Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Інститут комп'ютерних систем
Кафедра інформаційних систем

Попазов Петро Іванович
студент групи АІ-221

ЗВІТ

з виробничої практики

Спеціальність:
122 Комп’ютерні науки
Освітня програма: Комп’ютерні науки

Керівник від НУОП	Керівник від
доцент, к.т.н.	SoftServe
(науковий ступінь, вчене звання)	(назва підприємства, організації, установи)
Бабілунга О.Ю.	
(прізвище та ініціали) (підпис)	(прізвище та ініціали) (підпис)
«____» _____20___ р.	«____» _____20___ р.

Одеса – 2025

ЗМІСТ

ВСТУП

Виробнича практика є важливим етапом підготовки здобувачів першого (бакалаврського) рівня вищої освіти за спеціальністю «122 Комп'ютерні науки», що надає можливість студентам закріпити теоретичні знання, отримані під час навчання, і набути практичних навичок, необхідних для майбутньої професійної діяльності.

Основна мета виробничої практики – ознайомлення студентів з роботою IT-підрозділів підприємств, використанням сучасних комп'ютерних технологій та засобів програмування, а також розвиток вмінь самостійної діяльності в умовах реального виробництва.

Програма виробничої практики розроблена відповідно до навчального плану спеціальності та включає такі ключові завдання:

– знайомство з підприємством, його структурою та рівнем комп'ютеризації;

– закріплення теоретичних знань у сфері інформаційних технологій та програмування;

– виконання індивідуального завдання, яке відповідає профілю підготовки студента;

– розробка та тестування програмного забезпечення, аналіз інформаційних потоків, дослідження методів обробки даних.

В рамках цієї роботи також буде виконано індивідуальне завдання на тему «Розробка та дослідження моделей машинного навчання для попереднього діагностування хвороби Альцгеймера», яке надається з кафедри.

1 АНАЛІЗ СТАНУ КОМП'ЮТЕРІЗАЦІЇ ПІДПРИЄМСТВА

SoftServe (укр. Софтсерв) — українська IT-компанія, що працює у сфері розробки програмного забезпечення та надання консультаційних послуг [1]. Головні офіси компанії розташовані у Львові й Остіні (штат Техас, США), понад 10000 працівників працюють в офісах компанії в Європі, США, Канаді, Латинській Америці, Сінгапурі і Дубаї. SoftServe є однією з найбільших компаній-розробників програмного забезпечення у Центральній та Східній Європі та є другою найбільшою IT-компанією України за кількістю співробітників.

SoftServe займається сервісами повного циклу виробництва: від консультацій, дизайн-сервісів, створення концепту технологічного продукту чи програми і їх розробки. Компанія має значний досвід у розробці програмного забезпечення, хмарних технологіях, штучному інтелекті, великих даних, робототехніці, кібербезпеці, доповненій реальності та дослідженні і розробці (R&D). У R&D-відділі науковці та інженери працюють над інноваційними технологіями, такими як квантові обчислення, сенсорні технології, штучний інтелект для медицини та хімії та інші. У 2021 році SoftServe увійшов до рейтингу 25 найрозумніших компаній України, що ґрунтувався на оцінці R&D відділів. Експертиза компанії також включає роботу з партнерськими платформами та технологіями, зокрема SoftServe є: провідним консалтинговим партнером Amazon Web Services, основним сервісним партнером платформи Google Cloud, партнером Microsoft, NVIDIA та ін.

SoftServe активно використовує передові технології та обладнання для реалізації своїх проєктів у сферах штучного інтелекту, машинного навчання та робототехніки. Компанія є елітним партнером NVIDIA, що дозволяє їй впроваджувати найсучасніші рішення на базі обладнання та платформ цього виробника. Зокрема, SoftServe використовує платформу NVIDIA Jetson для розробки автономних пристроїв та вбудованих застосунків, а також NVIDIA Omniverse для створення фотореалістичних віртуальних середовищ. У своїх

проєктах компанія застосовує технології комп'ютерного зору, цифрових двійників та генеративного штучного інтелекту, що дозволяє створювати інноваційні рішення для різних галузей, включаючи виробництво, логістику та обслуговування клієнтів [2].

На жаль, конкретна інформація про моделі комп'ютерів, які використовує компанія SoftServe у своїй внутрішній діяльності, не є публічно доступною. Однак, відомо, що SoftServe активно співпрацює з провідними технологічними партнерами, такими як Amazon Web Services, Google Cloud, Microsoft та NVIDIA, що дозволяє їй використовувати найсучасніші апаратні та програмні рішення у своїх проєктах. Наприклад, компанія є елітним партнером NVIDIA, що дозволяє їй впроваджувати передові рішення на базі обладнання та платформ цього виробника.

Крім того, SoftServe бере активну участь у розвитку освітніх ініціатив. У 2024 році компанія обладнала комп'ютерну лабораторію в Львівському національному університеті імені Івана Франка 30-ма сучасними комп'ютерами марки Lenovo та моніторами Samsung.

2 ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

2.1 Аналіз існуючих технологій машинного навчання у медичній діагностиці хвороби Альцгеймера

2.1.1 Сучасні підходи до діагностування хвороби Альцгеймера

Хвороба Альцгеймера – це невиліковне неврологічне захворювання, що характеризується прогресуючою втратою когнітивних навичок людини, таких як пам'ять, мислення, здатність приймати рішення та виконувати повсякденні завдання. Хвороба зазвичай починається з незначного порушення пам'яті, але з часом симптоми погіршуються, охоплюючи поведінкові та емоційні зміни.

Першими проявами хвороби Альцгеймера можуть бути: втрата пам’яті, труднощі при виконанні знайомих завдань, проблеми з мовою, дезорієнтація в часі і місці, погане або слабке судження, проблеми з абстрактним мисленням, неправильне розміщення речей, зміни в настрої або поведінці. Захворювання супроводжується соціальною та професійною дезадаптацією [3].

Виявлення хвороби Альцгеймера (далі – ХА) потребує комплексного підходу, що включає клінічні оцінки, інструментальні методи діагностики та лабораторні аналізи. Першим етапом є оцінка лікарем симптомів пацієнта, таких як порушення пам’яті, дезорієнтація у просторі та часі, труднощі з мовою та зміни в поведінці. Під час обстеження можуть застосовуватися когнітивні тести, зокрема Мінімальний тест на психічний стан та Монреальська шкала когнітивної оцінки, які допомагають оцінити ступінь порушень. Для більш детального обстеження використовуються нейровізуалізаційні методи, такі як магнітно-резонансна томографія (МРТ) та комп’ютерна томографія (КТ), що дозволяють виявити атрофію мозку або виключити інші патології. Позитронно-емісійна томографія (ПЕТ) може допомогти виявити накопичення амілоїдних бляшок у мозку.

Додатково проводяться лабораторні аналізи для оцінки рівнів амілоїду та тау-білків у спинномозковій рідині, а також аналізи крові для виключення інших причин когнітивних порушень, таких як дефіцит вітаміну B12 чи порушення функції щитоподібної залози. У деяких випадках, особливо при ранньому розвитку хвороби, може бути рекомендоване генетичне тестування на наявність мутацій генів APOE-e4, PSEN1 та PSEN2. Важливою частиною діагностики є психіатрична оцінка для виключення депресії чи тривожних розладів, які можуть мати схожі симптоми. Завдяки ранньому виявленню хвороби Альцгеймера стає можливим своєчасне розроблення ефективного плану терапії та підтримки для пацієнта й його сім'ї.

2.1.2 Огляд методів машинного навчання для аналізу медичних даних

Для галузі охорони здоров’я алгоритми машинного навчання є особливо цінними, оскільки вони можуть допомогти зрозуміти та надати якісь висновки за аналізом величезних обсягів медичних даних, які щодня генеруються в електронних медичних записах. Використання машинного навчання в медицині, може допомогти знайти закономірності в медичних даних, які неможливо було б знайти вручну або це б зайняло велику кількість ресурсів.

Як машинне навчання використовується в охороні здоров'я? Наприклад, моделі можуть аналізувати зображення сітківки, щоб виявляти діабетичну ретинопатію, прогнозувати серцево-судинні ризики за допомогою електронних медичних записів або допомагати в ранньому виявленні ракових пухлин за допомогою візуалізації [4].

Традиційно діагностика захворювання покладалася на досвід медичного працівника та певною мірою на його інтуїцію, а також на наявні тестиле оскільки машинне навчання може швидше аналізувати величезні обсяги даних і визначати тонкі закономірності та кореляції набагато точніше, ніж люди. Наприклад, аналізуючи індивідуальні дані пацієнта, включаючи фактори способу життя,

генетичну схильність і попередні захворювання, машинне навчання використовується для прогнозування ймовірності розвитку у пацієнта певної хвороби, її початку та прогресування, це дає змогу медичному персоналу час та можливість для надання індивідуальної допомоги та впровадження профілактичних заходів і стратегій моніторингу до того, як хвороба стане симптоматичною, що покращує результати для пацієнтів [5].

Оскільки зростає потреба в персоналізованих ліках і методах лікування, також зростає потреба в «розумніших» медичних записах. Завдяки ML заклади охорони здоров'я можуть користуватися медичними записами десятиліть, не витрачаючи час на їх аналіз. Оскільки ML прагне економити час, гроші та зусилля людей, розмір його ринку з роками зростає в галузі охорони здоров'я. Grand View Research передбачає, що ринок електронних медичних записів для амбулаторного та лікарняного використання зросте в ціні до 2027 року. Медичні записи також мають бути розумнішими, щоб надавати кращі діагнози, клінічні пропозиції щодо лікування тощо [6].

Методи класифікації, такі як логістична регресія (Logistic Rregression) та дерева рішень (Decision Trees), використовуються для визначення наявності чи відсутності захворювання. Метод Support Vector Machines (SVM) ефективний для складних медичних задач, зокрема для аналізу генетичних даних. Методи регресії, включаючи лінійну та поліноміальну регресію, дозволяють прогнозувати медичні показники, такі як рівень глюкози в крові або ризик серцевих захворювань.

Методи кластеризації, такі як K-Means та DBSCAN, дозволяють групувати пацієнтів на основі схожих медичних параметрів або виявляти аномалії у даних. Байєсові методи, зокрема наївний Байєс та байєсові мережі, допомагають моделювати причинно-наслідкові зв'язки між симптомами та діагнозами. Ансамблеві методи, такі як Random Forest та Gradient Boosting, демонструють високу точність у прогнозуванні медичних результатів і класифікації захворювань.

Завдяки ML сфера охорони здоров'я підвищує точність діагностики, запроваджує персоналізований підхід до лікування та значно скорочує час

витрачений для аналізу даних. Машинне навчання відкриває широкі перспективи для розвитку медицини, покращуючи якість обслуговування пацієнтів та підвищуючи ефективність клінічної практики.

2.1.3 Використання штучного інтелекту та нейромереж у медицині

Глибоке навчання, підмножина машинного навчання, використовує нейронні мережі з кількома рівнями для розуміння складних шаблонів у даних. У сфері охорони здоров'я глибоке навчання продемонструвало надзвичайний успіх у інтерпретації медичних зображень, таких як рентгенівські промені, МРТ-сканування та патологічні слайди, часто досягаючи точності, порівнянної з точністю експертів-людей або перевершуючи її. Здатність моделей глибокого навчання автоматично вивчати представлення функцій з даних без необхідності ручного вилучення функцій робить їх особливо придатними для завдань, де людині важко вказати відповідні функції.

Штучний інтелект набирає популярності, як інструменти скринінгу для широкого кола галузей медицини, включаючи неврологію. У цій галузі моделі штучного інтелекту використовувалися для локалізації інсультів, ідентифікації епілептичних патернів та обчислення патернів на ПЕТ-зображенні (Позитронно-емісійна томографія). Крім того, ШІ можна використовувати для діагностики та лікування ХА. Наприклад, за допомогою алгоритмів машинного навчання, можна навчити моделі для прогнозування пацієнтів з найбільшим ризиком розвитку ХА на таких демографічних даних, як вік, стать і фактори способу життя (харчування, вживання спиртних напоїв, паління, фізичне навантаження).

Глибоке навчання, представлене конволюційними нейронними мережами (CNN), є незамінним для аналізу медичних зображень, таких як МРТ та КТ, тоді як рекурентні нейронні мережі (RNN) застосовуються для аналізу часових рядів, наприклад, електрокардіограм.

Наприклад, у радіології моделі машинного навчання навчені досліджувати медичні зображення, такі як рентгенівські промені, магнітно-резонансна томографія та комп'ютерна томографія, виявляючи аномалії, такі як пухлини чи ураження, з надзвичайною точністю. Ці моделі особливо корисні для ранньої діагностики таких захворювань, як рак або серцево-судинні захворювання, де раннє виявлення має вирішальне значення для успішних результатів лікування та виживання пацієнтів.

Якщо дані мають природну та незмінну структуру суміжності, наприклад зображення, «згорточна» нейронна мережа (CNN) може стати у нагоді, підкреслюючи локальні зв'язки, особливо на ранніх рівнях моделі. Коли дані мають сильний часовий компонент (наприклад, часові ряди), рекурентні нейронні мережі (RNN) можуть моделювати часові послідовності подій. Проте як RNN, так і CNN важко сприйняти залежності між точками даних, які часово або просторово віддалені одна від одної. ANN навчаються різними способами за допомогою варіацій алгоритму зворотного поширення. Типовим сценарієм є парадигма навчання під наглядом, коли моделі вчаться зіставляти вхідні дані (наприклад, рентгенівське сканування грудної клітки або групу результатів лабораторних досліджень) із міткою (наприклад, діагноз пневмонія чи прогноз початку метаболічного стану) [7].

Неконтрольоване навчання використовується, щоб робити висновки з наборів даних, що складаються з вхідних даних без будь-яких позначених анотацій. У медицині метою часто є групування пацієнтів відповідно до їхніх клінічних характеристик, щоб ідентифікувати нові підтипи та фенотипи захворювань (а саме, стратифікацію пацієнтів), що може інформувати про більш персоналізовану клінічну допомогу або забезпечити шляхи для майбутніх досліджень. Як приклад, неконтрольоване навчання використовувалося для субфенотипування пацієнтів із спричиненим сепсисом ГПН таким чином, щоб інформувати про основну фізіологію та смертність пацієнтів [8].

Нещодавно глибоке навчання було застосовано для обробки агрегованих EHR, включаючи як структуровані (наприклад, діагноз, ліки, лабораторні тести), так і неструктуровані (наприклад, клінічні примітки з довільним текстом) [9]. Також застосовувалась обробка природної мови (NLP) для отримання цінних ідей із неструктурованих медичних текстів, включаючи клінічні нотатки, звіти про патологію та наукову літературу. Моделі НЛП допомагають у прийнятті клінічних рішень і допомагають визначити потенційну взаємодію ліків або побічні ефекти.

Кілька робіт застосували глибоке навчання для прогнозування захворювань на основі клінічного стану пацієнта. Лю та ін. [10] використовували чотиришарову CNN для прогнозування застійної серцевої недостатності та хронічного обструктивного захворювання легень і показали значні переваги порівняно з базовими показниками. RNN із прихованими блоками довготривалої короткочасної пам’яті (LSTM), об’єднанням і вбудовуванням слів використовувалися в DeepCare [11], наскрізній глибокій динамічній мережі, яка визначає поточні стани хвороби та прогнозує майбутні медичні результати. Чой та ін. [12] для розробки Doctor AI, наскрізної моделі, яка використовує історію пацієнта для прогнозування діагнозів і ліків для наступних зустрічей. Оцінка показала значно кращу запам’ятовуваність, ніж дрібні базові лінії, і хорошу можливість узагальнення завдяки адаптації отриманої моделі від однієї установи до іншої без втрати суттєвої точності.

2.2 Розробка моделей машинного навчання для попереднього діагностування хвороби Альцгеймера

2.2.1 Обґрунтування вибору та попередня обробка медичних даних пацієнтів із хворобою Альцгеймера

Для навчання моделей було обрано дата сет з веб-сайту Kaggle [13]. Цей набір даних містить 2150 записи пацієнтів з різних країн, різних національностей, віку і т.д., які дають уявлення про фактори ризику хвороби Альцгеймера, включає

демографічні, медичні та когнітивні тести, а також параметри способу життя з цільовою змінною «Діагноз» (1389 пацієнтів діагностовано негативно, а 760 – позитивно). Доступні наступні ознаки:

1. PatientID: унікальний ідентифікатор пацієнта (від 4751 до 6900).

2. Age: Вік пацієнтів (від 60 до 90 років).

3. Gender: Стать пацієнта (0 - Чоловік, 1 - Жінка).

4. Ethnicity: Етнічна приналежність пацієнта.

5. EducationLevel: Рівень освіти.

6. BMI: Індекс маси тіла (від 15 до 40).

7. Smoking: Статус куріння (0 - Ні, 1 - Так).

8. AlcoholConsumption: Щотижневе споживання алкоголю (від 0 до 20 одиниць).

9. PhysicalActivity: Щотижнева фізична активність (від 0 до 10 годин).

10. DietQuality: Оцінка якості харчування (від 0 до 10).

11. SleepQuality: Оцінка якості сну (від 4 до 10).

12. FamilyHistoryAlzheimers: Історія хвороби Альцгеймера в сім'ї (0 - Ні, 1 - Так).

13. CardiovascularDisease: Наявність серцево-судинних захворювань (0 - Ні, 1 - Так).

14. Diabetes: Наявність діабету (0 - Ні, 1 - Так).

15. Depression: Наявність депресії (0 - Ні, 1 - Так).

16. HeadInjury: Наявність черепно-мозкової травми (0 - Ні, 1 - Так).

17. Hypertension: Наявність гіпертонії (0 - Ні, 1 - Так).

18. SystolicBP: Систолічний артеріальний тиск (від 90 до 180 мм рт. ст.).

19. DiastolicBP: Діастолічний артеріальний тиск (від 60 до 120 мм рт. ст.).

20. CholesterolTotal: Загальний рівень холестерину (від 150 до 300 мг/дл).

21. CholesterolLDL: Рівень холестерину ЛПНЩ (від 50 до 200 мг/дл).

22. CholesterolHDL: Рівень холестерину ЛПВЩ (від 20 до 100 мг/дл).

23. CholesterolTriglycerides: Рівень тригліцеридів (від 50 до 400 мг/дл).

24. MMSE: Результат Mini-Mental State Examination (від 0 до 30, нижчі значення вказують на когнітивне порушення).

25. FunctionalAssessment: Оцінка функціонального стану (від 0 до 10, нижчі значення вказують на більше порушення).

26. MemoryComplaints: Наявність скарг на пам'ять (0 - Ні, 1 - Так).

27. BehavioralProblems: Наявність поведінкових проблем (0 - Ні, 1 - Так).

28. ADL: Оцінка виконання повсякденних завдань (від 0 до 10, нижчі значення вказують на більше порушення).

29. Confusion: Наявність сплутаності свідомості (0 - Ні, 1 - Так).

30. Disorientation: Наявність дезорієнтації (0 - Ні, 1 - Так).

31. PersonalityChanges: Наявність змін особистості (0 - Ні, 1 - Так).

32. DifficultyCompletingTasks: Труднощі у виконанні завдань (0 - Ні, 1 - Так).

33. Forgetfulness: Наявність забудькуватості (0 - Ні, 1 - Так).

34. Diagnosis: Діагноз хвороби Альцгеймера (0 - Ні, 1 - Так).

Важливим кроком у роботі з великими даними – це, по-перше, аналіз на наявність пустих значень та дублікатів. Не було виявлено ані дублікатів, ані пустих значень.

StandardScaler стандартизує функції, видаляючи середнє значення та масштабуючи їх до одиниці дисперсії. Математично він виконує:

$$z = \frac{x-u}{s}$$

де x – значення ознаки;

u – середнє значення ознаки;

s — стандартне відхилення ознаки

Іншим інструментом для розуміння вхідних даних є кореляційна матриця. Кореляційна матриця є корисним інструментом для з’ясування того, як різні змінні пов’язані одна з одною. Дивлячись на коефіцієнти кореляції між двома змінними, ми можемо дізнатися, як вони пов’язані і як зміни в одній змінній можуть вплинути

на інші змінні. За результатами аналізу матриці (рис. 1), можна зробити висновок, що ознаки не залежать одна від одною, а діагноз ХА найбільше залежить від когнітивних тестів та скарг пацієнта.

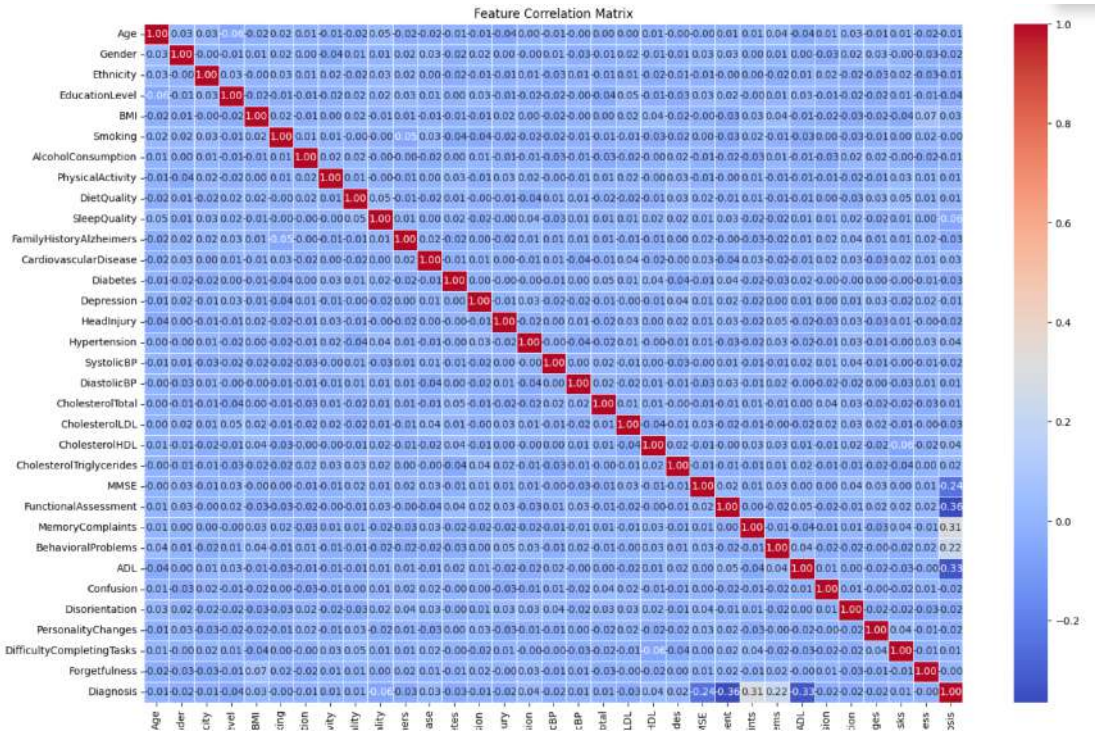

Рисунок 1 – Кореляційна матриця

2.2.2 Обґрунтування вибору та опис моделей машинного навчання для попереднього діагностування хвороби Альцгеймера

Попередня діагностика хвороби Альцгеймера є складною задачею, яка потребує аналізу великої кількості різнорідних даних, включаючи медичні зображення, генетичну інформацію, нейропсихологічні тести та клінічні показники. У цій роботі використовуються записи про хворих із відповідними ознаками та правдивою міткою (чи є у пацієнта ХА). Задача поставлена – бінарна класифікація. Буде використано наступні моделі машинного навчання із учителем:

1. Logistic Regression. Логістична регресія – це алгоритм машинного навчання з учителем, який використовується для завдань класифікації, мета якого полягає в

тому, щоб передбачити ймовірність того, що екземпляр належить до даного класу чи ні. Логістична регресія використовується для бінарної класифікації, використовується сигмоїдну функцію (рис. 2), яка приймає вхідні дані як незалежні змінні та віддає ймовірність між 0 і 1.

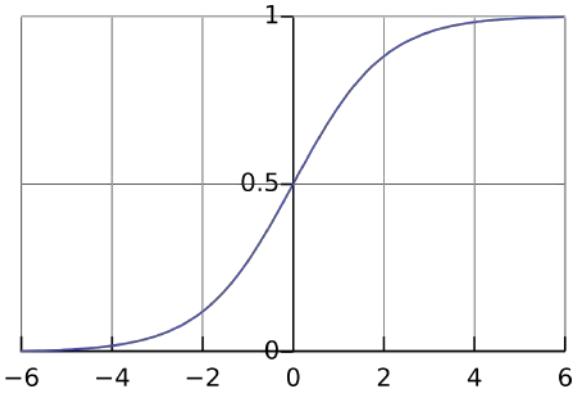

Рисунок 2 – Сигмоїдна функція

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Логістична регресія працює наступним чином: якщо прогнозована ймовірність перевищує порогове значення (зазвичай 0.5), результат класифікується як 1; інакше класифікується як 0. Працює шляхом мінімізації функції втрат функція крос-ентропії.

$$J(\Theta) = \frac{1}{m}[\sum_{i=1}^{m} - y^{(i)}\ log(h_0(x^{(i)})) + (1 - y^{(i)})log(1 - log(h_0(x^{(i)})]$$

Для того, щоб логістична регресія показувала хороші результати, потрібно, щоб:

– кожна ознака не залежить від іншої. тобто немає кореляції між будь-якими вхідними змінними, що і було доведено у розділі з попередньою обробкою даних;

– передбачається, що залежна змінна має бути двійковою або дихотомічною, тобто вона може приймати лише два значення (хворий або ні). Для більш ніж двох категорій використовуються функції SoftMax;

– у наборі даних не повинно бути викидів. У результаті попередньої обробки викидів не було виявлено;

– розмір вибірки достатньо великий (2150 записи).

2. Глибоке навчання. Глибоке навчання (рис. 3) — це підмножина машинного навчання, яка використовує багатошарові нейронні мережі, які називаються глибокими нейронними мережами, для імітації складної здатності людського мозку приймати рішення. Такий метод навчання складаються з кількох шарів взаємопов’язаних вузлів, кожен з яких будується на попередньому шарі для уточнення й оптимізації прогнозу чи категоризації. Цей хід обчислень через мережу називається прямим поширенням (foward propagation). Вхідний і вихідний шари глибокої нейронної мережі називаються видимими шарами. Вхідний рівень – це місце, де модель глибокого навчання отримує дані для обробки, а вихідний рівень – це місце, де робиться остаточний прогноз або класифікація.

Інший процес, який називається зворотним розповсюдженням (back propagation), використовує алгоритми, такі як градієнтний спуск, для обчислення помилок у передбаченнях, а потім коригує ваги та зміщення функції, переміщаючись назад через шари для навчання моделі. Разом пряме та зворотне поширення дають змогу нейронній мережі робити прогнози та виправляти будь-які помилки. З часом алгоритм стає точнішим [14].

Нейронна мережа (далі – НМ) — це математична модель, яка імітує роботу людського мозку для вирішення різних задач машинного навчання, таких як класифікація, регресія, обробка зображень та природної мови. Вона складається з великої кількості простих обчислювальних елементів, званих нейронами, які з'єднані між собою та працюють спільно для аналізу вхідних даних і надання вихідних результатів. Зазвичай складається з вхідного, прихованих та вихідного шарів.

Під час навчання мережа отримує дані та налаштовує ваги (коефіцієнти) зв'язків між нейронами для мінімізації похибки. Використовуються методи, як зворотне розповсюдження помилки (backpropagation) та градієнтний спуск. навчання модель може прогнозувати нові дані, використовуючи отримані ваги.

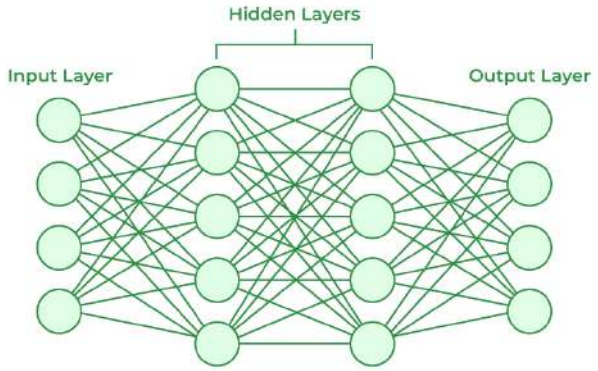

Рисунок 3 – Приклад моделі глибокого навчання

З переваг на лінійною регресією мажна видідити те, що НМ здатна виявити приховані зв’зки між даними, але для ефективного вірішення регресійної задачі потребує набагто більше навчадтних даних, ніж лінійнна регресія.

НМ може добре підійти для вирішення поставленої задачі, так як НМ особливо ефективні для завдань, пов’язаних зі складними взаємозв’язками ознак, оскільки вони можуть знаходити нелінійні зв’язки в даних. Однак є кілька застережень. Однією з головних проблем є розмір наборів медичних даних, які часто невеликі, що у випадку обраного дата сету не є проблемою, адже дата сет налічує 75000+ записів, адже це потенційно може призвести до перетренування (overfitting) НМ.

3. Random Forest. Random (рис. 4) — ансамблевий метод машинного навчання для класифікації, регресії та інших завдань, який працює за допомогою побудови численних дерев прийняття рішень під час тренування моделі й продукує моду для класів (класифікацій) або усереднений прогноз (регресія) побудованих дерев.

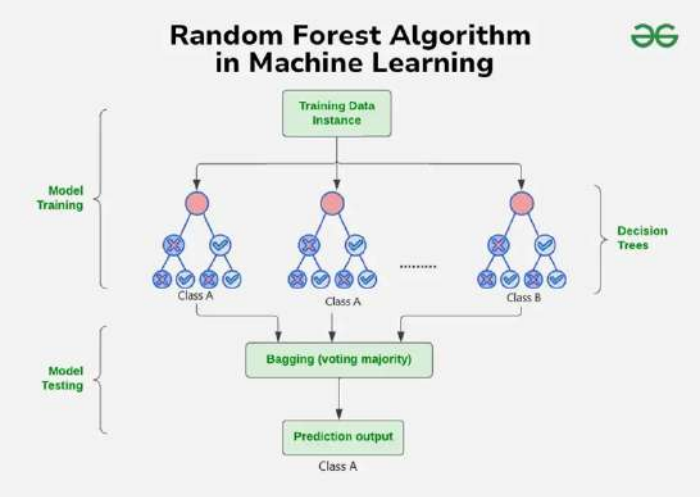

Рисунок 4 – Random Forest

Дерево рішень (рис. 5) — це техніка класифікації, яка будує моделі, які входять до структури дерева при застосуванні до набору даних. Ця техніка розбиває набір даних на підмножини, доки не буде розроблено повне дерево. Результуюча структура дерева складається з гілок, кореневого вузла, вузлів прийняття рішень і листових вузлів. Гілки зображують одну з можливих альтернатив або курсів дій, доступних у кожному вузлі. Кореневий вузол — це найвищий вузол у структурі дерева, який відображає найкращий предиктор. Вузли прийняття рішень складаються з кількох гілок, а листові вузли являють собою класифікацію рішення [15].

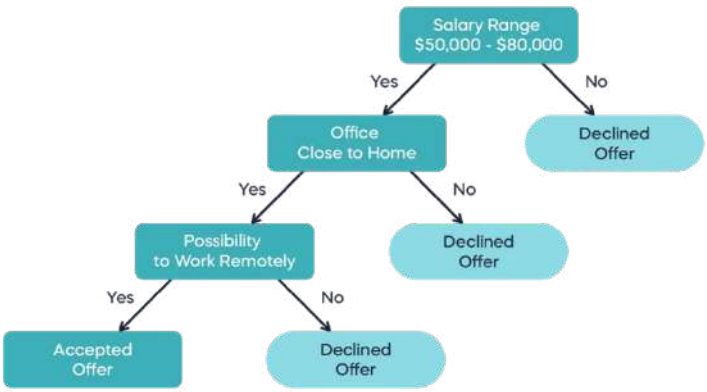

Рисунок 5 – Дерево рішень

З кореневого вузла дерево задає серію запитань «так/ні». Кожне запитання розроблено для поділу даних на підмножини на основі певних атрибутів. Наприклад, якщо перше запитання: «Заробітна платня знаходиться у $50-80 тис.?», відповідь визначить, за якою гілкою дерева слідувати. Залежно від відповіді на кожне запитання ви слідкуєте за різними гілками. Якщо відповідь «Так», рішення іде одним шляхом, якщо «Ні», іде іншим шляхом. Це розгалуження продовжується через послідовність рішень. Дотримуючись кожної гілки, отримується більше питань, які розбивають дані на менші групи.

Алгоритм вибирає розподіл, який максимізує чистоту розбиття (тобто мінімізує домішки). Неформально, домішки є мірою однорідності міток у вузлі (рис. 6)

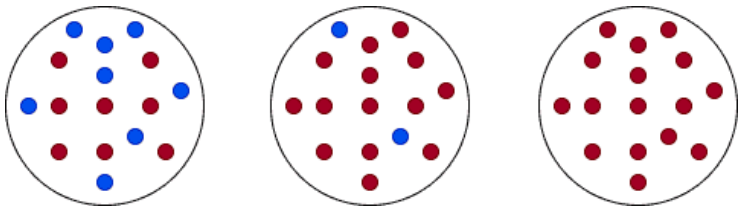

Рисунок 6 – Приклад однорідності множини

Індекс Джині пов’язаний з ймовірністю неправильної класифікації випадкової вибірки. Припустимо, що набір даних T містить приклади з n класів. Його індекс Джині, gini(T), визначається як

$$\text{gini(T)} = 1 - \sum_{\text{j=1}} p_j^2$$

де p_j — відносна частота класу j у T, тобто ймовірність того, що випадково вибраний об’єкт належить до класу j. Індекс Джині може призводити до значень у межах інтервалу [0, 0,5]. Мінімальне значення нуль відповідає вузлу, що містить елементи того самого класу. Якщо це відбувається, вузол називається чистим. Максимальне значення 0,5 відповідає найвищій домішці вузла.

У статистиці ентропія є мірою інформації. Припустимо, що набір даних T, пов'язаний з вузлом, містить приклади з n класів. Тоді його ентропія дорівнює.

$$\text{gini(T)} = -\sum_{\text{j=1}} p_j * \log\,(p_j)$$

де p_j — відносна частота класу j у T. Ентропія приймає значення з [0, 1]. Як і у випадку з індексом Джіні, вузол є чистим, коли ентропія(T) приймає своє мінімальне значення, нуль, і нечистим, коли воно приймає найвище значення, 1 [16].

Під час обчислення приросту інформації за допомогою ентропії алгоритм дерева рішень оцінює, наскільки добре певна функція розбиває набір даних, вимірюючи зменшення невизначеності (ентропії) після поділу. Ентропія кількісно визначає безлад або домішки в даних, причому вищі значення вказують на більш змішаний розподіл класів, а нижчі значення вказують на більш чисті вузли. Спочатку обчислюється ентропія батьківського вузла. Потім для кожного потенційного розбиття обчислюється ентропія дочірніх вузлів, зважена пропорцією вибірок у кожному дочірньому вузлі відносно батьківського. Приріст інформації виходить шляхом віднімання зваженої ентропії дочірніх вузлів від ентропії батьківського вузла. Функція, яка забезпечує найвищий інформаційний приріст, вибирається для поділу, оскільки вона найбільш ефективно зменшує невизначеність і призводить до кращої роздільності класів. Цей процес триває рекурсивно, доки дерево повністю не виросте або не будуть виконані критерії зупинки.

Ансамблеві методи навчання складаються з набору класифікаторів, наприклад, дерева рішень, а їхні прогнози агрегуються для визначення найпопулярнішого результату. Найвідомішими методами ансамблю є беггінг (bagging), також відомий як бутстрапова агрегація, і посилення (boosting) [17].

Беггінг насамперед має на меті зменшити дисперсію (variance) та запобігти перенавчанню шляхом незалежного навчання кількох моделей на різних

підмножинах навчальних даних. Процес передбачає створення кількох наборів даних за допомогою випадкової вибірки із заміною з початкового навчального набору. Потім кожен зразок використовується для навчання базової моделі, наприклад дерева рішень. Остаточний прогноз отримують шляхом усереднення результатів регресійних завдань. Добре відомим прикладом bagging є алгоритм Random Forest, який вводить додаткову випадковість шляхом вибору випадкових функцій для кожного дерева рішень. Пакетування ефективне для стабілізації прогнозів моделі та обробки шумових даних, але воно вимагає, щоб базові моделі не були сильно зміщені для досягнення оптимальної продуктивності.

З іншого боку, boosting зосереджується на зменшенні зміщення (bias) та покращенні точності прогнозування шляхом послідовного навчання моделей. При такому підході кожна наступна модель виправляє помилки, допущені попередньою. Неправильно класифікованим зразкам надається більша вага, щоб надати підказку наступій моделі звернути на них більше уваги. Остаточний прогноз робиться шляхом поєднання зважених результатів усіх моделей. Алгоритми boosting, такі як AdaBoost, Gradient Boosting Machines (GBM), XGBoost, LightGBM і CatBoost, широко визнані своєю високою точністю та здатністю фіксувати складні шаблони в даних.

Random Forest є композицією (ансамблем) безлічі вирішальних дерев, що дозволяє знизити проблему перенавчання та підвищити точність порівняно з одним деревом. Прогноз виходить у результаті агрегування відповідей безлічі дерев. Тренування дерев відбувається незалежно один від одного (на різних підмножинах), що не просто вирішує проблему побудови однакових дерев на тому самому наборі даних, але і робить цей алгоритм дуже ефектвним для застосування в системах розподілених обчислень. [18].

Таким чином Random Forest — це потужний алгоритм для застосування у сфері медицини завдяки його здатності обробляти складні, багатовимірні та шумні дані, які зазвичай зустрічаються у записах пацієнтів. Також здатен знайти нелінійні зв’язки між медичними характеристиками, надає оцінки важливості ознак для

інтерпретації та вирішує проблему перенавчання шляхом усереднення прогнозів з кількох дерев рішень.

4. Ensample Learning. Ансамблеве навчання — це потужна техніка машинного навчання, яка поєднує кілька моделей для підвищення продуктивності.

Накопичування (або накопичуване узагальнення) — це метод ансамблевого навчання, яка поєднує кілька базових моделей для покращення ефективності прогнозування, де у стекуванні використовується метанавчальник (або модель змішування), щоб дізнатися, як найкраще поєднати прогнози базових моделей. Основна ідея полягає в тому, що різні моделі фіксують різні аспекти даних, а модель вищого рівня може навчитися робити кращі кінцеві прогнози, використовуючи свої сильні сторони.

У моделі накопичення базові учні (також звані моделями рівня 0) навчаються незалежно на одному наборі даних. Їхні передбачення потім використовуються як вхідні характеристики для метамоделі (або моделі рівня 1), яка вивчає, як їх оптимально поєднувати. Ця мета-модель зазвичай є простою моделлю, наприклад, логістична регресія, або більш потужною моделлю, як-от XGBoost, залежно від складності проблеми. Ключовою перевагою стекування є те, що воно вивчає оптимальну комбінацію прогнозів, а не просто їх усереднює, як у м’якому чи жорсткому голосуванні.

Стекування особливо добре працює, коли поєднується різноманітні моделі з різними перевагами – комбінація лінійних моделей моделей на основі дерева і моделей глибокого навчання. Кожна базова модель пропонує унікальну перспективу: лінійні моделі фіксують взаємозв’язки в структурованих даних, деревовидні моделі обробляють взаємодії та нелінійність, а нейронні мережі чудово справляються з розпізнаванням складних патернів. Склавши ці моделі в один ряд, можна досягти вищої точності та міцності, ніж будь-яка окрема модель

2.2.3 Розробка моделей машинного навчання для попереднього діагностування хвороби Альцгеймера

Було розпроблено наступні моделі:

1. Logistic Regression. Було розроблено модель, яка виконує завдання класифікації машинного навчання, щоб передбачити діагноз хвороби Альцгеймера за допомогою логістичної регресії, використовуючи мову програмування Python та бібліотеку scikit-learn.

Програмний код починається з імпорту основних бібліотек для обробки даних (pandas, numpy), візуалізації (matplotlib, seaborn) і машинного навчання (scikit-learn). Потім набір даних попередньо обробляється шляхом поділу на функції X і цільові мітки y, де цільова змінна «Діагноз» відокремлена від решти функцій. Виконується поділ усього набору підмножини: 70% для навчання, 15% для крос-валідації та 15% для тестування.

Далі модель логістичної регресії ініціалізується з конкретними гіперпараметрами: регуляризацією L2 (пеналті='l2'), збалансованими вагами класу та 1000 ітераціями для конвергенції. 5-кратна стратифікована перехресна перевірка застосована для оцінки стабільності моделі, запобігаючи перенавчанню. Середня точність перехресної перевірки обчислюється для забезпечення загальної оцінки узагальнення моделі. Нарешті, модель логістичної регресії навчається за допомогою навчального набору (X_train, y_train), готуючи її для оцінки на основі перевірки та тестування наборів даних, що описане у наступному розділі.

2. Глибоке навчання. Програмний код розроблено для створення та навчання моделі глибокого навчання для бінарної класифікації за допомогою TensorFlow і Keras. Він починається з імпорту основних бібліотек, таких як pandas і numpy для обробки даних, tensorflow для операцій глибокого навчання, sklearn для розділення набору даних і метрик оцінки, а також matplotlib.pyplot і seaborn для візуалізації. Щоб забезпечити відтворюваність, сценарій встановлює фіксоване випадкове початкове число за допомогою tf.random.set_seed(42) і np.random.seed(42), що гарантує узгоджені результати під час кількох запусків.

Потім набір даних готується шляхом відділення цільової змінної, «діагноз», від набору функцій. Матриця ознак, X, складається з усіх стовпців, крім цільової

змінної, тоді як цільовий вектор, y, містить лише стовпець діагноз. Набір даних розділено на набори для навчання, перевірки та тестування за допомогою train_test_split.

Створно функцію create_model для створення нейронної мережі. Модель (рис. 7)складається з вхідного шару, за яким слідують три приховані шари, кожен із зменшеною кількістю нейронів: 64, 32 і 16. Кожен шар використовує функцію активації ReLU, яка допомагає в навчанні глибоких мереж, зменшуючи проблему зникнення градієнта. Пакетна нормалізація застосовується після кожного прихованого шару, щоб стабілізувати та пришвидшити навчання, у той час як відсівання використовується на першому шарі зі швидкістю 20%, щоб запобігти перенавчанню. Вихідний рівень містить один нейрон із сигмоїдною функцією.

Для оптимізації навчання включено два зворотних виклики. Перший, EarlyStopping, відстежує втрату перевірки та припиняє навчання, якщо протягом 10 послідовних епох не спостерігається жодного покращення, відновлюючи найкращі ваги моделі до того, як відбудеться переобладнання. Другий, ReduceLROnPlateau, динамічно знижує швидкість навчання в 0,5 рази, якщо втрати перевірки стагнують протягом 5 епох, запобігаючи застряганню моделі в поганих локальних мінімумах. Потім модель компілюється за допомогою оптимізатора Адама з початковою швидкістю навчання 0,001, двійковою крос-ентропією як функцією втрат, а також точністю та AUC як показниками оцінки.

Layer (type)	Output Shape	Param #
dense_8 (Dense)	(None, 64)	2,112
batch_normalization_6 (BatchNormalization)	(None, 64)	256
dropout_2 (Dropout)	(None, 64)	0
dense_9 (Dense)	(None, 32)	2,080
batch_normalization_7 (BatchNormalization)	(None, 32)	128
dense_10 (Dense)	(None, 16)	528
batch_normalization_8 (BatchNormalization)	(None, 16)	64
dense_11 (Dense)	(None, 1)	17

Рисунок 7 – Інформація про модель

Навчання виконується за допомогою функції model.fit, де модель навчається до 75 епох із розміром пакету 64. Дані перевірки включені для моніторингу продуктивності, а попередньо визначені зворотні виклики забезпечують ефективне навчання шляхом ранньої зупинки або коригування швидкості навчання за потреби. Після завершення навчання для тестового набору робляться прогнози за допомогою model.predict(X_test), що генерує оцінки ймовірності. Потім ці ймовірності перетворюються на мітки класу (0 або 1) із використанням порогу 0,5, що дозволяє моделі приймати рішення щодо бінарної класифікації.

3. Random Forest. Щоб забезпечити відтворюваність, встановлюється випадкове початкове число 42. Потім він завантажує набір даних, відокремлюючи ознаки X і цільову змінну y, де y представляє мітки діагнозу. Набір даних розділений на набори для навчання та тестування з використанням співвідношення 80-20 із збереженням розподілу класів через стратифікацію, що гарантує, що навчальні та тестові набори мають однакову пропорцію класів.

Приклад одного з дерев рішень (рис. 8) та приклад текстовий (додаток А).

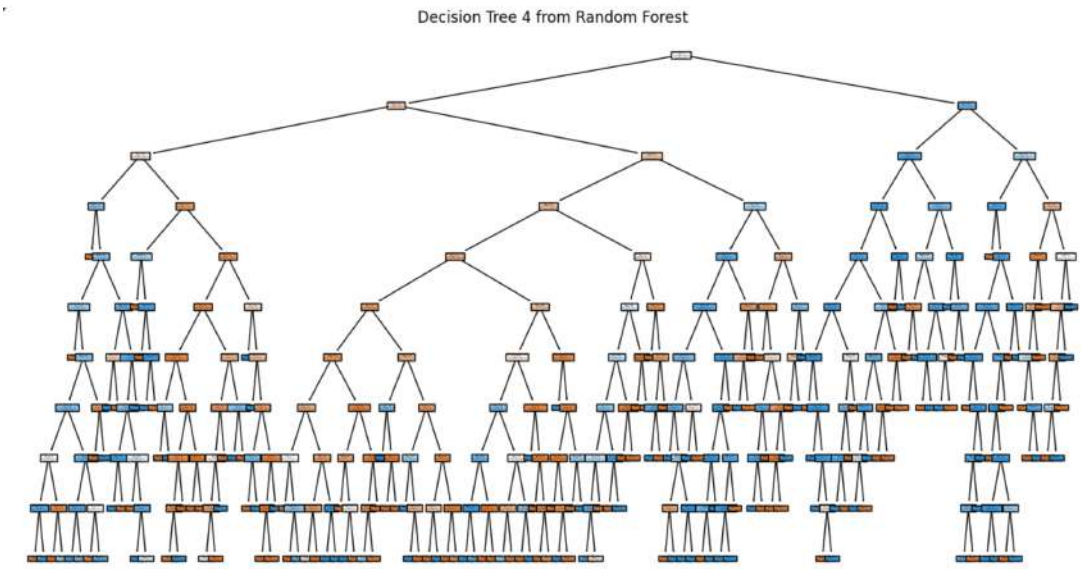

Рисунок 8 – Інформація про модель

Далі визначається класифікатор випадкового лісу зі 100 деревами рішень, максимальною глибиною 10 і class_weight="balanced" для обробки дисбалансу класів. Після перехресної перевірки модель навчається на всьому навчальному наборі даних за допомогою. Після навчання він робить прогнози як для навчального, так і для тестового наборів (y_pred_train і y_pred_test). Модель також передбачає ймовірності класу (y_pred_proba_test), вилучаючи ймовірність позитивного класу для подальшої оцінки. Прогнозовані значення потім можна використовувати для обчислення різних показників ефективності, таких як точність, оцінка ROC-AUC, криві точності-пригадування та матриці плутанини, щоб оцінити класифікатор.

4. Ensemble Learning. Модель (рис. 9) реалізує навчання ансамблю накопичення, де кілька базових моделей, а саме – логістична регресія, Random Forest і нейронна мережа навчаються на наборі даних, а їхні прогнози об’єднуються за допомогою метамоделі XGBoost.

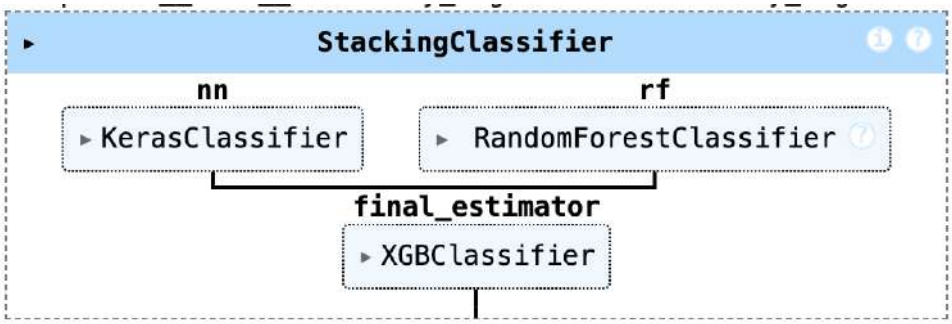

Рисунок 8 – Структура моделі ансамблевого навчання

Набір даних спочатку розбивається на набори для навчання, перевірки та тестування за допомогою стратифікованої вибірки. Базові моделі навчаються незалежно на навчальному наборі, а їхні прогнози служать вхідними функціями для метамоделі. Нейронна мережа, реалізована за допомогою TensorFlow/Keras, має кілька повністю пов’язаних рівнів із нормалізацією. Модель випадкового лісу зі 100 деревами та глибиною 10 фіксує нелінійні шаблони, тоді як логістична регресія з регуляризацією L2 забезпечує міцну основу для класифікації.

Після навчання базових моделей набір даних поєднується з даними перевірки, і новий набір даних створюється з використанням передбачень базових моделей. Ці прогнози стають вхідними даними для метамоделі XGBoost, яка вчиться оптимально зважувати внески кожної базової моделі. StackingClassifier від sklearn використовується для навчання ансамблю, мета-модель навчається на вихідних даних імовірності, а не на необроблених прогнозах. Такий підхід дозволяє метамоделі приймати більш обґрунтовані рішення, особливо у випадках, коли певні моделі більш впевнені у своїх прогнозах, ніж інші.

2.2.4 Тестування моделей машинного навчання для попереднього діагностування хвороби Альцгеймера та оптимізація параметрів

Моделі оцінюються за допомогою кількох показників ефективності та методів візуалізації:

1. Оцінка точності моделі обчислюється як для перевірки, так і для тестових наборів, вимірюється відсоток правильно класифікованих екземплярів.

2. Звіт про класифікацію надає докладні показники ефективності, включаючи точність, запам’ятовування та оцінку F1 для кожного класу.

3. Матриця плутанин показує кількість справжніх позитивних результатів (TP), помилкових позитивних результатів (FP), справжніх негативних результатів (TN) і помилкових негативних результатів (FN).

4. Крива Precision-Recall показує здатність моделі збалансувати між влучністю і відтворенням аналізується за допомогою кривої precision-recall, що допомагає в оцінці продуктивності, особливо при роботі з незбалансованими наборами даних.

5. Крива робочих характеристик демонструє компроміс між частотою істинного позитивного результату (TPR) і коефіцієнтом помилкового позитивного результату (FPR).

Проведено оцінку наступних моделей:

1. Logistic Regression. Модель логістичної регресії демонструє високу загальну ефективність, досягнувши точності на тренувальному сеті 82,92% і точності на тестовому сеті 81,42%. Результати вказують на те, що модель добре узагальнює нові дані, однак варто приділити увагу на класифікаційний звіт (рис. 8).

```
Validation Set Accuracy: 82.92%

Test Set Accuracy: 81.42%

Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.78      0.84       209
           1       0.68      0.89      0.77       114

    accuracy                           0.81       323
   macro avg       0.80      0.83      0.81       323
weighted avg       0.84      0.81      0.82       323
```

Рисунок 8 – Звіт з класифікації

Для класу 0 (негативні випадки) модель досягає високої точності 0,93, що означає, що коли вона передбачає негативний випадок, воно є правильним у 93% випадків. Однак повнота (rcall) становить 0,78, що свідчить про те, що не вдається визначити 22% фактичних негативних випадків. З іншого боку, для класу 1 (позитивні випадки) точність падає до 0,68, що вказує на те, що лише 68% позитивних прогнозів правильні. Однак повнота набагато вище — 0,89, тобто модель успішно визначає більшість позитивних випадків, але ціною деяких помилкових спрацьовувань. Матриця плутанини, на основі якої були зібрані дані наведені вище (рис. 9).

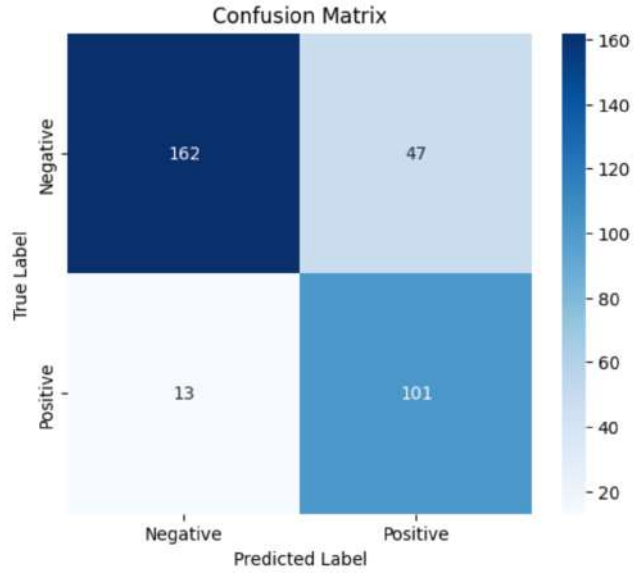

Рисунок 9 – Матриця плутанини

Загальна продуктивність моделі, яка відображається середнім показником F1 макросу 0,81 і зваженим показником F1 0,82, свідчить про добре збалансовану модель. Однак компроміс між влучністю (percission) та повнотю (recall) для позитивного класу вказує на те, що модель надає перевагу мінімізації помилкових негативів над помилково позитивними результатами. Це може бути корисним у сценаріях, коли пропуск позитивного випадку має серйозні наслідки, наприклад медичну діагностику.

Крива PR (рис. 10) вказує на ефективну модель логістичної регресії, хоча може бути місце для вдосконалення в регіоні з високим рівнем відтворенням, де точність падає.

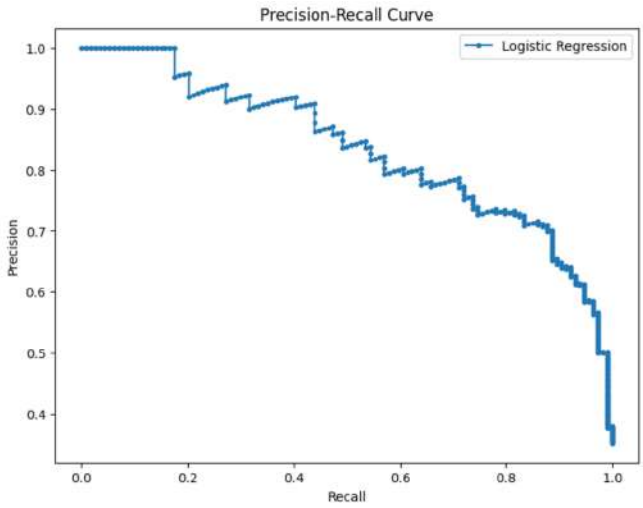

Рисунок 10 – Крива PR

Показник AUC (площа під кривою) 0,90 (рис. 11) вказує на сильну дискримінаційну здатність, тобто модель ефективна для розрізнення позитивних і негативних класів.

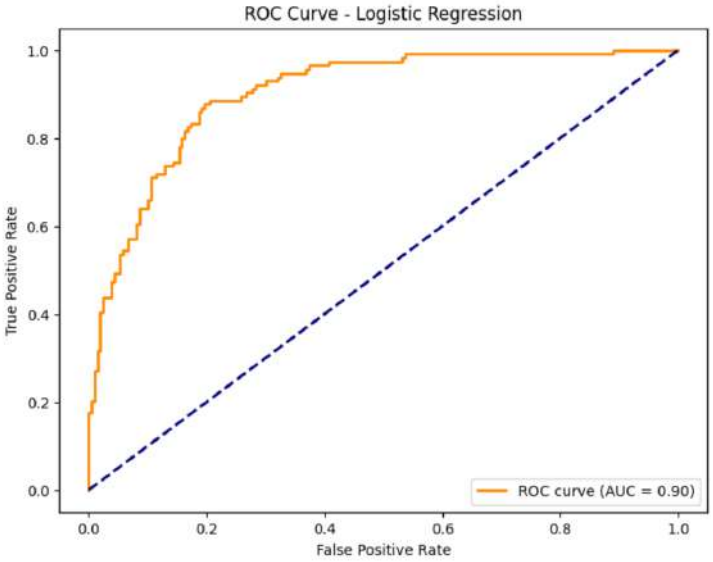

Рисунок 11 – Крива ROC

2. Deep Learning. Модель досягає відносно високої точності 83% (рис. 12). Влучність становить 0,79, тобто коли модель передбачає позитивний випадок (клас 1), вона правильна в 79% випадків. Однак повнота для класу 1 нижче – 0,71, показуючи, що модель пропускає близько 29% фактичних позитивних результатів.

Цей компроміс відображено в оцінці F1 0,75, який врівноважує влучність і повноту. Матриця плутанини, на основі якої були зібрані дані наведені вище (рис. 13).

```
Test Accuracy: 0.83
Precision: 0.79
Recall: 0.71
F1 Score: 0.75

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.90      0.87       277
           1       0.79      0.71      0.75       153

    accuracy                           0.83       430
   macro avg       0.82      0.80      0.81       430
weighted avg       0.83      0.83      0.83       430
```

Рисунок 12 – Звіт з класифікації

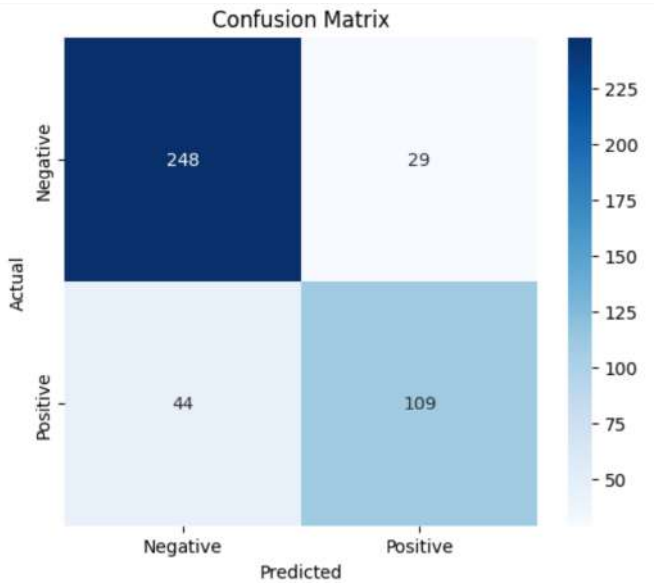

Рисунок 13 – Матриця плутанини

Крива Precision-Recall (рис. 14) вказує на те, що модель підтримує високу влучність на нижчих рівнях повноти, але влучність поступово знижується в міру збільшення повноти. Різке падіння на високих рівнях повноти підкреслює зростання помилкових спрацьовувань.

Показник AUC (площа під кривою) 0,90 (рис. 15) вказує на сильну дискримінаційну здатність, тобто модель ефективна для розрізнення позитивних і негативних класів.

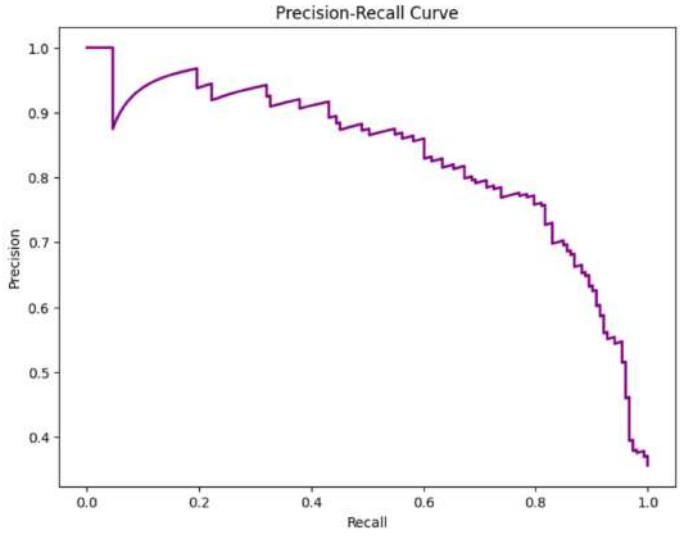

Рисунок 14 – Крива PR

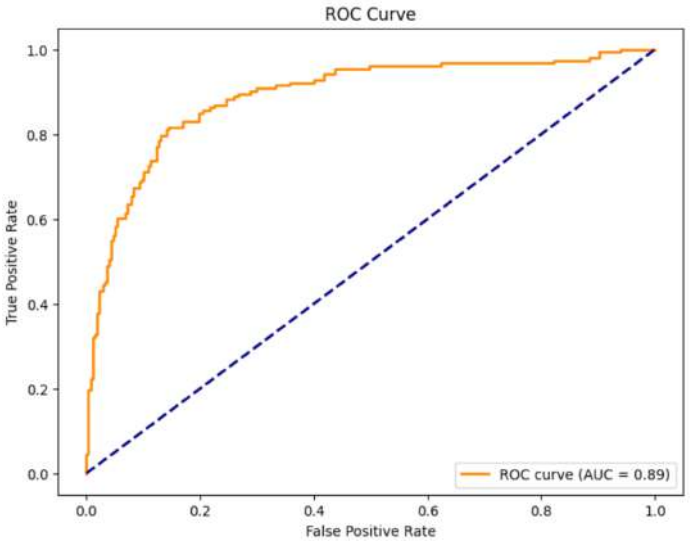

Рисунок 15 – Крива ROC

3. Random Forest. Модель Random Forest демонструє високу точність, досягнувши 93,72% і високого ROC-AUC 0,9432 (рис. 16, 17), що вказує на чудову дискримінаційну здатність. Показник F1 0,9085 відображає хороший баланс між влучністю та повнотою. Крім того, PR-AUC 0,9282 (рис. 16) свідчить про те, що модель добре справляється з дисбалансом класів. Мінімальна дисперсія в AUC

перехресної перевірки (0,9549 ± 0,0080) вказує на стабільну ефективність різних підмножин даних.

```
Cross-Validation AUC: 0.9549 ± 0.0080
Train Accuracy: 0.9878
Test Accuracy: 0.9372
Test ROC-AUC: 0.9432
Test PR-AUC: 0.9282
Test F1-Score: 0.9085

Classification Report on Test Set:
              precision    recall  f1-score   support

           0       0.94      0.97      0.95       278
           1       0.94      0.88      0.91       152

    accuracy                           0.94       430
   macro avg       0.94      0.92      0.93       430
weighted avg       0.94      0.94      0.94       430


Confusion Matrix:
[[269   9]
 [ 18 134]]
```

Рисунок 16 – Звіт з класифікації

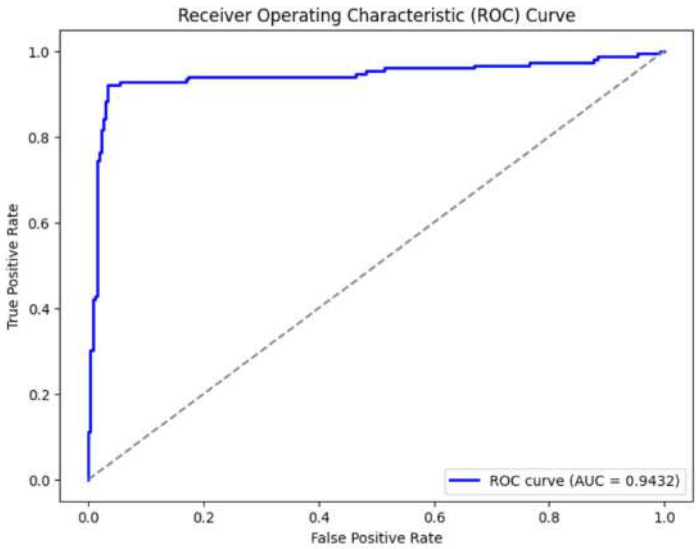

Рисунок 17 – Крива ROC

4. Ensemble Learning. Модель накопичення в цій оцінці досягла найвищої точності (94,43%) (рис. 18). Показники оцінки підкреслюють ефективність моделі в розрізненні позитивних і негативних класів. Оцінка AUC 0,9315 свідчить про те, що модель накопичення має сильну дискримінаційну здатність, займаючи друге місце після випадкового лісу (0,9509). Однак стекування трохи перевершує Random

Forest щодо загальної точності та повноти для класу 1 (є захворення), що робить модель кращим вибором. Модель показала високу влучність (0,97 для класу 0 (не має захворення), 0,91 для класу 1 (є захворення) і повнота (0,95 для класу 0, 0,94 для класу 1).

```
Stacking Ensemble Test Accuracy: 0.9443
Stacking Ensemble Test AUC: 0.9315
Stacking Ensemble Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       209
           1       0.91      0.94      0.92       114

    accuracy                           0.94       323
   macro avg       0.94      0.94      0.94       323
weighted avg       0.95      0.94      0.94       323
```

Рисунок 18 – Звіт з класифікації

Модель ансамблевого навчання показала трохи кращі результати ніж Random Forest, однак, оскільки Random Forest уже був високооптимізованим і ефективним, модель стекування не забезпечила суттєвого підвищення загальної точності або AUC.

2.2.5 Висновки щодо точності попереднього діагностування хвороби Альцгеймера

За результатми дослілжень було сформовано порівняльну таблицю моделей машинного навчання для попереднього діагностування хвороби Альцгеймера (табл. 1).

Таблиця 1 – Порівняльна таблиця моделей машинного навчання

Модель	Точність	AUC	Влучність класа 0	Повнота класа 0	Влучність класа 1	Повнота класа 1
Логістина Регресія	0,8142	0,9025	0,93	0,78	0,63	0,89

Глибоке навчання	0,8019	0,8857	0,85	0,84	0,71	0,74
Random Forest	0,9412	0,9509	0,95	0,96	0,93	0,90
Ensemble Learning (stacking)	0,9443	0,9315	0,97	0,95	0,91	0,94

Перед виконанням завдання було поставлено деякі питання, на які можна надати відповідь після виконання дослідження:

1. «Який метод машинного навчання має найбільшу точність саме у контексті медичних прогнозувань?»

Найбільшу точність показала модель ансамблевого навчання з точністю у 0,9443, але модель випадкового лісу з точністю на тренувальному наборі 98,78% не відстає. Основна причина через яку було отримано невеликий приріст у точності у моделі ансамблевого навчання – «слабкі» моделі логістичної регресії та глибокого навчання.

2. «Який вплив мало конструювання ознак (feature engineering) в прогнозуванні неврологічних захворювань?»

Конструювання ознак майже не вплинуло на точність прогнозування неврологічних захворювань, оскільки приріст становив лише близько 1%. Це може пояснюватися тим, що вихідні медичні показники вже були достатньо інформативними. Крім того, неврологічні захворювання мають складну природу, яку важко виразити через прості ознаки, наявні у датасеті. Також можливим фактором є обмеженість або шум (що є у будь-яких медичних записах) у даних, що робить додаткове конструювання ознак малоефективним.

3. «Яка кореляція між складністю моделі та її ефективністю?»

Кореляція між складністю моделі та її ефективністю не є лінійною. На початкових етапах збільшення складності моделі (наприклад, використання більшої кількості нейронів або шарів у нейромережах) покращило її продуктивність, оскільки модель краще «розуміє» складні патерни в даних. Однак занадто складна модель призвела до перенавчання.

4. "Які є найефективніші методи боротьби з дисбалансом класів?»

За результатами досліджень найефективнішим методом для боротьби з дисбалансом класів виявився використання ансамблевої моделі – Random Forest.

ЗАГАЛЬНІ ВИСНОВКИ

Виробнича практика стала важливим етапом навчального процесу, дозволивши закріпити отримані теоретичні знання та набути практичних навичок у сфері комп’ютерних наук. В ході практики було опановано сучасні методи роботи з інформаційними системами, вдосконалено навички програмування та аналізу даних, а також отримано цінний досвід роботи в реальних умовах.

Виконання індивідуального завдання сприяло розвитку самостійності, відповідальності та вміння вирішувати прикладні завдання в межах професійної діяльності. Аналіз моделей показав, що вибір алгоритму значно впливає на якість прогнозування, але не завжди складніші моделі дають кращі результати. Важливу роль відіграє якість даних, їх збалансованість та належна підготовка.

Більш інтерпретовані моделі можуть демонструвати високу ефективність без надмірної складності, тоді як нейромережі не завжди суттєво покращують результати, а навпаки приносять проблему перенавчання. Конструювання ознак мало незначний вплив, що свідчитить про достатню інформативність вихідних даних. Загалом, успіх моделі залежить від правильного балансу між її гнучкістю, здатністю узагальнювати дані та відповідністю конкретному завданню.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. https://www.bsmu.edu.ua/blog/hvoroba-alczgejmera-suchasni-mozhlyvosti-diagnostyky-i-likuvann (дата звернення 16.02.2025)

2. https://web.archive.org/web/20151221173031/http://softserve.ua/ua/company/about- us (дата звернення 16.02.2025)

3. https://www.softserveinc.com/uk-ua/blog/softserve-add-jetson-platform-nvidia-elite-partner (дата звернення 16.02.2025)

4. https://www.foreseemed.com/blog/machine-learning-in-healthcare (дата звернення 16.02.2025)

5. https://eithealth.eu/news-article/machine-learning-in-healthcare-uses-benefits-and-pioneers-in-the-field/ (дата звернення 16.02.2025)

6. https://www.n-ix.com/machine-learning-in-healthcare/ (дата звернення 16.02.2025)

7. https://pmc.ncbi.nlm.nih.gov/articles/PMC10103223/#B4 (дата звернення 16.02.2025)

8. Chaudhary K, Vaid A, Duffy Á, et al. Utilization of deep learning for subphenotype identification in sepsis-associated acute kidney injury. Clin J Am Soc Nephrol.

9. https://pmc.ncbi.nlm.nih.gov/articles/PMC6455466/#sec5 (дата звернення 16.02.2025)

10. Liu C, Wang F, Hu J, et al. Risk prediction with electronic health records: a deep learning approach. In: ACM International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, 2015, 705–14.

11. Pham T, Tran T, Phung D, et al. DeepCare: a deep dynamic memory model for predictive medicine. arXiv 2016. https://arxiv.org/abs/1602.00357.

12. Choi E, Bahadori MT, Schuetz A, et al. Doctor AI: predicting clinical events via recurrent neural networks. arXiv 2015. http://arxiv.org/abs/1511.05942v11

13. https://www.kaggle.com/datasets/rabieelkharoua/alzheimers-disease-dataset?resource=download (дата звернення 16.02.2025)

14. https://www.ibm.com/think/topics/deep-learning (дата звернення 16.02.2025)

15. https://www.researchgate.net/publication/352647862_Medical_Data_Classification _using_Machine_Learning_Techniques (дата звернення 16.02.2025)

16. https://www.baeldung.com/cs/impurity-entropy-gini-index (дата звернення 16.02.2025)

17. https://www.ibm.com/think/topics/random-forest (дата звернення 16.02.2025)

18. https://habr.com/ru/articles/320726 (дата звернення 16.02.2025)

ДОДАТОК А – ПРАВИЛА ДЕРЕВА РІШЕНЬ

Decision Tree Rules:

```
|--- CholesterolTriglycerides <= 1.37
|   |--- ADL <= -0.10
|   |   |--- PhysicalActivity <= 0.36
|   |   |   |--- BMI <= -1.70
|   |   |   |   |--- DiastolicBP <= -1.64
|   |   |   |   |   |--- class: 1.0
|   |   |   |   |--- DiastolicBP >  -1.64
|   |   |   |   |   |--- class: 0.0
|   |   |   |--- BMI >  -1.70
|   |   |   |   |--- FunctionalAssessment <= -0.08
|   |   |   |   |   |--- CholesterolTriglycerides <= -1.68
|   |   |   |   |   |   |--- SleepQuality <= -0.04
|   |   |   |   |   |   |   |--- class: 0.0
|   |   |   |   |   |   |--- SleepQuality >  -0.04
|   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |   |   |--- CholesterolTriglycerides >  -1.68
|   |   |   |   |   |   |--- MMSE <= 1.06
|   |   |   |   |   |   |   |--- DiastolicBP <= -0.08
|   |   |   |   |   |   |   |   |--- FuxnctionalAssessment <= -0.33
|   |   |   |   |   |   |   |   |   |--- PhysicalActivity <= -0.72
|   |   |   |   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |   |   |   |   |   |   |--- PhysicalActivity >  -0.72
|   |   |   |   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |   |   |   |   |   |--- FunctionalAssessment >  -0.33
|   |   |   |   |   |   |   |   |   |--- CholesterolLDL <= -0.14
|   |   |   |   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |   |   |   |   |   |   |--- CholesterolLDL >  -0.14
|   |   |   |   |   |   |   |   |   |   |--- class: 0.0
|   |   |   |   |   |   |   |--- DiastolicBP >  -0.08
|   |   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |   |   |   |--- MMSE >  1.06
|   |   |   |   |   |   |   |--- CholesterolTriglycerides <= 1.08
|   |   |   |   |   |   |   |   |--- class: 0.0
|   |   |   |   |   |   |   |--- CholesterolTriglycerides >  1.08
|   |   |   |   |   |   |   |   |--- Hypertension <= 0.99
|   |   |   |   |   |   |   |   |   |--- class: 0.0
|   |   |   |   |   |   |   |   |--- Hypertension >  0.99
|   |   |   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |--- FunctionalAssessment >  -0.08
|   |   |   |   |--- BMI <= -1.43
|   |   |   |   |   |--- Forgetfulness <= 0.43
|   |   |   |   |   |   |--- AlcoholConsumption <= 0.19
|   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |   |   |   |--- AlcoholConsumption >  0.19
|   |   |   |   |   |   |   |--- PersonalityChanges <= 0.98
```

```
|  |  |  |  |  |  |  |  |  |--- class: 0.0
|  |  |  |  |  |  |  |  |--- PersonalityChanges >  0.98
|  |  |  |  |  |  |  |  |  |--- class: 1.0
|  |  |  |  |  |  |--- Forgetfulness >  0.43
|  |  |  |  |  |  |  |--- BMI <= -1.50
|  |  |  |  |  |  |  |  |--- class: 0.0
|  |  |  |  |  |  |  |--- BMI >  -1.50
|  |  |  |  |  |  |  |  |--- class: 0.0
|  |  |  |  |  |--- BMI >  -1.43
|  |  |  |  |  |  |--- MMSE <= 0.97
|  |  |  |  |  |  |  |--- AlcoholConsumption <= -1.13
|  |  |  |  |  |  |  |  |--- ADL <= -1.68
|  |  |  |  |  |  |  |  |  |--- class: 1.0
|  |  |  |  |  |  |  |  |--- ADL >  -1.68
|  |  |  |  |  |  |  |  |  |--- CholesterolTriglycerides <= -0.92
|  |  |  |  |  |  |  |  |  |  |--- class: 1.0
|  |  |  |  |  |  |  |  |  |--- CholesterolTriglycerides >  -0.92
|  |  |  |  |  |  |  |  |  |  |--- class: 0.0
|  |  |  |  |  |  |  |--- AlcoholConsumption >  -1.13
|  |  |  |  |  |  |  |  |--- CholesterolTriglycerides <= -0.55
|  |  |  |  |  |  |  |  |  |--- CholesterolTotal <= 0.96
|  |  |  |  |  |  |  |  |  |  |--- class: 1.0
|  |  |  |  |  |  |  |  |  |--- CholesterolTotal >  0.96
|  |  |  |  |  |  |  |  |  |  |--- class: 1.0
|  |  |  |  |  |  |  |  |--- CholesterolTriglycerides >  -0.55
|  |  |  |  |  |  |  |  |  |--- MemoryComplaints <= 0.72
|  |  |  |  |  |  |  |  |  |  |--- class: 0.0
|  |  |  |  |  |  |  |  |  |--- MemoryComplaints >  0.72
|  |  |  |  |  |  |  |  |  |  |--- class: 1.0
|  |  |  |  |  |  |--- MMSE >  0.97
|  |  |  |  |  |  |  |--- SystolicBP <= -1.67
|  |  |  |  |  |  |  |  |--- class: 0.0
|  |  |  |  |  |  |  |--- SystolicBP >  -1.67
|  |  |  |  |  |  |  |  |--- class: 0.0
|  |  |--- PhysicalActivity >  0.36
|  |  |  |--- SleepQuality <= -1.67
|  |  |  |  |--- class: 0.0
|  |  |  |--- SleepQuality >  -1.67
|  |  |  |  |--- FunctionalAssessment <= -0.00
|  |  |  |  |  |--- CholesterolTotal <= -1.63
|  |  |  |  |  |  |--- ADL <= -1.18
|  |  |  |  |  |  |  |--- class: 0.0
|  |  |  |  |  |  |--- ADL >  -1.18
|  |  |  |  |  |  |  |--- class: 1.0
|  |  |  |  |  |--- CholesterolTotal >  -1.63
|  |  |  |  |  |  |--- CholesterolLDL <= -1.15
|  |  |  |  |  |  |  |--- ADL <= -1.21
|  |  |  |  |  |  |  |  |--- class: 0.0
|  |  |  |  |  |  |  |--- ADL >  -1.21
…
```