МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Таврійський державний агротехнологічний університет
імені Дмитра Моторного
Факультет енергетики і комп'ютерних технологій

ЗАТВЕРДЖУЮ
Зав. каф. «Комп'ютерні науки»
доц._____ ___ Сергій ШАРОВ
«16»_червня__2025 р.

Пояснювальна записка

до кваліфікаційної роботи здобувача СВО Бакалавр
(ступінь вищої освіти)

на тему: «Інформаційна система прогнозування захворювань методами машинного навчання»

52/4КНД.9683878.000000ПЗ

Виконав: здобувач вищої освіти 4 курсу, групи 41КН
спеціальності 122 Комп'ютерні науки
за ОПП Комп'ютерні науки
(шифр і назва спеціальності та ОПП)

_____ Владислав КУЗНЄЦОВ
(підпис)

Керівник д.т.н., професор _____ Віра МАЛКІНА
(підпис)

Консультант доц. _____ Лариса БОЛТЯНСЬКА
(підпис)

Консультант доц. _____ Михайло ЗОРЯ
(підпис)

Нормоконтроль, ст.викл. _____ Ольга ЗІНОВ'ЄВА
(підпис)

Рецензент, доц._____ Юрій СІЦИЛІЦИН
(підпис)

Запоріжжя – 2025 рік

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Таврійський державний агротехнологічний університет імені Дмитра Моторного

Факультет: Енергетики і комп'ютерних технологій
Кафедра: Комп'ютерні науки
Ступінь вищої освіти: Магістр
Спеціальність: 122 Комп'ютерні науки
ОПП: Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри_КН
к.пед.н., доц. _____ Сергій ШАРОВ
«10»_ жовтня _2024_ року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

_ КУЗНЄЦОВ Владислав Вячеславович_____
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи__«Інформаційна система прогнозування захворювань методами машинного навчання»
керівник проєкту __ПІБ, к.пед.н., доцент_____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом університету від "30" вересня_2024_року №_452-С
2. Строк подання здобувачем вищої освіти роботи __14 червня 2025_____
3. Вихідні дані до роботи: дані обстеження об'єкту, статистичні дані, технічне завдання на дипломне проєктування, матеріали виробничих практик, нормативні документи, науково-технічна література, електронні ресурси та ін._____
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)____
___1. Опис предметної області та аналіз існуючих рішень._____
___2. Специфікація вимог до експертної системи._____
___3. Вибір та обґрунтування технологій розробки експертної системи.___
___4. Дослідна експлуатація експертної системи._____
___5. Тестування експертної системи._____
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників)
Діаграма використання, діаграми IDEF0, розкадровка інтерфейсу, структурна схема, візуальні компоненти інтерфейсу, алгоритм роботи, інтерфейс програми
Презентація – 11 слайдів

1. Титульний слайд
2. Предмет, об'єкт, мета
3. Завдання дослідження
4. Порівняльна характеристика аналогів програмного продукту
5. Діаграма варіантів використання

6. Інструментальні засоби
7. Інтерфейс і робота експертної системи
8. Інтерфейс і робота експертної системи
9. Тестування
10. Висновок
11. Дякую за увагу

6. Консультанти розділів кваліфікаційної роботи

Розділ/ Підрозділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4.3	Болтянська Л. О.	15.10.2025	15.06.2025
4.4	Зоря М. В.	15.10.2025	15.06.2025

7. Дата видачі завдання___10 жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз предметної області	03.10.2024	Виконано
2	Специфікація вимог до експертної системи	17.10.2024	Виконано
3	Проектування експертної системи	28.10.2024	Виконано
4	Обґрунтування інструментальних засобів	18.11.2024	Виконано
5	Розробка експертної системи	16.12.2024	Виконано
6	Тестування та налагодження експертної системи	11.04.2025	Виконано
7	Підпис керівником роботи	14.06.2025	Виконано
8	Підпис завідувачем кафедри	16.06.2025	Виконано

Здобувач вищої освіти _____ _Владислав КУЗНЄЦОВ
(підпис) (власне ім'я та прізвище)

Керівник кваліфікаційної роботи _____ _Віра МАЛКІНА_
(підпис) (власне ім'я та прізвище)

РЕФЕРАТ

Кваліфікаційна робота: 136 с., 15 рисунків, 8 табл., 25 джерел.

Актуальність: Онкологічні захворювання є однією з ключових проблем сучасної охорони здоров'я. Рання та точна діагностика значно підвищує шанси на успішне лікування. Методи машинного навчання відкривають нові можливості для створення допоміжних інструментів, здатних аналізувати медичні дані та прогнозувати ризики, однак існує потреба в доступних, гнучких та адаптивних системах для широкого впровадження.

Об'єкт дослідження: Процес прогнозування ризику розвитку онкологічних захворювань на основі аналізу цитологічних даних пацієнтів.

Предмет дослідження: Інформаційна система для прогнозування онкологічних ризиків, що включає модулі машинного навчання, адаптивного донавчання моделі та автоматичної оптимізації гіперпараметрів.

Мета роботи: Розробка та реалізація інформаційної системи «OncoScreen Assist», що надає зручний веб-інтерфейс для прогнозування ризику онкологічних захворювань та дозволяє адаптувати прогностичну модель на основі нових даних.

Завдання:

– Проаналізувати предметну область та існуючі аналоги.
– Обґрунтувати вибір методів та технологій (XGBoost, Optuna, Flask).
– Розробити архітектуру та реалізувати модулі системи.
– Створити користувацький веб-інтерфейс.
– Провести тестування та оцінити якість розробленої системи.

Основні результати: Розроблено повнофункціональну інформаційну систему «OncoScreen Assist». Реалізовано унікальний механізм адаптивного донавчання моделі з автоматичною оптимізацією гіперпараметрів, що є ключовою перевагою. Створено інтуїтивно зрозумілий веб-інтерфейс. Тестування підтвердило високу прогностичну здатність системи (точність 94.44%, ROC AUC 0.9879).

Практичне значення дослідження: Розроблена система може бути використана як безкоштовний допоміжний інструмент у медичних закладах для попереднього скринінгу пацієнтів, що сприятиме ранньому виявленню ризиків та оптимізації діагностичних процесів.

Ключові слова: машинне навчання, прогнозування ризику, онкологія, XGBoost, Optuna, адаптивне донавчання, інформаційна система, веб-інтерфейс, оптимізація гіперпараметрів.

SUMMARY

Qualification Thesis: 136 p., 15 fig., 8 tab., 25 ref.

Actuality: Oncological diseases are a key challenge in modern healthcare. Early and accurate diagnosis significantly improves the chances of successful treatment. Machine learning methods offer new opportunities for creating auxiliary tools capable of analyzing medical data and predicting risks. However, there is a need for accessible, flexible, and adaptive systems for widespread implementation.

Object of Research: The process of predicting the risk of developing oncological diseases based on the analysis of patient cytological data.

Subject of Research: An information system for oncology risk prediction, which includes modules for machine learning, adaptive model retraining, and automatic hyperparameter optimization.

Purpose of the Thesis: To develop and implement the «OncoScreen Assist» information system, which provides a user-friendly web interface for predicting the risk of oncological diseases and allows for the adaptation of the prognostic model based on new data.

Tasks:

- To analyze the subject area and existing analogues.
- To justify the choice of methods and technologies (XGBoost, Optuna, Flask).
- To design the architecture and implement the system modules.
- To create a user-friendly web interface.
- To test and evaluate the quality of the developed system.

Main Results: A fully functional information system, «OncoScreen Assist,» has been developed. A unique mechanism for adaptive model retraining with automatic hyperparameter optimization, which is a key advantage, has been implemented. An intuitive web interface has been created. Testing confirmed the high predictive ability of the system (accuracy 94.44%, ROC AUC 0.9879).

Practical Significance: The developed system can be used as a free auxiliary tool in medical institutions for preliminary patient screening, which will contribute to the early detection of risks and the optimization of diagnostic processes.

Keywords: machine learning, risk prediction, oncology, XGBoost, Optuna, adaptive retraining, information system, web interface, hyperparameter optimization.

ЗМІСТ

ВСТУП

Актуальність теми. У сучасному світі онкологічні захворювання залишаються однією з найсерйозніших загроз для здоров'я населення, що підкреслює критичну важливість їхньої своєчасної та точної діагностики. Раннє виявлення патологій значно підвищує шанси на успішне лікування та покращує якість життя пацієнтів. Останніми роками методи машинного навчання (МН) та штучного інтелекту (ШІ) демонструють значний потенціал у трансформації підходів до медичної діагностики [15, 19], пропонуючи нові інструменти для аналізу великих обсягів даних [3, 9] та виявлення прихованих закономірностей, недоступних для традиційних методів [7, 12, 17].

Розробка інформаційних систем (ІС), здатних прогнозувати ризик розвитку захворювань на основі клінічних та інструментальних даних, стає все більш затребуваною. Такі системи можуть слугувати цінним допоміжним інструментом для медичних працівників, сприяючи об'єктивнішій оцінці стану пацієнта та допомагаючи у прийнятті клінічних рішень, особливо на етапі попереднього скринінгу. Проте багато наявних рішень або статичні, або вимагають складних процедур для адаптації до нових даних чи специфіки конкретного медичного закладу. Існує потреба в адаптивних системах, які могли б не лише давати точні прогнози, а й безперервно покращувати свою якість у міру накопичення нових даних, а також бути доступними та простими у використанні для медичного персоналу, який не має глибоких технічних знань. Важливим аспектом також є кросплатформність таких рішень, що забезпечує їхню доступність у різноманітній ІТ-інфраструктурі медичних закладів.

Ступінь розробленості проблеми. Проблематиці застосування методів машинного навчання для завдань медичної діагностики, зокрема прогнозування онкологічних захворювань, присвячена значна кількість наукових праць як вітчизняних, так і зарубіжних авторів. Досліджено різноманітні алгоритми класифікації, методи обробки медичних зображень та аналізу генетичних да-

них. Водночас питання створення комплексних інформаційних систем, що поєднують високу точність прогнозування з адаптивністю моделі завдяки механізму донавчання на накопичуваних даних та інтуїтивно зрозумілим користувацьким інтерфейсом, потребують подальшого вивчення та практичної реалізації.

Мета. Метою цієї дипломної роботи є розробка інформаційної системи для прогнозування ризику онкологічних захворювань із використанням методів машинного навчання, що має можливість донавчання моделі та зручний користувацький веб-інтерфейс.

Для досягнення поставленої мети було сформульовано такі завдання:

1. Провести аналіз предметної області, наявних підходів та програмних інструментів для прогнозування захворювань методами машинного навчання.
2. Обрати й теоретично обґрунтувати метод машинного навчання (класифікатор XGBoost) та стек технологій для реалізації інформаційної системи.
3. Розробити алгоритм попередньої обробки вхідних даних, що включає масштабування ознак, та методику відбору найінформативніших ознак.
4. Реалізувати програмні модулі для навчання класифікаційної моделі та здійснення прогнозування ризику захворювання.
5. Розробити та інтегрувати в систему механізм донавчання (перенавчання) моделі на основі нових, накопичуваних даних, що включає оновлення параметрів масштабування та перевизначення набору ключових ознак.
6. Створити користувацький вебінтерфейс для взаємодії із системою, що забезпечує завантаження даних, отримання прогнозів та ініціювання процесу донавчання моделі.
7. Провести тестування розробленої інформаційної системи та виконати оцінку якості прогнозування з використанням релевантних метрик.

Об'єкт дослідження. Об'єктом дослідження є процес прогнозування ризику розвитку онкологічних захворювань на основі аналізу деперсоналізованих даних про пацієнтів.

Предмет дослідження. Предметом дослідження є інформаційна система, що реалізує методи машинного навчання для класифікації та прогнозування ризику онкологічних захворювань і включає функціонал адаптивного донавчання моделі.

Методи дослідження. Під час виконання дипломної роботи використовувалися такі методи:

– теоретичні: аналіз і синтез наукової та технічної літератури за темою дослідження, системний аналіз;

– методи машинного навчання: методи класифікації (зокрема, алгоритм градієнтного бустингу XGBoost), методи оцінки важливості та відбору ознак;

– статистичні методи: методи оцінки якості класифікаційних моделей (точність (Accuracy), повнота (Recall), точність прогнозу (Precision), F1-міра, аналіз матриці помилок);

– методи розробки програмного забезпечення: об'єктно-орієнтоване програмування, розробка веб-застосунків (з використанням мов програмування Python, фреймворку Flask, технологій HTML, CSS, JavaScript), контейнеризація застосунків (Docker).

Практичне значення отриманих результатів. Розроблена інформаційна система може бути використана як допоміжний інструмент у клінічній практиці для проведення попередньої оцінки ризику онкологічних захворювань, сприяючи ранньому виявленню пацієнтів групи ризику та їхньому своєчасному направленню на поглиблене обстеження. Реалізована функція донавчання моделі на нових даних дозволяє адаптувати систему до специфіки даних конкретного медичного закладу за умови накопичення достатнього обсягу локальної верифікованої інформації, потенційно підвищуючи точність прогнозів для локальної популяції пацієнтів. Інтуїтивно зрозумілий вебінтерфейс та використання вебтехнологій забезпечують кросплатформну доступність системи та простоту її експлуатації медичним персоналом без спеціальних навичок у галузі програмування чи аналізу даних.

Структура та обсяг роботи. Дипломна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 25 найменувань та 6 додатків. Загальний обсяг роботи становить 64 сторінок машинописного тексту.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ

1.1 Узагальнена характеристика предметної області

Онкологічні захворювання є однією з найбільш значущих медико-соціальних проблем сучасного суспільства в усьому світі, і Україна не є винятком. Злоякісні новоутворення посідають провідні позиції у структурі смертності та інвалідизації населення, завдаючи істотної економічної шкоди державі та значно знижуючи якість життя пацієнтів та їхніх родин. За даними Національного канцер-реєстру України, щорічно реєструється значна кількість нових випадків раку. Наприклад, навіть до повномасштабного вторгнення, у 2020-2021 роках, кількість уперше виявлених випадків злоякісних новоутворень становила понад 120 тисяч щорічно, а смертність від раку залишалася на високому рівні, будучи однією з основних причин передчасної смерті населення [5, 8].

Ключовим фактором, що визначає успішність лікування онкологічних захворювань та прогноз для пацієнта, є своєчасність їх виявлення. Діагностика раку на ранніх стадіях, коли пухлина локалізована і не встигла дати метастази, дозволяє застосовувати більш щадні та ефективні методи лікування, що суттєво підвищує шанси на повне одужання або досягнення тривалої ремісії. Однак рання діагностика часто ускладнена через неспецифічність симптомів на початкових етапах, недостатню онкологічну настороженість як серед населення, так і серед лікарів первинної ланки, а також обмеженість ресурсів для проведення масових скринінгових програм.

У цьому контексті інформаційні технології (ІТ) і, зокрема, методи машинного навчання (МН) відкривають нові перспективи для вдосконалення процесів діагностики та прогнозування в онкології [4]. Штучний інтелект зда-

тний аналізувати величезні масиви медичних даних, включно з клінічними показниками, результатами лабораторних та інструментальних досліджень, даними медичної візуалізації та навіть генетичною інформацією, виявляючи неочевидні для людини закономірності та предиктори розвитку захворювань. Системи на основі МН можуть обробляти комплексні набори ознак, що характеризують стан пацієнта або особливості новоутворення, і на їх основі будувати прогностичні моделі.

Основними завданнями, що вирішуються в предметній області онкології за допомогою інформаційних систем, які використовують машинне навчання, є:

– скринінг та формування груп ризику: автоматизований аналіз даних великої кількості пацієнтів для виявлення осіб із підвищеною ймовірністю розвитку онкологічних захворювань, що дозволяє оптимізувати профілактичні заходи та сфокусувати діагностичні зусилля;

– підтримка прийняття лікарських рішень: надання лікарям об'єктивної, заснованої на даних, оцінки ризику захворювання або ймовірності певного результату, що може слугувати допоміжним інструментом під час встановлення попереднього діагнозу або вибору тактики ведення пацієнта;

– зниження навантаження на медичних фахівців: автоматизація рутинних етапів аналізу даних, що дозволяє лікарям сконцентруватися на складніших випадках та безпосередній взаємодії з пацієнтами;

– персоналізація підходів: у перспективі аналіз індивідуальних даних пацієнта може сприяти розробці персоналізованих стратегій профілактики та лікування;

– покращення результатів лікування завдяки ранньому втручанню: сприяючи більш ранньому виявленню захворювань, такі системи опосередковано впливають на підвищення ефективності лікування.

Дані, що використовуються для побудови прогностичних моделей в онкології, є надзвичайно різноманітними. Вони можуть включати демографічну інформацію, анамнестичні дані, результати загальних та біохімічних аналізів

крові, дані цитологічних і гістологічних досліджень (наприклад, характеристики клітин, отриманих під час біопсії, як-от розмір, форма, текстура ядра та цитоплазми тощо), результати променевої діагностики (КТ, МРТ, УЗД), генетичні маркери та багато іншого. У контексті цієї роботи передбачається використання набору числових ознак, отриманих у результаті аналізу клітинних характеристик, для побудови моделі прогнозування ризику онкологічного процесу.

1.1.1 Цитологічні основи прогнозування онкологічних захворювань: від морфології клітини до ознак для машинного навчання

Основою для більшості сучасних методів діагностики та прогнозування онкологічних захворювань слугує аналіз морфологічних характеристик клітин, що отримуються з біологічних зразків пацієнта. Візуальна оцінка клітинних структур під мікроскопом дозволяє виявляти ознаки атипії та злоякісної трансформації. Саме ці видимі зміни лягли в основу набору ознак, що використовуються в даній роботі для побудови моделі машинного навчання [13, 24].

На рисунку 1.1 представлена ілюстрація мікроскопічного препарату, подібного до тих, що аналізуються в цитологічних лабораторіях (дане зображення надано з дослідницьких матеріалів фірми-партнера).

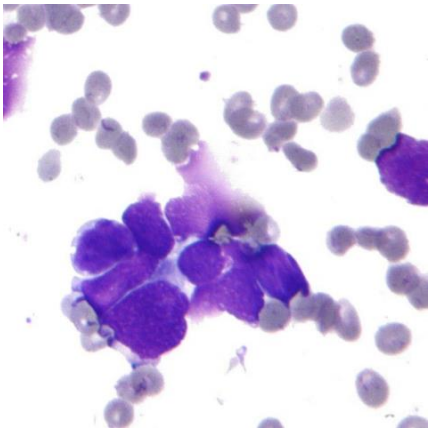

Рисунок 1.1 – Приклад мікроскопічного зображення клітинного препарату, що використовується для цитологічного аналізу

На зображенні видно клітини різної форми та розміру. В контексті онкодіагностики особлива увага приділяється великим клітинам з вираженими ядрами. Кількісна оцінка різних параметрів цих ядер та самих клітин дозволяє формувати набір числових ознак для подальшого аналізу методами машинного навчання. Злоякісні клітини часто демонструють низку характерних морфологічних відхилень від норми.

Ознаки, на яких навчається прогностична модель в даній інформаційній системі, є кількісними дескрипторами цих візуальних клітинних характеристик. Розглянемо основні з них:

– radius_mean (середній радіус ядра): Вимірюється як середня відстань від центру ядра до точок на його периметрі. Збільшення середнього радіуса ядер є одним з частих ознак атипії, оскільки ядра злоякісних клітин часто стають більшими за розміром (анізонуклеоз);

– texture_mean (середня текстура ядра): Кількісно описує варіабельність відтінків сірого всередині ядра, що відображає неоднорідність розподілу хроматину. Для злоякісних клітин характерний більш грубий, нерівномірно розподілений хроматин, що призводить до збільшення цього показника;

– perimeter_mean (середній периметр ядра): Довжина контуру ядра. Збільшений та нерівний периметр, часто пов'язаний зі збільшенням розміру та неправильною формою ядра, є характерною ознакою малігнізації;

– area_mean (середня площа ядра): Площа, яку займає ядро клітини. Аналогічно радіусу та периметру, збільшення середньої площі ядер може свідчити про злоякісний процес;

– smoothness_mean (середня гладкість контуру ядра): Оцінює локальні варіації довжин радіус-векторів, проведених від центру до точок на контурі ядра. Більш «колючі», нерівні контури злоякісних ядер призводять до вищих значень цього показника порівняно з гладкими контурами нормальних клітин;

– compactness_mean (середня компактність ядра): Розраховується як (периметр^2 / площа - 1). Цей безрозмірний показник характеризує, наскільки

форма ядра близька до ідеального кола. Злоякісні клітини часто мають менш компактні, більш витягнуті або неправильної форми ядра;

– concavity_mean (середня увігнутість контуру ядра): Кількісно оцінює ступінь та кількість увігнутостей (впадин, «затоків») на контурі ядра. Наявність глибоких та численних увігнутостей є характерною ознакою злоякісних клітин, контури ядер яких часто стають нерівними та фестончатими;

– concave points_mean (середня кількість увігнутих точок на контурі ядра): Підраховує кількість увігнутих ділянок на периметрі ядра. Цей показник також відображає ступінь нерегулярності форми ядра, яка збільшується при малігнізації;

– symmetry_mean (середня симетрія ядра): Оцінює, наскільки ядро симетричне. Злоякісні клітини часто втрачають симетрію форми ядра, що проявляється у вищих значеннях асиметрії (або нижчих значеннях симетрії, залежно від способу розрахунку);

– fractal_dimension_mean (середня фрактальна розмірність контуру ядра): Математичний параметр, що описує складність та «порізаність» контуру ядра. Більш складні, нерівні, «мереживні» контури злоякісних ядер мають вищу фрактальну розмірність.

Для кожної з цих десяти основних морфологічних характеристик в наборі даних також розраховуються:

– стандартна помилка (_se – standard error): Відображає варіабельність або розкид значень даної ознаки серед виміряних клітин в одному зразку. Більша стандартна помилка може вказувати на вищий ступінь гетерогенності (різноманітності) клітин, що також може бути пов'язано з патологічним процесом;

– найгірше або максимальне значення (_worst): Представляє середнє значення трьох найбільших (або «найгірших» з точки зору відхилення від норми) значень даної ознаки, виявлених у клітинах зразка. Ці «найгірші» показ-

ники часто є більш чутливими індикаторами злоякісності, ніж середні значення, оскільки навіть невелика кількість клітин з вираженою атипією може бути діагностично значущою.

Комплексний аналіз усіх цих 30 ознак (10 середніх, 10 стандартних помилок та 10 «найгірших» значень) дозволяє моделі машинного навчання виявляти складні патерни, характерні для онкологічних захворювань, та робити прогностичні висновки. Таким чином, інформаційна система автоматизує та об'єктивізує процес аналізу, який традиційно виконується спеціалістом-цитологом на основі візуальної оцінки.

1.1.2 Роль гіперпараметрів та їх автоматичної оптимізації в системі

Ефективність прогностичної моделі XGBoost, що є ядром системи «OncoScreen Assist», критично залежить від правильного налаштування її гіперпараметрів [14]. Це зовнішні конфігураційні змінні, які задаються до початку навчання моделі та керують її поведінкою, складністю та здатністю до узагальнення.

Ручний підбір оптимальної комбінації гіперпараметрів є неефективним та практично неможливим через їхню велику кількість та складну взаємодію. Тому для забезпечення максимальної точності та адаптивності системи було реалізовано механізм автоматичної оптимізації гіперпараметрів (HPO – Hyperparameter Optimization).

У даній системі цей процес реалізовано за допомогою фреймворку Optuna [10] та інтегровано в функціонал донавчання моделі. Процес оптимізації включає наступні ключові етапи, що безпосередньо відображені в програмному коді:

1. Визначення цільової функції: В якості метрики для оптимізації було обрано ROC AUC, оскільки вона є інтегральним показником якості бінарної класифікації, стійким до дисбалансу класів. Оцінка проводиться за допомогою

5-кратної стратифікованої крос-валідації (StratifiedKFold), що забезпечує надійність результату.

2. Визначення простору пошуку гіперпараметрів: В ході дослідження (study.optimize) Optuna ітеративно підбирає значення для ключових гіперпараметрів XGBoost. Для оптимізації було обрано саме цей набір гіперпараметрів, оскільки вони мають найбільший вплив на баланс між точністю та складністю моделі, дозволяючи досягти максимального приросту якості за прийнятний час обчислень. Розглянемо їх вплив на модель:

– n_estimators (кількість дерев): Визначає розмір ансамблю. Недостатня кількість дерев може призвести до недонавчання (underfitting), коли модель занадто проста і не здатна вловити складні залежності в даних. Надмірна кількість, хоча і може підвищити точність, значно збільшує час навчання та ризик перенавчання (overfitting), коли модель «запам'ятовує» тренувальні дані замість того, щоб узагальнювати.

– learning_rate (швидкість навчання): Цей параметр масштабує внесок кожного нового дерева. Низьке значення (напр. 0.01) змушує модель навчатися повільніше і обережніше, що робить її більш стійкою до перенавчання, але вимагає більшої кількості n_estimators. Високе значення прискорює навчання, але може призвести до того, що модель «пропустить» оптимальне рішення.

– max_depth (максимальна глибина дерева): Контролює складність кожного окремого дерева в ансамблі. Невелике значення (напр. 3-5) обмежує модель, змушуючи її створювати прості правила, що добре узагальнюються. Велика глибина дозволяє моделі будувати дуже складні правила для виявлення специфічних патернів, але це значно підвищує ризик перенавчання на шумі в даних;

– subsample та colsample_bytree (параметри стохастичності): subsample визначає частку рядків (пацієнтів), а colsample_bytree — частку ознак (цитологічних параметрів), що випадковим чином обираються для побудови кожного

нового дерева. Використання значень менше 1.0 (напр., 0.8) вносить стохастичність у процес навчання, що є ефективним методом боротьби з перенавчанням, оскільки кожне дерево бачить дещо різний зріз даних.

Для забезпечення відтворюваності результатів та прозорості процесу оптимізації, в таблиці 1.1 наведено простір пошуку, що використовувався фреймворком Optuna, а також оптимальні значення, знайдені в результаті дослідження для фінальної моделі.

Таблиця 1.1 – Простір пошуку та оптимальні значення гіперпараметрів

Гіперпараметр	Діапазон пошуку (простір пошуку)	Оптимальне значення
n_estimators	50..300 (крок 25)	225
learning_rate	0,01..0,3 (крок 0,05)	0,15
max_depth	3..9 (крок 1)	7
subsample	0,5..1,0 (крок 0.1)	0,75
colsample_bytree	0,5..1,0 (крок 0.1)	0,9
gamma	0,0..0,5 (крок 0.1)	0,2

Вибір саме такого простору пошуку дозволив збалансувати дослідження як простих, так і складних конфігурацій моделі, що в підсумку дало змогу досягти високої прогностичної точності.

3. Пошук оптимальних значень: Optuna проводить задану кількість ітерацій (N_OPTUNA_TRIALS = 25), на кожній з яких тестує нову комбінацію гіперпараметрів, знаходячи баланс між складністю моделі та її узагальнюючою здатністю.

4. Тренування фінальної моделі: Після завершення дослідження система використовує найкращу знайдену комбінацію гіперпараметрів

(study.best_trial.params) для навчання фінальної прогностичної моделі на оновленому наборі даних.

Хоча існують і інші гіперпараметри (наприклад, min_child_weight), їх вплив зазвичай є більш тонким. Фокусування на наведеному наборі дозволяє ефективно керувати основними аспектами поведінки моделі: її розміром, швидкістю навчання, складністю та стійкістю до перенавчання, що є оптимальною стратегією для даної задачі.

Такий підхід дозволяє системі «OncoScreen Assist» не просто донавчатися на нових даних, а робити це максимально ефективно, автоматично адаптуючи складність та характеристики моделі для досягнення найкращої прогностичної здатності. Це є однією з ключових переваг розробленого рішення.

1.2 Огляд і аналіз існуючих аналогів системи

Для визначення місця розробленої інформаційної системи OncoScreen Assist на ринку сучасних діагностичних рішень було проведено порівняльний аналіз з двома відомими комерційними аналогами: iCAD Profound AI Suite та Grail Galleri. Ці аналоги були обрані, оскільки вони представляють два різні передові підходи до діагностики: аналіз медичних зображень за допомогою ШІ та аналіз крові методом «рідкої біопсії».

Порівняння проводилося за ключовими технічними, економічними та експлуатаційними характеристиками, такими як тип продукту, використовувані дані, метрики ефективності, модель розгортання, вартість, прозорість, можливість адаптації та вимоги до інфраструктури. Результати порівняльного аналізу представлені в таблиці 1.2.

Аналіз результатів порівняння

Як видно з таблиці, розроблена система OncoScreen Assist займає унікальну нішу. На відміну від дорогих, вузькоспеціалізованих та закритих комерційних систем, вона пропонує гнучке, прозоре та доступне рішення.

Таблиця 1.2 – Порівняльний аналіз розробленої системи з комерційними аналогами

Характеристика (критерій порівняння)	OncoScreen Assist (наша розробка)	iCAD Profound AI Suite	Grail Galleri
1	2	3	4
Тип продукту	Програмний інструмент для класифікації ризику на основі табличних даних + донавчання; легкий GUI	Система аналізу зображень на основі ШІ та оцінки ризику з мамограм	Мульти-раковий скринінговий тест на основі аналізу крові (рідка біопсія)
Основний тип вхідних даних/Метод	Табличні CSV-дані (цитологічні/морфологічні ознаки клітин); алгоритмічна модель (XGBoost з HPO)	Мамографічні зображення (2D/3D) + аналіз ШІ	Метилювання циркулюючої вільної ДНК (cfDNA) в крові
Ключові метрики ефективності (заявлені/типові)	Залежать від моделі/даних; донавчання дозволяє ітеративне покращення; повний ROC/матриця помилок для пояснення	AUC ~0.82 (DBT); 2.4x точніше, ніж Gail/Tyrer-Cuzick	Стадія I: 16.8%, Стадія II: 40.4% чутливість; загальна специфічність 99.5%

Продовження таблиці 1.2

1	2	3	4
Розгортання/Застосування	Локально на ПК/ноутбуці (Windows/macOS/Linux); завантаження CSV в один клік; швидке отримання результату (1-5 сек)	Хмарне або локальне (on-prem); інтеграція з PACS у понад 50 системах	Зразки крові відправляються до центральної лабораторії; обробка займає тижні
Вартість використання (приблизна)	Безкоштовно та з відкритим вихідним кодом – без витрат на використання; ідеально для бюджетних або дослідницьких завдань	Ліцензування дороге; інтеграція в радіологію; зазвичай дуже висока вартість	~ $949/тест (США, роздріб); аналітичні лабораторії – відшкодування на розгляді
Прозорість та інтерпретованість	Повністю відкритий вихідний код; візуалізація важливості ознак, ROC, матриця помилок, кастомізовані пороги	Пропрієтарні нейронні мережі; виведення карт важливості, оцінки ризику	Пропрієтарні cfDNA алгоритми; обмежена публічна інтерпретованість
Власність моделі та Донавчання	Повністю керується користувачем; донавчання локально; інтеграція нових локальних даних; контроль версій	Керується вендором; самонавчання не типове	Закрита; оновлення через лабораторію вендора

Продовження таблиці 1.2

1	2	3	4
Вимоги до інфраструктури	Локальна машина; відсутність потреби в EHR/PACS; працює на стандартному ноутбуці/ПК	Інтеграція з PACS/RIS; GPU/обчислювальні ресурси для навчання; IT-підтримка	Флеботомія, центральна лабораторія, логістика, регуляторна інфраструктура
Ідеальний сценарій використання	Дослідження; клінічні умови з обмеженими ресурсами; навчальні проекти; використання, орієнтоване на прозорість	Радіологічні центри для персоналізованого скринінгу раку молочної залози	Популяційний мульти-раковий скринінг; ринок ранніх послідовників
Обмеження	Не сертифіковано; відсутнє схвалення регуляторних органів; точність залежить від якості вхідних даних	Висока вартість; обмежено мамографією; пропрієтарність; потребує доступу до зображень	Висока вартість тесту; низька чутливість для ранніх стадій; відшкодування не гарантоване; невизначеність з регулюванням

Ключовими перевагами та відмінними рисами OncoScreen Assist є:

– доступність та низькі вимоги до інфраструктури: Система не потребує дорогого обладнання, інтеграції з PACS/EHR чи спеціалізованих лабораторій.

Вона може бути розгорнута на звичайному персональному комп'ютері, що робить її ідеальною для установ з обмеженими ресурсами, дослідницьких груп та освітніх цілей;

– прозорість та інтерпретованість: Завдяки відкритому вихідному коду та вбудованим інструментам візуалізації (важливість ознак, ROC-криві, матриця помилок) користувач має повне уявлення про те, як модель приймає рішення. Це кардинально відрізняє її від «чорних скриньок» пропрієтарних аналогів;

– повний контроль та адаптивність: Функція донавчання (перенавчання) на локальних даних є головною конкурентною перевагою. Користувач може самостійно адаптувати та покращувати модель на основі власних, верифікованих даних, підвищуючи її точність для специфіки конкретної популяції пацієнтів. Комерційні аналоги такої гнучкості не надають;

– нульова вартість: Безкоштовне розповсюдження усуває фінансовий бар'єр для впровадження сучасних методів аналізу даних у клінічну чи наукову практику.

Водночас основним обмеженням є відсутність на даний момент медичної сертифікації, що визначає її поточний статус як допоміжного або дослідницького інструменту, а не засобу для встановлення остаточного діагнозу. Тим не менш, для завдань попереднього скринінгу, формування груп ризику та наукових досліджень OncoScreen Assist є потужним та унікальним рішенням на ринку.

1.3 Розробка технічного завдання

На основі проведеного аналізу предметної області, огляду існуючих аналогів та з урахуванням виявлених потреб було сформульовано технічне завдання (ТЗ) на розробку інформаційної системи прогнозування ризику онко-

логічних захворювань методами машинного навчання. Технічне завдання є основоположним документом, що визначає основні вимоги до системи, її функціональність, архітектуру та етапи розробки.

1. Основні положення:

– найменування системи: Інформаційна система прогнозування ризику онкологічних захворювань (далі – Система);

– призначення системи: Надання інструменту для медичних працівників для оцінки ризику наявності онкологічного процесу на основі аналізу цитологічних ознак клітин пацієнта з використанням моделі машинного навчання. Система повинна підтримувати адаптацію моделі на нових даних;

– підстава для розробки: Дипломна робота. Потреба у створенні доступного, крос-платформного та адаптивного інструменту для підтримки прийняття рішень в онкодіагностиці.

2. Вимоги до системи:

– завантаження вхідних даних пацієнта у форматі CSV-файлу з уніфікованими цитологічними ознаками;

– попередня обробка даних, включаючи масштабування ознак;

– прогнозування ризику онкологічного захворювання з відображенням ступеня впевненості;

– збереження результатів прогнозу;

– механізм донавчання моделі на основі нових верифікованих даних;

– об'єднання нових даних з наявним набором;

– перенавчання масштабувальника ознак (StandardScaler);

– переоцінка найбільш інформативних ознак;

– підбір оптимальних гіперпараметрів моделі XGBoost з використанням сучасних методів оптимізації (наприклад, Optuna);

– перенавчання моделі на оновленому наборі даних;

– збереження оновленої моделі, скейлера та ознак;

– надання зворотного зв'язку про статус виконання дій (аналіз, донавчання);

– забезпечення стабільної роботи при коректних даних;

– обробка типових помилок (невірний формат, відсутність обов'язкових значень);

– ітуїтивно зрозумілий інтерфейс для користувачів без спеціальної IT-підготовки;

– простий механізм завантаження файлів та отримання результатів;

– візуальне розділення функцій прогнозу і донавчання;

– реалізація у вигляді веб-додатку з крос-платформною підтримкою (доступ через браузер у Windows, MacOS, Linux);

– бекенд реалізований на Python із використанням Flask;

– модель машинного навчання реалізується з використанням бібліотеки XGBoost;

– для оптимізації гіперпараметрів використовується Optuna;

– можливість контейнеризації через Docker.

3. Етапи розробки:

– аналіз предметної області та існуючих рішень (завершено);

– розробка технічного завдання (поточний етап);

– проєктування архітектури системи та структури зберігання даних;

– розробка моделі машинного навчання (навчання, відбір ознак, оптимізація);

– розробка серверної частини веб-додатку;

– розробка клієнтської частини веб-додатку;

– реалізація механізму донавчання моделі;

– тестування та налагодження системи;

– підготовка документації.

4. Порядок контролю та приймання:

– приймання системи здійснюється шляхом демонстрації працездатності всіх заявлених функцій відповідно до ТЗ;

– оцінка якості моделі проводиться за метриками Accuracy, Recall, Precision, F1-score на тестовій вибірці. Пріоритет — Recall для класу з високим ризиком.

Дане технічне завдання слугує основою для подальшої розробки та реалізації інформаційної системи.

РОЗДІЛ 2
СПЕЦИФІКАЦІЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Глосарій

Для забезпечення однозначного розуміння предметної області та технічних аспектів, представлених у даній кваліфікаційній роботі, у цьому розділі наведено глосарій. Він містить визначення ключових термінів, понять та скорочень, що використовуються в подальшому викладі. Глосарій структурований для зручності та охоплює поняття з онкології, машинного навчання, метрик оцінки якості моделей та програмної інженерії, що є фундаментальними для повного розуміння системи, що розробляється.

Таблиця 2.1 – Глосарій ключових термінів та понять

Термін	Визначення
1	2
Онкологічне захворювання (рак)	Загальна назва для великої групи захворювань, що характеризуються неконтрольованим ростом аномальних клітин, здатних проростати в сусідні тканини та поширюватися на інші органи.
Прогнозування ризику	Процес оцінки ймовірності виникнення або наявності певного захворювання (в даному випадку – онкологічного) у пацієнта на основі аналізу його індивідуальних даних.

Продовження таблиці 2.1

1	2
Машинне навчання (МН)	Галузь штучного інтелекту, що вивчає методи побудови алгоритмів, здатних навчатися на основі даних та робити прогнози або приймати рішення без явного програмування.
Класифікація (в машинному навчанні)	Задача машинного навчання, що полягає у віднесенні об'єкта до одного з попередньо визначених класів на основі його ознак (наприклад, «високий ризик» / «низький ризик») [1, 11].
XGBoost (Extreme Gradient Boosting)	Ефективний та широко використовуваний алгоритм машинного навчання, заснований на методі градієнтного бустингу [16] на деревах рішень; відомий своєю високою точністю.
Оптимізація гіперпараметрів (HPO - Hyperparameter Optimization)	Процес автоматичного підбору найкращих значень для параметрів моделі машинного навчання (гіперпараметрів), які не визначаються в ході самого навчання, з метою покращення її якості.
Optuna	Сучасний програмний фреймворк, призначений для автоматизації процесу оптимізації гіперпараметрів моделей машинного навчання.

Продовження таблиці 2.1

1	2
TPE (Tree-structured Parzen Estimator)	Байєсівський алгоритм оптимізації, що використовується у фреймворку Optuna для ефективного пошуку оптимальних гіперпараметрів.
Ознака (в машинному навчанні) / Предиктор (Feature / Predictor)	Індивідуальна вимірювана властивість або характеристика спостережуваного явища (об'єкта), що використовується моделлю машинного навчання для прогнозування.
Цитологічні ознаки	Кількісні та якісні характеристики клітин та їх структур (наприклад, ядра), що визначаються при мікроскопічному дослідженні та використовуються для діагностики.
StandardScaler (стандартизація даних)	Метод попередньої обробки числових ознак, що полягає у їх масштабуванні таким чином, щоб середнє значення стало рівним нулю, а стандартне відхилення – одиниці.
Важливість ознак (Feature Importance)	Міра, що показує відносний внесок кожної вхідної ознаки у здатність моделі машинного навчання робити точні прогнози.

Продовження таблиці 2.1

1	2
Донавчання моделі (перенавчання моделі)	Процес оновлення або повної перебудови існуючої моделі машинного навчання на основі нових, додатково отриманих даних, з метою покращення її точності або адаптації.
Тестова вибірка (Test Set)	Частина набору даних, що не використовувалася при навчанні моделі, а призначена для незалежної оцінки її здатності до узагальнення та якості прогнозування.
Точність (Accuracy)	Частка правильно класифікованих об'єктів від загальної кількості об'єктів у вибірці.
Повнота (Recall / Sensitivity)	Частка правильно ідентифікованих об'єктів позитивного класу серед усіх об'єктів, що реально належать до позитивного класу.
Точність прогнозу (Precision)	Частка правильно ідентифікованих об'єктів позитивного класу серед усіх об'єктів, які модель віднесла до позитивного класу.
F1-міра (F1-Score)	Гармонійне середнє між точністю прогнозу (Precision) та повнотою (Recall), що дає збалансовану оцінку якості класифікатора.

Продовження таблиці 2.1

1	2
AUC ROC (Area Under the ROC Curve)	Площа під ROC-кривою (кривою робочих характеристик приймача); інтегральна метрика, що відображає здатність моделі розрізняти класи при різних порогах класифікації.
Матриця помилок (Confusion Matrix)	Таблиця, що відображає результати роботи класифікатора, показуючи кількість істинно позитивних (TP), хибно позитивних (FP), істинно негативних (TN) та хибно негативних (FN) прогнозів.
Веб-додаток	Програмний додаток, доступ до якого здійснюється через веб-браузер по мережі (інтернет або локальна мережа).
Flask	Мікрофреймворк для розробки вебдодатків на мові програмування Python, відомий своєю простотою та гнучкістю.
Docker (контейнеризація)	Платформа для розробки, доставки та запуску додатків у стандартизованих ізольованих середовищах, що називаються контейнерами.
CSV (Comma-Separated Values)	Текстовий формат файлу, призначений для представлення табличних даних, де значення в стовпцях розділені комами (або іншим роздільником).

Продовження таблиці 2.1

1	2
Користувацький інтерфейс (GUI - Graphical User Interface)	Графічна оболонка програми, що забезпечує взаємодію користувача з системою за допомогою візуальних елементів (кнопки, меню, вікна тощо).
Бекенд (Backend)	Серверна частина веб-додатка, відповідальна за обробку логіки, роботу з даними та взаємодію з моделлю машинного навчання.
Фронтенд (Frontend)	Клієнтська частина веб-додатка, що відображається у браузері користувача та відповідає за візуальне представлення інформації та взаємодію з користувачем.

2.2 Концептуальна модель використання інформаційної системи

Концептуальна модель використання інформаційної системи «OncoScreen Assist» представлена у вигляді UML-діаграми варіантів використання (рисунок 2.1). Дана діаграма ілюструє основні функціональні можливості системи з точки зору взаємодії з нею ключових користувачів.

Основним актором системи є Лаборант (або Медичний працівник). Він безпосередньо взаємодіє з інтерфейсом системи для виконання аналітичних та допоміжних задач. Результати роботи системи, зокрема звіти з прогнозом, призначені для подальшого використання Лікарем при прийнятті клінічних рішень.

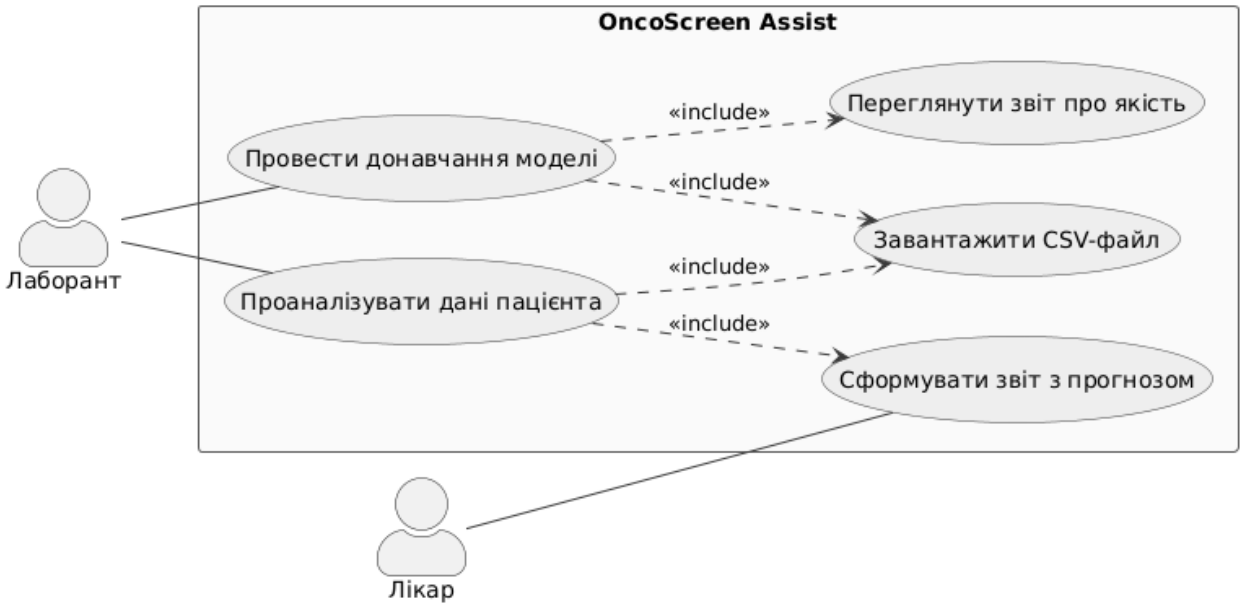

Рисунок 2.1 – Діаграма варіантів використання інформаційної системи «OncoScreen Assist»

Діаграма варіантів використання відображає наступні ключові сценарії взаємодії Лаборанта з системою:

- завантажити дані пацієнта: Дозволяє Лаборанту завантажувати вхідні дані пацієнта у вигляді CSV-файлу для аналізу;
- отримати прогноз: На основі завантажених даних Лаборант ініціює процес прогнозування для отримання оцінки ризику онкологічного захворювання;
- зберегти прогноз: Надає можливість збереження отриманого прогнозу у вигляді файлу-звіту для документування та передачі Лікарю;
- завантажити дані для донавчання: Дозволяє Лаборанту завантажувати нові набори верифікованих даних для оновлення та покращення прогностичної моделі;
- ініціювати донавчання (з HPO): Запускає процес повного перенавчання прогностичної моделі, включаючи автоматичну оптимізацію її гіперпараметрів;
- переглянути оцінку нової моделі: Надає Лаборанту доступ до детальних метрик якості оновленої моделі після завершення процесу донавчання.

Таким чином, діаграма варіантів використання наочно демонструє ключові сценарії взаємодії користувача з інформаційною системою «OncoScreen Assist» та її основний функціональний спектр, орієнтований на прогнозування та адаптацію моделі.

2.3 Розробка функціональної моделі

Функціональна модель інформаційної системи «OncoScreen Assist» описує основні процеси, що виконуються системою, та взаємозв'язки між ними на більш абстрактному рівні. Для візуалізації функціональної моделі використано нотацію, близьку до стандарту IDEF0.

На рисунку 2.2 представлена контекстна діаграма (рівень A-0), що відображає систему «OncoScreen Assist» як єдиний функціональний блок та її основні взаємозв'язки із зовнішнім середовищем.

Як видно з контекстної діаграми, основними входами (Inputs) для системи є «CSV-дані пацієнта», «CSV-дані для донавчання» та «Запит на аналіз/донавчання» від Лаборанта. Управління (Controls) роботою системи здійснюється за допомогою «Алгоритмів МО (XGBoost, Optuna)», «Вимог до формату даних» та «Налаштувань користувача». Механізмами (Mechanisms) реалізації є сам «Лаборант» як оператор та «Обчислювальні ресурси (комп'ютер, ПЗ системи)». На виході (Outputs) система генерує «Звіт: прогноз ризику (PDF)», «Звіт: якість донавченої моделі», «Оновлену модель та артефакти» (що зберігаються на сервері) та «Повідомлення/Результати» для Лаборанта.

Для більш детального аналізу основна функція системи «Прогнозування ризику онкозахворювань та адаптація моделі» була декомпозована на дві ключові підфункції, як показано на діаграмі декомпозиції першого рівня (рисунок 2.3).

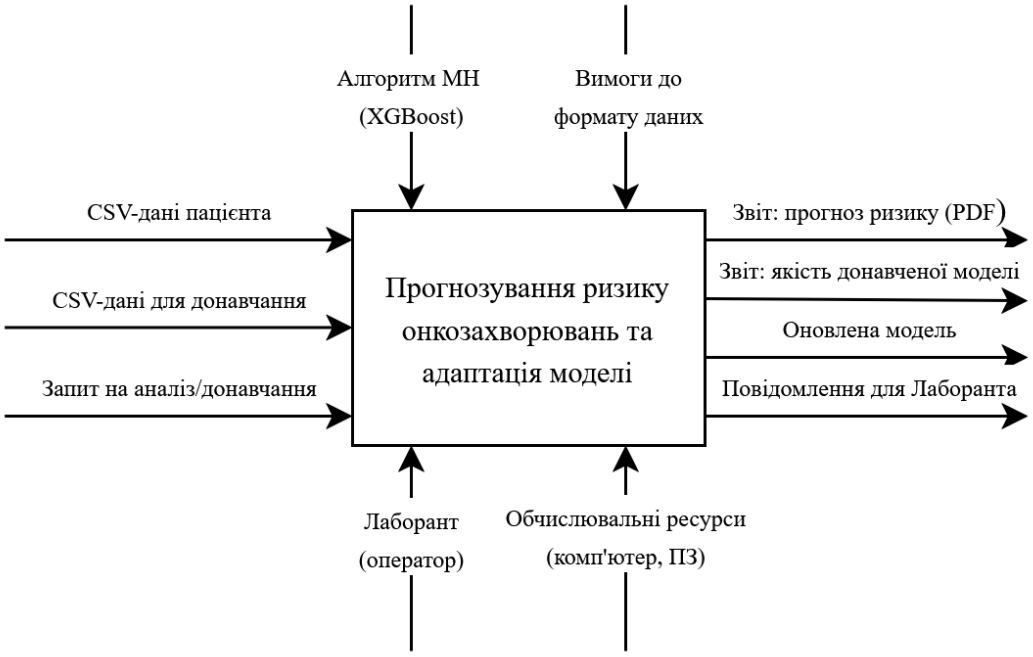

Рисунок 2.2 – Контекстна діаграма функціональної моделі системи «OncoScreen Assist»

Підфункція A1 «Прогнозування ризику захворювання» відповідає за обробку даних конкретного пацієнта та надання прогностичної оцінки.

– Входи: «CSV-дані пацієнта», «Запит на аналіз» (від Лаборанта), «Оновлена модель/Артефакти» (від підфункції A2).
– Виходи: «Звіт: Прогноз Ризику (PDF)», «Результат прогнозу» (для Лаборанта).
– Управління: «Алгоритм XGBoost (навчений)», «Правила обробки даних».
– Механізми: Спільні для системи (Лаборант, ПЗ, ресурси).

Підфункція A2 «Адаптація та донавчання прогностичної моделі» забезпечує оновлення та покращення якості прогностичної моделі.

– Входи: «CSV-дані для донавчання», «Запит на донавчання» (від Лаборанта).
– Виходи: «Звіт: Якість Донавченої Моделі», «Оновлена Модель та Артефакти» (які потім стають входом/управлінням для A1).
– Управління: «Алгоритми HPO (Optuna)», «Критерії якості моделі».

– Механізми: Спільні для системи.

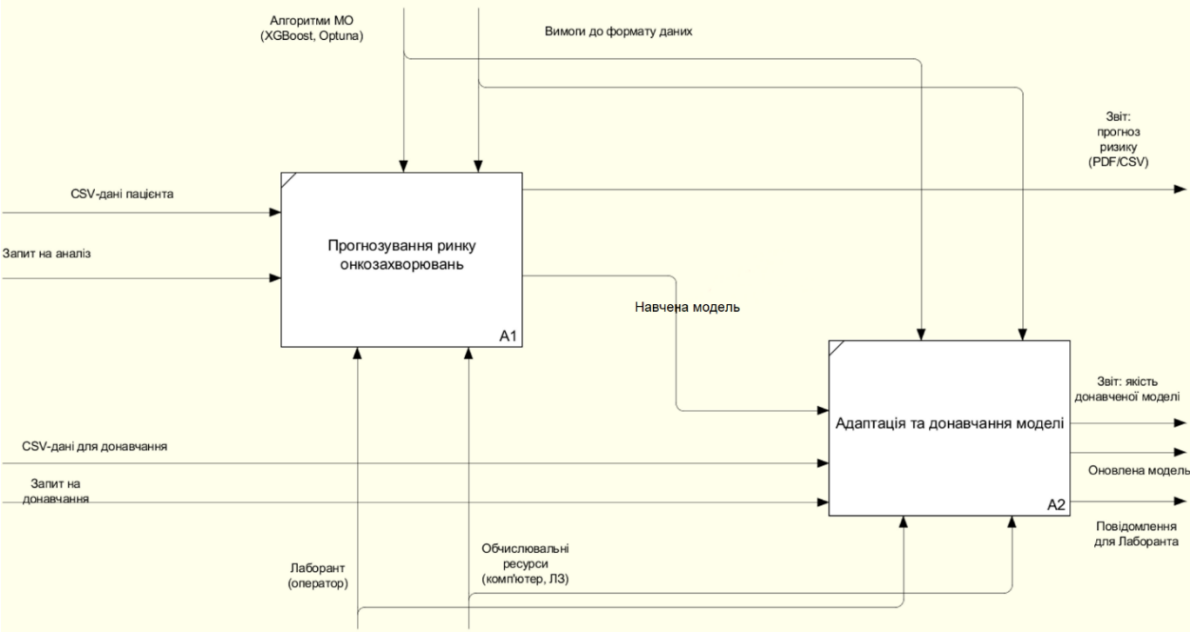

Рисунок 2.3 – Діаграма декомпозиції першого рівня функціональної моделі системи «OncoScreen Assist»

Для подальшої деталізації поведінки системи та візуалізації логіки взаємодії користувача з її основними функціями було розроблено діаграму діяльності. Ця діаграма наочно демонструє послідовність кроків, які виконуються в рамках ключових сценаріїв роботи системи.

Як показано на діаграмі (рисунок 2.4), робочий процес системи має циклічний характер. Після запуску додатку система переходить у стан очікування дій користувача. Користувач може ініціювати один з двох основних сценаріїв:

– Сценарій прогнозування, що включає завантаження даних пацієнта, отримання прогнозу та опціональне збереження звіту у форматі PDF.
– Сценарій донавчання, який передбачає завантаження даних для донавчання, запуск процесу перенавчання моделі та аналіз звіту про якість оновленої моделі.

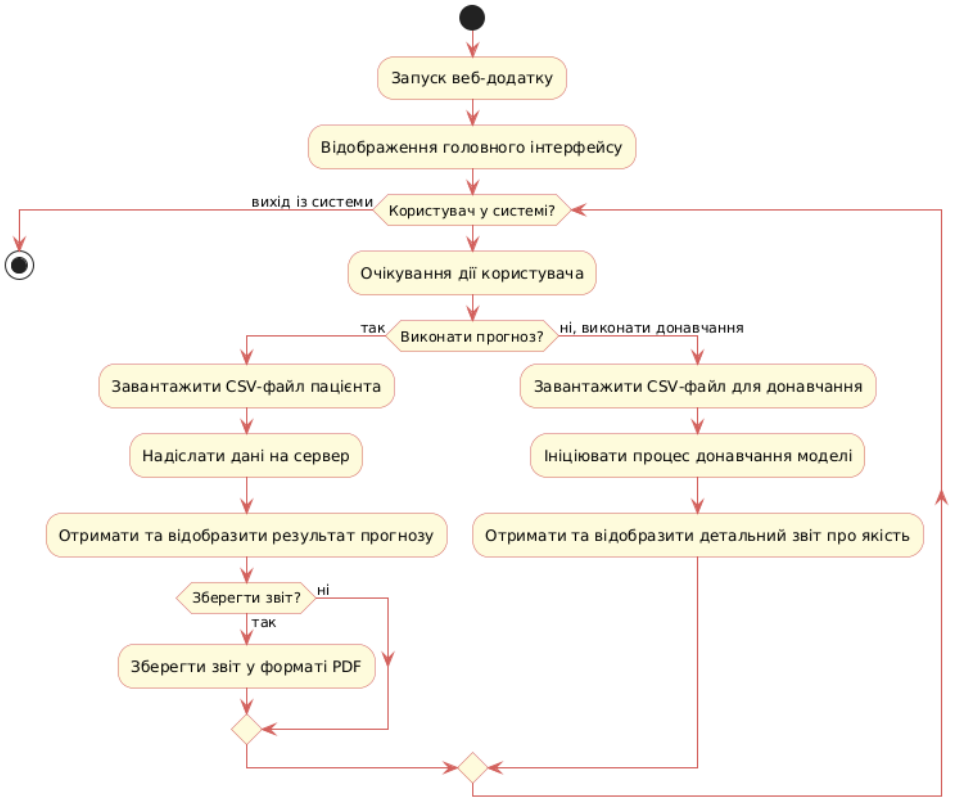

Рисунок 2.4 – Діаграма діяльності для основних сценаріїв використання системи

Дана функціональна модель дозволяє чітко структурувати процеси, що відбуваються в системі, та визначити інформаційні потоки, що є необхідною основою для подальшого проектування та реалізації інформаційної системи «OncoScreen Assist».

РОЗДІЛ 3
ОПИС ПРИЙНЯТИХ ПРОЄКТНИХ ТА ТЕХНОЛОГІЧНИХ РІШЕНЬ

3.1 Розробка об’єктної моделі

При розробці інформаційної системи «OncoScreen Assist» основний акцент було зроблено на реалізації функціональних модулів та взаємодії компонентів, а не на побудові складної ієрархічної об'єктної моделі. Тим не менш, для чіткого розуміння структури даних, з якими оперує система, можна виділити наступні ключові концептуальні об'єкти (сутності даних):

1. Дані Пацієнта (Patient Data): представляють набір кількісних цитологічних ознак для конкретного пацієнта, що слугують входом для модуля прогнозування. Структурно це табличні дані (понад 30 ознак), які під час обробки в системі завантажуються з CSV-файлу та представляються у вигляді об'єкта DataFrame бібліотеки Pandas.
2. Прогностична Модель (Prediction Model): центральний компонент, що генерує прогноз ризику. Це класифікатор XGBoost, ключовими атрибутами якого є навчені параметри та оптимальні гіперпараметри. В системі представлена об'єктом моделі XGBoost, серіалізованим у файл model.bin.
3. Масштабувальник Даних (Data Scaler): відповідає за стандартизацію вхідних числових ознак. Це об'єкт StandardScaler з бібліотеки Scikit-learn, навчені параметри якого (середні значення та стандартні відхилення) зберігаються у файлі scaler.joblib.
4. Список Ключових Ознак (Top Features List): містить перелік назв найбільш інформативних цитологічних ознак, відібраних на основі їх важливості. В системі це список рядків, що зберігається у JSON-файлі (features.json).

5. Навчальні Дані / Майстер-набір Даних (Training Data / Master Training Data): акумульований набір даних (ознаки та підтверджені діагнози), що використовується для навчання та донавчання моделі. Представлений у вигляді CSV-файлу (master_training_data.csv), який оновлюється користувачем.
6. Результат Прогнозу (Prediction Output): вихідна інформація для користувача, що включає категорію ризику та ймовірність прогнозу. В системі формується як текстовий рядок та JSON-об'єкт, з можливістю збереження у звіт.
7. Звіт про Якість Моделі (Model Evaluation Report) [20]: інформація, що генерується після донавчання моделі. Включає метрики якості (Accuracy, AUC, Матриця помилок, Звіт про класифікацію) та дані про важливість ознак, що відображаються в інтерфейсі.

3.2 Розробка архітектури

Інформаційна система «OncoScreen Assist» спроектована та реалізована з використанням сучасної клієнт-серверної архітектури [6]. Такий підхід забезпечує чітке розділення логіки представлення (клієнтська частина) та логіки обробки даних і машинного навчання (серверна частина), що сприяє гнучкості розробки, масштабованості та можливості крос-платформенного доступу до системи через стандартний веб-браузер. Загальна архітектурна схема системи представлена на рисунку 3.1.

Основними компонентами архітектури є клієнтська частина (Frontend), серверна частина (Backend) та сховище даних і артефактів моделі.

Клієнтська частина (Frontend) реалізована як веб-інтерфейс, що функціонує у веб-браузері користувача (Лаборанта). Для її розробки використано стандартні веб-технології: HTML для структурування контенту, CSS для стилізації та JavaScript для забезпечення інтерактивності. Основним призначенням клієнтської частини є відображення графічного інтерфейсу, надання елементів

управління для взаємодії з системою (завантаження CSV-файлів, ініціалізація процесів), відправка HTTP-запитів на серверну частину та відображення отриманих результатів прогнозування і звітів про якість моделі.

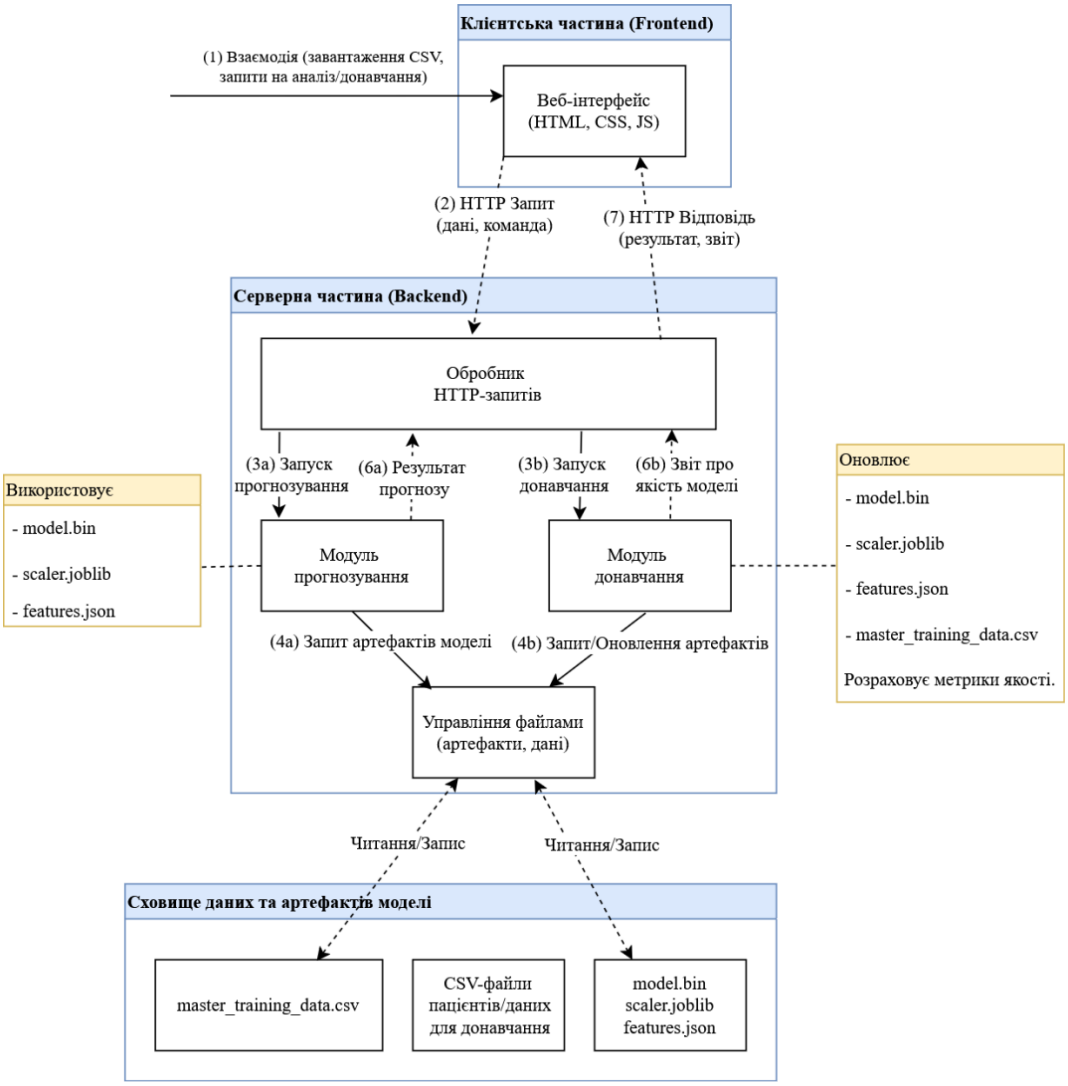

Рисунок 3.1 – Архітектурна схема інформаційної системи «OncoScreen Assist»

Серверна частина (Backend) розроблена на мові програмування Python з використанням мікрофреймворку Flask та відповідає за всю основну логіку роботи системи. Вона включає наступні ключові компоненти:

– Обробник HTTP-запитів: Приймає HTTP-запити від клієнтської частини, аналізує їх та передає управління відповідним модулям обробки, а також формує та відправляє HTTP-відповіді.

- Модуль прогнозування: Активується при запиті на аналіз даних пацієнта. Він завантажує актуальні артефакти моделі (XGBoost модель, масштабувальник даних, список ключових ознак), здійснює попередню обробку вхідних даних та генерує прогноз ризику.
- Модуль донавчання (з HPO - Optuna): Відповідає за повний цикл адаптації та перенавчання прогностичної моделі. Цей модуль об'єднує нові верифіковані дані з існуючим майстер-набором, автоматично перенавчає масштабувальник, переоцінює важливість діагностичних ознак, оптимізує гіперпараметри моделі XGBoost за допомогою Optuna та перенавчає основну модель, розраховуючи детальні метрики її якості.
- Управління файлами (артефакти, дані): Забезпечує операції читання та запису усіх необхідних файлів, включаючи артефакти моделі, майстер-файл навчальних даних та тимчасові файли, що завантажуються користувачем.

Сховище даних та артефактів моделі реалізовано на базі локальної файлової системи сервера. Воно зберігає серіалізовану навчену модель (model.bin), масштабувальник даних (scaler.joblib), список ключових ознак (features.json), акумульований майстер-набір навчальних даних (master_training_data.csv) та тимчасові CSV-файли.

Взаємодія компонентів в системі відбувається за чітко визначеними потоками, як показано на рисунку 3.1. У сценарії прогнозування, Лаборант через клієнтську частину (потік 1) відправляє запит з даними на сервер (потік 2). Серверна частина, задіявши модуль прогнозування та модуль управління файлами (потоки 3a, 4a), обробляє дані, генерує прогноз (потік 6a) та повертає результат клієнту (потік 7). У сценарії донавчання, Лаборант завантажує дані для оновлення (потік 1), які передаються на сервер (потік 2). Модуль донавчання на серверній частині обробляє ці дані, взаємодіє з файловим сховищем для оновлення майстер-набору та артефактів моделі (потоки 3b, 4b), виконує перенавчання та повертає звіт про якість нової моделі клієнту (потоки 6b, 7).

Для забезпечення переносимості, легкості розгортання та ізоляції середовища, система «OncoScreen Assist» була контейнеризована з використанням

технології Docker. Це дозволяє запускати додаток на будь-якій платформі, що підтримує Docker, без необхідності складного налаштування залежностей.

Обрана архітектура забезпечує необхідну гнучкість, модульність та можливість подальшого розвитку системи.

3.3 Проєктування інтерфейсу програмної системи

Проектування користувацького інтерфейсу (UI) інформаційної системи «OncoScreen Assist» здійснювалося з урахуванням наступних ключових принципів: простота, інтуїтивна зрозумілість та орієнтація на користувача без спеціалізованих технічних навичок (медичного працівника, лаборанта). Основною метою було створення зручного інструменту для швидкого отримання прогностичної інформації та ефективного управління процесом донавчання моделі. Інтерфейс реалізований у вигляді єдиної веб-сторінки з чітким функціональним зонуванням, що забезпечує легкість навігації та мінімізує час на освоєння системи. Для розробки використано стандартні веб-технології HTML, CSS та JavaScript.

Для візуалізації логічної структури та взаємозв'язку між компонентами інтерфейсу було розроблено відповідну схему (рисунок 3.2).

Рисунок 3.2 – Структурна схема веб-додатку «OncoScreen Assist»

Інтерфейс системи логічно розділений на дві основні функціональні зони: зону прогнозування ризику та зону донавчання моделі.

1. Інтерфейс прогнозування ризику:

Головний екран системи надає користувачеві можливість виконати основну функцію – отримати прогноз ризику онкологічного захворювання для пацієнта (рисунок 3.3).

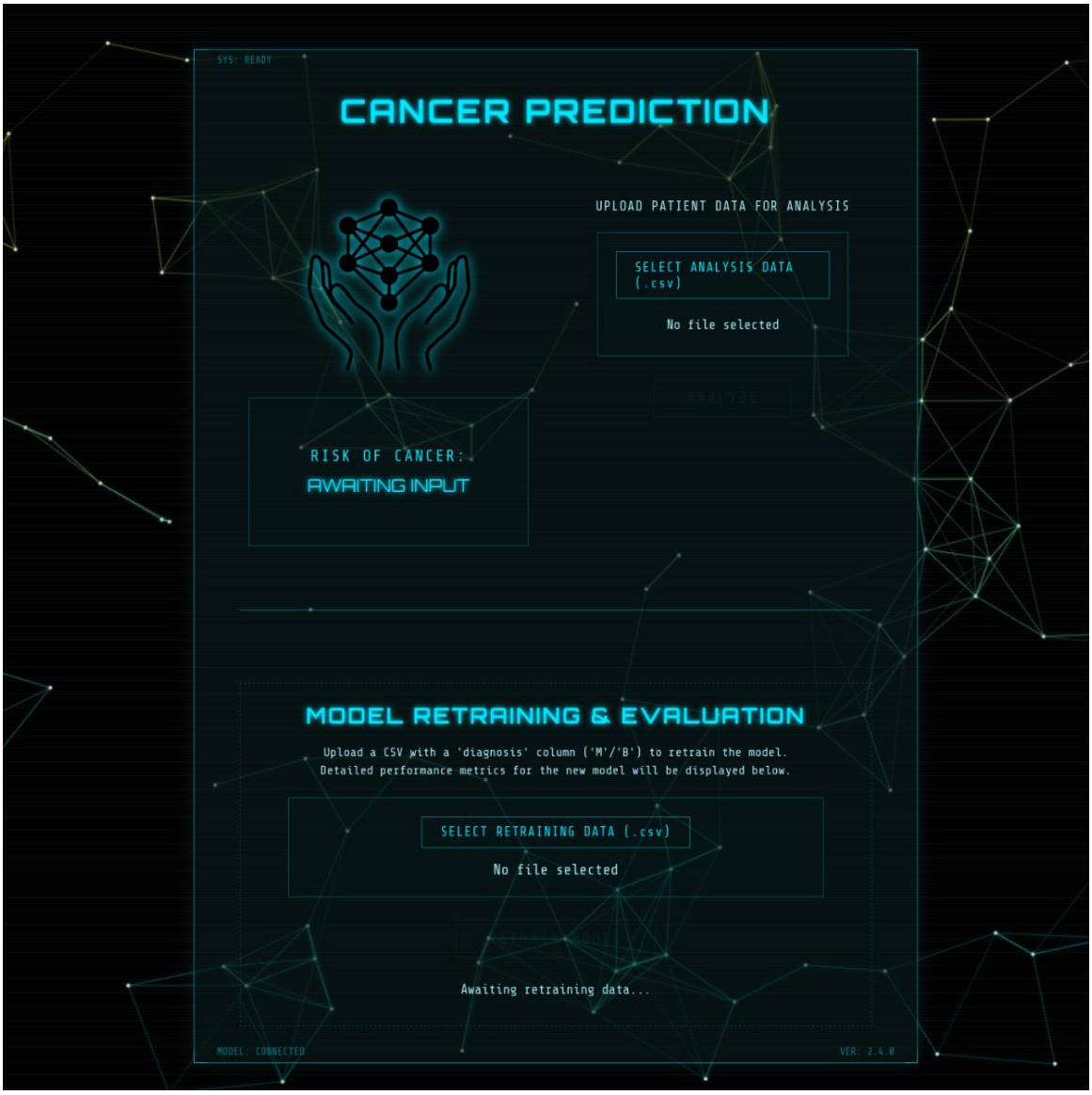

Рисунок 3.3 – Головний екран системи «OncoScreen Assist»: секція прогнозування ризику

У верхній частині інтерфейсу розташована секція «CANCER PREDICTION». Користувачеві пропонується обрати CSV-файл з даними пацієнта за допомогою кнопки «SELECT ANALYSIS DATA (.csv)». Після вибору

файлу його назва відображається поруч із кнопкою. Для запуску аналізу передбачена кнопка «ANALYZE». Початково, до виконання аналізу, у полі «RISK OF CANCER» відображається статус «AWAITING INPUT» (Очікування вводу).

Після натискання кнопки «ANALYZE» та успішної обробки даних сервером, система відображає результат прогнозування безпосередньо в полі «RISK OF CANCER» (рисунок 3.4).

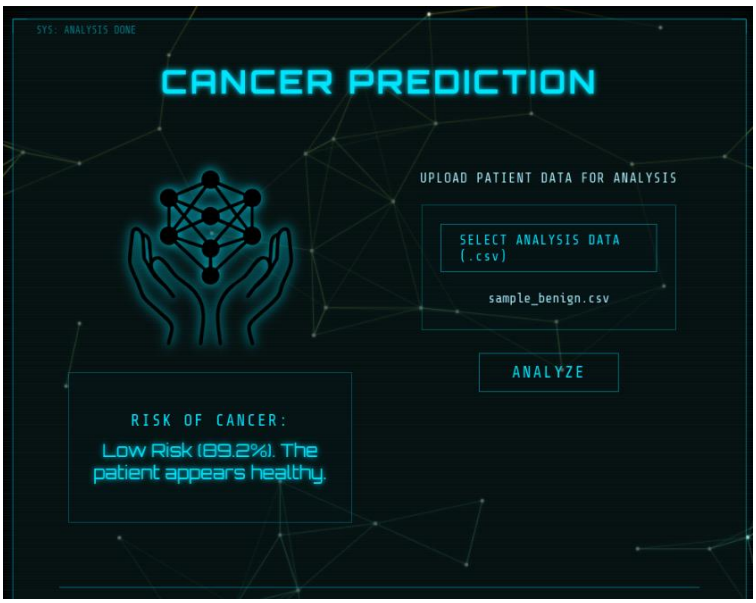

Рисунок 3.4 – Відображення результату прогнозування ризику в системі «OncoScreen Assist»

Результат включає категоріальний висновок (наприклад, «High Risk» або «Low Risk»), відсоток впевненості моделі у цьому прогнозі, а також коротку текстову рекомендацію (наприклад, «Further diagnosis is recommended» або «The patient appears healthy»). Такий формат представлення результатів є лаконічним та інформативним для швидкої оцінки ситуації Лаборантом.

2. Інтерфейс донавчання моделі та оцінки її якості:

Нижня частина головного екрану присвячена функціоналу донавчання та оцінки моделі – «MODEL RETRAINING & EVALUATION». Ця секція дозволяє користувачеві ініціювати процес оновлення прогностичної моделі на

основі нових верифікованих даних. Користувач обирає CSV-файл з даними для донавчання за допомогою кнопки «SELECT RETRAINING DATA (.csv)» та запускає процес натисканням кнопки «RETRAIN MODEL».

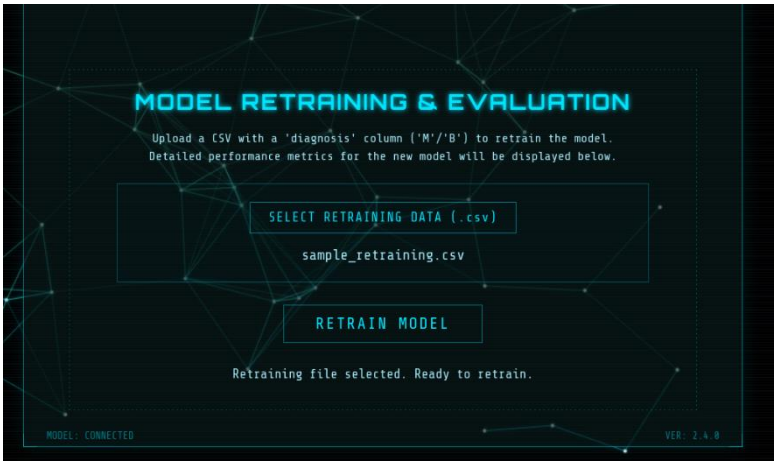

Рисунок 3.5 – Інтерфейс ініціалізації процесу донавчання моделі

Система інформує користувача про хід процесу донавчання (наприклад, повідомленням «Retraining in progress...» або «Model successfully retrained with Hyperparameter Optimization»). Після успішного завершення повного циклу донавчання, включаючи оптимізацію гіперпараметрів, система відображає детальний звіт про якість оновленої моделі (рисунок 3.6).

Даний звіт включає наступні ключові блоки інформації:

– «Overall Performance»: Загальні метрики якості, такі як Точність (Accuracy) та AUC ROC Score;

– «Confusion Matrix»: Матриця помилок, що наочно демонструє кількість істинно позитивних, хибно позитивних, істинно негативних та хибно негативних прогнозів;

– «Top Feature Importances»: Візуалізація відносної важливості топ-ознак, які використовуються оновленою моделлю;

– «Classification Report»: Детальний звіт про класифікацію, що містить показники Precision, Recall, F1-Score та Support для кожного класу («Benign» та «Malignant»).

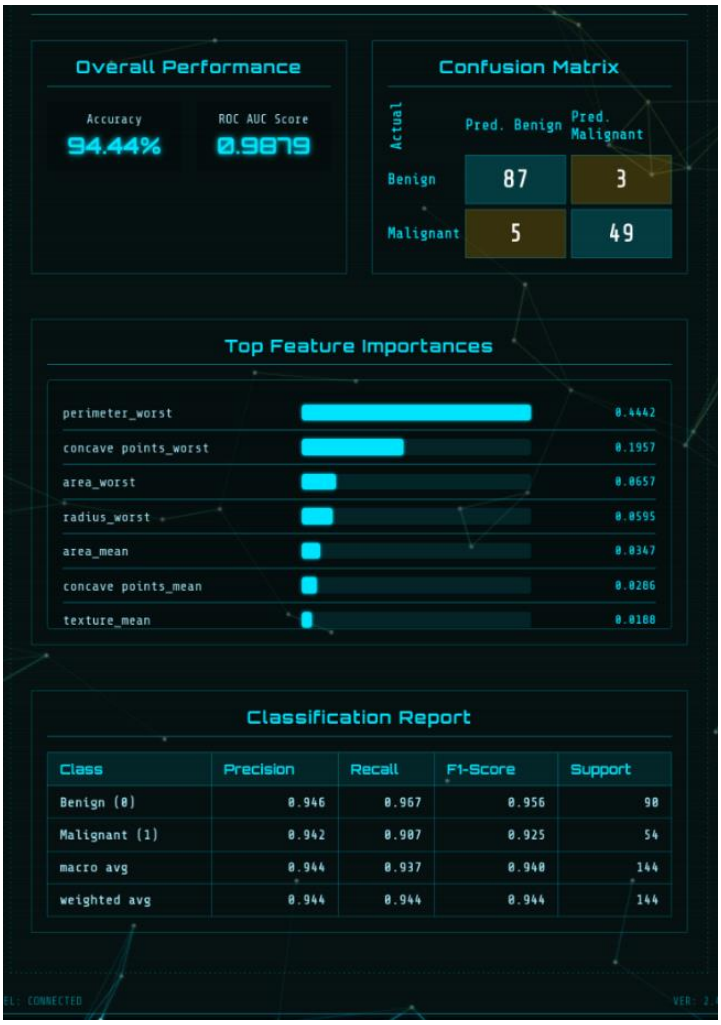

Рисунок 3.6 – Інтерфейс відображення результатів донавчання та оцінки якості моделі в системі «OncoScreen Assist»

Таке комплексне представлення результатів оцінки дозволяє користувачеві об'єктивно оцінити якість донавченої моделі та прийняти обґрунтоване рішення щодо її подальшого використання в практичній роботі.

Обраний мінімалістичний дизайн та чітке функціональне зонування спрямовані на забезпечення максимальної простоти використання системи та концентрації користувача на виконанні основних завдань: отриманні прогнозу та, за необхідності, донавчанні моделі. Розділення інтерфейсу на логічні блоки прогнозування та донавчання дозволяє уникнути плутанини та робить процес взаємодії з системою послідовним та логічним. Детальне відображення метрик якості після донавчання підвищує прозорість роботи системи та довіру до її результатів.

3.4 Обґрунтування вибору мови програмування та технологій

Вибір технологічного стеку для розробки інформаційної системи «OncoScreen Assist» ґрунтувався на низці ключових критеріїв, серед яких ефективність для вирішення задач машинного навчання, швидкість та гнучкість розробки, доступність потужних бібліотек, можливість створення крос-платформенного веб-інтерфейсу, легкість розгортання та, що особливо важливо для системи медичного призначення, зрілість та надійність обраних технологій.

Мова програмування Python була обрана як основна для реалізації серверної частини (бекенду) та всієї логіки машинного навчання [2, 25]. Python надає величезну екосистему бібліотек, спеціально призначених для наукових обчислень, аналізу даних та машинного навчання, таких як Pandas, NumPy, Scikit-learn, XGBoost та Optuna, що значно прискорює процес розробки та дозволяє використовувати найсучасніші алгоритми [21, 23]. Простота синтаксису Python та велика активна спільнота також сприяли швидкому прототипуванню та реалізації складних функціональних модулів.

Для створення веб-інтерфейсу та серверної частини спочатку розглядалися різні підходи. Ідея розробки десктопного додатку (наприклад, з використанням фреймворку PyQt) була відхилена через потенційні проблеми із забезпеченням крос-платформенності та складністю підтримки на різноманітних операційних системах, що часто зустрічаються у медичних закладах. Натомість було прийнято рішення про реалізацію системи у вигляді веб-додатку, що забезпечує універсальний доступ через будь-який сучасний веб-браузер.

В якості веб-фреймворку для реалізації серверної частини (бекенду) було обрано Flask [18]. Хоча розглядалася альтернатива у вигляді більш комплексного фреймворку Django, який надає широкий набір вбудованих інструментів для розробки великих веб-додатків, для цілей даного проекту перевагу було віддано Flask. Це рішення зумовлене його легковаговістю, мінімалістич-

ністю та гнучкістю. Flask не нав'язує жорсткої структури та дозволяє розробнику підключати лише необхідні компоненти, що робить його ідеальним для створення спеціалізованих сервісів та API. На відміну від новітніх, менш перевірених фреймворків (наприклад, Gradio, який також розглядався), Flask є зрілою та стабільною технологією з добре налагодженими процесами розробки та великою кількістю документації, що є критично важливим для систем медичного призначення. Висока продуктивність Flask при обробці запитів також стала важливим фактором.

Для побудови прогностичної моделі було обрано алгоритм XGBoost. На етапі попередніх досліджень розглядалася можливість використання інших ансамблевих методів, зокрема, Випадкого лісу (Random Forest), відомого своєю простотою. Однак, експерименти показали, що XGBoost демонструє вищу точність та кращу узагальнюючу здатність на цільових даних, що є типовим для цього алгоритму в задачах класифікації на табличних даних. Можливість XGBoost оцінювати важливість ознак також стала важливим фактором для реалізації механізму їх автоматичного відбору.

Для автоматичної оптимізації гіперпараметрів моделі XGBoost було обрано фреймворк Optuna. Альтернативою міг слугувати GridSearchCV з бібліотеки Scikit-learn, проте Optuna, що використовує більш просунуті байєсівські методи оптимізації (зокрема, алгоритм TPE – Tree-structured Parzen Estimator), показав значно вищу ефективність та швидкість знаходження оптимальних комбінацій гіперпараметрів на заданому наборі даних, що скоротило час, необхідний для донавчання моделі.

Клієнтська частина (фронтенд) системи реалізована з використанням стандартних та перевірених часом веб-технологій: HTML, CSS та JavaScript. Такий вибір забезпечує максимальну крос-браузерність, крос-платформеність та не створює додаткових залежностей для кінцевого користувача, якому для роботи з системою потрібен лише веб-браузер.

Для забезпечення легкості розгортання, переносимості та ізоляції середовища було використано технологію контейнеризації Docker [22]. Це дозволяє запакувати додаток з усіма його залежностями в єдиний контейнер, який може бути запущений на будь-якій системі, що підтримує Docker, значно спрощуючи процес інсталяції та підтримки системи в медичних установах.

Таким чином, обраний технологічний стек є збалансованим поєднанням потужних інструментів для машинного навчання, гнучких засобів веб-розробки та надійних технологій для розгортання, що дозволило створити функціональну, адаптивну та зручну у використанні інформаційну систему «OncoScreen Assist».

3.5 Опис програмної реалізації

Програмна реалізація інформаційної системи «OncoScreen Assist» виконана з використанням мови програмування Python та мікрофреймворку Flask для серверної частини, а також стандартних веб-технологій (HTML, CSS, JavaScript) для клієнтської частини. Основна логіка, пов'язана з обробкою даних, машинним навчанням, прогнозуванням та донавчанням моделі, зосереджена на сервері.

1. Загальна структура серверного додатку (Backend)

– Серверний додаток, побудований на Flask, є ядром системи. Він відповідає за прийом запитів від клієнта, обробку даних, взаємодію з моделлю машинного навчання та повернення результатів.

– Ініціалізація: При старті додатку відбувається завантаження в пам'ять основних артефактів моделі: навченої моделі XGBoost (model.bin), об'єкта масштабувальника даних (scaler.joblib) та списку ключових діагностичних ознак (features.json).

Ключові ендпоінти:

– / (Головний): Відображення основного веб-інтерфейсу.

– /analyze (метод POST): Отримання прогнозу ризику захворювання.

– /retrain (метод POST): Ініціалізація процесу донавчання прогностичної моделі.

2. Реалізація модуля прогнозування ризику (ендпоінт /analyze)

– даний модуль відповідає за надання прогнозу на основі даних пацієнта. Алгоритм роботи включає наступні кроки:

– прийом та валідація вхідних даних: Ендпоінт приймає CSV-файл, перевіряє його наявність, коректність формату, повноту набору з 30 ознак та можливість конвертації значень у числовий формат;

– попередня обробка даних: Завантажені дані масштабуються за допомогою об'єкта StandardScaler, після чого з них відбираються ключові діагностичні ознаки;

– отримання прогнозу: Підготовлені дані передаються моделі XGBoost для генерації прогнозу класу та розрахунку ймовірностей;

– формування відповіді: Результат інтерпретується у текстовий опис («Низький ризик» / «Високий ризик») та разом з відсотком впевненості формується у JSON-відповідь для клієнта.

3. Реалізація модуля донавчання моделі з оптимізацією гіперпараметрів (ендпоінт /retrain)

– процес донавчання є комплексним та забезпечує адаптивність системи;

– валідація та оновлення даних: Після валідації вхідного CSV-файлу для донавчання (включаючи перевірку колонки diagnosis), нові дані об'єднуються з майстер-набором master_training_data.csv. Перевіряється наявність достатньої кількості записів (мінімум 50).

Повний цикл перенавчання артефактів:

– масштабувальник: Новий StandardScaler навчається на всіх 30 ознаках з повного оновленого майстер-набору та зберігається;

– відбір ознак: На всіх 30 масштабованих ознаках навчається тимчасова модель XGBoost для визначення та збереження оновленого списку топ-N найбільш важливих ознак (features.json);

– оптимізація гіперпараметрів (HPO): Фреймворк Optuna (з алгоритмом TPE) використовується для підбору оптимальних гіперпараметрів для фінальної моделі XGBoost. Оптимізація проводиться на відібраних топ-N масштабованих ознаках з майстер-набору, цільовою функцією є максимізація середнього ROC AUC при 5-фолдовій стратифікованій крос-валідації. Оптимізуються такі гіперпараметри, як n_estimators, learning_rate, max_depth та інші;

– навчання фінальної моделі: Модель XGBoost навчається на топ-N ознаках (з майстер-набору, розділеного на тренувальну та тестову вибірки) з використанням знайдених оптимальних гіперпараметрів та зберігається;

– оцінка якості та відповідь: Якість донавченої моделі оцінюється на тестовій вибірці (Accuracy, AUC, Матриця помилок, Звіт про класифікацію). Клієнту повертається JSON з повідомленням про успіх та розрахованими метриками;

– перезавантаження артефактів: Оновлені модель, скейлер та список ознак перезавантажуються в пам'ять сервера.

4. Реалізація клієнтської частини (Frontend)

Клієнтська частина, реалізована за допомогою HTML, CSS та JavaScript, забезпечує користувацький інтерфейс. JavaScript обробляє дії користувача, формує та відправляє асинхронні HTTP-запити (Fetch API) на бекенд, а також обробляє отримані JSON-відповіді для динамічного відображення результатів та звітів.

5. Контейнеризація за допомогою Docker

Для забезпечення високого рівня переносимості, легкості розгортання та ізоляції робочого середовища, вся система «OncoScreen Assist» була контейнеризована з використанням технології Docker. Було розроблено Dockerfile, який визначає образ додатку на основі офіційного образу python:3.9-slim. У процесі побудови образу виконується копіювання коду додатку та встановлення всіх необхідних залежностей з файлу requirements.txt. Робочою директорією всередині контейнера визначено /app, а сам додаток експонує порт 80. Запуск Flask-сервера всередині контейнера здійснюється командою flask run.

Додатково, для зручності розробки та управління розгортанням, створено файл docker-compose.yml, що описує сервіс web, який будує образ, прокидає порти та монтує директорію з кодом для «живого» оновлення. Такий підхід значно спрощує процес розгортання системи та забезпечує консистентність робочого середовища.

РОЗДІЛ 4
ДОСЛІДНА ЕКСПЛУАТАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Інструкція користувача ІС

Інформаційна система «OncoScreen Assist» розроблена з акцентом на простоту та інтуїтивність використання медичним персоналом без спеціальної технічної підготовки. Дана інструкція описує основні кроки для роботи з системою. Для роботи з системою користувачеві потрібен лише сучасний веб-браузер та доступ до CSV-файлів з даними.

Важливо зазначити, що система «OncoScreen Assist» постачається з попередньо навченою базовою моделлю. Ця модель пройшла ретельне тестування на валідаційному наборі даних та продемонструвала наступні показники ефективності. Ці метрики підтверджують високу надійність прогнозів "із коробки". Функціонал донавчання дозволяє користувачеві адаптувати та потенційно покращити ці показники на специфічних локальних даних.

Система працює у двох основних режимах: прогнозування ризику та донавчання моделі.

1. Робота в режимі прогнозування ризику.

– Крок 1: Початковий екран системи.

При запуску системи користувач бачить головний інтерфейс, готовий до роботи. У верхній частині екрана розташована секція «CANCER PREDICTION» для виконання прогнозу.

– Крок 2: Вибір файлу з даними пацієнта.

Для аналізу даних пацієнта користувач натискає кнопку «SELECT ANALYSIS DATA (.csv)». Це відкриває стандартне діалогове вікно операційної системи, де необхідно обрати підготовлений CSV-файл. Після вибору назва файлу з'явиться поруч із кнопкою, що свідчить про готовність до аналізу.

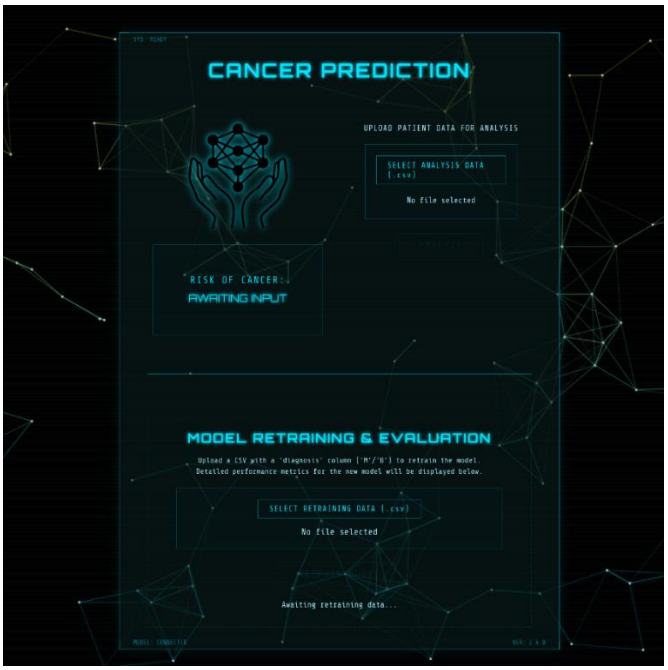

Рисунок 4.1 – Початковий вигляд інтерфейсу системи «OncoScreen Assist»

– Крок 3: Запуск аналізу та отримання результату.

Після вибору файлу користувач натискає кнопку «ANALYZE». Система надсилає дані на сервер для обробки. Після завершення аналізу (зазвичай кілька секунд) у полі «RISK OF CANCER» відображається результат прогнозу.

Результат містить категоріальний висновок («Low Risk» або «High Risk»), відсоток впевненості моделі та коротку текстову рекомендацію. Наприклад, для пацієнта з низьким ризиком система може показати «Low Risk. The patient appears healthy».

2. Робота в режимі донавчання моделі

Цей режим дозволяє оновити прогностичну модель на основі нових, верифікованих даних, підвищуючи її точність та адаптуючи до специфіки локальної вибірки пацієнтів.

– Крок 1: Вибір файлу з даними для донавчання.

У нижній секції інтерфейсу «MODEL RETRAINING & EVALUATION» користувач натискає кнопку «SELECT RETRAINING DATA (.csv)» та обирає підготовлений CSV-файл з новими даними, що містять підтверджені діагнози.

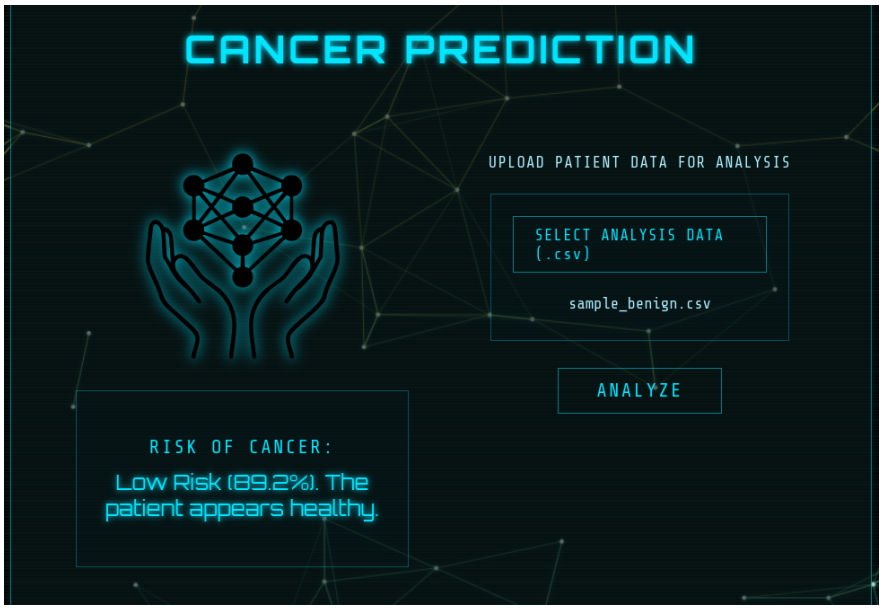

Рисунок 4.2 – Приклад результату прогнозування з результатом «Low Risk»

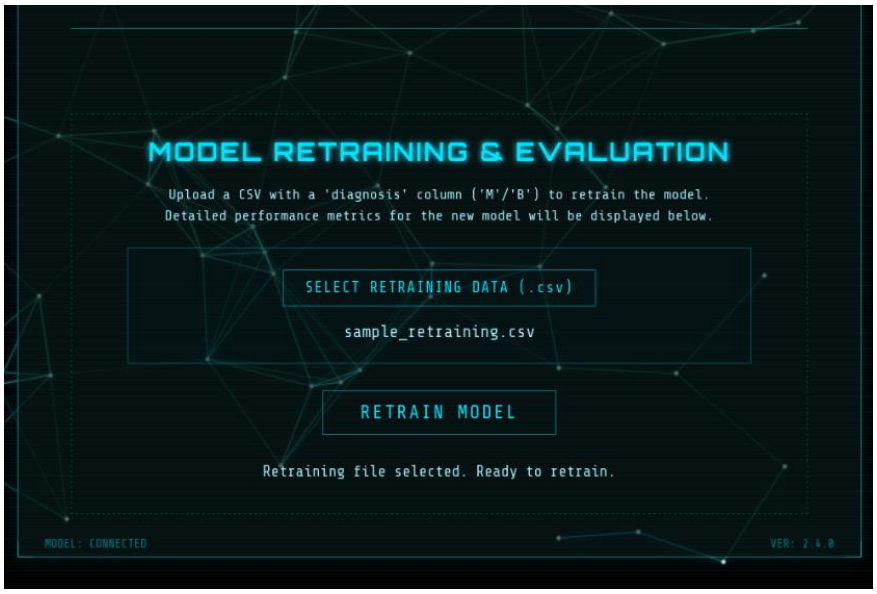

Рисунок 4.3 – Вибір файлу для донавчання моделі

– Крок 2: Запуск процесу донавчання.

Після вибору файлу користувач натискає кнопку «RETRAIN MODEL». Система ініціює повний цикл перенавчання моделі, що включає об'єднання даних, перенавчання масштабувальника ознак, автоматичну оптимізацію гіперпараметрів (HPO) та навчання нової версії моделі. Під час цього процесу система інформує користувача про його перебіг повідомленням «Retraining in progress...».

– Крок 3: Аналіз результатів оновленої моделі.

Після успішного завершення процесу донавчання система відображає детальний звіт про якість нової моделі. Цей звіт дозволяє користувачу об'єктивно оцінити характеристики оновленої моделі перед її подальшим використанням. Звіт містить загальну продуктивність (Accuracy, ROC AUC Score), матрицю помилок (Confusion Matrix), візуалізацію важливості ознак (Top Feature Importances) та детальний звіт про класифікацію (Classification Report).

Ключовою перевагою розробленої системи є надання користувачеві (медичному працівнику) повного та прозорого звіту про якість новоствореної моделі (рисунок 4.4). Це дозволяє не «сліпо» довіряти системі, а приймати обґрунтоване рішення щодо її подальшого використання. Детальний звіт надає відповіді на критично важливі питання:

– Загальна ефективність (Overall Performance): Наскільки нова модель є точною в цілому? Чи покращився показник ROC AUC порівняно з попередньою версією?
– Аналіз помилок (Confusion Matrix): Де саме модель робить помилки? Найважливіше — чи не збільшилась кількість хибно негативних прогнозів (FN), тобто випадків, коли злоякісний процес був пропущений?
– Детальна класифікація (Classification Report): Наскільки можна довіряти кожному конкретному вердикту моделі (наприклад, показник Recall для класу «Злоякісна» показує, який відсоток реальних захворювань модель змогла виявити).

Таким чином, користувач отримує всі необхідні інструменти для самостійної валідації результатів донавчання, що кардинально відрізняє дану систему від пропрієтарних аналогів типу «чорна скринька».

Для ще більшої зручності та наочності, користувачеві пропонується наступна таблиця-пам'ятка (таблиця 4.1) для швидкої інтерпретації ключового показника якості — ROC AUC.

Таблиця 4.1 – Інтерпретація значень метрики ROC AUC

Значення ROC AUC	Інтерпретація якості моделі
1	2
0.90 – 1.00	Відмінна. Модель має виняткову здатність до розрізнення класів.
0.80 – 0.89	Добра. Модель є надійною для практичного застосування.
0.70 – 0.79	Задовільна. Модель можна використовувати, але з обережністю.
0.60 – 0.69	Слабка. Модель потребує покращення, не рекомендується для клінічного використання.
0.50 – 0.59	Незадовільна. Продуктивність моделі не краща за випадкове вгадування.

4.2 Тестування та оцінка ефективності прогностичної моделі

Після розробки та реалізації інформаційної системи «OncoScreen Assist» було проведено її комплексне тестування з метою перевірки працездатності всіх функціональних модулів та детальної оцінки якості розробленої прогностичної моделі. Тестування проводилося в середовищі, що емулює робоче місце користувача, з використанням стандартного веб-браузера для доступу до системи. Оцінка якості моделі здійснювалася на відкладеній тестовій вибірці, сформованій з оновленого майстер-набору даних (master_training_data.csv) після завершення процесу донавчання та автоматичної оптимізації гіперпараметрів за допомогою фреймворку Optuna.

Рисунок 4.4 – Детальний звіт про якість донавченої моделі

Функціональне тестування системи включало перевірку наступних ключових аспектів:

– коректність завантаження CSV-файлів з даними пацієнтів для прогнозування та з даними для донавчання моделі;

– правильність отримання прогнозу ризику (категорія та відсоток впевненості) для різних наборів вхідних цитологічних ознак;

– успішне виконання повного циклу донавчання моделі, включаючи накопичення даних, перенавчання масштабувальника, переоцінку важливості ознак, оптимізацію гіперпараметрів та навчання нової моделі;

– адекватність та повнота відображення результатів прогнозування та детальних метрик оцінки якості моделі в користувацькому інтерфейсі;

– обробка системою потенційних помилок, таких як завантаження файлу некоректного формату або з неповним набором даних.

За результатами функціонального тестування було встановлено, що всі основні функціональні можливості системи «OncoScreen Assist» працюють коректно та відповідно до вимог, визначених у технічному завданні.

Оцінка якості прогностичної моделі. Ключовим етапом тестування була кількісна оцінка ефективності фінальної прогностичної моделі XGBoost. На тестовій вибірці, що не використовувалася при навчанні моделі та оптимізації гіперпараметрів, були розраховані наступні показники якості:

Загальна ефективність моделі характеризується наступними метриками:

– Загальна Точність (Accuracy): 94.44%. Це означає, що модель правильно класифікувала понад 94% випадків у тестовій вибірці;
– Порівняння ROC AUC на навчальній та тестовій вибірках. Для перевірки моделі на наявність перенавчання було проведено порівняння метрики ROC AUC. На навчальній вибірці показник склав 0.9923, а на тестовій вибірці — 0.9879. Дуже близькі значення цих показників свідчать про відмінну здатність моделі до узагальнення та відсутність перенавчання. Високе значення площі під ROC-кривою підтверджує її виняткову роздільну здатність, тобто здатність ефективно відрізняти пацієнтів з високим ризиком онкологічного захворювання від пацієнтів з низьким ризиком.

Для більш детального аналізу результатів класифікації було побудовано Матрицю помилок (таблиця 4.2).

Таблиця 4.2 – Матриця помилок для оцінки якості прогностичної моделі.

	Прогноз: Доброякісна (0)	Прогноз: Злоякісна (1)
Факт: Доброякісна (0)	87	3
Факт: Злоякісна (1)	5	49

Аналіз Матриці помилок показує:

– Істинно негативні результати (87): Модель правильно ідентифікувала 87 доброякісних випадків.

– Істинно позитивні результати (49): Модель правильно визначила 49 злоякісних випадків.
– Хибно позитивні результати (3): у 3 випадках модель помилково класифікувала доброякісний процес як злоякісний. Це може призвести до непотрібного занепокоєння пацієнта та додаткових обстежень, однак не несе ризику пропустити захворювання.
– Хибно негативні результати (5): у 5 випадках модель помилково визначила злоякісний процес як доброякісний. Це найбільш критичний тип помилки в медичній діагностиці, оскільки може призвести до затримки лікування. Показник повноти (Recall) для класу "Злоякісна", що становить 90.7% (таблиця 4.3), якраз і відображає здатність моделі мінімізувати кількість таких пропущених випадків.

Детальні показники якості для кожного класу представлені у Звіті про класифікацію (таблиця 4.3).

Таблиця 4.3 – Звіт про класифікацію для оцінки якості прогностичної моделі.

Клас	Precision	Recall	F1-Score	Support
Доброякісна (0)	0.946	0.967	0.956	90
Злоякісна (1)	0.942	0.907	0.925	54
Macro Avg	0.944	0.937	0.940	144
Weighted Avg	0.944	0.944	0.944	144

Зі звіту про класифікацію видно, що для класу «Злоякісна (1)» (пацієнти з високим ризиком) досягнуто наступних показників:

– Precision (Точність прогнозу) = 0.942: З усіх пацієнтів, яких система віднесла до групи високого ризику, 94.2% дійсно мали злоякісний процес.

– Recall (Повнота / Чутливість) = 0.907: Система змогла правильно ідентифікувати 90.7% усіх пацієнтів, які реально мали злоякісний процес. Цей

показник є особливо важливим, оскільки він відображає здатність моделі мінімізувати кількість пропущених випадків захворювання (хибно негативних результатів).

– F1-Score = 0.925: Збалансована метрика, що враховує як Precision, так і Recall, також демонструє високу якість класифікації для цього класу.

Аналіз важливості ознак, проведений після донавчання моделі, показав, що найбільший внесок у прийняття рішення моделлю роблять наступні цитологічні параметри (таблиця 4.4).

Аналіз важливості ознак (таблиця 4.4) показує, що найбільший внесок у прийняття рішення моделлю роблять параметри, що характеризують "найгірші" (тобто найбільш атипові) клітини у зразку, зокрема perimeter_worst, concave points_worst та area_worst.

Таблиця 4.4 – Топ-7 найбільш важливих діагностичних ознак.

Ознака	Важливість (Importance)
perimeter_worst	0.4442
concave points_worst	0.1957
area_worst	0.0657
radius_worst	0.0595
area_mean	0.0347
concave points_mean	0.0286
texture_mean	0.0188

Це повністю узгоджується з принципами цитологічної діагностики, де наявність навіть невеликої кількості клітин із вираженими ознаками атипії (великий нерівний периметр, значна площа ядра, наявність увігнутостей контуру) є ключовим індикатором злоякісного процесу. Таким чином, модель навчилася спиратися на найбільш клінічно значущі характеристики.

Це свідчить про те, що модель спирається на медично значущі характеристики клітин, пов'язані з їх розмірами, формою та атиповими змінами контурів, що корелює з принципами цитологічної діагностики.

Важливо зазначити, що представлені метрики якості відображають стан моделі на конкретний момент часу. Завдяки реалізованому функціоналу донавчання, при надходженні нових верифікованих даних та ініціалізації процесу перенавчання, характеристики моделі будуть оновлюватися. Система надає користувачеві інструменти для оцінки якості кожної нової версії моделі, що дозволяє відстежувати її еволюцію та підтримувати високий рівень прогностичної ефективності.

4.3 Техніко-економічні показники розробки інформаційної системи

Для об'єктивної оцінки ефективності впровадження інформаційної системи необхідно розрахувати основні техніко-економічні показники. Розрахунок проводиться за аналогією з комерційним проєктом для визначення його кошторисної вартості та потенційної економічної ефективності.

Витрати, пов'язані з розробкою продукту

До даних витрат належать витрати на оплату праці розробників з нарахуваннями, амортизаційні відрахування, витрати на електроенергію та інші супутні витрати.

Витрати на оплату праці з нарахуваннями – складаються з витрат на основну заробітну плату, додаткову заробітну плату та нарахувань згідно чинного законодавства (22%).

Витрати на основну заробітну плату – визначаються виходячи з місячного посадового окладу та витрат часу, понесених на розробку даного продукту.

Розрахунок основної оплати праці здійснюється за формулою:

$$\text{ЗПосн} = \left(\frac{\text{ЗПміс}}{\text{ФРЧ}}\right) * \text{Тр} \qquad (4.1)$$

де:

– ЗПосн – основна заробітна плата, грн;
– ЗПміс – місячний посадовий оклад розробника, на підприємстві становить 25 000 грн/міс;
– ФРЧ – місячний фонд робочого часу, становить 168 (21 день по 8 год) год./міс;
– Тр – трудомісткість (затрати часу) виготовлення програмного продукту, становить 160 год.

Таким чином:

$$ЗПосн = \left(\frac{25000}{168}\right) * 160 \approx 23\,810 \text{ грн.}$$

Додаткова заробітна плата передбачає різноманітні види доплат за рівень кваліфікації. Приймемо відсоток нарахувань додаткової заробітної плати на рівні 15%.

$$ЗПдод = ЗПосн * \left(\frac{\%дод}{100}\right) \quad (4.2)$$

де %дод – відсоток нарахувань додаткової заробітної плати, %.

$$ЗПдод = 23810 * \left(\frac{15}{100}\right) \approx 3\,572 \text{ грн.}$$

Нарахування на заробітну плату здійснюють від суми основної та додаткової зарплати. Розмір нарахувань становить 22%.

$$Нзп = (ЗПосн + ЗПдод) * \left(\frac{\%нар}{100}\right) \quad (4.3)$$

де %нар – відсоток нарахувань заробітної плати, %.

$$Нзп = (23810 + 3572) * \left(\frac{22}{100}\right) \approx 6\,024 \text{ грн.}$$

Розрахунок загальної суми витрат на оплату праці з нарахуваннями розробника наведемо у вигляді таблиці 4.5.

Таблиця 4.5 – Розрахунок витрат на оплату праці з нарахуваннями розробників

Найменування посади	Основна ЗП, грн.	Додаткова ЗП, грн.	Нарахування (ЄСВ), грн.	Всього витрат на оплату праці, грн.
Розробник	23 810	3 572	6 024	33 406

Розрахунок суми амортизаційних відрахувань обладнання, яке використовується при розробці продукту, здійснюється прямолінійним методом. При створенні продукту використовувався ноутбук та принтер. Відповідно до податкового кодексу України, ноутбук (ПК) відносять до 4 групи (термін використання 2 роки, норма амортизації 50%), а принтер – до 6 групи (термін використання 5 років, норма амортизації 20%).

Розрахунок річної суми амортизаційних відрахувань:

$$\text{Аріч} = \text{ББобл} * \left(\frac{\%\text{аморт}}{100}\right) \quad (4.4)$$

де:

– ББобл - ціна придбання, або балансова вартість обладнання, грн.

– %аморт – річний відсоток нарахувань амортизації, %.

$$\text{Аріч.ноут.} = 30000 * \left(\frac{50}{100}\right) = 15\,000 \text{ грн.}$$

$$\text{Аріч.принт.} = 6000 * \left(\frac{20}{100}\right) = 1\,200 \text{ грн.}$$

Сума амортизаційних відрахувань, що увійде до собівартості програмного продукту, розраховується з врахуванням часу використання обладнання (1 місяць).

$$\text{А} = (\text{Аріч.ноут.} + \text{Аріч.принт.}) * \left(\frac{\text{Твик(міс)}}{12}\right) \quad (4.5)$$

$$\text{А} = (15000 + 1200) * \left(\frac{1}{12}\right) = 1\,350 \text{ грн.}$$

Витрати на електроенергію – визначаються відповідно до потужності обладнання, фактичного часу роботи та ціни на електроенергію.

$$\text{Вел} = \text{П} * \text{Квп} * \text{Тр} * \text{Ц1кВт} \quad (4.6)$$

де:

– П – потужність обладнання (ноутбук + принтер), приймаємо 0.15 кВт.
– Квп – коефіцієнт використання потужності, приймаємо 0.95.
– Тр – трудомісткість, 160 год.
– Ц1кВт – ціна 1 кВт·год, 2.64 грн.

$$\text{Вел} = 0.15 * 0.95 * 160 * 2.64 \approx 60 \text{ грн.}$$

Витрати на створення та розміщення продукту

Сукупні витрати на створення та розміщення програмного продукту складаються з загальної суми витрат.

$$\text{Соб.прод} = \text{Врозр} + \text{Вінш} \quad (4.7)$$

де:

– Врозр – витрати на розробку (оплата праці, амортизація, електроенергія), грн.
– Вінш – інші витрати, які не включені до вищеперелічених (приймаємо 10%).

$$\text{Врозр} = 33406 + 1350 + 60 = 34\,816 \text{ грн.}$$

$$\text{Вінш} = 34816 * 0.10 = 3\,482 \text{ грн.}$$

Загальна вартість створення (Вствор):

$$\text{Вствор} = 34816 + 3482 = 38\,298 \text{ грн.}$$

Розрахунок ціни реалізації та окупності

Собівартість одиниці (Спрод): Припустимо, проєкт виконується на замовлення для установки на 5 робочих місць.

$$\text{Спрод} = \frac{38298}{5} = 7\,660 \frac{\text{грн.}}{\text{од}}.$$

Ціна реалізації (Цр): Приймаємо рентабельність Р = 50%, ПДВ = 20%.

$$Цр = 7660 * (1 + 0.50) * (1 + 0.20) \approx 13\,788 \text{ грн.}$$

Чистий прибуток (ЧП):

$$ЧП = \left(\left(\frac{Цр}{1.2}\right) - Спрод\right) * 5 * (1 - 0.18) \approx 15\,623 \text{ грн.}$$

Термін окупності (Ток):

$$Ток = \frac{Вствор}{ЧП} = \frac{38298}{15623} \approx 2.45 \text{ роки.}$$

Таким чином, розраховані показники демонструють економічну обґрунтованість розробки, навіть при розгляді проєкту в рамках комерційного замовлення.

4.4 Охорона праці

Виконання кваліфікаційної роботи з розробки програмного забезпечення вимагало тривалої роботи за персональним комп'ютером. Тому організація робочого місця та дотримання норм безпеки є ключовими для збереження здоров'я та забезпечення високої продуктивності. Усі роботи проводились відповідно до вимог ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» та загальних правил електро- та пожежної безпеки.

Робота над проєктом здійснювалася в умовах домашнього офісу, де було створено безпечне та ергономічне робоче середовище:

– Робоче місце: Використовувався стіл стандартної висоти (75 см) та ергономічне крісло, що забезпечувало правильну поставу. Монітор було встановлено на рівні очей на відстані 60-70 см.

– Обладнання: Робота велася на сучасному ноутбуці з якісним дисплеєм, що мінімізує мерехтіння. Для зручності використовувалися зовнішня клавіатура та миша.

– Режим праці та відпочинку: Для профілактики втоми було встановлено регламентований режим роботи: після кожних 50 хвилин безперервної

роботи виконувалася 10-хвилинна перерва для відпочинку очей та виконання фізичних вправ.

– Електробезпека: Усе обладнання підключалося до мережі через стабілізатор напруги для захисту від можливих перебоїв в електромережі. Кабелі були впорядковані та захищені від механічних пошкоджень.

– Пожежна безпека: Приміщення відповідає нормам пожежної безпеки для житлових будівель. Було виключено використання несправного обладнання та перевантаження електромережі.

Для забезпечення комфортних умов праці важливим є достатній рівень освітлення. Робоче приміщення має наступні характеристики: площа S = 20 м², висота H = 2.8 м. Природне освітлення забезпечується двома вікнами загальною площею S_0 = 4.5 м², розташованими на одній стіні. Згідно з БНіП II-4-79, коефіцієнт природної освітленості (КПО) при бічному освітленні має становити не менше $e_н$ = 1,5%.

Розрахуємо нормоване значення КПО для даного приміщення, враховуючи коефіцієнт світлового клімату m = 0.9 та коефіцієнт сонячності C = 0.8 (вікна виходять на південь):

$$e_{норм} = e_н * m * C = 1.5 * 0.9 * 0.8 = 1.08\%$$

Розрахуємо фактичне значення КПО за формулою для бічного освітлення:

$$e = \left(\tau^0 * \eta^0 * \frac{S^0}{\left(K^3 * K_{зд} * S_п\right)}\right) * 100\% \quad (4.8)$$

де:

– τ_0 – загальний коефіцієнт світлопропускання;

– η_0 – світлова характеристика вікон;

– S_0 – площа світлових прорізів, м² (4.5 м²);

– $K_з$ – коефіцієнт запасу (приймемо 1.4);

– $K_{зд}$ – коефіцієнт, що враховує відбите світло (приймемо 1.2);

– $S_п$ – площа підлоги приміщення, м² (20 м²).

Загальний коефіцієнт світлопропускання τ_0 розраховується як:

$$\tau_0 = \tau_1 * \tau_2 * \tau_3 * \tau_4 \quad (4.9)$$

де:

– τ_1 (склопакет) = 0.8;

– τ_2 (металопластикова рама) = 0.7;

– τ_3 (несучі конструкції) = 1.0;

– τ_4 (сонцезахист, штори) = 0.9.

$$\tau_0 = 0.8 * 0.7 * 1.0 * 0.9 = 0.504$$

Світлова характеристика вікон η_0 для даного співвідношення розмірів приміщення становить 12.

Підставимо значення у формулу (4.8):

$$e = \left(\frac{0.504 * 12 * 4.5}{1.4 * 1.2 * 20}\right) * 100\% \approx 0.81\%$$

Отримане значення КПО (0.81%) є дещо нижчим за нормоване (1.08%). Це свідчить про необхідність використання додаткового штучного освітлення протягом робочого дня для створення комфортних умов праці. В приміщенні встановлені світлодіодні лампи, що забезпечують достатній рівень освітленості на робочій поверхні.

ВИСНОВКИ

У ході виконання даної кваліфікаційної роботи було розроблено інформаційну систему «OncoScreen Assist». За результатами роботи було досягнуто поставленої мети та повністю виконано всі дослідницькі завдання. Основні висновки є наступними:

1. Проведено аналіз предметної області та існуючих програмних аналогів (iCAD Profound AI Suite, Grail Galleri), що дозволило виявити незайняту нішу для гнучкого, прозорого та безкоштовного інструменту. На основі аналізу було обґрунтовано вибір технологічного стеку: Python для серверної логіки, XGBoost як прогностична модель, Optuna для автоматичної оптимізації гіперпараметрів та Flask для створення веб-сервісу.
2. Розроблено архітектуру та ключові модулі системи, що забезпечують її гнучкість та адаптивність. Центральним елементом архітектури є унікальний модуль адаптивного донавчання, який дозволяє оновлювати модель на основі нових даних користувача, що є ключовою перевагою порівняно зі статичними комерційними системами.
3. Реалізовано зручний крос-платформенний веб-інтерфейс, що забезпечує легку взаємодію з системою без необхідності спеціальної технічної підготовки. Інтерфейс логічно розділений на функціональні блоки прогнозування та донавчання, а також надає деталізовану візуалізацію результатів та метрик якості моделі.
4. Проведено комплексне тестування та оцінку якості системи. Тестування підтвердило коректну роботу всіх функціональних модулів та високу прогностичну здатність фінальної моделі. На відкладеній тестовій вибірці було досягнуто показників точності (Accuracy) 94.44% та ROC AUC 0.9879, що підтверджує практичну цінність та надійність розробленого рішення для використання у медичній практиці.

СПИСОК ЛІТЕРАТУРИ

1. Войтко В. В., Месюра В. І. Системи штучного інтелекту: Навчальний посібник. Вінниця: ВНТУ, 2017. 164 с.
2. Ковалюк Т. В. Основи програмування на Python. Львів: Видавництво Львівської політехніки, 2019. 321 с.
3. Кулик Я. В., Пасічник В. В. Інтелектуальний аналіз даних в медичних інформаційних системах. Вісник Національного університету «Львівська політехніка». 2020. № 901. С. 87-95.
4. Попова М. О., Лисенко С. М. Застосування методів машинного навчання для задач біоінформатики. Біополімери і клітина. 2019. Т. 35, № 2. С. 101-112.
5. Рижов А. Ю., Клименко В. І. Аналіз та прогнозування показників онкологічної захворюваності в Україні. Український журнал медицини, біології та спорту. 2021. Т. 6, № 5. С. 139-145.
6. Сидоренко В. В. Архітектура програмного забезпечення: Навчальний посібник. Харків: ХНУРЕ, 2019. 180 с.
7. Субботін С. О. Нейронні мережі та глибоке навчання: Навчальний посібник. Запоріжжя: ЗНТУ, 2021. 245 с.
8. Федоренко З. П., Сумкіна О. В., Горох Є. Л., Гулак Л. О., Куценко Л. Б. та ін. Рак в Україні, 2020-2021. Захворюваність, смертність, показники діяльності онкологічної служби. Бюлетень Національного канцер-реєстру України. Київ, 2022. №23. 136 с.
9. Шкляр В. Б., Жебка В. В. Методи та моделі аналізу даних: Навчальний посібник. Київ: КПІ ім. Ігоря Сікорського, 2018. 212 с.
10. Akiba T., Sano S., Yanase T., Ohta T., Koyama M. Optuna: A Next-generation Hyperparameter Optimization Framework. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019. P. 2623–2631.

11. Alpaydin E. Introduction to Machine Learning. 4th ed. MIT Press, 2020. 640 p.
12. Bishop C. M. Pattern Recognition and Machine Learning. Springer, 2006. 738 p.
13. Borkowski K., Płaza M. The Use of Machine Learning in the Diagnosis of Breast Cancer Based on Cytological Images: A Systematic Review. Journal of Clinical Medicine. 2021. Vol. 10, No. 16. P. 3636.
14. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. P. 785–794.
15. Esteva A. et al. A guide to deep learning in healthcare. Nature Medicine. 2019. Vol. 25. P. 24-29.
16. Friedman J. H. Greedy function approximation: A gradient boosting machine. Annals of Statistics. 2001. Vol. 29, No. 5. P. 1189-1232.
17. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. 800 p.
18. Grinberg M. Flask Web Development: Developing Web Applications with Python. 2nd ed. O'Reilly Media, 2018. 320 p.
19. Kourou K., Exarchos T. P., Exarchos K. P., Karamouzis M. V., Fotiadis D. I. Machine learning applications in cancer prognosis and prediction. Computational and Structural Biotechnology Journal. 2015. Vol. 13. P. 8-17.
20. Lundberg S. M., Lee S.-I. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems. 2017. Vol. 30.
21. McKinney W. Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference. 2010. P. 56-61.
22. Merkel D. Docker: Lightweight Linux Containers for Consistent Development and Deployment. Linux Journal. 2014. Vol. 2014, No. 239.
23. Pedregosa F. et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011. Vol. 12. P. 2825-2830.

24. Robbins S. L., Cotran R. S., Kumar V. Robbins and Cotran Pathologic Basis of Disease. 10th ed. Philadelphia: Elsevier, 2020. 1344 p.
25. Van Rossum G. The Python Language Reference Manual. Python Software Foundation, 2023. Available at: https://docs.python.org/3/reference/

ДОДАТОК А

Код фронтенду (index.html)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Cancer Detection Interface</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}" />
</head>
<body>
  <canvas id="neuron-canvas"></canvas>

  <div class="peripheral-text pt-top-left">SEQUENCE ANALYSIS</div>
  <div class="peripheral-text pt-top-right" id="system-clock">--:--:--</div>
  <div class="peripheral-text pt-bottom-left">SYSTEM OPERATIONAL</div>
  <div class="peripheral-text pt-bottom-right">Vladyslav Kuznietsov @ Tomorrow Labs 2025</div>

  <div class="container">
    <div class="corner-bl"></div>
    <div class="corner-br"></div>
    <span class="corner-label label-tl" id="status-label">SYS: READY</span>
    <span class="corner-label label-bl">MODEL: CONNECTED</span>
    <span class="corner-label label-br">VER: 2.4.0</span>

    <h1 class="main-title">CANCER PREDICTION</h1>
```

```
        <!-- Existing Prediction Panels -->
        <div class="left-panel">
          <div class="graphics-area">
            <img src="{{ url_for('static', filename='network-graphic.svg') }}"
id="network-graphic" alt="Network Graphic" />
          </div>
          <div class="result-area">
            <h2>RISK OF CANCER:</h2>
            <span id="result-text"></span>
          </div>
        </div>

        <div class="right-panel">
          <p>UPLOAD PATIENT DATA FOR ANALYSIS</p>
          <div class="file-upload-area">
            <label for="csv-file" class="file-button">SELECT ANALYSIS DATA
(.csv)</label>
            <input type="file" id="csv-file" accept=".csv" hidden />
            <span id="file-name">No file selected</span>
          </div>
          <button id="detect-button" class="detect-button"
disabled>ANALYZE</button>
        </div>

        <hr class="panel-divider">

        <div class="retrain-panel">
          <h2>MODEL RETRAINING & EVALUATION</h2>
```

```
<p>Upload a CSV with a 'diagnosis' column ('M'/'B') to retrain the model. Detailed performance metrics for the new model will be displayed below.</p>
<div class="file-upload-area">
  <label for="retrain-csv-file" class="file-button">SELECT RETRAINING DATA (.csv)</label>
  <input type="file" id="retrain-csv-file" accept=".csv" hidden />
  <span id="retrain-file-name">No file selected</span>
</div>
<button id="retrain-button" class="detect-button" disabled>RETRAIN MODEL</button>
<div id="retrain-status" class="retrain-status-message">Awaiting retraining data...</div>

<div id="metrics-container" class="metrics-container" style="display: none;">

  <!-- Overall Performance -->
  <div class="metric-section" id="overall-metrics-section">
    <h4>Overall Performance</h4>
    <div id="overall-metrics-display" class="overall-metrics-grid">
      <!-- Populated by JS -->
    </div>
  </div>

  <!-- Confusion Matrix -->
  <div class="metric-section" id="confusion-matrix-section">
    <h4>Confusion Matrix</h4>
    <div id="confusion-matrix-display">
      <!-- Populated by JS -->
```

```
            </div>
        </div>

        <!-- Feature Importances -->
        <div class="metric-section" id="feature-importance-section">
            <h4>Top Feature Importances</h4>
            <div id="feature-importance-display">
                <!-- Populated by JS -->
            </div>
        </div>

        <!-- Classification Report -->
        <div class="metric-section" id="classification-report-section">
            <h4>Classification Report</h4>
            <div id="classification-report-display">
                <!-- Populated by JS -->
            </div>
        </div>

    </div>

  </div>

 </div>

 <script src="{{ url_for('static', filename='script.js') }}"></script>
</body>
</html>
```

ДОДАТОК Б

Код фронтенду (style.css)

```
@import
url('https://fonts.googleapis.com/css2?family=Orbitron:wght@400;700&family=Share+Tech+Mono&display=swap');

:root {
    --glow-color: #00e5ff;
    --text-color: #adefff;
    --border-color: rgba(0, 229, 255, 0.3);
    --border-color-medium: rgba(0, 229, 255, 0.5);
    --bg-color: #02060a;
    --container-bg: rgba(10, 30, 30, 0.6);
    --disabled-color: rgba(0, 229, 255, 0.2);
    --glow-shadow: 0 0 6px var(--glow-color);
    --scanline-color: rgba(173, 239, 255, 0.04);

    --neuron-line-base-alpha: 0.15;
    --neuron-gradient-top: 255, 180, 0; /* Yellow/Orange */
    --neuron-gradient-bottom: 0, 229, 255; /* Teal/Cyan */
    --neuron-dot-color: rgba(255, 255, 255, 0.9);
    --neuron-dot-glow-color: rgba(200, 255, 255, 0.4);
    --neuron-line-glow-color: rgba(0, 229, 255, 0.4);

    --space-sm: 10px;
    --space-md: 15px;
    --space-lg: 20px;
    --space-xl: 25px;
    --space-xxl: 40px;
```

```
        /* -- Add a variable for the result color -- */
        --result-glow-color: var(--glow-color); /* Default to base glow */
    }

    * {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
    }

    body {
        font-family: 'Share Tech Mono', monospace;
        background-color: var(--bg-color);
        color: var(--text-color);
        display: flex;
        justify-content: center;
        align-items: flex-start;
        min-height: 100vh;
        padding: 50px var(--space-lg);
        position: relative;
        overflow-y: auto; /* Changed from 'hidden' to enable vertical scrollbar when needed */
    }

    body::after {
        content: '';
        position: fixed;
        inset: 0;
        background: linear-gradient( var(--scanline-color) 1px, transparent 1px );
        background-size: 100% 3px;
```

```
            z-index: 2;
            pointer-events: none;
            animation: flicker 0.15s infinite;
        }
        @keyframes flicker { 0% { opacity: 1; } 50% { opacity: 0.95; } 100% { opacity: 1; } }

        #neuron-canvas {
            position: fixed;
            inset: 0;
            z-index: 1;
            pointer-events: none;
        }

        .container {
            display: flex;
            flex-wrap: wrap;
            gap: var(--space-xxl);
            padding: var(--space-xxl) 50px;
            background-color: var(--container-bg);
            border: 1px solid var(--border-color-medium);
            box-shadow: 0 0 20px rgba(0, 229, 255, 0.1);
            max-width: 800px;
            width: 100%;
            z-index: 10;
            position: relative;
        }

         .container::before, .container::after,
        .corner-bl, .corner-br {
```

```
            content: '';
            position: absolute;
            width: var(--space-md); height: var(--space-md);
            border-color: var(--border-color-medium);
            border-style: solid;
            pointer-events: none;
            z-index: 11;
        }
        .container::before { top: -1px; left: -1px; border-width: 1px 0 0 1px; }
        .container::after { top: -1px; right: -1px; border-width: 1px 1px 0 0; }
        .corner-bl { bottom: -1px; left: -1px; border-width: 0 0 1px 1px; }
        .corner-br { bottom: -1px; right: -1px; border-width: 0 1px 1px 0; }

        .corner-label {
            position: absolute;
            font-size: 0.7em;
            color: var(--border-color-medium);
            pointer-events: none;
            z-index: 11;
        }
        .label-tl { top: 5px; left: var(--space-xl); }
        .label-bl { bottom: 5px; left: var(--space-xl); }
        .label-br { bottom: 5px; right: var(--space-xl); }

        #status-label { transition: color 0.5s ease; }
        #status-label.processing { color: var(--glow-color); text-shadow: var(--glow-shadow); }

        .main-title {
            width: 100%;
```

```
    text-align: center;
    font-family: 'Orbitron', sans-serif;
    font-size: 2.2em;
    margin-bottom: var(--space-lg);
    color: var(--glow-color);
    text-shadow: var(--glow-shadow);
    letter-spacing: 3px;
    padding: 5px 0;
}

.left-panel, .right-panel {
    flex: 1;
    min-width: 280px;
    display: flex;
    flex-direction: column;
    align-items: center;
    padding: var(--space-sm);
}
.left-panel { justify-content: space-around; }

.graphics-area {
    position: relative;
    width: 180px; height: 180px;
    margin-bottom: var(--space-xl);
}

#network-graphic, #hands-graphic {
    position: absolute;
    width: 100%; height: auto;
    left: 0;
```

```
}

#network-graphic {
    top: 0;
    z-index: 10;
    filter: brightness(125%) saturate(90%) drop-shadow(var(--glow-shadow));
}

#hands-graphic {
    bottom: -30px;
    z-index: 5;
    opacity: .5;
    filter: drop-shadow(var(--glow-shadow));
}

#hands-graphic path {
    stroke-width: 1px;
    stroke: var(--text-color);
    fill: none;
    transition: stroke .3s ease;
}

.result-area {
    /* Fixed dimensions as requested */
    width: 309px;
    height: 163.8px;

    /* Styling for content within the fixed block */
    display: flex;
    flex-direction: column;
```

```
    justify-content: center; /* Vertically center content */
    align-items: center;   /* Horizontally center content block */
    overflow: hidden;      /* Clip content that exceeds dimensions */

    /* Existing styles */
    text-align: center;
    margin-top: var(--space-md);
    padding: var(--space-sm); /* Add some internal padding */
    border: 1px solid var(--border-color);
    background: rgba(0,30,40,.1);
    box-sizing: border-box; /* Ensure padding is within width/height */
}

.result-area h2 {
    font-size: 1.1em;
    margin-bottom: var(--space-sm);
    color: var(--glow-color);
    font-weight: 400;
    letter-spacing: 2px;
    text-shadow: none;
}

#result-text {
    font-size: 1.2em;
    line-height: 1.25;
    display: block;
    width: 100%;
    font-family: 'Orbitron', sans-serif;
    color: var(--result-glow-color);
    text-shadow: 0 0 6px var(--result-glow-color);
```

```
            transition: color 0.5s ease, text-shadow 0.5s ease;
            text-align: center;
            box-sizing: border-box;
            white-space: normal;
            overflow-wrap: break-word;
        }

        #result-text.typing {
            border-right: 3px solid var(--result-glow-color);
            white-space: nowrap; /* Essential for typing effect, text stays on one line */
            overflow: visible; /* Hide the part of the single line that goes beyond the element's JS-set width */
            /* Add a subtle transition for the width update during typing */
            transition: color 0.5s ease, text-shadow 0.5s ease, border-color 0.5s ease;
            animation: typing-cursor steps(1) .6s infinite;
            text-align: left;
        }

        #result-text:not(.typing) {
            border-right-color: transparent;
            animation: none;
            /* width: 100%; is already set on #result-text, JS will clear inline style */
            white-space: normal;            /* Allow text to wrap */
            overflow-wrap: break-word;      /* Break long words if necessary */
            /* Parent .result-area handles overall overflow clipping */
        }

        @keyframes typing-cursor {
            0%, 100% { border-color: transparent; }
            50% { border-color: var(--result-glow-color); }
```

```
}


.right-panel h2 {
    font-size: 1.2em;
    margin-bottom: var(--space-sm);
    color: var(--glow-color);
    font-weight: 400;
    letter-spacing: 1px;
    text-shadow: none;
    text-align: center;
}
.right-panel p {
    margin-bottom: var(--space-lg);
    font-size: .9em;
    color: var(--text-color);
    letter-spacing: 1px;
    text-shadow: none;
    text-align: center;
}

.file-upload-area {
    display: flex;
    flex-direction: column;
    align-items: center;
    gap: var(--space-md);
    margin-bottom: var(--space-xl);
    width: 90%;
    padding: var(--space-lg);
    border: 1px solid var(--border-color);
```

```
            background: rgba(0,30,40,.1);
            position: relative;
        }
        .file-button {
            display: inline-block;
            padding: 8px 20px;
            border: 1px solid var(--border-color-medium);
            background: transparent;
            color: var(--glow-color);
            cursor: pointer;
            transition: background .2s, border-color .2s, color .2s;
            font-family: 'Share Tech Mono', monospace;
            font-size: .9em;
            letter-spacing: 1px;
            box-shadow: none;
        }
        .file-button:hover { background: rgba(0,229,255,.1); border-color: var(--glow-color); }

        #file-name {
            font-size: .9em;
            color: var(--text-color);
            min-height: 1.2em;
            max-width: 95%;
            overflow: hidden;
            text-overflow: ellipsis;
            white-space: nowrap;
            text-align: center;
            padding: 5px;
        }
```

```
        .detect-button {
            font-family: 'Share Tech Mono', monospace;
            padding: 10px 35px;
            font-size: 1.1em;
            background: transparent;
            border: 1px solid var(--border-color);
            color: var(--text-color);
            cursor: pointer;
            position: relative;
            transition: background-color .2s ease, border-color .2s ease, color .2s ease,
opacity .3s ease;
            min-width: 150px;
            opacity: .6;
            letter-spacing: 2px;
            text-shadow: none;
            box-shadow: none;
        }
        .detect-button:not(:disabled) {
            opacity: 1;
            color: var(--glow-color);
            border-color: var(--border-color-medium);
        }
        .detect-button:not(:disabled):hover { background-color: rgba(0,229,255,.1);
border-color: var(--glow-color); }
        .detect-button:disabled {
            cursor: not-allowed;
            opacity: .3;
            color: var(--disabled-color);
            border-color: var(--disabled-color);
```

```
}
.detect-button.processing {
    color: var(--glow-color);
    border-color: var(--glow-color);
}

.peripheral-text {
    position: fixed;
    font-size: 0.8em;
    color: var(--text-color);
    opacity: 0.7;
    letter-spacing: 1px;
    z-index: 15;
    text-shadow: 0 0 3px var(--text-color);
    pointer-events: none;
}
.pt-top-left { top: 30px; left: 40px; }
.pt-top-right { top: 30px; right: 40px; text-align: right; }
.pt-bottom-left { bottom: 30px; left: 40px; }
.pt-bottom-right { bottom: 30px; right: 40px; text-align: right; }
.pt-bottom-left::before {
    content: '●'; color: #33ff57; margin-right: 8px;
    animation: blink 1.5s linear infinite; display: inline-block;
}
@keyframes blink { 0%, 100% { opacity: 1; } 50% { opacity: 0.2; } }

#system-clock { transition: color 0.5s ease; }

@media (max-width: 850px) {
    .container {
```

```
            max-width: 95%;
            padding: var(--space-lg);
            gap: var(--space-lg);
        }
        .main-title { font-size: 1.8em; }
        .peripheral-text:not(#system-clock):not(#status-label) { display: none; }
        #system-clock { top: var(--space-sm); right: var(--space-sm); font-size: 0.7em; opacity: 0.6; z-index: 15; display: block; }
        .corner-label:not(#status-label) { display: none; }
        #status-label { top: var(--space-sm); left: var(--space-sm); font-size: 0.7em; opacity: 0.6; z-index: 15; display: block; }
    }
    @media (max-width: 600px) {
        .container { flex-direction: column; }
        .left-panel { order: 2; margin-top: var(--space-sm); width: 100%; }
        .right-panel { order: 1; width: 100%; }
        .graphics-area { width: 150px; height: 150px; margin-top: var(--space-sm); margin-bottom: var(--space-md);}
        /* #result-text font-size adjusted globally, check if further media query needed */
        .detect-button { padding: 8px 25px; font-size: 1em;}
        .file-button { font-size: 0.8em; padding: 6px 15px;}
    }

    .panel-divider {
        width: 100%;
        border: none;
        height: 1px;
        background-color: var(--border-color-medium);
        margin: var(--space-lg) 0;
```

```
}

.retrain-panel {
    width: 100%; /* Take full width of the container row */
    display: flex;
    flex-direction: column;
    align-items: center;
    padding: var(--space-lg);
    border: 1px dashed var(--border-color); /* Dashed border to differentiate */
    margin-top: var(--space-lg);
}

.retrain-panel h2 {
    font-family: 'Orbitron', sans-serif;
    font-size: 1.5em;
    margin-bottom: var(--space-md);
    color: var(--glow-color);
    text-shadow: var(--glow-shadow);
    letter-spacing: 2px;
}

.retrain-panel p {
    font-size: 0.85em;
    color: var(--text-color);
    margin-bottom: var(--space-lg);
    text-align: center;
    max-width: 80%;
    line-height: 1.5;
}
```

```
/* Can reuse .file-upload-area and .detect-button styles */

.retrain-status-message {
    margin-top: var(--space-md);
    padding: var(--space-sm) var(--space-md);
    font-size: 0.9em;
    color: var(--text-color);
    border: 1px solid transparent; /* Initially transparent border */
    min-height: 2em;
    text-align: center;
}

.retrain-status-message.processing {
    color: var(--glow-color);
    border-color: var(--glow-color);
    text-shadow: var(--glow-shadow);
}

.retrain-status-message.success {
    color: #33ff57; /* Greenish for success */
    border-color: #33ff57;
}

.retrain-status-message.error {
    color: #ff4d4d; /* Reddish for error */
    border-color: #ff4d4d;
}

.metric-section {
    width: 100%; /* Make sections take full width if not already */
```

```
    margin-bottom: var(--space-lg);
    text-align: left; /* Align section content to the left */
}

.metric-section h4 {
    font-size: 1em;
    color: var(--text-color); /* Or var(--glow-color) if preferred */
    margin-bottom: var(--space-sm);
    border-bottom: 1px solid var(--border-color-medium);
    padding-bottom: 5px;
    text-align: center; /* Center the heading itself */
}

#feature-importance-display {
    font-size: 0.85em;
    line-height: 1.6;
    background-color: rgba(0,0,0,0.1); /* Darker background for the list */
    padding: var(--space-md); /* More padding */
    border-radius: 4px;
    max-height: 250px; /* Increased max height */
    overflow-y: auto;  /* Scroll for longer lists */
    border: 1px solid var(--border-color); /* Added border */
}

#feature-importance-display ul {
    list-style-type: none;
    padding-left: 0;
    margin: 0;
}
```

```
        #feature-importance-display li {
            display: flex;
            justify-content: space-between;
            align-items: center; /* Vertically align items in the list item */
            padding: 6px 0; /* Increased padding */
            border-bottom: 1px solid var(--border-color-medium); /* Slightly stronger
border */
        }
        #feature-importance-display li:last-child {
            border-bottom: none;
        }

        #feature-importance-display .feature-name {
            color: var(--text-color);
            flex-basis: 40%; /* Give more defined space to feature name */
            padding-right: var(--space-sm);
            white-space: nowrap;
            overflow: hidden;
            text-overflow: ellipsis; /* Handle very long feature names */
        }

        #feature-importance-display .importance-bar-container {
            flex-basis: 40%; /* Define space for bar */
            background-color: rgba(var(--glow-color-rgb, 0, 229, 255), 0.1);
            border-radius: 3px; /* Slightly more rounded */
            height: 1.2em; /* Slightly taller bar */
            position: relative;
            margin: 0 var(--space-sm);
        }
```

```
#feature-importance-display .importance-bar {
    background-color: var(--glow-color);
    height: 100%;
    border-radius: 3px;
    transition: width 0.5s ease-in-out;
    box-shadow: 0 0 4px var(--glow-color); /* Add subtle glow to the bar */
}

#feature-importance-display .feature-score {
    color: var(--glow-color);
    font-weight: bold;
    flex-basis: 20%; /* Define space for score */
    text-align: right;
    font-size: 0.9em; /* Slightly smaller score text */
}

/* ... (keep all existing CSS) ... */

.retrain-panel h2 {
    font-family: 'Orbitron', sans-serif;
    font-size: 1.5em;
    margin-bottom: var(--space-md);
    color: var(--glow-color);
    text-shadow: var(--glow-shadow);
    letter-spacing: 2px;
}

/* ... (keep existing retrain-panel styles) ... */

.retrain-status-message.error {
```

```
    color: #ff4d4d; /* Reddish for error */
    border-color: #ff4d4d;
}

/* --- NEW: Metrics Container --- */
.metrics-container {
    width: 100%;
    margin-top: var(--space-xl);
    display: grid;
    grid-template-columns: 1fr 1fr; /* Two-column layout */
    gap: var(--space-xl);
    border-top: 1px solid var(--border-color-medium);
    padding-top: var(--space-xl);
}

.metric-section {
    width: 100%;
    background-color: rgba(0,0,0,0.1);
    border: 1px solid var(--border-color);
    padding: var(--space-md);
}

.metric-section h4 {
    font-size: 1.1em;
    color: var(--glow-color);
    margin-bottom: var(--space-md);
    border-bottom: 1px solid var(--border-color-medium);
    padding-bottom: 8px;
    text-align: center;
    font-family: 'Orbitron', sans-serif;
```

```
    letter-spacing: 1px;
}

/* --- Specific section styling --- */
#feature-importance-section,
#classification-report-section {
    grid-column: span 2; /* Make these sections span both columns */
}

/* --- Overall Performance Grid --- */
.overall-metrics-grid {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: var(--space-md);
    text-align: center;
}
.metric-item {
    padding: var(--space-sm);
    background-color: rgba(0,0,0,0.2);
}
.metric-item .metric-label {
    font-size: 0.8em;
    color: var(--text-color);
    display: block;
    margin-bottom: 5px;
}
.metric-item .metric-value {
    font-size: 1.4em;
    color: var(--glow-color);
    font-weight: bold;
```

```
    font-family: 'Orbitron', sans-serif;
    text-shadow: var(--glow-shadow);
}

/* --- NEW: Confusion Matrix Styling --- */
#confusion-matrix-display {
    display: grid;
    grid-template-columns: auto 1fr 1fr;
    grid-template-rows: auto 1fr 1fr;
    gap: 5px;
    align-items: center;
    font-family: 'Share Tech Mono', monospace;
}
.cm-cell {
    padding: var(--space-sm);
    text-align: center;
    border: 1px solid var(--border-color);
}
.cm-header {
    font-weight: bold;
    color: var(--glow-color);
    background: transparent;
    border: none;
    font-size: 0.9em;
}
.cm-label {
    font-weight: bold;
    writing-mode: vertical-rl;
    text-orientation: mixed;
    transform: rotate(180deg);
```

```
    color: var(--glow-color);
    border: none;
    background: transparent;
    font-size: 0.9em;
}
.cm-value {
    font-size: 1.5em;
    font-weight: bold;
}
.cm-tn, .cm-tp { background-color: rgba(0, 229, 255, 0.2); color: #fff; } /* Correct predictions */
.cm-fn, .cm-fp { background-color: rgba(255, 180, 0, 0.2); color: #fff; } /* Incorrect predictions */


/* Feature Importance (existing styles are good, just minor adjustments if needed) */
#feature-importance-display {
    font-size: 0.85em;
    line-height: 1.6;
    padding: var(--space-md);
    max-height: 250px;
    overflow-y: auto;
}

/* ... (keep existing feature importance styles) ... */

/* --- NEW: Classification Report Table --- */
#classification-report-display table {
    width: 100%;
```

```
    border-collapse: collapse;
    font-size: 0.85em;
}
#classification-report-display th,
#classification-report-display td {
    border: 1px solid var(--border-color);
    padding: 8px 12px;
    text-align: left;
}
#classification-report-display th {
    background-color: rgba(0, 229, 255, 0.1);
    color: var(--glow-color);
    font-family: 'Orbitron', sans-serif;
}
#classification-report-display tbody tr:hover {
    background-color: rgba(0, 229, 255, 0.05);
}
#classification-report-display td {
    color: var(--text-color);
}
#classification-report-display .report-label {
    font-weight: bold;
}
#classification-report-display .report-value {
    text-align: right;
    font-family: 'Share Tech Mono', monospace;
    font-weight: bold;
}

/* Media query for smaller screens */
```

```
@media (max-width: 850px) {
    /* ... (keep existing media queries) ... */
    .metrics-container {
        grid-template-columns: 1fr; /* Stack metrics on smaller screens */
    }
    #feature-importance-section,
    #classification-report-section {
        grid-column: span 1; /* Reset span for single column */
    }
}
```

ДОДАТОК В

Код фронтенду (script.js)

```
// --- Constants and Configuration ---
const ANALYSIS_MIN_TIME = 1200;
const ANALYSIS_RANDOM_TIME = 1800;
const TYPING_DELAY = 100;
const TYPING_END_PAUSE = 800;
const PARTICLE_SPEED_MULTIPLIER_ANALYZING = 2.5;
const NUM_PARTICLES_BASE = 150;
const MAX_CONNECT_DISTANCE = 150;
const PARTICLE_BASE_SPEED = 0.3;
const PARTICLE_RADIUS = 2.0;
const DRAW_PARTICLES = true;
const PARTICLE_COUNT_MIN = 50;
const PARTICLE_COUNT_MAX = 300;
const LINE_GLOW_BLUR = 5;
const DOT_GLOW_BLUR = 3;
const PLACEHOLDER_TEXT = "AWAITING INPUT";

const resultColors = {
   "VERY LOW":  "#00e5ff",
   "LOW":       "#40d9bf",
   "MEDIUM":    "#80cd7f",
   "HIGH":      "#c0c13f",
   "VERY HIGH": "#FFB400",
   "DEFAULT":   "#00e5ff"
};

// --- DOM Elements ---
const fileInput = document.getElementById("csv-file");
```

```
        const fileNameDisplay = document.getElementById("file-name");
        const detectButton = document.getElementById("detect-button");
        const resultText = document.getElementById("result-text");
        const statusLabel = document.getElementById("status-label");
        const clockDisplay = document.getElementById("system-clock");

        // --- Retraining and Metrics DOM Elements ---
        const retrainFileInput = document.getElementById("retrain-csv-file");
        const retrainFileNameDisplay = document.getElementById("retrain-file-name");
        const retrainButton = document.getElementById("retrain-button");
        const retrainStatusDisplay = document.getElementById("retrain-status");
        const metricsContainer = document.getElementById("metrics-container");

        // --- State Variables ---
        let typingInterval;
        let isAnalyzing = false;
        let isRetraining = false;

        // --- Canvas Variables ---
        const canvas = document.getElementById('neuron-canvas');
        const ctx = canvas.getContext('2d');
        let particles = [];
        let canvasWidth = window.innerWidth;
        let canvasHeight = window.innerHeight;

        // --- Style Variables ---
        const computedStyle = getComputedStyle(document.documentElement);
        const baseLineAlpha = parseFloat(computedStyle.getPropertyValue('--neuron-line-base-alpha')) || 0.15;
```

```
        const lineGlowColor = computedStyle.getPropertyValue('--neuron-line-glow-color');
        const dotColor = computedStyle.getPropertyValue('--neuron-dot-color');
        const dotGlowColor = computedStyle.getPropertyValue('--neuron-dot-glow-color');
        const gradientTopRGB = computedStyle.getPropertyValue('--neuron-gradient-top').split(',').map(s => parseInt(s.trim()));
        const gradientBottomRGB = computedStyle.getPropertyValue('--neuron-gradient-bottom').split(',').map(s => parseInt(s.trim()));
        const rootBgColorMatch = computedStyle.getPropertyValue('--bg-color').match(/\d+/g);
        const bgColorCanvas = rootBgColorMatch ? `rgba(${rootBgColorMatch.join(', ')}, 0.15)` : 'rgba(2, 6, 10, 0.15)';

        // --- Event Listeners ---
        fileInput.addEventListener("change", handleFileChange);
        detectButton.addEventListener("click", handleDetectClick);
        window.addEventListener('resize', resizeCanvas);

        if (retrainFileInput) {
            retrainFileInput.addEventListener("change", handleRetrainFileChange);
        }
        if (retrainButton) {
            retrainButton.addEventListener("click", handleRetrainClick);
        }

        // --- Analysis Functions ---

        function setPlaceholderText() {
            resultText.textContent = PLACEHOLDER_TEXT;
```

```
            resultText.style.setProperty('--result-glow-color',
resultColors.DEFAULT);
        }

        function handleFileChange() {
            if (fileInput.files.length > 0) {
                fileNameDisplay.textContent = fileInput.files[0].name;
                detectButton.disabled = false;
                statusLabel.textContent = "FILE LOADED";
                 setPlaceholderText();
            } else {
                fileNameDisplay.textContent = "No file selected";
                detectButton.disabled = true;
                statusLabel.textContent = "SYS: READY";
                setPlaceholderText();
            }
        }

        function displayResult(text, isError = false) {
            resultText.textContent = text;
            let resultColor = resultColors.DEFAULT;
            if (!isError) {
                if (text.toLowerCase().includes("high risk")) {
                    resultColor = resultColors["VERY HIGH"];
                } else if (text.toLowerCase().includes("low risk")) {
                    resultColor = resultColors["VERY LOW"];
                }
            }
            resultText.style.setProperty('--result-glow-color', resultColor);
        }
```

```
        function handleDetectClick() {
            if (fileInput.files.length === 0 || isAnalyzing) return;

            isAnalyzing = true;
            detectButton.disabled = true;
            detectButton.textContent = "ANALYZING...";
            detectButton.classList.add("processing");
            statusLabel.textContent = "SYS: PROCESSING";
            statusLabel.classList.add("processing");
            resultText.textContent = "Processing...";
            resultText.style.setProperty('--result-glow-color',
resultColors.DEFAULT);

            const file = fileInput.files[0];
            const formData = new FormData();
            formData.append("csv_file", file);

            fetch("/analyze", {
                method: "POST",
                body: formData
            })
            .then(response => {
                if (!response.ok) {
                    return response.json().then(errData => {
                        throw new Error(errData.error || `Server error: ${response.status}`);
                    });
                }
                return response.json();
            })
```

```
            .then(data => {
                isAnalyzing = false;
                detectButton.classList.remove("processing");
                detectButton.textContent = "ANALYZE";
                detectButton.disabled = !(fileInput.files.length > 0);
                statusLabel.classList.remove("processing");

                if (data.error) {
                    statusLabel.textContent = "SYS: ERROR";
                    displayResult("ERROR: " + data.error, true);
                } else if (data.result) {
                    statusLabel.textContent = fileInput.files.length > 0 ? "SYS: ANALYSIS DONE" : "SYS: READY";
                    displayResult(data.result);
                }
            })
            .catch(error => {
                isAnalyzing = false;
                detectButton.disabled = !(fileInput.files.length > 0);
                detectButton.classList.remove("processing");
                detectButton.textContent = "ANALYZE";
                statusLabel.textContent = "SYS: FAILED";
                statusLabel.classList.remove("processing");
                displayResult("ANALYSIS FAILED: " + error.message, true);
            });
        }

        function updateClock() {
            const now = new Date();
            const h = String(now.getHours()).padStart(2, '0');
```

```
            const m = String(now.getMinutes()).padStart(2, '0');
            const s = String(now.getSeconds()).padStart(2, '0');
            if (clockDisplay) { clockDisplay.textContent = `${h}:${m}:${s}`; }
        }

        // --- Retraining and Metrics Functions ---

        function handleRetrainFileChange() {
            if (retrainFileInput.files.length > 0) {
                retrainFileNameDisplay.textContent = retrainFileInput.files[0].name;
                retrainButton.disabled = false;
                retrainStatusDisplay.textContent = "Retraining file selected. Ready to retrain.";
                retrainStatusDisplay.className = 'retrain-status-message';
                if (metricsContainer) metricsContainer.style.display = 'none'; // Hide old metrics on new file selection
            } else {
                retrainFileNameDisplay.textContent = "No file selected";
                retrainButton.disabled = true;
                retrainStatusDisplay.textContent = "Awaiting retraining data...";
                retrainStatusDisplay.className = 'retrain-status-message';
            }
        }

        function handleRetrainClick() {
            if (retrainFileInput.files.length === 0 || isRetraining) return;

            isRetraining = true;
            retrainButton.disabled = true;
            retrainButton.textContent = "RETRAINING...";
```

```
        retrainStatusDisplay.textContent = "Retraining in progress... This may take several minutes.";
        retrainStatusDisplay.className = 'retrain-status-message processing';
        if (metricsContainer) metricsContainer.style.display = 'none'; // Hide while processing

        const file = retrainFileInput.files[0];
        const formData = new FormData();
        formData.append("retrain_csv_file", file);

        fetch("/retrain", {
            method: "POST",
            body: formData
        })
        .then(response => {
            if (!response.ok) {
                return response.json().then(errData => { throw new Error(errData.error || `Server error: ${response.status}`); });
            }
            return response.json();
        })
        .then(data => {
            isRetraining = false;
            retrainButton.textContent = "RETRAIN MODEL";
            retrainButton.disabled = !(retrainFileInput.files.length > 0);

            if (data.error) {
                retrainStatusDisplay.textContent = "Retraining Failed: " + data.error;
                retrainStatusDisplay.className = 'retrain-status-message error';
            } else if (data.message) {
```

```
                retrainStatusDisplay.textContent = data.message;
                retrainStatusDisplay.className = 'retrain-status-message success';
                if (statusLabel) statusLabel.textContent = "SYS: MODEL UPDATED";

                // Call rendering functions for all the new data
                if (data.final_model_accuracy && data.roc_auc_score) {
                    displayOverallMetrics(data.final_model_accuracy, data.roc_auc_score);
                }
                if (data.confusion_matrix) {
                    displayConfusionMatrix(data.confusion_matrix);
                }
                if (data.classification_report) {
                    displayClassificationReport(data.classification_report);
                }
                if (data.top_feature_importances) {
                    displayFeatureImportances(data.top_feature_importances);
                }

                if (metricsContainer) metricsContainer.style.display = 'grid'; // Show the populated container
            }
        })
        .catch(error => {
            isRetraining = false;
            retrainButton.disabled = !(retrainFileInput.files.length > 0);
            retrainButton.textContent = "RETRAIN MODEL";
            retrainStatusDisplay.textContent = "Retraining Request Failed: " + error.message;
```

```
            retrainStatusDisplay.className = 'retrain-status-message error';
        });
    }

    function displayOverallMetrics(accuracy, roc_auc) {
        const display = document.getElementById('overall-metrics-display');
        if (!display) return;
        display.innerHTML = `
            <div class="metric-item">
                <span class="metric-label">Accuracy</span>
                <span class="metric-value">${(parseFloat(accuracy) * 100).toFixed(2)}%</span>
            </div>
            <div class="metric-item">
                <span class="metric-label">ROC AUC Score</span>
                <span class="metric-value">${parseFloat(roc_auc).toFixed(4)}</span>
            </div>
        `;
    }

    function displayConfusionMatrix(matrix) {
        const display = document.getElementById('confusion-matrix-display');
        if (!display || !matrix || matrix.length < 2 || matrix[0].length < 2) return;

        // Assuming matrix is [[TN, FP], [FN, TP]]
        const [TN, FP] = matrix[0];
        const [FN, TP] = matrix[1];

        display.innerHTML = `
```

```
            <div class="cm-label">Actual</div>
            <div class="cm-header">Pred. Benign</div>
            <div class="cm-header">Pred. Malignant</div>

            <div class="cm-header">Benign</div>
            <div class="cm-cell cm-value cm-tn" title="True Negative">${TN}</div>
            <div class="cm-cell cm-value cm-fp" title="False Positive">${FP}</div>

            <div class="cm-header">Malignant</div>
            <div class="cm-cell cm-value cm-fn" title="False Negative">${FN}</div>
            <div class="cm-cell cm-value cm-tp" title="True Positive">${TP}</div>
        `;
    }

    function displayClassificationReport(report) {
        const display = document.getElementById('classification-report-display');
        if (!display || !report) return;

        const table = document.createElement('table');
        table.innerHTML = `
            <thead>
                <tr>
                    <th>Class</th>
                    <th>Precision</th>
                    <th>Recall</th>
                    <th>F1-Score</th>
```

```
                <th>Support</th>
            </tr>
        </thead>
        <tbody></tbody>
    `;
    const tbody = table.querySelector('tbody');
    const classOrder = ['Benign (0)', 'Malignant (1)', 'macro avg', 'weighted avg'];

    classOrder.forEach(className => {
        if (report[className]) {
            const data = report[className];
            const row = document.createElement('tr');
            row.innerHTML = `
                <td class="report-label">${className}</td>
                <td class="report-value">${data.precision !== undefined ? data.precision.toFixed(3) : 'N/A'}</td>
                <td class="report-value">${data.recall !== undefined ? data.recall.toFixed(3) : 'N/A'}</td>
                <td class="report-value">${data['f1-score'] !== undefined ? data['f1-score'].toFixed(3) : 'N/A'}</td>
                <td class="report-value">${data.support !== undefined ? data.support : 'N/A'}</td>
            `;
            tbody.appendChild(row);
        }
    });

    display.innerHTML = ''; // Clear previous content
    display.appendChild(table);
```

```
    }

    function displayFeatureImportances(importances) {
        const display = document.getElementById('feature-importance-display');
        if (!display || !importances || importances.length === 0) {
            if(display) display.innerHTML = '<p>No feature importance data to display.</p>';
            return;
        }

        display.innerHTML = '';
        const maxImportance = Math.max(...importances.map(item => item.importance));
        const list = document.createElement('ul');

        importances.forEach(item => {
            const listItem = document.createElement('li');
            const barWidth = maxImportance > 0 ? (item.importance / maxImportance) * 100 : 0;
            listItem.innerHTML = `
                <span class="feature-name" title="${item.feature}">${item.feature}</span>
                <div class="importance-bar-container">
                    <div class="importance-bar" style="width: ${barWidth}%;"></div>
                </div>
                <span class="feature-score">${item.importance.toFixed(4)}</span>
            `;
            list.appendChild(listItem);
        });
```

```
        display.appendChild(list);
    }


    // --- Canvas Drawing and Animation ---
    function resizeCanvas() {
        canvasWidth = window.innerWidth;
        canvasHeight = window.innerHeight;
        canvas.width = canvasWidth;
        canvas.height = canvasHeight;
        initParticles();
    }

    class Particle {
        constructor() {
            this.x = Math.random() * canvasWidth;
            this.y = Math.random() * canvasHeight;
            this.baseVx = (Math.random() - 0.5) * PARTICLE_BASE_SPEED * 2;
            this.baseVy = (Math.random() - 0.5) * PARTICLE_BASE_SPEED * 2;
        }

        update() {
            const currentSpeedMultiplier = (isAnalyzing || isRetraining) ?
PARTICLE_SPEED_MULTIPLIER_ANALYZING : 1.0;
            const vx = this.baseVx * currentSpeedMultiplier;
            const vy = this.baseVy * currentSpeedMultiplier;

            this.x += vx;
            this.y += vy;
```

```
            if (this.x < -PARTICLE_RADIUS * 2) this.x = canvasWidth + PARTICLE_RADIUS * 2;
            else if (this.x > canvasWidth + PARTICLE_RADIUS * 2) this.x = -PARTICLE_RADIUS * 2;
            if (this.y < -PARTICLE_RADIUS * 2) this.y = canvasHeight + PARTICLE_RADIUS * 2;
            else if (this.y > canvasHeight + PARTICLE_RADIUS * 2) this.y = -PARTICLE_RADIUS * 2;
        }

        draw() {
            if (!DRAW_PARTICLES) return;
            ctx.save();
            ctx.shadowBlur = DOT_GLOW_BLUR;
            ctx.shadowColor = dotGlowColor;
            ctx.beginPath();
            ctx.arc(this.x, this.y, PARTICLE_RADIUS, 0, Math.PI * 2);
            ctx.fillStyle = dotColor;
            ctx.fill();
            ctx.restore();
        }
    }

    function initParticles() {
        particles = [];
        const refArea = 1920 * 1080;
        const currentArea = canvasWidth * canvasHeight;
        const adjustedNumParticles = Math.floor(NUM_PARTICLES_BASE * (currentArea / refArea));
```

```
        const finalParticleCount = Math.max(PARTICLE_COUNT_MIN, Math.min(PARTICLE_COUNT_MAX, adjustedNumParticles));

        for (let i = 0; i < finalParticleCount; i++) {
            particles.push(new Particle());
        }
    }

    function drawConnections() {
        ctx.save();
        ctx.shadowBlur = (isAnalyzing || isRetraining) ? LINE_GLOW_BLUR * 1.5 : LINE_GLOW_BLUR;
        ctx.shadowColor = lineGlowColor;
        ctx.lineWidth = 1.0;

        for (let i = 0; i < particles.length; i++) {
            for (let j = i + 1; j < particles.length; j++) {
                const p1 = particles[i];
                const p2 = particles[j];
                const dx = p1.x - p2.x;
                const dy = p1.y - p2.y;
                const distance = Math.sqrt(dx * dx + dy * dy);

                if (distance < MAX_CONNECT_DISTANCE) {
                    const opacity = Math.min(1, baseLineAlpha + (1 - distance / MAX_CONNECT_DISTANCE) * (1 - baseLineAlpha));
                    const avgY = (p1.y + p2.y) / 2;
                    const t = Math.max(0, Math.min(1, avgY / canvasHeight));
```

```
                const r = Math.round(gradientTopRGB[0] +
(gradientBottomRGB[0] - gradientTopRGB[0]) * t);
                const g = Math.round(gradientTopRGB[1] +
(gradientBottomRGB[1] - gradientTopRGB[1]) * t);
                const b = Math.round(gradientTopRGB[2] +
(gradientBottomRGB[2] - gradientTopRGB[2]) * t);

                ctx.strokeStyle = `rgba(${r}, ${g}, ${b}, ${opacity.toFixed(3)})`;
                ctx.beginPath();
                ctx.moveTo(p1.x, p1.y);
                ctx.lineTo(p2.x, p2.y);
                ctx.stroke();
            }
        }
    }
    ctx.restore();
}

function animate() {
    ctx.fillStyle = bgColorCanvas;
    ctx.fillRect(0, 0, canvasWidth, canvasHeight);

    particles.forEach(p => {
        p.update();
        p.draw();
    });

    drawConnections();
    requestAnimationFrame(animate);
}
```

```
// --- Initialization on Load ---
document.addEventListener('DOMContentLoaded', () => {
    setPlaceholderText();
    updateClock();
    setInterval(updateClock, 1000);

    if (canvas) {
        resizeCanvas();
        animate();
    } else {
        console.warn("Neuron canvas element not found.");
    }

    // Initialize UI states for both file inputs
    handleFileChange();
    if (retrainFileInput) {
        handleRetrainFileChange();
    }
});
```

ДОДАТОК Г

Код бекенду (main.py)

```
import os
import traceback
import pandas as pd
import xgboost as xgb
import joblib
import json
from flask import Flask, request, jsonify, render_template
from sklearn.model_selection import train_test_split, StratifiedKFold, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import (accuracy_score, recall_score, make_scorer,
                             classification_report, roc_auc_score, confusion_matrix)

import optuna

app = Flask(__name__)

# --- Configuration ---
MODEL_DIR = os.path.join(os.path.dirname(os.path.abspath(__file__)), 'model')
MODEL_FILENAME = 'model.bin'
SCALER_FILENAME = 'scaler.joblib'
TOP_FEATURES_FILENAME = 'features.json'
MASTER_TRAINING_DATA_FILENAME = 'master_training_data.csv'

MODEL_PATH = os.path.join(MODEL_DIR, MODEL_FILENAME)
SCALER_PATH = os.path.join(MODEL_DIR, SCALER_FILENAME)
```

```
    TOP_FEATURES_PATH = os.path.join(MODEL_DIR, TOP_FEATURES_FILENAME)
    MASTER_TRAINING_DATA_PATH = os.path.join(MODEL_DIR, MASTER_TRAINING_DATA_FILENAME)

    N_OPTUNA_TRIALS = 25
    N_FEATURES_TO_SELECT = 10

    # --- Global In-Memory Artifacts ---
    xgb_model_instance = None
    scaler_instance = None
    top_feature_names_list = None

    ALL_INPUT_FEATURE_COLUMNS = [
        'radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean', 'smoothness_mean',
        'compactness_mean', 'concavity_mean', 'concave points_mean', 'symmetry_mean',
        'fractal_dimension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'area_se',
        'smoothness_se', 'compactness_se', 'concavity_se', 'concave points_se',
        'symmetry_se', 'fractal_dimension_se', 'radius_worst', 'texture_worst',
        'perimeter_worst', 'area_worst', 'smoothness_worst', 'compactness_worst',
        'concavity_worst', 'concave points_worst', 'symmetry_worst', 'fractal_dimension_worst'
    ]
    RETRAINING_REQUIRED_COLUMNS = ['diagnosis'] + ALL_INPUT_FEATURE_COLUMNS

    def load_global_artifacts():
        # ... (function is unchanged) ...
```

```
    global xgb_model_instance, scaler_instance, top_feature_names_list
    all_loaded_successfully = True
    try:
        xgb_model_instance = xgb.XGBClassifier()
        xgb_model_instance.load_model(MODEL_PATH)
        print(f"--- Model '{MODEL_FILENAME}' loaded successfully ---")
    except Exception as e:
        xgb_model_instance = None; all_loaded_successfully = False
        print(f"--- !!! Error loading model '{MODEL_FILENAME}': {e} !!! ---
")
    try:
        scaler_instance = joblib.load(SCALER_PATH)
        print(f"--- Scaler '{SCALER_FILENAME}' loaded successfully ---")
    except Exception as e:
        scaler_instance = None; all_loaded_successfully = False
        print(f"--- !!! Error loading scaler '{SCALER_FILENAME}': {e} !!! ---
")
    try:
        with open(TOP_FEATURES_PATH, 'r') as f:
            top_feature_names_list = json.load(f)
        print(f"--- Top feature names loaded successfully:
{top_feature_names_list} ---")
    except Exception as e:
        top_feature_names_list = None; all_loaded_successfully = False
        print(f"--- !!! Error loading top features
'{TOP_FEATURES_FILENAME}': {e} !!! ---")
    return all_loaded_successfully

load_global_artifacts()
```

```
@app.route('/')
def index():
    return render_template('index.html')

@app.route('/analyze', methods=['POST'])
def analyze():
    # ... (endpoint is unchanged) ...
    global xgb_model_instance, scaler_instance, top_feature_names_list
    if not all([xgb_model_instance, scaler_instance, top_feature_names_list]):
        missing = []
        if not xgb_model_instance: missing.append("Model")
        if not scaler_instance: missing.append("Scaler")
        if not top_feature_names_list: missing.append("Top feature list")
        return jsonify({"error": f"{', '.join(missing)} not loaded. Check logs."}),
500
    if 'csv_file' not in request.files: return jsonify({"error": "No CSV file."}),
400
    file = request.files['csv_file']
    if file.filename == '': return jsonify({"error": "No selected file."}), 400
    if file and file.filename.endswith('.csv'):
        try:
            data_df = pd.read_csv(file)
            missing_cols = [col for col in ALL_INPUT_FEATURE_COLUMNS
if col not in data_df.columns]
            if missing_cols: return jsonify({"error": f"Missing columns: {',
'.join(missing_cols)}"}), 400
            data_for_scaling = data_df[ALL_INPUT_FEATURE_COLUMNS]
            try: data_for_scaling = data_for_scaling.astype(float)
            except ValueError as ve: return jsonify({"error": f"Data not numeric:
{ve}"}), 400
```

```
                scaled_all_features_np = scaler_instance.transform(data_for_scaling)
                scaled_all_features_df = pd.DataFrame(scaled_all_features_np, columns=ALL_INPUT_FEATURE_COLUMNS)
                try: data_for_model = scaled_all_features_df[top_feature_names_list]
                except KeyError as ke: return jsonify({"error": f"Feature not found after scaling: {ke}"}), 500
                prediction_encoded = xgb_model_instance.predict(data_for_model)
                probability_scores = xgb_model_instance.predict_proba(data_for_model)
                if len(prediction_encoded) > 0:
                    pred_code = prediction_encoded[0]; probas = probability_scores[0]
                    result_text = f"High Risk ({probas[1]*100:.1f}%)." if pred_code == 1 else f"Low Risk ({probas[0]*100:.1f}%)."
                    result_text += " Further diagnosis recommended." if pred_code == 1 else " The patient appears healthy."
                else: result_text = "No data rows in CSV."
                return jsonify({"result": result_text})
            except pd.errors.EmptyDataError: return jsonify({"error": "CSV empty."}), 400
            except Exception as e: app.logger.error(f"Analyze error: {e}\n{traceback.format_exc()}"); return jsonify({"error": f"Server error: {str(e)}"}), 500
        else: return jsonify({"error": "Invalid file type."}), 400


    @app.route('/retrain', methods=['POST'])
    def retrain_model_with_hpo_endpoint():
        global xgb_model_instance, scaler_instance, top_feature_names_list

        app.logger.info("Received request to /retrain with HPO.")
```

```
        if 'retrain_csv_file' not in request.files:
            return jsonify({"error": "No retraining CSV file provided."}), 400
        file = request.files['retrain_csv_file']
        if not (file and file.filename.endswith('.csv')):
            return jsonify({"error": "Invalid file type for retraining. Please upload a CSV."}), 400

        try:
            new_data_df = pd.read_csv(file)
            # --- 1. Validate & Prepare Data ---
            missing_cols = [col for col in RETRAINING_REQUIRED_COLUMNS if col not in new_data_df.columns]
            if missing_cols: return jsonify({"error": f"Retraining CSV missing: {', '.join(missing_cols)}"}), 400
            new_data_df['diagnosis'] = new_data_df['diagnosis'].map({'M': 1, 'B': 0}).fillna(-1)
            if (new_data_df['diagnosis'] == -1).any(): return jsonify({"error": "Invalid 'diagnosis' values."}), 400
            for col in ALL_INPUT_FEATURE_COLUMNS:
                new_data_df[col] = pd.to_numeric(new_data_df[col])

            # --- 2. Update Master Data ---
            if os.path.exists(MASTER_TRAINING_DATA_PATH):
                master_df = pd.read_csv(MASTER_TRAINING_DATA_PATH)
                master_df = pd.concat([master_df, new_data_df[RETRAINING_REQUIRED_COLUMNS]], ignore_index=True)
            else:
                master_df = new_data_df[RETRAINING_REQUIRED_COLUMNS]
            master_df.to_csv(MASTER_TRAINING_DATA_PATH, index=False)
```

```
        app.logger.info(f"Master training data updated. Total records: {len(master_df)}")

        if len(master_df) < 50: # Check for a reasonable amount of data
            return jsonify({"error": f"Not enough data for robust retraining ({len(master_df)} records). Min 50 required."}), 400

        # --- 3. Prepare full dataset ---
        y_full = master_df['diagnosis']
        X_full_unscaled = master_df[ALL_INPUT_FEATURE_COLUMNS]

        # --- 4. Re-fit Scaler ---
        current_scaler = StandardScaler()
        X_full_scaled_np = current_scaler.fit_transform(X_full_unscaled)
        X_full_scaled_df = pd.DataFrame(X_full_scaled_np, columns=ALL_INPUT_FEATURE_COLUMNS)
        joblib.dump(current_scaler, SCALER_PATH)

        # --- 5. Determine Top N Features ---
        model_for_fs = xgb.XGBClassifier(n_estimators=100, random_state=42)
        model_for_fs.fit(X_full_scaled_df, y_full)
        feature_importances_series = pd.Series(model_for_fs.feature_importances_, index=ALL_INPUT_FEATURE_COLUMNS)
        top_n_importances = feature_importances_series.sort_values(ascending=False).head(N_FEATURES_TO_SELECT)
        current_top_features_for_hpo = list(top_n_importances.index)
```

```
            feature_importances_for_json = [{"feature": name, "importance": float(score)} for name, score in top_n_importances.items()]
            X_hpo_data = X_full_scaled_df[current_top_features_for_hpo]

            # --- 6. Optuna HPO Study ---
            app.logger.info(f"Starting Optuna HPO with {N_OPTUNA_TRIALS} trials...")
            def objective(trial):
                params = {
                    'objective': 'binary:logistic', 'eval_metric': 'logloss',
                    'n_estimators': trial.suggest_int('n_estimators', 50, 300, step=25),
                    'learning_rate': trial.suggest_float('learning_rate', 0.01, 0.3, log=True),
                    'max_depth': trial.suggest_int('max_depth', 3, 9),
                    'subsample': trial.suggest_float('subsample', 0.5, 1.0),
                    'colsample_bytree': trial.suggest_float('colsample_bytree', 0.5, 1.0),
                    'gamma': trial.suggest_float('gamma', 0.0, 0.5),
                }
                model = xgb.XGBClassifier(**params)
                cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
                # Use roc_auc for optimization as it's a good overall metric
                score = cross_val_score(model, X_hpo_data, y_full, cv=cv, scoring='roc_auc').mean()
                return score

            study = optuna.create_study(direction='maximize')
            study.optimize(objective, n_trials=N_OPTUNA_TRIALS, timeout=300)
            best_hpo_params = study.best_trial.params
```

```
        app.logger.info(f"Optuna HPO complete. Best ROC AUC in CV: {study.best_trial.value:.4f}")

        # --- 7. Train Final Model ---
        X_final_train, X_final_test, y_final_train, y_final_test = train_test_split(
            X_hpo_data, y_full, test_size=0.25, stratify=y_full
        )
        final_xgb_model = xgb.XGBClassifier(**best_hpo_params)
        final_xgb_model.fit(X_final_train, y_final_train)
        final_xgb_model.save_model(MODEL_PATH)
        with open(TOP_FEATURES_PATH, 'w') as f:
            json.dump(current_top_features_for_hpo, f)

        # --- 8. Evaluate final model and generate detailed reports ---
        y_final_pred = final_xgb_model.predict(X_final_test)
        y_pred_proba = final_xgb_model.predict_proba(X_final_test)[:, 1]

        final_accuracy = accuracy_score(y_final_test, y_final_pred)
        report_dict = classification_report(y_final_test, y_final_pred, target_names=['Benign (0)', 'Malignant (1)'], output_dict=True)
        cm = confusion_matrix(y_final_test, y_final_pred)
        cm_list = cm.tolist()
        roc_auc = roc_auc_score(y_final_test, y_pred_proba)

        app.logger.info(f"Final Model - Accuracy: {final_accuracy:.4f}, ROC AUC: {roc_auc:.4f}")
        app.logger.info(f"Classification Report:\n{classification_report(y_final_test, y_final_pred, target_names=['Benign (0)', 'Malignant (1)'])}")
        app.logger.info(f"Confusion Matrix:\n{cm}")
```

```
            # --- 9. Reload artifacts & Return response ---
            if not load_global_artifacts():
                return jsonify({"error": "Retraining completed, but failed to reload artifacts."}), 500

            # --- Return all metrics in the JSON response ---
            return jsonify({
                "message": "Model successfully retrained with Hyperparameter Optimization.",
                "final_model_accuracy": f"{final_accuracy:.4f}",
                "roc_auc_score": f"{roc_auc:.4f}",
                "classification_report": report_dict,
                "confusion_matrix": cm_list,
                "top_feature_importances": feature_importances_for_json
            })

        except Exception as e:
            app.logger.error(f"Error during HPO retraining: {e}\n{traceback.format_exc()}")
            return jsonify({"error": f"An error occurred during retraining: {str(e)}"}), 500

    if __name__ == '__main__':
        if not os.path.exists(MODEL_DIR): os.makedirs(MODEL_DIR)
        app.run(host='0.0.0.0', port=int(os.environ.get("FLASK_RUN_PORT", 5000)), debug=True)
```

ДОДАТОК Д

Код до контейнеризації (Dockerfile)

```
FROM python:3.9-slim

COPY . .
RUN pip install --no-cache-dir -r requirements.txt

WORKDIR /app

EXPOSE 80

ENV FLASK_APP=main.py
ENV FLASK_RUN_HOST=0.0.0.0
ENV FLASK_RUN_PORT=80

CMD ["flask", "run"]
```

ДОДАТОК Е

Код до контейнеризації (docker-compose.yml)

```
---
version: '3.8'
services:
  web:
    build: .
    ports:
      - "5000:80"
    volumes:
      - ./app:/app
```