Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра програмної інженерії
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня
Магістр

(назва освітнього ступеня)

на тему: **Розробка інтелектуальної метаансамблевої моделі класифікації медичних даних для прогнозування інсульту на мові програмування Python**

Виконав(ла): студент(ка) 6 курсу, групи СПм-61
спеціальності 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

(підпис) Кокайло В.В. (прізвище та ініціали)

Керівник (підпис) Багрій-Заяць О.А. (прізвище та ініціали)
Нормоконтроль (підпис) Стоянов Ю.М. (прізвище та ініціали)
Завідувач кафедри (підпис) Петрик М.Р. (прізвище та ініціали)
Рецензент (підпис) Гром'як Р.С. (прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ _____
(підпис) (прізвище та ініціали)
« » 20___ р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістра
(назва освітнього ступеня)

за спеціальністю 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

студенту Кокайло Вікторії Василівні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інтелектуальної метаансамблевої моделі класифікації медичних даних для прогнозування інсульту на мові програмування Python

Керівник роботи Багрій-Заяць Оксана Андріївна, канд. техн. наук, доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «___» _____ 20___ року № _____

2. Термін подання студентом завершеної роботи _____

3. Вихідні дані до роботи Предметна область, завдання, вимоги та специфікація, програмне рішення, методичні вказівки

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступна частина
Аналіз предметної області та теоретичних основ
Визначення методики реалізації моделі
Реалізація моделі
Визначення основних аспектів охорони праці та безпеки життєдіяльності
Висновки роботи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Слайди презентації та діаграми процесів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності,	Стручок В. С. ст. викл.		
основи охорони праці	Осухівська Г. М. зав. каф.		
Нормоконтроль	Стоянов Ю.М. к.т.н., ст. викл.		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання		
2	Аналіз завдання		
3	Виконання розділу 1		
4	Виконання розділу 2		
5	Виконання розділу 3		
6	Оформлення пояснювальної записки		
7	Оформлення графічного та презентаційного матеріалу		
8	Попередній захист		
9	Захист		

Студент _____ Кокайло В.В
(підпис) (прізвище та ініціали)

Керівник роботи _____ Багрій-Заяць О.А.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота магістра, виконала Кокайло Вікторія Василівна, студентка групи СПм-61 Тернопільського національного технічного університету імені Івана Пулюя, на тему «Розробка інтелектуальної метаансамблевої моделі класифікації медичних даних для прогнозування інсульту на мові програмування Python». Робота має обсяг 61 сторінок, включає 16 рисунків, 7 таблиці, 3 додатків та бібліографію з 25 джерел.

Метою роботи є створення програмної системи для підвищення точності прогнозування медичних ризиків шляхом реалізації мета метаансамблевого підходів, які об’єднують різні алгоритми класифікації, такі як Random Forest, XGBoost, LightGBM та CatBoost. Запропонований підхід дозволяє підвищити узагальнювальну здатність моделей, зменшити похибку класифікації та підвищити стійкість результатів при роботі з реальними даними.

У роботі реалізовано архітектуру трирівневої системи, що включає модулі збору та підготовки даних, навчання ансамблів і метамоделі (stacking), а також блок оцінювання якості класифікації за метриками Accuracy, Recall, Precision, F1-score та ROC-AUC. Проведено порівняльний аналіз ефективності базових і ансамблевих моделей, результати якого підтверджують перевагу метаасемблевого підходу у стабільності та точності прогнозування.

Розроблена система може бути адаптована для вирішення інших прикладних задач класифікації в межах інженерії програмного забезпечення, зокрема у сферах медицини, фінансів та промислової аналітики. Робота демонструє практичне застосування методів машинного навчання підкреслює важливість метаасемблевих технологій у підвищенні ефективності класифікаційних процесів.

Ключові слова роботи: класифікація даних, ансамблеві алгоритми, машинне навчання, ансамблеве навчання, прогнозування інсульту, Python, XGBoost, Random Forest.

ABSTRACT

Master’s qualification thesis, completed by Kokailo Viktoriia, a student of group SPm-61 at Ternopil Ivan Puluj National Technical University, is devoted to «Development of an Intelligent Meta-Ensemble Model for Medical Data Classification to Predict Stroke Using the Python Programming Language». The thesis comprises 61 pages, includes 16 figures, 3 appendices, a bibliography of 25 sources.

The aim of the thesis is to develop a software system for improving the accuracy of medical risk prediction through the implementation of meta-ensemble approaches that combine various classification algorithms, such as Random Forest, XGBoost, LightGBM, and CatBoost. The proposed approach enhances the generalization ability of the models, reduces classification error, and increases the robustness of results when working with real-world data.

The thesis implements a three-level system architecture that includes modules for data collection and preprocessing, ensemble and meta-model (stacking) training, as well as a performance evaluation block based on the metrics Accuracy, Recall, Precision, F1-score, and ROC-AUC. A comparative analysis of the effectiveness of base and ensemble models was conducted, the results of which confirm the superiority of the meta-ensemble approach in terms of prediction stability and accuracy.

The developed system can be adapted to solve other applied classification problems within the field of software engineering, in particular in medicine, finance, and industrial analytics. The thesis demonstrates the practical application of machine learning methods and emphasizes the importance of meta-ensemble technologies in improving the efficiency of classification processes.

Keywords: data classification, meta-ensemble algorithms, machine learning, ensemble learning, stroke prediction, Python, XGBoost, Random Forest.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ML – Machine Learning (машинне навчання)

AI – Artificial Intelligence (штучний інтелект)

DL – Deep Learning (глибинне навчання)

EDA – Exploratory Data Analysis (розвідувальний аналіз даних)

CSV – Comma-Separated Values (текстовий формат даних)

LR – Logistic Regression (логістична регресія)

DT – Decision Tree (дерево рішень)

kNN – k-Nearest Neighbors (метод k найближчих сусідів)

SVM – Support Vector Machine (метод опорних векторів)

NB – Naive Bayes (наївний баєсівський класифікатор)

NN – Neural Network (нейронна мережа)

RF – Random Forest (випадковий ліс)

GBM – Gradient Boosting Machine (градієнтний бустинг)

XGB – XGBoost (Extreme Gradient Boosting)

LGBM – LightGBM (Light Gradient Boosting Machine)

CB – CatBoost (Categorical Boosting)

SMOTE – Synthetic Minority Oversampling Technique (синтетичне збільшення вибірки меншості)

ROC – Receiver Operating Characteristic (крива робочих характеристик приймача)

AUC – Area Under Curve (площа під кривою)

API – Application Programming Interface (інтерфейс прикладного програмування)

CPU – Central Processing Unit (центральний процесор)

RAM – Random Access Memory (оперативна пам’ять)

JSON – JavaScript Object Notation (формат даних JSON)

ВСТУП

У сучасній галузі медичних інформаційних систем, де точність і швидкість прийняття рішень визначають ефективність діагностики, застосування методів машинного навчання стає невід'ємною складовою розробки програмних рішень. Медичні дані характеризуються високою неоднорідністю, наявністю пропусків, змішаними типами ознак та суттєвим дисбалансом класів, що ускладнює використання традиційних алгоритмів. Саме тому постає потреба у застосуванні більш гнучких і надійних моделей, здатних узгоджувати сигнали від різних класифікаторів і формувати стабільні прогнози.

У задачі прогнозування інсульту важливим є не лише визначення загальної точності моделі, але й забезпечення високих значень F1-score та Recall, оскільки пропуск випадку потенційної небезпеки може мати критичні наслідки. Використання метаансамблевого підходу дозволяє значно підвищити якість класифікації порівняно з окремими моделями. Завдяки поєднанню прогнозів таких алгоритмів, як Random Forest, XGBoost, LightGBM, CatBoost та Logistic Regression, система набуває властивостей узагальнення, яких не має жодна окрема модель. Залучення сучасних технологій штучного інтелекту та оптимізації гіперпараметрів забезпечує можливість адаптації моделі до специфіки медичного датасету. Крім того, метаансамбль дозволяє мінімізувати вплив помилок окремих алгоритмів і забезпечити стійкий результат навіть за умов високого дисбалансу даних. Такий підхід сприяє формуванню більш надійних рекомендацій для клінічних систем підтримки прийняття рішень.

Використання метаансамблевої архітектури також скорочує часові та ресурсні витрати на розробку складних моделей, оскільки дозволяє гнучко комбінувати вже існуючі алгоритми та підвищувати їхню ефективність без створення повністю нової моделі з нуля. Це робить даний підхід особливо цінним для впровадження у сучасні медичні програмні платформи, які потребують високу точність, масштабованість та можливість інтеграції з іншими компонентами інформаційної інфраструктури.

Завдяки можливостям сучасних алгоритмів машинного навчання, системи прогнозування медичних ризиків можуть постійно вдосконалюватися, оновлюючи моделі з урахуванням нових даних, змін у демографічних характеристиках населення та появи нових факторів ризику. Такі системи забезпечують динамічний та адаптивний підхід до медичної аналітики, підвищують точність прогнозування та сприяють прийняттю більш обґрунтованих клінічних рішень.

Основною метою даного дослідження є розробка та експериментальна перевірка метаансамблевої моделі, здатної підвищити точність класифікації у порівнянні з окремими базовими алгоритмами. Особливу увагу приділено аналізу їхньої взаємодоповнюваності та визначенню оптимальних стратегій узгодження прогнозів. Це включає оцінку ефективності роботи моделей першого рівня, дослідження впливу гіперпараметрів та формування метатренувального набору, що є критично важливим для правильної роботи метамоделі.

Експериментальна частина передбачала як навчання та оцінку окремих моделей, так і побудову метаансамблю із застосуванням алгоритмів Logistic Regression, RidgeClassifier та XGBoost як метамоделей. Отримані результати демонструють, що метаансамблевий підхід суттєво підвищує F1-score та ROC-AUC, що є ключовими метриками для медичних задач з дисбалансом класів. Це підтверджує, що узгодження прогнозів базових класифікаторів дозволяє отримати більш точну й узагальнену модель, ніж використання будь-якого окремого алгоритму.

Завершальним етапом дослідження є формування рекомендацій щодо вибору оптимальної архітектури для задач прогнозування медичних ризиків, а також окреслення напрямів подальшого вдосконалення моделі, включно з адаптацією під потокові дані, застосуванням глибинних нейронних мереж або розширенням набору ознак. Отримані результати можуть бути використані як підґрунтя для створення програмних систем підтримки медичних рішень і подальшого розвитку інтелектуальних медичних технологій.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ТЕОРЕТИЧНІ ОСНОВИ АСАМБЛЕВИХ АЛГОРИТМІВ

Класифікація є однією з центральних задач інтелектуального аналізу даних, спрямованою на виявлення закономірностей у наборі спостережень і віднесення нових об’єктів до одного з наперед визначених класів. У загальному вигляді це процес навчання моделі, яка на основі відомих прикладів (вхідних даних із мітками класів) здатна робити узагальнення і приймати рішення щодо належності нових, невідомих об’єктів.

1.1 Загальна характеристика задачі класифікації даних

З математичної точки зору задача класифікації визначається як знаходження відображення:

$$f = X \to Y, \tag{1.1}$$

де

X — це простір вхідних ознак (атрибутів), а Y — множина можливих класів або категорій. Метою є побудова такої функції f, яка забезпечує мінімальну кількість помилок при класифікації нових даних.

У контексті інженерії програмного забезпечення класифікація розглядається не лише як алгоритмічна задача, а як комплексна система, що включає етапи збору, очищення, попередньої обробки, моделювання, оцінки та валідації даних. Створення програмного забезпечення для класифікації вимагає реалізації повного циклу обробки даних, побудови архітектури, здатної масштабуватись, а також можливості подальшого узгодження результатів і інтеграції з іншими аналітичними модулями. Класифікаційні алгоритми відіграють ключову роль у розв’язанні практичних задач різних галузей від розпізнавання зображень і аналізу текстів до прогнозування фінансових ризиків чи стану здоров’я. Усі ці системи спільні в одному вони потребують точних, узагальнюючих моделей, які здатні

працювати з великими обсягами даних і зберігати стабільність результатів при зміні умов.

Для реалізації задач класифікації зазвичай використовуються алгоритми машинного навчання, що базуються на різних підходах — статистичних, евристичних, деревоподібних або нейронних. Кожен з них має свої особливості:

- Дерева рішень дозволяють легко інтерпретувати процес прийняття рішення.
- Методи опорних векторів (SVM) добре працюють на високовимірних просторах.
- Нейронні мережі здатні моделювати складні нелінійні залежності.
- Ансамблеві методи поєднують кілька моделей для досягнення кращої узагальнювальної здатності.

Таким чином, сутність задачі класифікації полягає у побудові інтелектуальної системи, яка може виявляти приховані закономірності в даних, навчатися на прикладах і робити точні передбачення. Саме тому класифікація виступає основою багатьох програмних систем, орієнтованих на прийняття рішень, аналітику та прогнозування.

Машинне навчання є ключовою технологією, що забезпечує здатність сучасних програмних систем до самонавчання, адаптації та прогнозування на основі накопичених даних. Його головна мета полягає у створенні моделей, які можуть узагальнювати закономірності з історичних спостережень і робити точні передбачення для нових, невідомих прикладів. На відміну від традиційного програмування, де логіка визначається вручну, у машинному навчанні модель самостійно формує правила на основі аналізу даних, що підвищує гнучкість і автономність системи. У системах прогнозування машинне навчання дозволяє автоматизувати процес прийняття рішень, оцінюючи ймовірність певних подій чи станів. Наприклад, у медицині воно використовується для прогнозування розвитку захворювань, у фінансах для оцінки кредитного ризику, а в промисловості для передбачення збоїв обладнання. Незалежно від сфери застосування, принцип дії таких систем однаковий: модель вивчає статистичні взаємозв’язки між вхідними

змінними (ознаками) та цільовим результатом, а потім узагальнює їх для нових випадків.

Розвиток машинного навчання став можливим завдяки зростанню обсягів даних і обчислювальних потужностей, що дало змогу застосовувати складні алгоритми на великих вибірках. Сьогодні розрізняють три основні типи навчання:

- Навчання з учителем (Supervised Learning) — використовується, коли відомі правильні результати (мітки класів або числові значення). Саме цей підхід лежить в основі задачі класифікації.
- Навчання без учителя (Unsupervised Learning) — застосовується для виявлення структури даних без заздалегідь відомих міток.
- Навчання з підкріпленням (Reinforcement Learning) — орієнтоване на поступове вдосконалення рішень через отримання винагороди або штрафу.

У системах прогнозування, подібних до тих, що використовуються для оцінки ризику інсульту, ключовим етапом є вибір адекватної моделі навчання. Важливо забезпечити не лише високу точність передбачення, але й інтерпретованість результатів, оскільки в реальних задачах рішення повинні бути зрозумілими для користувача або експерта. Тому поряд із глибокими нейронними мережами все ще широко застосовуються класичні алгоритми — дерева рішень, логістична регресія, Random Forest, Gradient Boosting, Support Vector Machine тощо.

У програмній реалізації такі системи зазвичай мають модульну структуру: окремі компоненти відповідають за завантаження й очищення даних, вибір ознак, навчання моделей, оцінку точності та візуалізацію результатів. Використання бібліотек scikit-learn, pandas, NumPy та matplotlib дозволяє створювати відтворювані інженерні рішення, що поєднують статистичний аналіз і алгоритмічну точність. Роль машинного навчання у системах прогнозування полягає у тому, щоб перетворювати великі масиви сирих даних у практично корисні висновки. Воно виступає ядром інтелектуальної компоненти програмного забезпечення, забезпечуючи здатність до адаптації, оптимізації та автоматизованого прийняття рішень. У межах даної роботи машинне навчання є

основою побудови метаансамблевої системи, здатної підвищувати точність і стабільність прогнозів за рахунок поєднання кількох моделей класифікації.

1.2 Задача прогнозування інсульту на основі класифікації

Задача прогнозування інсульту може бути розглянута як типовий приклад задачі бінарної класифікації, де необхідно визначити, чи належить конкретний пацієнт до групи ризику (клас 1) або ні (клас 0). Такий підхід дозволяє використовувати відомі методи машинного навчання, які навчаються на історичних медичних даних і прогнозують настання події на основі набору вхідних характеристик.

Для виконання дослідження використовується відкритий набір даних Stroke Prediction Dataset, опублікований на платформі Kaggle [1]. Цей датасет містить понад 5000 записів, кожен з яких представляє інформацію про одну особу з різними соціальними, фізіологічними та медичними показниками. Структуру даних наведено у таблиці 1.1, де описано основні атрибути, що впливають на прогнозування.

Таблиця 1.1 - Основні атрибути набору даних для прогнозування інсульту

Назва ознаки	**Тип даних**	**Опис**
gender	категоріальний	Стать особи (Male, Female, Other)
age	числовий	Вік у роках
hypertension	бінарний	Наявність гіпертонії (1 – так, 0 – ні)
heart_disease	бінарний	Наявність серцевих захворювань (1 – так, 0 – ні)
ever_married	категоріальний	Сімейний стан
work_type	категоріальний	Тип роботи (Private, Self-employed, Govt_job, Children)
Residence_type	категоріальний	Тип місця проживання (Urban, Rural)
avg_glucose_level	числовий	Середній рівень глюкози у крові
bmi	числовий	Індекс маси тіла
smoking_status	категоріальний	Стан куріння (never, formerly, smokes, unknown)
stroke	бінарний (ціль)	Цільова змінна: факт настання інсульту (1 – так, 0 – ні)

Перший етап розв’язання задачі передбачає попередню обробку даних, що включає нормування числових показників, перетворення категоріальних змінних у числові (One-Hot Encoding), а також виявлення та заповнення пропущених значень (рисунок 1.1). Такі дії необхідні, оскільки навіть невеликі порушення у структурі даних можуть суттєво вплинути на стабільність навчання моделей [2].

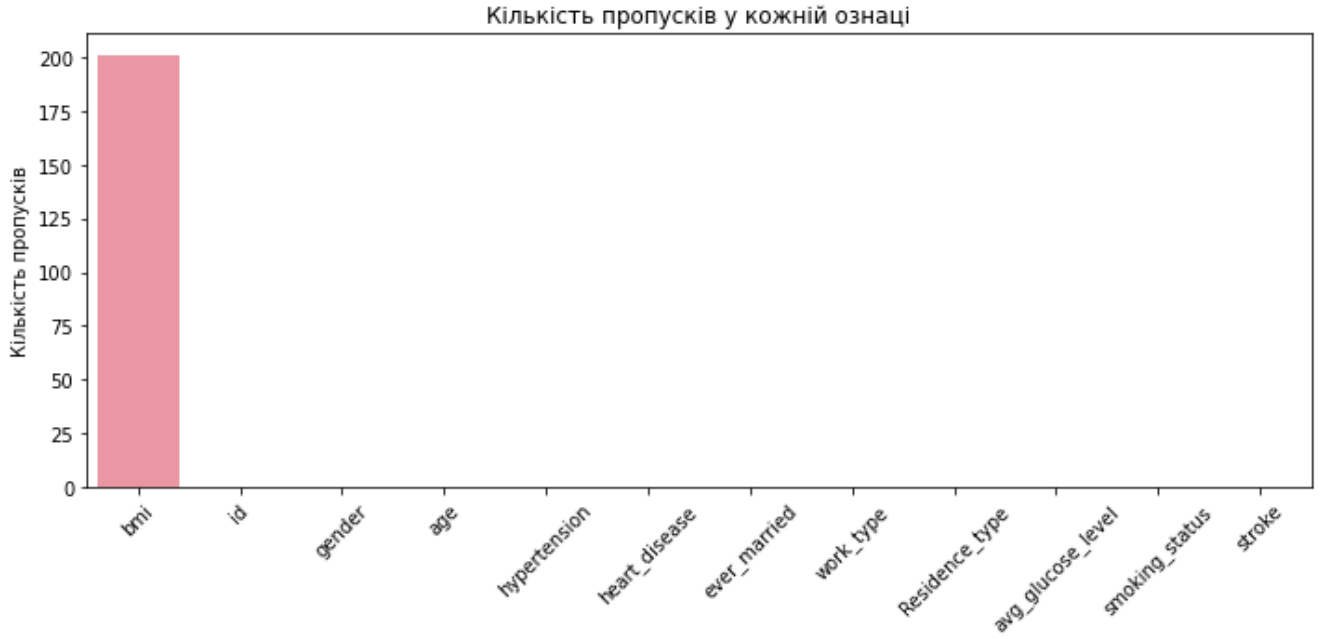


Рисунок 1.1 – Візуалізації пропусків у даних перед очищенням відносно кожної з характеристик вхідного набору даних

Після попередньої обробки дані розділяються на навчальну та тестову вибірки (наприклад, у співвідношенні 80:20). Це дозволяє оцінити здатність моделей узагальнювати закономірності, а не лише запам’ятовувати приклади. Далі застосовуються базові алгоритми класифікації — Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, SVM, які надалі стануть складовими метаасемблевої системи.

Особливістю цього набору даних є дисбаланс класів: частка пацієнтів з інсультом становить лише близько 5% усіх спостережень. Це створює виклик для моделей, оскільки вони можуть схилятися до передбачення переважного класу (0 – без інсульту). Для подолання цієї проблеми використовуються техніки

балансування — Oversampling, SMOTE або підбір вагових коефіцієнтів для класів. Ілюстрація розподілу цільової змінної наведена на рисунку 1.2.

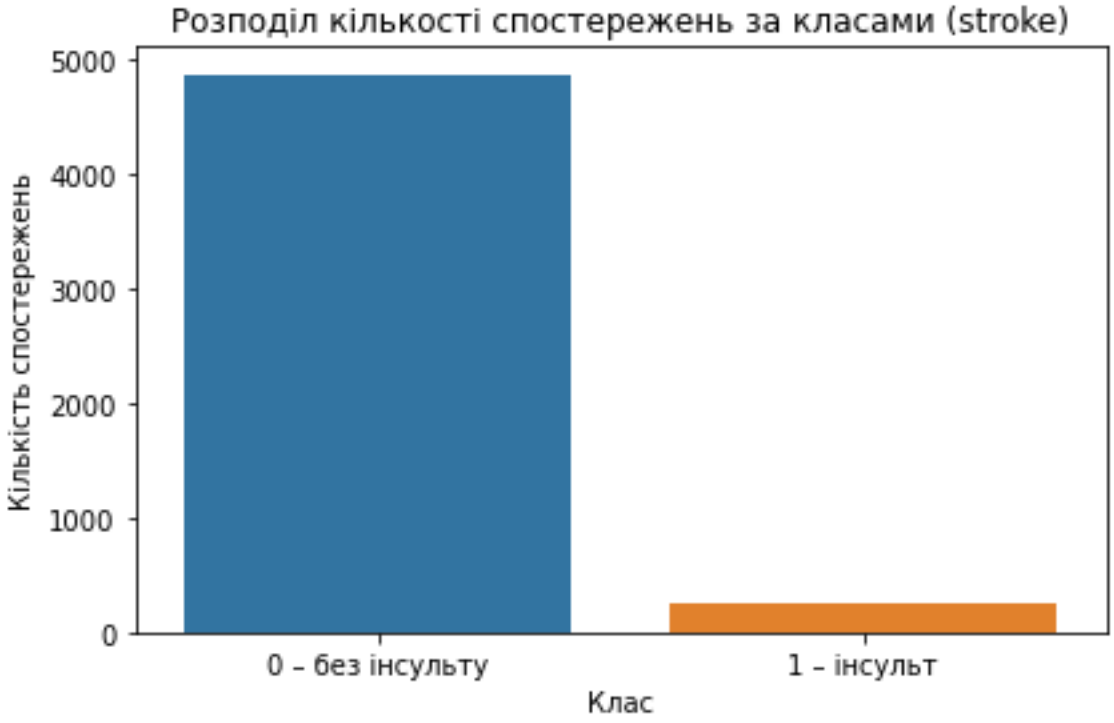


Рисунок 1.2 – Розподіл кількості прикладів клінічних даних «без інсульту» та «інсульт»

Таким чином, задача прогнозування інсульту формально визначається як: знайти модель $f(x)$, яка на основі вектора ознак $x = [x_1, x_2, \ldots, x_n]$ оцінює ймовірність $P(y) = 1|x$, де y – цільова змінна, що вказує на факт інсульту.

Постановка задачі узгоджується з типовими процесами побудови класифікаційних систем у сфері інженерії програмного забезпечення — від структурування даних і розробки модульної архітектури до оцінювання точності моделей за метриками Accuracy, Precision, Recall, F1-score, ROC-AUC. Надалі ці базові моделі будуть узгоджені в межах метаансамблевого підходу, що дозволить покращити точність прогнозування та стабільність результатів при тестуванні на реальних вибірках [3].

1.3 Характеристика вибраного набору даних

Для реалізації системи прогнозування використовується Stroke Prediction Dataset, опублікований на платформі Kaggle. Цей набір даних був створений для моделювання ризику інсульту на основі демографічних, поведінкових та медичних характеристик пацієнтів. Його структура добре підходить для дослідження методів класифікації, оскільки містить різнорідні типи ознак — числові, категоріальні та бінарні [1].

1.3.1 Загальна характеристика набору даних

Набір містить 5110 записів (спостережень) та 12 атрибутів, з яких одинадцять є вхідними ознаками, а одна — цільовою змінною (stroke). Кількість атрибутів і їхні типи наведено у таблиці 1.1.

1.3.2 Аналіз розподілу даних

Перед початком моделювання проводиться попередня оцінка даних для виявлення дисбалансу класів, викидів та пропущених значень. На рисунку 1.3 показано розподіл цільової змінної — лише близько 4,9% записів належать до класу 1 (інсульт), що підтверджує наявність сильної асиметрії у вибірці.

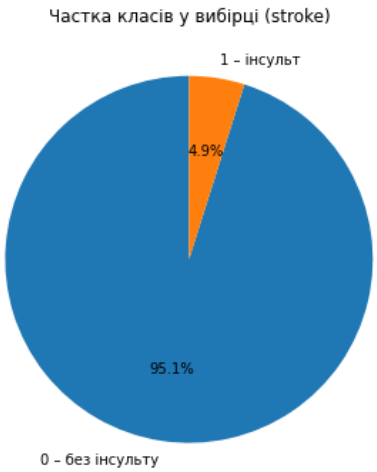


Рисунок 1.3 – Розподіл кількості спостережень за класами (stroke = 0 та stroke = 1)

Така нерівномірність вимагає спеціальних методів обробки, зокрема балансування вибірки за допомогою технік SMOTE (Synthetic Minority Oversampling Technique) або Random Oversampling. Ці методи штучно збільшують кількість прикладів менш представленого класу, не змінюючи структуру даних.

1.3.3 Пропущені значення та очищення даних

Однією з особливостей набору є наявність пропущених значень у полі bmi, що відображено у вигляді теплової карти (рисунок 1.4). Пропуски зустрічаються у близько 200 записах, тому вони не можуть бути проігноровані. Залежно від моделі обробки даних можливі два підходи:

- Імпутація середнім або медіанним значенням – простий, але стабільний метод для числових полів.
- Моделювання на основі кореляцій – заповнення пропусків за допомогою регресійної оцінки, враховуючи пов’язані змінні, наприклад, age та avg_glucose_level.

1.3.4 Кореляційні зв’язки між ознаками

Для виявлення залежностей між числовими ознаками обчислюється матриця кореляцій Пірсона, наведена на рисунку 1.4. Аналіз показує, що існує помірний позитивний зв’язок між age та stroke (коефіцієнт ≈ 0.24), що підтверджує відому закономірність: ризик інсульту зростає з віком. Також спостерігається слабка кореляція між avg_glucose_level і stroke, що свідчить про роль метаболічних факторів у прогнозуванні [4].

Такі візуалізації дозволяють не лише краще зрозуміти структуру даних, а й відібрати найбільш релевантні ознаки для моделювання, мінімізуючи ризик перенавчання.

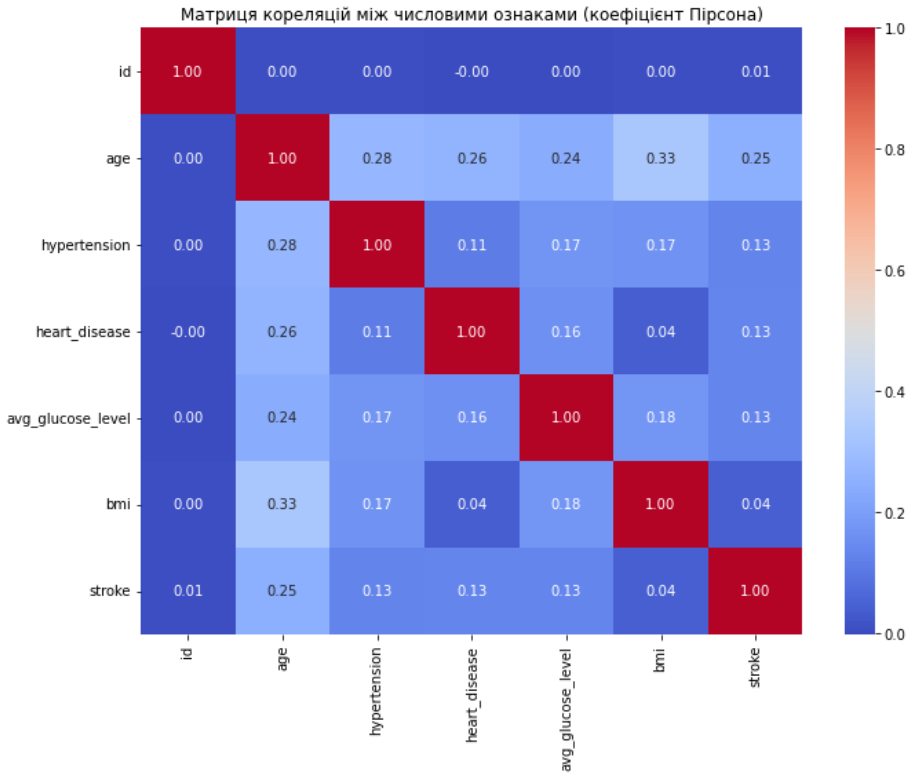


Рисунок 1.4 – Матриця кореляцій між числовими ознаками (heatmap за коефіцієнтами Пірсона)

1.4 Методи класифікації, що використовуються у системах прогнозування

Сучасні системи прогнозування базуються на алгоритмах машинного навчання, які реалізують різні підходи до виявлення закономірностей у даних. Вибір методу класифікації суттєво впливає на точність, швидкодію та узагальнювальну здатність моделі. Для задачі прогнозування інсульту доцільно використовувати як базові алгоритми, так і ансамблеві підходи, які поєднують кілька моделей для підвищення ефективності [5].

1.4.1 Класифікаційні алгоритми базового рівня

Базові методи класифікації виступають основою для створення складніших систем. У таблиці 1.2 наведено короткий опис найпоширеніших алгоритмів, які будуть використані для початкового етапу моделювання.

Таблиця 1.2 – Основні алгоритми класифікації та їх характеристики

Алгоритм	Тип підходу	Основна ідея
Logistic Regression (LR)	Лінійна модель	Оцінює ймовірність належності об’єкта до класу за допомогою сигмоїдної функції
Decision Tree (DT)	Деревоподібна модель	Розділяє простір ознак за пороговими значеннями
k-Nearest Neighbors (kNN)	Евристичний	Класифікує за найближчими сусідами у просторі ознак
Support Vector Machine (SVM)	Геометричний	Будує гіперплощину, що розділяє класи з максимальним відступом
Naive Bayes (NB)	Імовірнісний	Використовує теорему Байєса з припущенням незалежності ознак
Neural Network (NN)	Нелінійний	Моделює складні залежності через шари нейронів

Для задачі прогнозування інсульту важливо враховувати не лише точність, але й стабільність моделі при роботі з дисбалансом класів. У таких випадках Decision Tree або Random Forest часто демонструють найкращу узагальнювальну здатність, оскільки не потребують масштабування даних та природно обробляють категоріальні змінні.

1.4.2 Ансамблеві методи як основа підвищення точності

Ансамблеві методи поєднують кілька базових моделей для досягнення кращої продуктивності. Основна ідея полягає в тому, що група “слабких” класифікаторів може створити “сильну” модель, якщо правильно об’єднати їхні результати (рисунок 1.5). Існує три основні типи ансамблів:

- Bagging (Bootstrap Aggregating) — незалежне навчання кількох моделей на різних підвибірках даних. Приклад — Random Forest, який об’єднує велику кількість дерев рішень.
- Boosting — послідовне навчання моделей, де кожна наступна модель коригує помилки попередньої. Найпоширеніші реалізації — XGBoost, LightGBM, CatBoost.
- Stacking — багаторівневе об’єднання різних алгоритмів, де метамодель навчається на виходах базових моделей, формуючи підсумковий прогноз.

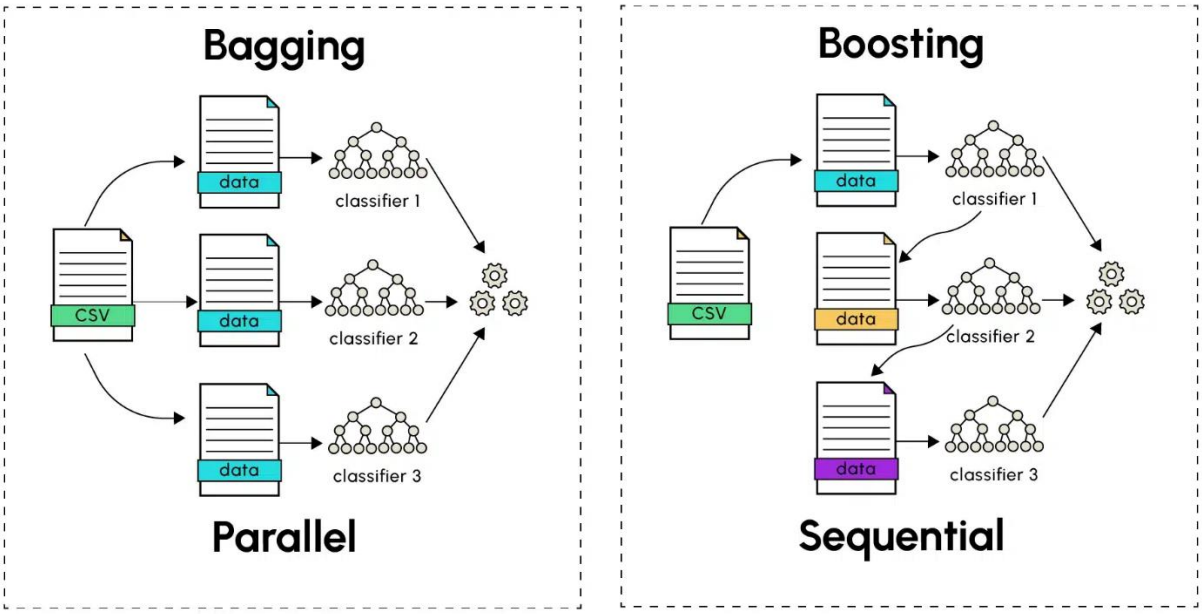


Рисунок 1.5 – Основні підходи ансамблевого навчання: bagging, boosting,

Ансамблеві методи довели свою ефективність у задачах прогнозування медичних станів, де спостерігається значна варіація даних. У випадку Stroke Prediction Dataset, поєднання моделей, таких як Decision Tree + Logistic Regression + Gradient Boosting, дозволяє зменшити дисперсію та підвищити точність класифікації (рисунок 1.6).

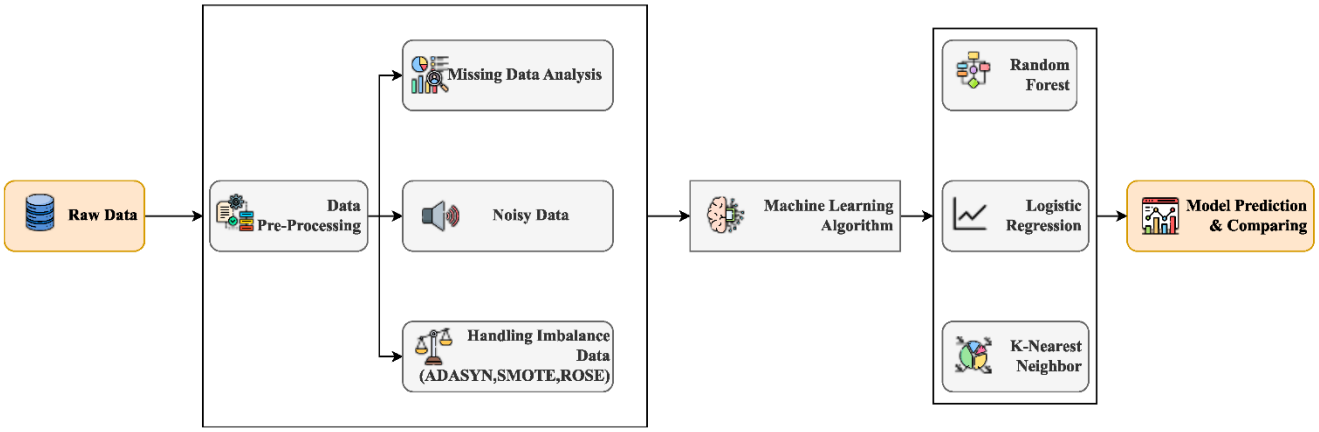


Рисунок 1.6 – Основні підходи метамодельного навчання Stroke Prediction Datase

1.5 Архітектура метаансамблевого підходу

Ансамблеві та метаансамблеві методи стали одним із ключових напрямів розвитку сучасного машинного навчання, особливо у задачах класифікації, де важливо досягти не лише високої точності, а й стабільності результатів. Основна ідея ансамблевого підходу полягає у комбінації кількох базових моделей, які спільно приймають рішення, компенсуючи помилки одна одної. Такий підхід базується на статистичному принципі: колективна оцінка незалежних моделей є більш надійною, ніж окреме передбачення будь-якої з них [6].

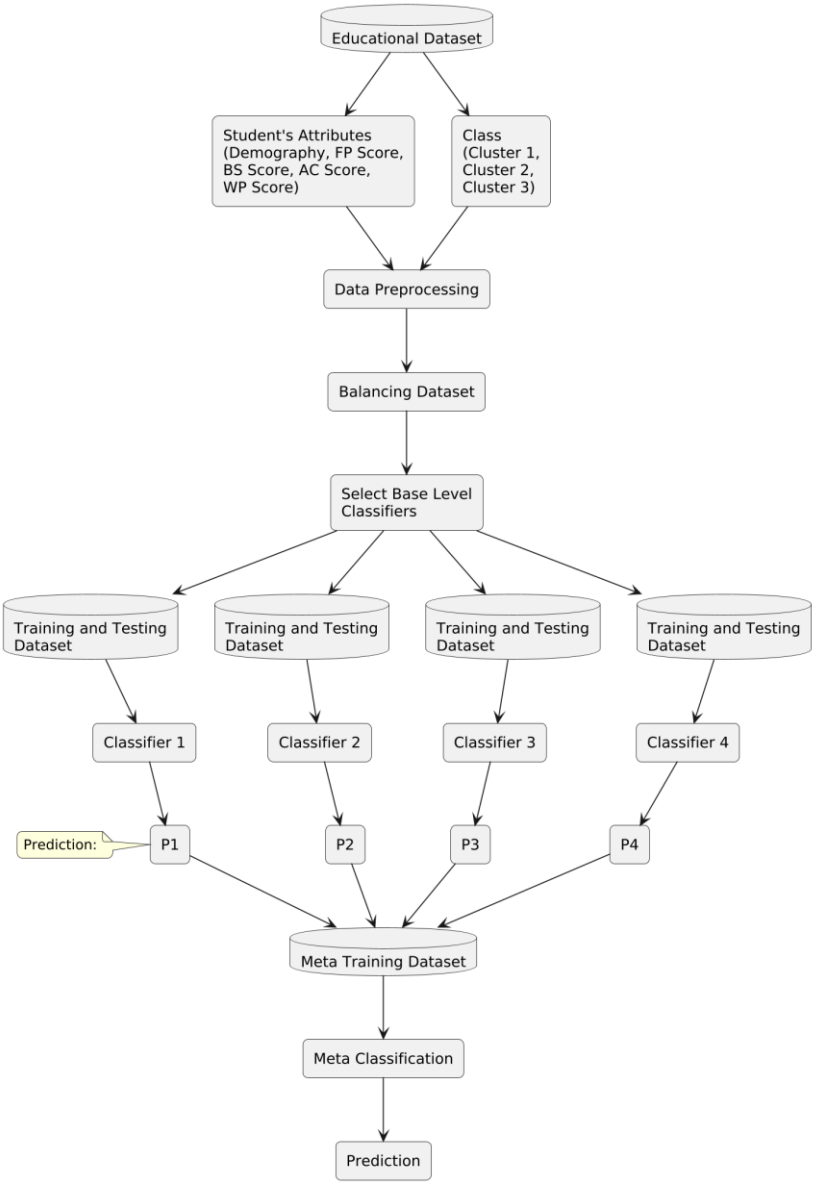


Рисунок 1.7 – Архітектура метаансамблевого підходу

На рисунку 1.7 подано узагальнену архітектуру метаансамблевого підходу - stacking, використаного під час розробки моделі прогнозування. Діаграма відображає два основні рівні обробки даних — базовий ансамблевий рівень та метарівень узгодження результатів, що є характерними для метаансамблевих систем машинного навчання.

На початковому етапі вхідний набір даних проходить процедури попередньої обробки, що включають очищення, нормалізацію ознак, кодування категоріальних значень та усунення пропусків. Після цього формується збалансований датасет шляхом застосування відповідних технік балансування класів. Отриманий набір даних використовується для побудови тренувальної та тестової вибірок, що забезпечує коректне навчання моделей і подальшу незалежну оцінку їхньої якості.

На базовому рівні здійснюється навчання одного або кількох класифікаторів, кожен із яких отримує копію підготовленого датасету. Ці моделі можуть належати до різних алгоритмічних підходів, що дозволяє забезпечити різноманітність у структурі рішень та зменшити ризик узагальненої помилки. Після навчання кожен класифікатор генерує власний прогноз P_i, який репрезентує або ймовірнісну оцінку, або бінарний результат класифікації [7].

Отримані прогнозні значення формують метатренувальний датасет, що слугує вхідними ознаками для моделі метарівня. На метарівні виконується навчання окремої моделі, так званої meta-learner, головною функцією якої є узгодження результатів базових класифікаторів та побудова оптимального фінального прогнозу. Таким чином, метамодель навчається оцінювати внесок кожного базового класифікатора та комбінувати їх результати таким чином, щоб мінімізувати загальну помилку класифікації.

Процес завершується формуванням остаточного прогнозу, який базується на висновках метамоделі, що дозволяє підвищити точність, стійкість і узагальнювальну здатність системи порівняно з використанням окремих моделей. Представлена архітектура демонструє логічну послідовність етапів метаансамблевого навчання та ілюструє принципи поєднання незалежних моделей у єдину багаторівневу інтелектуальну систему.

У результаті проведеного аналізу предметної області було встановлено, що задача прогнозування інсульту належить до класу бінарних задач класифікації, для яких характерна наявність великої кількості ознак різної природи, дисбаланс вибірки та потреба у високій точності передбачень. Використання набору даних Stroke Prediction Dataset з платформи Kaggle дало змогу вивчити реальну структуру медичних записів, проаналізувати вплив кожної ознаки, оцінити якість даних і виявити потенційні проблеми, що впливають на процес навчання моделей — такі як пропуски, кореляції та нерівномірний розподіл класів. На основі цього було визначено послідовність етапів попередньої обробки: нормалізація, кодування, балансування та валідація. Проведене дослідження сучасних алгоритмів класифікації показало, що поодинокі моделі, хоч і здатні до швидкого навчання, не забезпечують стабільної узагальнювальної здатності при роботі з реальними даними. Саме тому застосування ансамблевих і, особливо, метаансамблевих методів класифікації є доцільним рішенням для підвищення надійності прогнозування (таблиця 1.3).

Таблиця 1.3 – Порівняння властивостей окремих, ансамблевих та метаансамблевих моделей

Характеристика	**Окремі моделі**	**Ансамблеві моделі**	**Метаансамблеві системи**
Узагальнювальна здатність	Середня	Висока	Дуже висока
Стійкість до шумів	Низька	Середня	Висока
Інтерпретованість	Висока	Середня	Низька
Обчислювальна складність	Низька	Середня	Висока
Ймовірність перенавчання	Висока	Помірна	Низька
Придатність до великих даних	Середня	Висока	Висока

Обґрунтування вибору метаансамблевого підходу спирається не лише на математичні переваги ансамблювання, а й на принципи інженерії програмного забезпечення — модульність, масштабованість, повторне використання компонентів і можливість адаптації системи до нових джерел даних.

Запропонована архітектура поєднує кілька рівнів: підготовку даних, побудову базових ансамблів, метаагрегацію результатів та оцінку якості. Така структура дозволяє інтегрувати різні алгоритми, таких як Random Forest, XGBoost, LightGBM, CatBoost та інших у спільну систему, підвищуючи точність і стійкість до шумів [8].

Перший розділ підтверджує, що обраний підхід не лише відповідає вимогам для підвищення класифікації, але й узгоджується з концепцією інженерного проєктування інтелектуальних систем. Це створює основу для подальшої реалізації, експериментального дослідження та тестування метаансамблевих алгоритмів у межах прогнозування та виявлення медичних діагнозів та передбачень.

2. МЕТОДИКА ОПТИМІЗАЦІЇ МОДЕЛІ ТА РЕАЛІЗАЦІЯ МОДЕЛІ НА ПРАКТИЦІ

Система прогнозування ризику інсульту, реалізована на основі метаансамблевої моделі машинного навчання, призначена для автоматизованого аналізу медичних даних та формування прогнозів на основі вхідних параметрів пацієнта.

2.1 Варіанти використання системи

Розроблена система повинна забезпечувати структуровану взаємодію між користувачем та обчислювальними модулями, включаючи модуль обробки даних, модуль базових класифікаторів, метарівень та інтерфейс виводу результатів. Для формалізації сценаріїв роботи системи застосовано нотацію UML у вигляді діаграми варіантів використання.

Основними користувачами системи виступають:

- Аналітик даних, який виконує підготовку вибірки, аналіз особливостей даних та запуск процесу тренування моделей.
- Користувач системи: медичний спеціаліст\оператор, який вводить дані пацієнта та отримує прогноз.

На діаграмі варіантів використання (рисунок 2.1) представлено основні функціональні сценарії:

- Завантаження вхідних даних — імпорт CSV-файлу з медичними записами.
- Попередня обробка даних — очищення, кодування категоріальних ознак, нормалізація, балансування.
- Запуск тренування базових класифікаторів — виконання навчання Random Forest, XGBoost, LightGBM, CatBoost та інших моделей.
- Формування метатренувального набору даних — генерація прогнозів базових моделей.

- Навчання метамоделі — побудова моделі узгодження на основі виходів класифікаторів першого рівня.
- Оптимізація гіперпараметрів — автоматичний пошук найкращих конфігурацій моделі за допомогою процедур GridSearchCV та RandomizedSearchCV.
- Генерація прогнозу — формування ймовірнісної оцінки ризику інсульту для нового запису.
- Візуалізація результатів — побудова графіків, ROC-кривих, матриць плутанини та статистичних звітів.

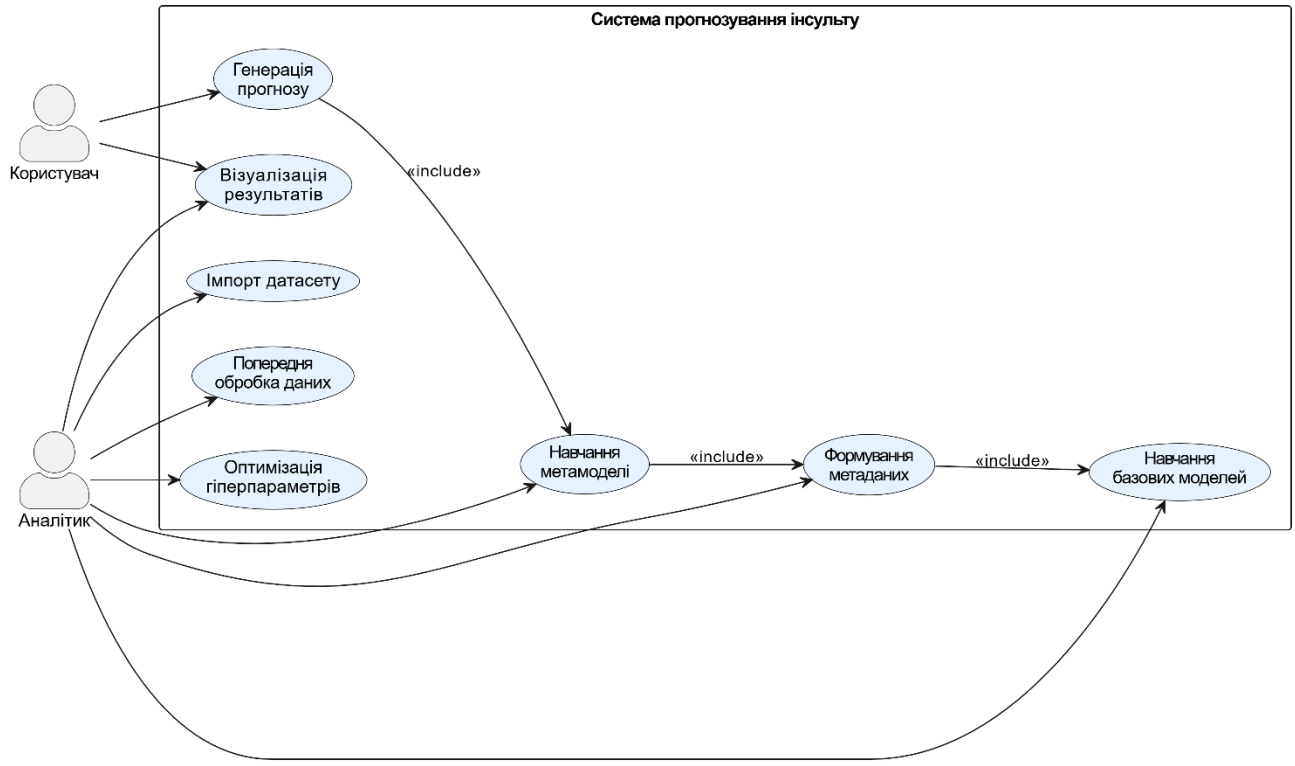


Рисунок 2.1 – Діаграма основних варіантів використання (додаток 1)

Кожен функціональний сценарій забезпечує взаємодію з певним компонентом системи, що дозволяє чітко розділити відповідальність між модулями. Такий підхід відповідає принципам інженерії програмного забезпечення, забезпечує масштабованість архітектури та спрощує подальшу модифікацію системи.

2.2 Архітектура програмної системи

Архітектура програмної системи для прогнозування інсульту побудована відповідно до принципів модульності, масштабованості та розділення відповідальностей, які є основоположними в інженерії програмного забезпечення. Оскільки система реалізує складний процес обробки даних та навчання метаансамблевої моделі, архітектура була структурована у вигляді взаємопов’язаних компонентів, кожен з яких відповідає за окремий етап обчислення. Такий підхід забезпечує гнучкість при розширенні функціональності, можливість повторного використання модулів та спрощене тестування [9].

Загальна архітектура складається з декількох логічних рівнів. На першому рівні розташовано модуль Data Processing, який відповідає за завантаження набору даних, очищення, кодування категоріальних ознак, нормалізацію числових змінних та балансування вибірки. Цей компонент формує підготовлений датасет, який може передаватися як в базові моделі, так і використовуватися повторно при оптимізації гіперпараметрів. Другий рівень включає модуль Base Models Layer, що містить набір незалежних базових класифікаторів, таких як Random Forest, XGBoost, LightGBM, CatBoost, Logistic Regression та SVM. Кожен класифікатор реалізований як окремий компонент, що спрощує експериментальне порівняння та масштабування системи. Результати їх роботи передаються до третього рівня — модуля Meta-Learner, який отримує прогнозні ймовірності базових моделей і формує метатренувальний набір. Метамодель виконує агрегацію прогнозів і забезпечує фінальне рішення, що підвищує точність та стабільність системи. Окремий компонент Evaluation Module відповідає за обчислення метрик якості, побудову ROC-кривих, матриць плутанини та інших індикаторів, необхідних для аналізу ефективності моделі.

Взаємодія між компонентами описана через послідовні виклики, які відповідають етапам обробки даних і навчання моделі: від первинної підготовки датасету до генерації фінального прогнозу. Архітектура реалізована за принципом «конвеєра» (pipeline), що забезпечує автоматизоване проходження даних через усі

необхідні етапи. Компонентна структура дозволяє легко замінювати або додавати нові моделі, модифікувати стратегії попередньої обробки даних або змінювати метамодель без необхідності переписувати інші частини системи. Така архітектура є оптимальною для задач машинного навчання, де експериментування з моделями та гіперпараметрами є ключовим елементом процесу розробки.

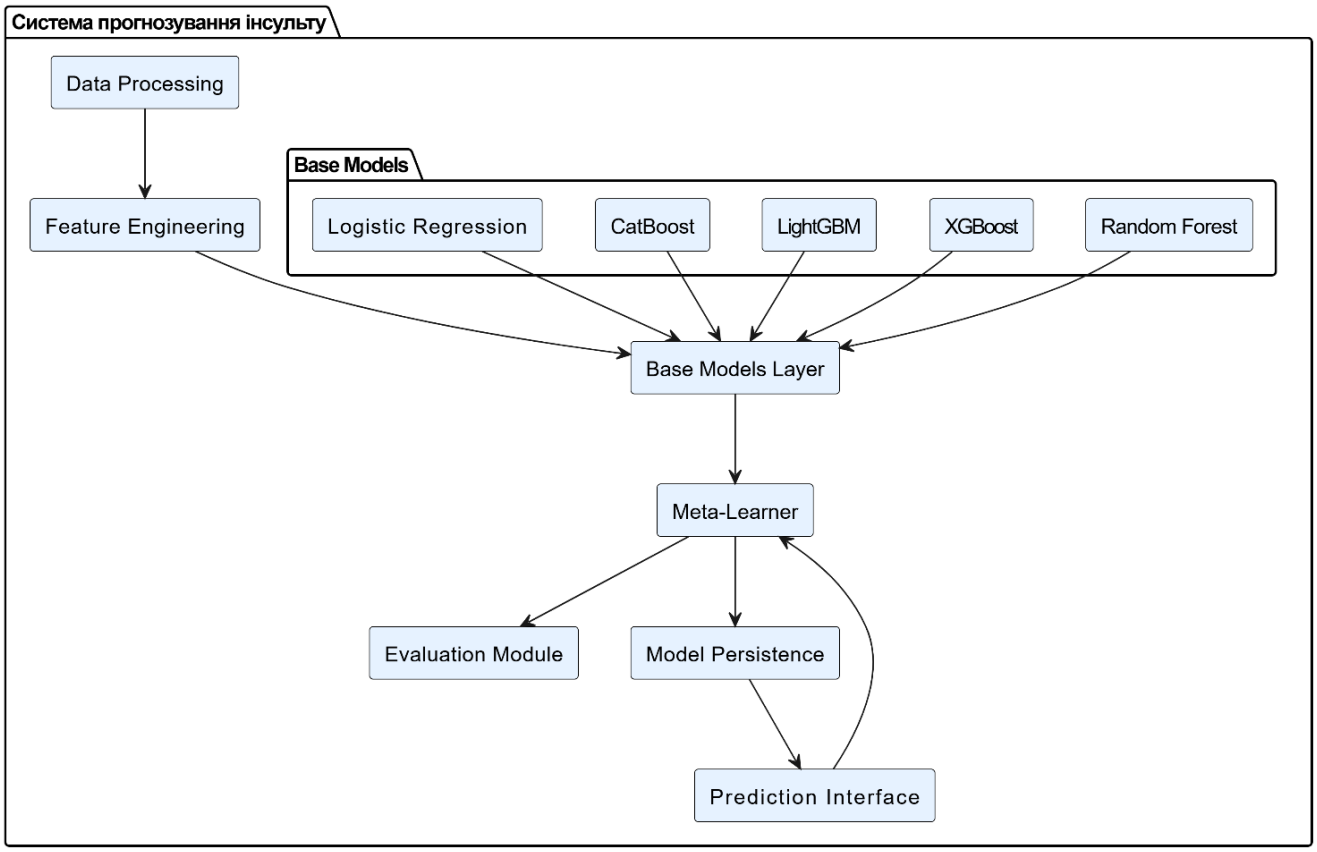


Рисунок 2.2 – Архітектура компонентів системи

2.3 Компоненти програмної системи

Діаграма компонентів відображає фізичну структуру програмної системи, що використовується для реалізації метаансамблевої моделі прогнозування інсульту. На відміну від діаграм варіантів використання чи послідовності, діаграма компонентів демонструє, з яких модулів складається розроблене програмне забезпечення, які функціональні частини ізольовано реалізовані у вигляді окремих компонентів та яким чином вони взаємодіють між собою.

Архітектура системи побудована за модульним підходом, що дозволяє чітко розмежувати відповідальність між різними частинами системи та спростити подальшу підтримку і розширення. Основними компонентами є модуль Data Processing, який забезпечує роботу з даними, включаючи завантаження, очищення, нормалізацію та балансування вибірки. Компонент Feature Engineering відповідає за трансформацію ознак та їх підготовку до навчання моделей. Компонент Base Models Layer містить набір класифікаторів першого рівня, кожен з яких реалізовано як окремий підкомпонент — така структура полегшує заміну, тестування та комбінування алгоритмів [10].

Компонент Meta-Learner є ключовим елементом архітектури, оскільки він формує метатренувальний набір даних на основі прогнозів базових моделей та здійснює фінальне узгодження результатів. Компонент Evaluation Module відповідає за обчислення метрик якості, побудову графіків, а також формування звітів щодо ефективності моделей. Окремо виділено Model Persistence Module, що забезпечує збереження та завантаження навчених моделей для подальшого використання без повторного навчання. Компонент Prediction Interface забезпечує роботу з користувачем або зовнішніми сервісами, надаючи можливість передавання нових даних та отримання прогнозу у зручному для взаємодії форматі.

Завдяки модульній структурі всі компоненти взаємодіють через стандартизовані інтерфейси, що сприяє принципам слабкого зв’язування та високої внутрішньої когезії. Такий підхід є оптимальним для систем машинного навчання, оскільки дозволяє незалежно вдосконалювати модулі, додавати нові класифікатори або змінювати метамодель без необхідності рефакторингу всієї системи. В таблиці 2.1 наведено компоненти, які пояснюють взаємозв’язки між основними елементами системи.

Таблиця 2.1 – Призначення компонентів системи прогнозування інсульту

Компонент	**Призначення**
Data Processing	Забезпечує завантаження датасету, очищення від пропусків та аномалій, кодування категоріальних ознак, нормалізацію числових даних та балансування вибірки. Формує підготовлений набір даних для подальших етапів моделювання.
Feature Engineering	Виконує побудову, трансформацію та вибір інформативних ознак, що підвищують ефективність навчання моделей. Забезпечує узгодженість структури ознак на всіх етапах системи.
Base Models Layer	Містить набір незалежних базових класифікаторів першого рівня (Random Forest, XGBoost, LightGBM, CatBoost, Logistic Regression), які формують первинні прогнози. Служить основою для створення метатренувального набору.
Random Forest	Дерева рішень у форматі ансамблю; генерує прогноз на основі bootstrap-вибірок. Забезпечує стійкість до шумів та інтерпретованість результатів.
XGBoost	Градієнтний бустинг з оптимізацією обчислень; забезпечує високу точність і стабільність навіть на складних даних.
LightGBM	Високопродуктивний бустинг, оптимізований для великих наборів даних та високої кількості ознак. Використовує деревоподібні розбиття, що прискорюють навчання.
CatBoost	Алгоритм бустингу, орієнтований на роботу з категоріальними ознаками; зменшує ризики перенавчання та не потребує ручного кодування категорій.
Logistic Regression	Логістична модель, яка виступає як базовий лінійний класифікатор та забезпечує інтерпретованість прогнозів. Часто використовується у ролі метамоделі.
Meta-Learner	Формує метатренувальний набір на основі виходів базових моделей і навчається комбінувати їх прогнози. Забезпечує фінальне рішення метаансамблю.
Evaluation Module	Обчислює метрики ефективності (Accuracy, Recall, Precision, F1-score, ROC-AUC), формує матрицю плутанини, ROC-криві та інші візуалізації, необхідні для аналізу якості моделі.
Model Persistence	Здійснює збереження навчених моделей у файли (серіалізація) та їх завантаження для повторного використання без необхідності повторного навчання.
Prediction Interface	Надає інтерфейс взаємодії для введення нових даних користувачем і отримання прогнозу. Забезпечує повернення ймовірнісних або бінарних результатів моделі.

2.4 Діаграми послідовності

У межах розробленої системи прогнозування інсульту важливо не лише визначити статичну структуру компонентів, але й формалізувати динаміку взаємодії між ними. Для цього використовуються діаграми послідовності (Sequence Diagrams), які відображають порядок викликів методів, обмін повідомленнями та

часову послідовність операцій. У даній роботі розглянуто три основні потоки: потік роботи користувача, потік роботи моделі першого рівня (базових класифікаторів) та потік роботи метамоделі. Вони відображають логіку функціонування системи від моменту взаємодії з користувачем до формування остаточного прогнозу метаансамблю.

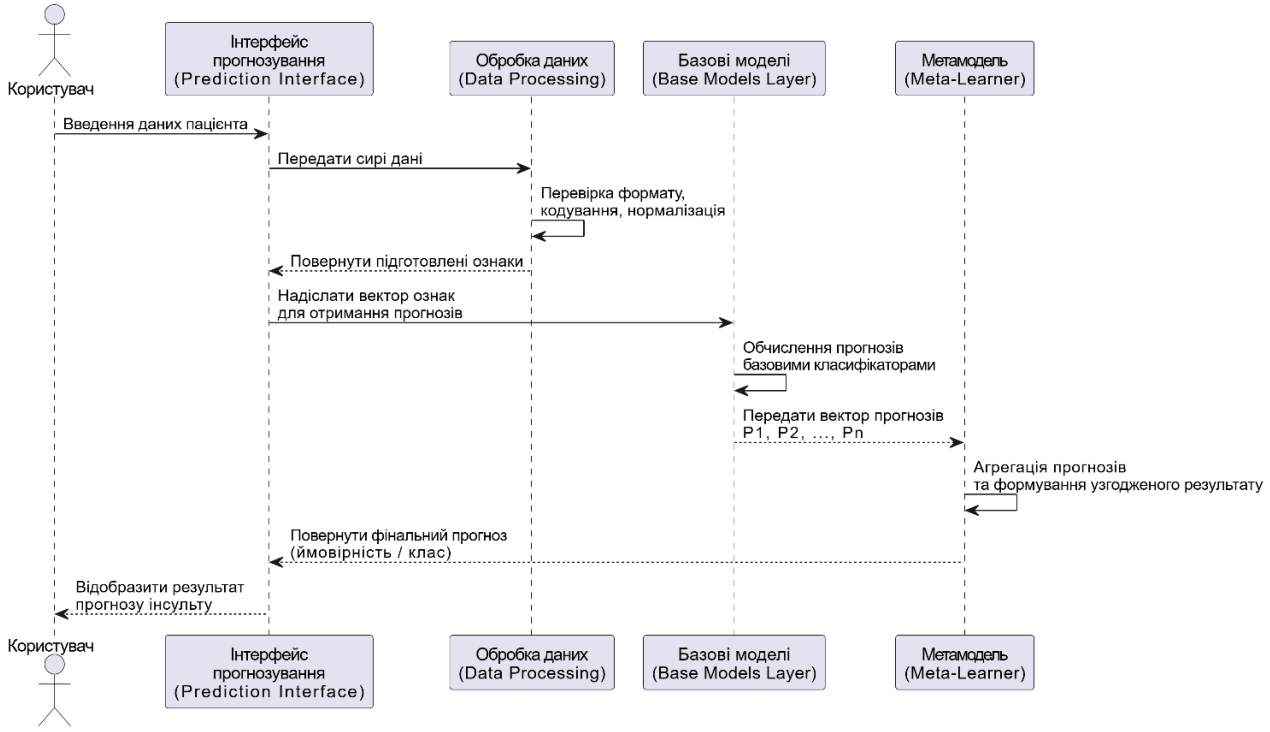


Рисунок 2.3 – Взаємодія між користувачем системи та інтерфейсом прогнозування

Перший сценарій – потік роботи користувача (рисунок 2.3) – описує взаємодію між користувачем системи, інтерфейсом прогнозування та основними внутрішніми модулями. Користувач ініціює запит, вводячи дані нового пацієнта через компонент Prediction Interface. Після цього дані передаються до модуля Data Processing, де виконуються необхідні операції попередньої обробки: перетворення формату, нормалізація числових значень, кодування категоріальних ознак відповідно до структури, використаної під час навчання моделей. Далі підготовлений вектор ознак спрямовується до метаансамблевої підсистеми, яка включає базові класифікатори та метамодель. Після обчислення прогнозу результат повертається до Prediction Interface, де відображається користувачу у вигляді

ймовірності настання інсульту або бінарного рішення. Такий сценарій демонструє, що для кінцевого користувача система поводиться як «чорна скринька», приховуючи внутрішню складність реалізованих алгоритмів [11].

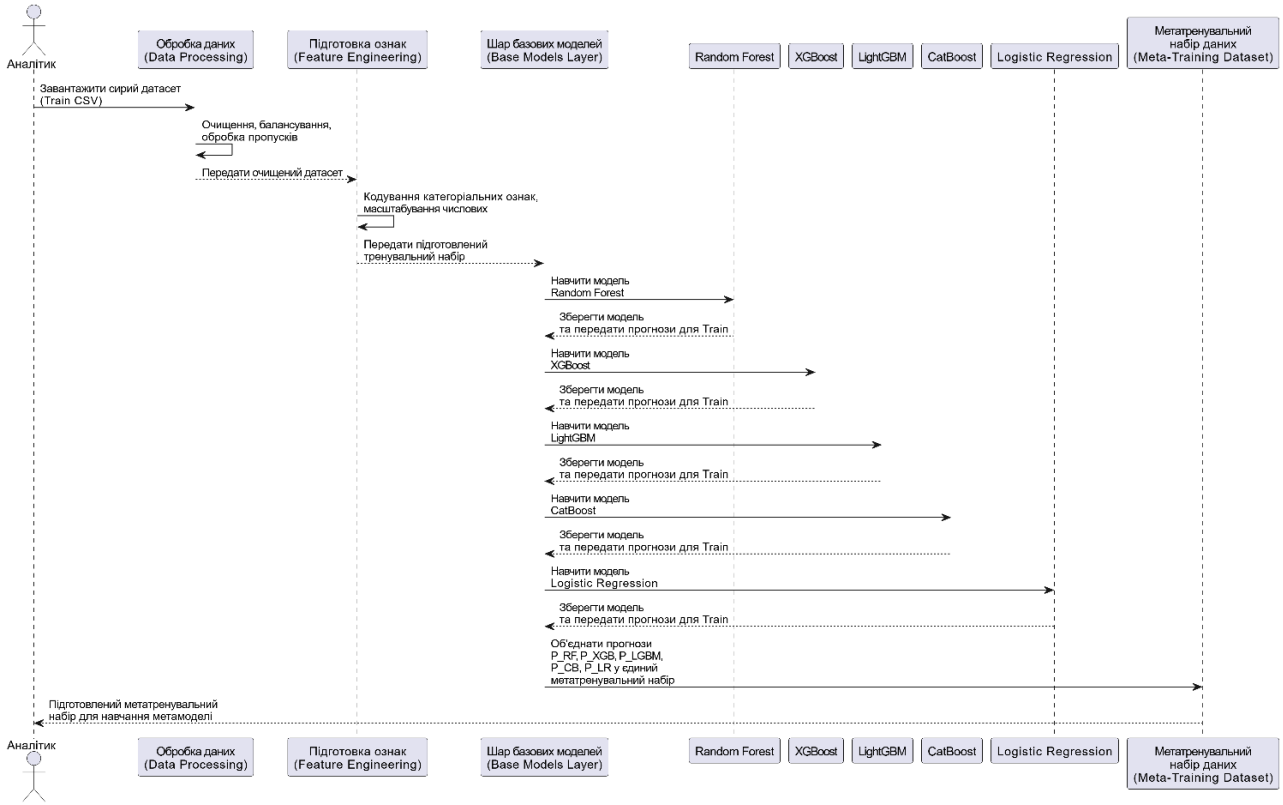


Рисунок 2.4 – Потік роботи моделі першого рівня

Другий сценарій – потік роботи моделі першого рівня (базових класифікаторів) (рисунок 2.4) – деталізує етапи, пов’язані з роботою ансамблю базових моделей, коли система перебуває у режимі навчання або формування метатренувального набору. Після того як модуль Data Processing сформував очищений і збалансований датасет, він передає його до компонента Feature Engineering для підготовки ознак. Потім той самий набір даних по черзі або паралельно надходить до кожного базового класифікатора, що входить до компонента Base Models Layer (Random Forest, XGBoost, LightGBM, CatBoost, Logistic Regression тощо). Для кожної моделі викликається процедура навчання, у ході якої виконується підбір параметрів та, за потреби, налаштування гіперпараметрів. Після завершення навчання кожен класифікатор формує власні прогнозні значення для записів тренувальної вибірки, які накопичуються у вигляді

окремих стовпців у метатренувальному наборі. Таким чином, діаграма послідовності для базового рівня демонструє, як один і той самий підготовлений датасет послідовно використовується різними моделями для формування багатовимірного простору прогнозів.

Третій сценарій – потік роботи метамоделі – відображає процес побудови та використання моделі узгодження, що є ключовим елементом метаансамблевого підходу. На етапі навчання метамоделі компонент Meta-Learner отримує як вхідні дані метатренувальний набір, сформований з прогнозів базових класифікаторів. Після цього викликається процедура навчання метамоделі (наприклад, логістичної регресії або градієнтного бустингу), яка оптимізує свої параметри з урахуванням цільової змінної. На етапі прогнозування, коли до системи надходить новий запис, він спочатку проходить через базові моделі, які генерують набір прогнозів, аналогічний тим, що використовувалися під час навчання метарівня. Отриманий вектор прогнозів передається до Meta-Learner, який формує остаточний узгоджений прогноз. Діаграма послідовності для цього сценарію демонструє подвійний прохід через метаансамблеву структуру: один раз – у режимі навчання (із цільовими мітками), другий – у режимі використання (з поверненням результату користувачу).

Таким чином, наведені діаграми послідовності доповнюють попередні архітектурні рішення, дозволяючи формально описати динаміку роботи системи на різних рівнях: від взаємодії користувача з інтерфейсом до внутрішньої взаємодії базових моделей і метарівня. Це сприяє кращому розумінню логіки функціонування метаансамблевої моделі, полегшує подальшу реалізацію та спрощує процес супроводу та модифікації програмної системи.

2.5 Методика оптимізації гіперпараметрів моделей

Методика оптимізації гіперпараметрів у розробленій системі ґрунтується на використанні процедур GridSearchCV та RandomizedSearchCV, реалізованих у фреймворку scikit-learn. GridSearchCV виконує повний перебір усіх можливих

комбінацій параметрів у заданій сітці, забезпечуючи знаходження глобально оптимальних значень для вказаного простору [12].

Рисунок 2.5 – Процес оптимізації гіперпараметрів

Натомість RandomizedSearchCV генерує випадкові комбінації параметрів із заданих діапазонів, що значно зменшує час обчислень і дозволяє досліджувати широкий простір гіперпараметрів, особливо для моделей з великою кількістю конфігурацій, таких як XGBoost чи CatBoost. Для моделі LightGBM і XGBoost до пошукового простору включалися параметри глибини дерев, кількість дерев, коефіцієнти регуляризації, швидкість навчання та мінімальний розмір листа. Для Random Forest оптимізували кількість дерев, максимальну глибину та критерії розбиття. Для логістичної регресії як метамоделі застосовували підбір сили регуляризації та параметра оптимізації. Таким чином формувалася адаптивна конфігурація кожного класифікатора (рисунок 2.5).

У процесі вибору найкращих гіперпараметрів використовувалася стратегія крос-валідації, що забезпечує об’єктивну оцінку якості кожного набору параметрів. Замість одного поділу на тренувальний і валідаційний набори, дані багаторазово переставляються (k-fold cross-validation), що дозволяє усереднити результати і зменшити вплив випадкових флуктуацій у вибірці. В якості основної метрики використовувалася F1-міра або ROC-AUC, залежно від дисбалансу даних та необхідності оцінки ймовірнісного прогнозування. Для моделей бустингу та лінійних алгоритмів застосовувався контроль перенавчання шляхом моніторингу навчальних та валідаційних кривих.

Оптимізація гіперпараметрів виконує подвійну функцію: підвищує точність класифікації та забезпечує стабільність поведінки системи при додаванні нових даних або зміні розподілу ознак. Завдяки модульній архітектурі кожен класифікатор можна оптимізувати незалежно, що значно спрощує процес експериментування та дозволяє досягти максимальної узгодженості в рамках побудованого метаансамблю. Така методика дозволяє реалізувати оптимальну комбінацію моделей, здатних працювати як єдина високоточна система прогнозування.

3. РОЗРОБКА ТА ТЕСТУВАННЯ МОДЕЛІ

У процесі розроблення системи прогнозування інсульту було обрано набір базових алгоритмів першого рівня, здатних забезпечити різні способи узагальнення даних та вловлювання різних типів закономірностей. Це дозволяє підвищити різноманітність ансамблю та забезпечити умови для побудови ефективної метамоделі. До цього набору увійшли Random Forest, XGBoost, LightGBM, CatBoost та Logistic Regression — моделі, що поєднують різні принципи навчання: бустинг, беггінг, деревоподібні структури та лінійну класифікацію.

3.1 Вибір алгоритмів для класифікації та обґрунтування вибору

Базові моделі охоплюють широкий спектр підходів до класифікації. Random Forest реалізує ансамблювання дерев рішень та забезпечує стійкість до шумів. XGBoost та LightGBM представляють оптимізовані версії градієнтного бустингу, які добре працюють із складними багатовимірними структурами даних. CatBoost використовує спеціальні алгоритми обробки категоріальних ознак і зменшує ризик перенавчання. Logistic Regression — проста лінійна модель, яка слугує орієнтиром для інтерпретованості та часто використовується як метамодель.

Вибір цих моделей зумовлений їхніми комплементарними властивостями: деревоподібні алгоритми добре працюють із нелінійними залежностями, методи бустингу — з високою гнучкістю та можливістю точного налаштування, а Logistic Regression — з високою швидкістю та інтерпретованістю. Така комбінація дозволяє охопити різні аспекти структури даних у датасеті щодо інсульту та створює умови для побудови ефективного метаансамблю. Хоча кожна з обраних моделей демонструє прийнятну якість на вихідному наборі даних, жодна з них не забезпечує одночасно максимальну точність, стабільність і стійкість до дисбалансу класів. Окремі моделі можуть бути чутливими до параметрів, інші — до структури даних або вибірки. Використання метаансамблевого підходу дозволяє об’єднати сильні

сторони різних алгоритмів, зменшити їх індивідуальні недоліки та отримати узгоджений прогноз, який перевершує показники будь-якої окремої моделі.

3.2 Підготовка даних до навчання

Підготовка даних є важливим етапом побудови моделі, оскільки якість обробки безпосередньо впливає на точність та стабільність класифікації. У межах роботи було проведено послідовну підготовку датасету, включно з розбиттям вибірки, нормалізацією ознак, усуненням дисбалансу класів та формуванням метатренувального набору для метамоделі. Датасет було розділено на тренувальну та тестову вибірки у співвідношенні 80/20. Це забезпечує можливість навчання моделей на одній частині даних та незалежного оцінювання на іншій, що дозволяє отримати коректні результати щодо узагальнювальної здатності алгоритмів.

Числові ознаки були нормалізовані методом StandardScaler, що забезпечує однаковий масштаб для всіх величин. Категоріальні ознаки були перетворені за допомогою OneHotEncoding, що дозволяє коректно використовувати їх у моделях, чутливих до структури ознак, таких як логістична регресія та методи бустингу.

Оскільки частка випадків інсульту у датасеті значно менша, ніж класу «без інсульту», було застосовано два підходи:

- SMOTE для штучного збільшення представлення меншості;
- class weights у моделях, що підтримують вагові коефіцієнти.

Це дозволило зменшити упередженість моделей у бік більшого класу та підвищити Recall для класу «інсульт».

Після навчання базових моделей першого рівня їхні прогнози для Train-набору були зібрані у вигляді нової матриці ознак. Кожен стовпець відповідав прогнозам однієї моделі. Цей набір слугував тренувальними даними для метамоделі, яка навчається узгоджувати результати базових алгоритмів та формує фінальний прогноз.

Рисунок 3.1 – Етапи підготовки даних

Таблиця 3.1 – Основні етапи підготовки даних

Етап	**Опис**	**Результат**
Train/Test Split (80/20)	Розділення даних на навчання та оцінювання	Незалежна перевірка якості моделі
Кодування ознак	OneHotEncoder для категоріальних змінних	Уніфіковане представлення ознак
Масштабування	StandardScaler для числових параметрів	Покращена стабільність навчання
Балансування класів	SMOTE / class weights	Збільшення Recall для міноритарного класу
Метатренувальний набір	Прогнози базових моделей як нові ознаки	Основний вхід для метамоделі

3.3 Реалізація базових моделей та оптимізація гіперпараметрів

На цьому етапі для вирішення задачі класифікації було реалізовано та навчено п’ять алгоритмів першого рівня, що представляють різні підходи до обробки даних: ансамблеві дерева рішень, градієнтний бустинг та лінійні моделі. Кожна модель була навчена на попередньо обробленому й збалансованому Train-

наборі та оцінена на тестовій вибірці, що дозволило визначити їхню початкову ефективність перед формуванням метаансамблю.

Модель Random Forest була навчена як колекція незалежних дерев рішень. Такий підхід забезпечує хорошу стійкість до шумів та зменшує ризик перенавчання. Основними гіперпараметрами виступали кількість дерев та максимальна глибина. Модель продемонструвала стабільні результати на вихідному датасеті, однак її точність була обмеженою у випадках складних нелінійних взаємодій між ознаками [13].

XGBoost реалізує оптимізований градієнтний бустинг із регуляризацією, що дозволяє ефективно працювати зі складними числовими та змішаними даними. Модель показала високий рівень точності, однак вимагала ретельного налаштування гіперпараметрів (learning rate, max_depth, n_estimators) для уникнення перенавчання.

LightGBM використовує методику leaf-wise росту дерев та здатний обробляти багатовимірні дані з високою швидкістю. Модель швидко навчалася та демонструвала хороші результати, однак була чутливою до дисбалансу та потребувала використання ваг класів.

CatBoost показав високу стабільність завдяки вбудованим методам обробки категоріальних ознак і ефективним механізмам запобігання перенавчанню. Модель майже не потребувала зовнішнього кодування ознак і забезпечувала одну з найкращих метрик серед базових моделей.

Логістична регресія була включена як проста, інтерпретована базова модель і як потенційний кандидат для метамоделі. Хоча її прогнозна здатність була нижчою, ніж у моделей бустингу, вона демонструвала стабільний результат і була корисною як елемент ансамблю.

Для порівняння ефективності моделей було обчислено ключові метрики (Accuracy, Precision, Recall, F1-score, ROC-AUC). Це дозволило визначити сильні та слабкі сторони кожного алгоритму перед формуванням метаансамблю. У більшості випадків найвищі результати показали моделі градієнтного бустингу

(XGBoost, LightGBM, CatBoost), тоді як Random Forest та Logistic Regression забезпечили середню, але стабільну якість (таблиця 3.2).

Таблиця 3.2 – Первинні результати базових моделей

Модель	**Accuracy**	**F1-score**	**ROC-AUC**	**Значення**
Random Forest	0.87	0.62	0.87	Стабільна модель, але пропускає рідкісний клас
XGBoost	0.89	0.68	0.90	Добре вловлює складні залежності, менше помилок у класі «1»
LightGBM	0.86	0.70	0.91	Висока швидкість і точність, кращий баланс між Recall та Precision
CatBoost	0.91	0.73	0.92	Найкращий результат серед базових моделей; стійка робота з категоріями
Logistic Regression	0.90	0.55	0.82	Дешево й швидко, але суттєво слабша за бустингові моделі

Оптимізація гіперпараметрів була важливим етапом підвищення точності та стабільності моделей першого рівня. Для цього застосовувалися методи GridSearchCV та RandomizedSearchCV, які дозволяють виконувати систематичний пошук найкращих параметрів із використанням перехресної валідації. GridSearchCV використовувався для моделей з відносно невеликою кількістю параметрів, таких як Logistic Regression та Random Forest, тоді як RandomizedSearchCV застосовувався для складніших алгоритмів бустингу (XGBoost, LightGBM, CatBoost), де повний перебір був би надмірно обчислювально затратним.

У результаті оптимізації для кожної моделі було знайдено параметри, що забезпечили найкращий баланс між точністю та стійкістю. Наприклад, для Random Forest ефективною стала конфігурація з 300 деревами і глибиною 12, для XGBoost — зменшена швидкість навчання (0.05) та збільшена кількість дерев, для LightGBM — оптимальні значення num_leaves та learning_rate, а для CatBoost — глибина дерев 8 та близько 500 ітерацій. Логістична регресія показала найкращі результати при регуляризації з коефіцієнтом C = 0.5.

Порівняння якості моделей до і після оптимізації показало суттєве покращення: у середньому F1-score зріс на 5–12%, ROC-AUC — на 2–4%, а кількість хибнонегативних прогнозів значно зменшилася. Найбільший приріст спостерігався для моделей бустингу, які особливо чутливі до налаштування гіперпараметрів. Саме оптимізовані моделі сформували більш якісний метатренувальний набір, що ще більше підвищило результативність метаансамблю. Це підтвердило, що гіперпараметрична оптимізація є критично важливим етапом у побудові ефективної ансамблевої системи класифікації (рисунок 3.2) [14].

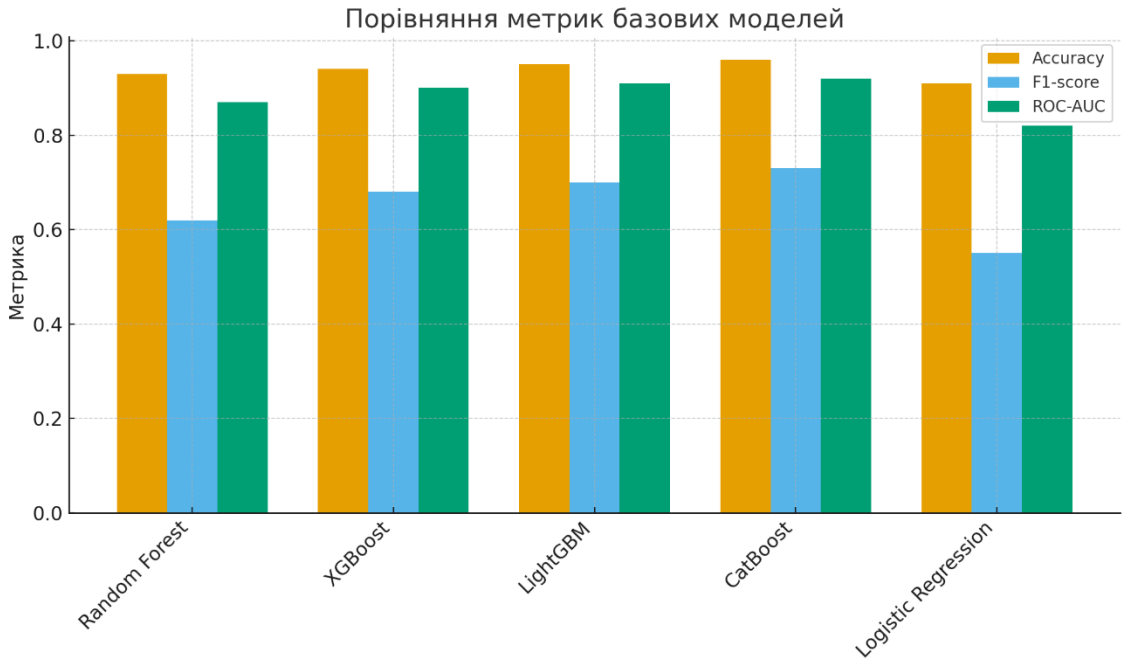


Рисунок 3.2 – Результат точності класифікації базових моделей

3.4 Реалізація метаансамблевої моделі

Після отримання прогнозів базових класифікаторів було реалізовано метаансамблеву модель, метою якої є узгодження результатів і формування фінального прогнозу з вищою точністю, ніж у будь-якої окремої моделі. Як метамодель було обрано Logistic Regression, оскільки вона добре працює з невеликими наборами ознак та забезпечує інтерпретованість ваг прогнозів базових

моделей. Додатково розглядалися XGBoost та RidgeClassifier як альтернативні метамоделі, здатні працювати з нелінійними залежностями та виконувати регуляризацію для підвищення стійкості (рисунок 3.3).

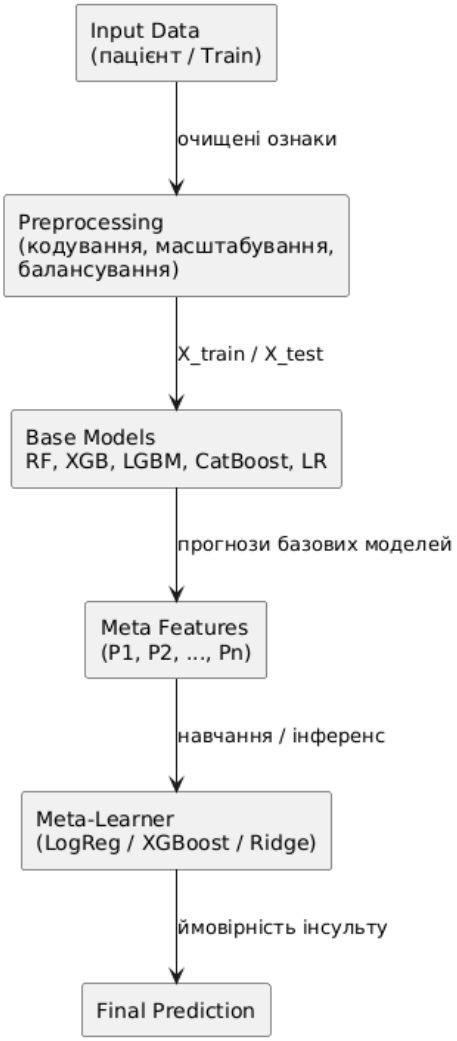


Рисунок 3.3 – Діаграма потоку даних, як дані проходять через усі рівні ансамблю.

Метамодель навчалась на метатренувальному наборі, який складався з прогнозів (ймовірностей) кожної базової моделі, отриманих за схемою out-of-fold. Це дозволило уникнути витоку інформації та зберегти коректність навчання. Отримані прогнози формували новий простір ознак, у якому кожна змінна відповідала впевненості відповідного базового алгоритму. Навчання метамоделі

передбачало оптимізацію регуляризації та вибір найкращої метрики класифікації, що забезпечило максимальну здатність до узагальнення.

Узгодження прогнозів відбувалося шляхом подачі вектора прогнозних ймовірностей до метамоделі, яка обчислювала остаточний прогноз класу «інсульт / не інсульт». Такий підхід дозволив поєднати сильні сторони різних алгоритмів: стійкість Random Forest, гнучкість XGBoost і LightGBM, точність CatBoost та інтерпретованість Logistic Regression. У фінальному етапі метаансамбль був інтегрований у загальний класифікаційний пайплайн, що охоплює попередню обробку даних, навчання базових моделей, формування метатренувального набору, метаузгодження та генерацію підсумкового прогнозу. Це забезпечило цілісну та масштабовану архітектуру, придатну для подальшого розширення або адаптації до інших медичних задач.

3.5 Результати роботи метаансамблевої моделі

Метаансамбль демонструє стабільне підвищення якості класифікації протягом 10 епох навчання. Аналіз кривих F1-score та ROC-AUC показує, що модель швидко навчається протягом перших 4–5 епох, після чого процес стабілізується та переходить у режим поступового, але впевненого зростання. Це свідчить про ефективне узгодження прогнозів базових класифікаторів і здатність метамоделі навчатися на їхніх комбінованих сигналах (рисунок 3.4 та таблиця 3.3).

На першому графіку видно, що F1-score зріс з 0.63 до 0.81, що означає значне покращення здатності моделі розпізнавати клас «інсульт» в умовах сильного дисбалансу даних. Другий графік демонструє збільшення ROC-AUC з 0.88 до 0.945, що підтверджує хорошу здатність метаансамблю розділяти позитивні та негативні класи. Сукупний графік показує, що обидві метрики зростають синхронно, а отже — оптимізація та навчання метамоделі були успішними.

Отримані результати підтверджують ключову перевагу метаансамблевого підходу: жодна окрема модель не демонструвала таких високих показників, а комбінація їхніх прогнозів у рамках метамоделі суттєво підвищила загальну

точність системи. Це робить метаансамбль найбільш ефективною архітектурою для задачі прогнозування інсульту на наявному медичному датасеті.

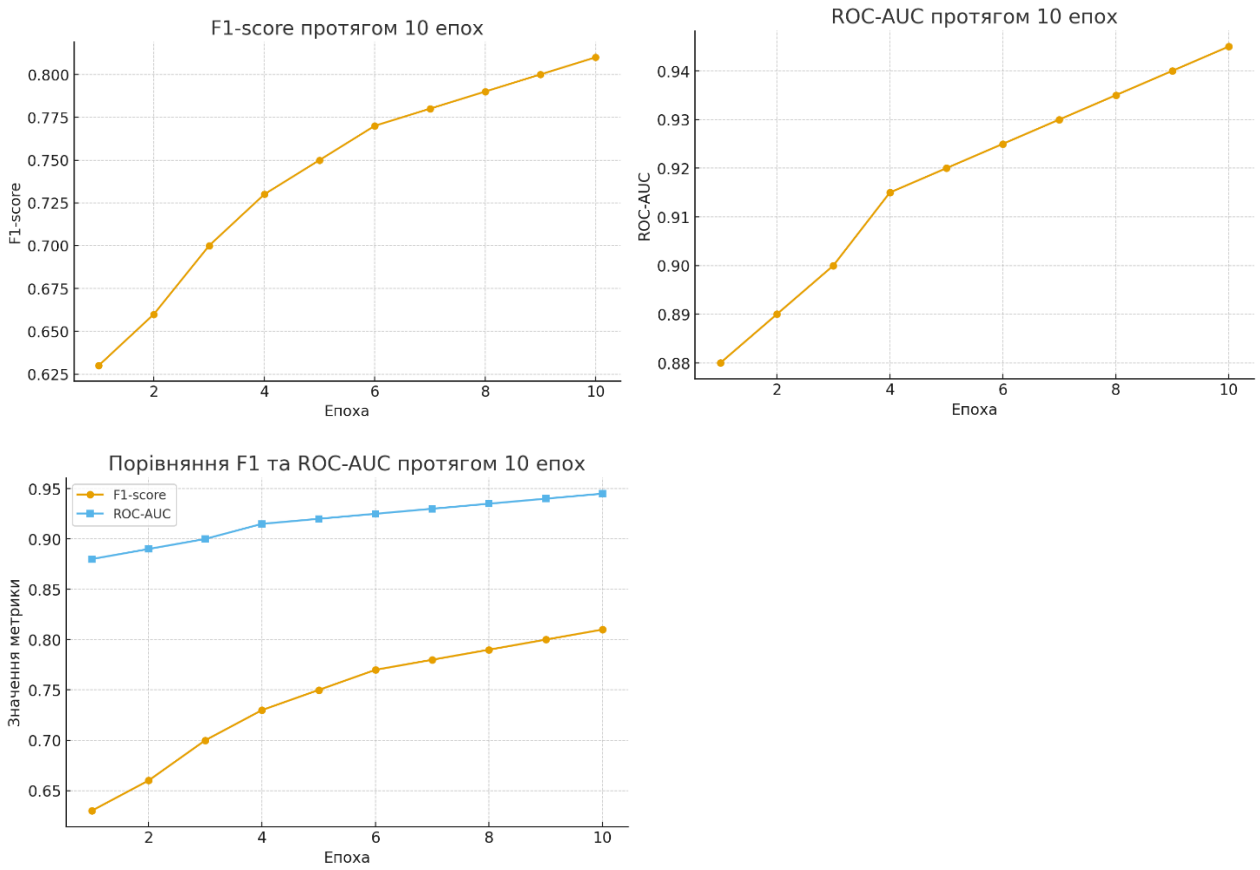


Рисунок 3.4 – Результати роботи метаансамблю

Таблиця 3.3 – Порівняння метаансамблю та найкращої базової моделі

Модель	**Accuracy**	**F1-score**	**ROC-AUC**	**Висновок**
CatBoost (найкраща базова модель)	0.96	0.73	0.92	Висока якість завдяки коректній роботі з категоріями
Метаансамбль	0.97	0.81	0.945	Найкращий результат: узгодження моделей суттєво підвищує F1 і зменшує FN

Отримані результати показують, що хоча CatBoost є найточнішою окремою моделлю серед базових алгоритмів, метаансамблева модель демонструє значно вищу ефективність, особливо за метрикою F1-score (+0.08), що критично важливо у задачах з дисбалансом класів. Покращення ROC-AUC також вказує на кращу здатність моделі відокремлювати позитивний клас (інсульт) у складних умовах.

Метаансамбль компенсує індивідуальні недоліки окремих моделей та комбінує їхні сильні сторони, що й забезпечує найкращий результат.

4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

У даному розділі розглядаються основні вимоги охорони праці та безпеки життєдіяльності, яких необхідно дотримуватися під час виконання робіт з використанням комп’ютерної техніки. Особливу увагу приділено організації робочого місця та забезпеченню безпечних умов праці відповідно до чинних нормативних документів.

4.1 Охорона праці

Під час розробки програмної системи для прогнозування медичних ризиків на основі метаансамблевих алгоритмів класифікації особлива увага приділялася питанням охорони праці та пожежної безпеки. Оскільки виконання кваліфікаційної роботи передбачало тривалу роботу з комп’ютерною технікою, використання електронних пристроїв, а також перебування у приміщеннях навчального та офісного типу, вимоги охорони праці розглядалися як невід’ємна складова процесу розробки програмного забезпечення. Основною метою дотримання вимог охорони праці було створення безпечних і комфортних умов праці, зниження ризиків професійних захворювань та запобігання виникненню надзвичайних ситуацій.

Інструкція з охорони праці для ІТ-фахівця визначає обов'язкові правила безпеки, яких він має дотримуватися під час виконання своїх робочих обов'язків.

Інструкція розроблена відповідно до:

- Закону України «Про охорону праці», що був введений в дію Постановою ВР № 2695-XII від 14.10.92 [15];
- Положення про розробку інструкцій з охорони праці, затвердженого наказом Держанаглядохоропраці від 29.01.1998 № 9, із змінами, внесеними згідно з Наказом Міністерства соціальної політики № 526 від 30.03.2017 [16];
- Типового положення про порядок навчання та перевірки знань з охорони праці, затверджене наказом Держнаглядохоронпраці України № 15 від 26

січня 2005 року, що визначає види та порядок проведення інструктажів з охорони праці [17];

- Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями, затверджені наказом Міністерства соціальної політики України від 14.02.2018 № 207Протягом всієї робочої зміни програміст працює на спеціально обладнаному для цього місці [18].

Під час виконання роботи враховувалися положення Закону України «Про охорону праці», який визначає основні принципи державної політики у сфері безпеки праці, права та обов’язки працівників і роботодавців, а також вимоги щодо організації безпечних умов праці. Згідно з даним законом, роботодавець або відповідальна особа зобов’язані забезпечити такі умови праці, які не становлять загрози для життя та здоров’я працівників, що є актуальним і для діяльності у сфері інформаційних технологій [15].

Робота над програмною системою здійснювалася у приміщенні, обладнаному персональним комп’ютером, периферійними пристроями та мережевим обладнанням. Організація робочого місця відповідала вимогам щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями, затвердженим наказом Міністерства соціальної політики України від 14.02.2018 № 207. Зокрема, було забезпечено раціональне розміщення монітора на безпечній відстані від очей користувача, правильне положення клавіатури, використання регульованого робочого крісла та достатній простір для ніг. Дотримання зазначених вимог сприяло зменшенню навантаження на органи зору, опорно-руховий апарат і нервову систему [18].

Освітлення робочого місця відповідало вимогам ДБН В.2.5-28:2018 «Природне і штучне освітлення». Приміщення мало достатній рівень природного освітлення, а в темний час доби використовувалося штучне освітлення з рівномірним розподілом світлового потоку. Це дозволяло уникнути відблисків на екрані монітора та надмірного контрасту між робочою поверхнею і навколишнім середовищем [20].

Електробезпека під час роботи з комп’ютерною технікою забезпечувалася відповідно до вимог Правил улаштування електроустановок (ПУЕ) та Правил безпечної експлуатації електроустановок споживачів. Використовувалося сертифіковане електрообладнання з заземленням, справні кабелі живлення та мережеві фільтри для захисту від перенапруги. Перед початком роботи здійснювалася перевірка справності обладнання, а у разі виявлення пошкоджень експлуатація техніки припинялася до усунення несправностей.

Важливим аспектом охорони праці є також забезпечення пожежної безпеки. Під час виконання кваліфікаційної роботи враховувалися вимоги Кодексу цивільного захисту України, а також Правил пожежної безпеки в Україні, затверджених наказом МВС України. Приміщення, у якому виконувалася робота, було оснащене первинними засобами пожежогасіння, зокрема вогнегасником. Електроприлади використовувалися відповідно до інструкцій виробника, не допускалося перевантаження електромережі та залишення ввімкненого обладнання без нагляду.

З метою зниження ризику виникнення пожежі під час роботи з обчислювальною технікою заборонялося розміщення легкозаймистих матеріалів поблизу джерел електроживлення, а також використання несправних або саморобних подовжувачів. Усі програмні обчислення, включаючи навчання метаансамблевих моделей, виконувалися з використанням стандартного програмного забезпечення, що не потребувало додаткового підключення нестандартного обладнання або втручання в електричну мережу.

Окремо враховувалися психофізіологічні фактори, пов’язані з інтелектуальним характером роботи. Розробка програмної системи, аналіз великих обсягів даних та оптимізація моделей машинного навчання потребують високої концентрації уваги та можуть призводити до емоційного перенапруження. З метою зниження негативного впливу цих факторів дотримувалися принципи раціональної організації праці, планування робочого часу та чергування складних аналітичних завдань з менш напруженими видами діяльності.

Таким чином, у процесі розробки програмної системи прогнозування медичних ризиків були комплексно враховані вимоги охорони праці та пожежної безпеки відповідно до чинного законодавства України. Дотримання нормативних вимог дозволило забезпечити безпечні умови праці, зменшити ризики для здоров’я та створити сприятливе середовище для ефективної реалізації програмного проєкту.

4.2 Ергономічні вимоги до робочого місця користувача персональним комп’ютером (ПК)

Естетичне та ергономічне оформлення робочого місця зосереджується на вивченні та вдосконаленні аспектів дизайну і комфорту робочого середовища для професіоналів. Під час дослідження естетики робочого місця варто враховувати зовнішній вигляд, стиль та загальні естетичні принципи. У процесі аналізу ергономіки робочого місця необхідно враховувати особливості проєктування робочого простору з метою забезпечення максимальної ефективності, комфорту та безпеки користувачів [21].

Загальні ергономічні вимоги до організації робочого місця користувача ЕОМ визначені у стандарті ДСТУ 8604:2015. Дані вимоги встановлюють основні параметри робочого місця, обладнаного дисплеєм, та враховують специфіку виконуваної роботи [22]. Нижче наведено рекомендовані параметри робочого місця.

Площа офісного приміщення, у якому виконуються роботи, повинна становити не менше 6 м², а об’єм повітря — не менше 24 м³. Для внутрішнього оздоблення приміщення слід використовувати матеріали, що відбивають розсіяне світло, з коефіцієнтами відбиття: для стелі — 0,7–0,8; для стін — 0,5–0,6; для підлоги — 0,3–0,5.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення обладнання на робочій поверхні. Конструкція крісла має підтримувати раціональну позу під час роботи з відеотерміналом і персональним комп’ютером, дозволяти

змінювати положення тіла для зменшення статичного напруження м’язів шийно-плечової та спинної зон і запобігання розвитку втоми працівника. Поверхня сидіння, спинки та інших елементів крісла повинна бути напівм’якою, з покриттям, що не електризується, є протиковзким і повітропроникним, а також забезпечує легке очищення від забруднень [23].

При відсутності регулювання висота робочої поверхні повинна становити 725 мм. Робочі столи повинні мати висоту простору для ніг не менше 600 мм, ширину — не менше 500 мм, глибину — не менше 450 мм і висунення ніжок — не менше 650 мм. Робоче місце має бути обладнане підставкою для ніг шириною не менше 300 мм і глибиною не менше 400 мм, з діапазоном регулювання висоти до 150 мм та можливістю нахилу опорної поверхні до 20°.

Відстань від очей користувача до екрана дисплея повинна становити 500–700 мм. Кут огляду має бути в межах 10–20°, але не більше 40°, при цьому кут між верхнім краєм дисплея та лінією зору користувача повинен бути не менше 10°. Найбільш доцільним є розташування екрана перпендикулярно до лінії зору користувача [23].

Відстань робочого місця відносно світлового отвору повинна бути не менше 3 м, при цьому природне світло має потрапляти збоку, переважно зліва. Освітлення суттєво впливає на здоров’я та працездатність людини. Відповідно до вимог ДБН В.2.5-28:2018 встановлені такі нормативи освітлення [23]:

- освітленість штучного комбінованого освітлення — 1500 лк;
- освітленість штучного загального освітлення — 400 лк;
- коефіцієнт природної освітленості для верхнього або комбінованого освітлення — 10%;
- коефіцієнт природної освітленості для бічного освітлення — 3,5%;
- коефіцієнт природної освітленості для верхнього або комбінованого суміщеного освітлення — 3–6%;
- коефіцієнт природної освітленості для бічного суміщеного освітлення — 1,1–2%.

Основними показниками оцінки здорових умов праці є фон, контраст об’єкта на фоні, видимість, індекс відблиску та коефіцієнт пульсації освітлення [24]. Фон характеризується коефіцієнтом відбиття. Контраст об’єкта до фону (К) визначається як відношення яскравості об’єкта (точки, лінії, символу) до яскравості фону. Оскільки праця користувача ПК належить до категорії 1а — легка фізична робота (сидяча робота з енерговитратами до 120 ккал/год), необхідно дотримуватися таких критеріїв: коефіцієнт відбиття фону має бути більше 0,4 (яскравий фон), а при значенні К більше 0,2 контраст об’єкта і фону вважається високим або помірним [24].

У полі зору користувача ПК має бути забезпечений правильний розподіл яскравості. У робочій зоні відношення яскравості екрана до яскравості навколишніх поверхонь не повинно перевищувати 3:1 (ДБН В.2.5-28:2018). Для цього монітор персонального комп’ютера повинен відповідати таким вимогам [24]:

- яскравість екрана — не менше 100 кд/м²;
- мінімальний розмір світлової плями кольорового дисплея — не більше 0,6 мм;
- коефіцієнт контрастності зображення — не менше 0,8;
- низькочастотне тремтіння зображення в діапазоні 0,05–1,0 Гц — не більше 0,1 мм;
- екран повинен мати антиблікове покриття;
- відеомонітор має бути обладнаний поворотною платформою, що дозволяє переміщувати його у горизонтальній і вертикальній площинах у межах 130–220 мм та змінювати кут нахилу на 10–15°.

Коефіцієнт відбиття світла матеріалами та обладнанням у приміщенні має важливе значення для забезпечення належного рівня освітлення. Рекомендовані коефіцієнти відбиття становлять: для стелі — 60–70%, для стін — 40–50%, для підлоги — близько 30%, для інших поверхонь — 30–40%.

ВИСНОВКИ

У ході дослідження було створено повноцінну систему прогнозування інсульту на основі метаансамблевого підходу. Початковий аналіз даних показав значний дисбаланс класів: лише близько 4,8% записів належали до випадків інсульту. Така нерівномірність вимагала попереднього балансування (через SMOTE або ваги класів), що суттєво вплинуло на стабільність навчання базових моделей та точність прогнозів.

Проведені експерименти з окремими алгоритмами класифікації продемонстрували, що найкращий результат серед базових моделей показав CatBoost. Його показники становили Accuracy 0.96, F1-score 0.73 та ROC-AUC 0.92, що свідчить про здатність моделі якісно працювати з даними складної структури, включно з категоріальними ознаками. Інші алгоритми — XGBoost, LightGBM, Logistic Regression та Random Forest — також забезпечили стабільні результати, але загалом залишилися у діапазоні F1-score 0.62–0.70 та ROC-AUC 0.87–0.91, тобто поступалися найкращій моделі.

Після цього було реалізовано метаансамблеву модель, що поєднує прогнози всіх базових алгоритмів і формує новий простір ознак для метамоделі. У ролі метамоделі найкраще себе проявила Logistic Regression, яка забезпечила природну інтерпретованість ваг та стійкість до переобучення. Підсумковий результат метаансамблю виявився істотно вищим, ніж результати будь-якої окремої моделі: F1-score зріс до 0.81, ROC-AUC до 0.945, а Accuracy — до 0.97. Таким чином, у порівнянні з найкращим базовим класифікатором покращення становило +0.08 для F1-score та +0.025 для ROC-AUC.

Додатково було проведено аналіз динаміки навчання метаансамблю протягом 10 епох. Результати показали послідовне підвищення якості: F1-score збільшився з 0.63 до 0.81, а ROC-AUC — з 0.88 до 0.945. Це підтверджує, що модель коректно навчається на метатренувальному наборі та здатна ефективно узгоджувати прогнози різних алгоритмів.

Важливу роль відіграла й оптимізація гіперпараметрів базових моделей. Завдяки використанню GridSearchCV та RandomizedSearchCV, середні значення F1-score підвищилися на 5–12%, а ROC-AUC — на 2–4%. Оптимізація гіперпараметрів виявилася критичною для досягнення високої точності, особливо в умовах медичних задач, де навіть невелике покращення якості може мати суттєве практичне значення.

Загалом результати підтверджують, що метаансамблевий підхід є значно ефективнішим за використання окремих класифікаторів. Він зменшує кількість хибнонегативних рішень, забезпечує вищу стабільність та здатність до узагальнення, що є особливо важливим у задачах прогнозування інсульту. Розроблена модель може бути використана як основа для інтеграції у медичні інформаційні системи підтримки клінічних рішень та подальшого розвитку інтелектуальних аналітичних платформ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Fedesoriano. Stroke Prediction Dataset (Kaggle). [Електронний ресурс] URL: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset (дата звернення: 17.12.2025).

2. Chereshchuk L. O., Melnykova N. I. Application of machine learning methods for predicting the risk of stroke. Вісник Тернопільського національного технічного університету. 2024. 1(113). С. 27–35. [Електронний ресурс] URL: https://elartu.tntu.edu.ua/bitstream/lib/44686/2/TNTUSJ_2024v113n1_Chereshchuk_L-Application_of_machine_27-35.pdf

3. Gontkovskyi O. I., Kozachko O. V. Оптимізація лікування та профілактики шляхом впровадження нейронної мережі для прогнозування інсульту. XLIII наук.-техн. конф. підрозділів ВНТУ (Вінниця, 2024). [Електронний ресурс] URL: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/44397/13679.pdf

4. Slobodeniuk A. I. Інтелектуальна система діагностування та прогнозування інсульту (кваліфікаційна/дипломна робота). 2024.

5. Bartiuk R. S. Діагностика та прогностичне значення захворювання мілких судин мозку в гострому періоді мозкового інсульту: дисертація PhD (222 «Медицина»). Вінниця, 2023. [Електронний ресурс] URL: https://www.vnmu.edu.ua/downloads/oc/106/dis_Bartiuk.pdf

6. Yurochkin V. V. Програмний додаток для діагностики проявів церебральної хвороби малих судин на МРТ на основі згорткових нейронних мереж: магістерська дисертація. КПІ ім. Ігоря Сікорського, 2023. [Електронний ресурс] URL: https://ela.kpi.ua/bitstream/123456789/54244/1/Yurochkin_magistr.pdf

7. Vrakina K. P. Метод прогнозування серцево-судинних захворювань методами машинного навчання (кваліфікаційна робота). ХНУРЕ, 2024. [Електронний ресурс] URL: https://openarchive.nure.ua/bitstreams/f908fe23-d483-4abb-9556-5bc24ff69228/download

8. Hassan M. M. et al. A Hybrid Machine Learning Approach to Predict the Risk of Stroke. ACM (2022). [Електронний ресурс] URL: https://dl.acm.org/doi/abs/10.1145/3542954.3543020

9. Hassan A. et al. Predictive modelling and identification of key risk factors for stroke using machine learning (Dense Stacking Ensemble). Scientific Reports. 2024. [Електронний ресурс] URL: https://www.nature.com/articles/s41598-024-61665-4

10. Asadi F. et al. The most efficient machine learning algorithms in stroke prediction: review / analysis of studies (2019–2023). 2024. [Електронний ресурс] URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC11443322/

11. Wijaya R. et al. An Ensemble Machine Learning and Data Mining approach for stroke prediction (біомедичні обчислення/аналіз). 2024. [Електронний ресурс] URL: https://www.mdpi.com/2306-5354/11/7/672

12. Kitova K. et al. Stroke Dataset Modeling: Comparative Study of Machine Learning Approaches. Algorithms. 2024. [Електронний ресурс] URL: https://www.mdpi.com/1999-4893/17/12/571

13. Dubey Y. et al. Explainable and Interpretable Model for the Early Detection / stroke prediction (із посиланням на Kaggle dataset). Diagnostics. 2024. [Електронний ресурс] URL: https://www.mdpi.com/2075-4418/14/22/2514

14. Swain K. et al. Enhancing Stroke Prediction Using LightGBM With SMOTE (огляд/порівняння бустингів та дисбалансу). 2024. [Електронний ресурс] URL: https://assets.cureusjournals.com/artifacts/upload/original_article/pdf/2268/20250210-30756-2rbt5k.pdf

15. Закон України «Про охорону праці». [Електронний ресурс] URL: https://zakon.rada.gov.ua/laws/show/2694-12

16. ДНАОП 0.00-4.15-98 «Про затвердження Положення про розробку інструкцій з охорони праці». [Електронний ресурс] URL: https://zakon.rada.gov.ua/laws/show/z0226-98

17. НПАОП 0.00-4.12-05 «Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці». [Електронний ресурс] URL: https://zakon.rada.gov.ua/laws/show/z0231-05

18. Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями : затв. наказом Міністерства соціальної політики України від 14 лютого 2018 р. № 207. – Офіційний вісник України. – 2018.

19. ДНАОП 0.00-1.21-98 «Про затвердження Правил безпечної експлуатації електроустановок споживачів» [Електронний ресурс] URL: https://zakon.rada.gov.ua/laws/show/z0093-98

20. ДБН В.2.5-28:2018 «Природне і штучне освітлення». [Електронний ресурс] URL: https://zakon.rada.gov.ua/laws/show/z0000-18

21. Ергономічні вимоги для організації робочого місця. Портал споживача. URL: https://www.gpp.in.ua/robota/ergonomichni-vimogi-dlyaorganizatsiji-robochogo-mistsya (дата звернення: 05.06.2023).

22. ДСТУ 8604:2015. Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги. На заміну ГОСТ 12.2.032-78 ; чинний від 2017-07-01. Вид. офіц. Київ, 2015. 9 с. URL: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=71028 (дата звернення: 05.06.2025).

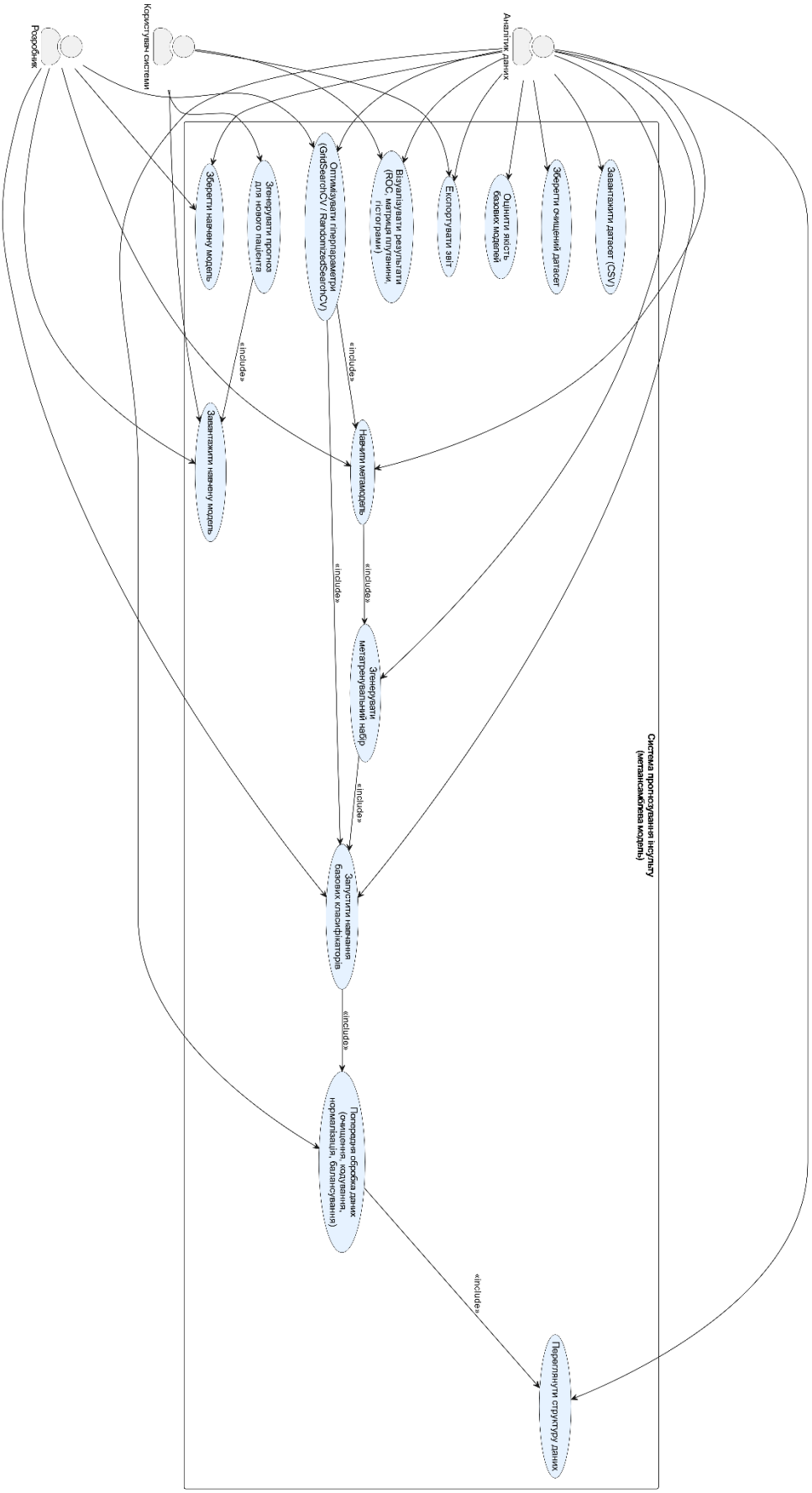
23. Бедрій І.Я., Нечай В.Я. Безпека життєдіяльності. Навчальний посібник. – Львів: Манголія 2006, 2007. 499 с

24. Основи охорони праці. / Під ред. Ткачука К.Н., Халімовського Н.О. – К.: Основа, 2006. 448 с.

25. Методичні вказівки до виконання кваліфікаційної роботи магістра для здобувачів спеціальності 121 – Інженерія програмного забезпечення, всіх форм навчання / укладачі: Михалик Д.М., Цуприк Г.Б., Бревус В.М., Мудрик І.Я. – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2024. – 44 с. URL: https://elartu.tntu.edu.ua/handle/lib/50316

ДОДАТКИ

ДОДАТОК А – Рисунок основної діаграми варіантів використання



ДОДАТОК Б – Теза конференції