Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ МЕТОДІВ ВІДНОВЛЕННЯ ПОШКОДЖЕНИХ ДАНИХ
(тема)

Виконав:
студент 2 курсу, групи ІНФм-22-1
Безверха Є.В.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Руденко Д.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Безверхій Єлизаветі Володимирівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів відновлення пошкоджених даних

затверджена наказом по університету від 3 листопада 2023 року № 1280Ст

2. Термін подання студентом роботи до екзаменаційної комісії 25 грудня 2023 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, дані інтернет-мережі, UCI репозиторій.

4. Перелік питань, що потрібно опрацювати в роботі

1. Аналіз предметної області.
2. Порівняльний аналіз відомих методів відновлення пошкоджених даних.
3. Підбір програмних засобів та програмна реалізація.
4. Експериментальні дослідження методів відновлення даних.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) аналіз предметної області, постановка задачі, адаптивне відновлення пропущених даних, аналіз отриманих результатів.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	03.11.2023	
2	Аналіз завдання, підбір літератури	04.11.23-06.11.23	
3	Аналіз літератури з досліджуваної проблеми	06.11.23-08.11.23	
4	Аналіз технічних засобів	08.11.23-09.11.23	
5	Розробка адаптивного методу відновлення даних	09.11.23-10.11.23	
6	Програмна реалізація	10.11.23-14.11.23	
7	Оформлення пояснювальної записки	15.11.23-30.11.23	
8	Перевірка на плагіат	10.12.2023	
9	Рецензування	20.12.2023	
10	Підготовка презентації та доповіді	25.12.2023	
11	Занесення роботи в електронний архів	04.01.2024	
12	Попередній захист кваліфікаційної роботи	04.01.2024	

Дата видачі завдання 3 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Руденко Д.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 62 с., 11 табл., 33 рис., 42 джерела.

ЧАСТКОВА ВІДСТАНЬ, ТЕСТОВА ВИБІРКА, ЗАПОВНЕННЯ ПРОПУЩЕНОГО ЗНАЧЕННЯ, ПРОПУЩЕНЕ ЗНАЧЕННЯ, РІВЕНЬ НАЛЕЖНОСТІ.

Об’єктом дослідження є відновлення пропущених значень в тестовій вибірці.

Метою дослідження є аналіз методів відновлення втрачених спостережень та вибір найкращого метода для відновлення викривлених пропусками даних в тестових наборах.

В роботі розглянуто аналіз методів заповнення пропущених спостережень, розглянуто переваги і недоліки класичних методів відновлення пропусків в даних. Обрано методи, які можна застосовувати у відновленні специфічних наборах даних, таких як медичні та клінічні дані.

Результатом роботи є вибір основних методів відновлення пропусків в тестових наборах даних, які можуть бути корисними для подальшого інтелектуального аналізу та різноманітних задачах прийняття рішень.

PARTIAL DISTANCE, TEST SAMPLE, FILL MISSING VALUE, MISSING VALUE, MEMBERSHIP LEVEL.

The object of the research is the restoration of missing values in the test sample.

The purpose of the research is to analyze the methods for restoring lost observations and to choose the best method for restoring data distorted by gaps in test sets.

The paper examines the analysis of methods for filling in missing observations, examines the advantages and disadvantages of classic methods of restoring gaps in data. Techniques are selected that can be applied in the recovery of specific data sets, such as medical and clinical data.

The result of the work is a selection of the main methods of gap recovery in test data sets, which can be useful for further intellectual analysis and various decision-making tasks.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШНМ – штучні нейронні мережі

MAR – Missing At Random (випадковий пропуск)

MCAR – Missing Completely At Random (абсолютно випадковий пропуск)

MNAR – Missing Not At Random (не випадковий пропуск)

MBND – Missing By Natural Design (природній пропуск)

EM – Expectation Maximization (алгоритм максимального очікування)

k-NN – *k*-Nearest Neighbors (метод *k* найближчих сусідів)

MLE – Maximum Likelihood Estimation (метод максимальної правдоподібності)

ВСТУП

Сьогодні технології обчислювального інтелекту досить часто і успішно використовуються у вирішенні складних завдань, які, як правило, не мають аналітичного рішення. Ці технології та особливо штучні нейронні мережі (ШНМ) набули широкого поширення для вирішення різних задач обробки сигналів, оптимізації, оптимального та адаптивного управління, розпізнавання образів, ідентифікації, прогнозування в реальному часі тощо. Створено реальні системи обробки зображень та комп'ютерного зору, управління аерокосмічними об'єктами, технічної та медичної діагностики, в економіці та фінансах, у військовій справі, управління рухом, в енергетиці, в криміналістиці, аналізу сигналів різної природи та інше, причому цей перелік розширюється практично щодня.

На даний момент є достатня кількість інформації про діяльність підприємств, лікарень, фірм, яка відображає діяльність цих об'єктів. Проаналізувавши цю інформацію, можна визначити об'єктивні закономірності за умови, що сформована таблиця містить фактичні дані, які відображають причинно-наслідкові залежності.

Ця інформація збирається протягом багатьох років (наприклад, темпи інфляції, рівень доходів населення, структура витрат населення, вартість послуг житлово-комунального господарства, стан промислового та сільськогосподарського виробництва, своєчасність виплати заробітної плати та пенсій тощо). Зрозуміло, що такі дані складно аналізувати вручну через великий обсяг інформації та складні нелінійні причинно-наслідкові залежності. Тому і виникла потреба у розробці нових методів аналізу та прогнозування, до яких належать машинні методи виявлення закономірностей.

В багатьох задачах інтелектуального аналізу даних, пов'язаних з обробкою кількісних емпіричних спостережень, дані можуть бути спотворені пропусками. Задачі відновлення таких спостережень приділялося достатньо

уваги, найбільш ефективними у цій ситуації виявилися підходи, засновані на математичному апараті обчислювального інтелекту і, перш за все, штучних нейронних мережах і нечітких систем, які вирішують задачі відновлення втрачених спостережень.

Проте, описані підходи до відновлення даних працездатні лише у випадках, коли вихідні дані задані апріорно, а сама таблиця «об'єкт-властивість» чи часовий ряд мають фіксовану кількість спостережень, тобто не змінюються у процесі обробки.

Для роботи з такими даними найефективнішими є штучні нейронні мережі, нейро-фаззі системи, гібридні системи, відомі своїми універсальними апроксимуючими властивостями та здатністю до навчання, під яким зазвичай розуміється можливість налаштування їх параметрів шляхом оптимізації деякого критерію якості (цільової функції, критерію навчання). У більш широкому значенні можна настроювати не тільки параметри, але і архітектуру системи.

В даний час існує цілий ряд підходів та рішень, хоча найбільшого поширення набув, так званий, конструктивний підхід, при якому система обчислювального інтелекту, стартуючи з найпростішої архітектури, поступово нарощує свою складність, одночасно налаштовуючи і свої параметри, до досягнення необхідної якості рішення задачі.

Зрозуміло, що для нормального функціонування нейронної мережі або гібридної системи ці відсутні дані повинні бути якимось чином відновлені. Природно, що в задачах, коли дані надходять на обробку послідовно в реальному часі, кількість пропусків наперед невідома, а сама послідовність має нестаціонарний характер, тому відомі підходи є малоефективними.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ

1.1 Моделі викривлених даних

Проблема втрати даних завжди була актуальною, оскільки втрати даних можуть статися з різних причин, включаючи випадкове видалення, пошкодження файлів, атаки хакерів, віруси, виходження з ладу обладнання та інші небезпечні ситуації. Однак з розвитком комп'ютерної техніки та технологій зберігання даних почали розвиватися методи відновлення втрачених даних.

Важливою подією в історії відновлення даних було винайдення резервного копіювання. Перші системи резервного копіювання даних виникли в 1950-х роках зі зростанням популярності комп'ютерів. Протягом 1960-х і 1970-х років, з розвитком магнітних стрічок, з'явилися більш ефективні засоби резервного копіювання.

Спотворені дані можуть бути охарактеризовані рядом моделей, що представлені на рисунку 1.1, який демонструє представлення даних у вигляді рядків та стовпців, де рядки – це номер спостереження, а стовпці, у свою чергу, відповідають різним змінним даних.

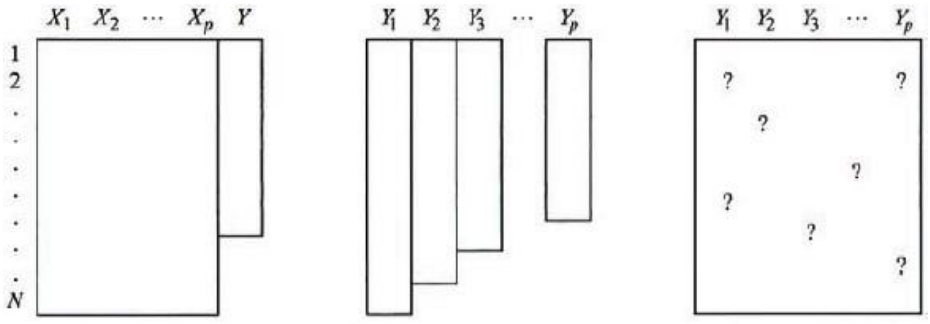

Рисунок 1.1 – Шаблони відсутніх даних: а) одновимірний шаблон; б) монотонний шаблон; в) довільний шаблон

Щодо спроб відновлення втрачених даних, перші спроби відновлення даних з носіїв даних виглядають не так вже й новими.

Для відновлення пропущених даних необхідно знати причину їх викривлення, оскільки це спрощує роботу з даними та підбір методу відновлення таких даних відіграє істотну роль. Змінна або функція набору даних розглядається як окремий механізм, який дає зрозуміти причину і вловити якусь закономірність втрати даних. У ході аналізу даних можуть бути отримані також і приховані закономірності. У такому разі дані мають різну природу пошкодження і діляться на чотири типи:

– випадковий пропуск (Missing At Random (MAR)), описує випадок, коли пропущене значення змінної спостереження не пов'язане з іншими пропусками даних;

– абсолютно випадковий пропуск (Missing Completely At Random (MCAR)) відноситься до того випадку, коли пропущене значення не відноситься ні до значень пропущених даних, ні до повних;

– не випадковий пропуск (Missing Not At Random (MNAR)), відноситься до випадку, коли пропуск пов'язаний з іншими пропущеними даними у вибірці;

– природній пропуск (Missing By Natural Design (MBND)), відноситься до випадку, коли дані не отримані спочатку, тобто не можуть бути виміряні фізично чи природно.

Сучасні методи відновлення даних використовують різні техніки, включаючи програмне забезпечення відновлення даних, технології відновлення на рівні файлової системи, а також спеціалізовані сервіси відновлення даних, які можуть відновлювати дані з фізично пошкоджених носіїв даних.

Узагальнюючи, проблема втрати даних і їх відновлення завжди була присутньою, але з розвитком технологій зберігання даних та відновлення їх становище суттєво поліпшилося з появою більш ефективних та надійних методів відновлення.

1.2 Аналіз методів відновлення даних

Задача відновлення пропущених значень полягає у заповненні пропусків у таблицях «об’єкт-властивість» та часових рядах найбільш правдоподібними значеннями з найменшою похибкою.

Задачам відновлення таких пропущених спостережень приділялася достатня увага [1-4], при цьому дуже ефективними в даній ситуації виявилися підходи, засновані на математичному апараті обчислювального інтелекту і, насамперед, штучних нейронних мережах [5-9]. Разом з тим, описані підходи до відновлення пропусків працездатні лише у випадках, коли вихідна таблиця даних задана апріорно і кількість її рядків або стовпців не може змінюватися в процесі обробки.

Методи відновлення даних можна поділити на кілька основних категорій в залежності від принципу їх роботи та застосування. Нижче наведено аналіз деяких основних методів відновлення даних:

– програмне забезпечення відновлення даних: зазвичай дешевше порівняно із зверненням до професійних служб відновлення даних. Деякі програми дозволяють відновлювати втрачені файли з видалених або пошкоджених носіїв даних. Простіше використання для невеликих обсягів даних. Можливість відновлення обмежена для важко доступних файлів або в разі серйозних пошкоджень носіїв. Не завжди допомагають у відновленні втрачених або пошкоджених файлів з високою точністю;

– професійні служби відновлення даних. Висока ефективність відновлення важко доступних даних. Використання спеціалізованого обладнання та програмного забезпечення. Здатність відновлювати дані з фізично пошкоджених носіїв. Високі витрати на послуги відновлення. Тривалий час відновлення, особливо в разі складних випадків;

– відновлення на рівні файлової системи. Швидкий доступ до видалених або втрачених файлів через системні резервні копії. Здатність відновлювати попередні версії файлів та видалені файли, які можуть бути

відновлені. Обмежена місткість системних резервних копій. Можливість втрати даних, які були створені після останньої резервної копії;

– застосування методів штучного інтелекту. Здатність генерувати втрачені або пошкоджені дані, що може бути корисно в певних сценаріях. Потенційно висока точність відновлення для різних типів даних. Складність налаштування та використання глибоких нейронних мереж. Обмеженість у відновленні деяких структур або форматів даних.

Алгоритми заповнення пропусків в таблицях даних можна поділити на дві групи: складні та прості.

До простих алгоритмів належать такі методи заповнення таблиць як:

– метод заповнення таблиць середнім арифметичним;

– алгоритм найближчого сусіда;

– метод багатовимірної лінійної регресії.

Складні алгоритми, своєю чергою, поділяються на частини: глобальні і локальні.

До глобальних алгоритмів належать:

– метод Барлетта;

– алгоритм resampling;

– алгоритми максимальної правдоподібності (МП);

– алгоритми максимального очікування (EM-expectation maximization);

– локальний – алгоритм ZET.

1.2.1 Прості алгоритми заповнення пропусків в таблицях

Найбільш поширені алгоритми заповнення таблиць ґрунтуються на методах інтелектуального аналізу даних (Data Mining)[10-12].

Метод заповнення середнім арифметичним. Цей метод досить широко використовується в більшості статистичних пакетів як засіб боротьби з пропусками в таблицях, найпростіший, але не точний метод заповнення

пропущених даних. Відповідно до цього методу, рекомендується заповнювати пропущені значення таблиць середніми значеннями величин, що є в даному стовпці.

Таким чином, невідоме значення $\hat{x}_{ki}$ в матриці X розміру $N \times n$ передбачається за формулою:

$$\hat{x}_{ki} = \frac{\sum_{i=1}^{n} \check{x}_{ki}}{n-1}. \quad (1.1)$$

Експериментальні дослідження показали неспроможність даного методу навіть на простих наборах даних, не кажучи вже про роботу з даними, що мають велику розмірність і надходять на обробку в реальному часі.

Метод *k*-найближчих сусідів (*k*-Nearest Neighbors або *k*-NN) – це простий і ефективний алгоритм для класифікації і регресії. Він використовує навчальний набір даних для передбачення класу (у випадку класифікації) або значення (у випадку регресії) для нового зразка шляхом знаходження k-найближчих до нього точок у навчальному наборі [13].

Кроки роботи алгоритму *k*-NN:

Крок 1. Вибір параметра *k*: визначить кількість сусідів (*k*), яку враховувати при передбаченні класу чи значення для нового зразка.

Крок 2. Вимірювання відстаней: визначте метрику відстані, таку як евклідова відстань, манхеттенська відстань, тощо.

Крок 3. Знаходження *k*-найближчих сусідів: знайдіть *k*-найближчих точок в навчальному наборі, які мають найменш відстань до нового зразка.

Крок 4. Визначення класу чи значення: якщо ви робите класифікацію, визначте клас, який найчастіше зустрічається серед *k*-найближчих сусідів. У випадку регресії, середнє значення *k*-найближчих сусідів може бути передбаченим значенням.

Переваги методу *k*-NN:

– простота та легкість реалізації;

– не потребує навчання моделі перед використанням;

– добре працює для відносно невеликих наборів даних та кількісних характеристик.

Недоліки методу *k*-NN:

– чутливість до масштабу даних та вибору параметра *k*;

– велика обчислювальна витрата, особливо для великих наборів даних;

– погано працює для наборів даних з великою кількістю ознак (зокрема, коли більше ознак, ніж спостережень).

Метод *k*-NN використовується у багатьох задачах, включаючи класифікацію текстів, розпізнавання образів, рекомендаційні системи та інші сфери. Правильний вибір параметра *k* та відповідна обробка та масштабування даних можуть покращити результати роботи алгоритму *k*-NN.

Метод багатовимірної лінійної регресії. На першому етапі роботи методу, список незалежних змінних порожній. Спочатку розглядаються всі лінійні залежності від однієї змінної і для кожної обчислюється стандартна помилка та значення F-статистики. Далі вибирається незалежна змінна, обчислюється мінімальна помилка та включається до списку, аналогічним чином відбувається з другою змінною. Такі ж дії відбуваються і у двовимірній лінійній моделі. Далі йде порівняння моделей та вибирається найкраща серед них. Таким чином на кожному кроці додається нова незалежна змінна.

Цей процес триває до тих пір, поки певний рівень достовірності не перестане покращуватися в порівнянні з попереднім. До недоліків даного методу слід віднести те, що даний метод погано працює з даними, в яких мало чисел і багато дискретних полів, а також може зовсім не виявити наявності будь-якої залежності в даних

1.2.2 Складні алгоритми заповнення пропусків в таблицях

Існують і складніші алгоритми заповнення пропусків. Метод Барлетта і resampling метод засновані на рівняннях багатовимірної лінійної регресії.

Метод Барлетта. В цьому методі спочатку пропущеного значення присвоюється початкове значення (наприклад, середнє арифметичне стовпця), далі йде побудова рівняння багатовимірної лінійної регресії з використанням початкового значення пропуску, і після аналізу цього рівняння передбачається пропущене значення.

Метод «resampling» – це загальна стратегія в області машинного навчання, де використовується підвибірка або перенавчання на даних, які вже були використані для навчання моделі. Цей підхід використовується для уникнення перенавчання (overfitting) та покращення загальної здатності моделі до узагальнення на нових даних.

Існують різні методи resampling, включаючи перехресну валідацію, створення нових підвибірок даних шляхом вибору даних з заміщенням, тощо. Користування різними методами resampling може допомогти у покращенні якості моделі та уникненні перенавчання на навчальних даних. Користуючись цими методами, можна отримати більш надійні оцінки ефективності моделі, особливо якщо ви маєте обмежені обсяги даних для навчання.

Метод максимальної правдоподібності (Maximum Likelihood Estimation, MLE) – це статистичний метод, який використовується для оцінки параметрів у статистичних моделях. Основна ідея полягає в тому, що ми шукаємо ті значення параметрів, при яких дані, які ми спостерігаємо, мають найбільшу ймовірність.

Для зрозуміння MLE давайте розглянемо приклад. Нехай у нас є вибірка з нормального розподілу з невідомими параметрами середнього (μ) і стандартного відхилення (σ). Ми хочемо знайти такі значення μ і σ, які найкраще пояснюють наші спостереження.

У MLE для нормального розподілу, функція правдоподібності, яка визначає ймовірність нашої вибірки при заданих параметрах μ і σ, виглядає так:

$$L(\mu,\sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}, \quad (1.2)$$

де x_i – i-те спостереження;

n – кількість спостережень.

MLE вирішує задачу максимізації функції правдоподібності відносно параметрів μ і σ.

Отже, метод максимальної правдоподібності надає нам спосіб знаходження параметрів моделі, які найбільш ймовірно пояснюють наші спостереження, враховуючи ймовірність того, що саме ці спостереження можуть виникнути при виборі цих параметрів.

Метод максимального очікування (Expectation-Maximization, EM) – це ітеративний статистичний метод, який використовується для розв’язання задач, пов’язаних з неповними чи відсутніми даними та моделями з латентними (прихованими) змінними. EM метод ділиться на два основних кроки: крок очікування (Expectation Step) та крок максимізації (Maximization Step).

Крок очікування. На цьому кроці EM оцінює параметри латентної моделі, враховуючи неповні або відсутні дані. Визначаються очікувані значення латентних змінних або параметрів, які недоступні для спостережень.

Крок максимізації. На цьому кроці EM оновлює оцінки параметрів моделі, використовуючи спостережені та розраховані очікувані значення з попереднього кроку. Параметри моделі оновлюються, щоб максимізувати функцію правдоподібності для спостережень, враховуючи розраховані на попередньому кроці очікувані значення.

Ці два кроки повторюються ітеративно до збіжності, тобто до того моменту, коли параметри моделі перестають змінюватися або змінюються дуже мало.

ЕМ метод є дуже потужним і широко використовується в статистиці та машинному навчанні, особливо в задачах, де є неповні або відсутні дані, таких як кластеризація, факторний аналіз, сегментація зображень та інші. Цей метод дозволяє розв'язувати складні завдання навіть у випадках, коли частина інформації втрачена чи недоступна.

Алгоритм ZET. Одним із найбільш ефективних локальних алгоритмів, призначених для заповнення пропущених значень в таблицях «об'єкт-властивість», є алгоритм ZET [11, 12]. В основу алгоритму входить три складові: гіпотеза надмірності, гіпотеза аналогічності та гіпотеза локальної компетентності.

Гіпотеза надмірності полягає в аналізі реальних таблиць, які мають надмірність у схожості між собою рядків (об'єктів) та стовпців (властивостей), що залежать один від одного. У разі відсутності надмірності перевага одного прогнозу іншому неможлива.

Гіпотеза аналогічності встановлює близькість об'єктів за n-ною властивістю у разі коли деяка пара об'єктів близька за значеннями (n -1).

Гіпотеза локальної компетентності стверджує, що надмірність має локальний характер, тобто. у кожного об'єкта є своя підмножина об'єктів-аналогів і кожна властивість має свою підмножину властивостей-аналогів. В прогнозування повинні брати участь тільки «компетентні» рядки та стовпці, які вибираються для кожного елемента, що прогнозується, окремо.

Умовно робота алгоритму складається із трьох кроків.

Крок 1. За вибраним пропущеним значенням з вихідної таблиці «об'єкт-властивість» вибирається підмножина «компетентних» рядків, а потім для вибраних рядків – підмножина «компетентних» стовпців.

Крок 2. Автоматичний вибір параметрів для передбачення мінімальної помилки.

Крок 3. Відбувається прогнозування.

Відновлення пропусків у таблицях «об’єкт-властивість» зручно реалізувати за допомогою штучної нейронної мережі. Для зручнішої роботи з даними, що надходять на обробку в реальному часі, пропонується використовувати адаптивні нейронні мережі. У даному випадку використовується нейронна мережа, утворена з *n*-паралельно працюючих найпростіших нейронів, що навчаються, а саме адаптивних лінійних елементів (ADALINE), які навчаються на основі модифікованої процедури Качмажа-Уідроу-Хоффа [14-17].

1.3 Адаптивний лінійний елемент Адаліна

Адаліна – це лінійний алгоритм класифікації та регресії, який може навчатися на основі вхідних даних та виправляти свої ваги відповідно до вихідних результатів (перевага цього алгоритму – можливість навчання на основі відхилень між передбаченими та реальними вихідними значеннями). Назва «Адаліна» виникла як скорочення від «адаптивний лінійний елемент» (Adaptive Linear Element). Адаліна є однією з реалізацій ідеї адаптивного навчання, в якій ваги моделі змінюються з кожним новим прикладом навчання.

Алгоритм роботи Адаліни:

Крок 1. Адаліна отримує вхідні дані та ваги для кожного входу.

Крок 2. Для кожного прикладу вхідних даних обчислюється ваговий сумар (загальна вага додатків і віднімань вхідних значень, взятих з відповідними вагами).

Крок 3. Результат обчислення подається на функцію активації (зазвичай це тангенс гіперболічний або лінійна функція, залежно від вибору).

Крок 4. Передбачене значення порівнюється з реальним результатом.

Крок 5. Ваги адаптуються за допомогою алгоритму градієнтного

спуску так, щоб зменшити різницю між передбаченим і реальним значенням.

Цей процес повторюється протягом багатьох ітерацій (епох), доки ваги не зійдуться або досягнуть певної зазначеної точності.

Одна з вагомих переваг Адалін – можливість навчати модель на основі великої кількості вхідних функцій та адаптуватися до змін в навчальних даних. На рисунку 1.2 представлена схема Адаліни.

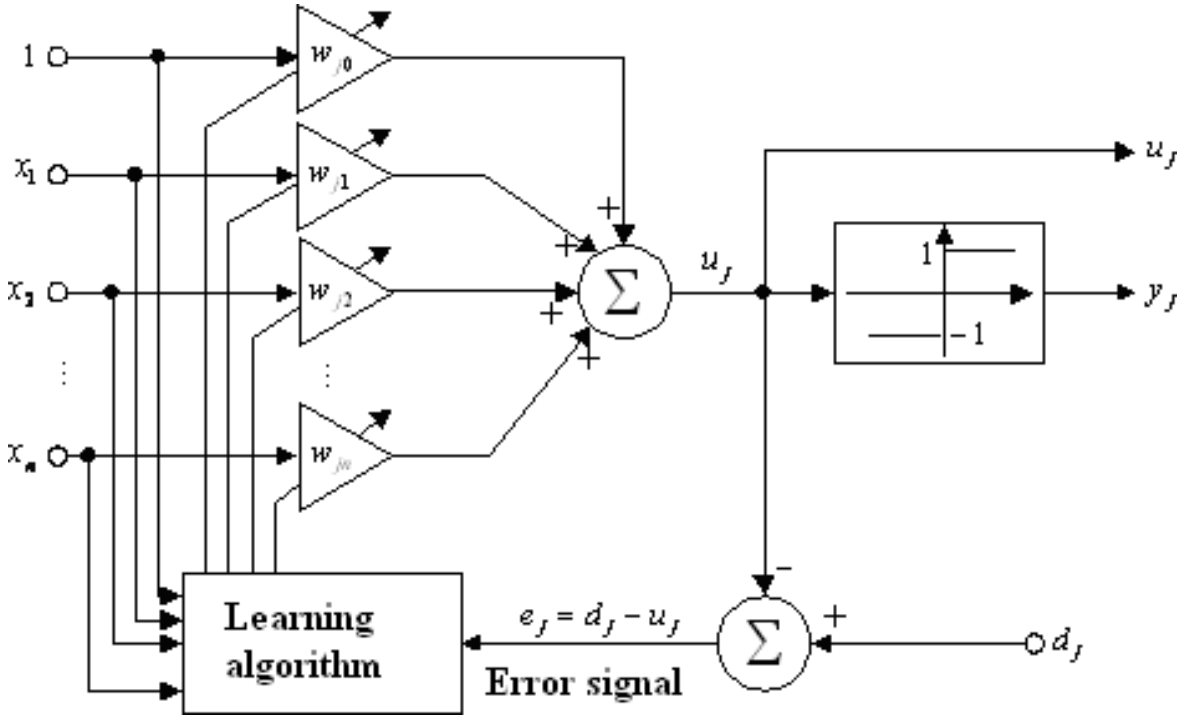

Рисунок 1.2 – Адаптивний лінійний елемент Адаліна

Адаліна може використовуватися як елементарний нейрон у складі штучних нейронних мережах, так і самостійно в задачах розпізнавання образів, обробки сигналів, реалізації логічних функцій. У цьому випадку Адаліни підходять для роботи з даними, які надходять на обробку в послідовному режимі.

1.4 Постановка задачі дослідження

Існує ціла низка підходів обробки та відновлення даних, спотворених пропусками, при цьому найбільш ефективними в даній ситуації виявилися підходи, засновані на математичному апараті обчислювального інтелекту, і насамперед штучних нейронних мережах. Разом з тим, описані підходи до відновлення пропусків працездатні лише у випадках, коли вихідна таблиця даних задана апріорно і кількість її рядків або стовпців не може змінюватися в процесі обробки. Існує безліч додатків, коли дані надходять на обробку послідовно в реальному часі, при цьому заздалегідь невідомо, який з векторів-образів, що надходять на обробку, містить пропуски.

Об’єктом дослідження є відновлення пропущених значень в тестовій вибірці.

Метою дослідження є аналіз методів відновлення втрачених спостережень та вибір найкращого метода для відновлення викривлених пропусками даних в тестових наборах.

Для досягнення мети необхідно вирішити такі завдання:

– аналіз та опис предметної області;

– постановка завдання;

– аналіз методів заповнення пропущених значень в наборах даних;

– підбір програмних засобів для реалізації та моделювання методу відновлення даних та програмна реалізація методу;

– порівняльний аналіз роботи обраного методу з відомими.

2 АДАПТИВНЕ ВІДНОВЛЕННЯ ДАНИХ З ПРОПУСКАМИ

2.1 Нейронна мережа

Нейронна мережа – це математична модель, яка імітує роботу нервової системи живих організмів. Це алгоритм, який може визначати складні залежності в даних, шукаючи патерни та взаємозв'язки. Нейронні мережі використовуються в різних сферах, таких як розпізнавання образів, прогнозування, класифікація, управління та багато інших.

Структура нейронної мережі включає в себе вузли, які називають нейронами, та зв'язки між ними. Кожен нейрон отримує вхідні сигнали, обчислює їх і видає вихідний сигнал. Це вхідні дані, які подаються на нейронну мережу, можуть бути числовими значеннями, текстом, зображеннями чи іншими формами даних [7, 12, 18]. На рисунку 2.1 продемонстрована класична нейронна мережа.

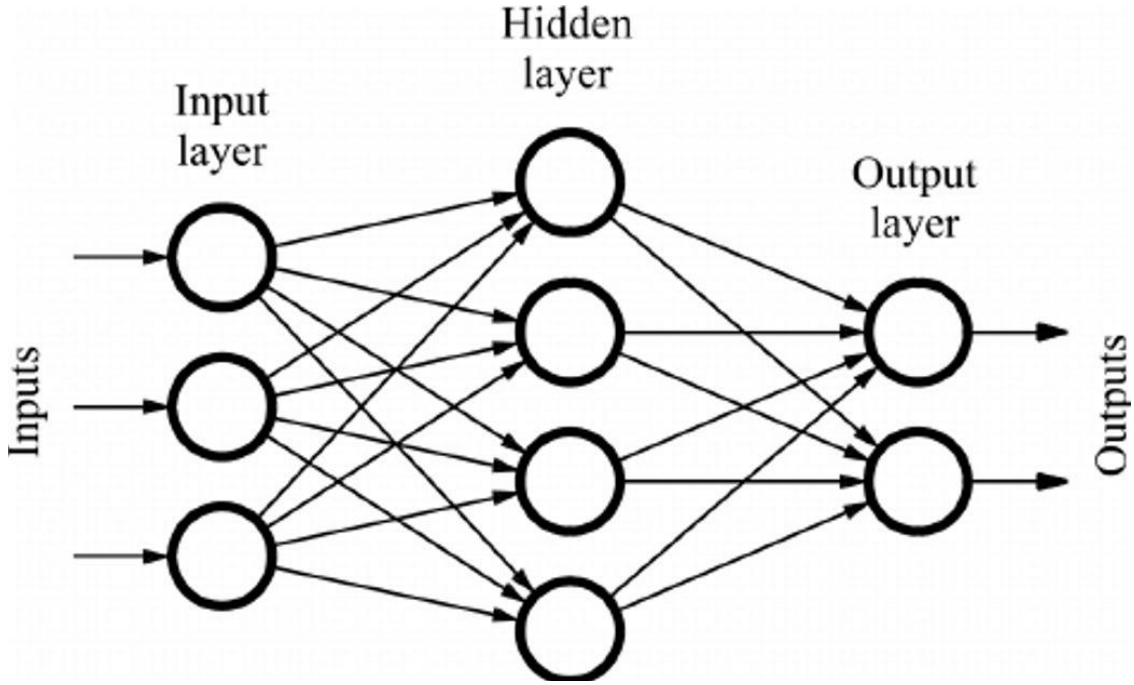

Рисунок 2.1 – Нейронна мережа

Нейронні мережі складаються з кількох шарів:

Шар 1. Вхідний шар в нейронній мережі – це перший шар нейронів, який отримує вхідні дані і передає їх для подальшої обробки мережею. Вхідні дані можуть бути числовими значеннями, текстом, зображеннями чи іншими формами даних, залежно від завдання, яке вирішується мережею.

Кожен нейрон в вхідному шарі представляє один атрибут або функцію вхідних даних. Наприклад, якщо мережа призначена для розпізнавання зображень, кожен нейрон в вхідному шарі може відповідати пікселю зображення.

Вхідні дані проходять через вхідний шар, і кожен нейрон в цьому шарі використовує свої ваги для обчислення значень вихідних сигналів. Ці вихідні сигнали потім передаються до прихованих шарів для подальшої обробки в нейронній мережі.

Шар 2. Приховані шари в нейронній мережі розташовані між вхідним і вихідним шарами. Кожен нейрон у прихованому шарі отримує вхідні сигнали з вхідного шару або з інших прихованих шарів, обчислює їх, і видає вихідний сигнал. Ці вихідні сигнали використовуються як вхід для нейронів у наступних шарах мережі.

Важливо відзначити, що мережі з більшою кількістю прихованих шарів і нейронів можуть вивчати складніші залежності в даних, але вони також потребують більше обчислювальних ресурсів та більше даних для навчання. Приховані шари допомагають нейронній мережі вивчати складні взаємозв’язки та робити більш точні прогнози чи класифікації.

У нейронних мережах існують різні архітектури з різною кількістю та розміщенням прихованих шарів. Наприклад, у звичайних нейронних мережах приховані шари розташовані між вхідним та вихідним шарами. У рекурентних нейронних мережах приховані шари можуть мати зв’язки, які повертаються до попередніх часових кроків у навчальних даних.

Шар 3. Вихідний шар нейронної мережі – це останній шар у структурі мережі, який видає результати обчислень або прогнози. Кожен нейрон у

вихідному шарі представляє собою конкретний вихід або класифікацію, залежно від типу задачі, яку розв’язує мережа.

У випадку задачі класифікації, кількість нейронів у вихідному шарі дорівнює кількості класів, які потрібно передбачити. Наприклад, якщо ми навчаємо нейронну мережу для класифікації зображень на три класи (наприклад, кіт, собака, кінь), то у вихідному шарі буде три нейрони, кожен з яких представляє один з цих класів.

У випадку задачі регресії, вихідний шар може мати один або кілька нейронів, які видають числові значення або вектори як відповіді на певний набір вхідних даних.

Вихідний шар обчислює значення за допомогою активаційної функції, яка може бути, наприклад, функцією *softmax* для задач класифікації або лінійною функцією для задач регресії. Ці значення можуть бути використані для прийняття рішення або подальшого аналізу в рамках конкретної задачі.

Кожен зв’язок між нейронами має вагу, яка визначає важливість цього зв’язку. Нейрони в мережі навчаються, оптимізуючи ці ваги під час тренування на навчальних даних.

Процес тренування нейронної мережі полягає в адаптації ваг зв’язків між нейронами так, щоб модель точно передбачала вихідні дані для вхідних даних. Після завершення тренування нейронна мережа може використовуватися для прогнозування або класифікації нових даних, з якими вона раніше не зустрічалася.

2.2 Ансамблі нейронних мереж

Ансамблі нейронних мереж (англ. Neural Network Ensembles) – це метод машинного навчання, який об’єднує кілька нейронних мереж для досягнення кращої точності та стабільності прогнозів порівняно з окремими моделями нейронних мереж, у яких одні й самі дані обробляються відразу

декількома *h*-паралельно працюючими штучними нейронними мережами [19-23].

Вихідні сигнали деяким чином комбінуються в загальну оцінку, яка дає уявлення про якість отриманих результатів за допомогою локальних мереж, що входять в ансамблі, як це показано на рисунку 2.2.

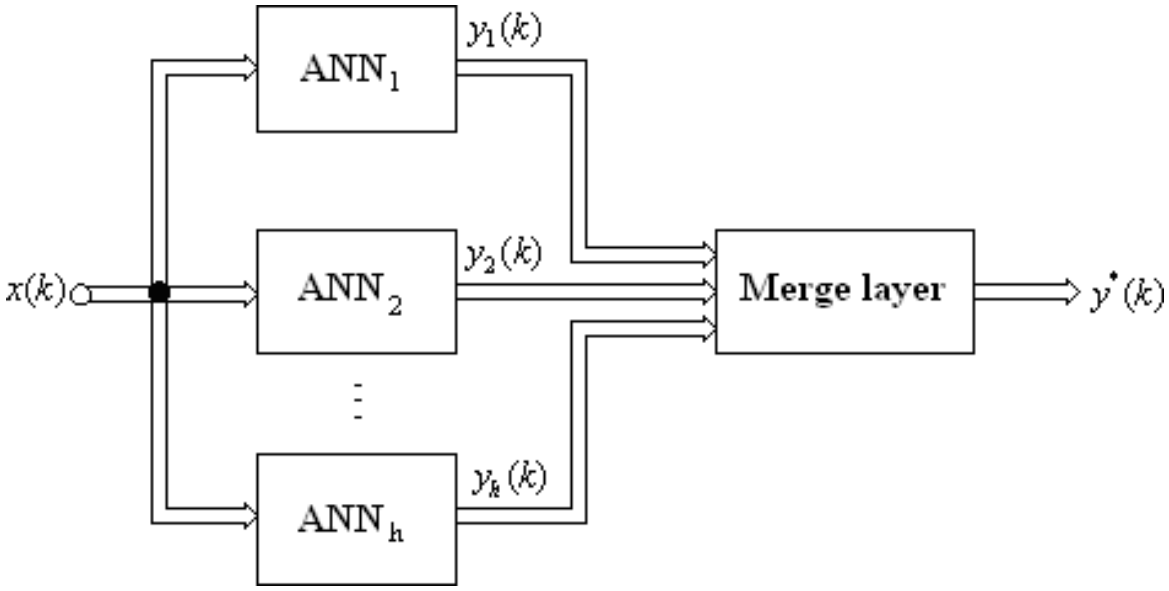

Рисунок 2.2 – Ансамбль *h* – паралельно працюючими штучними нейронними мережами

Найбільшого поширення до об’єднання штучних нейронних мереж в ансамблі набули два підходи, такі як модельний та заснований на зваженому усередненні. Дані підходи досить відрізняються один від одного, проте вони так само мають і схожість. Вони обидва використовують лінійну комбінацію вихідних сигналів своїх членів у тій чи іншій формі.

Модульний підхід має досить евристичний характер на відміну більш математично суворого зваженого усереднення, хоча й тут залишається елемент суб’єктивізму, пов’язаний з вибором членів ансамблю.

Ця задача зазвичай вирішується за допомогою тих чи інших евристик, хоча є більш-менш суворі результати, засновані на генетичному програмуванні або поступовому нарощуванні складності мереж-членів

ансамблю.

Ансамбль може включати в себе різні типи нейронних мереж, такі як глибокі нейронні мережі, згорткові нейронні мережі, рекурентні нейронні мережі тощо. Це дозволяє ансамблю використовувати різні аспекти даних та відмінності між моделями для покращення прогнозів.

Ансамблі зменшують ймовірність перенавчання, особливо, якщо вони використовуються з алгоритмами, які мають різні умови зупинки навчання, допомагають покращити стійкість моделі до випадкових змін в даних або до викидів, можуть забезпечити кращу узагальненість для нових, невідомих даних, оскільки вони комбінують декілька джерел інформації, можуть використовувати різні підходи, такі як голосування більшості (Majority Voting), середнє значення (Averaging), або зважене голосування (Weighted Voting), що робить їх гнучкими та адаптивними до різних ситуацій, може виявити вищий інтелектуальний потенціал, оскільки комбінує думки різних моделей.

Ансамблі нейронних мереж поділяються за типами.

Багатокласовий ансамбль (Multi-Class Ensemble) – це тип ансамблю нейронних мереж, який використовується для задачі класифікації, де існує багато класів, і кожен елемент може належати лише до одного з них. Такі задачі часто зустрічаються у сферах, де потрібно класифікувати об’єкти на декілька категорій. Він включає в себе кілька нейронних мереж, кожна з яких спробує вирішити задачу класифікації для всіх класів. Коли новий об’єкт потрапляє до ансамблю для класифікації, кожна модель у ансамблі видає свій прогноз. Клас, який отримав більше голосів, вважається прогнозом ансамблю. Використання різних моделей сприяє зменшенню ризику перенавчання, оскільки різні моделі можуть виявити різні підходи до навчання. Моделі можуть взаємодіяти між собою, підсилюючи сильні сторони один одного і компенсуючи слабкі сторони.

Ансамбль може бути більш стійким до викидів або неточностей в навчальних даних, оскільки різні моделі можуть реагувати по-різному на такі

випадки. Завдяки використанню декількох моделей ансамбль може забезпечувати кращу точність класифікації і загальне краще узагальнення для нових даних.

Багатокласові ансамблі є потужним інструментом для класифікації об'єктів на декілька класів і використовуються в різних сферах, включаючи комп'ютерне зорове сприйняття, медичні діагностики, фінансовий аналіз та багато інших;

Багатозадачний ансамбль (Multi-Task Ensemble) – це тип ансамблю нейронних мереж, який використовується для вирішення декількох схожих задач одночасно. У цьому випадку кожна модель у складі ансамблю навчається вирішувати різні, але пов’язані між собою задачі. Наприклад, в області комп’ютерного зору одна модель може відповідати за розпізнавання обличь, інша – за розпізнавання об’єктів, третя – за визначення позиції об’єктів. Кожна модель в ансамблі відповідає за вирішення певної задачі або кількох пов’язаних задач. Моделі навчаються одночасно, інформація, яку вони отримують в процесі навчання, може використовуватися для покращення результатів всіх задач. Вони можуть взаємодіяти між собою, обмінюючись інформацією та взаємодіючі в процесі прийняття рішень.

Ансамбль використовує загальну інформацію, яку навчилися розпізнавати всі моделі, для покращення результатів кожної конкретної задачі. Багатозадачні ансамблі ефективні для вирішення задач, які мають спільні аспекти або вимагають схожих навичок. Взаємодія моделей може компенсувати недоліки кожної окремої моделі, покращуючи загальні результати.

Багатозадачні ансамблі застосовуються у різних областях, де потрібно вирішувати декілька пов’язаних задач одночасно, таких як медична діагностика, машинне навчання, обробка природних мов та багато інших;

Багаторівневий ансамбль (Multi-Level Ensemble) – це тип ансамблю нейронних мереж, де використовуються кілька рівнів моделей для вирішення складних задач. Кожен рівень ансамблю може спеціалізуватися на різних

аспектах задачі, інтегруючи свої рішення на більш високому рівні для отримання кінцевого результату. Моделі розташовані на кількох рівнях, де кожен рівень може мати свою специфічну задачу або фокусуватися на певних аспектах вхідних даних. Рішення, отримані на нижчих рівнях, можуть використовуватися як вхідні дані для вищих рівнів, де вони агрегуються та композуються для отримання кінцевого результату. Різні рівні можуть спеціалізуватися на різних аспектах задачі або розпізнавати різні підзадачі, що допомагає вирішувати більш складні проблеми. Рішення, отримані з кожного рівня, інтегруються для отримання більш точного та надійного результату на вищому рівні. Система може бути адаптивною, здатною змінювати організацію рівнів або додавати нові рівні в залежності від складності задачі або якості отриманих результатів.

Багаторівневі ансамблі використовуються у завданнях, де складність задачі вимагає аналізу на різних рівнях деталізації, інтеграції різних аспектів або спеціалізованих даних для досягнення найкращого результату. Вони можуть бути корисні у великих системах машинного навчання, обробці великих обсягів даних та в інших задачах, де потрібна глибока аналітика та інтеграція різних джерел інформації;

Байєсівський ансамбль (Bayesian Ensemble) – це тип ансамблю нейронних мереж, який базується на байєсівській імовірності та теорії вибірки. У цьому типі ансамблю кожна модель навчається з використанням вибірки з можливими вагами для кожного зразка. Після навчання ці ваги можуть бути використані для оцінки надійності кожної моделі. Використання байєсівської імовірності для моделювання невизначеності та ваг моделей. Кожному зразку присвоюється вага, яка визначає його вплив на навчання моделі. Ваги зразків інтегруються в модель для урахування надійності та важливості кожної моделі [24-28].

Ансамбль може надавати оцінку надійності кожної моделі, дозволяючи враховувати невизначеність у рішенні. Можливість адаптації ваг зразків залежно від нової інформації або зміни у вхідних даних [29-33].

Байєсівські ансамблі дозволяють урахувати невизначеність та надати оцінку надійності прогнозів, особливо в умовах обмежених даних або у випадках, коли деякі моделі можуть бути менш надійними або нестійкими до змін в навколишньому середовищі.

Вибір конкретного типу ансамблю залежить від конкретної задачі та характеристик даних. Ансамблі нейронних мереж є потужним засобом у сучасних застосуваннях машинного навчання.

2.3 Адаптивне відновлення даних із пропусками

Адаптивне відновлення даних із пропусками – це процес заповнення відсутніх (пропущених) значень у наборі даних з використанням методів аналізу даних і статистики [15, 19, 23, 29]. Цей процес важливий в аналізі даних, оскільки багато алгоритмів машинного навчання та статистичних методів вимагають повного набору даних для ефективної роботи.

Нехай вихідною інформацією є $N \times n$ таблиця «об'єкт-властивість», яка містить інформацію про N об'єктах, кожен з яких описується $(1 \times n)$ – вектором-рядком ознак $x_i = (x_{i1}, ..., x_{ip}, ..., x_{ij}, ..., x_{in})$ при цьому передбачається, що N_G рядків можуть мати по одному пропуску, а $N_F = N - N_G$ заповнені повністю.

В процесі обробки таблиці необхідно заповнити пропуски так, щоб відновлені елементи були б у певному сенсі «найбільш правдоподібними» або «близькими» до апріорі невідомим закономірностям, що містяться в таблиці.

Представимо таблицю 2.1 як $(N \times n)$ – матрицю X, в якій відсутній один елемент x_{kj} або у більш загальному випадку відсутні N_G елементів.

Таблиця 2.1 – Таблиця «об’єкт-властивість»

	1	...	*p*	...	*j*	...	*n*
1	x_{11}	...	x_{1p}	...	x_{1j}	...	x_{1n}
...	...	...	...	...	...	...	...
i	x_{i1}	...	x_{ip}	...	x_{ij}	...	x_{in}
...	...	...	...	...	...	...	...
k	x_{k1}	...	x_{kp}	...	x_{kj}	...	x_{kn}
...	...	...	...	...	...	...	...
N	x_{N1}	...	x_{Np}	...	x_{Nj}	...	x_{Nn}

Передбачається [2-4], що між стовпцями $x_j = (x_{1j},...,x_{ij},...,x_{kj},...,x_{Nj})^T$ існує лінійна кореляція, на підставі якої і проводиться відновлення пропуску за допомогою регресії

$$\hat{x}_{kj} = w_{j0} + w_{j1}x_{k1} + w_{j2}x_{k2} + w_{j,j-1}x_{k,j-1} + w_{j,j+1}x_{k,j+1} + ... + w_{jn}x_{kn} \quad , \qquad (2.1)$$

або

$$\hat{x}_{kj} = \underset{\sim}{X}_{kj} w_j , \qquad (2.2)$$

де $w_j = (w_{j0}, w_{j1},..., w_{jn})^T$ – $(n \times 1)$ вектор параметрів, що підлягають визначенню;

$\underset{\sim}{X}_{kj} = (1, x_{k1},..., x_{k,j-1}, x_{k,j+1},..., x_{kn})$ вектор-рядок ознак k-того об’єкта без kj - того елемента та з одиницею на першій позиції.

Вектор невідомих параметрів w_j може бути знайдений за допомогою стандартного методу найменших квадратів, для чого із матриці X слід виключити k-ий рядок, j-й стовпець, додати зліва стовпець з одиниць та на

основі отриманої $((N-1)\times n)$ матриці X_j розрахувати оцінки параметрів

$$w_j = (X_j^T \mathrm{X}_j)^+ \mathrm{X}_j^T \mathrm{X}_j, \tag{2.3}$$

де $\mathrm{X}_j = (x_{1j},...,x_{ij},...,x_{k-1,j},x_{k+1,j},...,x_{Nj})^T$.

$(\bullet)^+$ – символ псевдообернення по Муру-Пенроузу.

Якщо ж пропуски є в N_G строках і в різних стовпцях, із матриці X виключаються всі ці рядки і на підставі отриманої усіченої $(N_F \times n)$ матриці n раз знаходяться вектори параметрів (2.1) для всіх $j = 1,2,...,n$.

Далі за допомогою рівняння (2.1) або (2.3) заповнюються всі прогалини отриманими оцінками $\hat{x}_{kj}$.

Розглянутий метод зручно представити у вигляді блок-схеми, наведеної на рисунку 2.3.

Цей метод нескладно поширити на випадок, коли дані про об’єкти в таблицю 2.1 надходять послідовно об’єкт за об’єктом.

Нехай без втрати спільності від початку задано інформацію у вигляді таблиці 2.1 з N рядками, з яких N_F заповнені повністю.

На підставі цих даних може бути побудована оцінка

$$w_j(N_F) = (X_j^T(N_F) X_j(N_F))^+ X_j^T(N_F) \mathrm{X}_j(N_F), \tag{2.4}$$

за допомогою якої відновлюється вихідна таблиця. З появою $(N+1)$-го спостереження у формі повністю заповненого рядка x_{N+1} оцінка (2.4) може бути скоригована за допомогою рекурентного методу найменших квадратів

$$\begin{cases} w_j(N_F+1) = w_j(N_F) + \dfrac{P_j(N_F)(x_{N+1,j} - \vec{X}_{N+1,j} w_j(N_F))}{1 + \vec{X}_{N+1,j} P_j(N_F) \vec{X}^T_{N+1,j}} \vec{X}^T_{N+1,j}, \\ P_j(N_F+1) = P_j(N_F) - \dfrac{P_j(N_F) \vec{X}^T_{N+1,j} \vec{X}_{N+1,j} P_j(N_F)}{1 + \vec{X}_{N+1,j} P_j(N_F) \vec{X}^T_{N+1,j}}, \end{cases} \quad (2.5)$$

після цього уточняються і відновлені значення $\hat{x}_{kj}$.

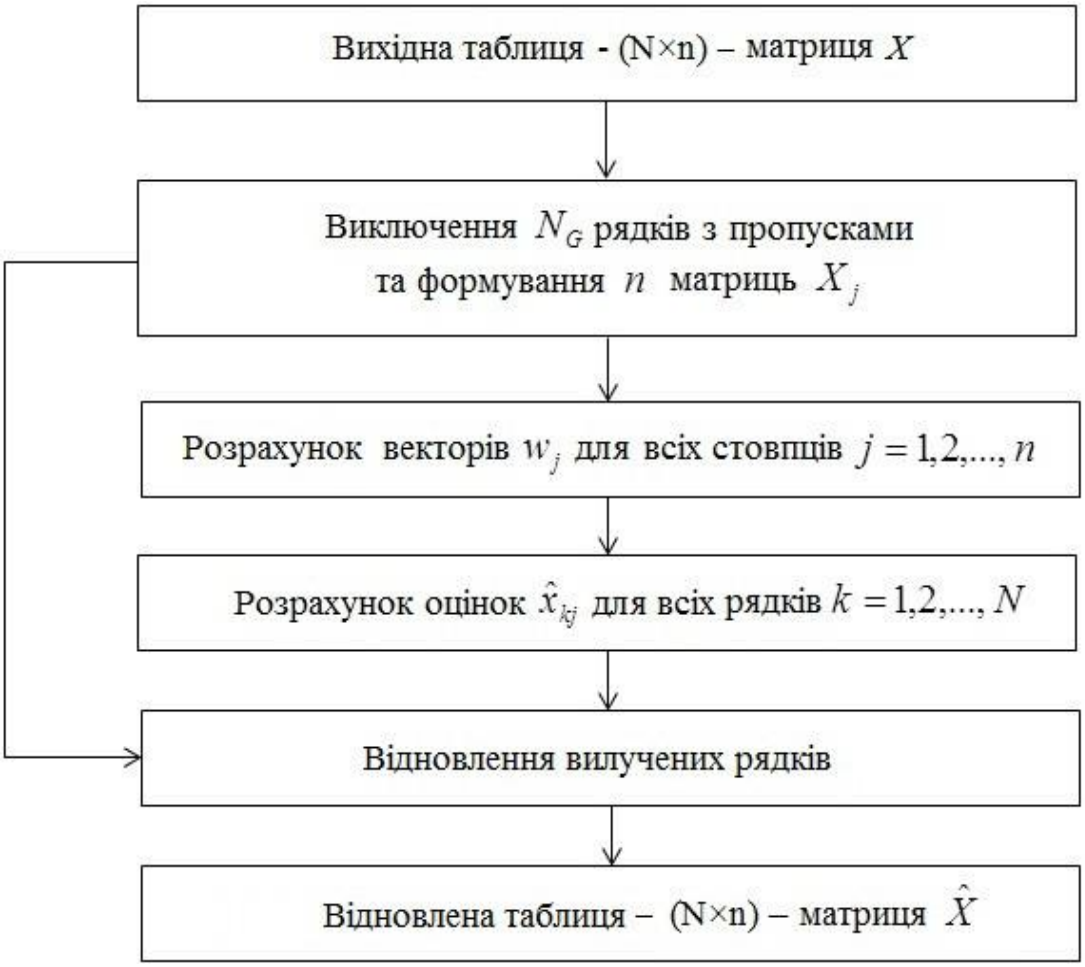

Рисунок 2.3 – Пакетний алгоритм заповнення пропусків

Якщо ж $(N+1)$ рядок містить пробіл, він пропускається і алгоритм (2.5) очікує появи повністю заповненого рядка, наприклад x_{N+2}, після чого обчислюється $w_j(N_F+1)$ за допомогою $(N+2)$-го спостереження та коригуються всі значення $\hat{x}_{kj}$, включаючи $\hat{x}_{N+1,j}$.

На початкових етапах обробки таблиці 2.1, коли число повністю заповнених рядків N_F порівняно з кількістю стовпців n, оцінки, отримані з допомогою методу найменших квадратів, характеризуються низькою точністю. У цій ситуації для послідовної обробки ефективніше застосування адаптивних алгоритмів навчання, що володіють як фільтруючими, так і слідкуючими (для нестаціонарних ситуацій) властивостями [12]. Розглянутий метод зручно представити у вигляді на рисунку 2.4.

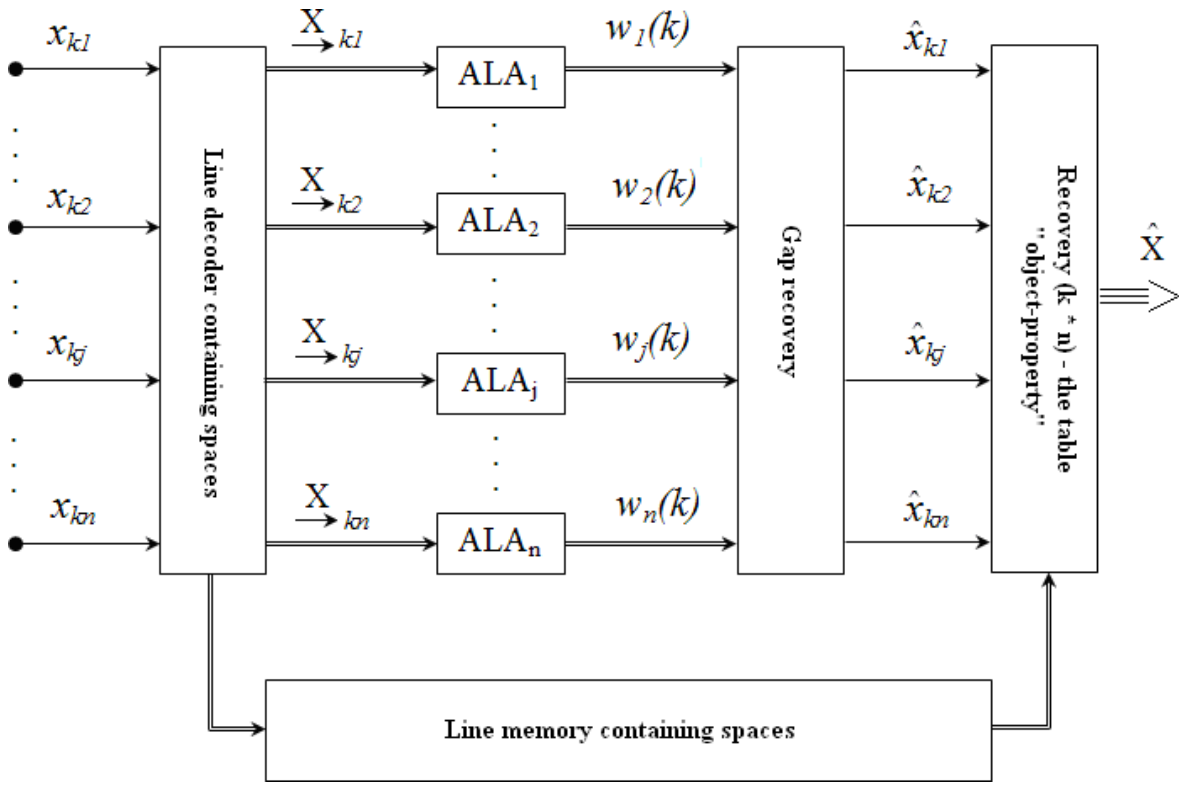

Рисунок 2.4 – Адаптивна нейромережева система для відновлення пропусків

Даний метод можна переписати із застосуванням адаптивних алгоритмів навчання у вигляді

$$\begin{cases} w_j(N_F+1) = w_j(N_F) + r_j^{-1}(N_F+1)(x_{N+1,j} - \mathrm{X}_{N+1,j}^T w_j(N_F))\mathrm{X}_{N+1,j}, \\ r_j(N_F+1) = \alpha r_j(N_F) + \|\mathrm{X}_{N+1,j}\|^2, \end{cases} \quad (2.6)$$

де $0 \le \alpha \le 1$ – параметр згладжування, що задає компроміс між згладжуванням і стеженням за характеристиками об’єктів, що змінюються.

Обробку інформації в режимі послідовного надходження даних зручно організувати на основі нейромережевої системи, основними елементами якої є n -паралельно працюючих адаптивних лінійних асоціаторів (ALA-Adaline) [14-16], які навчаються за допомогою формули (2.6). На рисунку 2.4 наведено схему цієї системи, яка не вимагає додаткових пояснень.

3 АПРОБАЦІЯ МЕТОДІВ ЗАПОВНЕННЯ ПРОПУЩЕНИХ ЗНАЧЕНЬ

Для роботи з даними специфічної властивості, а саме медичних даних, необхідно їх ретельно проаналізувати і зрозуміти залежність спостережень одне від одного.

Для використання підходів заповнення пропущених значень середнім значенням по ознаці серед пацієнтів з однаковим діагнозом або методу EM-оцінювання слід бути впевненим, що вибірка даних має нормальний або близький до нього закон розподілу. Якщо дані мають нормальний або близький до нього закон розподілу, використання цих методів може бути доцільним, оскільки вони передбачають розподіл значень, що дозволяє ефективно заповнювати пропущені дані.

Проте, якщо дійсної вибірка медичних даних має відмінний від нормального розподіл, то використання таких методів може призвести до неточних результатів. Медичні дані зазвичай можуть мати важкі хвости, асиметричність або інші відхилення від нормального розподілу. Це може бути викликано різними факторами, такими як клінічні особливості пацієнтів чи спосіб збору даних.

Щоб підтвердити, чи відповідає вибірка медичних даних нормальному розподілу, можна використовувати статистичні методи, такі як тест Шапіро-Уілка, тест Колмогорова-Смірнова чи графіки квантиль-квантиль (Q-Q plots). Ці методи допоможуть оцінити відповідність розподілу даних нормальному закону.

Якщо дані не відповідають нормальному розподілу, доцільно буде використовувати інші методи для заповнення пропущених значень, такі як методи, що базуються на розподілі квартилів, медіані або інших статистиках, які менше чутливі до відхилень від нормального розподілу.

Для дослідження закону розподілу найбільш відомих вибірок медичних даних з репозиторію UCI [34-37], які є вбудованими у бібліотеку sklearn.datasets у Python.

3.1 Графік квантіль-квантіль

Графік квантіль-квантіль (*Q-Q* plot або *QQ* plot) – це популярний графік для перевірки того, чи відповідають реальні дані теоретичному розподілу (зазвичай нормальному). Цей графік порівнює квантилі вибірки (спостережені квантилі) з квантилями теоретичного розподілу (очікувані квантилі) і візуально відображає розподіл даних.

Якщо дані точно відповідають нормальному розподілу, точки на графіку *Q-Q* лягають на пряму лінію, яка є діагоналлю графіку. Якщо точки розташовані приблизно вздовж цієї діагоналі, це може свідчити про те, що розподіл даних є близьким до нормального.

Однак, якщо точки графіка відхиляються від діагоналі, це може вказувати на відхилення від нормального розподілу. Наприклад, якщо точки утворюють «вигин» або «згин», це може вказувати на наявність важкого хвоста в розподілі, тобто значення, які відхиляються від середнього більше, ніж у нормальному розподілі.

Графік *Q-Q* є потужним інструментом для перевірки розподілу даних і дозволяє зрозуміти, наскільки дані відповідають теоретичному розподілу, що може бути корисним при виборі відповідних методів аналізу даних.

На графіку квантіль-квантіль (Q-Q plot) ідеальний збіг для розподілу буде показано лінією точок під кутом 45 градусів від нижнього лівого кута графіка до правого верхнього кута. Ця діагональна лінія є лінією, що відображає ідеальний збіг між спостережуваними квантилями вибірки і квантилями теоретичного розподілу (зазвичай нормального).

Якщо точки графіка розташовані вздовж цієї діагоналі, це означає

ідеальний збіг між вибіркою і нормальним розподілом. Якщо точки відхиляються від цієї діагоналі, це показує відхилення від очікуваного нормального розподілу. Такі відхилення можуть вказувати на наявність важкого хвоста, асиметрії або інших ненормальних властивостей у розподілі даних. Графік Q-Q є важливим інструментом для визначення цих відхилень та перевірки відповідності розподілу даних теоретичному розподілу.

У Python для побудови графіку *Q-Q* використовується функція qqplot() з бібліотеки statsmodels. Ця функція призначена для порівняння вибірки даних з гаусовим (нормальним) розподілом за допомогою графіку *Q-Q*.

Для використання функції qqplot(), спочатку потрібно імпортувати необхідні бібліотеки (рисунок 3.1).

```
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
```

Рисунок 3.1 – Частина програмного коду з файлу Diplom.ipynb імпортування необхідних бібліотек

можемо використовувати функцію qqplot() для побудови графіку *Q-Q*.

```
import numpy as np
import pandas as pd

from sclearn.datasets
import load_diabetes,load_breast_cancer, load_linnerud
line=load_linnerud()
breast=load_breast_cancer()
diabetes=load_diabetes()
```

Рисунок 3.2 – Частина програмного коду з файлу Diplom.ipynb для завантаження медичних даних

На прикладі diabetes dataset можна розглянути які дані містяться у датасеті. diabetes.data містить дані усіх ознак (рисунок 3.3), diabetes.target містить інформацію про цільову ознаку (рисунок 3.4), diabetes.DESCR містить опис ознак (рисунок 3.5), diabetes.feature_names містить перелік ознак.

```
array([[ 0.03807591,  0.05068012,  0.06169621, ..., -0.00259226,
         0.01990842, -0.01764613],
       [-0.00188202, -0.04464164, -0.05147406, ..., -0.03949338,
        -0.06832974, -0.09220405],
       [ 0.08529891,  0.05068012,  0.04445121, ..., -0.00259226,
         0.00286377, -0.02593034],
       ...,
       [ 0.04170844,  0.05068012, -0.01590626, ..., -0.01107952,
        -0.04687948,  0.01549073],
       [-0.04547248, -0.04464164,  0.03906215, ...,  0.02655962,
         0.04452837, -0.02593034],
       [-0.04547248, -0.04464164, -0.0730303 , ..., -0.03949338,
        -0.00421986,  0.00306441]])
```

Рисунок 3.3 – Частина вибірки медичних даних diabetes.data

```
array([151.,  75., 141., 206., 135.,  97., 138.,  63., 110., 310., 101.,
        69., 179., 185., 118., 171., 166., 144.,  97., 168.,  68.,  49.,
        68., 245., 184., 202., 137.,  85., 131., 283., 129.,  59., 341.,
        87.,  65., 102., 265., 276., 252.,  90., 100.,  55.,  61.,  92.,
       259.,  53., 190., 142.,  75., 142., 155., 225.,  59., 104., 182.,
       128.,  52.,  37., 170., 170.,  61., 144.,  52., 128.,  71., 163.,
       150.,  97., 160., 178.,  48., 270., 202., 111.,  85.,  42., 170.,
       200., 252., 113., 143.,  51.,  52., 210.,  65., 141.,  55., 134.,
        42., 111.,  98., 164.,  48.,  96.,  90., 162., 150., 279.,  92.,
        83., 128., 102., 302., 198.,  95.,  53., 134., 144., 232.,  81.,
       104.,  59., 246., 297., 258., 229., 275., 281., 179., 200., 200.,
       173., 180.,  84., 121., 161.,  99., 109., 115., 268., 274., 158.,
       107.,  83., 103., 272.,  85., 280., 336., 281., 118., 317., 235.,
        60., 174., 259., 178., 128.,  96., 126., 288.,  88., 292.,  71.,
       197., 186.,  25.,  84.,  96., 195.,  53., 217., 172., 131., 214.,
        59.,  70., 220., 268., 152.,  47.,  74., 295., 101., 151., 127.,
       237., 225.,  81., 151., 107.,  64., 138., 185., 265., 101., 137.,
       143., 141.,  79., 292., 178.,  91., 116.,  86., 122.,  72., 129.,
       142.,  90., 158.,  39., 196., 222., 277.,  99., 196., 202., 155.,
        77., 191.,  70.,  73.,  49.,  65., 263., 248., 296., 214., 185.,
        78.,  93., 252., 150.,  77., 208.,  77., 108., 160.,  53., 220.,
       154., 259.,  90., 246., 124.,  67.,  72., 257., 262., 275., 177.,
        71.,  47., 187., 125.,  78.,  51., 258., 215., 303., 243.,  91.,
       150., 310., 153., 346.,  63.,  89.,  50.,  39., 103., 308., 116.,
       145.,  74.,  45., 115., 264.,  87., 202., 127., 182., 241.,  66.,
        94., 283.,  64., 102., 200., 265.,  94., 230., 181., 156., 233.,
        60., 219.,  80.,  68., 332., 248.,  84., 200.,  55.,  85.,  89.,
        31., 129.,  83.])
```

Рисунок 3.4 – Частина вибірки медичних даних diabetes.target

```
'.. _diabetes_dataset:\n\nDiabetes dataset\n----------------\n\nTen baseline variables, age, sex, body ma
ss index, average blood\npressure, and six blood serum measurements were obtained for each of n =\n442 di
abetes patients, as well as the response of interest, a\nquantitative measure of disease progression one
year after baseline.\n\n**Data Set Characteristics:**\n\n  :Number of Instances: 442\n\n  :Number of Attr
ibutes: First 10 columns are numeric predictive values\n\n  :Target: Column 11 is a quantitative measure
of disease progression one year after baseline\n\n  :Attribute Information:\n      - Age\n      - Sex\n
- Body mass index\n      - Average blood pressure\n      - S1\n      - S2\n      - S3\n      - S4\n
- S5\n      - S6\n\nNote: Each of these 10 feature variables have been mean centered and scaled by the st
andard deviation times `n_samples` (i.e. the sum of squares of each column totals 1).\n\nSource URL:\nhtt
ps://www4.stat.ncsu.edu/~boos/var.select/diabetes.html\n\nFor more information see:\nBradley Efron, Trevo
r Hastie, Iain Johnstone and Robert Tibshirani (2004) "Least Angle Regression," Annals of Statistics (wit
h discussion), 407-499.\n(https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)'
```

Рисунок 3.5 – Опис ознак вибірки медичних даних diabetes.DESCR

Далі, побудуємо графік *QQ* з використанням функції qqplot() з statsmodels, розглянутий на рисунках (3.6-3.9).

```
from statsmodels.graphics.gofplots
import qqplot
from matplotlib
import pyplot
data=np.array(diabetes.data)
qqplot(data, line='s')
pyplot.show()
```

Рисунок 3.6 – Частина програмного коду з файлу Diplom.ipynb для побудування графіку QQ

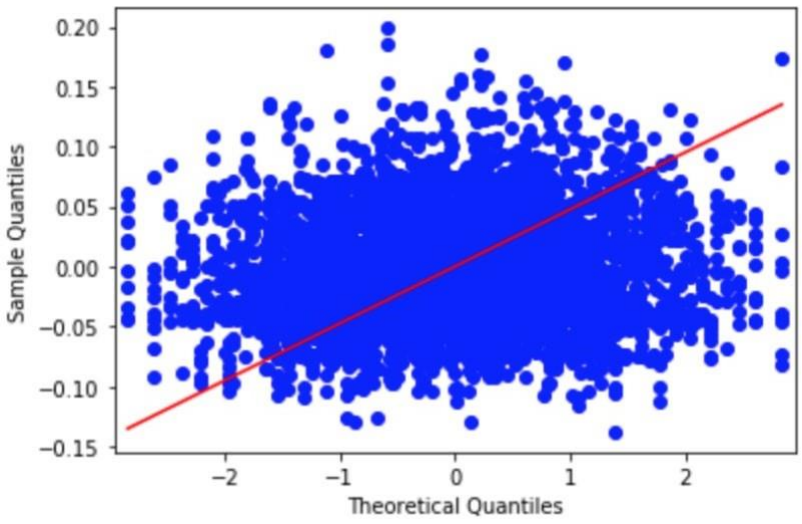

Рисунок 3.7 – Графік *QQ* для diabetes dataset

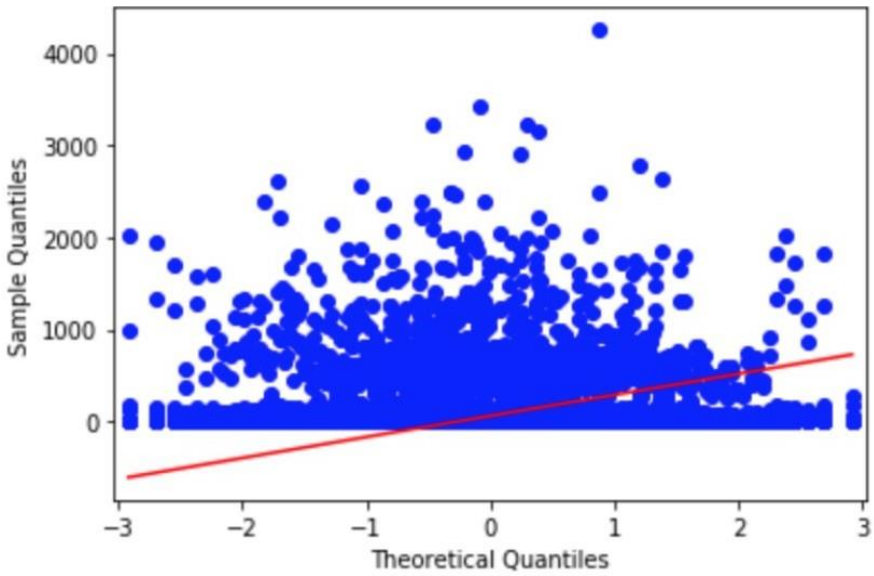

Рисунок 3.8 – Графік *QQ* для breast_cancer dataset

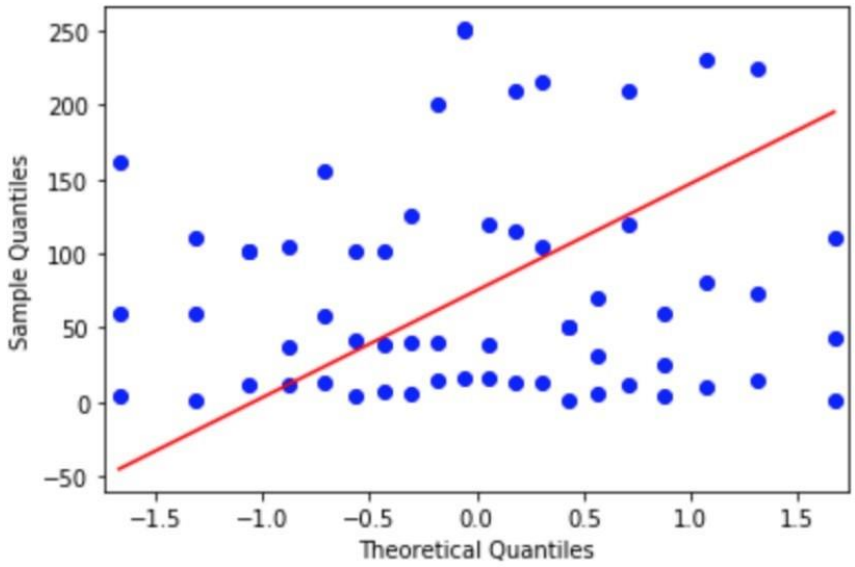

Рисунок 3.9 – Графік *QQ* для linnerud dataset

Візуальна оцінка за допомогою графіка Q-Q може надати вам загальне уявлення про те, наскільки вибірка подібна до нормального розподілу. Однак для чисельної оцінки неподільної придатності даних до нормального розподілу використовують різні статистичні методи [38, 39]. Один із найпоширеніших методів – це тест на нормальність Шапіро-Уілка.

3.2 Тест на нормальність Шапіро-Уілка

Тест на нормальність Шапіро-Уілка – це статистичний тест, який використовується для перевірки гіпотези про нормальний розподіл даних у вибірці. Нульова гіпотеза в цьому тесті відзначає, що дані взяті з певного розподілу (наприклад, нормального розподілу).

Основна ідея тесту полягає в порівнянні спостережених даних із тим, що можна очікувати від нормального розподілу. Якщо відмінності між спостереженими даними та очікуваними значеннями великі, тоді ми можемо відхилити нульову гіпотезу та висунути припущення, що дані не мають нормального розподілу.

У Python можна використовувати тест на нормальність Шапіро-Уілка за допомогою бібліотеки scipy.stats. Функція повертає *W*-статистику, розраховану тестом, і значення *p*.

Тест на нормальність Шапіро-Уілка повертає значення тестової статистики (stat) та *p*-значення (*p*).

Інтерпретуючи результати тесту, якщо *p*-значення менше встановленого рівня значущості (зазвичай 0.05), то можна відхилити нульову гіпотезу та припустити, що дані не мають нормального розподілу. В іншому випадку можна не відхиляти нульову гіпотезу та припустити, що дані мають нормальний розподіл.

W-статистика обчислюється як сума рангів однієї з вибірок після сортування всіх значень обох вибірок разом. *W*-статистика може бути використана для порівняння розподілу значень у двох групах і визначення, чи існують статистично значущі відмінності між ними.

У Манн-Уїтнівському тесті нульова гіпотеза припускає, що розподіли двох вибірок однакові.

Альтернативна гіпотеза вказує на те, що розподіли різняться. *W*-статистика служить основою для обчислення *p*-значення, яке допомагає приймати рішення про відхилення чи не відхилення нульової гіпотези.

Застосуємо тест Шапіро-Уілка для кожної з вибірок даних (рисунок 3.10-3.13) [36-39]

```
from scipy.stats
import shapiro
stat,p = shapiro(data)
print('staticstics=%.3f, p=%3f'% (stat,p))
alpha=0.05
if p>alhpa:
    print('Sample looks Gaussian(fail to reject H0)')
    else:
      print('Sample does not looks Gaussian (fail to reject H0)')
```

Рисунок 3.10 – Частина програмного коду з файлу Diplom.ipynb для розрахунку тест Шапіро-Уілка

```
Statistics=0.986, p=0.000
Sample does not look Gaussian (reject H0)
```

Рисунок 3.11 – Результат розрахунку тесту Шапіро-Уілка для diabetes.data

```
Statistics=0.295, p=0.000
Sample does not look Gaussian (reject H0)
```

Рисунок 3.12 – Результат розрахунку тесту Шапіро-Уілка для breast_cancer.data

```
Statistics=0.854, p=0.000
Sample does not look Gaussian (reject H0)
```

Рисунок 3.13 – Результат розрахунку тесту Шапіро-Уілка для linnerud.data

3.3 Експериментальні дослідження на реальних медичних даних

Для перевірки роботи розглянутих методів заповнення пропущених даних пропонуємо використати вибірку реальних даних захворювань відділення пульмонології. Детальний опис клінічної вибірки містить інформацію про 132 пацієнтів із захворюваннями пульмонології. Ця вибірка містить 104 ознаки для кожного пацієнта, які включають в себе інформацію про стать, вік, скарги, анамнез, об'єктивний опис стану пацієнта, клінічний аналіз крові, біохімічний аналіз крові, клінічний аналіз сечі, рентгенографію грудної клітки, ЕКГ і спірометричні дослідження.

Вибірка пацієнтів поділена на три категорії діагнозів: хронічне обструктивне захворювання легень (ХОЗЛ) з 46 пацієнтами, бронхіальна астма з 53 пацієнтами та пневмонія з 33 пацієнтами.

Усі дані медичних досліджень представлено у вигляді файлу формату Dataset.xlsx. Код завантаження даних з файлу продемонстровано на рисунку 3.14.

```
import numpy
import pandas
pandas.set_option('display.max_rows',500)
pandas.set_option(display.max_columns',500)
pandas.set_option(display.width',500)
file = pandas.execfile('Dataset.xlsx')
df = pandas.read_exel(
    file, sheet_name=0, header=1, Names=None, index_col=None,
    usecols=None, squeeze=False, dtype=None, engine=None, conventers=None,
    true_values=None, false_values=None, skiprows=None, nrows=None,...)
df.replace("ХОЗЛ",1,inplace=True)
df.replace("БА",2,inplace=True)
df.replace("ПНЕВМОНІЯ",3,inplace=True)
```

Рисунок 3.14 – Частина програмного коду з файлу Diplom.ipynb для завантаження даних до Python

Наступним кроком необхідно перевірити дані на нормальний закон розподілу для того, щоб обмежити можливі методи заповнення пропущених значень. Для цього необхідно використати програмний код з рисунків 3.2 та 3.6. В результаті отримаємо загальний вигляд графіку квантіль-квантіль (рисунок 3.15) та значення *W*-статистики та *p* для тесту Шапіро-Уілка (рисунок 3.16). З обох результатів добре видно, що дані зовсім не відповідають нормальному закону розподілу, тому для заповнення пропущених значень в таких даних не можна використовувати методи *EM*-оцінювання та заповнення середнім значенням.

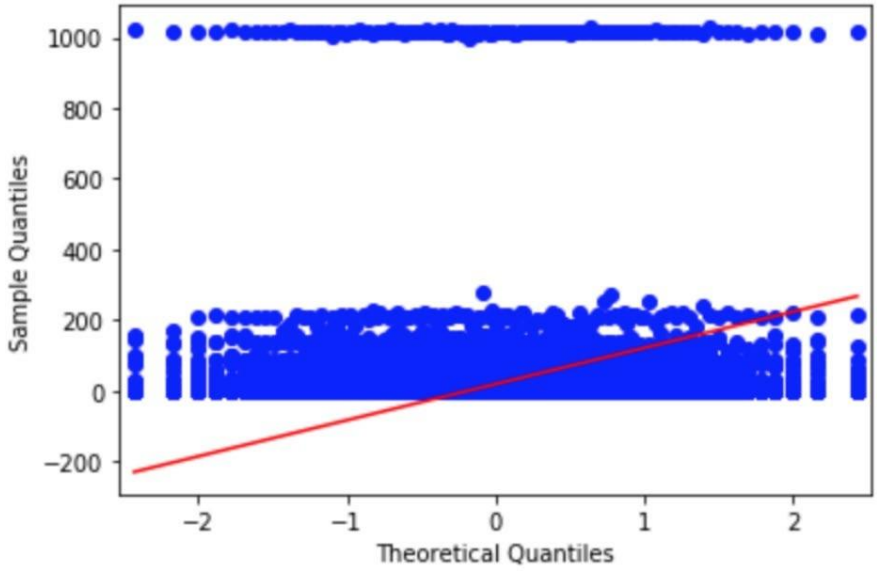

Рисунок 3.15 – Графік квантіль-квантіль для клінічних даних пацієнтів

```
Statistics=0.167, p=0.000
Sample does not look Gaussian (reject H0)
```

Рисунок 3.16 – Значення *W*-статистики та p для тесту Шапіро-Уілка

При аналізі цієї вибірки даних є оцінювання кількості пропущених значень. Для цього використовується код, наведений на рисунку 3.17.

```
gap = df.isnull().sum()
gap.to_excel("gap.xlsx")
```

Рисунок 3.17 – Код для підрахування кількості пропущених значень

В таблиці 3.1 наведено перелік лише тих ознак, які містять пропущені значення та кількість таких пропусків.

Таблиця 3.1 – Перелік тих ознак, які містять пропущені значення та кількість таких пропусків

Ознака	**Кількість пропусків**
ЧДД	10
АД сист	10
АД діаст	10
Пульс	7
Еритроцити	3
Hb	2
Колірний показник	8
Лейкоцити	2
Еозинофіли	3
Палочкоядерні	2
Сегментоядерні	2
Лімфоцити	2
Моноцити	3
СОЕ	4
Білірубін 1	55
Білірубін 2	55
Білірубін 3	55
АСТ	97

Продовження таблиці 3.1

АЛТ	97
Глюкоза	34
Сіалові кислоти	74
Серомукоїди	74
Питома вага	30
Білок	2
Лейк п/зр	2
Еритр неізм п/зр	2

Аналізуючи таблицю 3.1 можна зробити висновки, що загальна кількість пропущених значень складає 645, що є 4,698% від загальної кількості ознак (13728 ознак).

Наступним кроком є заповнення пропущених значень з використанням регресійного методу та методу, що базується на ансамблі Адалін (адаптивний метод). Для цього необхідно заздалегідь оцінити необхідність заповнення пропущеного значення при співвідношенні кількості пропущених значень в спостереженні відносно загальної кількості ознак. Добре видно, що в деяких ознаках пропущена більша половина ознак і заповнення пропущених значень в них є можливим, але не несе жадної інформативності для подальшого аналізу даних. Тому перед заповненням пропусків в даних слід видалити ту частину векторів-ознак, пропущені значення в яких перевищують 10% від загальної кількості ознак.

Програмна реалізація для видалення неінформативних спостережень, наведено на рисунку 3.18

Так, із вибірки медичних даних відділу пульмонології було видалено 9 векторів-ознак, кількість пропущених значень в яких перевищувала 10%, усі дані з пропусками було заповнено на основі двох методів: метод регресійного моделювання і метод, що базується на ансамблі Адалін (адаптивний метод), програмна реалізація якого наведена на рисунку 3.19, а його застосування

на рисунку 3.20.

```
reduceAllData = all.Data.copy()
columnCount = len(all.Data.index)
indexesToDelete = []
for x, row in allData.iterrows():
    nanCount = 0
    for y, value in row.iteritimes():
        if numpy.isnan(value):
            nanCount += 1
            if ((nanCount/columnCount)*100)>10:
                indexesToDelete.append(x)
                reduceAllData.drop(reduceAllData.index[indexesToDelete],inplace=True)
                avaibleData=reduceAllData.dropna()
                result.Data=reduceAllData.copy()
```

Рисунок 3.18 – Програмна реалізація для видалення неінформативних спостережень із вибірки даних

```
def getMissingValue(MissingValueX, MissingValueY, rowCount):
    distances = []
    for x, availableRow in availableData.iterrows():
        res = 0
        for y, value in allData.iloc[MissingValueX].iteritems():
            if pandas.isnull(value)!=True:
                res += abs(availableRow[y]-value)
                distances.append(res/rowCount)
                inverseDistancesSum = 0
                for distance in distances:
                    inverseDistancesSum += 1/distance
                    affiliationLevels = []
                    for distance in distances:
                        affiliationLevels.append((1/distance)/inverseDistancesSum)
                        MissingValue = 0
                        iterator = 0
                        for x, value in availableData[MissingValueY].iteritems():
                            MissingValue += value*affiliationLevels[iterator]
                            iterator += 1
                            return MissingValue
```

Рисунок 3.19 – Адаптивний метод заповнення пропущених значень адаптивном

```
for x, row in allData.iterrows():
    for y, value in row.iteritems():
        if pandas.isnull(value):
            resultData.loc[x,y] = getMissingValue(x, y, len(availableData.index)+1)
```

Рисунок 3.20 – Заповнення пропущених значень з використанням функції getMissingValue

3.4 Аналіз отриманих результатів

При обробці вибірки медичних даних хворих із відділення пульмонології, викривлених «штучно» пропусками, було відновлено і заповнено двома різними методами, які необхідно оцінити і обрати найкращих серед них.

На першому етапі проведемо аналіз кожної ознаки з використанням методу df.describe().

В таблиці 3.2 та таблиці 3.3 наведено основні статистичні характеристики даних з пропущеними значеннями.

Таблиця 3.2 – Основні статистичні характеристики даних з незаповненими пропусками

	ЧДД в хвилину	**АД сист**	**АД діаст**	**Пульс**	**Еритр.**	**Hb**	**Кольоровий показник**	**Лейкоцити**	**Еозинофіли**
Count	122	122	122	125	129	130	124	130	129
Mean	23,53	133,85	82,28	88,73	4,40	137,27	0,93	8,33	1,47
Std	3,38	15,18	11,52	14,04	0,48	15,03	0,05	5,03	1,02
Min	15	60	8	56	3,27	92	0,72	1	1
25%	22	120	80	80	4,2	127,25	0,9	6,2	1

Продовження таблиці 3.2

	ЧДД в хвилину	**АД сист**	**АД діаст**	**Пульс**	**Еритр.**	**Hb**	**Кольоровий показник**	**Лейкоцити**	**Еозинофіли**
50%	24	130	80	90	4,37	137	0,91	7,3	1
75%	26	140	90	95	4,51	148	0,94	9,2	2
Max	33	180	110	150	8,1	176	1,1	57,6	7

Таблиця 3.3 – Основні статистичні характеристики даних з незаповненими пропусками

	Палочко-ядерні	**Сегменто-ядерні**	**Лімфо-цити**	**Моно-цити**	**СОЕ**	**білок**	**лейк п/зр**	**еритр незм п/зр**
Count	130	130	130	129	128	130	130	130
Mean	5,8	62,95	24,36	4,55	16,3	0,01	4,37	0,38
Std	10,08	12,92	9,69	2,47	13,18	0,02	4,38	1,3
Min	1	3	3	2	2	0	0	0
25%	2	59	18	3	7	0	2	0
50%	3	64,5	25	4	11,5	0	3	0
75%	5,75	69	31	6	21	0	5	0
Max	69	87	69	20	74	0,09	40	8

Важливою складовою процесу заповнення є незмінність внутрішньої структури даних. Для відновлених даних, за допомогою регресійного моделювання основні статистичні характеристики наведено в таблиці 3.4 та таблиці 3.5.

Для даних, що було відновлено за допомогою адаптивного методу моделювання, основні статистичні характеристики наведено в таблиці 3.6 та таблиці 3.7.

Таблиця 3.4 – Основні статистичні характеристики даних з пропусками, заповненими на основі регресійного моделювання

	ЧДД в хвилину	**АД сист**	**АД діаст**	**Пульс**	**Еритр.**	**Hb**	**Кольоровий показник**	**Лейкоцити**	**Еозинофіли**
Count	132	132	132	132	132	132	132	132	132
Mean	23,78	134,12	81,3	88,55	4,40	137,27	0,93	8,31	1,46
Std	3,42	15,63	11,25	13,87	0,48	14,91	0,05	4,99	1,01
Min	15	60	8	56	3,27	92	0,72	1	1
25%	21,5	120	80	80	4,2	127,75	0,9	6,2	1
50%	23	130	80	88	4,39	137	0,91	7,35	1
75%	25,25	140	90	94,25	4,5	148	0,94	9,2	1,59
Max	33	180	110	150	8,1	176	1,1	57,6	7

Таблиця 3.5 – Основні статистичні характеристики даних з пропусками, заповненими на основі регресійного моделювання

	Палочко-ядерні	**Сегменто-ядерні**	**Лімфо-цити**	**Моно-цити**	**СОЕ**	**білок**	**лейк п/зр**	**еритр незм п/зр**
Count	132	132	132	132	132	132	132	132
Mean	5,83	62,99	24,36	4,55	16,33	0,01	4,38	0,38
Std	10	12,82	9,69	2,44	12,98	0,02	4,34	1,29
Min	1	3	3	2	2	0	0	0
25%	2	59	18	3	7,75	0	2	0
50%	3	65	24,6	4	12,5	0	3	0
75%	6	69	31	6	21	0	5	0
Max	69	87	69	20	74	0,09	40	8

Таблиця 3.6 – Основні статистичні характеристики даних з пропусками, заповненими на основі адаптивного методу

	ЧДД в хвилину	**АД сист**	**АД діаст**	**Пульс**	**Еритр.**	**Hb**	**Кольоровий показник**	**Лейкоцити**	**Еозинофіли**
Count	132	132	132	132	132	132	132	132	132
Mean	23,38	133,59	82,46	88,32	4,40	137,27	0,92	8,32	1,46
Std	3,29	14,62	11,09	13,76	0,48	14,91	0,05	5	1
Min	15	60	8	56	3,27	92	0,72	1	1
25%	21,5	120	80	80	4,2	127,75	0,9	6,2	1
50%	23	130	80	88	4,39	137	0,91	7,35	1
75%	25,25	140	90	94,25	4,5	148	0,94	9,2	1,59
Max	33	180	110	150	8,1	176	1,1	57,6	7

Таблиця 3.7 – Основні статистичні характеристики даних з пропусками, заповненими на основі адаптивного методу

	Палочко-ядерні	**Сегменто-ядерні**	**Лімфо-цити**	**Моно-цити**	**СОЕ**	**білок**	**лейк п/зр**	**еритр незм п/зр**
Count	132	132	132	132	132	132	132	132
Mean	5,83	62,99	24,36	4,55	16,33	0,01	4,38	0,38
Std	10	12,82	9,62	2,44	12,98	0,02	4,34	1,29
Min	1	3	3	2	2	0	0	0
25%	2	59	18	3	7,75	0	2	0
50%	3,5	65	24,6	4	12,5	0	3	0
75%	6	69	31	6	21	0	5	0
Max	69	87	69	20	74	0,09	40	8

При порівнянні таблиць добре видно, що є вкрай малі зміни за таких параметрів, як середнє значення (mean), стандартне відхилення (std), в деяких

ознаках також змінилися параметри розподілу між максимальним та мінімальним значенням.

Також було проведено порівняння тих самих статистичних ознак для тих ознак, що було видалено з чого отримано обґрунтування необхідності видалення показників з кількістю пропущених значень більшою за 10%.

Основні статистичні характеристики цих показників наведено в таблиці 3.8, результати відновлення ознак, що містять більше за 10% пропусків наведено в таблиці 3.9.

Порівняння цих двох таблиць дозволяє стверджувати, що при зростанні кількості пропущених значень істотно змінюються показники середнього значення, стандартного відхилення та розподілу між максимум та мінімумом, що представлено в таблиці 3.10.

Таблиця 3.8 – Основні статистичні характеристики ознак, що містять більше за 10% пропусків

	Білірубін 1	**Білірубін 2**	**Білірубін3**	**АСТ**	**АЛТ**	**Глюкоза**	**Сіал. Кисл.**	**Сіромуко-їди**	**Питома вага**
Count	77	77	77	35	32	98	58	58	102
Mean	13,11	3,82	10,24	37,01	29,01	5,25	184,4	0,18	1016,19
Std	3,49	1,07	8,01	14,32	15,63	1,63	36,57	0,060	4,38
Min	9,5	2	5,2	0,33	0,33	3,5	100	0,12	1000
25%	11	3,1	7,9	30,5	18,5	4,3	160	0,1425	1014
50%	12,2	3,8	8,7	34	27	4,9	190	0,18	1016
75%	13,9	4	10,1	43	35,5	5,575	200	0,2	1018
Max	31	9	75	69	88	16,5	280	0,46	1028

Таблиця 3.9 – Основні статистичні характеристики заповнених ознак, що містять більше за 10% пропусків

	Білірубін 1	**Білірубін 2**	**Білірубін3**	**АСТ**	**АЛТ**	**Глюкоза**	**Сіал. Кисл.**	**Сіромуко-їди**	**Питома вага**
Count	132	132	132	132	132	132	132	132	132
Mean	13,08	3,77	9,862	33,99	23,42	5,33	200,39	0,19	1016,2
Std	2,66	0,82	6,11	7,54	8,65	1,41	28,1	0,04	3,85
Min	9,5	2	5,2	0,33	0,33	3,5	100	0,12	1000
25%	11,85	3,5	8,4	32,41	32,41	4,67	190	0,18	1015
50%	12,87	3,71	9,2	32,99	32,99	5,39	210,23	0,2	1016,1
75%	13,21	4	9,59	33,66	33,66	5,64	214,01	0,21	1018
Max	31	9	75	69	88	16,5	280	0,46	1028

Таблиця 3.10 – Основні статистичні характеристики заповнених ознак, що містять більше за 10% пропусків

	Білірубін 1	**Білірубін 2**	**Білірубін3**	**АСТ**	**АЛТ**	**Глюкоза**	**Сіал. Кисл.**	**Сіромуко-їди**	**Питома вага**
Count	55	55	55	97	97	34	74	74	30
Mean	-0,02	-0,49	-0,37	-3,01	-5,57	0,08	15,9	0,012	0,027
Std	-0,83	-0,25	1,89	-6,77	-6,97	-0,22	-8,46	-0,018	-0,532
Min	0	0	0	0	0	0	0	0	0
25%	0,85	0,4	0,5	1,91	2,5	0,375	30	0,04	1
50%	0,687	-0,08	0,5	-1	-5,46	0,49	20,23	0,02	0,197
75%	-0,69	0	-0,5	-9,34	-13,53	0,06	14,01	0,01	0
Max	55	55	55	97	97	34	74	74	30

3.5 Перевірка якості заповнення пропущених даних шляхом кластеризації

Для перевірки якості заповнення даних пропонується попередньо необхідно провести нормування даних за допомогою методу MinMaxScaler з бібліотеки sklearn.preprocessing (рисунок 3.21)

```
from slkearn.preprocessing import MinMaxScaler
sckaler = MinMaxScaler(feature_range = (0,1))
sckaler_x = sckaler.fit(dfProc.values)
```

Рисунок 3.21 – Нормування вхідних даних вибірки

Після цього проводиться безпосередньо навчання моделі логістичної регресії за допомогою методу LogisticRegression з sklearn.linear_model. Для оцінки якості роботи методу слід провести оцінку роботи моделі за допомогою метрик з бібліотеки sklearn.metrics, результат роботи методу на даних, що заповнені адаптивним методом представлено на рисунку 3.22.

```
Error = 0.00757575757575757
Mean Absolute Error: 0.00757575757575757
Root mean squared error:  0.00757575757575757
R2 score:  0.9871332488546642
```

Рисунок 3.22 – Значення метрик MAE, RMSE та R^2

Візуалізація клінічних даних пульмонологічних досліджень на базі метода головних компонент приведена на рисунку 3.23 (зелений колір – пацієнти з бронхіальною астмою, червоний колір – пацієнти з ХОЗЛ, синій колір – пацієнти з пневмонією).

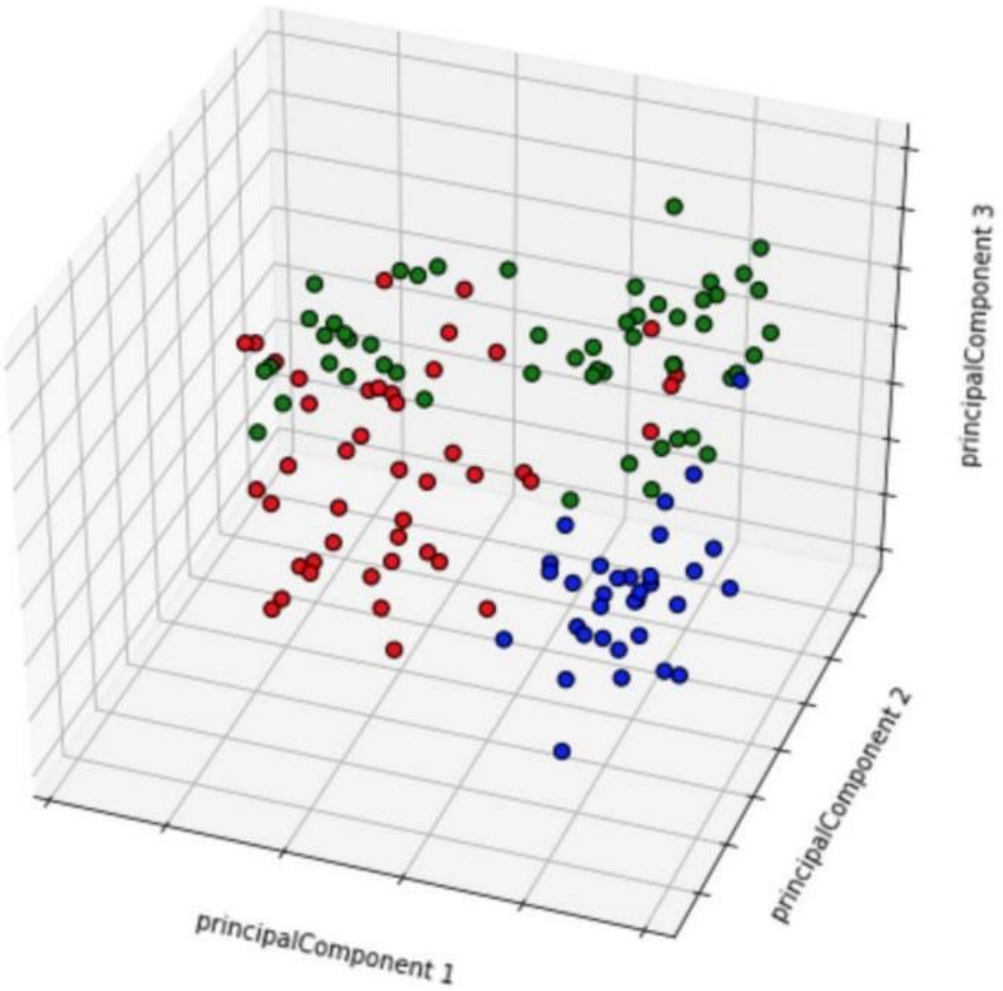

Рисунок 3.23 – Візуалізація клінічних даних пульмонологічних досліджень на базі метода головних компонент

Для порівняння використаного методу з іншими слід провести класифікацію вибірки, яка заповнена іншими методами. Для методу заповнення середніми значеннями результати наведено на рисунку 3.24, для методу регресійного моделювання – на рисунку 3.25.

```
Error = 0.09645467235
Mean Absolute Error: 0.09645467235
Root mean squared error: 0.09645467457
R2 score: 0.8534262748596
```

Рисунок 3.24 – Значення метрик MAE, RMSE та R^2

```
Error = 0.0094765985
Mean Absolute Error: 0.0094765985
Root mean squared error: 0.0094766575
R2 score: 0.9534262748596
```

Рисунок 3.25 – Значення метрик MAE, RMSE та R^2

Для того, щоб зрозуміти де саме модель логістичної регресії невірно класифікувала пацієнтів необхідно побудувати матрицю спряження – Confusion Matrix (рисунок 3.26).

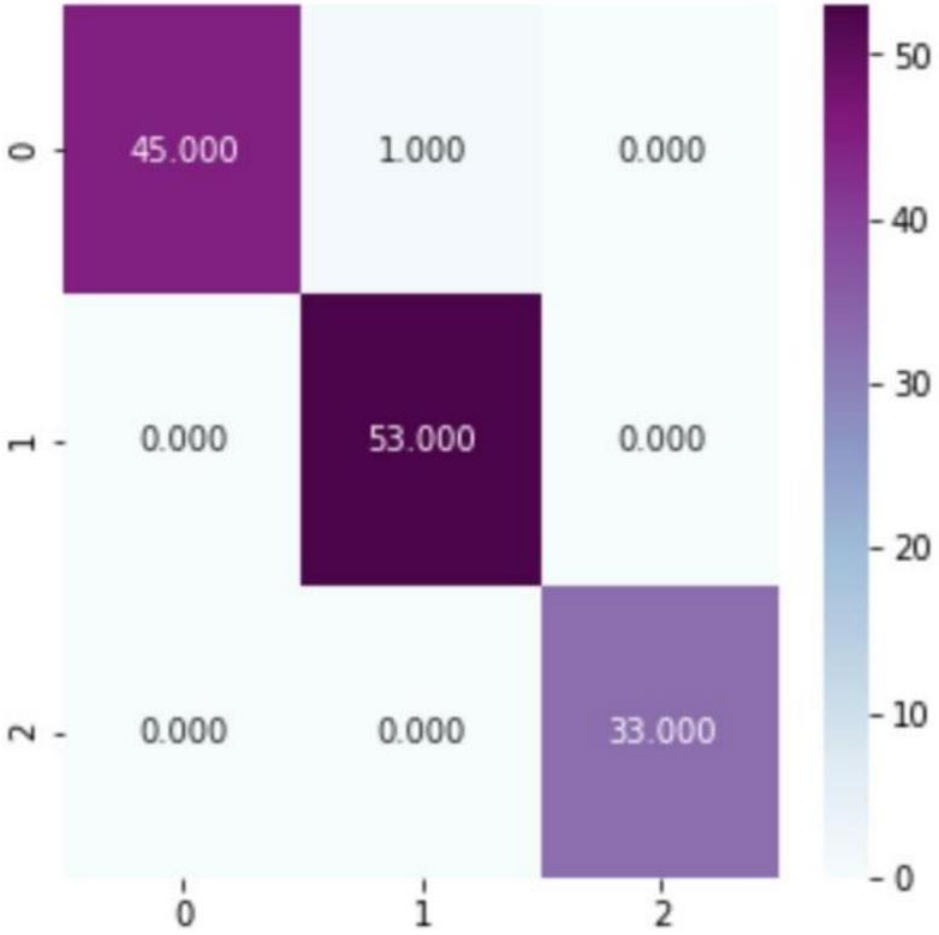

Рисунок 3.26 – Confusion Matrix

Програмна реалізація матриці спряження продемонстрована на рисунку 3.27.

```
from slkearn.metrix import confusion_matrix
import matplotlib.pyplot as plt
print (confusion_matrix(y,y_predicted))
import seaborn as sns
plt.figure(figsize=(5,5))
g = sns.heatmap(confusion_matrix(y,y_predicted),annot=True, fmt='.3f', cmap='BuPu')
```

Рисунок 3.27 – Програмна реалізація матриці спряження

Матриця спряження побудована лише для тої вибірки, яку було відновлено за допомогою адаптивного методу, оскільки з метрик добре видно, що інші методи заповнення дали гірші результати при подальшій кластеризації.

Таким чином можна зробити висновок, що адаптивний метод продемонстрував найкращій результат.

ВИСНОВКИ

У ході кваліфікаційної роботи було проведено ретельний аналіз наукових джерел з теми дослідження, зокрема, методів заповнення пропущених значень у вибірках даних. Перший етап аналізу стосувався вивчення видів даних, їх представлення та причини виникнення пропусків у даних, .

На основі проведеного аналізу була сформульована мета дослідження, яка полягала в аналізі методів заповнення пропущених значень та виборі найефективнішого методу, який максимально точно відновлює втрачені спостереження та кластеризує вибірки даних.

Для цільового аналізу була обрана вибірка клінічних даних в галузі пульмонології. Ця вибірка містила 645 пропущених значень при загальній кількості пацієнтів 132 та кількості ознак 104. Для заповнення пропущених значень були використані метод регресійного моделювання та метод, який ґрунтується на адаптивному підході.

Після того, як були заповнені пропущені значення, результати були перевірені шляхом аналізу статистичних характеристик даних до і після заповнення пропусків. Крім того, для оцінки якості заповнення та класифікації даних була використана методика логістичної регресії, з використанням метрик MAE (середня абсолютна помилка), RMSE (коренева середньоквадратична помилка) та R^2 (коефіцієнт детермінації), а також побудована Confusion Matrix для додаткової оцінки результатів кластеризації.

Запропонований адаптивний метод, що базується на ансамблі адалін продемонстрував найкращій результат.

Результати дослідження апробовано у вигляді тез доповідей під час IV Міжнародної науково-теоретичної конференції «МОДЕРНІЗАЦІЯ СУЧАСНОЇ НАУКИ: ДОСВІД ТА ТЕНДЕНЦІЇ», Сінгапур, [40] та VIII міжнародній науковій конференції «Розвиток науки у XXI столітті», Дортмунд, Німеччина [41].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Gorban, A. N., Kégl, B., Wunsch, D. C., & Zinovyev, A. Y. (Eds.). (2008). *Principal manifolds for data visualization and dimension reduction* (Vol. 58, pp. 96-130). Berlin: Springer.

2. Marwala, T. (Ed.). (2009). *Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques: Knowledge Optimization Techniques*. IGI Global.

3. Wunsch, D. C., Rossiev, A., & Gorban, A. N. (2000). Neural Network Modeling of Data with Gaps.

4. Tkacz, M. (2005). Artificial neural networks in incomplete data sets processing. In *Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM'05 Conference held in Gdansk, Poland, June 13–16, 2005* (pp. 577-583). Springer Berlin Heidelberg.

5. Kohonen, T. (1991). Self-organizing maps: Ophmization approaches. In *Artificial neural networks* (pp. 981-990). North-Holland.

6. Dracopoulos, D. C. (2013). *Evolutionary learning algorithms for neural adaptive control*. Springer.

7. Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

8. Braun, H. (2013). *Neuronale Netze: Optimierung durch Lernen und Evolution*. Springer-Verlag.

9. Haykin, S. (1998). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

10. Han, J., Pei, J., & Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann.

11. Gan, G., Ma, C., & Wu, J. (2020). *Data clustering: theory, algorithms, and applications*. Society for Industrial and Applied Mathematics.

12. А. Шафроненко Є. Бодянський, І. Плісс (2022). *Нечіткі методи*

інтелектуального аналізу даних. GlobeEdit.

13. Abnane, I., Idri, A., & Abran, A. (2023). Optimized fuzzy clustering-based k-nearest neighbors imputation for mixed missing data in software development effort estimation. *Journal of Software: Evolution and Process*, e2529.

14. Волкова, В. В., & Шафроненко, А. Ю. (2011). Нечітка кластеризація масивів даних з пропущеними значеннями. *Індуктивне моделювання складних систем*.

15. Bodyanskiy, Y., Shafronenko, A., & Volkova, V. (2012). Adaptive fuzzy probabilistic clustering of incomplete data. *INFORMATION MODELS & ANALYSES*, *112*.

16. Bodyanskiy, Y., Kolodyazhniy, V., & Stephan, A. (2001). An adaptive learning algorithm for a neuro-fuzzy network. In *Computational Intelligence. Theory and Applications: International Conference, 7th Fuzzy Days Dortmund, Germany, October 1–3, 2001 Proceedings 7* (pp. 68-75). Springer Berlin Heidelberg.

17. Шафроненко, А. Ю., & Бодянський, Є. В. (2023). Адаптивний підхід до нечіткої кластеризації на основі еволюційної оптимізації алгоритму сірих вовків. *Збірник наукових праць Харківського національного університету Повітряних Сил*, (1 (75)), 77-81.

18. Shafronenko, A., Bodyanskiy, Y., & Rudenko, D. (2020). *Neuro-fuzzy clustering of Distorted Data Using Cat Swarm Optimization*. LAP LAMBERT Academic Publishing.

19. Bodyanskiy, Y., Pliss, I., & Shafronenko, A. (2022). Adaptive neuro-fuzzy clustering of distorted data based on prototype-centroid strategy using evolutionary procedures. *Artificial Intelligence*, *27*, 239-244.

20. Плісс, І. П., Шевякова, А. Ю., & Шевякова, Ю. Ю. (2011). Нейромережеве відновлення пропусків у таблицях даних. *Наукові праці [Чорноморського державного університету імені Петра Могили]. Сер.: Комп'ютерні технології*, (160, Вип. 148), 59-61.

21. Krishnapuram, R., & Keller, J. M. (1993). A possibilistic approach to clustering. IEEE transactions on fuzzy systems, 1(2), 98-110.

22. Xu, R., Xu, J., & Wunsch, D. C. (2012). A comparison study of validity indices on swarm-intelligence-based clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(4), 1243-1256.

23. Shafronenko, A. Y., Bodyanskiy, Y. V., & Rudenko, D. A. (2020). Adaptive Neuro-Fuzzy Methods for Distorted Data Clustering.

24. Шафроненко, А. Ю., & Бодянський, Є. В. (2023). Нечітка достовірна кластеризація великих масивів даних з гіпереліпсоїдальними класами з довільною орієнтацією осей. *Наука і техніка Повітряних Сил Збройних Сил України*, (1 (50)), 93-99.

25. Shafronenko, A. Y., Kasatkina, N. V., Bodyanskiy, Y. V., & Shafronenko, Y. O. (2023). CREDIBILISTIC ROBUST ONLINE FUZZY CLUSTERING IN DATA STREAM MINING TASKS. *Radio Electronics, Computer Science, Control*, (3), 97-97.

26. Shafronenko, A., Dolotov, A., Bodyanskiy, Y., & Setlak, G. (2018, August). Fuzzy clustering of distorted observations based on optimal expansion using partial distances. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 327-330). IEEE.

27. Bodyanskiy, Y. V., Shafronenko, A., & Rudenko, D. (2019). Online Neuro Fuzzy Clustering of Data with Omissions and Outliers based on Completion Strategy. In *CMIS* (pp. 18-27).

28. Bodyanskiy, Y., Shafronenko, A., & Mashtalir, S. (2020). Online robust fuzzy clustering of data with omissions using similarity measure of special type. In Lecture Notes in Computational Intelligence and Decision Making*: Proceedings of the XV International Scientific Conference "Intellectual Systems of Decision Making and Problem of Computational Intelligence"(ISDMCI'2019), Ukraine, May 21–25, 2019 15 (pp. 637-646*). Springer International Publishing.

29. Bodyanskiy, Y. V., Shafronenko, A., & Klymova, I. (2021, April). Adaptive Recovery of Distorted Data Based on Credibilistic Fuzzy Clustering

Approach. In *COLINS* (pp. 6-15).

30. Shafronenko, A. Y., & Rudenko, D. A. (2020). ONLINE RECURRENT METHOD OF CREDIBILISTIC FUZZY CLUSTERING. *BBK 91*, 37.

31. Bodyanskiy, Y., Shafronenko, A., & Pliss, I. (2021). Правдоподібна нечітка кластеризація даних на основі еволюційного методу божевільних котів. *System research and information technologies*, (3), 110-119.

32. Шафроненко, А. Ю., Бодянський, Є. В., & Руденко, Д. О. (2023). Модифікований рекурентний метод достовірної нечіткої кластеризації з використанням оптимізаційної процедури на основі косяків риб. *Системи обробки інформації*, (1 (172)), 92-96.

33. Kalton, G. (1986). The treatment of missing survey data. *Survey methodology*, *12*, 1-16.

34. Asuncion, A., & Newman, D. (2007). UCI machine learning repository.

35. Frank, A. (2010). UCI machine learning repository. *http://archive. ics. uci. edu/ml*.

36. Agarap, A. F. M. (2018, February). On breast cancer detection: an application of machine learning algorithms on the wisconsin diagnostic dataset. In *Proceedings of the 2nd international conference on machine learning and soft computing* (pp. 5-9).

37. Hackeling, G. (2017). *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd.

38. Bodyanskiy, Y. V., Pliss, I. P., Shafronenko, A. Y., & Kalynychenko, O. V. (2022). НЕЧІТКА ДОВІРЧА КЛАСТЕРИЗАЦІЯ ДАНИХ НА ОСНОВІ АНАЛІЗУ ЩІЛЬНОСТІ РОЗПОДІЛУ ДАНИХ ТА ЇХ ПІКІВ. *Radio Electronics, Computer Science, Control*, (3), 58-58.

39. Bodyanskiy, Y., Shafronenko, A., Klymova, I., & Polyvoda, V. (2022). Robust Recurrent Credibilistic Modification of the Gustafson-Kessel Algorithm. In *Lecture Notes in Computational Intelligence and Decision Making: 2021 International Scientific Conference" Intellectual Systems of Decision-making and Problems of Computational Intelligence", Proceedings* (pp. 613-623). Springer

International Publishing.

40. Руденко, Д., Лотвінова, В., & Безверха, Є. (2023). НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ В ЗАДАЧАХ ОБРОБКИ ДАНИХ. *Collection of scientific papers «SCIENTIA»*, (September 22, 2023; Singapore, Singapore), 94-95.

41. Rudenko, D. O., Bezverkha, Ye. V., Lotvinova, V. V. (2023). NEURAL NETWORK RECOVERY OF OMISSIONS IN DATA SETS. *VIII international scientific conference*, (September 14-15, 2023; Dortmund, Germany), 87-88.