть розкриватися.",
        "precautions": ["Змочити уражену ділянку теплою водою", "Використовувати антибіотики", "Зняти кірки вологою компресійною тканиною", "Звернутися до лікаря"],
        "medications": ["Місцеві антибіотики", "Оральні антибіотики", "Антисептики", "Мазі", "Теплі компреси"],
        "workout": ["Дотримуватися гігієни", "Пити достатньо води", "Вживати продукти, багаті на поживні речовини", "Обмежити вживання цукру та солодких напоїв", "Додати в раціон продукти, багаті на вітамін С", "Звернутися до лікаря", "Дотримуватися медичних рекомендацій", "Уникати розчісування", "Приймати призначені антибіотики", "Дотримуватися догляду за раню"],
        "diets": ["Дієта при імпетіго", "Лікування антибіотиками", "Фрукти та овочі", "Гідратація", "Продукти, багаті на білок"]
    }
}

```

```

def helper_with_translation(original_dis):
    if original_dis is None:
        return None, None, None, None, None
    if original_dis in translation_maps:
        translated_data = translation_maps[original_dis]
        return (
            translated_data.get("description", "Опис відсутній."),
            translated_data.get("precautions", []),
            translated_data.get("medications", []),
            translated_data.get("diets", []),
            translated_data.get("workout", [])
        )
    else:

```

```

        st.warning(f"Переклад для хвороби '{original_dis}' не знайдено. Відображено
оригінальні дані.")

        desc = description[description['Disease'] ==
original_dis]['Description'].to_list()
        desc = " ".join(desc) if desc else "Опис відсутній."
        pre = precautions[precautions['Disease'] == original_dis][['Precaution_1',
'Precaution_2', 'Precaution_3', 'Precaution_4']].values.tolist()
        pre = pre[0] if pre else []
        med = medications[medications['Disease'] ==
original_dis]['Medication'].to_list()
        med = med if med else []
        die = diets[diets['Disease'] == original_dis]['Diet'].to_list()
        die = die if die else []
        wrkout = workout[workout['disease'] == original_dis]['workout'].to_list()
        wrkout = wrkout if wrkout else []
        return desc, pre, med, die, wrkout

st.image('1.png')
st.title("Система персоналізованих медичних рекомендацій :hospital: :health_worker:")
st.markdown("На основі методів машинного навчання")

st.markdown("---")

st.subheader("Введіть ваші симптоми: ☐")
st.info("Введіть симптоми через кому. Наприклад: свербіж, висип на шкіри, вузликові
висипи на шкірі")
user_symptoms_input = st.text_input(
    "Симптоми",
    placeholder="свербіж, висип на шкіри, вузликові висипи на шкірі"
)
if user_symptoms_input:
    # Обробка введених симптомів
    user_symptoms_uk = [s.strip().lower() for s in user_symptoms_input.split(',')]
    # Перекладаємо українські симптоми на англійські за допомогою словника
    valid_symptoms_en = [uk_to_en_symptoms_map[s] for s in user_symptoms_uk if s in
uk_to_en_symptoms_map]
    invalid_symptoms_uk = [s for s in user_symptoms_uk if s not in
uk_to_en_symptoms_map]
    if valid_symptoms_en:
        translated_disease, original_disease = get_predicted_value(valid_symptoms_en)
        if original_disease:
            desc, pre, med, die, wrkout = helper_with_translation(original_disease)

```

```

st.markdown("---")
st.subheader("Результат прогнозування :mag_right:")
st.success(f"**Передбачена хвороба:** {translated_disease} ☐")
st.subheader("Опис :scroll:")
st.write(desc)
st.subheader("Запобіжні заходи :shield:")
if pre:
    for i, p_i in enumerate(pre, 1):
        st.write(f"{i}. {p_i}")
else:
    st.write("Інформація відсутня.")
st.subheader("Медикаменти :syringe:")
if med:
    for i, m_i in enumerate(med, 1):
        st.write(f"{i}. {m_i}")
else:
    st.write("Інформація відсутня.")

st.subheader("Вправи :running_man:")
if wrkout:
    for i, w_i in enumerate(wrkout, 1):
        st.write(f"{i}. {w_i}")
else:
    st.write("Інформація відсутня.")
st.subheader("Дієта ☐")
if die:
    for i, d_i in enumerate(die, 1):
        st.write(f"{i}. {d_i}")
else:
    st.write("Інформація відсутня.")
else:
    st.warning("Не вдалося передбачити хворобу на основі введених симптомів.
☹️")
else:
    st.error("Будь ласка, введіть дійсні симптоми, які є в системі. ☐")
if invalid_symptoms_uk:
    st.info(f"Наступні симптоми не розпізнано та було проігноровано: {'',
'.join(invalid_symptoms_uk)}")

```