МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

другий (магістерський)

(рівень вищої освіти)

Дослідження методів прогнозування розвитку діабету серед людей-інсультників у різних умовах життєдіяльності

Виконав:
Студент II курсу, групи ІПЗм-21-2

Нурал Гулієв
(ім'я, прізвище)

Спеціальність 121 Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо – наукова
(освітньо-професійна або освітньо – наукова)

Керівник доц. Олексій НАЗАРОВ
(посада, ім'я, прізвище)

Допускається до захисту

Зав. Кафедри _____ З.В. Дудар

2023р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної Інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Програмна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Гулієву Нуралу Бахадур огли
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження методів прогнозування розвитку діабету серед людей-інсультників у різних умовах життєдіяльності

затверджена наказом університету від «29» 03 2023 р. № 302Ст

2. Термін подання студентом роботи до екзаменаційної комісії «01» 05 2023 р.

3. Вихідні дані до роботи встановлений календарний план роботи, методичні вказівки до оформлення пояснювальної записки.

4. Перелік питань, що потрібно опрацювати в роботі аналіз предметної галузі, огляд наявних математичних моделей, аналіз існуючих нейронних мереж та моделей прогнозування, оптимізація обраних алгоритмів.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	10.11.2022	виконано
2	Постановка задачі	20.11.2022	виконано
3	Здійснення огляду математичних моделей	25.12.2022	виконано
4	Аналіз існуючих моделей прогнозування	30.01.2023	виконано
5	Дослідження обраних алгоритмів	01.03.2023	виконано
6	Написання пояснювальної записки	15.04.2023	виконано
7	Підготовка презентації та доповіді	25.04.2023	виконано
8	Перевірка роботи на плагіат та нормоконтроль	08.05.2023	виконано
9	Захист кваліфікаційної роботи	16.05.2023	виконано

Дата видачі завдання 29 березня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Назаров О.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 90 с., 29 рис., 4 табл., 5 додатків, 50 джер.

БАГАТОШАРОВИЙ ПЕРЦЕПТРОН, НЕЙРОННА МЕРЕЖА, ПРОГНОЗУВАННЯ, ІНСУЛЬТ, ЦУКРОВИЙ ДІАБЕТ.

Об’єктом дослідження є методи прогнозування розвитку цукрового діабету.

Метою роботи є проведення Дослідження методів прогнозування розвитку діабету серед людей в різних умовах життєдіяльності.

У результаті роботи була здійснена підготовча робота для подальшого дослідження та розроблена документація для майбутньої системи прогнозування розвиту хвороби.

MULTILAYER PERCEPTRON, NEURAL NETWORK, FORECASTING, STROKE, DIABETES.

The object of research is methods of predicting the development of diabetes.

The purpose of the work is to conduct research on methods of predicting the development of diabetes among people in different conditions of life.

As a result of the work, preparatory work for further research was carried out and documentation was developed for the future system for predicting the development of the disease.

Умови публікації пояснювальної записки

Я, Гулієв Нурал Бахадур огли
(прізвище, ім’я, по батькові)

студент групи ІПЗм-21-2 здобувач вищої освіти на другому (магістерському) рівні

кафедра програмної інженерії
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему «Дослідження методів прогнозування розвитку діабету серед людей-інсультників у різних умовах життєдіяльності», що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлен з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП

Розвиток інсульту та його особливості впливу залежать від попередніх стану метаболізму головного мозку, церебральної гемодинаміки та нейроімуноендокринної системи. До головних причин, які можуть спровокувати порушення мозкового кровообігу, включають куріння, дисліпідемію, артеріальну гіпертензію та цукровий діабет.

Цукровий діабет – неінфекційна хвороба, яка нанесла удару 425 мільйонам людей, а до 2045 року кількість тільки зростатиме в 1.5 рази. Доведено, що він є незалежним та чинником розвитку інсульту.

Коли у крові забагато цукру, це негативно позначається на артеріях та кровоносних судинах. У людей, які мають це захворювання, частіше розвиваються атеросклерозні бляшки та тромбоутворення, що може спричинити закупорку серця та ішемічний інсульт.

Наявність цукрового діабету підвищує ризик та погіршує перебіг інсульту. За даними Фремінгеймського дослідження, кількість повторних випадків стає більше вдвічі.

Складність проблеми у тому, що недіагностованих випадків стільки ж, скільки й діагностованих, тому приблизно половина через запізнє вставновлення хвороби у людей страждає через нього та складності, які стають його наслідками.

У наш час подібні проблеми почали застосовувати машинне навчання. У 1950-1960 роках були спроби об’єднати існуючі на той час підходи створення нейронних мереж, з використанням яких з’явились можливості розрахунків, які кількісно описують ознаки людського інтелекту, запам’ятовування, аналізу та обробки інформації, що нагадувало роботу людського мозку. Медицина – одна з основних сфер діяльності людини, де різні алгоритми класифікаторів та нейронних мереж набувають все більше популярності з кожним роком. Особливо вони набули популярності в діагностиці захворювань.

Метою курсового проекту є проведення дослідження методів прогнозування розвитку діабету як серед звичайних людей, так і серед інсультників у різних умовах життєдіяльності задля запобігання розвитку інших хвороб.

Галуззю даного спостереження є своєчасне діагностування захворювання, яке важко виявити, але можливо запобігти його розвитку та подальшим ускладненням.

1 ОПИС ПРОБЛЕМНОЇ ГАЛУЗІ

1.1 Аналіз предметної області

Цукровий діабет – одне з найбільш поширених, соціальних хвороб ендокринної системи, поширеність якого перевищує 1% та з часом зростає. На сьогоднішній день воно стало пандемією, бо кількість хворих, на жаль, збільшується. Прогнозовано, що у 2030 році може нараховуватись близько 360 мільйонів хворих на ЦД, а у 2040 – майже в два рази більше, а саме 640 мільйонів. Захворюваність спостерігається у всіх вікових групах, найбільше серед людей віком від 45 до 64 років, та зустрічається в розвинених країнах, де присутня можливість діагностування в більш ранньому віці [1].

Цуровий діабет – це метаболічна група захворювань, яка характеризується порушенням секреції, дії інсуліну та гіперглікемією.

За сучасною класифікацією від 1999 року, яка модифікована в 2019 році, цукровий діабет поділяється на такі види:

- ЦД 1-типу – вид діабету, який виникає через руйнування β-клітин підшлункової залози під час аутоімунного або невизначеного процесу, що і стає причиною дефіциту інсуліну;
- ЦД 2-типу – найчастіший вид, який розвивається при порушенні секреції інсуліну через інсулінорезистентність;
- гібридні форми ЦД;
- гестаційний ЦД;
- ЦД відомої етіології;
- некласифікований ЦД [2].

На жаль, часто зустрічаються комбінації різних видів ЦД.

1.2 Постановка задачі

В наш час Всесвітня організація охорони здоров’я характеризує ЦД як неінфекційне захворювання, але є одним з десяти можливих причин смерті, бо тривалість життя хворих ним зменшується на 25%. Майже 80% випадків смертності спричинено серцево-судинними ускладненнями, а саме інфарктом або інсультом, інвалідністю[3].

Інсульт – це такий стан, при якому вражаються кровоносні судини у головному мозку, що виникає, коли вони блокуються тромбом або розриваються, через що не можуть переносити кисень і поживні речовини.

Ризики виникнення інсульту наступні:

- артеріальний тиск більше норми;
- холестерин більше норми;
- серцевий ритм;
- вага більше норми;
- цукровий діабет;
- надмірні стреси.

Канадські фахівці проводили дослідження: збирали дані про 12200 пацієнтів з діагнозом цукрового діабету 2-го типу, вік яких був більше 30 років, і 9.1% з них було діагностовано різні види інсульту впродовж 5 років. Вони дійшли до висновку, що довгострокова перспектива цукрового діабету призводить до макросудинних наслідків[4].

Відомо, що інсульт буває у людей, в яких цукровий діабет, у віці до 40 років в 3-4 рази та у людей після 40 років в 1.5-2 рази частіше, аніж у недіабетиків, бо вони мають специфічні особливості. В більшості випадків виникають ішемічні варіанти (у 65% атеротромботичний підтип), смертність становить 40.3-59.3%. Також вірогідність виникнення інсультів стає більшою через артеріальний тиск, який більше норми, що може супроводжуватися цереброкардіальним синдромом, розладами свідомості, пневмоніями. Таким чином уражається головний мозок, та повільно починається неврологічний дефіцит, але втрачені функції не повністю

відновлюються. Варто зазначити, у 46% випадків можна відстежити ознаки лейкоареозу в перивентрикулярній зоні, що показує, на скільки уражено церебральні судини[5].

На жаль, у більше 80% людей, хворих інсультом на фоні цукрового діабету, можуть бути рухові розлади.

Сьогодення не змогло повністю знайти причини розвитку інсульту, який виникає через цукровий діабет. Дослідники вважають, що в такому випадку інсульт – клінічний синдром макроангіопатії, зумовленої через порушення процесів обміну вуглеводів [6].

Вище перераховані приклади доводять, що інсульт та цукровий діабет небезпечні не тільки своїми наслідками, а й тим, що можливістю спричинення інших захворювань та один одного. Тому задачею даної кваліфікаційної роботи є дослідження методів прогнозування розвитку діабету серед людей-інсультників у різних умовах життєдіяльності та вибір найоптимальнішої моделі задля цього завдання.

2 МАТЕМАТИЧНЕ ПРЕДСТАВЛЕННЯ

2.1 Багатокритеріальна задача вибору моделі

Для вибору необхідного алгоритму застосовують багатокритеріальну задачу вибору моделі прогнозування, де для медичних спостережень розглядається множина альтернативних моделей, які різняться значеннями своїх характеристик, та необхідно знайти кращий варіант із запропонованих.

При розв’язуванні подібних задач можуть виникати труднощі через неоднозначність вибору. В таких випадках застосовують методи з двох груп: перші призначені для скорочення кількості критеріїв оцінки, при цьому роблять припущення для ранжування значень характеристик та порівняння усіх варіантів, а друга націлена на виключення поганих альтернатив до початку алгоритму порівняння.

Для даного дослідження кращим способом є один із методів першої групи, який включає такі, як метод згортки, метод граничних критеріїв, метод відстані, метод головного критерію.

Метод згортки – спосіб, в якому усі критерії альтернатив стають одним загальним. Найбільш часто вживаними є адитивна, мультиплікативна та максмінна згортки.

Адитивна подається наступним чином:

$$K(x) = \sum_{j=1}^{n} a_j K_j(x), \quad (1)$$

де K(x) – загальний критерій для альтернативи $x \in X$;

$(K_1(x), \dots K_j(x), \dots K_n(x))$ – набір вихідних критеріїв;

n – число, яке описує їх кількість;

a_j – нормуючий множник, вага особливості характеристики альтернативи.

Найкращий із альтернатив розраховується наступним чином:

$$x^* = arg \max_{x \in x} K(x) . \quad (2)$$

Тобто розв'язком є найбільше значення, розраховане за допомогою згортки.

Мультиплікативна згортка обчислюється за формулою:

$$K(x) = П_{j=1}^{n} K_j^{a_j}(x). \quad (3)$$

Формула максмінної згортки:

$$K(x) = \min_j a_j K_j(x). \quad (4)$$

Найкращі розв'язки мультиплікативної та максмінної згорток теж розраховуються за формулою (2).

Метод граничних критеріїв необхідний для розв'язку задач планування та проектування, коли задаються порогові значення критеріїв $k_j(x) \geq k_{j^0};\ j = 1 \ldots, n.$

Формула:

$$K(x) = \min_j \left(\frac{K_j(x)}{K_{j0}(x)}\right). \quad (5)$$

Найкраще рішення розраховується теж за формулою (2).

Метод відстані застосовує додаткову метрику відстань. Припустимо наявної інформації $(K_0, \ldots, K_{0n})$ достатньо для вибору ідеального розв'язку. Розрахуємо для кожної альтернативи відстань до значення максимуму d(x). В такому разі найкраща альтернатива знаходиться за формулою:

$$x^* = arg \min_{x \in x} d(x) \quad (6)$$

Метрикою відстані можуть бути функції Міньковського, Махаланобіса.

Метод головного критерію замінює багатокритеріальну задачу на однокритеріальну, але з обмеженнями, також потрібно при цьому знати порогові значення неголовних критеріїв.

Припустимо, в нас достатньо інформації для виділення головного критерію, який за вагою більший за інших, то краще рішення обчислюється наступним чином:

$$x^* = arg \max_{x \in x} K_0(x), \quad (7)$$

при умові, що значення інших мірил не перевищують зазначених порогових.

Слід звернути увагу, що зі способами першої групи застосовують методи з другої, зокрема принцип Парето. В багатокритеріальних задачах використовують цей метод: найкращу альтернативу по суперечливим критеріям обирають серед усіх можливих, визначених множиною Парето. Тобто, метод з першої групи працює зі звуженою множиною альтернатив, яка включає тільки ті, які важко порівняти за допомогою значень критеріїв: одна альтернатива має більший показник по одному з критеріїв, але менший по іншому [7].

З принципом Парето пов’язаний принцип рівноваги, також відомий під назвою принцип Неша, який дозволяє зменшити множину альтернатив, коли визначається не найкраще рішення, а колективне, що підтримується кожним із розв’язків, що означає, що вони поступаються оцінкам своїх критеріїв.

На жаль, існують ситуації, коли важко знаходити рішення в багатокритеріальних задачах через неконтрольовані та неочікувані впливові параметри, які можуть надходити з навколишнього середовища.

Тут у нагоді може стати метод гарантованого результату, що полягає у визначенні найгіршої реакції, тому неможливо знайти найкраще рішення, але можливо – гарантоване, ймовірність якої достатньо висока [30].

Проаналізувавши методи двох груп, слід обрати необхідний для проведення дослідження. Не можна одразу виключати усі можливі альтернативи, тому методи другої групи не підходять. Та оскільки важко знайти пороги мірил, то краще застосовувати одну із згорток.

Виберемо адитивну, бо мультиплікативна вимагає нормалізації значень від 0 до 1, при якій при 0 матимемо 0, не дивлячись на інші пріоритети критеріїв.

Спочатку необхідно визначити критерії, за якими буде проходити оцінка усіх пропонованих альтернатив, після чого для кожного з них обчислюються ваги – їх важливість у прийнятті рішення, яка з них краща та підходить більше за інших.

Усі критерії відрізняються своїми показниками: вони бувають як якісні, так і кількісні. Даний метод працює з останніми, тому у випадку наявності перших, слід їх замінити відповідними значеннями другого типу.

Після готових кількісних даних можна виключити деякі з альтернатив за принципом Парето, якщо знайдуться такі, що по усім критеріям гірші за інші, а потім необхідно проводити нормалізацію. Оцінки за критеріями відрізняються шкалами своїх значень, тобто, маса вимірюється в кілограмах, швидкість в м/с або в с, тому правильної оцінки значень слід нормувати їх у проміжку значень від 0 до 1. Часто чим вище значення, тим краще, але, звичайно, буває і навпаки, що залежить від умови поставленої задачі.

Одним із способів нормалізації є поділ значення критерію на максимальне, що застосовують у випадку максимізації значень, а в оберненому – одиницю ділять на це значення.

На наступному кроці визначаються вагові коефіцієнти задля ранжування критеріїв. Ваговий коефіцієнт – множник, який визначає важливість того, на скільки даний критерій може вплинути на остаточний вибір.

У нас є n критеріїв: найпотужніший матиме значення параметру n, поділене на n, менш важливий – $n-1$, поділене на n, і так далі. Або можна іншим способом: сумуємо показники кожного критерію та ділимо одиницю на цю суму.

Залишається обчислити значення згортки для усіх альтернатив, а потім порівняти: просумувати для кожної альтернативи суму добутків усіх значень мірил та їх вагових коефіцієнтів. Проведемо такий експеримент для вибору оптимальнішої моделі для поставленої задачі, але спочатку опишемо кожен із можливих варіантів.

2.2 Моделі

2.2.1 Багатошаровий перцептрон

За останні декади діагностування захворювань почали використовувати інтелектуальний аналіз даних, основою якого є алгоритми та математичні підходи. Результативними в прогнозуванні та діагностуванні є саме нейронні мережі [8].

Японські фахівці медичної школи в Кагаве побудували та навчили нейронну мережу, яка буде спроможна практично без помилок прогнозувати у хворих печінково-клітинну карциному та можливі результати резекції печінки.

Троїцький інститут інноваційних та термоядерних досліджень реалізував нейронну мережу, яка була призначена на поради вибору методу лікування базалькоклітинного раку шкіри на основі прогнозу ровитку рецидиву. Кількість людей, які мають білу та тонку шкіру, з базаліомою займають третю частину від усіх хворих онкологічними захворюваннями[9].

Різноманітні можливості використання нейромереж в медицині, тому і різна їх архітектура. Відомий голландський дослідник Герберт Каппен з університету в Німегене досяг значних результатів у прогнозі лікування рака яєчника, але він використовував у своїй роботі не багатошаровий парцептрон, а машини Больцмана – нейромережі, які оцінюють ймовірності настання певних подій або явищ[10].

Краща необхідна нейромережа була обрана при дослідженні широко поширених, що було репрезентовано в науковій конференції 7th International Conference on Computational Linguistics and Intelligent Systems (Scopus) April 20–21, 2023 at National Technical University “Kharkiv Polytechnic Institute” (Kharkiv, Ukraine) [11]. Було запропоновано реалізувати багатошаровий перцептрон.

У наш час більш за все набули популярності саме багатошарові моделі (наприклад, Multilayer Perceptron MLP) із нелінійною функцією активації.

Багатошаровий перцептрон (багатошарова повнозв’язна нейронна мережа) – нейронна мережа, в якій сигнал, який входить, проходить через декілька шарів та стає вихідним. Шари містять вироджені нейрони та можуть не враховуватися в

загальній кількості. Багатошаровий перцептрон містить також проміжні шари, крім вхідного та вихідного(рис. 2.1).

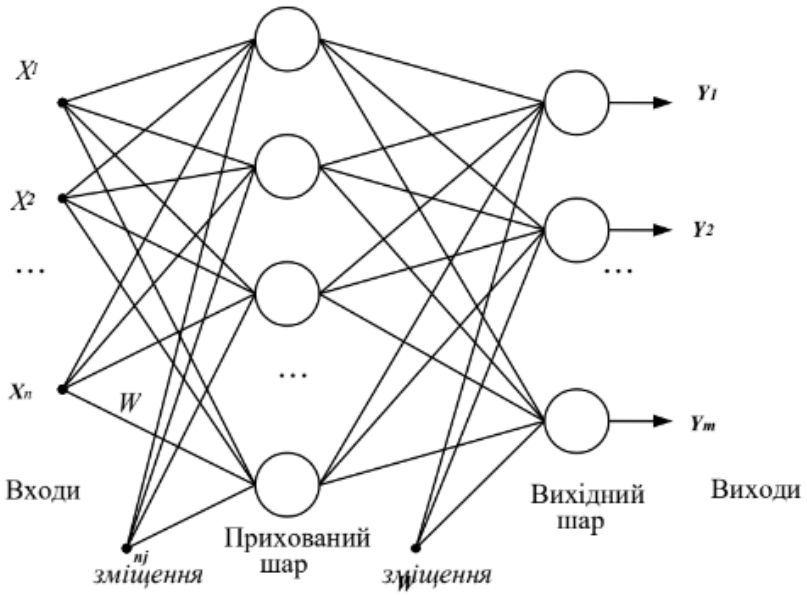

Рисунок 2.1 - Багатошаровий перцептрон (виконано самостійно)

Перед початком навчання мережі ваги та пороги ініціалізуються випадковими малими значеннями. В залежності від того, як проходить навчання, поверхні відгуку елементів зсуваються у потрібне положення, а значення синаптичних зв’язків змінюються.

При побудові моделей задачею є налаштування та коригування ваг протягом навчання.

Рівень активації нейрона – це сума вхідних параметрів, яка є простою функцією входів. Потім проходить перетворення активації за допомогою нелінійної кривої (частіше використовують «сігмавидну»). При комбінації скалярної «сігмавидної» функції та лінійної функції кількох змінних можна знайти характерний профіль «сігмавидного схилу», що видає елементи першого прошарку проміжного шару багатошарової нейронної мережі прямого поширення, а саме MLP. Якщо змінити ваги та порогові значення, поверхня відгуку відповідно зміниться також. Схил стає крутіший при збільшенні значення вагових коефіцієнтів.

Елемент нейронної мережі видає результуючий сигнал на основі сигналів, які надходять, маючи декілька входів із різними ваговими коефіцієнтами. Сигнали сумуються відповідно до синаптичних зв’язків входів, по яких вони надійшли, і сума порівнюється з пороговим значенням: якщо вона менше за його показник, то дорівнює нулю, в протилежному випадку – одиниці[12].

Кожен елемент має один вхід, по якому проходить компонента вхідного вектора. Елементи проміжного шару отримують сигнали від елементів вхідного та мають m входів від нього. Ефективність зв'язків між ними оцінюється вагами, які приймають участь в обчисленні остаточного результау роботи нейронної мережі.

Роботу багатошарового перцептрону відобразимо за допомогою формул:

$$S_{jl} = \sum w_{ijl}\, x_{ijl}, \tag{8}$$

$$Y_{jl} = F(S_{ij}\, \theta_{il}), \tag{9}$$

$$X_{ji(l+1)} = Y_{ij}\,, \tag{10}$$

де i – номер входу,

j – номер нейрона в шарі,

l – номер шару,

x_{ijl} – і-й вхідний сигнал j-го нейрона в шарі l,

w_{ijl} – ваговий коефіцієнт і-го входу j-го нейрона в шарі l,

S_{jl} – сигнал S_j-го нейрона в шарі l,

Y_{jl} – вихідний сигнал нейрона,

θ_{il} - пороговий рівень j-го нейрона в шарі l.

Вагові коефіцієнти впливають на навчання. Ваги обираються так, щоб загальна середньоквадратична помилка даних навчальної вибірки була якомога мінімальною. Існує декілька способів, які допомагають досягти цього. Після того, як перцептрон пройшов етап навчання, проводять тестування – оцінку роботи готової нейронної мережі, для чого навчальну вибірку розділяють на дві частини: першу використовують власне для навчання, а другу з відомими очікуваними

значеннями застосовують для тестування. Відсоток правильно обчислених результатів – оцінка якості написаного алгоритму. Помилкою мережі можна вважати $E^s = \|d^s - y^s\|$ для кожної пари (x^s, d^s). Іноді показник якості навчання обчислюють за формулою:

$$\mathrm{E} = \frac{1}{2} \sum_s \sum_j (d_j^s - y_j^s)^2. \quad (11)$$

Рідше використовують середню відносну помилку, формула якої:

$$\sigma = \frac{1}{SN_o} \sum_s \sum_j \left(\frac{|d_j^s - y_j^s| + 1}{|d_j^s| + 1} - 1\right) 100\%. \quad (12)$$

Її плюс у тому, що її значення, інтервал якого від 0 до 100%, не залежить від кількості елементів навчальної вибірки.

Окрім нейромереж, у роботі також досліджено класифікатор, регресію та побудовано дерево рішень[13].

2.2.2 Метод k-найближчих сусідів (k-nearest neighbour)

Метод k-найближчих сусідів (k-nearest neighbor або kNN) – це алгоритм класифікації на рівні дерева рішень, який застосовується в різних задачах машинного навчання та є одним із найпростіших та найзрозуміліших підходів.

Інтуїтивно він працює наступним чином: об’єкт звертає увагу на сусідів – яких з них більше, таким же є він сам. Метод базується на гіпотезі компактності, яка означає, що якщо метрика відстані використана доречно, то подібні образи завжди знаходитимуться ближче.

У випадку задачі класифікації даний метод спростовує твердження, що клас елемента визначається класом, який найбільш поширений серед його k сусідів, класи яких вже відомі [14].

У випадку задачі регресії об’єкт набуває середнє значення серед значень своїх k сусідів.

Окрім опису алгоритму, слід враховувати також його математичний апарат: Евклідова метрика (Euclidean distance) – найменша можлива відстань між двома точками, яка обчислюється як корінь із квадрату відстані між ними:

$$s = \sqrt{\sum_{i=1}^{n} (x_c - x)^2}, \quad (13)$$

де n – загальна кількість сусідів;

x_c – ознака сусіднього елемента;

x – ознака елемента, для якого обчислюється значення метрики.

2.2.3 Логістична регресія

Класичний метод аналізу логістична регресія – це один із видів множинної регресії, який базується на математичному підрахунку та пошуку зв’язків серед залежної та незалежних змінних, після чого це використовується для прогнозування значення фактору на основі ознак[15].

Регресійна модель подається у вигляді наступної формули:

$$y = F(x_1, x_2, \ldots, x_n). \quad (14)$$

Множинна регресія передбачає лінійну залежність між фактором та ознаками, які впливають на його значення, що видно з формули:

$$y = a + w_1 * x_1 + w_2 * x_2 + \ldots + w_n * x_n, \quad (15)$$

де a – зміщення;

w_i - ваговий коефіцієнт;

x_i – значення ознаки елемента.

Логістична регресія – це статичний апарат, який використовує логічну функцію, яка розраховує значення *y*, як сигмоїдна функція від значення зваженої суми за формулою:

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (16)$$

Після побудови моделі її графік набуває вигляду S-подібної кривої, як показано нижче (рис. 2.2).

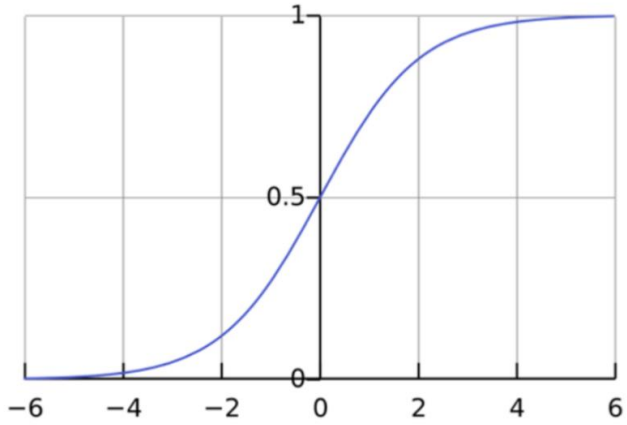

Рисунок 2.2 – Графік логітної функції(виконано самостійно)

Як видно по графіку, функція повертає тільки значення *y* проміжку від 0 до 1, та якщо воно менше визначеного раніше порогу (частіше поріг дорівнює 0.5), то прогнозований фактор – 0, в іншому випадку – 1.

При навчанні модель будується при умові зменшення кількості помилок за допомогою функції втрат Log Loss та коригування синаптичних зв'язків методом градієнтного спуску[16].

У машинному навчанні функція втрат – це функція, яка відображає подію на дійсне число, яке представляє, на скільки обчислене значення фактора моделлю

відрізняється від очікуваного. Задачею оптимізації є мінімізація значення цієї функції, формула якої:

$$L(y_{pred}, y) = -\left(y * log(y_{pred}) + (1-y) * log(1 - y_{pred})\right). \quad (17)$$

Градієнтний спуск – алгоритм оптимізації, в якому використовуються значення, пропорційні протилежному градієнту, для обчислення локального мінімуму функції.

Алгоритм пошуку точки мінімуму наступний:

1) обирається початкова точка M_0;
2) обчислюємо на ній градієнт F;
3) виконуємо антиградієнтний етап:

$$x_1 = x_0 - a_1 * grad\, f(M_0), \quad (18)$$

де $a_1 > 0$, та значення функції нової точки M_1 менше, ніж у точці M_0;

4) якщо попередня умова не виконується, то зменшуємо крок $a_2 = a_1 / 2$ та повторюємо попередні кроки.

Поки не буде знайдено найменше значення функції, або якщо воно почне збільшуватись, алгоритм продовжуватиме працювати[17].

Далі після коригування вагових зв'язків та зміщення застосовуватимемо наступні формули:

$$w_{new} = w_{old} - learningRate * \frac{dL}{dw}, \quad (19)$$

$$a_{new} = a_{old} - learningRate * \frac{dL}{da}, \quad (20)$$

де $\frac{dL}{dw_i}$ – частинна похідна функції втрат за вагами,

$\frac{dL}{da}$ - частинна похідна функції втрат за зміщенням.

Повторюємо етапи аналізу, поки не буде досягнуто бажану якість моделі або визначену кількість епох навчання.

Як бачимо, на відміну від існуючих методів машинного навчання даний алгоритм має наступні переваги:

- простота – регресію набагато легше реалізувати;
- швидкість – дана модель здатна обробляти велику кількість інформації, оскільки вони потребують меншої кількості обчислювальної здатності;
- гнучкість – метод можна застосовувати для пошуку відповідей на питання, коли кількість можливих варіантів результату не важлива;
- наочність – аналіз дозволяє усувати проблеми через просту реалізацію та можливість розуміння роботи внутрішніх процесів.

2.2.4 Дерево класифікації

Дерево рішень (класифікації) – метод аналізу даних, який використовується в статистиці та машинному навчанні. Воно утворюється із таких елементів, як корінь, гілки та листя. Гілками є ознаки, від яких залежне значення цільової функції, яке знаходиться в листях[18].

Після побудови моделі класифікація елемента здійснюється проходом дерева в залежності від значень його атрибутів до одного з листів дерева.

Прикладом дерева класифікації є структура, відображена на рисунку 2.3.

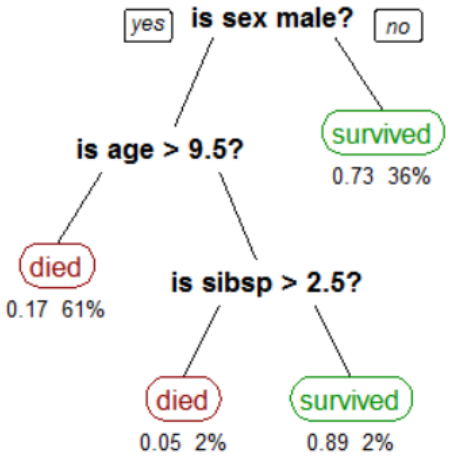

Рисунок 2.3 – Загальний вигляд дерева рішень [19]

Дерево будується на основі тренувальних даних такис чином: обирається атрибут, який стає коренем:

1) для всіх значень цієї ознаки залишаємо ті, які дорівнюють певному значенню;
2) рекурсивно продовжуємо з цього кореня.

Корінь та підкорені визначаються на основі методу Коефіцієнту Джині (Gini Impurity)[20]. Gini Impurity – ймовірність того, що елемент неправильно марковано, яка розраховується за формулою:

$$I_G\,(n) = \,1\, - \,\sum_{i\,=\,1}^{J}(p_i)^2. \qquad (21)$$

Ідея дерева рішень полягає у формуванні запитів, по яким виконується прохід моделі. Атрибути обираються таким чином, щоб дані чіткіше розділялися та вузли містили велику кількість елементів даних, які належить до одного класу. Процес повторюється з використанням «жадібної» рекурсивної процедури, поки не буде досягнуто максимальної глибини, чи кожне листя не міститиме елементи суто одного з можливих класів, або закінчиться кількість елементів тренувальної вибірки.

Важливим фактом є те, що при побудові моделі необхідно регулювати глибину дерева для збереження оптимальності його розміру. При цьому потрібно

зменшувати розмір структури дерева, не зменшуючи при цьому точність її прогнозу[21].

Дана операція виконується зверху вниз або знизу вгору. При першому способі скорочення починається з кореня, а при другому – зменшується кількість листя. Знизу з листя вузол обирається, як найпопулярніший клас.

3 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

Розглянемо практичну частину побудови моделей прогнозування та класифікації. Опишемо експерименти для обраних алгоритмів:

- багатошаровий перцептрон;
- дерево класифікації;
- метод k-найближчих сусідів;
- логістична регресія.

3.1 Пошук даних

Завантажимо датасети з даного джерела та розглянемо кожен з них https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset, які є відповідями на телефонне опитування на тему здоров'я, яке щорічно проводить Центр контролю та профілактики захворювань США (CDC). Ця система нагляду за поведінковими факторами ризику (BRFSS) збирає відповіді від понад 400000 американців про ризиковану для здоров'я поведінку, хронічні захворювання та використання профілактичних послуг. Опитування проводиться щороку з 1984 року, а на Kaggle доступні результати за 2015 рік у форматі csv-файлу.

Таблиця даних першого файлу наведено нижче на рисунку 3.1. Датасет містить 253680 записів.

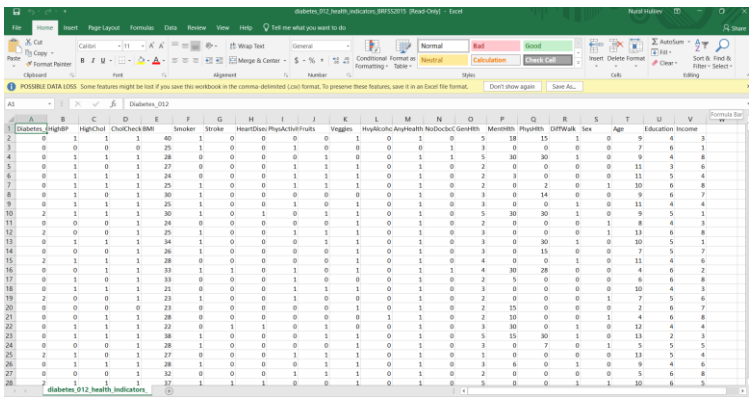

Рисунок 3.1 – Дані першого датасету[22]

Кожен рядок містить наступні атрибути:

- Diabetes_012 - при значенні 0 залежна змінна означає, що пацієнт не хворіє на діабет, а при 1 – результат безуспішний (див. рис. 3.2);

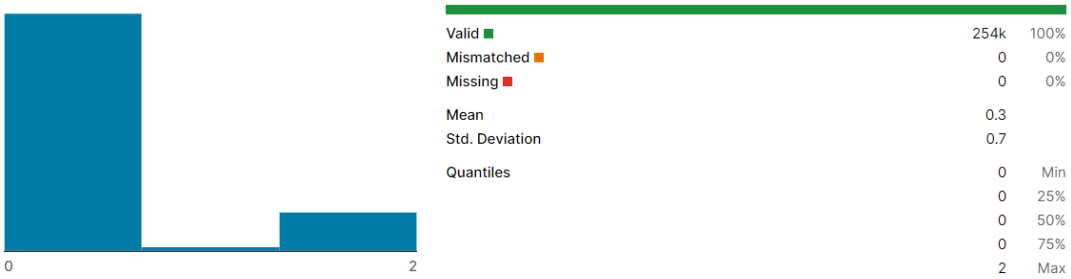

Рисунок 3.2 - Diabetes_012[22]

- HighBP (див. рис. 3.3);

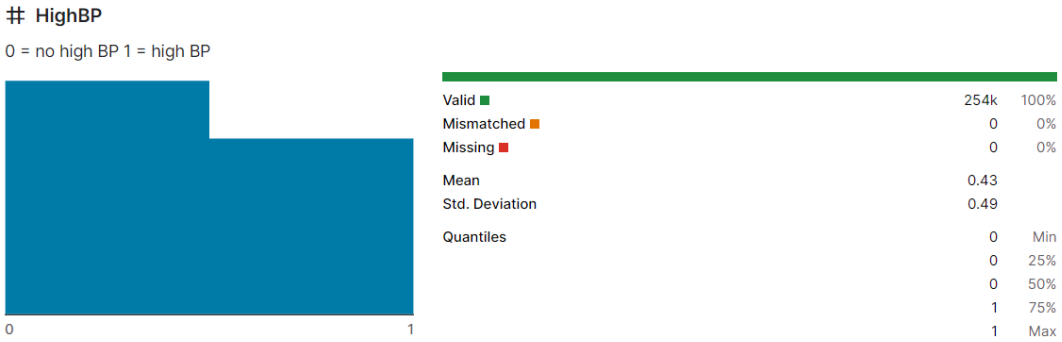

Рисунок 3.3 – HighBP[22]

- HighChol (див. рис.3.4);

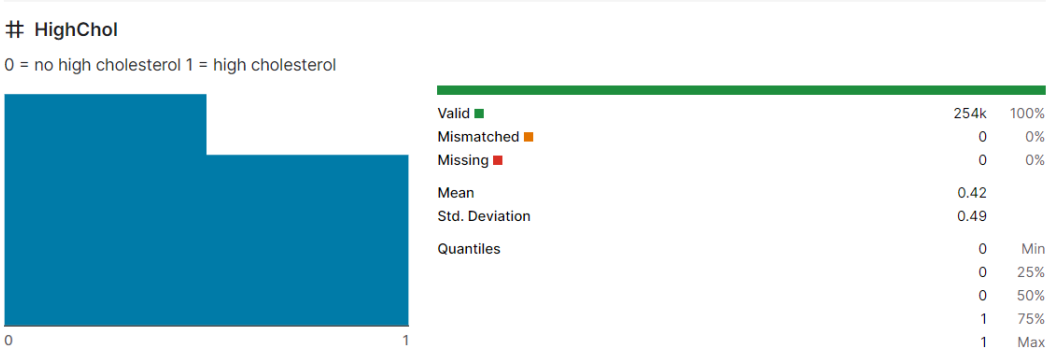

Рисунок 3.4 – HighChol[22]

- CholCheck (див. рис. 3.5);

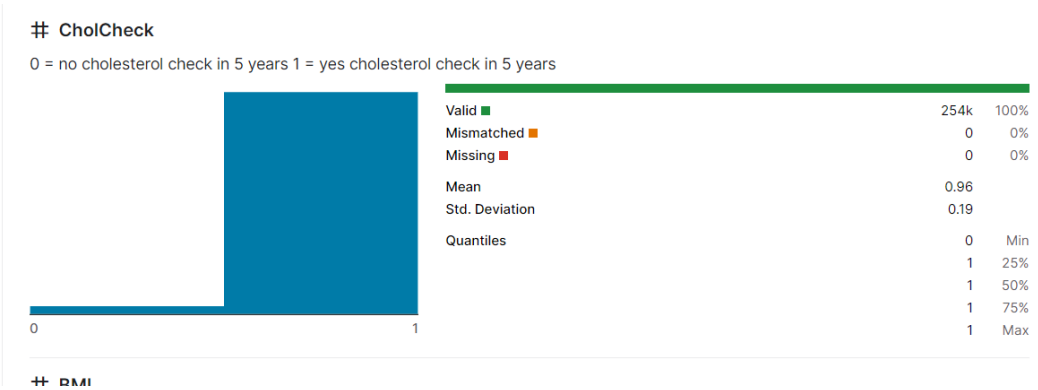

Рисунок 3.5 – CholCheck[22]

- BMI (див. рис. 3.6);

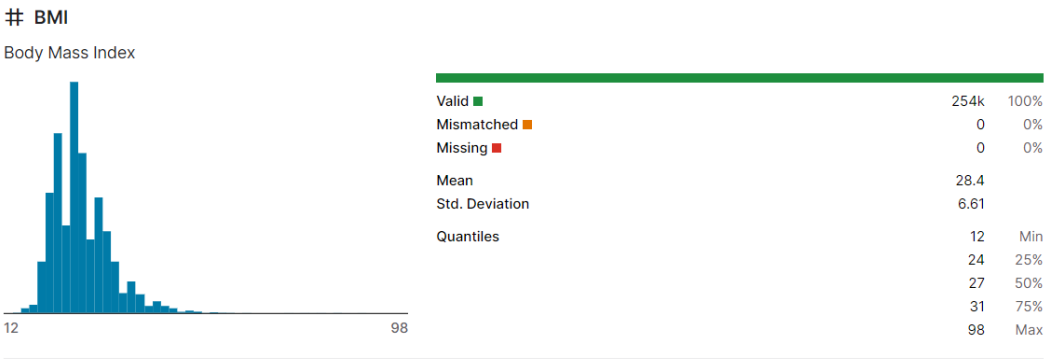

Рисунок 3.6 – BMI[22]

- Smoker (див. рис. 3.7);

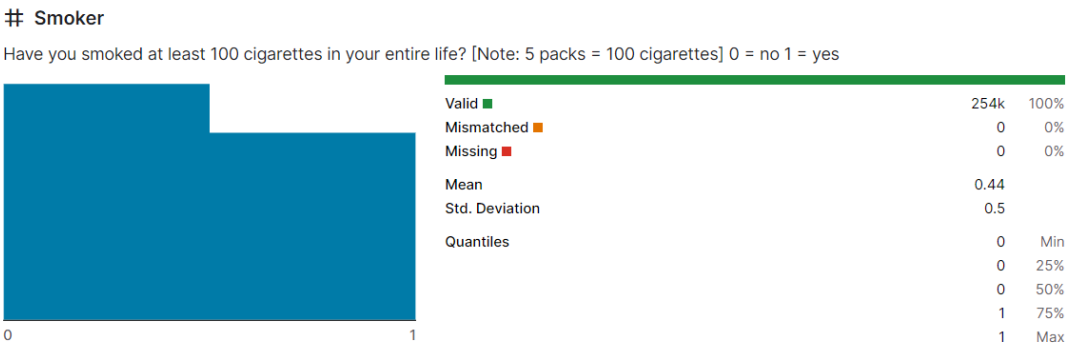

Рисунок 3.7 – Smoker[22]

- Stroke (див. рис. 3.8);

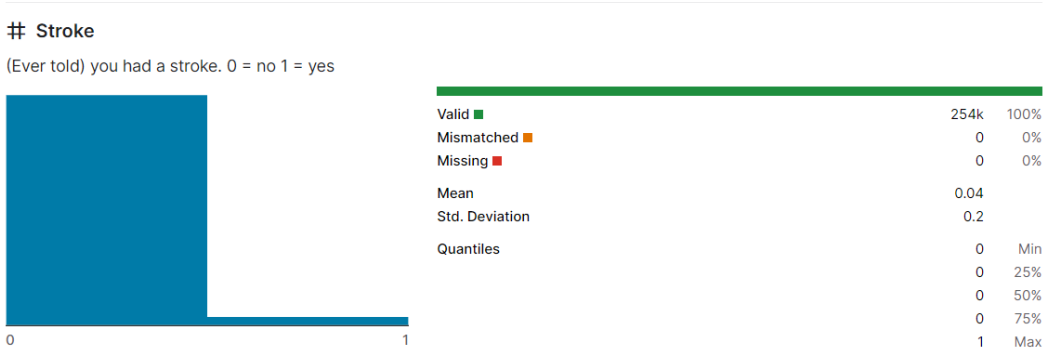

Рисунок 3.8 – Stroke[22]

- HeartDiseaseorAttack (див. рис.3.9);

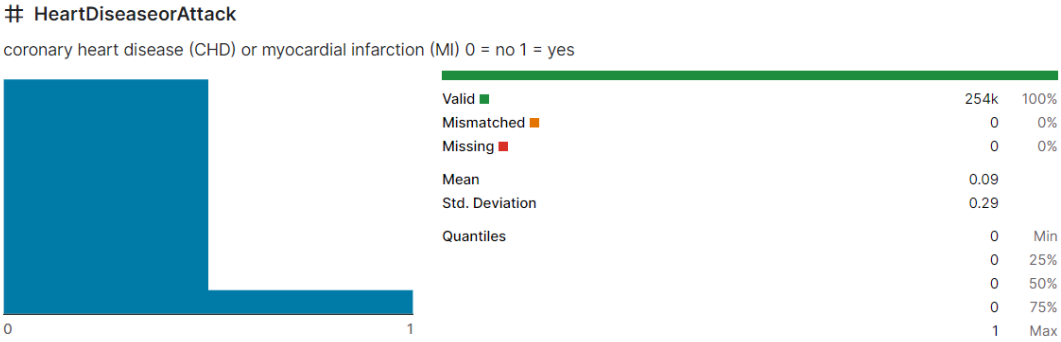

Рисунок 3.9 – HeartDiseaseorAttack[22]

- PhysActivity (див. рис. 3.10);

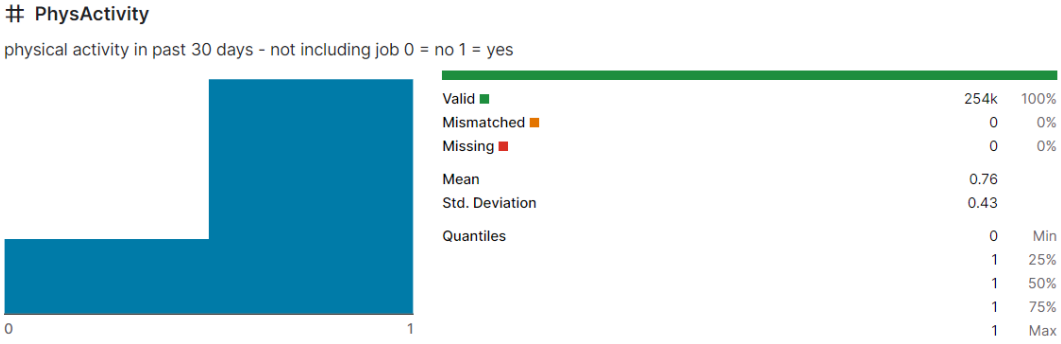

Рисунок 3.10 – PhysActivity[22]

- Fruits (див. рис. 3.11);

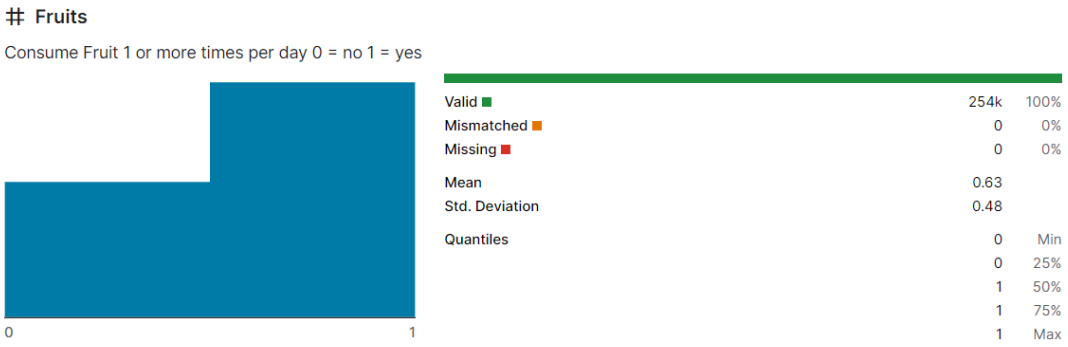

Рисунок 3.11 - Fruits [22]

- Veggies (див. рис. 3.12);

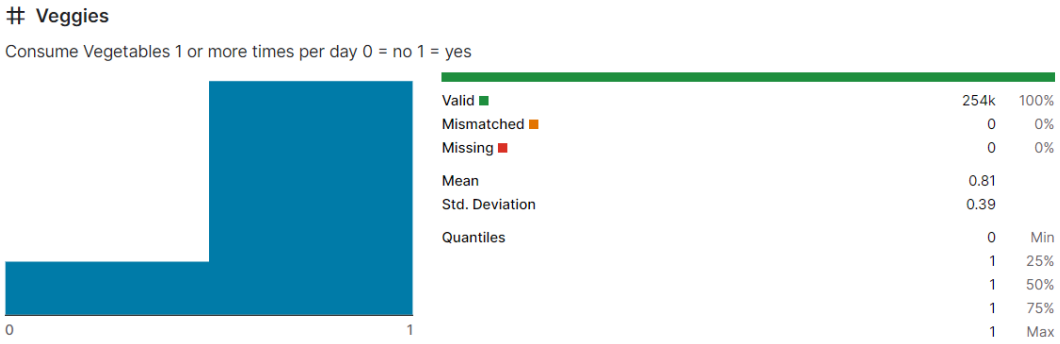

Рисунок 3.12 – Veggies[22]

- HvyAlcoholConsump (див. рис. 3.13);

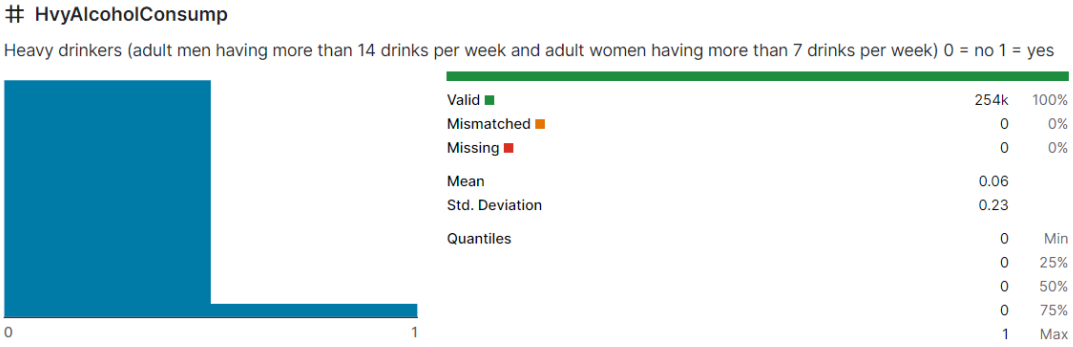

Рисунок 3.13 – HvyAlcoholConsump[22]

- AnyHealthcare (див. рис. 3.14);

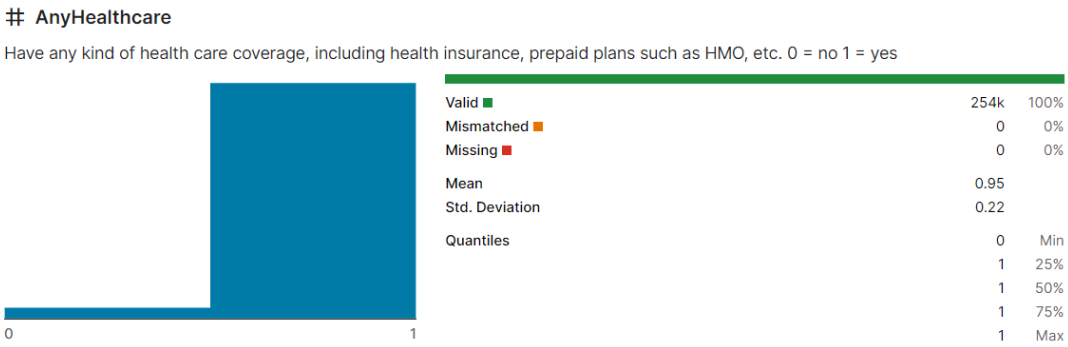

Рисунок 3.14 – AnyHealthcare[22]

- NoDocbcCost (див. рис. 3.15);

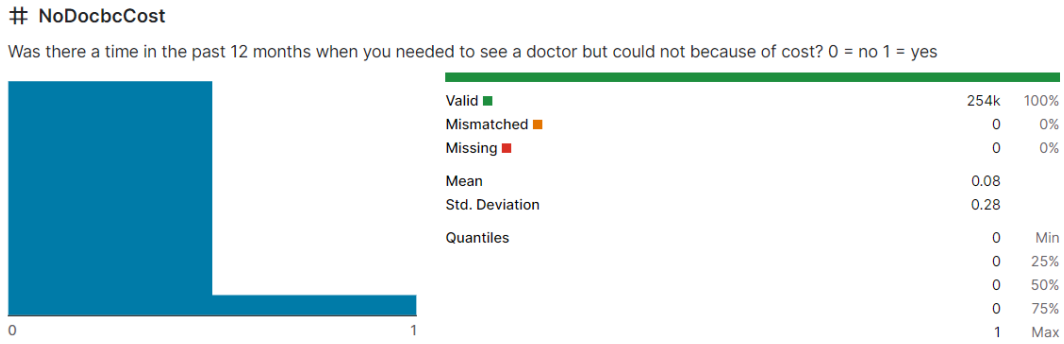

Рисунок 3.15 – NoDocbcCost[22]

- GenHlth (див. рис. 3.16);

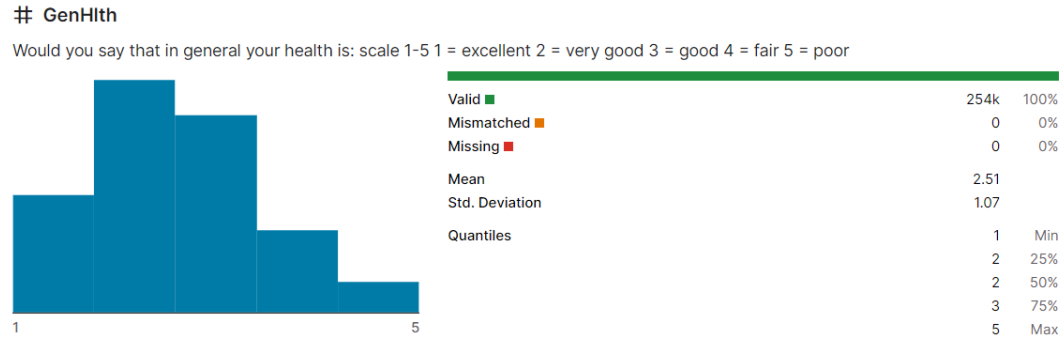

Рисунок 3.16 – GenHlth[22]

- MentHlth (див. рис. 3.17);

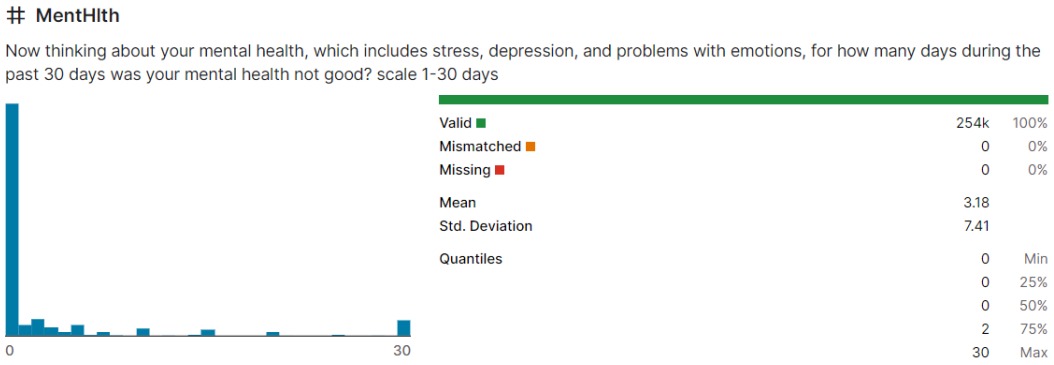

Рисунок 3.17 – MentHlth[22]

- PhysHlth (див. рис. 3.18);

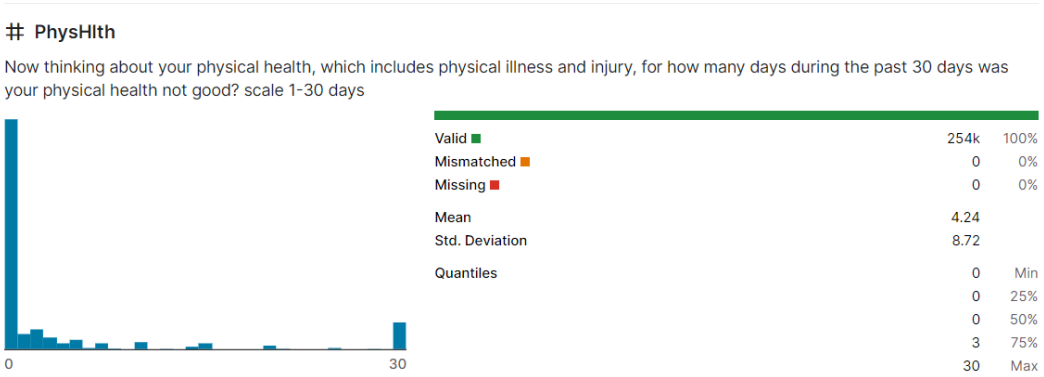

Рисунок 3.18 – PhysHlth[22]

- DiffWalk (див. рис. 3.19);

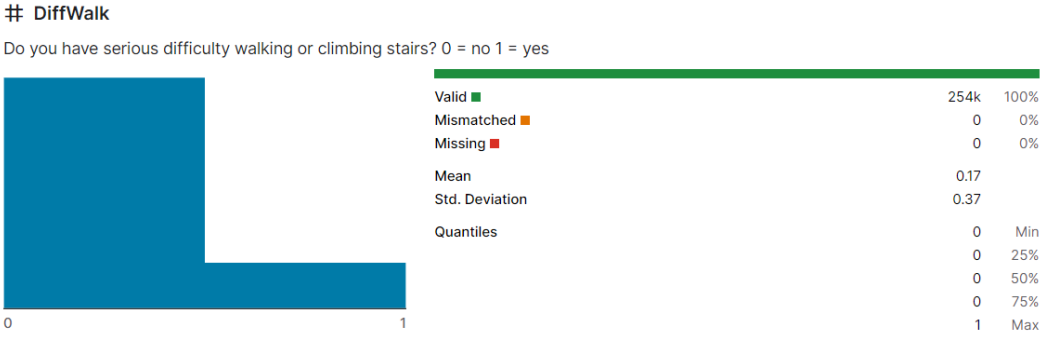

Рисунок 3.19 – DiffWalk[22]

- Sex (див. рис. 3.20);

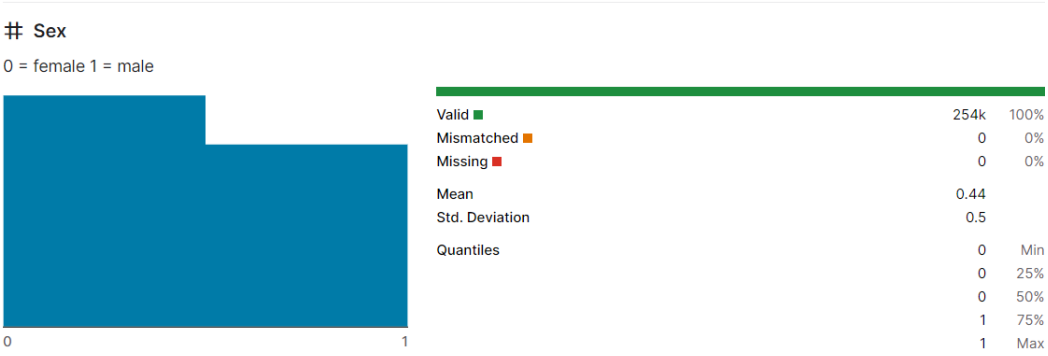

Рисунок 3.20 – Sex[22]

- Age (див. рис. 3.21);

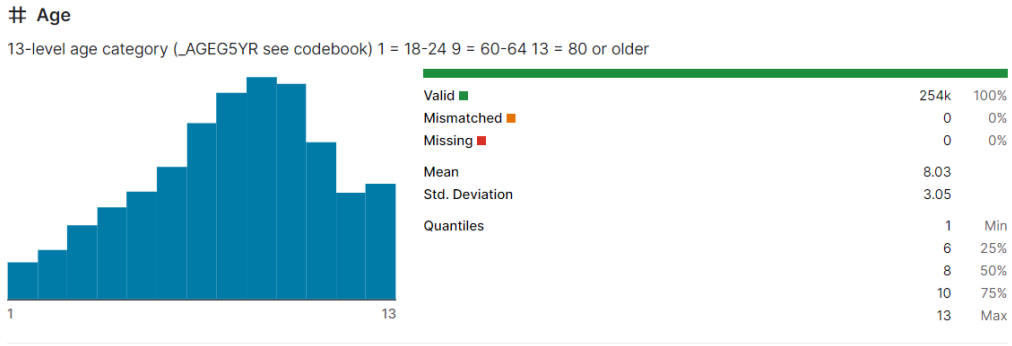

Рисунок 3.21 – Age[22]

- Education (див. рис. 3.22);

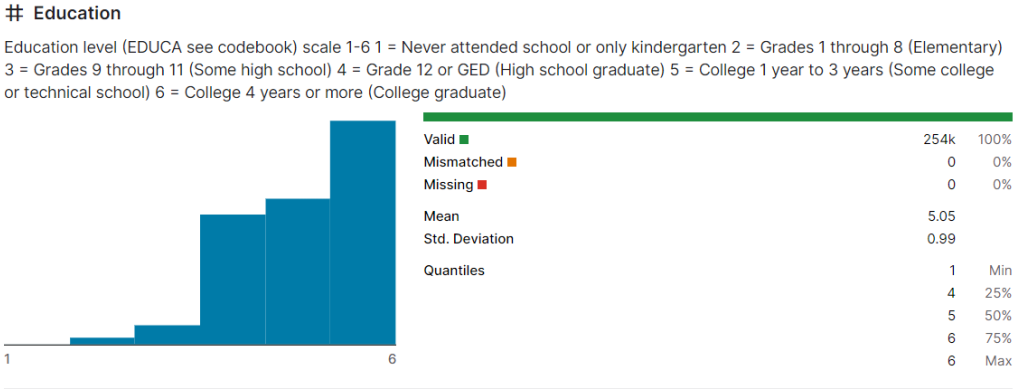

Рисунок 3.22 – Education[22]

- Income (див. рис. 3.23).

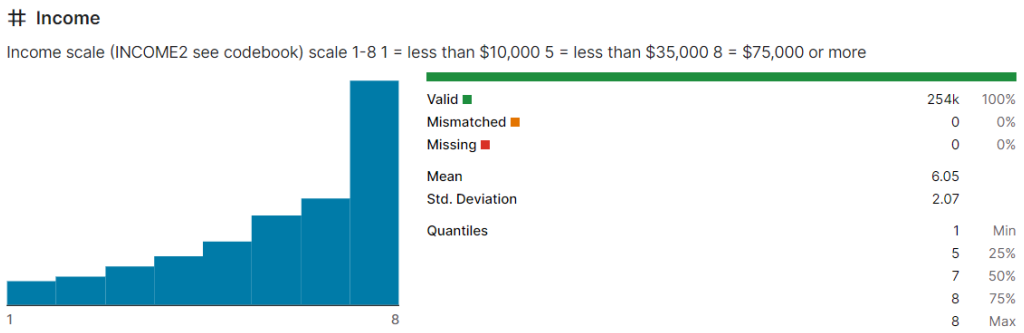

Рисунок 3.23 – Income[22]

Датасет другого набору містить атрибути першого, кількість записів 253680. Дані зображено на рисунку 3.24 нижче.

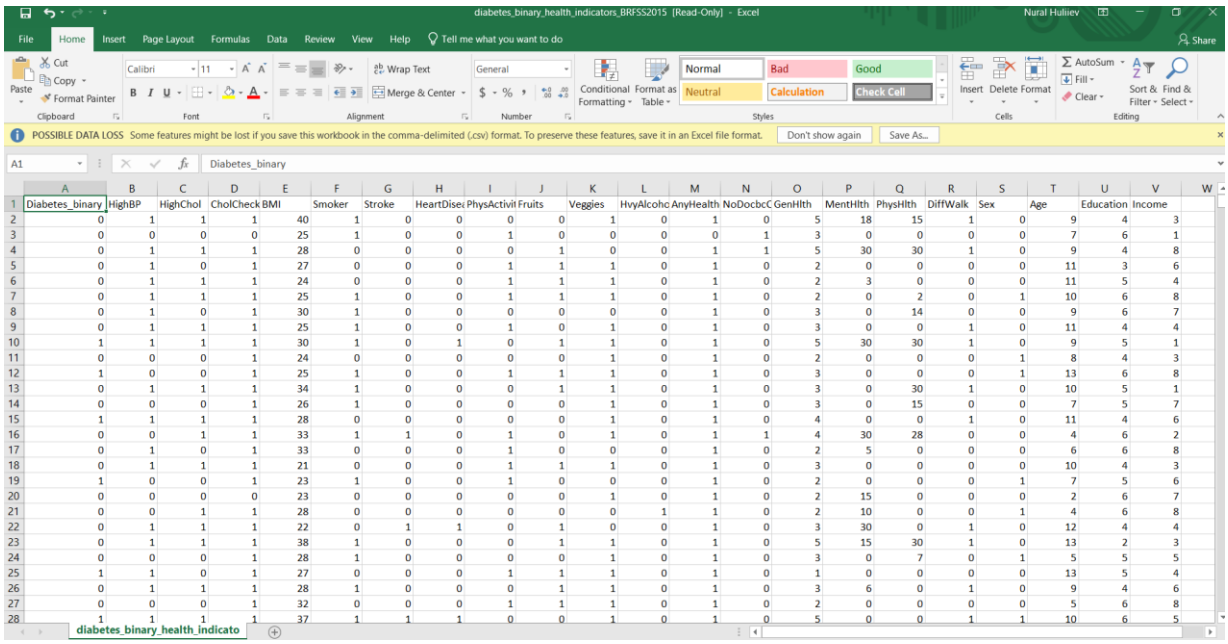

Рисунок 3.24 - Другий датасет(виконано самостійно)

Перейдемо до нормалізації наборів даних.

3.2 Нормалізація даних

У кожному датасеті дані мають різний проміжок значень: від 0 до 1 та від 0 до 100, тому необхідно нормалізувати дані, що можна зробити наступним чином [23]:

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}}. \quad (22)$$

Приклад нормалізації даних першого датасету приведено нижче.

```
def normalisePatientRow(csvData):
   obj = {}
   obj["Diabetes_012"] = csvData["Diabetes_012"]
   obj["HighBP"] = csvData["HighBP"]
   obj["HighChol"] = csvData["HighChol"]
   obj["CholCheck"] = csvData["CholCheck"]
   obj["BMI"] = normaliseParameter(csvData["BMI"], 12, 98)
   obj["Smoker"] = csvData["Smoker"]
   obj["Stroke"] = csvData["Stroke"]
   obj["HeartDiseaseorAttack"] = csvData["HeartDiseaseorAttack"]
   obj["PhysActivity"] = csvData["PhysActivity"]
   obj["Fruits"] = csvData["Fruits"]
   obj["Veggies"] = csvData["Veggies"]
   obj["HvyAlcoholConsump"] = csvData["HvyAlcoholConsump"]
   obj["AnyHealthcare"] = csvData["AnyHealthcare"]
   obj["NoDocbcCost"] = csvData["NoDocbcCost"]
   obj["GenHlth"] = normaliseParameter(csvData["GenHlth"], 1, 5)
   obj["MentHlth"] = normaliseParameter(csvData["MentHlth"], 0,
30)
   obj["PhysHlth"] = normaliseParameter(csvData["PhysHlth"], 0,
30)
   obj["DiffWalk"] = csvData["DiffWalk"]
   obj["Sex"] = csvData["Sex"]
   obj["Age"] = normaliseParameter(csvData["Age"], 1, 13)
   obj["Education"] = normaliseParameter(csvData["Education"],
1, 6)
   obj["Income"] = normaliseParameter(csvData["Income"], 1, 8)

   return obj

def normaliseParameter(value, xMin, xMax):
   return (int(float(value)) - xMin) / (xMax - xMin)
```

Перший метод створює об’єкт запису датасету та при необхідності нормалізує значення, використовуючи мінімальне, максимальне та поточне значення.

3.3 Побудова моделей

Наступним етапом після підготовки даних є тренування моделі, для цього поділемо дані для кожного з методів прогнозування у такому співвідношенні, що 80% даних – тренувальна вибірка, а 20% - тестова.

3.3.1 Багатошаровий перцептрон

Перцептрон матиме 3 шари: перший складатиметься з такої ж кількості нейронів, як і кількість незалежних змінних-атрибутів запису, прихований матиме 2 нейрони для подальшої обробки, а останній – результуючий – один, буде видавати остаточне значення ймовірності приналежності об’єкта до одного з можливих класів: якщо воно більше за визначений поріг, то проноз видаватиме 1, а якщо менше – 0.

Нехай, маємо такі початкові умови для першого набору даних:

- поріг значень – 0.5;
- кількість епох – 100;
- тренувальна вибірка – 200000;
- тестувальна вибірка – 50000.

Алгоритм проходитиме наступним чином:

1) на вхід модель отримує черговий об’єкт;
2) перший шар заповнюється значеннями його атрибутів;
3) обчислюємо зважені суми для кожного з двох шарів прихованого шару;
4) розраховуємо значення сигмоїдної активації для кожного з наступних нейронів;

5) повторюємо ті ж операції для нейронів прихованого шару, які входять далі до результуючого нейрона;
6) коригуємо вагові коефіцієнти.

Повторюємо алгоритм та розраховуємо глобальну помилку за допомогою методу градієнтного спуску, допоки вона не стане мінімальною або поки не дійдемо до максимальної кількості епох навчання[24].

```
def predictForAllData():

   n = random.randint(1000, 10000)
   alpha = random.random() / n
   learningRate = -random.random() / n
   epoches = 100

   w = generateWeights()
   errors = []


   for i in range(epoches):
       results = []
       for row in data:
           layers = fillLayers(w, row, alpha)
           results.append(layers[2]["Activation"])
           w = correctWeights(w, row, layers, learningRate)
       print("ERROR: " + str(calculateGlobalError(results)))
       errors.append(calculateGlobalError(results))
       i+=1
   accuracy(testData, w, alpha)
```

3.3.2 Дерево класифікації

Будуватимемо дерево класифікації. Розглянемо алгоритм для першого набору даних. На початку маємо наступні умови:

- тренувальна вибірка – 200000;
- тестувальна вибірка – 50000.

Основою алгоритму є Gini Impurity дерева – показник, який розраховується як мінімальне значення з усіх можливих Gini Impurity атрибутів об’єкта. Він відображає, на скільки даний атрибут може розмежувати дані за значенням головної залежної ознаки, а чим його значення менше, тим менше помилок він може спричинити. У випадку, коли значення атрибуту не бінарні, до уваги беруться

усі його можливі показники, сортуються, та для кожних двох значень обчислюється їхнє середнє та розраховується для них Gini Impurity.

Після вибору відповідного атрибуту будується корінь з даного ключа, додається ліва гілка, яка містить дані з позитивним значенням по даному атрибуту, а також права – з негативним. Далі обирається за корінь гілка, кількість даних якої більша [25-27].

```
    public static Node CalculateTotalGiniImpurityForAllData(List<Dictionary<string, int>> people)
        {
            var rootTotalGiniImpurity = CalculateTotalGiniImpurityAmongKeys(people, Keys);
            var rootColumnKey = rootTotalGiniImpurity?.Key;

            var treeNode = new Node();
            treeNode.Key = rootColumnKey;
            treeNode.ComparedValue = rootTotalGiniImpurity.Value;
            treeNode.Left = new Node()
            {
                TruePeople = people.Where(p => p[rootColumnKey] == 1 && p["Diabetes_012"] == 1).ToList(),
                FalsePeople = people.Where(p => p[rootColumnKey] == 1 && p["Diabetes_012"] == 0).ToList(),
            };
            treeNode.Right = new Node()
            {
                TruePeople = people.Where(p => p[rootColumnKey] == 0 && p["Diabetes_012"] == 1).ToList(),
                FalsePeople = people.Where(p => p[rootColumnKey] == 0 && p["Diabetes_012"] == 0).ToList(),
            };

            if (treeNode.Left.IsLeaf())
            {
                treeNode.Left.Result = 1;
            }

            if (treeNode.Right.IsLeaf())
            {
                treeNode.Right.Result = 0;
            }

            Keys.Remove(rootColumnKey);

            if (treeNode.Left.People.Count > treeNode.Right.People.Count)
            {
                AddChildrenIfIsImpure(treeNode.Left);
                AddChildrenIfIsImpure(treeNode.Right);
            }
```

```
            else          if          (treeNode.Left.People.Count          <
treeNode.Right.People.Count)
            {
                AddChildrenIfIsImpure(treeNode.Right);
                AddChildrenIfIsImpure(treeNode.Left);
            }

            ReadNode(treeNode);

            return treeNode;
        }
```

Дія алгоритму закінчується, коли дерево містить усі можливі атрибути або гілка стає листям, тобто містить чітко дані одного з можливих класів.

3.3.3 Метод k-найближчих сусідів

Початкові умови:

- навчальна вибірка – 25000;
- тестова вибірка – 30000;
- кількість найближчих сусідів – 3.

Даний класифікатор працює таким чином: чим сусідів певного класу більше, тим ймовірніше, що об’єкт належить до даного класу також. Число k тут виступає за кількість сусідів, які ближче за інших знаходяться до чергового елемента. Відстань у цьому алгоритмі розраховується за евклідовою метрикою. Метод не потребує навчання, тому одразу проводитиметься для елементів тестової вибірки серед записів тренувального набору[28]. Тому він навантажує систему через високомасштабні розрахунки відстаней від усіх елементів тестової вибірки до усіх сусідів з навчальної вибірки.

3.3.4 Логістична регресія

Проведемо експеримент для першого датасету за допомогою логістичної регресії, головною метою якої є розподіл об’єктів у два класи[29-30].

На початку маємо:

– кількість епох – 100;

– тренувальна вибірка – 200000;

– тестувальна вибірка – 50000.

Алгоритм такий:

– ініціалізація синаптичних зв’язків та зміщення;

```
int inputVectorLength = inputVectors[0].Length - 1;
weights = new double[inputVectorLength + 1];
for (int i = 0; i < inputVectorLength; i++)
{
     weights[0] = 0;
}
```

– розрахунок лінійної комбінації вхідних ознак;

```
var value = weights[0];
int inputVectorLength = inputVector.Length - 1;
for (int i = 0; i < inputVectorLength; i++)
{
       value += weights[i + 1] * inputVector[i];
 }
```

– обчислення ймовірного значення за допомогою сигмоїдної функції;

```
return 1 / (1 + Math.Pow(Math.E, -(value)));
```

– визначення функції втрат (log loss);

```
public double CalculateLogLoss(double value, double predictedValue,
int trainDataCount)
        {var logLoss = value * Math.Log(predictedValue) + (1 -
value) * (Math.Log(1 - predictedValue));

            return logLoss * ((-1.0) / trainDataCount);}
```

– коригування вагових зв’язків та зміщення за допомогою градієнтного спуску:

```
for (int k = 0; k < epoches; k++)
```

```
{
  double logLoss = 0;
  var predictedValues = new double[inputVectors.Length];
  for (int i = 0; i < inputVectors.Length; i++)
  {
        var                    predictedValue                    =
CalculateLogisticRegression(inputVectors[i], weights);
        predictedValues[i] = predictedValue;
        weights[0]  +=  l  *  (inputVectors[i][inputVectorLength]  -
predictedValue);
        for (int j = 0; j < inputVectorLength; j++)
        {
        weights[j + 1] += l * (inputVectors[i][inputVectorLength] -
predictedValue)  *  predictedValue  *  (1  -  predictedValue)  *
inputVectors[i][j];
        }}
for (int i = 0; i < inputVectors.Length; i++)
  {
   logLoss += CalculateLogLoss(inputVectors[i][inputVectorLength],
predictedValues[i], trainDataCount);

  Console.WriteLine(logLoss);
  points.Add(new Point(k, logLoss));}
```

Дія алгоритму закінчується, якщо функція втрат набула потрібного значення або дійшов до максимальної кількості епох навчання.

4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Проведено експеримент для першого датасету з початковими умовами, даними вище, та розраховано якість алгоритму на основі обчислення значень метрик [31-33].

Проаналізуємо перший набір даних. При побудові нейромережі функція помилки з кожною епохою повертає менше значення, що можна також прослідкувати на графіку, наведеному на рисунку 4.1.

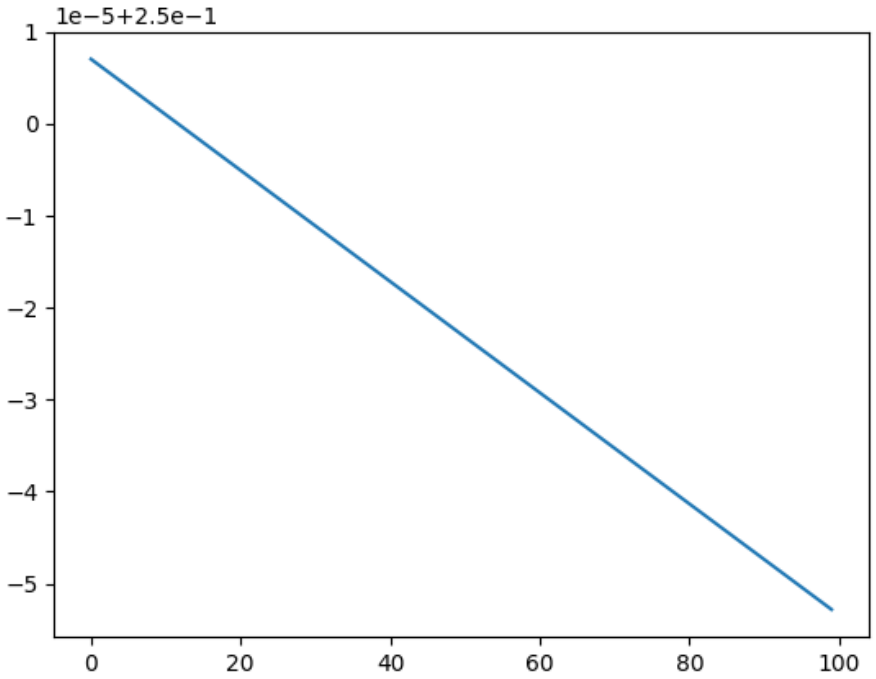

Рисунок 4.1 – Функція помилки перцептрона(виконано самостійно)

Те ж саме стосується помилки, розрахованої з кожною ітерацією при регресійному аналізі, як на рисунку 4.2.

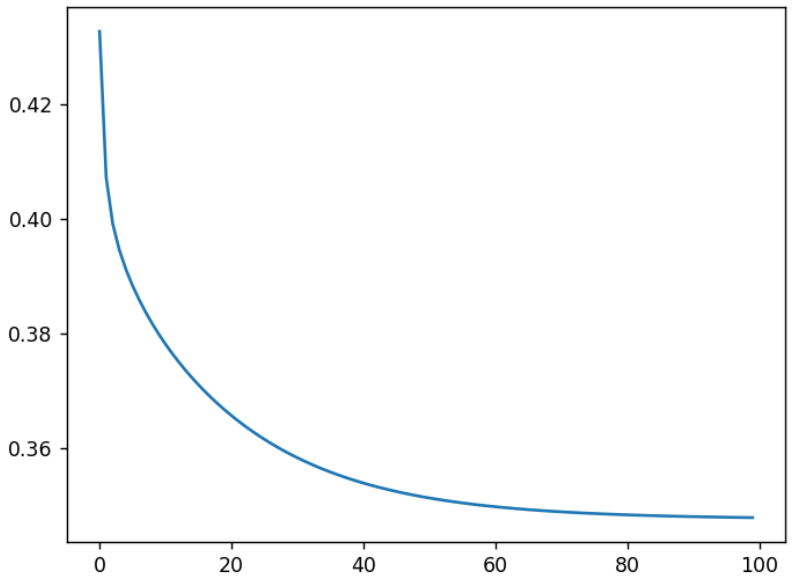

Рисунок 4.2 – Функція помилки логістичної регресії (виконано самостійно)

Тепер промоделюємо та вирішимо векторну оптимізацію:

Потрібно обрати якомога кращу модель з великим відсотком достовірності результатів.

Проведемо інформаційну підготовку прийняття рішення з вибору необхідного алгоритму:

1) опишемо множину альтернатив:
– багатошаровий перцептрон (БШП);
– логістична регресія;
– дерево рішень;
– метод k-найближчих сусідів.
2) опишемо критерії вибору:
– precision– кількість об’єктів, які за алгоритмом належать до позитивного класу, що є правдивим твердженням;
– accuracy – кількість правильно прогнозованих результатів;
– recall – вказує на кількість позитивно визначених елементів з усього об’єму цього класу;
– f-міра – критерій якості, який об’єднує показники precision та recall.
3) опишемо шкали оцінок за критеріями:

– усі показники мають значення у проміжку між 0 та 1.

Наведемо модель поставленої задачі у вигляді таблиці з відомими даними в (див. табл. 4.1).

Таблиця 4.1 – Вихідні дані

	Багатошаровий перцептрон	Дерево класифікації	k-найближчих сусідів	Логістична регресія
Accuracy	84.854	82.8	99.78	85.5575
Precision	1	0.845	0.9974	0.8683
Recall	0.84854	1	1	0.978
F-міра	0.91	0.9162	0.9987	0.92

Для точної оцінки показники мають бути в одному проміжку, тому проведемо нормалізацію даних: поділимо значення точності на еталонне – 100 і матимемо наступне (див. табл. 4.2):

Таблиця 4.2 – Таблиця з нормалізованими даними

	Багатошаровий перцептрон	Дерево класифікації	k-найближчих сусідів	Логістична регресія
Accuracy	0.84854	0.828	0.9978	0.855575
Precision	1	0.845	0.9974	0.8683
Recall	0.84854	1	1	0.978
F-міра	0.91	0.9162	0.9987	0.92

На даному етапі мережі не порівняльні за принципом Парето, тому проведемо лінійну адитивну згортку з нормуючими множниками для усієї моделі задачі та розглянемо результати експерименту (див. табл. 4.3) [34-35].

Таблиця 4.3 – Проведення лінійної адитивної згортки

	Нормуючий множник	Багатошаровий перцептрон	Дерево класифікації	k-найближчих сусідів	Логістична регресія
Accuracy	0.2832929	0.84854	0.828	0.9978	0.855575
Precision	0.2694909	1	0.845	0.9974	0.8683
Recall	0.2613327	0.84854	1	1	0.978
F-міра	0.2670298	0.91	0.9162	0.9987	0.92
		0.97462471	0.96827181	1.079475332	0.977628

Як бачимо, за лінійною адитивною згорткою алгоритм k-найближчих сусідів краще прогнозує приналежність об’єктів до певного класу, але, маючи на увазі, що дослідження проводилося не для усієї вибірки, а для частини даних через довготривалу роботу методу, тдана альтернатива підходить у випадку невеликої кількості даних, тому кращим варіантом є саме логістична регресія для обробки великої кількості інформації. Якщо повторити розрахунки для другого датасету (див. табл. 4.4), результат буде тим самим.

Таблиця 4.4 – Показники дослідження другого датасету

	Багатошаровий перцептрон	Дерево класифікації	k-найближчих сусідів	Логістична регресія
Accuracy	86.482	82.8	99.68	86.925
Precision	0.86482	0.8419	0.9967	0.8796
Recall	1	1	0.9995	0.9833
F-міра	0.9275	0.9141	0.9833	0.9286

Отже, дослідження показує, що оптимальнішим методом прогнозування розвитку діабету серед інсультників та звичаних людей є саме регресійний аналіз.

5 ОПТИМІЗАЦІЯ АЛГОРИТМІВ

Реалізовані алгоритми виконують поставлену мету – прогнозують можливе виникнення та розвиток діабету, як серед звичайних людей, так і серед інсультників. Але дані моделі можна ще оптимізувати за допомогою наступних способів [36].

Багатошаровий перцептрон та логістична регресія застосовують методи оптимізації першого порядку, а саме –метод градієнтного спуску[37-39].

Як відомо, його ідея в тому, синаптичні ваги змінюються ітеративно в прямому або протилежному напрямку цільової функції градієнтів. Оновлення значень, які відображають, як атрибути елементів впливають на залежну ознаку, продовжується до моменту досягнення найоптимальніших результатів. Швидкість роботи даного процесу, а саме – кількість ітерацій, залежить від значення параметру навчання.

Даний метод просто реалізувати при будуванні моделей у машинному навчанні, однак він має декілька недоліків.

Перший з них – у випадку великої кількості даних алгоритм ускладнюється довгими надлишковими обчисленнями. В цому разі в нагоді може стати стохатистичний градієнтний спуск (SGD) [40]. Цей спосіб використовує випадкову обрану вибірку для коригування градієнта при кожній ітерації, не обчислюючи його точне значення, тобто яке даний метод оцінює. Кількість записів даних не впливає на роботу алгоритму, та він здатний досягти сублінійної швидкості збіжності. Тому стохатистичний градієнтний спуск витрачає менше часу, оновлюючи параметри моделей, не займаючи його великомасштабними розрахунками [41].

Друга проблема полягає у визначенні оптимальнішого параметру навчання моделі, точніше кажучи – гіперпараметру [42]. Гіперпараметр – параметр, від якого залежить процес будування моделі, швидкість та точність результатів. При виборі такого найкращого значення, яке можна застосувати в машинному навчанні,

використовуються різні методи, які повертають кортеж гіперпараметрів та втрат, а саме:

- пошук по гратці;
- випадковий пошук;
- байєсова оптимізація;
- оптимізація на основі градієнтів;
- еволюційна оптимізація;
- на основі заселення.

У дослідженні застосовано налаштування гіперпараметрів методом випадкового пошуку, тому надалі можна виключити його з варіантів модифікації та оптимізації реалізованих алгоритмів [43-45].

Дерево класифікації також підлягає коригуванню своєї побудови за допомогою використання іншого методу вибору чергового атрибуту кореню та підкоренів, від яких розгалужуватимуться ліва та права гілки. У роботі було обрано алгоритм ID3, за яким атрибути визначаються на основі значення Gini Impurity. Також існує метод C4.5 – поліпшена та розширена методика, розроблена Росом Куінланом.

C4.5 працює так само, як і ID3, але застосовує концепцію інформаційної ентропії. На кожному вузлі майбутнього дерева обирається атрибут, який краще за інших розбиває вибірку на підмножини за критерієм нормалізованого інформаційного приросту – тобто різниці ентропії. Ознака з найбільшим значенням показника стає атрибутом вузла. Після алгоритм продовжується на поділених наборах [46].

Даний підхід кращий за ID3 такими перевагами:

- здатний обробляти атрибути з різними витратами;
- здатний обробляти тренувальний набір, в якому деякі атрибути не мають взагалі значення;
- здатний видаляти зайві гілки, які не приймають участь у прийнятті рішень.

Метод k-найближчих сусідів може бути оптимізовано вибором іншої метрики розрахунку відстані між сусідами.

Описані вище методи змінять побудовані моделі задля прогнозування більш точного результату, але також може виникнути випадок перенавчання, що уособлює собою «запам’ятовування» тренувальних даних замість «навчання» визначення узагальнених значень. Перенавчена модель погано видає остаточні результати через істотну реакцію на другорядні відхилення тренувальної вибірки. Можливими шляхами усунення виникнення даного недоліку є регуляризація, перехресне затвердження, байєсові параметри [47-48].

ВИСНОВКИ

Моделі машинного навчання – ключ до нових відкриттів в наш час, переважно це стосується медицини, де вони застосовуються для діагностування та прогнозування виникнення можливих захворювань.

До сьогодення проведено чимало спостережень, посвячених вічній проблемі людства – винайти тригери розвитку хвороб, для чого використовуються різні моделі.

У ході кваліфікаційної роботи встановлено, що основою застосування нейронних мереж у медицині для прогнозування та діагностування є побудова багатошарового перцептрону. Наприклад, варіативна точність функціонування нейронної мережі задля діагностування серцево-судинних захворювань коливалась від 64 до 94%. Це були моделі багатошарового перцептрону з двома прихованими шарами із точністю понад 90%, навчання яких базувались на генетичних алгоритмах[49-50].

Головними перевагами обраної моделі в медицині є:

- можливість пошуку взаємозв’язків при дуже складних ситуаціях, коли це здається неможливим або це важко помітити при оцінці ситуаційного становища;
- також завдяки своїй здатності до навчання їй властиво знаходити розв’язки проблем навіть при відсутності апріорного знання про вхідні дані, розвитку дослідженого явища, залежності між параметрами, вхідними даними та очікуваними результатами;
- точність прогнозів не залежить від наявності різнотипних, менш інформативних або пропущених даних.

Але, не дивлячись на корисність моделей машинного навчання, вони мають ряд недоліків:

- навчання потребує деякого часу, нейронна мережа має пройти етапи доучування під час багатократного застосування, а більші показники вхідного значення потребують більше часу;

– іноді реалізація вимагає використання відповідного програмного забезпечення.

Мета завдання досягнута за рахунок побудови моделі, здатної спрогнозувати розвиток цукрового діабету задля запобігання виникнення інсульту та кваліфікаційна робота пройшла апробацію на науковій конференції 7th International Conference on Computational Linguistics and Intelligent Systems (Scopus) April 20–21, 2023 at National Technical University “Kharkiv Polytechnic Institute” (Kharkiv, Ukraine).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ – ПЕРСПЕКТИВНА СФЕРА НАУКИ І СУСПІЛЬСТВА | Наукові конференції. Грудень / Декабрь 2013 | Наукові конференції. URL: http://oldconf.neasmo.org.ua/node/139.

2. КОНФЕРЕНЦІЇ ВНТУ електронні наукові видання. URL: https://conferences.vntu.edu.ua/index.php/mn/mn2020/paper/viewFile/8673/7359\ .

3. Конференції Державного університету «Житомирська політехніка». URL: https://conf.ztu.edu.ua/wp-content/uploads/2019/12/5.pdf.

4. Прогнозування атеросклерозу за допомогою штучної нейронної мережі. URL: https://cyberleninka.ru/article/n/prognozuvannya-aterosklerozu-za-dopomogoyu-shtuchnoyi-neyronnoyi-merezhi/viewer.

5. Учасники проектів Вікімедіа. Нейронні мережі в медицині – Вікіпедія. Вікіпедія. URL: https://uk.m.wikipedia.org/wiki/Нейронні_мережі_в_медицині.

6. CORE – Aggregating the world’s open access research papers. URL: https://core.ac.uk/download/pdf/322991189.pdf (дата звернення: 14.12.2022).

7. Конференції Державного університету «Житомирська політехніка». URL: https://conf.ztu.edu.ua/wp-content/uploads/2019/12/5.pdf.

8. Використання нейронних мереж – перспективна сфера науки і суспільства | Наукові конференції. Грудень / Декабрь 2013 | Наукові конференції. URL: http://oldconf.neasmo.org.ua/node/139.

9. S. Kaushik, A. Choudhury, S. Natarajan, L. A. Pickett and V. Dutt, "Medicine Expenditure Prediction via a Variance- Based Generative Adversarial Network," in IEEE Access, vol. 8, pp. 110947-110958, 2020, doi: 10.1109/ACCESS.2020.3002346.

10. J. Jaruenpunyasak and R. Duangsoithong, "Empirical Analysis of Feature Reduction in Deep Learning and Conventional Methods for Foot Image Classification," in IEEE Access, vol. 9, pp. 53133-53145, 2021, doi: 10.1109/ACCESS.2021.3069625.

11. Selection of Artificial Neural Networks for Disease Prediction / I. Iryna Kyrychenko et al. Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems. 2023. Volume I, Machine Learning Workshop.

12. Ageing, Apr. 2020, [online] Available: https://www.un.org/en/global-issues/ageing/.

13. Osteoarthritis, Apr. 2020, [online] Available: https://www.who.int/medicines/areas/priority_medicines/Ch6_12Osteo.pdf.

14. [N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 886-893, Jun. 2005.

15. C. Kalyoncu and, "GTCLC: Leaf classification method using multiple descriptors", IET Comput. Vis., vol. 10, no. 7, pp. 700-708, Jun. 2016.

16. Y. Qi and G. Zhang, "Strategy of active learning support vector machine for image retrieval", IET Comput. Vis., vol. 10, no. 1, pp. 87-94, Feb. 2016.

17. T. George, S. P. Potty and S. Jose, "Smile detection from still images using KNN algorithm", Proc. Int. Conf. Control Instrum. Commun. Comput. Technol. (ICCICCT), pp. 461-465, Jul. 2014.

18. G. Khan, A. Siddiqi, G. Khan, S. Q. Wahla and S. Samyan, "Geometric positions and optical flow based emotion detection using MLP and reduced dimensions", IET Image Process., vol. 13, no. 4, pp. 634-643, Mar. 2019.

19. Учасники проектів Вікімедіа. Дерево ухвалення рішень – Вікіпедія. Вікіпедія. URL: https://uk.wikipedia.org/wiki/Дерево_ухвалення_рішень (дата звернення: 05.04.2023).

20. S. Verma, M. A. Razzaque, U. Sangtongdee, C. Arpnikanondt, B. Tassaneetrithep and A. Hossain, "Digital Diagnosis of Hand, Foot, and Mouth Disease Using Hybrid Deep Neural Networks," in IEEE Access, vol. 9, pp. 143481-143494, 2021, doi: 10.1109/ACCESS.2021.3120199.

21. N. Alamdari, K. Tavakolian, M. Alhashim and R. Fazel-Rezai, "Detection and classification of acne lesions in acne patients: A mobile application", Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT), pp. 0739-0743, May 2016.

22. Diabetes Health Indicators Dataset. Kaggle: Your Machine Learning and Data Science Community. URL: https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset (date of access: 05.04.2023).

23. S. Abdul-Rahman, A. Khairil Norhan, M. Yusoff, A. Mohamed and S. Mutalib, "Dermatology diagnosis with feature selection methods and artificial neural network", Proc. IEEE-EMBS Conf. Biomed. Eng. Sci., pp. 371-376, Dec. 2012.

24. T. A. Rimi, N. Sultana and M. F. A. Foysal, "Derm-NN: Skin diseases detection using convolutional neural network", Proc. 4th Int. Conf. Intell. Comput. Control Syst. (ICICCS), pp. 1205-1209, May 2020.

25. M. F. Aryan, W. Krathu, C. Arpnikanondt and B. Tassaneetrithep, "Image recognition for detecting hand foot and mouth disease", Proc. 11th Int. Conf. Adv. Inf. Technol., pp. 1-11, Jul. 2020.

26. When Should I Suspect Hand Foot and Mouth Disease, Aug. 2020, [online] Available: https://cks.nice.org.uk/topics/hand-foot-mouth-disease/diagnosis/when-t%o-suspect-hand-foot-mouth-disease/.

27. Y. Liu, A. Jain, C. Eng, D. H. Way, K. Lee, P. Bui, et al., "A deep learning system for differential diagnosis of skin diseases", Nature Med., vol. 26, no. 6, pp. 900-908, 2020.

28. A. Taravat, S. Proud, S. Peronaci, F. D. Frate and N. Oppelt, "Multilayer perceptron neural networks model for meteosat second generation SEVIRI daytime cloud masking", Remote Sens., vol. 7, no. 2, pp. 1529-1539, 2015.

29. G. -M. Lin and H. -C. Zeng, "Electrocardiographic Machine Learning to Predict Mitral Valve Prolapse in Young Adults," in IEEE Access, vol. 9, pp. 103132-103140, 2021, doi: 10.1109/ACCESS.2021.3098039.

30. L. A. Freed, D. Levy, R. A. Levine, M. G. Larson, J. C. Evans, D. L. Fuller, et al., "Prevalence and clinical outcome of mitral-valve prolapse", New England J. Med., vol. 341, no. 1, pp. 1-7, Jul. 1999

31. R. B. Devereux, E. C. Jones, M. J. Roman, B. V. Howard, R. R. Fabsitz, J. E. Liu, et al., "Prevalence and correlates of mitral valve prolapse in a population-based sample of American Indians: The strong heart study", Amer. J. Med., vol. 111, no. 9, pp. 679-685, Dec. 2001.

32. J. M. Flack, J. H. Kvasnicka, J. M. Gardin, S. S. Gidding, T. A. Manolio and D. R. Jacobs, "Anthropometric and physiologic correlates of mitral valve prolapse in a

biethnic cohort of young adults: The CARDIA study", Amer. Heart J., vol. 138, no. 3, pp. 486-492, Sep. 1999.

33. R. B. Devereux, I. Hawkins, R. Kramer-Fox, E. M. Lutas, I. W. Hammond, M. C. Spitzer, et al., "Complications of mitral valve prolapse. Disproportionate occurrence in men and older patients", Am. J. Med., vol. 81, no. 5, pp. 751-758, Nov. 1986.

34. P.-Y. Liu, K.-Z. Tsai, Y.-P. Lin, C.-S. Lin, H.-C. Zeng, E. Takimoto, et al., "Prevalence and characteristics of mitral valve prolapse in military young adults in Taiwan of the CHIEF heart study", Sci. Rep., vol. 11, no. 1, pp. 2719, Feb. 2021.

35. R. A. Levine et al., "Mitral valve disease–morphology and mechanisms", Nature Rev. Cardiol., vol. 12, no. 12, pp. 689-710, Dec. 2015.

36. C. V. Leier, T. D. Call, P. K. Fulkerson and C. F. Wooley, "The spectrum of cardiac defects in the Ehlers-Danlos syndrome types I and III", Ann. Int. Med., vol. 92, pp. 171-178, Feb. 1980.

37. K. Hirata, F. Triposkiadis, E. Sparks, J. Bowen, H. Boudoulas and C. F. Wooley, "The Marfan syndrome: Cardiovascular physical findings and diagnostic correlates", Amer. Heart J., vol. 123, no. 3, pp. 743-752, Mar. 1992.

38. G. H. Tison, J. Zhang, F. N. Delling and R. C. Deo, "Automated and interpretable patient ECG profiles for disease detection tracking and discovery", Circulation Cardiovascular Qual. Outcomes, vol. 12, no. 9, Sep. 2019

39. CORE – Aggregating the world’s open access research papers. URL: https://core.ac.uk/download/pdf/322991189.pdf (date of the application: 14.12.2022).

40. S. Trenn, "Multilayer perceptrons: Approximation order and necessary number of hidden units", IEEE Trans. Neural Netw., vol. 19, no. 5, pp. 836-844, May 2008.

41. E. Romero and J. M. Sopena, "Performing feature selection with multilayer perceptrons", IEEE Trans. Neural Netw., vol. 19, no. 3, pp. 431-441, Mar. 2008.

42. H. Seo and D. -H. Cho, "Cancer-Related Gene Signature Selection Based on Boosted Regression for Multilayer Perceptron," in IEEE Access, vol. 8, pp. 64992-65004, 2020, doi: 10.1109/ACCESS.2020.2985414.

43. Al.Hak, L. A. (2022). Diabetes Prediction Using Binary Grey Wolf Optimization and Decision Tree. International Journal of Computing, 21(4), 489-494. https://doi.org/10.47839/ijc.21.4.2785.

44. Mienye, I. D., Sun, Y., & Wang, Z. (2020). IMPROVED PREDICTIVE SPARSE DECOMPOSITION METHOD WITH DENSENET FOR PREDICTION OF LUNG CANCER. International Journal of Computing, 19(4), 533-541. https://doi.org/10.47839/ijc.19.4.1986

45. Qeethara Kadhim Al-Shayea and Itedal S.H. Bahia, "Urinarysystem Dieseases Diagnosis Using Artificial neural networks",IJCSNS, Vol.10, No.7, July 2010.

46. Sharonova, N., Kyrychenko, I., Gruzdo, I., Tereshchenko, G., "Generalized Semantic Analysis Algorithm of Natural Language Texts for Various Functional Style Types", 2022 6th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2022), 2022. – CEUR-WS 3171, 2022, ISSN 16130073. - Volume I: Main, PP. 16 - 26.

47. Sharonova, N., Kyrychenko, I., Tereshchenko, G., "Application of big data methods in E-learning systems", 2021 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), 2021. – CEUR-WS, 2021, ISSN 16130073. - Volume 2870, PP. 1302-1311.

48. Gruzdo, I., Kyrychenko, I., Tereshchenko, G., Shanidze, N. Metrics applicable for evaluating software at the design stage. 2021 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), 2021. – CEUR-WS, 2021, ISSN 16130073. - Volume 2870, PP. 916-936.

49. Ворочек О.Г, Дударь В.В. Исследование интеллектуальных средств анализа и прогнозирования. Журнал «Вестник» Херсонского национального технического университета №4(27), 2007.

50. Даниленко В. Д., Назаров О. С. Дослідження методів прогнозування для потреб автомобільного підприємства щодо забезпечення матеріальних запасів. "Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління" : Дванадцята міжнар. науково-техн. конф., 27 квіт. 2022 р.