

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра програмної інженерії

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка системи для неврології

Виконав: студент IV курсу, групи СП-42
121 Інженерія програмного
спеціальності забезпечення

(шифр і назва спеціальності)

	Яковів Б.Ю.
(підпис)	(прізвище та ініціали)
Керівник	Пастух О.А.
(підпис)	(прізвище та ініціали)
Нормоконтроль	Стоянов Ю.М.
(підпис)	(прізвище та ініціали)
Завідувач кафедри	Петрик М.Р.
(підпис)	(прізвище та ініціали)
Рецензент	
(підпис)	(прізвище та ініціали)

Тернопіль

2024

АНОТАЦІЯ

За результатами Всесвітньої Організації Охорони Здоров'я в найближчі десять, п'ятнадцять років відбудуться зміни у рейтингу кількості захворювань людей із серцево-судинних захворювань на психоневрологічні захворювання. Таким чином, логічно випливає висновок щодо потреби медичного діагностування психоневрологічних захворювань на ранніх стадіях. До того ж це потрібно робити у телемедичний спосіб. Окрім цього, потрібно забезпечувати високу точність та надійність медичного діагностування. Для цього необхідно розробляти нові інтелектуалізовані технічні засоби, зокрема, інтелектуалізовані програми, інтелектуалізоване програмне забезпечення, тощо. Наприклад, так діють у технологічній компанії IBM, які створили величезну кількість інтелектуалізованих хмарних програмних систем і навіть цілих платформ для медицини у вигляді віртуальних клінік, віртуальних медичних діагностичних центрів, тощо. У даній кваліфікаційній роботі на здобуття освітнього ступеня бакалавр спроектовано та зконструйовано інтелектуалізоване програмне забезпечення для мелемедичної неврології, зокрема, для діагностування неврологічної патології у вигляді захворювання Паркінсона. В основі програмного забезпечення використано регресійне дерево прийняття рішень для вирішення завдання регресійного рівня (ступені) захворюваності на хворобу Паркінсона. У роботі забезпечено можливість зберігання програмних компонентів, які попередньо були навчені на даних і ці програмні компоненти можна використовувати у подальшому сторонніми програмами. Виконано повне метрологічне дослідження створеного програмного забезпечення у вигляді коефіцієнта детермінації, середнього значення відносної похибки, середнього значення квадрата відхилення вимірювальних значень відносно їх середніх. Створено можливість програмної візуалізації метрологічних характеристик, які підвищують наочність сприйняття.

ANNOTATION

According to the results of the World Health Organization, in the next ten to fifteen years there will be changes in the rating of the number of human diseases from cardiovascular diseases to psycho-neurological diseases. Thus, a logical conclusion follows regarding the need for medical diagnosis of psycho-neurological diseases in the early stages. In addition, it should be done telemedically. In addition, it is necessary to ensure high accuracy and reliability of medical diagnosis. For this purpose, it is necessary to develop new intellectualized technical means, in particular, intellectualized programs, intellectualized software, etc. For example, this is how the technology company IVM operates, which has created a huge number of intellectualized cloud-based software systems and even entire platforms for medicine in the form of virtual clinics, virtual medical diagnostic centers, etc. In this qualification work for obtaining a bachelor's degree, intellectualized software for medical neurology, in particular, for diagnosing neurological pathology in the form of Parkinson's disease, was designed and constructed. The basis of the software is a regression decision-making tree used to solve the problem of the regression level (degree) of the incidence of Parkinson's disease. The work provides the possibility of storing software components that were previously trained on data and these software components can be used later by third-party programs. A full metrological study of the created software was performed in the form of the coefficient of determination, the average value of the relative error, the average value of the square of the deviation of the measurement values relative to their averages. The possibility of software visualization of metrological characteristics has been created, which increases the clarity of perception.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	9
ВСТУП.....	10
РОЗДІЛ 1. КОНСТРУКТИВНИЙ АНАЛІЗ СТАНУ ПРОГРАМНИХ СИСТЕМ ДЛЯ ДІАГНОСТУВАННЯ НЕЙРОДЕГЕНЕРАТИВНИХ ЗАХВОРЮВАНЬ.....	11
1.1. Епідемічний стан нейродегенеративних захворювань.....	11
1.2. Технічні засоби для діагностування нейродегенеративних захворювань.....	14
1.3. Нейротехнологічні рішення усунення нейродегенеративних патологій.....	18
1.4. Мета і завдання бакалаврської роботи.....	20
РОЗДІЛ 2. ПРАКТИЧНЕ ПРОЕКТУВАННЯ СИСТЕМИ ДІАГНОСТУВАННЯ НЕВРОЛОГІЧНОГО ЗАХВОРЮВАННЯ ПАРКЕНСОНА.....	21
2.1. Інтерпретація введених даних для системи діагностування рівня захворюваності на хворобу Паркінсона.....	22
2.2. Використання бібліотек і фреймворків із репозиторію анаконди у проекті.....	25
2.3. Програмний інструмент для введення даних у скрипт проекту.....	26
.	
2.4. Проектування програмного конвеєра для апріорної переробки інформації.....	28
2.5. Програмний інструмент для утворення навчальної та тестової множин даних.....	30

2.6. Програмний інструмент для проектування регресійних дерев рішень.....	31
2.7. Інструмент для кросс-валідації регресійних дерев рішень	34
2.8. Інструменти для оцінювання метрик якості регресійних дерев рішень.....	35
2.9. Програмні інструменти для гіпероптимізації регресійних дерев рішень.....	40
2.10. Програмний інструмент для збереження та завантаження регресійних дерев рішень.....	42
РОЗДІЛ 3. ПРИКЛАДНА РЕАЛІЗАЦІЯ СИСТЕМИ ДІАГНОСТУВАННЯ НЕВРОЛОГІЧНОГО ЗАХВОРЮВАННЯ ПАРКІНСОНА.....	44
3.1. Задіяння бібліотек та фреймворків у пайтонівських скриптах проекту.....	44
.	
3.2. Уведення набору даних в систему діагностування захворювання Паркенсона.....	44
3.3. Апріорна переробка набору даних хворих на захворювання Паркенсона.....	45
3.4. Розщеплення набору даних на характеристики хворих та рівень захворюваності на Паркінсона.....	46
3.5. Кросс Валідація регресійних дерев прийняття рішень при діагностуванні захворювання Паркінсона.....	47
3.6. Рандомне формування навчального та тестових наборів даних.....	49
3.7. Навчання, тестування та прийняття рішень на основі регресійних дерев прийняття рішень при діагностування захворювання Паркінсона.....	50
3.8. Формування фрейму метрик якості для регресійних дерев прийняття рішень у діагностуванні хвороби Паркенсона.....	51

3.9. Візуалізація значень метрики якості для регресійних дерев прийняття рішень для визначення рівня хвороби Паркенсона.....	52
3.10. Зберігання та використання регресійних дерев прийняття рішень для визначення рівня хвороби Паркінсона.....	53
3.11. Гіпер Оптимізація регресійних дерев прийняття рішень для визначення рівня хвороби Паркінсона.....	54
3.11.1. Гіпер Оптимізація регресійних дерев прийняття рішень для визначення рівня хвороби Паркінсона на основі GridSearchCV.....	54
3.11.2. Гіпер Оптимізація регресійних дерев прийняття рішень для визначення рівня хвороби Паркінсона на основі Random Search CV.....	55
.	
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....	56
.	
4.1. Моделювання та прогнозування небезпечних ситуацій.....	56
4.2. Механізація транспортування та встановлення важких за вагою деталей.....	58
ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	62
ДОДАТКИ.....	64
.	
Додаток А Технічне завдання	65
Додаток Б. Тези доповіді на конференції.....	70
Додаток В. Диск із кваліфікаційною роботою бакалавра.....	73

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

ПЗ – програмне забезпечення

ПС ДНЗ – програмна система для діагностування неврологічних захворювань

EEG – електроенцефалографічні сигнали

fMRI – функціональна магніто-резонансна томографія

НЗ – неврологічні захворювання

ВСТУП

За результатами Всесвітньої Організації Охорони Здоров'я в найближчі десять, п'ятнадцять років відбудуться зміни у тренді кількості захворювань людей із серцево-судинних захворювань на психоневрологічні захворювання. Тому виникає потреба медичного діагностування психо-неврологічних захворювань на ранніх стадіях зростання даного тренду. Серед психоневрологічних захворювань часто має місце неврологічне захворювання Паркінсона. Для його запобігання необхідна рання медична діагностика з високою точністю та достовірністю. А це можливо за умови використання сучасного програмного забезпечення в основі роботи якого використовуються інтелектуалізовані алгоритми. Саме так діють високотехнологічні компанії, на кшталт, IBM, яка створила великі медичні інтернет платформи онлайн клінік, онлайн медичних діагностичних центрів, тощо. На основі різноманітних даних, що надходять від пацієнтів, вони можуть фактично миттєво з високою точністю та достовірністю встановити діагноз захворювання пацієнта і до того ж з низькою оплатою для нього, оскільки їх інтернет медична діагностика масштабується на весь світовий ринок.

Висока точність та достовірність медичних діагностичних рішень таких інтернет платформ для медичної діагностики базується на знаннях великої кількості експертів високого рівня у конкретній сфері медицини, що реалізуються в інтелектуалізованих алгоритмах програмного забезпечення.

У даній роботі пропонується розглянути розробку програмного забезпечення на основі інтелектуалізованих алгоритмів для систем у сфері неврологічної діагностики степені захворюваності на хворобу Паркінсона.

РОЗДІЛ 1. КОНСТРУКТИВНИЙ АНАЛІЗ СТАНУ ПРОГРАМНИХ СИСТЕМ ДЛЯ ДІАГНОСТУВАННЯ НЕЙРОДЕГЕНЕРАТИВНИХ ЗАХВОРЮВАНЬ

Відповідно до даних Всесвітньої організації охорони здоров'я через десять – п'ятнадцять років серцево-судинні захворювання не будуть займати домінуюче місце серед населення планети Земля. Їх місце займе епідемія різного роду нейродегенеративних патологій. Тому гостро постає питання упередження виникнення епідемічного зростання психоневрологічних захворювань для уникнення проблем людської цивілізації. Причиною такого роду захворювань людства, а відповідно потреба його діагностування обумовлена багатьма чинниками, зокрема, великими потоками інформації, які йдуть з різних джерел. Причому – це є експоненційно великі потоки інформації. Збільшення реактивності реагування на ті чи інші інформаційні зміни. Шириною спектру різноманітності інформації та багато іншими факторами.

Це все разом у сукупності є фундаментальною потребою у діагностиці психоневрологічних захворювань, зокрема, тог, що розглядається у даній роботі – неврологічне захворювання Паркінсона. А для високоточної (прецизійної) роботи діагностичних пристроїв потрібна розробка програмного забезпечення з використанням інтелектуалізованих програмних засобів.

1.1 Епідемічний стан нейродегенеративних захворювань

Як зазначено вище на даний момент часу зростає кількість психоневрологічних захворювань, зокрема патології Паркінсона, у людей не лише літнього віку, як наведено на рисунку нище, але і у людей молодшого віку. Фактично

такого виду неврологічні патології, можна сказати "помолоділи", а для збереження здоров'я населенню необхідно точно, надійно і швидко діагностувати неврологічну патологію Паркінсона.



Рис.1.1 – Нейродегенеративна патологія – деменція

У роботі в основу створеного програмного забезпечення покладено використання різних характеристик людей. Наприклад, їх вік, амплітуда тремтіння кінцівок, частота тремтіння кінцівок, тощо. Виявляється, що такого роду характеристику дають можливість оцінювати ступінь захворюваності пацієнтів на неврологічне захворювання Паркінсона.

Причиною захворюваності на неврологічну патологію Паркінсона є недостатній рівень синтезу нейромедіатора (нейротрансмітера, нейро месенджера) дофаміну. Оскільки нейромедіатор дофамін синтезується за умов задоволення людиною від умов життя, то відповідно його концентрація залежить від хороших умов проживання людини, тому потреба у покращенні життя людини є надзвичайно важливою.

Діагностування неврологічної патології Паркінсона, може виконуватися і на основі інших характеристик. Наприклад, на електрофізіологічних характеристиках мозкової діяльності. Для наочності доречно розглянути рисунок, який наведений нище.



Рис.1.2 – Неврологічне захворювання (захворювання Паркінсона)

До характеристик належать електроенцефалографічні сигнали, магнітоенцефалографічні сигнали, зображення магніто-резонансного томографа та багато інших характеристик. Всі ці підходи можуть використовуватися для одержання додаткових атрибутів, які використовуватимуться програмним забезпеченням при діагностуванні патології Паркінсона. Використання більшої кількості інформації дає можливість збільшувати потенційну точність діагностування, підвищувати його достовірність, забезпечувати його стійкість роботи та багато інших метрологічних характеристик.

У даній роботі вихідними даними розглядається лінійно апроксимована змінна стандарту визначення ступеня захворюваності на Паркінсона. Таким чином, діагностування зводиться до вирішення задачі регресії, оскільки вихідні дані мають

континуальну природу. Відповідно точність роботи програмного забезпечення визначається також метриками якості, що використовуються у регресійному аналізі, на кшталт, середнє значення абсолютної помилки, середнє значення квадратичної помилки, тощо.

1.2 Технічні засоби для діагностування нейродегенеративних захворювань

Як було зазначено раніше до даних які використовуються у роботі – це різносторонні характеристики людини корисно використовувати інші атрибути, наприклад електроенцефалографічні. сигнали. Корисно розглянути рисунок, який наведено нище.

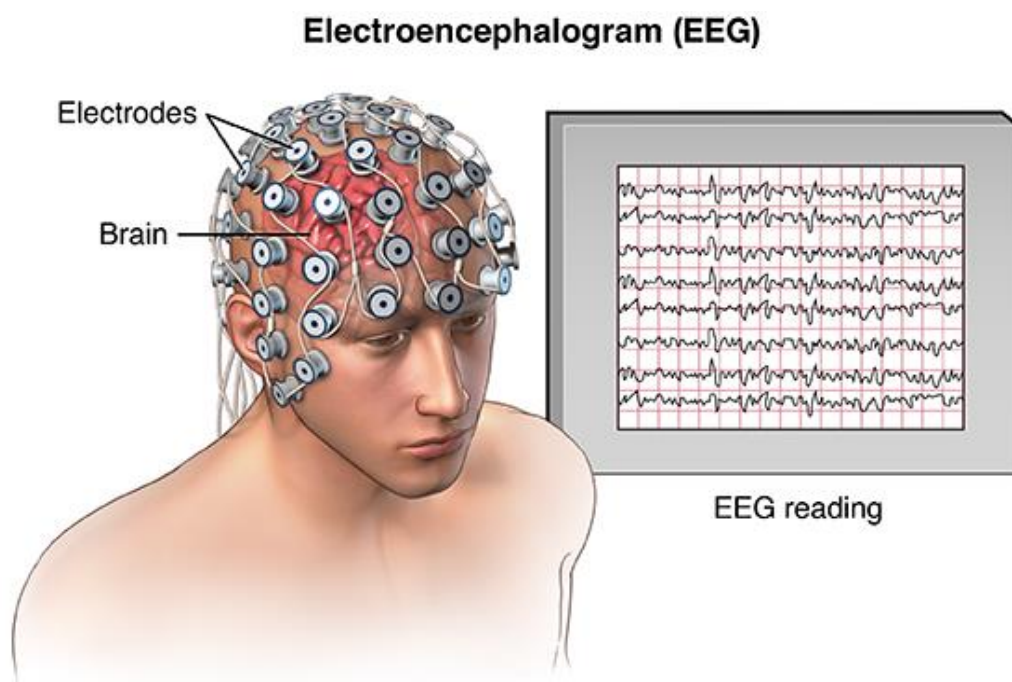


Рис.1.3 – Експериментальні вимірювання (записи) EEG для діагностування неврологічних захворювань

Також дуже часто на практиці використовують зображення магніто резонансного томографа. Ці зображення можуть давати можливість для підвищення точності діагностування неврологічної патології Паркінсона і досліджувати локалізованість тих чи інших місць його у головному мозку. Для наочності розгляньмо зображення, що наведено нище на рисунку.



Рис.1.4 – Експериментальні вимірювання (записи) fMRI для діагностування неврологічних захворювань

Якщо порівнювати технічні засоби для діагностування Паркінсона патології, то кожен з них має свої, як переваги так і недоліки. Наприклад, електроенцефалограф дає змогу реактивно у часі вимірювати значення електричної активності мозку, тобто його перевагою є часова реактивність. Проте, недоліком яким він володіє є погана роздільна здатність у просторі. Абсолютно все навпаки має місце у магніто резонансного томографа. Зображення головного мозку під різними кутами зрізу мають надзвичайно високоточну роздільну здатність по простору і це дає змогу до визначення локальної зони виникнення збурень, що спричиняють ознаки Паркінсона. Тому магніторезонансна томографія має велику перевагу над електроенцефалографом у просторовому розумінні. Проте вона також не є без слабких сторін у порівнянні з електроенцефалографом, магніторезонансна томографія не має достатнього рівня часової реактивності. Таким чином багато

корисної інформації про патологічний стан мозку залишаються швидко змінними у часі, а тому їх не детектується. І це суттєвий недолік магніторезонансних томографів у порівнянні з електроенцефалографами.

Внаслідок вищенаведеного впливає ідея одночасно-паралельного вимірювання стану активності головного мозку, тобто вимірювання його стану на основі одночасного визначення електроенцефалографічної активності та зображень магніто резонансного томографа, як наведено на рисунку нище.



Рис.1.5 – Одночасні експериментальні вимірювання (записи) EEG-fMRI
для діагностування неврологічних захворювань

Одночасне діагностування стану головного мозку на основі електроенцефалографічних сигналів та магніторезонансних томографічних зображень дають багато додаткових атрибутів для діагностування, а тому дають потенційну можливість для підвищення точності, стабільності, інформативності у процесі діагностування неврологічної патології Паркінсона. І цим самим

досягається для діагностики Паркінсона топології висока часова реактивність та висока просторова роздільна здатність.

Окремо важливим питанням є питання візуалізації результатів діагностування. Досить інформативною візуалізацією є форма представлення у вигляді "вибух сонця". Приклад такої візуалізації можна побачити на подалі наведеному рисунку.

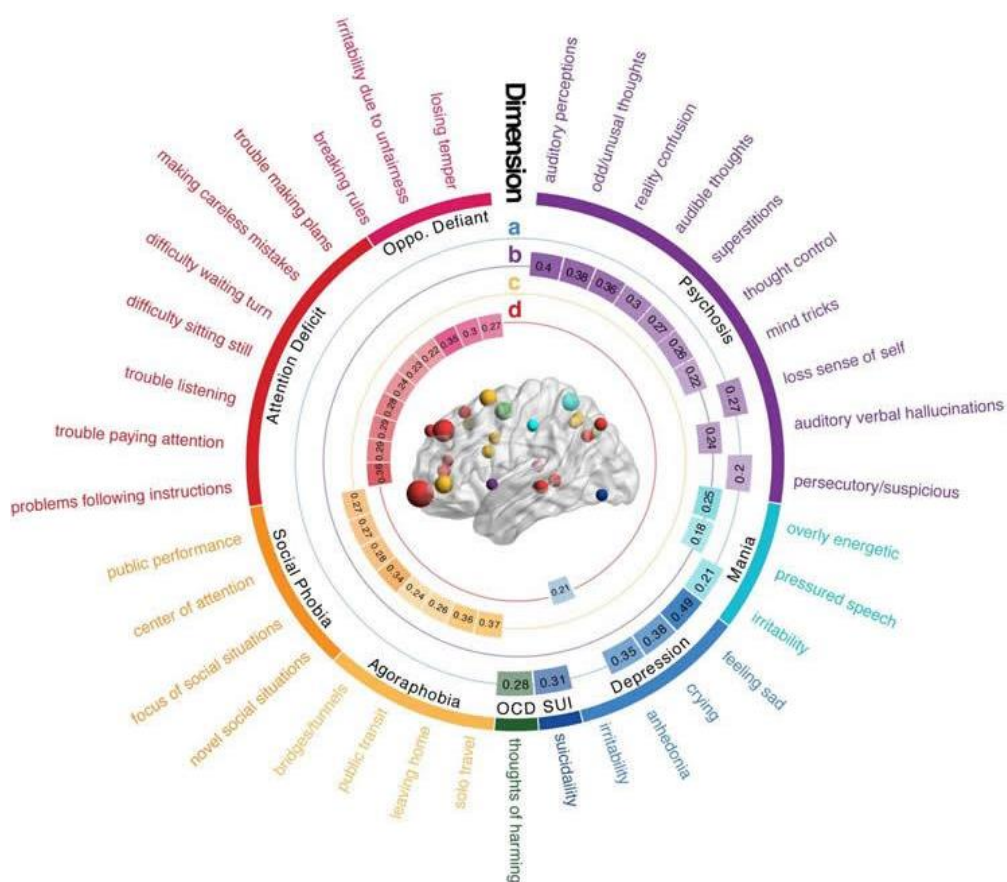


Рис.1.6 – Візуалізація результатів діагностування неврологічних захворювань

Всі ці дані вимірювання є вхідними даними для програмного забезпечення на основі якого працює та чи інша діагностична система. Різна структура даних, їх форма, тип даних вимагають додаткової апріорної обробки цих даних. А це у свою чергу вимагає додаткових затрат обчислювальних засобів, на кшталт, процесорів,

пам'яті та інших комп'ютерних засобів. Тому таке програмне забезпечення може потенційно розгортатися навіть на кластерних комп'ютерних системах і керування ними, зокрема, їх фізичними ядрами, потоками, пам'яттю на різних робочих станціях комп'ютерного кластеру.

Які технології використовуються для діагностування Паркінсона патології та програмне забезпечення розгляньмо у наступному підрозділі даного розділу.

1.3 Нейро Технологічні рішення усунення нейродегенеративних патологій

Діагностування Паркінсона захворювання призводить до розгляду різних видів терапії захворювання Паркінсона. Від нейрохірургічного втручання – видалення частини мозку – інвазивне лікування до неінвазивного лікування на основі різного виду і форми електростимулювання та магніто стимулювання.

У сьогодення технології дозволяють виконувати вживлення нейро трансплантатів, як наведено на рисунку.

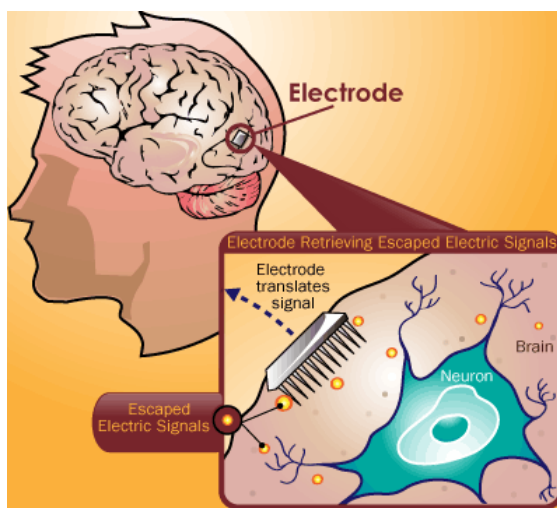


Рис.1.6 – Нейро Технологічні рішення для усунення неврологічних патологій

Нейротрансплантація вживаються прямо в мозок, тобто прямо у кору головного мозку. Звісно, що цим досягається максимальна терапія, максимальний рівень дії програмного забезпечення його команд через нейрочип.

Завдяки нейрочипа вдається досягти прямого програмного втручання у зміни стану мозку пацієнта і при потребі керувати станами роботи головного мозку. Тому розробка програмного забезпечення є надзвичайно актуальною.

Підсумовуючи вище наведений матеріал підійдемо до формування мети роботи та завдань, які вирішуватимуться у цій роботі.

Як зазначалося раніше точність, достовірність діагностування неврологічної патології Паркінсона є надзвичайно важливим завданням, а тому вирішення даного завдання можливе при використанні сучасних програмних інструментів, зокрема, інтелектуалізованих алгоритмів. За основу інтелектуалізованих алгоритмів у роботі буде використано алгоритми машинного навчання.

Зокрема, у роботі використано логічну парадигму машинного навчання – це регресійні дерева прийняття рішень для визначення лінійно апроксимаційного показника степені захворюваності на неврологічну патологію Паркінсона.

1.4 Мета і завдання бакалаврської роботи

Мета бакалаврської кваліфікаційної роботи.

Спроекувати та сконструювати програмне забезпечення на основі інтелектуалізованих алгоритмів для систем у сфері неврологічної діагностики ступені захворюваності на хворобу Паркінсона.

Завдання бакалаврської кваліфікаційної роботи.

1. На основі інтелектуалізованих алгоритмів спроекувати та сконструювати програмне забезпечення для системи неврологічної діагностики ступені захворюваності на хворобу Паркінсона.

2. Виконати оптимізацію гіпер параметрів спроектованого та сконструйованого програмного забезпечення для системи неврологічної діагностики ступені захворюваності на хворобу Паркінсона.

РОЗДІЛ 2. ПРАКТИЧНЕ ПРОЕКТУВАННЯ СИСТЕМИ ДІАГНОСТУВАННЯ НЕВРОЛОГІЧНОГО ЗАХВОРЮВАННЯ ПАРКІНСОНА

Практичне проектування системи неврологічного діагностування захворювання Паркінсона є не простою задачею і потребує детального розгляду процесу проектування. Необхідно, аналогічно, як писав Дональд Кнут розглянути процес проектування з точки зору алгоритмів і структур даних.

Проектування на рівні алгоритмічного програмування виконуватиметься з використанням програмних компонентів технології програмування пайтон, зокрема його бібліотек пандас, скітльорн, нумпай, матплотліб, тощо. Центральним програмним компонентом є регресійне дерево рішень. Саме цей програмний компонент буде використовуватися для встановлення рівня захворюваності пацієнта на Паркесонівську патологію. Бібліотеки нумпай та пандас використовуватимуться для маніпуляції даними пацієнтів, які підозрілі на неврологічне Паркесонівську патологію. Бібліотека матплотлиб використовуватиметься для візуалізації графічних оцінок метрологічних даних. Також використовуватимуться при проектуванні інші програмні компоненти.

Наступним етапом у проектуванні є структура даних з розгляду якої і розпочнеться даний розділ.

Проектування процесу введення даних у проект, який буде практичним представленням результатів з допомогою технології пайтон і відповідно використовуватимуться методи для введення даних у вигляді двовимірного масиву, який матиме обгортку у вигляді об'єкта даних пандас. Зручність створення програмного об'єкта пандас полягає в тому, що пандас має багато методів для маніпулювання даними та графічного виведення цих даних. Тому використовуватиметься метод читання із модуля пандас. Слід зауважити, що пандас,

як і нумпай забезпечує відразу паралельне читання даних, тому відбувається експоненційне прискорення при читанні, що прискорює роботу програмного забезпечення і більш продуктивно використовує програмні інструменти.

2.1 Інтерпретація введених даних для системи діагностування рівня захворюваності на хворобу Паркінсона

Для початку зазначимо, що дані пацієнтів читаються із файлу з розширенням csv за допомогою бібліотеки пандас із методом читання із вказанням гіпер параметрами структуру файлу. Метод читання має гіпер параметр для керування вказівкою де знаходяться назви змінних, тобто із вказанням індексу рядка. Також даний метод керується гіпер параметром, який вказує на роздільник між числовими даними у файлі та вказанням значення ще одного гіпер параметра для розділення цілої та дробової частин. А основним аргументом при цьому є адреса знаходження самого файлу.

Після читання файлу із даними одержується фрейм, який є об'єктом даних бібліотеки пандас. Для подальшого наочного розуміння суті введених даних далі розглянемо змістовну наповненість змінних.

Розглянемо природу (інтерпретацію) введених даних для програм. забезпечення, оскільки вона суттєво допоможе для формування процесів уявлення при практичному проектуванні на практиці.

Нехай у нас є фрейдівської структури дані df з відповідними ознаками

$$f_1(x^-), f_2(x^-), \dots, f_{21}(x^-) = y:$$

$f_1(x)$ – цілочисельне числове значення, яке ідентифікує кожен окремий об'єкт;

$f_2(x)$ – фізіологічний вік пацієнта;

$f_3(x)$ – стать суб'єкта: '0' – чоловіча стать, '1' – жіноча стать;

$f_4(x)$ – час з моменту реабілітації у експерименті. Ціла частина представляє собою кількість днів з моменту реабілітації;

$f_5(x), f_6(x), f_7(x), f_8(x), f_9(x)$ – сукупність показників зміни основної частоти тремтіння кінцівок;

$f_{10}(x), f_{11}(x), f_{12}(x), f_{13}(x), f_{14}(x), f_{15}(x)$ – сукупність показників зміни амплітуди тремтіння кінцівок;

$f_{16}(x), f_{17}(x)$ – показники відношення шумових даних до компоненти тональності у голосі;

$f_{18}(x)$ – нелінійний динамічно-орієнтований показник складності;

$f_{19}(x)$ – фрактальна структура сигналу масштабованого показника;

$f_{20}(x)$ – нелінійний показник варіації основної частоти;

$f_{21}(x) = y$ – цільова змінна. Загальний рівень (ступінь, який лінійно інтерпольований) уніфікованою рейтинговою шкалою хвороби Паркінсона, який використовується для моніторингу та спостереження протікання хвороби Паркінсона. Рейтингова шкала UPD найбільш використовувана шкала у клінічних дослідженнях хвороби Паркінсона.

Всі ознаки (атрибути), які представлятимуть собою колонки двовимірного масиву мають відповідні скорочення своїх назв.

Використання методів читання введених даних за допомогою інструментів пандас полегшує при проектуванні візуальне сприйняття даних завдяки градієнту кольору, який вказує на варіабельність значень змінних, тобто значень в межах окремих колонок. Також при проектуванні таким чином процесу уведення даних градієнт дасть можливість користувачеві проєктованого програмного забезпечення візуалізувати викиди у даних, тобто елементи не подібні до інших елементів даних.

На рисунку нижче наведено візуалізацію введених даних, наведено частинний вид, як видно з даного рисунку колонки представляють собою дані – характеристики пацієнтів та дані цільової змінної, яка характеризує стан неврологічної

Паркенсонівської патології. Таким чином, рисунок нище демонструє, як лікар-діагност у галузі неврології може бачити візуально дані.

Index	subject#	age	sex	test_time	total UPDR	Jitter(%)	Jitter(Abs)	Jitter:RAP	Jitter:PQ05	Jitter:DDP	Shimmer	immer(d)	immer:APC	immer:APC	immer:APQ	immer:DD	NHR	HNR
0	1	72	0	5.6431	34.398	0.00662	3.38e-05	0.00401	0.00317	0.01204	0.02565	0.23	0.01438	0.01309	0.01662	0.04314	0.01429	21.64
1	1	72	0	12.666	34.894	0.003	1.68e-05	0.00132	0.0015	0.00395	0.02024	0.179	0.00994	0.01072	0.01689	0.02982	0.011112	27.183
2	1	72	0	19.681	35.389	0.00481	2.462e-05	0.00205	0.00208	0.00616	0.01675	0.181	0.00734	0.00844	0.01458	0.02202	0.02022	23.047
3	1	72	0	25.647	35.81	0.00528	2.657e-05	0.00191	0.00264	0.00573	0.02309	0.327	0.01106	0.01265	0.01963	0.03317	0.027837	24.445
4	1	72	0	33.642	36.375	0.00335	2.014e-05	0.00093	0.0013	0.00278	0.01703	0.176	0.00679	0.00929	0.01819	0.02036	0.011625	26.126
5	1	72	0	40.652	36.87	0.00353	2.29e-05	0.00119	0.00159	0.00357	0.02227	0.214	0.01006	0.01337	0.02263	0.03019	0.009438	22.946
6	1	72	0	47.649	37.363	0.00422	2.404e-05	0.00212	0.00221	0.00637	0.04352	0.445	0.02376	0.02621	0.03488	0.07128	0.01326	22.506
7	1	72	0	54.64	37.857	0.00476	2.471e-05	0.00226	0.00259	0.00678	0.02191	0.212	0.00979	0.01462	0.01911	0.02937	0.027969	22.929
8	1	72	0	61.669	38.353	0.00432	2.854e-05	0.00156	0.00207	0.00468	0.04296	0.371	0.01774	0.02134	0.03451	0.05323	0.013381	22.078
9	1	72	0	68.688	38.849	0.00496	2.702e-05	0.00258	0.00253	0.00773	0.0361	0.31	0.0203	0.0197	0.02569	0.06089	0.018021	22.606
10	1	72	0	75.653	39.34	0.00465	2.553e-05	0.00238	0.0026	0.00715	0.02132	0.188	0.01069	0.01214	0.01844	0.03206	0.017443	25.672
11	1	72	0	82.653	39.834	0.00537	3.216e-05	0.00236	0.00278	0.00709	0.02377	0.282	0.01001	0.01375	0.02395	0.03003	0.017115	24.204
12	1	72	0	89.635	40.412	0.00524	3.287e-05	0.00235	0.00251	0.00704	0.02493	0.24	0.01176	0.01395	0.02019	0.03528	0.011876	22.203
13	1	72	0	96.633	41.034	0.00354	2.388e-05	0.00142	0.0015	0.00427	0.02107	0.171	0.00847	0.0104	0.0192	0.0254	0.015008	24.614
14	1	72	0	103.64	41.657	0.0053	3.181e-05	0.00241	0.00231	0.00724	0.02791	0.291	0.0131	0.0126	0.02069	0.0393	0.018093	23.533
15	1	72	0	110.65	42.28	0.00456	2.908e-05	0.00152	0.00194	0.00457	0.02878	0.264	0.01379	0.01494	0.02309	0.04138	0.020181	22.203
16	1	72	0	117.66	42.904	0.00693	3.93e-05	0.00329	0.00285	0.00987	0.0281	0.274	0.01468	0.0143	0.01952	0.04405	0.04198	20.878
17	1	72	0	124.64	43.524	0.00652	3.783e-05	0.00313	0.00311	0.0094	0.03011	0.32	0.01603	0.01733	0.02293	0.0481	0.031634	22.212
18	1	72	0	131.64	44.146	0.00571	3.711e-05	0.00296	0.00293	0.00889	0.02522	0.223	0.0126	0.01466	0.02145	0.0378	0.031546	23.129
19	1	72	0	139.69	44.861	0.00372	2.221e-05	0.00181	0.00195	0.00542	0.0323	0.288	0.01458	0.01732	0.02908	0.04373	0.010976	22.939

Рис.2.1 – Частина ознак пацієнтів, що підлягають діагностуванню на неврологічне захворювання Паркінсона

Використанням методу введення вхідних даних проектування програмного забезпечення не завершується. Тому у наступному підрозділі даного розділу розглядаються інші бібліотеки пайтона для обробки даних.

З візуалізації даних градієнтом кольору видно, що можна значення привести до одного діапазону їх варіювання, тобто можна, наприклад, їх стандартизувати – відняти середнє значення та розділити на середньоквадратичне значення. Тоді кожна змінна з однаковою величиною ваги буде впливати на регресійне дерево рішень при прогнозуванні ступені захворюваності на неврологічну патологію, а цим самим можна навіть визначити яку стратегію розробляти для організації процесу неврологічної терапії, зокрема, терапії захворювання Паркінсона. Таким чином, використання програмних компонент з бібліотеки пандас з точки зору проектування

дає широкий спектр функціональних можливостей при практичному конструюванні проекту даної кваліфікаційної бакалаврської роботи.

2.2 Використання бібліотек і фреймворків із репозиторію анаконди у проекті

Розпочнемо із послідовного процесу, по крокам проектувати програмну систему із описом в деталях та їх обґрунтуванням доцільності використання у програмному проекті.

Як було зазначено раніше використання інструментів пандас є корисним не лише для читання введених даних, але і для їх візуалізації, зокрема за допомогою градієнту кольору можна бачити диференціацію значень.

Пул використовуваних платонівських фреймворків та бібліотек можна розглянути нище по тексту.

```
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
import matplotlib.pyplot as plt
from sklearn.externals import joblib
```

Як видно, пул програмних компонентів базується на використанні бібліотеки скітлерн із дистрибутиву пайтона, зокрема, її програмного елеа – це регресійного дерева рішень, з модуля селекції моделей використання перехресної-перевірки функції, функції навчання та розщеплення даних на навчальну та тестову вибірки даних, функції метрик якості, що характеризують якість вирішення регресійних завдань на основі коефіцієнта детермінації, метрики, що представляє собою. Середнє

значення абсолютної похибки, середнє значення середнього значення квадрату відхилення, використання гіпер оптимізаторів, решіткового пошуку та гіпер оптимізатора випадкового пошуку, використання бібліотеки матплотлиб для різного роду підходів для візуалізації графіків, використання джобліб бібліотеки для зберігання та імпортування алгоритму у вигляді регресійного дерева рішень.

2.3 Програмний інструмент для введення даних у скрипт проекту

Програмний інструмент пандас для введення вхідних даних має для свого управління цілий набір гіпер параметрів. Серед яких є багато принципових, що допомагають краще організовувати процес читання даних.

До першого гіпер параметру слід віднести гіпер параметр, який задає адресу до файлового місцезнаходження, тобто можна сказати диск, папку, при потребі підпапку місця знаходження файлу з даними.

Інший гіпер параметр, який використовується – це гіпер параметр, який вказує на номер індекса рядка в якому знаходяться назви змінних, оскільки фреймова структура даних передбачає назви колонок із стрічковим типом даних. Слід відзначити, що назви колонок можуть бути розміщені у довільному рядку.

Наступним важливим гіпер параметром у проектуванні програмного забезпечення є гіпер параметр роздільник числових елементів даних у файлі. Цей гіпер параметр є важливим з причини того, що роздільниками можуть бути довільні знаки і їх можна обрати, на кшталт, роздільника елементів у файлі.

Наступним важливим гіпер параметром є роздільником між цілою частиною числового значення та його дробовою частиною, адже вона також може бути

представлена різним символьним знаком – комою, крапкою, крапка з комою, тощо. Тому це важливо при проектуванні програмного забезпечення, оскільки це забезпечує гнучкість при практичній розробці програмного забезпечення. Загальний вид програмної компоненти із її всіма гіпер параметрами є таким, як наведено нище, із абсолютно всі керуючими гіпер параметрами.

```
pandas.read_csv(  
    filepath_or_buffer, *, sep=_NoDefault.no_default,  
    delimiter=None, header='infer',  
    names=_NoDefault.no_default,  
    index_col=None, usecols=None,  
    dtype=None, engine=None,  
    converters=None, true_values=None,  
    false_values=None,  
    skipinitialspace=False,  
    skiprows=None, skipfooter=0,  
    nrows=None, na_values=None,  
    keep_default_na=True,  
    na_filter=True,  
    verbose=_NoDefault.no_default,  
    skip_blank_lines=True, parse_dates=None,  
    infer_datetime_format=_NoDefault.no_default,  
    keep_date_col=_NoDefault.no_default,  
    date_parser=_NoDefault.no_default,  
    date_format=None, dayfirst=False,  
    cache_dates=True, iterator=False,  
    chunksize=None, compression='infer',  
    thousands=None, decimal='.',
```

```

lineterminator=None, quotechar='\"',
quoting=0, doublequote=True,
escapechar=None, comment=None,
encoding=None, encoding_errors='strict',
dialect=None, on_bad_lines='error',
delim_whitespace=_NoDefault.no_default,
low_memory=True, memory_map=False,
float_precision=None, storage_options=None,
dtype_backend=_NoDefault.no_default
)

```

Фактичне використання при проектуванні програмного забезпечення такого програмного засобу дає у перспективі розробнику гнучкість у маніпулюванні при процесі розробки проекту.

Далі перейдімо до розгляду процесу проектування програмного забезпечення у вигляді конвеєра операцій – програмних компонент.

2.4 Проектування програмного конвеєра для апріорної переробки інформації

Першою операцією у цьому конвеєрі програмних компонент для апріорної обробки даних буде компонента, яка видаляє ідентичні рядки даних, аби регресійне дерево при діагностуванні не зациклюватися знаннями на тих даних, яких є просто на просто більше. Ця операція також представляє собою один із методів бібліотеки пандас і з його англійської назви випливає, що він витирає однакові рядки даних. також ним можна керувати за допомогою гіпер параметру колонками, тобто витирати подібні колонки.

Нище наведений даний метод бібліотеки пандас із всіма керуючими гіпер параметрами.

```
DataFrame.drop_duplicates(
subset=None,
*,
keep='first',
inplace=False,
ignore_index=False
)
```

Основним гіпер параметром при управлінні методом витирання подібних рядків є гіпер параметр, що вказує на номер рядка. Також його можна використати при нумерації колонок, але іншим індексом вказати номер колонки і тоді труться ідентичні колонки із фрейму даних.

Наступним методом, який завжди слідує після витирання дублів рядків використовується метод перевстановлення індексів рядків, оскільки після процесу витирання нумерація рядків порушується і подальші цикли неможливо виконувати над такими даними, тому перевстановлення індексів є вкрай особливим методом. Хоча, якщо апріорі відомо, що подібних рядків не має, то тоді не потрібно, перевстановлювати рядки, але і попереднє витирання їх тоді також є без потреби.

```
DataFrame.reset_index(
level=None,
*,
drop=False,
inplace=False,
col_level=0,
col_fill='',
allow_duplicates=_NoDefault.no_default,
names=None
)
```

Ключовими гіпер параметрами для керування даним методом перевстановлення індексів – переномерацією індексів у фреймі даних пандас є

гіперпараметри, що задають індекси колонок, наприклад початковий та кінцевий індекси колонок.

Також тут є багато інших важливих гіпер параметрів для керування даним методом, наприклад, можна програмно спроектувати, щоб цей метод одночасно виконував дві операції, а саме витирав однакові рядки та одночасно пронумерував індекси рядків у фреймі даних.

Одним із принципово важливих гіпер параметрів – це є гіпер параметр, який відповідає за запис результату виконання даних дій у змінну. Назва гіпер параметру – зміна місця має булеві значення. Якщо булеве значення є фальшу, то зміни над даними виконуються, але вони можуть бути записаними у іншу змінну. В такому режимі і працює даний метод по замовчуванню. Якщо ж його значення перевести у булеве значення істини, то він виконає дію і результат дії запишиться у цю ж змінну, яка використала метод пандаса перевстановлення індексів.

Послідовно перейдемо до наступної операції у пайплайн апріорної обробки даних – функції розбиття на тестову та навчальну вибіркою даних.

2.5 Програмний інструмент для утворення навчальної та тестової множин даних

Використання розбиття на тестову та навчальну вибірки даних виконується за допомогою скітлернівської функції трен тест спліт, яка має також великий ряд гіпер параметрів для свого керування, наприклад, розмір тестової вибірки чи розмір навчальної вибірки, які задаються нормованим числом із діапазону від нуля до

одиниці. Нище наведено платонівська нотація даної функції з усіма її керуючими гіпер параметрами.

```
sklearn.model_selection.train_test_split(
    *arrays,
    test_size=None,
    train_size=None,
    random_state=None,
    shuffle=True,
    stratify=None)
```

В розрізі вище наведеної функції правильно було би згадати ученого, завдяки працям якого ця процедура розділ на навчальну та тестову вибірки, як парадигма індуктивного навчання виникли. Фактично проектування програмного забезпечення на основі розбиття на навчальну і тестову множини даних потрібне для навчання регресійного дерева рішень завдяки академіку Національної Академії Наук України Івахненко, який розробив метод групового аргументу і цим самим створив нову парадигму логіки навчання машин і вважається творцем технологій глибокого навчання і інших напрямів машинного навчання.

Далі у роботі розглядається програмна компонента, яка реалізує регресійне дерево рішень.

2.6 Програмний інструмент для проектування регресійних дерев рішень

Проектування основного елемента прийняття рішень щодо встановлення діагнозу Паркінсона виконана за допомогою наступної програмної компоненти скітлернівської бібліотеки, яка наведене нище і з якої видно всі гіпер параметри для

керування нею при її навчання та тестуванні з метою максимізації точності прийнятих нею рішень щодо діагностування паркінсона.

```
sklearn.tree.DecisionTreeRegressor (  
*,  
criterion='squared_error',  
splitter='best',  
max_depth=None,  
min_samples_split=2,  
min_samples_leaf=1,  
min_weight_fraction_leaf=0.0,  
max_features=None  
random_state=None,  
max_leaf_nodes=None,  
min_impurity_decrease=0.0,  
ccp_alpha=0.0,  
monotonic_cst=None  
)
```

Регресійне дерево рішень для діагностування паркінсона в результаті проектування на цій програмній компоненті дає можливість керувати такими його різновидністю.

Одним із перших тут згаданих гіпер параметрів можна назвати гіпер параметр для керування вибору критерія на основі якого дерево вибирає ознаку по якій воно буде у вершині розщепити дані , наприклад це може бути критерій Джині, або критерій інформаційної ентропії Клода Шеннона, або дисперсійний критерій. Фактично цей гіпер параметр забезпечує гнучкість процесу розробки програмного забезпечення. Наступним Гіпер Параметром є гіпер параметр, який дає змогу керувати висотою дерева. З однієї сторони він дає гнучкість технології у розробці по висоті дерева і цим самим досягається висока точність прийнятих рішень з однієї сторони, але з іншої сторони збільшуються витрати обчислювальних ресурсів, а значить витрати на практичну реалізацію розроблених рішень.

Важливим гіпер параметром для керування також є параметр, або критерій зупинки росту та розщеплення дерева ж кількість листків у дереві, іншим гіпер

параметром по керуванню процесу зупинки розсту дерева є гіпер параметр, що задає кількість ярусів у дереві.

Також можна керувати ростом та розгалуженням дерев, якщо керувати значенням гіпер параметру, який відповідає кількості не термінальних вершин у регресійному дереві рішень щодо діагностування неврологічної Паркенсонівської патології.

Окремо важливим є поняття у регресійних дерев рішень – це прунінг, тобто після їх росту та розщеплення їх обрізка. Цей процес прунінгу дає можливість при проектуванні закласти гнучкість у процес конструювання практичної реалізації регресійного дерева рішень.

Дана програмна компонента скітлерн бібліотеки дає змогу організовувати у вигляді атрибутів цієї програмної компоненти селекцію характеристик по їх значимості, а тому можна при практичній реалізації впливати на підвищення точності при прийнятті рішень регресійним деревом рішень щодо континуальної вихідної змінної, яка є лінійною апроксимацією ступеня стандарту захворювання на патологію Паркінсона. Проте необхідно правильно організувати процес тестування регресійного дерева рішення, що можливо при використанні скітлернівської бібліотеки кросс-валідації (перехресної перевірки).

2.7 Інструмент для кросс-валідації регресійних дерев рішень

Перед тим, як описати і обґрунтувати чому при проектуванні використовується саме ця скітлернівська функція приводиться її вид.

```
sklearn.model_selection.cross_val_score(
    estimator,
    X,
    y=None,
    *,
    groups=None,
    scoring=None,
    cv=None,
    n_jobs=None,
    verbose=0,
    fit_params=None,
    params=None,
    pre_dispatch='2*n_jobs',
    error_score=nan)
```

Дана скітлернівська функція має, як аргументи так і атрибути. На вхід вона бере два аргументи – це вхідні дані, що представляють собою вище описані у цьому розділі характеристики пацієнтів, що підлягають медичному діагностуванню на неврологічну патологію та вихідні дані, що представляють собою континуально змінну, яка характеризує ступінь захворюваності на неврологічну патологію паркінсона, степінь представляє собою лінійну апроксимацію міжнародного показника стану глибини патології Паркінсона.

До ключових гіпер параметрів для управління перехресною перевіркою відноситься гіпер параметр, що задає об'єм тестової множини даних, а відповідно і об'єм навчальної множини даних на основі кількості фолдів. Чим більше задається фолдів тим точніше оцінюється точність, але витрачається більше обчислювальних ресурсів. І навпаки, якщо вибирається менша кількість фолдів, то менше

витрачається обчислювальних ресурсів, але і точність досягається меншою по своєму значенню.

Важливим питанням при проектуванні даного оцінювача на основі регресійного дерева рішень є вибір метрик якості для регресійного завдання і у цій роботі були обрані такі регресійні метрики якості, які описані у наступному пункті даного розділу.

2.8 Інструменти для оцінювання метрик якості регресійних дерев рішень

Важливим завданням у проєкті є оцінювання метрологічних характеристик якості вирішення регресійного завдання, оскільки медична діагностика неврологічної Паркесонвської патології представляє собою регресійне завдання. Важливість такого завдання обумовлено тим, що технічні засоби діагностики тих чи інших захворювань повинні бути гарантоздатними системами, тобто системами, які працюють з високою ступінню надійності та точності, оскільки ціна помилки є відчутною – це вироблення протоколу лікування пацієнта від захворювання, яке продіагностоване. Помилка в роботі програмної діагностичної системи – це ймовірність виникнення загрози у плані можливості неправильного призначення лікування. Таким чином оцінювання точності роботи системи – це надзвичайно важливе завдання. А тому у роботі розглядається випадок оцінювання точності роботи програмної діагностики неврологічної Паркінсона патології на основі трьох метрик якості, які використовуються у завданнях регресії.

Три метрики використовуються з багатьох причин, по-перше, необхідно різносторонньо побачити роботу програмної системи, по-друге, різні регресійні метрики якості щодо регресійної задачі мають різну інтерпретацію, а значить можна

на системному рівні зробити висновок щодо якості діагностики програмним забезпеченням.

Першою метрикою якості вирішення діагностичного регресійного завдання є коефіцієнт детермінації, тому проектування щодо подальшого вирішення його у коді планується на основі використання функції, що обчислює коефіцієнт детермінації і при цьому використати програмний засіб скітлернівської бібліотеки.

```
sklearn.metrics.r2_score(
y_true,
y_pred,
*,
sample_weight=None,
multioutput='uniform_average',
force_finite=True
)
```

Даний програмний компонент на свій вхід буде приймати два аргументи – перший це реальні значення вимірювальної величини, яка є вихідними даними, а другий аргумент – це прогнозовані значення цільової змінної на основі регресійного дерева рішень.

Значення коефіцієнта детермінації лежать у діапазоні від мінус нескінченності до плюс одиниці. Але ефективність алгоритму при цьому вважається значимо прийнятною, якщо його значення лежить у діапазоні від нуля цілих п'яти десятих до плюс одиниці. І чим більше значення у цьому діапазоні тим вважається є ефективнішим у роботі є регресійний алгоритм, тобто регресійне дерево рішень. Також існують інші інтерпретації для коефіцієнта детермінації, але у даній роботі вони не будуть задіюватися, оскільки їх інтерпретація не представляє інтересу у прикладному розумінні.

Іншою метрикою якості щодо оцінювання роботи регресійного дерева рішень при діагностуванні неврологічної Паркесонівської патології є метрика – середнє значення абсолютної похибки. При проектуванні обчислювального процесу для неї планується також використати програмний об'єкт скітлернівської бібліотеки із

модуля метрики, зокрема регресійну метрику якості. Оскільки її інтерпретація – це середнє значення абсолютної похибки, то звісно чим менше її значення тим краще працює регресійне дерево рішень при діагностуванні. Діапазон варіювання середнього значення абсолютної похибки є нормованим і лежить у діапазоні від нуля до одиниці. Тому чим значення середнього ближче до нуля тим краще, з вищою точністю встановлюється діагноз на захворювання чи не захворювання Паркенсона. З нормованості діапазону його варіювання від нуля до одиниці випливає, що значення середнього абсолютного можна виразити у відсотковому співвідношенні.

Для практичної реалізації у проекті планується використання бібліотеки скітлерн, зокрема її функцію, що має вид, який наведений далі по тексту.

```
sklearn.metrics.mean_absolute_error(
y_true,
y_pred,
*,
sample_weight=None,
multioutput='uniform_average'
)
```

Ця функція також, як і попередній коефіцієнт детермінації, на вхід приймає два аргументи. Один із аргументів – це реальні значення цільової функції – вихідна змінна, а другий аргумент – це значення спрогнозовано регресійним деревом рішень, як видно із гіпер параметрів вище наведеної функції.

Оскільки дана функція має гіпер параметр у вигляді вагової функції усіх значень, то можна цим гіпер параметром керувати вагою різних значень цільової змінної (вихідних даних). Тобто можна керувати пріоритетністю значень в діапазоні варіювання.

Також ця функція має гіпер параметр для управління середнім значення абсолютної похибки із різним розподілом ймовірностей. Але у даній роботі при проектуванні програмного забезпечення планується використовувати значення по замовчуванню, тобто використовувати рівномірний розподіл ймовірностей для множини значень вихідних даних.

Наступною метрику для оцінювання регресійного діагностування неврологічної Паркесонівської патології планується використання середнього значення квадрату різниці між реальним значенням діагностичної величини – цільової змінної (вихідної змінної). Процес проектування планує її використання з причини того, що вона не залежить від знаку цільової змінної і її значення не може бути з компенсованим, а тому її цінність також є прийнятною.

Для єе практичного кодового обчислення планується використати також скітлернівську бібліотеку та її модуль метрики для регресії, як наведено нище по роботі.

```
sklearn.metrics.mean_squared_error(
y_true,
y_pred,
*,
sample_weight=None,
multioutput='uniform_average',
squared='deprecated'
)
```

Як видно із кодового представлення у неї є також ряд своїх гіпер параметрів для керування її значеннями. По-перше, вона, як обидві попередні метрики має два аргументи – це реальні значення діагностичної величини, яка характеризує ступінь захворюваності на паркінсона, а також другий аргумент – це прогнозовані значення регресійним деревом рішень ступінь захворюваності на паркінсона. Також функція, яка реалізує дану метрику має гіпер параметр, який може керувати вагою значень з діапазону варіювання. Проте у даній кваліфікаційній планується використовувати її значення за замовчуванням, тобто використовувати рівномірний закон розподілу ймовірностей.

Діапазон значень даної метрики лежить у межах від нуля до одиниці і чим менше її значення тим краще діагностує неврологічну патологію регресійне дерево рішень щодо Паркенсонівської патології.

Підводячи підсумок даного підрозділу другого розділу можна стверджувати, що використання програмних компонент скітлернівської бібліотеки дає змогу різносторонньо представити метрологічну характеристику роботи регресійного дерева рішень щодо діагностування захворюваності пацієнта на неврологічну патологію Паркінсона.

Просуваючись по логіці процесу проектування програмного забезпечення у наступному підрозділі розглядатиметься проектування процесів гіпер оптимізації регресійного дерева рішень.

2.9 Програмні інструменти для гіпер оптимізації регресійних дерев рішень

Для ефективного використання регресійного дерева рішень при діагностиці патології Паркінсона у цій роботі на стадії проектування програмного забезпечення планується використання двох програмних гіпер оптимізаторів. Один з яких представляє собою перебір всіх значень із ґратки значень усіх гіпер параметрів – це ґрид серч програмний гіпер оптимізатор. Також планується використати стохастичний програмний гіпер оптимізатор, який організовує випадковий пошук по ґратці значень гіпер параметрів для регресійного дерева рішень. Його перевага полягає у тому, що він не використовує метод грубої сили – повний перебор значень по ґратці значень гіпер параметрів, а тому він є легшим у обчислювальному розумінні в порівнянні з ґрид серчем. Цей програмний гіпер оптимізатор називається рандом серч. Вони обидва є програмними об'єктами бібліотеки скітлерн і мають такий вигляд.

```
class sklearn.model_selection.GridSearchCV(  
    estimator,  
    param_grid,  
    *,  
    scoring=None,  
    n_jobs=None,  
    refit=True,  
    cv=None,  
    verbose=0,  
    pre_dispatch='2*n_jobs',  
    error_score=nan,  
    return_train_score=False  
)
```

```

class sklearn.model_selection.RandomizedSearchCV(
    estimator,
    param_distributions,
    *,
    n_iter=10,
    scoring=None,
    n_jobs=None,
    refit=True,
    cv=None,
    verbose=0,
    pre_dispatch='2*n_jobs',
    random_state=None,
    error_score=nan,
    return_train_score=False
)

```

Важливе місце посідає набір їх гіпер параметрів, які дають можливість при проектуванні програмного забезпечення гнучкості конструювання його. Тобто керувати оптимальністю його роботи на практиці. Розгляньмо завдяки чому досягається висока гнучкість практичної роботи цих обох програмних гіпер оптимізаторів. Першим параметром є вказання, який саме алгоритм машинного навчання із скітлернівських бібліотеки підлягатиме гіпер оптимізації. У випадку даної роботи – це буде регресійне дерево рішень для діагностування Паркінсона патології. Другим позиційним гіпер параметром є створення словникового типу даних гіпер параметрів регресійного дерева рішень і для кожного із них діапазонів варіювання їх значень. Причому діапазони значень можуть бути як числовими множинами так і не числовими множинами, наприклад, множиною стрічкових значень для випадку, що розглядається у цій бакалаврській роботі гіпер параметри будуть наступними – це кількість ярусів у регресійного дерева, кількість листків цього дерева, критерій розщеплення у вершині на різну кількість гілок, у цій роботі розглядається не регресійне дерево рішень, також можна використовувати багато інших гіпер параметрів для цього алгоритму діагностування, але у цьому процесі проектування не передбачається подальша гіпер оптимізація на основі інших гіпер параметрів. Окремо слід зауважити, що обидва програмні гіпер оптимізатори

можуть розпаралелювати свою роботу по процесорним ядрам обчислювальної машини.

Остаточним підрозділом цього розділу два – проектування є підрозділ, який представляє програмні компоненти для зберігання регресійного дерева рішень для можливості його використання сторонніми програмами при потребі, а також його завантаження при потребі використання його у режимі діагностування патології паркенсона, тобто в режимі прийняття рішень.

Наступний підрозділ використовувати в основному програмні об'єкти, методи та атрибути бібліотеки джоблаб. Ця бібліотека дає змогу не лише зберігати регресійне дерево рішень для діагностування неврологічної патології Паркінсона, але і зберігати його усі гіпер параметри та конструкцію. Також вона завдяки програмному об'єкта загрузки дає можливість завантажити це регресійне дерево рішень у інший скрипт програмного проекту і шляхом його використання в режимі прийняття рішень економляться затрати комп'ютерних ресурсів процесорів та його пам'яті, тому при проектуванні обрано саме цю бібліотеку.

2.10 Програмний інструмент для збереження та завантаження регресійних дерев рішень

Проектування процесу програмного зберігання планується реалізувати так, як наведено послідовно по тексту.

```
joblib.dump(
value,
filename,
compress=0,
protocol=None,
cache_size=None
)
```

Процес завантаження збереженого програмного об'єкта у питоно夫斯基й скрипт програмного проекту у кодовому представленні має ви такий.

```
joblib.load(  
filename,  
mmap_mode=None  
)
```

На цьому процес проектування програмного забезпечення у цій кваліфікаційній бакалаврській роботі завершено і у наступному розділі розглядатиметься процес практичного конструювання програмного забезпечення для діагностування неврологічної патології Паркінсона у пацієнтів.

РОЗДІЛ 3. ПРИКЛАДНА РЕАЛІЗАЦІЯ СИСТЕМИ ДІАГНОСТУВАННЯ НЕВРОЛОГІЧНОГО ЗАХВОРЮВАННЯ ПАРКІНСОНА

3.1 Задіяння бібліотек та фреймворків у пайтонівських скриптах проекту

Практична реалізація імпортування всіх необхідних пайтонівських фреймворків, бібліотек, функцій та модулів до скриптів практичного проекту цієї бакалаврської роботи у кодовому представленні виглядає наступним чином.

```
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
import matplotlib.pyplot as plt
from sklearn.externals import joblib
```

Як видно із вищенаведеного кодового представлення у практичному конструюванні використовуються пайтонівські бібліотеки скітлерн та пандас. А також використовуються їх окремі модулі та функції.

3.2 Введення набору даних в систему діагностування захворювання Паркінсона

Розглянемо першим кроком введення вхідних даних у конструйований пайтонівський проект. Для цього при конструюванні використовується той же алгоритм, що ми розглядали у попередньому розділі при проектуванні проекту даної бакалаврської роботи.

Виконується читання файлу текстового формату із розширенням "csv" інструментами – пайтонівськими методами так як видно нище у кодовому представлені.

```
"""Read dataframe"""
df = pd.read_csv("./PARKINSONS_TELEMONITORING/PARKINSONS_TELEMONITORING.csv",
                 delimiter=";", decimal=".", header=0)
```

Як видно з коду у даному проекті використовуються гіпер параметри для розділення числових даних у файлі та розділення цілої частини та дробової частини чисел, а також вказано, що назви змінних розміщені у нульовому рядку даного файлу.

Далі буде описано процес попередньої обробки потоку вхідних даних у даному проекті, що конструюється.

3.3 Априорна переробка набору даних хворих на захворювання Паркінсона

Попередня обробка потоку вхідних даних складається з наступних кроків. Першим кроком є викидання із вхідного потоку даних ідентичних дублікатів з метою того, щоб прийняття рішення машиною не мало систематичної похибки. Причому суттєвим нюансом тут є гіпер параметр, який відповідає за прописування змін у ту ж змінну чи іншу змінну. Значення гіпер параметру зміни результату дії за замовчуванням прописує результат операції в іншу змінну, тому у скрипті значення даного гіпер параметру змінено на значення логічної істинності і тепер зміни прописуються у ту ж саму змінну і цим самим економляться ресурси пам'яті на зберігання різних змінних. Наступним кроком є використання методу пандас, що перевстановлює індекси рядків, тобто переіндексовує рядки фрейму даних, оскільки

порядок у них може змінитися після витирання дублікатів – однакових рядів даних. Аналогічно цей метод, як і попередній має гіпер параметр для управління куди прописувати результат операції, його значення – значення гіпер параметру заміни місця переводиться в стан логічної одиниці. Завершення цієї частини кодового рішення виконується формуванням змінної, яка представляє собою список імен колонок, тобто списком назв змінних.

```
"""Preprocessing data"""
#Removing duplicate rows
df.drop_duplicates(subset=None, keep='first', inplace=True)
#Reset index of dataframe
df.reset_index(drop=True, inplace=True)
#Return column names of dataframe
column_names = list(df.columns)
```

Далі по тексту йтиме мова про розділення на вхідні та вихідні дані для регресійного дерева рішення для діагностування неврологічної патології Паркінсона.

3.4 Розщеплення набору даних на характеристики хворих та рівень захворюваності на Паркінсона

Розщеплення даних на вхідний потік даних та вихідний потік даних здійснюється адресуванням до назв колонок.

```
"""Splitting independent 'x' and dependent 'y' variables of dataframe 'df'"""
x = df[column_names[0:len(column_names)-1]].values
y = df[column_names[len(column_names)-1]].values
```

Крім цього метод велюкс перетворює вхідний та вихідний потоки даних у нумпаївські масиви даних, що забезпечує на практиці безперебійну роботу

програмних об'єктів скриптів практично реконструйованого проекту у даній кваліфікаційній бакалаврській роботі. Причина такого перетворення обумовлена практичним досвідом автора при програмуванні і виявленні зупинки програм при обставинах подачі фреймів даних на ті чи інші програмні компоненти різних пайтонівських бібліотек та фреймворків.

Наступним кроком даного розділу конструювання програмного забезпечення є проведення перехресної перевірки якості роботи регресійного дерева рішень щодо діагностування неврологічної патології Паркінсона.

3.5 Крос Валідація регресійних дерев прийняття рішень при діагностуванні захворювання Паркінсона

При практичному конструюванні спершу потрібно ініціалізувати із скітлернівської бібліотеки регресійне дерево та функцію оцінювання кросс-валідації.

```
"""Model cross-validation"""
#Initialization
dt = DecisionTreeRegressor()

#Compute the 'r2' by cross-validation
r2 = cross_val_score(dt, x, y, cv=5, scoring='r2').mean()
#Compute the 'mae' by cross-validation
mae = -1*cross_val_score(dt, x, y, cv=5,
                          scoring='neg_mean_absolute_error').mean()
#Compute the 'mse' by cross-validation
mse = -1*cross_val_score(dt, x, y, cv=5,
                          scoring='neg_mean_squared_error').mean()
```

Як видно із кодового представлення функція кросс валідації має два вхідних аргументи – це позиційні аргументи, перший є вхідний потік даних, а другий – це є вихідний потік даних.

Також є ряд гіпер параметрів для керування нею – це вибір кількості фолдів на яких регресійне дерево навчатиметься та тестуватимуть і на вибір метрик якості яких обчислюється середнє значення.

Також є гіпер параметр для вказування який тип метрики якості слід обчислювати для оцінювання якості роботи алгоритму діагностування неврологічної патології Паркінсона.

Із кодового представлення видно, що у цьому модулі програмування обчислюються усі метрики якості – коефіцієнт детермінації, середнє значення абсолютної помилки, середнє значення квадрата помилки.

У наступному підрозділі даного розділу розглядатиметься практичне конструювання процесу синтезу навчального і тестового множин даних.

3.6 Рандомне формування навчального та тестових наборів даних

В основі рандомного формування навчальної та тестової множини даних використовується на практиці скітлернівська функція трейн тест спліт, яка має два позиційні аргументи – вхідний потік даних та вихідний потік даних, а також гіперпараметр для керування – це розмір тестової множини даних, який встановлено у стані замовчування і дорівнює двадцять п'ять відсотків від усієї множини даних.

```
"""Splitting on training sample and test sample"""  
x_train, x_test, y_train, y_test = train_test_split(x, y,  
                                                    test_size=0.25)
```

Дана скітлернівська функція трейн тест спліт для вхідного та вихідного потоків даних повертає чотири позиційні вихідні змінні з яких першою є вхідна множина даних навчального набору, друга змінна – це вхідна множина даних тестового набору, третя змінна – це множина цільових даних для навчального вхідного набору, і четверта змінна – це вихідні дані для тестової множини даних.

У наступному підрозділі даного розділу розглядатиметься практичне конструювання процесу навчання та прийняття рішень регресійним деревом діагностування неврологічної патології Паркінсона.

3.7 Навчання, тестування та прийняття рішень на основі регресійних дерев прийняття рішень при діагностування захворювання Паркінсона

Розглянемо послідовно кроки даного кодового пайплайн у, якій наведено далі.

```
"""Training and testing model"""
#Initialization
dt = DecisionTreeRegressor()
#Training
dt.fit(x_train, y_train)
#Prediction
y_pred = dt.predict(x_test)

#Compute the 'r2'
r2_score = r2_score(y_test, y_pred)
#Compute the 'mae'
mae_score = mean_absolute_error(y_test, y_pred)
#Compute the 'mse'
mse_score = mean_squared_error(y_test, y_pred)
```

Відбувається практична ініціалізація регресійного дерева рішень, потім на основі методу підгонки виконується практичне конструювання етапу навчання регресійного дерева і у подальшому – це переведення його у етап прийняття рішень, причому чіткого прийняття рішень на основі методу предикшн, хоча програмні інструменти дають можливість практично сконструювати і сов класифікаційні рішення і цим самим повернути ранжовані діагностичні рішення із відповідними їм імовірностями. Далі виконується згідно практичного кодового пайплайн обчислення метрик якості – коефіцієнта детермінації, середнього значення абсолютної похибки та середнього значення квадрату різниці між реальним значення та прогнозом від регресійного дерева щодо діагнозу наявності тієї чи іншої ступені Паркінсона.

3.8 Формування фрейму метрик якості для регресійних дерев прийняття рішень у діагностуванні хвороби Паркінсона

Для конструктивності наведені кодові обчислення метрик якості формуються у єдиний фрейм даних, щоб збільшити структурування даних у середині проекту скриптів.

```
"""Metrics"""
print("Dataframe of metrics:",
      pd.DataFrame([r2, r2_score, mae, mae_score, mse, mse_score],
                    index=["r2", "r2_score", "mae", "mae_score", "mse", "mse_score"]))
```

Для цього використовуються пандасівські методи, на кшталт, методу для створення фрейму даних.

Використовується метод ДатаФрейм і видаються відповідні значення гіпер параметрам, тобто назви для індексів колонок, а також іншим гіпер параметром формуються із змінних самі значення колонок для даного пандасівського фрейму даних. Таким чином, організовується внутрішня структурація даних у практичному проекті із скриптами пайтона. У наступному підрозділі даного розділу розглядатиметься практичне конструювання процесів візуалізації значень метрик якості для регресійної задачі діагностування паркінсона патології.

3.9 Візуалізація значення метрики якості для регресійних дерев прийн. рішень для визначення рівня хвороби Паркінсона

Кодове представлення процесу візуалізації якості вирішення регресії представлено у коді нище, а також нище наведено графік, який візуалізує якість діагностичних рішень регрес. деревом.

```
"""Visualization of actual values vs. predicted values"""
fig, ax = plt.subplots()
ax.plot(y_test, y_pred, 'o')
ax.plot(y_test, y_test)
ax.set(xlabel='y_test', ylabel='y_pred',
       title='Visualization of actual values vs. predicted values')
ax.grid()
plt.show()
```

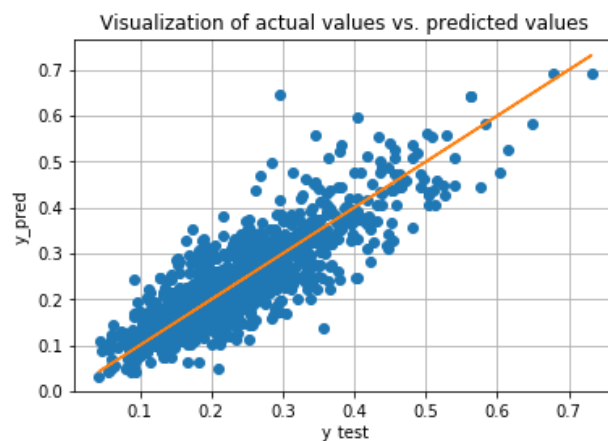


Рис.2.1 – Візуалізація точності регресійн. дерев прийн. рішень для визначення рівня захворюваності на хворобу Паркінсона

Для візуалізації використовується наступні гіпер параметри – це виставлення по вісі ординат реальних значень вихідної змінної, а по вісі абсцис прийнятих рішень регр. деревом щодо діагностичного рішення.

У наступному підрозділі даного розділу практичного конструювання програмного забезпечення розглядається зберігання та імпортування попередньо вивченої моделі навчання машинного у вигляді регресійного. дерева.

3.10 Зберігання та використання регрес. дерев прийня. рішень для визначення рівня хвороби Паркінсона

В основі практичної реалізації використовується кодове рішення на основі бібліотеки джобліб технології програмування пайтона, як видно нище по тексту.

```
"""Saving trained model. Loading trained model.
Decision making by trained model"""
#Saving trained model
joblib.dump(dt, 'reg.pkl')
#Loading trained model
dt = joblib.load('reg.pkl')
#Decision making by trained model
result = dt.predict(x_test)
```

Регресор – регресійне дер. зберігається, тобто пакується у файловий об'єкт на основі бібліотеки джобліб і методу дампу і тоді попередньо вивчений алгоритм машинного навчання – регресійне дерево для діагностування неврологічної патології Паркінсона збережеться у файлі з розширенням pkl. І тепер його можуть використовувати як скрипти пайтонівського проекту так і сторонні програми іншого мета проекту. Також дана бібліотека дає можливість за допомогою методу лод завантажити це попередньо навчене регресійне дерево рішень щодо діагностування неврологічної патології Паркінсона.

3.11 Гіпер Оптимізація регресійних дерев прийняття рішень для визначення рівня хвороби Паркінсона

Розглядається практичне конструювання двох гіпер оптимізаторів.

3.11.1 Гіпер Оптимізація регресійних дерев прийняття рішень для визначення рівня хвороби Паркінсона на основі GridSearchCV

```

"""Optimization of hyper-parameters based on GridSearchCV"""
#Label of the DecisionTreeClassifier
dt = DecisionTreeRegressor(criterion='mse', min_samples_split=2)

#Optimization
hyper_parameters = {'criterion':('mse', 'mae'),
                    'min_samples_split':[2, 3, 4]}
grid_optimal_dt = GridSearchCV(dt, hyper_parameters,
                               scoring='neg_mean_absolute_error')
grid_optimal_dt.fit(x, y)

#Hyper-parameters of optimal estimator
grid_optimal_dt.best_params_

#Computing mean cross-validation accuracy for optimal estimator
grid_optimal_dt_cross_val_accuracy = cross_val_score(grid_optimal_dt, x, y, cv=3,
                                                       scoring='neg_mean_absolute_error')
grid_optimal_dt_mean_accuracy = grid_optimal_dt_cross_val_accuracy.mean()

#Predict based on optimal estimator
y_pred_grid = grid_optimal_dt.predict(x_test)

```

3.11.2 Гіпер Оптимізація регресійних дерев прийняття рішень для визначення рівня хвороби Паркінсона на основі Random Search CV

```

"""Optimization of hyper-parameters based on RandomizedSearchCV"""
#Label of the DecisionTreeClassifier
dt = DecisionTreeRegressor(criterion='mse', min_samples_split=2)

#Optimization
hyper_parameters = {'criterion':('mse', 'mae'),
                    'min_samples_split':[2, 3, 4]}
random_optimal_dt = RandomizedSearchCV(dt, hyper_parameters,
                                       scoring='neg_mean_absolute_error')
random_optimal_dt.fit(x, y)

#Hyper-parameters of optimal estimator
random_optimal_dt.best_params_

#Computing mean cross-validation accuracy for optimal estimator
random_optimal_dt_cross_val_accuracy = cross_val_score(random_optimal_dt, x, y, cv=3,
                                                         scoring='neg_mean_absolute_error')
random_optimal_dt_mean_accuracy = random_optimal_dt_cross_val_accuracy.mean()

#Predict based on optimal estimator
y_pred_random = random_optimal_dt.predict(x_test)

```

Таким чином, у цьому останньому підрозділі даного розділу наведено останній крок у практичному конструюванні програмного забезпечення, яке є основою для підтримки прийняття рішень лікарем-діагностом відносно ступені неврологічної патології Паркінсона.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Моделювання та прогнозування небезпечних ситуацій

Жодна система чи операція не гарантує абсолютної безпеки. Та все ж доки ми не маємо 100% безпеки, ми намагаємося, наскільки це можливо, наблизитися до цієї мети. З плином часу різні заходи та методи, які використовуються для вирішення відповідних задач, удосконалюються, збільшуючи наші можливості у дослідженні систем, визначенні небезпек, виключенні або контролі за цими небезпеками, зниженні ризику до прийнятного рівня при роботі з цими системами. Аналіз небезпек починають з грубого дослідження, яке дозволяє в основному ідентифікувати джерела небезпек. Потім, при необхідності, дослідження можуть бути поглиблені і може бути виконаний детальний якісний аналіз. Методи цих аналізів та прийоми, які використовуються при їх виконанні, відомі під різними назвами. Нижче наведені основні з цих загальних інструментів.

Типи аналізу [10, 11]:

- попередній аналіз небезпек (ПАН)
- системний аналіз небезпек (САН)
- підсистемний аналіз небезпек (ПАН)
- аналіз безпеки робіт та обслуговування (АНРО).

Методи та прийоми, що використовуються при аналізах [10, 11]:

- аналіз пошкоджень та викликаного ними ефекту (АПВЕ)
- аналіз дерева помилок (АДП)
- аналіз ризику помилок (АРП)
- прорахунки менеджменту та дерево ризику (ПМДР)
- аналіз потоків та перешкод енергії (АППЕ)

- аналіз поетапного наближення (АПН)
- програмний аналіз небезпек (ПАН)
- аналіз загальних причин поломки (АКПП)
- причинно-наслідковий аналіз (В НА)
- аналіз дерева подій (АДПд)

Попередній аналіз небезпек - це аналіз загальних груп небезпек, присутніх в системі, їх розвитку та рекомендації щодо контролю. ПАН є першою спробою в процесі безпеки систем визначити та класифікувати небезпеки, які мають місце в системі. Проте в багатьох випадках цьому аналізу може передувати підготовка попереднього переліку небезпек.

ПАН звичайно виконується у такому порядку [10, 11]:

- вивчають технічні характеристики об'єкта, системи чи процесу, а також джерела енергії, що використовуються, робоче середовище, матеріали; встановлюють їхні небезпечні та шкідливі властивості;
- визначають закони, стандарти, правила, дія яких розповсюджується на даний об'єкт, систему чи процес;
- перевіряють технічну документацію на її відповідність законам, правилам, принципам і нормам безпеки;
- складають перелік небезпек, в якому зазначають ідентифіковані джерела небезпек (системи, підсистеми, компоненти), чинники, що викликають шкоду, потенційні небезпечні ситуації, виявлені недоліки.

Особливу увагу при ПАН приділяють вибухопожежонебезпечним та токсичним речовинам, можливим неконтрольованим реакціям та підвищенню тиску. Виявлені небезпеки досліджуються за допомогою інших методів аналізу. Основні питання, на які треба відповісти під час ПАН, допомагають уникнути

неповного аналізу безпеки системи. Проведення ПАН спрощується використанням матриць, анкет, списків і таблиць.

Аналіз логічного дерева подій (АЛДП) є важливим інструментом у процесі системної безпеки, особливо для оцінки складних систем. Цей метод дозволяє досліджувати умови, що можуть призвести до небажаних наслідків. АЛДП допомагає створити промислову програму техніки безпеки. Для побудови дерева помилок необхідно мати глибокі знання про систему чи процес. Створення дерева починається з визначення кінцевої події, яка може бути загальною (відмова системи) чи специфічною (порушення роботи компонента).

4.2 Механізація транспортування та встановлення важких за вагою деталей

Вимоги безпеки до транспортування вантажів повинні бути такими.

1. На підприємстві, що виконує вантажопідіймальні роботи кранами, мають бути розроблені та затверджені у встановленому порядку проект виконання робіт, технологічні карти складування вантажів, навантаження і розвантаження рухомого складу. У технологічних картах зазначаються місця розміщення стропальників під час переміщення вантажів і передбачається можливість їх виходу на естакади або навісні площадки.

2. Стоянку електрокарів та авто- і електронавантажувачів необхідно влаштовувати в гаражах або на критих площадках. Електрокари та авто- і електронавантажувачі встановлюють так, щоб не захарашувати проходи, проїзди, виходи.

3. Відкриті площадки для розміщення автомобілів повинні мати розмітку, яка виконується незмивною фарбою або іншими способами і визначає місце для установки автомобілів та проїздів. Уздовж стін, біля яких встановлюють автомобілі, повинен бути передбачений відбійний брус висотою: для автомобілів I категорії - 0,12 м; II і III категорій - 0,3 м; IV категорії - 0,4 м.

4. Причіпні візки до транспортних засобів повинні мати чітке визначення номінальної вантажопідймальності і гранично допустимої швидкості руху.

5. Під час перевезення довгомірних вантажів на автомобілі із причепом необхідно застосовувати поворотні кола із пристосуванням, яке закріплює коло при русі автомашины без вантажу.

6. Під час перевезення довгомірних вантажів їх необхідно надійно закріпити до причепа. При одночасному перевезенні довгомірних вантажів різної довжини коротші вантажі необхідно розташовувати зверху.

7. Транспортувати вантаж необхідно плавно, без різкого гальмування та крутих поворотів навантажувача.

8. Довгомірний або нестійкий вантаж при транспортуванні необхідно обв'язати так, щоб виключити можливість його розвалу чи падіння.

9. Під час транспортування вантажу необхідно витримувати відстань між габаритом навантажувача або вантажу, що транспортується, і устаткуванням, штабелями не менше ніж 0,5 м.

Вимоги безпеки до процесів проведення вантажно-розвантажувальних робіт.

1. Вантажно-розвантажувальні роботи повинні проводитися під керівництвом посадової особи або працівника, відповідального за безпечне проведення цих робіт.

2. Посадова особа або працівник, відповідальний за безпечне проведення вантажно-розвантажувальних робіт, повинен:

визначити спосіб навантаження і розвантаження;

зробити розміщення робочої сили;

перевірити справність пристосувань і засобів індивідуального захисту;

встановити порядок обміну умовними сигналами між стропальником, що подає сигнали, і машиністом підйимально-транспортного устаткування при провадженні робіт механічним способом;

провести інструктаж працівникам перед початком робіт.

3. До великовагових належать вантажі, які мають масу більше 500 кг.

4. Роботи з переміщення великовагового устаткування та машин дозволяються за наявності проекту вантажно-розвантажувальних робіт або технологічних карт складання вантажу.

5. Забороняється перебування працівників у місцях переміщення вантажу або його можливого перевертання.

6. Кантувати великовагові вантажі дозволяється на майданчиках, які мають огорожу.

7. Переміщення вантажів необхідно проводити плавно, без ривків і ударів. Залишати у висячому положенні піднятий вантаж, а також перебування працівників у зоні піднімання (опускання) вантажу забороняється.

8. Не допускається підтримувати, розвертати та направляти вантаж, що перебуває у висячому положенні, безпосередньо руками.

ВИСНОВКИ

1. Виконано проектування та на основі нього сконструйовано програмне забезпечення для діагностування неврологічного захворювання, зокрема, захворювання Паркінсона, що дає змогу реалізовувати підтримку прийняття рішення щодо діагностування патології лікарем-діагностом у сфері неврології і цим самим підвищити точність та достовірність встановленого ним діагнозу.

2. Виконано проектування та на основі нього сконструйовано програмні компоненти гіпер оптимізації програмного забезпечення, що дало покращення точності підтримки прийняття рішень у діагностиці неврологічного захворювання Паркінсона.

3. На основі програмних об'єктів бібліотеки Scikit-learn програмно оцінено метрологічні характеристики: коефіцієнт детермінації, середнє значення відносної похибки програмного забезпечення діагностування патології Паркінсона, що підтверджує коректність його використання у підтримці прийняття рішень невропатологом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Pastukh O., Petryk M., Bachynskiy M., Mudryk I., Stefanyshyn V. Processing of Cerebral Cortex Neurosignals from EEG Sensors and Recognizing Specific Types of Mechanical Movements Elements of Patient Limbs under the Cognitive Feedback Influences // International Workshop on Computer Information Technologies in Industry 4.0. 2023. №3468. С. 61–70.
2. Matplotlib 3.8.3 documentation. URL: <https://matplotlib.org/stable/index.html> (дата звернення: 07.03.2024)
3. Yasniy O., Pastukh O., Didych I., Yatsyshyn V., Chykhira I. Application of machine learning for modeling of 6061–T651 aluminum alloy stress– strain diagram // Procedia Structural Integrity. 2023. №48. С. 183-189.
4. Пастух О.А., Яцишин В.В., Жаровський Р.О., Шаблій Н.Р. Software tool for productivity metrics measure of relational database management system // Математичне моделювання. 2023. №1 (48). С.7-17.
5. Pastukh O., Stefanyshyn V., Baran I., Yakymenko I., Vasylyuk V. Mathematics and software for controlling mobile software devices based on brain activity signals // The International Workshop on Information Technologies: Theoretical and Applied Problems (ITTAP-2023). 2023. №3628. С. 684–689.
6. Yatsyshyn V., Pastukh O., Palamar A., Zharovskyi R. Technology of relational database management systems performance evaluation during computer systems design // Scientific Journal of TNTU. 2023. №1. С. 54–65.
7. Pastukh O., Petryk M., Bachynskiy M., Mudryk I., Stefanyshyn V. Processing of Cerebral Cortex Neurosignals from EEG Sensors and Recognizing Specific Types of Mechanical Movements Elements of Patient Limbs under the Cognitive Feedback Influences // International Workshop on Computer Information Technologies in Industry 4.0. 2023. №3468. С. 61–70.

8. Pandas documentation. URL: <https://pandas.pydata.org/docs/> (дата звернення: 07.03.2024)
9. NumPy documentation. URL: <https://numpy.org/devdocs/> (дата звернення: 07.03.2024)
10. Мелех Л.В. Безпека життєдіяльності та охорона праці: Навчальний посібник. Львів: Львівський державний університет внутрішніх справ. 2022. 219 с.
11. Сокурєнко В.В. Безпека життєдіяльності та охорона праці: Підручник. Харків: Харків. нац. ун-т внутр. справ. 2021. 308 с.

ДОДАТКИ

Додаток А

Технічне завдання

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

Кафедра програмної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедри ПІ
_____ Петрик М.Р.
"___" _____ 2024 р

РОЗРОБКА СИСТЕМИ ДЛЯ НЕВРОЛОГІЇ

ТЕХНІЧНЕ ЗАВДАННЯ

Вид робіт:

Кваліфікаційна робота

на здобуття освітнього ступеня "Бакалавр"
спеціальність 121 "Інженерія програмного забезпечення"

"УЗГОДЖЕНО"

керівник кваліфікаційної роботи

_____ д.т.н., проф. Пастух О.А.

"___" _____ 2024 р.

"ВИКОНАВЕЦЬ"

студент групи СП-42

_____ Яковів Б.Ю.

"___" _____ 2024 р.

Тернопіль 2024

ЗМІСТ

1. Підстави до розробки
2. Призначення до розробки
3. Вимоги до програмної системи
 - 3.1. Функціональні характеристики
 - 3.2. Склад та параметри технічних засобів
 - 3.3. Інформаційна та програмна сполучність
4. Стадії розробки
5. Програмна документація
6. Порядок контролю та приймання

1. ПІДСТАВИ ДО РОЗРОБКИ

Розробка програмної системи виконується у відповідності до графіку навчального плану на 2023 – 2024 н.р. та згідно наказу ректора Тернопільського національного технічного університету імені Івана Пулюя на виконання кваліфікаційної роботи на здобуття освітнього ступеня бакалавр.

Тема кваліфікаційної роботи на здобуття освітнього ступеня бакалавр: "Розробка системи для неврології" призначена для створення програмного інструменту у сфері діагностики неврологічних патологій.

2. ПРИЗНАЧЕННЯ РОЗРОБКИ

Програмна система – "Розробка системи для неврології" призначена для медичного діагностування неврологічних захворювань, зокрема, патології Паркінсона. Дана система може виконувати віддалену діагностику хворого на Паркінсона, тобто присвячена напряму телемедицини.

В основу такої інтелектуалізованої системи діагностики захворювання Паркінсона покладено використання програмний інструментів з репозиторію Анаконда, зокрема, такі інструменти: скітлерн, пандас, нумпай, тощо. Зокрема інтелектуалізованим програмним інструментом виступає регресійне дерево рішень із логічної парадигми алгоритмів машинного навчання.

Програмне забезпечення на основі характеристик людини визначає рівень чи ступінь її захворюваності, відповідно після чого можна людині прийняти рішення розпочати її терапію чи ні.

3. ВИМОГИ ДО ПРОГРАМНОЇ СИСТЕМИ

3.1. Функціональні характеристики

- формувати та візуалізувати дані діагностування щодо степені захворюваності на Паркінсона

- формувати та візуалізувати метрики точності діагностики

- оцінювати точність та достовірність діагностики

3.2. Склад та параметри технічних засобів

1) персональний комп'ютер з 4 Гб оперативної пам'яті, встановленою системою Windows 8, 8.1, 10. Не менше 200 Мб вільного місця на жорсткому диску. Двоядерний процесор із двома потоками та тактовою частотою від 1.2 GHz і більше.

2) наявність встановленого дистрибутиву Анаконди та ядра мови програмування Python із бібліотеками Pandas, Scikit-learn, Numpy.

3.3. Інформаційна та програмна сполучність

Програмна система повинна коректно функціонувати в операційній системі Windows різних поколінь. Програмні модулі програмної системи повинні працювати з високою ступінню надійності та точності. Розробку виконувати з використанням бібліотек та технологій мови Python в середовищі програмування Spyder.

4. СТАДІЇ РОЗРОБКИ

В ході реалізації роботи програмна система повинна пройти крізь стадії життєвого циклу розробки програмного забезпечення:

- аналіз предметної області

- проектування архітектури програмної системи

- практичне конструювання програмної системи

- реалізація певних елементів візуалізації даних

- тестування якості роботи окремих програмних модулів програмної системи
- оформлення супровідної документації
- здача роботи

5. ПРОГРАМНА ДОКУМЕНТАЦІЯ

Для розробленої системи для діагностування захворюваності на патологію Паркінсона повинні бути розроблені наступні документи:

- пояснювальна записка
- технічне завдання
- презентаційний матеріал
- додатки

6. ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Розроблена програмна система має виконувати всі вимоги, що складаються з перерахованих у даному технічному завданні.

Приймання проводиться спеціально створеною екзаменаційною комісією у відведені терміни, що вказані у календарному плані завдання кваліфікаційної роботи на здобуття освітнього ступеня бакалавр.

Тези доповіді на конференції

Міністерство освіти і науки України
 Тернопільський національний технічний університет
 імені Івана Пулюя
 Маріборський університет (Словенія)
 Технічний університет в Кошице (Словаччина)
 Каунаський технологічний університет (Литва)
 Львівський національний університет
 імені Івана Франка,
 Гірничо-металургійна академія ім. Станіслава Сташиця (Польща)
 Луцький національний технічний університет,
 Чернівецький національний університет
 імені Юрія Федьковича,
 Вроцлавський економічний університет (Польща)
 Університет технологій та економіки
 імені Хелени Ходковської (Польща)
 Донбаська державна машинобудівна академія



Студентське наукове
 товариство



VII МІЖНАРОДНА
студентська науково - технічна конференція
"ПРИРОДНИЧІ ТА ГУМАНІТАРНІ
НАУКИ.

АКТУАЛЬНІ ПИТАННЯ"

25-26 квітня 2024 р.

(збірник тез конференції)

Тернопіль 2024

Марків К. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КРУЇЗНОЇ КОМПАНІЇ З ВИКОРИСТАННЯМ SPRING FRAMEWORK	331
Музика В. АСИНХРОННІСТЬ ПРОГРАМУВАННЯ В РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	332
Петришин Я., Марцинюк Я. РОЛЬ СІ/СД У ПІДВИЩЕННІ ЕФЕКТИВНОСТІ ТА НАДІЙНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	333
Бурса В., Мудрик І. АВТОМАТИЗОВАНА СИСТЕМА ОЦІНКИ РІВНЯ ЗНАННЯ ІНОЗЕМНОЇ МОВИ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ	335
Антонюк Д., Пастух О. АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ АНСАМБЛЮВАННЯ АЛГОРИТМІВ	336
Чорна Х., Пастух О. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ АНАЛІЗУ ТА ПРОГНОЗУ ПАСАЖИРОПЕРЕВЕЗЕНЬ	337
Бабинець К., Пастух О. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ КЛАСИФІКАЦІЇ ТЕКСТОВОЇ ІНФОРМАЦІЇ	338
Бойко Д., Пастух О. ПРОГРАМА ДЛЯ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО КРЕДИТУВАННЯ	339
Яковів Б., Пастух О. ПРОГРАМНА СИСТЕМА ДЛЯ НЕВРОЛОГІЧНОЇ ДІАГНОСТИКИ	340
Гашинський Р., Мельник Н. АВТОМАТИЗОВАНИЙ АНАЛІЗ СХЕМИ ДОКУМЕНТІВ З ЧАСТКОВО-СТРУКТУРОВАНИМИ ДАНИМИ	341

УДК 629.735.072.8.08:004(043.2)

Яковів Б. – ст. гр. СП-42, Пастух О.

Тернопільський національний технічний університет імені Івана Пулюя

ПРОГРАМНА СИСТЕМА ДЛЯ НЕВРОЛОГІЧНОЇ ДІАГНОСТИКИ

Науковий керівник: д.т.н., професор Пастух О.А.

Yakoviv B., Pastukh O.

Ternopil Ivan Puluj National Technical University

SOFTWARE SYSTEM FOR NEUROLOGICAL DIAGNOSTICS

Supervisor: Dr.Sc., Professor Pastukh O.A.

Всесвітня Організація Охорони Здоров'я вказує на те, що у найближчі 15 років неврологічні захворювання займуть перше місце серед людей, витіснивши серцево-судинні захворювання.

Таким чином, щоб уникнути такого екстрапольційного прогнозу, необхідно розвивати експрес діагностику на основі нового програмного забезпечення. Зокрема, це потрібно для діагностування неврологічної патології Паркінсона.

У роботі пропонується програмне забезпечення для діагностування захворювання Паркінсона. Нижче на рис.1 наведено архітектуру програмного забезпечення у вигляді конвеєра операцій.

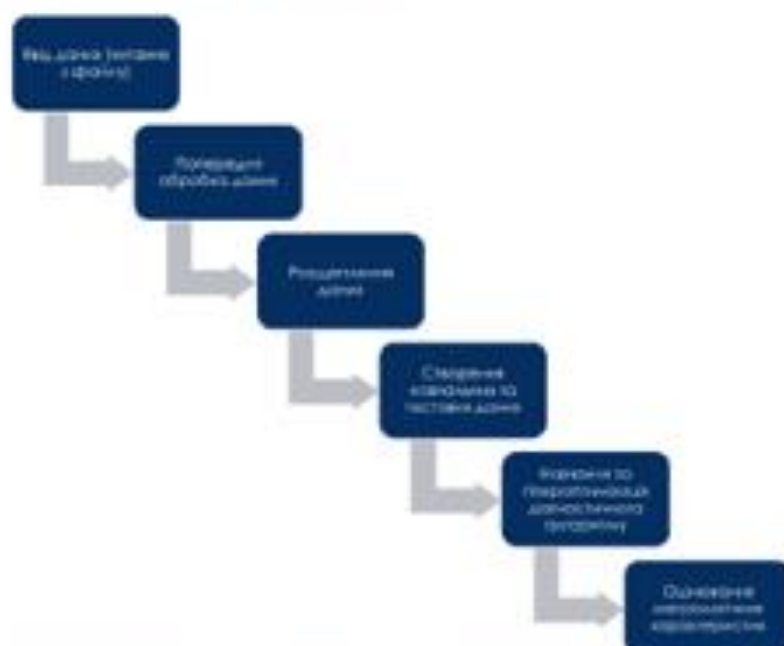


Рис. 1 – Конвеєр програмних операцій для автоматичної діагностики неврологічної патології Паркінсона

Зображена на рис.1 архітектура програмного забезпечення у вигляді конвеєра операцій практично реалізована з використанням технології пайтон.

Додаток В

Диск із кваліфікаційною роботою бакалавра