МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: **«Дослідження впливу алгоритмів попередньої обробки даних на результати створення моделей машинного навчання»**

Виконав: студент групи Іт-61

Спеціальності 126 «Інформаційні системи та технології»
(шифр і назва)
Троць Андрій Мар`янович
(Прізвище та ініціали)

Керівник: д.т.н., професор Тригуба А.М.
(Прізвище та ініціали)

Рецензент: к.т.н., доцент Кригуль Р.Є.
(Прізвище та ініціали)

ДУБЛЯНИ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Другий (магістерський) рівень вищої освіти
Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри_____

д.т.н., проф. А.М. Тригуба
«____» _____2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Троцю Андрію Мар`яновичу

1. Тема роботи: «Дослідження впливу алгоритмів попередньої обробки даних на результати створення моделей машинного навчання»

Керівник роботи Тригуба Анатолій Миколайович, професор
затверджені наказом по університету від 12.09.2024 року № 616/к-с.

2. Строк подання студентом роботи 10.12.2024 р.

3. Вихідні дані до роботи: *дані для оцінення алгоритмів попередньої обробки; алгоритми створення моделей машинного навчання.*

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)
Вступ.
Аналіз стану дослідження алгоритмів попередньої обробки даних.
Теоретичні основи попередньої обробки даних та її впливу на моделі машинного навчання.
Результати дослідження впливу алгоритмів попередньої обробки даних на результати створення моделей машинного навчання.
Охорона праці та безпека у надзвичайних ситуаціях.
Визначення показників ефективності від використання алгоритмів попередньої обробки даних на результати створення моделей машинного навчання.
Висновки та пропозиції.
Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов»язкових слайдів): *аналіз стану дослідження алгоритмів попередньої обробки даних; теоретичні основи попередньої обробки даних та її впливу на моделі машинного навчання; результати дослідження впливу алгоритмів попередньої обробки даних на результати створення моделей машинного навчання; визначення показників ефективності від використання алгоритмів попередньої обробки даних на результати створення моделей машинного навчання.*

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 5	*Тригуба А.М., зав. кафедри інформаційних технологій*		
4	*Городецький І.М., доцент кафедри фізики, інженерної графіки та безпеки виробництва*		

7. Дата видачі завдання 12 вересня 2024 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	Примітка
1	*Написання першого розділу*	*12.09-20.09.24*	
2	*Виконання другого розділу та аркушів ілюстраційного матеріалу до нього*	*21.09-14.10.24*	
3.	*Виконання третього розділу та аркушів ілюстраційного матеріалу до нього*	*15.010-10.11.24*	
4.	*Написання розділу «Охорона праці та безпека у надзвичайних ситуаціях»*	*11.11-20.11.24*	
5.	*Оцінення ефективності запропонованої системи*	*21.11-30.30.24*	
6.	*Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу*	*01-04.12.24*	
7.	*Завершення роботи вцілому*	*05-10.12.24*	

Студент _____ Троць А.М.
(підпис)

Керівник роботи _____ Тригуба А.М.
(підпис)

УДК 004.032.26:004.8

Дослідження впливу алгоритмів попередньої обробки даних на результати створення моделей машинного навчання.

Троць А.М. Кафедра інформаційних технологій – Дубляни, ЛНУП, 2024.

Кваліфікаційна робота: 76 с. текст. част., 22 рис., 8 табл., 15 арк. ілюстраційного матеріалу, 52 джерела.

Виконано аналіз стану використання та попередньої обробки даних. Подано етапи попередньої обробки даних. Проведено аналіз стану дослідження алгоритмів попередньої обробки даних. Сформульовано завдання кваліфікаційної роботи.

Проаналізовано основи машинного навчання та роль попередньої обробки даних. Наведено основні етапи інтеграції даних. Показано вплив алгоритмів обробки даних на точність і ефективність моделей. Проаналізовано використання нормалізації та стандартизації, усунення пропусків у даних, видалення аномальних значень, вибір ключових ознак та вибір та попередній аналіз даних для машинного навчання.

Здійснено формування експерименту дослідження впливу алгоритмів попередньої обробки даних на результати створення моделей прогнозування. Здійснена попередня обробка даних для машинного навчання моделей. Подано результати створення моделей машинного навчання за використання різних алгоритмів попередньої обробки даних. Наведено результати навчання моделей прогнозування оцінок студентів з математики за використання різних алгоритмів попередньої обробки даних.

Розроблено заходи із охорони праці. Проведено визначення показників ефективності від використання алгоритмів попередньої обробки даних на результати створення моделей машинного навчання.

Ключові слова: алгоритми, попередня обробка даних, моделі, машинне навчання, ефективність.

ЗМІСТ

ВСТУП

Сьогодні дані стали ключовим ресурсом, що досягає успішності бізнесу, наукових досліджень та технологічного прогресу. Великі масиви інформації, які накопичуються в різних сферах, вимагають ефективного аналізу та використання [26]. Машинне навчання, як підрозділ штучного інтелекту, пропонує інструменти для автоматизованого аналізу та прогнозування на основі цих даних. Однак ефективність моделей машинного навчання значною мірою залежить від якості даних, які використовуються для навчання.

Алгоритми попередньої обробки даних мають ключове значення для забезпечення високої точності моделей, після чого вони впливають на видалення шуму, заповнення пропусків, масштабування ознак та інші аспекти, які застосовують придатність даних до аналізу [17]. Попередня обробка дозволяє мінімізувати вплив некоректних, неповних або нерелевантних даних на результати, що особливо важливо в умовах складних багатовимірних завдань.

Актуальність теми кваліфікаційної роботи зумовлена швидким зростанням обсягів даних та цінністю їх ефективного використання. У різних сферах, таких як медицина, фінанси, агропромисловий комплекс, транспорт, якість обробки даних прямо впливають на точність прогнозів і рішень, які приймаються на основі моделей машинного навчання. поки в наявності багато методів попередньої обробки, проблема вибору оптимального алгоритму для конкретної задачі залишається відкритою [27].

Об’єкт дослідження – процес створення моделей машинного навчання.

Предмет дослідження – вплив алгоритмів обробки даних на точність і продуктивність моделей машинного навчання.

Мета дослідження – розробка рекомендацій щодо вибору та застосування алгоритмів попередньої обробки даних для підвищення ефективності моделей машинного навчання.

Новизна роботи полягає в комплексному підході до оцінки впливу алгоритмів попередньої обробки даних на результати навчання моделей та розробці універсальних рекомендацій для підвищення точності прогнозів у різних сферах.

Результати дослідження пропонується використовувати в різних прикладних задачах, таких як прогнозування врожайності, аналіз ризиків, оптимізація виробничих процесів тощо. Таким чином, ця робота має теоретичне та практичне значення, спрямоване на вдосконалення процесів роботи з даними та створення високоякісних моделей машинного навчання.

РОЗДІЛ 1.
АНАЛІЗ СТАНУ ДОСЛІДЖЕННЯ АЛГОРИТМІВ ПОПЕРЕДНЬОЇ ОБРОБКИ ДАНИХ

1.1. Аналіз стану використання та попередньої обробки даних

У сучасному світі цифровізація набуває величезних масштабів. І державні, і приватні компанії переводять свій бізнес в онлайн-формат. Отже, кожен день величезна кількість інформації генерується цифровим способом від користувачів. З кожним роком кількість інформації стрімко зростає. Всю цю інформацію необхідно транспортувати, зберігати та обробляти. Також не існує єдиного формату для даних, вони можуть зберігатися в різних формах і структурах. Для цього потрібні величезні обчислювальні потужності та використання новітніх технологій у сфері інженерії даних [12]. При цьому виникають збої, що призводять до неточностей даних або погіршення їх якості.

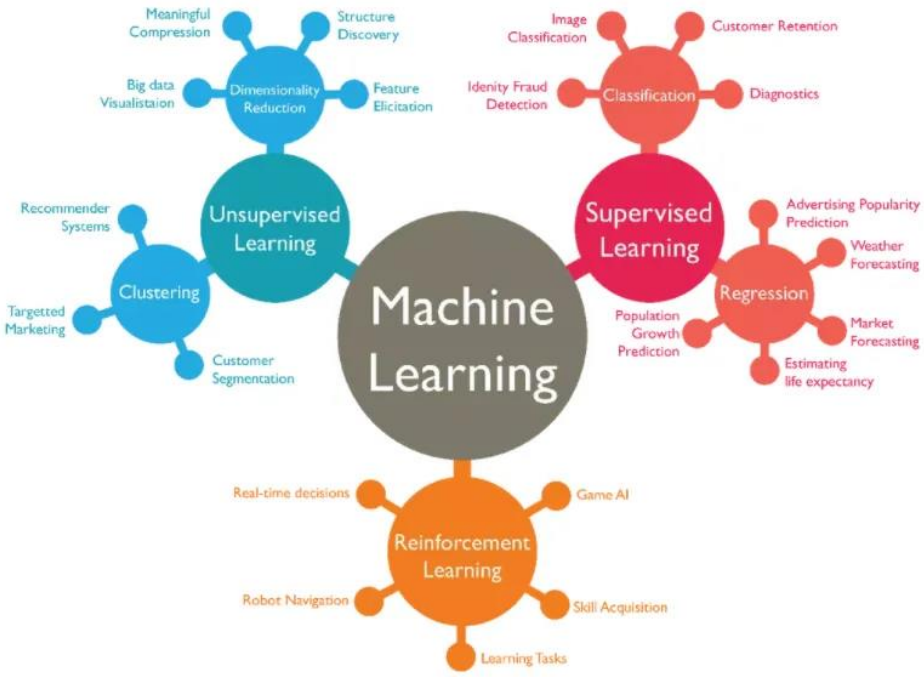

Рисунок 1.1 – Використання даних у алгоритмах машинного навчання [27]

Продуктивність і ефективність алгоритмів машинного навчання [27] нерозривно пов’язані з характеристиками та якістю [33] вхідних даних. В епоху великих даних [10], оскільки обсяг і різноманітність доступних джерел даних продовжує розширюватися, важливість підготовки цих даних стає все більш очевидною.

Заслуговує на увагу процес введення даних. Додавати нові дані до існуючого сервера даних можна з різних причин, щоб оновити базу даних новими даними та додати більше різноманітних даних для покращення продуктивності моделей машинного навчання. Або виправлення помилок вихідного набору даних зазвичай виконується автоматизовано за допомогою деяких зручних інструментів.

Рисунок 1.2 – Формування масивів даних

Є три способи додавання даних до існуючого сервера даних:

- ✓ пакетне вставлення – дані вставляються масово, зазвичай у фіксований час;
- ✓ впровадження в режимі реального часу – дані вводяться одразу після їх створення;

- ✓ введення потоку – дані вводяться безперервним потоком. Часто використовується в режимі реального часу.

Приватні компанії зараз дуже зацікавлені у впровадженні новітніх технологій з використанням машинного навчання та штучного інтелекту [19] для своїх потреб. Останнім часом активно відкриваються стартапи у сфері штучного інтелекту та машинного навчання, у розвиток цих галузей інвестуються величезні кошти і ця тенденція буде тільки продовжуватися. Одними з ключових переваг автоматизації з використанням штучного інтелекту та алгоритмів машинного навчання є збільшення продуктивності, часу та економічної ефективності, зменшення людських помилок, прискорення прийняття бізнес-рішень, прогнозування переваг клієнтів та максимізація продажів [38]. Великі компанії, яким потрібно аналізувати та працювати з великою кількістю даних, мають цілі команди, які контролюють і підтримують якість даних. Це ще раз доводить важливість якості даних для подальшого використання.

1.2. Етапи попередньої обробки даних

На рис. 1.3 представлені етапи попередньої обробки даних. Очищення даних – це процес виявлення неправильних або зашумлених даних і їх виправлення або видалення з набору даних. Загалом, він працює над виявленням і заміною неповних, неточних, нерелевантних або будь-яких інших шумових даних і записів. Хоча методи, що використовуються, відрізняються залежно від вимог моделі, основні кроки: Видаліть нерелевантні або повторювані дані

Це часто трапляється в наборі даних. Коли дані об’єднуються з іншого джерела, скопіюйте дані або дані з кількох клієнтів. Існує можливість створення дублікатів даних.

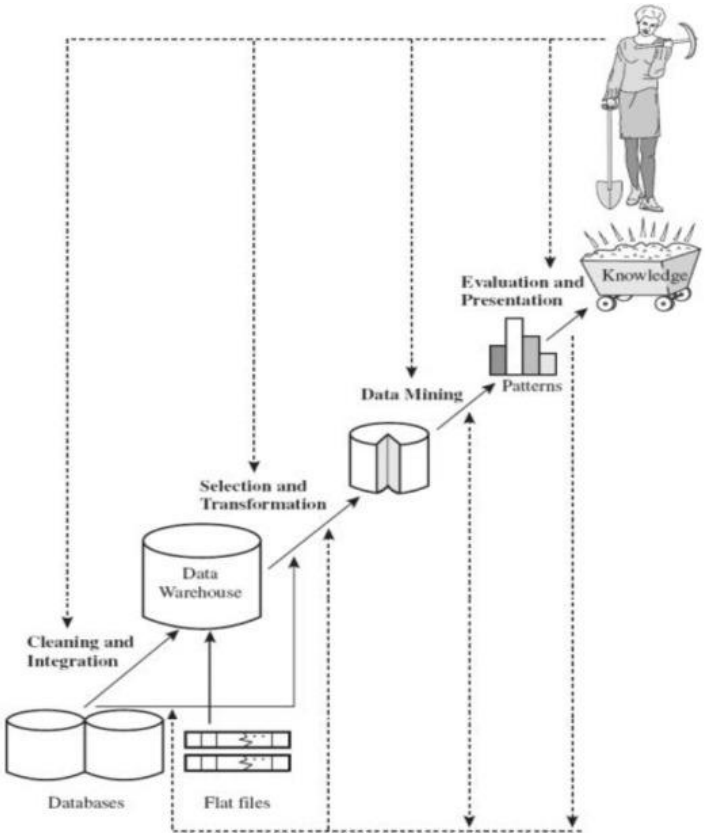

Рисунок 1.3 – Етапи попередньої обробки даних

Невідповідності виникають через неправильне маркування категорій або класів. Це також може виникнути через дивні угоди про найменування, помилки або неправильне використання великих літер.

Зазвичай у наборі даних відсутні значення певних стовпців. Проблема може виникнути через правила перевірки даних або збір даних. Але необхідно враховувати відсутні значення, оскільки це може усунути особливість моделі через відсутні значення. Якщо розумна кількість значень відсутня, то прості методи інтерполяції можуть заповнити ці питання. Найпоширеніший метод, який використовується для боротьби з цим, полягає в використанні середніх, медіанних або модових значень щодо характеристик моделі. а. Непослідовні цінності

Це може бути через людську помилку або згенеровано під час роботи з первинними даними. Отже, для процесу оцінки даних стає необхідним дізнатися тип даних об’єкта та перевірити, чи всі об’єкти даних належать до одного типу.

Остаточний набір даних повинен відповісти на такі запитання після процесу:

- ✓ Чи мають ці дані сенс?
- ✓ Чи відповідають дані спеціальним правилам поля?
- ✓ Це обґрунтовує чи спростовує ознаку моделі?
- ✓ Чи здатний він ідентифікувати шаблони в даних?

Неузгоджені дані можуть призвести до помилкових висновків і прогнозів. Як наслідок, високоякісні дані мають відповідати таким критеріям: валідність (обмеження, діапазон, шаблони та інші), точність, повнота, послідовність та однорідність.

У подальшому виконується етап обробки шуму. Якщо шум продовжується в класі після виявлення гучних явищ, є три підходи до боротьби з ним. По-перше, шум можна ігнорувати, якщо модель достатньо міцна, щоб витримати надмірну підгонку. По-друге, шум у наборі даних можна відфільтрувати, змінити, відшліфувати або змінити мітки. Якщо дані з атрибутом зберігаються, такі методи, як фільтрація чи полірування помилкового значення атрибута, видалення його з набору даних або імпутація, можуть передбачити, що потрібно очистити, і виявити більше підозрілих значень [15].

Це техніка для зменшення впливу незначних помилок спостереження. Значення розділяються на невеликі відсіки у вихідних даних, а потім замінюються загальними значеннями, отриманими для цього відсіку. Це згладжує вхідні дані та, у випадку короткого набору даних, може знизити ймовірність переобладнання [14, 39].

При цьому виділяють два методи групування:

- ✓ Групування рівної частоти – прив’язки мають однакову частоту;
- ✓ Групування однакової ширини – контейнери мають однакову ширину з діапазоном кожного контейнера, який обчислюється як:

$$[\min+w], [\min+2w], \ldots, [\min+nw], \quad (1.1)$$

де w = (max-min) / (кількість контейнерів).

Регресія – це техніка керованого машинного навчання, яка використовується для прогнозування безперервності. Вона встановлює зв’язок між змінними шляхом оцінки того, як змінна впливає на іншу. Щоб оцінити прогнози за допомогою алгоритму регресії, необхідно враховувати показники дисперсії та зміщення.

Дисперсія – це величина, на яку змінюється оцінка цільової функції, якщо навчальні дані відрізняються. Щоб уникнути помилкових прогнозів, дисперсія моделі повинна бути низькою.

Зміщення – алгоритм, який має тенденцію постійно вивчати неправильні речі, не враховуючи всю інформацію даних. Якщо в наборі даних є невідповідності, як-от відсутні значення, менша кількість атрибутів або помилки, це може призвести до упередженого результату. Отже, щоб отримати точність, потрібно було підтримувати низький зсув моделі.

1.3. Аналіз стану дослідження алгоритмів попередньої обробки даних

Крім того, попередня обробка даних ще не може бути повністю автоматизована, оскільки це досить складний процес (рис. 1.4), який може включати різні техніки, алгоритми та повинен враховувати специфіку даних і завдання, щоб вибрати методи та досягти найкращих результатів [29].

Рисунок 1.4 – Схема процесу попередньої обробки даних

Це процес зменшення початкового обсягу даних і представлення їх у значно меншому обсязі (рис. 1.5). Це забезпечує цілісність даних, одночасно зменшуючи дані. Коли дані мають значну важливість, їх необхідно скоротити. Дізнатися потрібну інформацію може стати важко, а обробка складних запитів може зайняти багато часу.

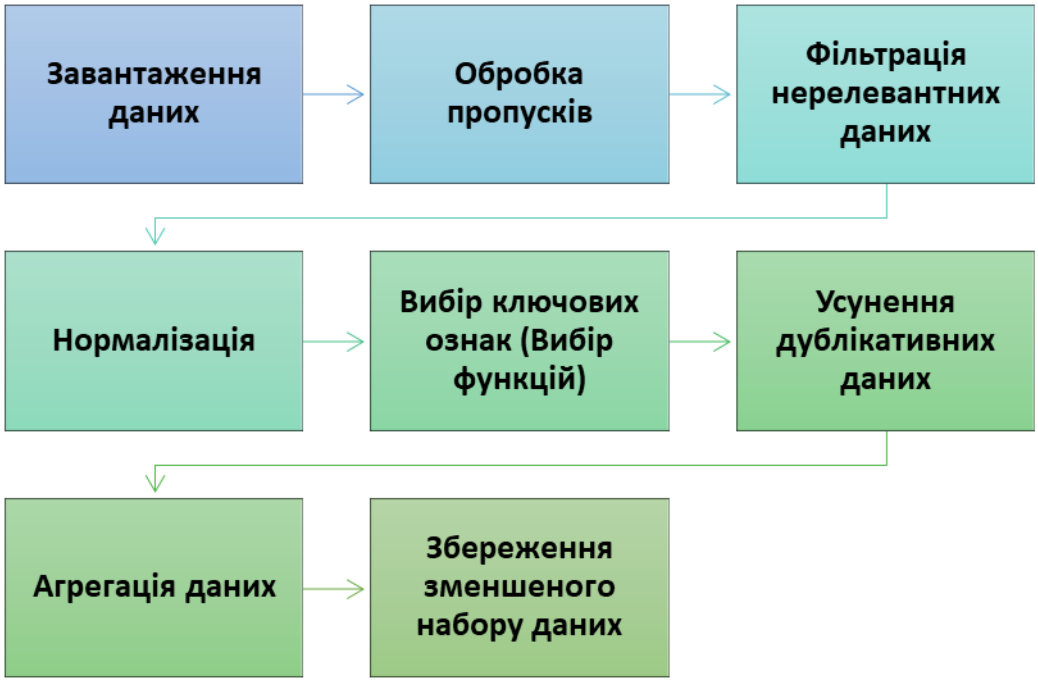

Рисунок 1.5 – Процес зменшення початкового обсягу даних

Є кілька статей, у яких вчені досліджували вплив якості даних на різні моделі машинного навчання в різних областях. Однак у цій статті дослідження буде проведено у зв’язку зі сферою освіти та пов’язаними даними.

При прогнозуванні діабету за допомогою моделей машинного навчання авторам вдалося підвищити ефективність моделі за допомогою методів попередньої обробки даних, таких як: вставка відсутніх значень і вибір ознак [31].

Аналіз роботи [51] з нейронними мережами для прогнозування стану внутрішнього середовища, автори роблять висновок, що окреме прогнозування кількох змінних без попередньої обробки даних може дати такі ж точні прогнози, як і одночасне прогнозування з попередньою обробкою даних, однак

обчислювальні витрати на навчання кількох нейронних мереж для слід враховувати окреме прогнозування.

В іншій статті [22] вчені вирішили з’ясувати, як обробка даних вплине на моделі машинного навчання в задачах прогнозування. У результаті виявилося, що обробка даних може мати сильний позитивний вплив на результати та якість прогнозу, але також може мати негативний вплив на ефективність прогнозу з використанням методів машинного навчання.

Автори іншої роботи [17] досліджували можливості 6 різних алгоритмів, а також методи попередньої обробки даних у задачі класифікації сигналу електроенцефалограми для визначення сонливості водія за кермом за допомогою машинного навчання. У результаті вони прийшли до висновку, що тип алгоритму, який використовується для вирішення проблеми, має більший вплив на результати, ніж попередня обробка. Однак обробка даних також покращує результати моделювання. Крім того, у ситуаціях, коли підготовка даних неможлива, краще використовувати деревоподібні алгоритми машинного навчання.

Інша робота [8] використовувала алгоритми машинного навчання для прогнозування забруднення повітря. Також було оцінено вплив попередньої обробки даних і вибору функцій. В результаті при використанні цих методів досягнуто кращої точності та ефективності моделей.

В іншому дослідженні [35] про попередню обробку ближнього інфрачервоного спектру дослідники дійшли висновку, що попередня обробка даних має великий вплив на невеликі набори даних. Зі збільшенням обсягу даних методи попередньої обробки втрачають свою ефективність. Моделі, навчені на великій кількості даних, більш точні.

Ще одна стаття [13] досліджувала вплив обробки даних на аналіз корпоративних даних. Було розглянуто різні методи попередньої обробки, а також різні алгоритми машинного навчання. У результаті було емпірично доведено, що деякі з алгоритмів попередньої обробки мають істотний вплив на точність прогнозування.

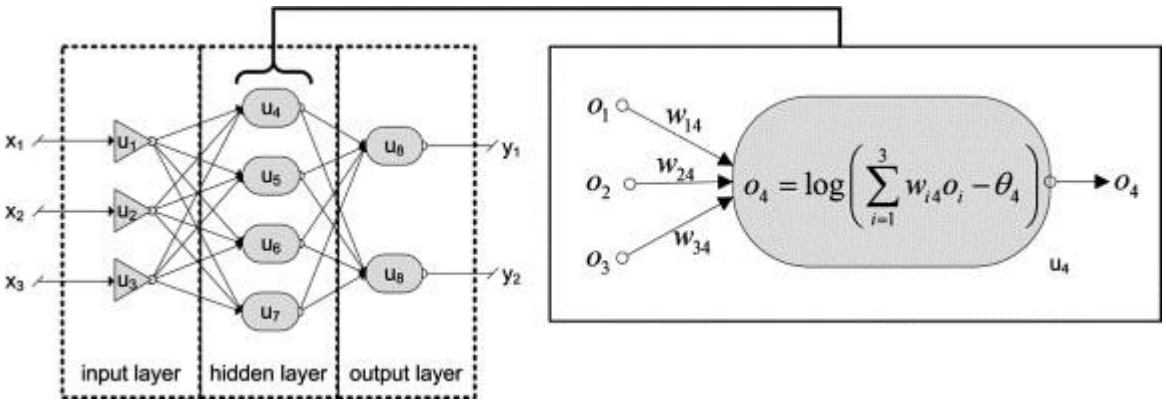

Рисунок 1.6 – Трирівневий MLP, що показує обробку інформації всередині вузла, використовуючи зважену суму як вхідну функцію, логістичну функцію як функцію сигмоїдної активації та функцію виводу ідентичності [13]

У випадку обробки неструктурованих медичних даних [24] дослідники також проаналізували найвпливовіші методи попередньої обробки даних. У результаті виявилося, що для аналізу рукописного тексту найбільш ефективними етапами є нормалізація та виправлення помилок.

Машинне навчання в галузі освіти продовжує розвиватися. Знаходяться нові шляхи застосування цих технологій. У цьому дослідженні розглядатиметься використання машинного навчання для аналізу та прогнозування оцінок студентів на іспитах залежно від показників їхнього життя та сім'ї. Машинне навчання також можна використовувати для досліджень, автоматизації управління, вдосконалення онлайн-навчання, персоналізації навчання, створення розумних додатків.

Візуалізація даних [9] також є важливим кроком у процесі вирішення задачі за допомогою алгоритмів машинного навчання. Візуалізація застосовується на різних етапах процесу вирішення задачі.

Незважаючи на очевидну важливість попередньої обробки даних, повне розуміння її впливу в реальному світі залишається динамічною сферою досліджень і застосування. Складність цього питання виникає через взаємодію різноманітних методів попередньої обробки, унікальних характеристик різних наборів даних і специфіки моделей машинного навчання. Таким чином, вплив

попередньої обробки даних не є однозначним, а натомість залежить від контексту.

Крім того, існує обмежене уявлення про те, як різні методи попередньої обробки впливають на стійкість моделей машинного навчання до шумних даних, викидів. Цей брак знань може перешкодити ширшому впровадженню найкращих практик попередньої обробки даних, обмежуючи потенціал машинного навчання в реальних програмах.

У зв’язку з цим тема впливу якості даних на моделі машинного навчання в освіті потребує додаткових досліджень.

1.4. Завдання кваліфікаційної роботи

Попередня обробка даних є ключовим етапом машинного навчання, оскільки якість даних значно впливає на точність і ефективність моделі. В умовах зростання обсягів інформації важливо вибрати оптимальні алгоритми для підвищення продуктивності. Існує потреба в детальному дослідженні алгоритмів обробки результатів моделювання. Це дослідження сприяти ефективнішому використанню даних у різних прикладних задачах. Практична значущість роботи ускладнюється у підвищенні точності та стійкості моделей машинного навчання.

Таким чином, виконання кваліфікаційної роботи на тему «Дослідження впливу алгоритмів попередньої обробки даних на результати створення моделей машинного навчання» є досить актуальним завданням у сучасних умовах. У зв’язку з цим були сформульовано наступні завдання роботи:

- ✓ провести аналіз існуючих алгоритмів попередньої обробки даних;
- ✓ розробити методику оцінки їх впливу на результати моделей;
- ✓ провести експерименти з використанням різних алгоритмів на тестових наборах даних;
- ✓ виділити найефективніші алгоритми для конкретних типів завдань;

- ✓ розробити практичні рекомендації щодо вибору алгоритмів обробки даних.

РОЗДІЛ 2.
ТЕОРЕТИЧНІ ОСНОВИ ПОПЕРЕДНЬОЇ ОБРОБКИ ДАНИХ ТА ЇЇ ВПЛИВУ НА МОДЕЛІ МАШИННОГО НАВЧАННЯ

2.1. Основи машинного навчання та роль попередньої обробки даних

Машинне навчання (ML) – це підгалузь штучного інтелекту, яка зосереджена на створених алгоритмах, здатних автоматично виявляти закономірності в даних і використовувати їх для прогнозування чи класифікації. Основою машинного навчання є робота з того, що підкреслює важливість попередньої обробки, оскільки якість даних напряму впливає на ефективність моделей.

Алгоритми машинного навчання виділяються на такі основні категорії, які представлено на рис. 2.1.

Рисунок 2.1 – Алгоритми машинного навчання, які потребують попередньої обробки даних

Навчання з учителем (Supervised Learning) Використовує дані з мітками (labelled data), де кожен запис має відповідь (наприклад, класифікація зображення).

Навчання без учителя (Unsupervised Learning) використовує дані без міток, орієнтуючись на виявлення структури у даних (кластеризація, зменшення розмірності).

Навчання з підкріпленням (Reinforcement Learning) передбачає, що моделі навчаються шляхом взаємодії з середовищем і отримують винагороди за дії, що ведуть до мети.

Процес побудови моделей машинного навчання включає кілька етапів, представлених на рисунку 2.2.

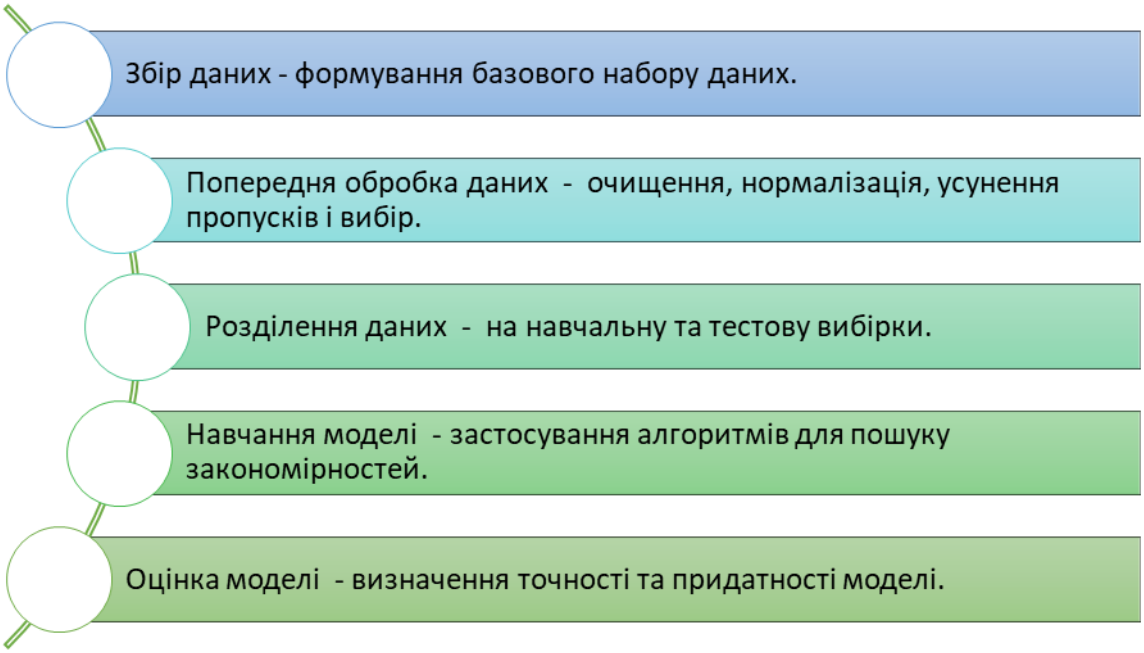

Рисунок 2.2 – Етапи побудови моделі машинного навчання

Попередня обробка даних є ключовим етапом у процесі машинного навчання. Вона забезпечує підготовку даних до аналізу та побудови моделей. Основні завдання цього етапу:

✓ очищення даних – видалення шуму, дублікативних або аномальних записів;

✓ усунення пропусків – заповнення відсутніх значень середнім, медіаною або за допомогою моделей;

✓ нормалізація даних – приведення числових ознак до одного масштабу для забезпечення коректної роботи алгоритмів. Формула мін-макс нормалізації:

$$X_{норм} = \frac{X - \min(X)}{\max(X) - \min(X)}, \qquad (2.1)$$

Де X – вихідне значення; $\min(X)$, $\max(X)$ – мінімальне та максимальне значення ознаки.

✓ кодування категорійних ознак: перетворення текстових даних у числові (наприклад, One-Hot Encoding).

Наведемо приклад впливу обробки даних на точність моделі (табл. 2.1).

Таблиця 2.1 – Вплив обробки даних на точність моделей

Алгоритм	Без обробки даних (точність)	Після обробки (точність)
Логістична регресія	68%	85%
Випадковий ліс	72%	89%

Попередня обробка даних є невід'ємною частиною процесу машинного навчання. Вона дозволяє підвищити якість вхідних даних, що напряму впливає на точність, швидкість і стабільність моделей. Без цього етапу навіть найкращі алгоритми машинного навчання можуть дати неточні або нерелевантні результати.

2.1. Основні етапи інтеграції даних

Інтеграція даних у попередній обробці передбачає об'єднання кількох різнорідних джерел даних у єдине сховище даних і єдине представлення даних. Його можна визначити як GSM (глобальна схема, джерело гетерогенної схеми

та відображення між джерелом і глобальною схемою). Він має в основному два підходи – підхід із жорстким зв’язком і підхід із слабким зв’язком [39].

Інтеграція даних є першим етапом попередньої обробки, спрямованої на об’єднання даних із різними джерелами в єдину структуру для подальшого аналізу. Процес інтеграції забезпечує придатність даних, усуває дублікати та вирішує проблеми різних форматів і структур даних. Для цього використовують математичні моделі та алгоритми, які дозволяють досягти високої якості інтеграції.

Розглянемо основні етапи інтеграції даних (рис. 2.3).

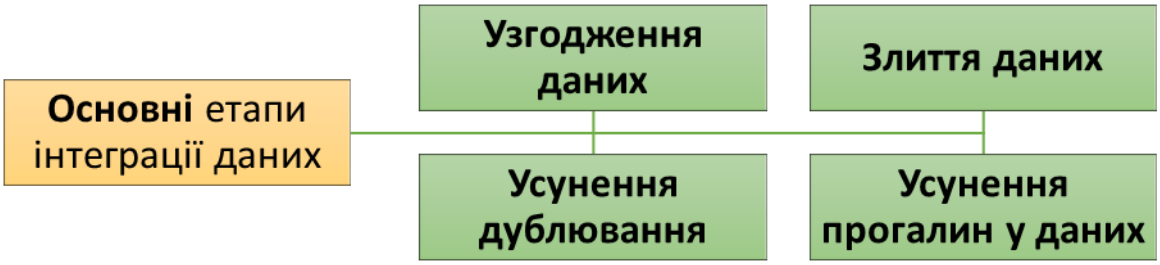

Рисунок 2.3 – Основні етапи інтеграції даних

Узгодження передбачає перетворення даних із різних джерел в єдиний формат. Для цього виконується перетворення даних:

$$X_{ij}^{*} = \frac{X_{ij} - \min(X_j)}{\max(X_j) - \min(X_j)}, \qquad (2.2)$$

де X_{ij} – значення *i*-го запису для *j*-го атрибута; $\min(X_j)$, $\max(X_j)$ – мінімальне та максимальне значення *j*-го атрибута; X_{ij}^{*} – нормалізоване значення.

Дублікати можуть виникати через одночасне внесення даних із різних джерел. Для їх виявлення застосування метрики схожості, наприклад, коефіцієнти Жаккара:

$$S(A,B) = \frac{|A \cap B|}{|A \cup B|}, \qquad (2.3)$$

де A, B – множини атрибутів двох записів; $S(A,B)$ – міра схожості між записами.

Якщо $S(A,B)>T$, де T – заданий поріг, знаєте, що записи є дубльованими.

Злиття даних вимагає об'єднання інформації з кількох джерел. Використовується агрегація значень за допомогою функцій середнього, суми або медіани. Наприклад, для обчислення середнього значення:

$$X_j^{mean} = \frac{1}{n}\sum_{i=1}^{n} X_{ij}, \qquad (2.4)$$

де n – кількість записів; X_j^{mean} – середнє значення для атрибута j.

Для заповнення пропущених значень можуть використовуватися статистичні методи або алгоритми машинного навчання. Один із підходів – заміна середнім значенням:

$$X_{ij} = \frac{\sum_{k=1}^{m} X_{kj}}{m}, \qquad (2.5)$$

де X_{ij} – заповнене значення для i-го запису та j-го атрибута; m – кількість наявних значень у стовпці j.

Загальний процес інтеграції можна представити у вигляді математичної моделі. Нехай задані k джерел даних $D_1, D_2, \ldots, D_k$, тоді інтегроване множина даних $D*$ визначається як:

$$D^* = \bigcup_{i=1}^{k} \hat{O}(D_i), \qquad (2.6)$$

де D_i – i-те джерело даних; $\Phi(D_i)$ – функція обробки даних із джерела D_i, яка включає нормалізацію, узгодження форматів, очищення та агрегацію.

Алгоритми інтеграції можуть обґрунтовуватися на:

- ✓ регресійних моделей для прогнозування відсутніх даних;
- ✓ Кластеризація для об'єднання схожих записів;
- ✓ Байєсових методів для врахування невизначеностей у джерелах.

Результатом інтеграції є набір даних, готовий до подальшого аналізу та моделювання. Ефективна інтеграція забезпечує точність і надійність моделей машинного навчання, що підтверджує значимість цього етапу у всьому процесі обробки даних.

2.3. Вплив алгоритмів обробки даних на точність і ефективність моделей

Для розуміння впливу алгоритмів попередньої обробки даних на точність і ефективність моделей машинного навчання необхідно провести аналіз наукових досліджень, в яких використовувалися різні методи обробки. Нижче наведено результати огляду ключових публікацій.

2.3.1. Використання нормалізації та стандартизації

У роботі [11] досліджено вплив нормалізації та стандартизації на точність моделей логістичної регресії, методів опорних векторів (SVM) та багатошарових нейронних мереж. Автори показали, що нормалізація даних суттєво покращує точність моделей, які чутливі до масштабу ознаки. Для SVM точність зросла на 15%, а для логістичної регресії – на 12%.

Таблиця 2.2 – Вплив нормалізації на точність моделей [11]

Модель	Без нормалізації (%)	З нормалізацією (%)
Логістична регресія	73	85
Метод опорних векторів	70	85
Нейронна мережа	78	90

Формула нормалізації, використана в роботі:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \qquad (2.7)$$

де x – значення ознаки, $\min(x)$, $\max(x)$ – мінімальне та максимальне значення ознаки.

2.3.2. Усунення пропусків у даних

У роботі [18] досліджено вплив на заповнення пропущених значень на моделі дерев рішень та Random Forest. Автори використовували такі методи заповнення:

- ✓ заповнення середнім значенням;
- ✓ інтерполяція;
- ✓ видалення записів із пропусками.

Найкращі результати показали заповнення середнім значенням. Точність Random Forest зросла з 80% до 88%.

2.3.3. Видалення аномальних значень

Робота [16] була присвячена дослідженню впливу видалення аномалій на точність моделі лінійної регресії. Автори використовували методи:

Z-оцінка: видалення записів, де $z>3$, за формулою:

$$z = \sigma x - \mu, \tag{2.8}$$

де x – значення ознаки; μ – середнє значення; σ – стандартне відхилення.

Після видалення аномалії середня абсолютна похибка (MAE) зменшилася на 20%.

2.3.4. Вибір ключових ознак

У публікації [49] проаналізовано методи вибору ознак, такі як методи головних компонентів (PCA) та відбір на основі ознак (важливість ознак) у Random Forest. Автори показали, що скорочення розмірності набору даних від 50 до 10 знаків за допомогою PCA скоротило час навчання моделей на 40%, зберігаючи точність на рівнях 92%.

Таблиця 2.3 – Вплив вибору ознак на ефективність моделей [49]

Метод	Кількість ознак	Час навчання (с)	Точність (%)
Без вибору ознак	50	12	92
PCA	10	7	92
Важливість функції	15	8	93

Аналіз наукових праць демонструє, що використання алгоритмів обробки даних суттєво підвищує як точність, так і ефективність моделей машинного навчання. Нормалізація даних є обов'язковою для моделей, чутливих до масштабу ознаки. Усунення пропусків і видалення аномалій підвищують стабільність і точність моделей. Вибір ключових ознак знижує обчислювальні витрати, зберігаючи або навіть покращуючи точність прогнозів. Ці результати показують на важливість використання алгоритмів показу як невід'ємної частини процесу машинного навчання.

2.4. Вибір та попередній аналіз даних для машинного навчання

Для дослідження впливу алгоритмів попередньої обробки даних на результати створення моделей машинного навчання нами використано набори даних із відкритих джерел. Набір даних [23], який буде використано для завдання прогнозування з використанням алгоритмів машинного навчання, містить різноманітну інформацію про студентів. Набір даних, використаний у дослідженні, є розширеною версією оригінального набору даних «Оцінки студентських іспитів» [1]. Він містить велику кількість стовпців і записів. Він також включає неточності в даних, такі як відсутні значення та неінформативні стовпці. Цей набір даних використовується для навчання моделей прогнозуванню оцінок студентів. Розміри набору даних становлять 30641 рядок на 14 стовпців. Цільовою змінною в цьому дослідженні, яка буде прогнозована, є результат іспиту студентів з математики.

Основні атрибути цього набору даних:

1. Gender – цей атрибут вказує на стать студентів (чоловічий чи жіночий);
2. Race/ethnicity – студенти поділяються на групи на основі їхньої расової чи етнічної приналежності, позначені як групи A, B, C, D, E;
3. Parental Education – цей атрибут представляє найвищий рівень освіти, досягнутий батьками учнів, із категоріями: «середня школа», «якийсь коледж», «ступінь молодшого спеціаліста», «ступінь бакалавра», «ступінь магістра»;
4. Lunch Type – цей атрибут описує тип обіду, який отримують учні, із варіантами «стандартний» або «безкоштовний/знижений»;
5. Test Preparation Course – вказує, чи завершив студент курс підготовки до тесту, з варіантами «завершено» або «жодного»;
6. Parent Marital Status – цей атрибут вказує на сімейний стан батьків (одружений, неодружений, вдівець, розлучений);
7. Practice Sport – частота занять студента спортом, яка може бути: «ніколи», «інколи» або «регулярно»;
8. Is First Child – вказує, чи є студент першою дитиною в сім’ї (так/ні);
9. Number of Siblings – кількість братів і сестер у студента (від 0 до 7);
10. Transport Means – тип транспорту, яким користується студент, щоб дістатися до місця навчання («шкільний автобус» або «приватний»);
11. Weekly Study Hours – кількість годин самостійної підготовки студента протягом тижня;
12. Math Score – оцінка, яку студент отримав на частині іспиту з математики;
13. Reading Score – оцінка, яку студент отримав на частині іспиту з читання;
14. Writing Score – оцінка, яку студент отримав на письмовій частині іспиту.

Таблиця 2.4 – Фрагмент використаного набору даних

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	MathScore	ReadingScore	WritingScore
0	0	female	group B	bachelor's degree	standard	none	72	72	74
1	1	female	group C	some college	standard	completed	69	90	88
2	2	female	group B	master's degree	standard	none	90	95	93
3	3	male	group A	associate's degree	free/reduced	none	47	57	44
4	4	male	group C	some college	standard	none	76	78	75
...	...	...	...	...	...	...	...	...	...
30636	995	male	group C	some high school	standard	none	56	47	51
30637	996	male	group E	associate's degree	free/reduced	none	74	75	72
30638	997	male	group C	some college	standard	none	36	29	27
30639	998	male	group A	some high school	free/reduced	completed	43	34	39
30640	999	female	group D	associate's degree	standard	none	52	68	66

30641 rows × 9 columns

Більшість змінних є категоричними. Це означає, що в процесі підготовки даних їх потрібно буде оцифрувати для подальшого навчання моделей машинного навчання. Цей набір даних було створено, щоб вивчити зв’язок між демографічними характеристиками студентів, підготовкою та успішністю.

Набір даних дозволяє дослідникам даних досліджувати різні аспекти успішності студентів і розуміти, як демографічні чинники та підготовка впливають на результати іспитів. Його можна використовувати для таких завдань, як прогнозне моделювання, кластеризація та статистичний аналіз. Дослідники часто використовують цей набір даних, щоб досліджувати відмінності в успішності студентів на основі демографічних ознак і розробити розуміння факторів, які можуть покращити результати студентів. Набір даних є цінним для освітніх досліджень. Це чудовий ресурс для вивчення освітніх даних і проведення різноманітних аналізів.

На етапі вивчення даних корисно використовувати інструменти візуалізації, щоб краще зрозуміти завдання та шляхи вирішення проблеми. Аналіз стовпця етнічної приналежності за допомогою кругової діаграми відкриває уявлення про різноманітність студентської популяції та дозволяє досліджувати потенційні відмінності в академічній успішності, пов’язані з

расовою та етнічною приналежністю. За расою студенти розподіляються таким чином, як представлено на рис. 2.4.

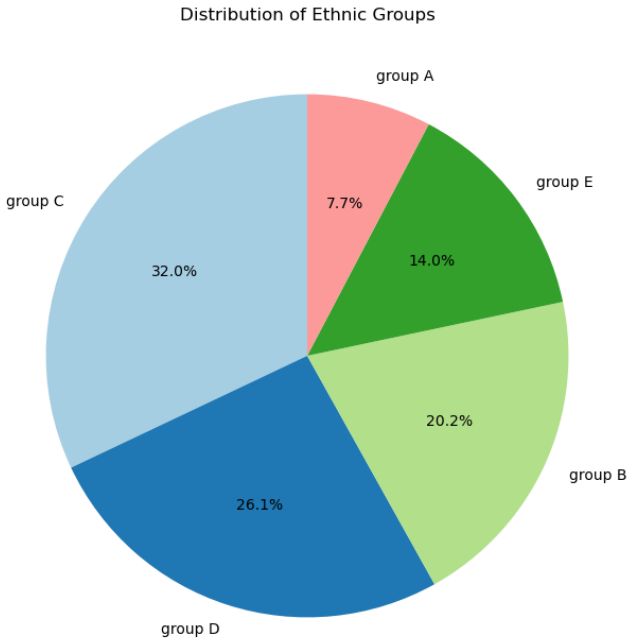

Рисунок 2.4 – Розподіл студентів за расою

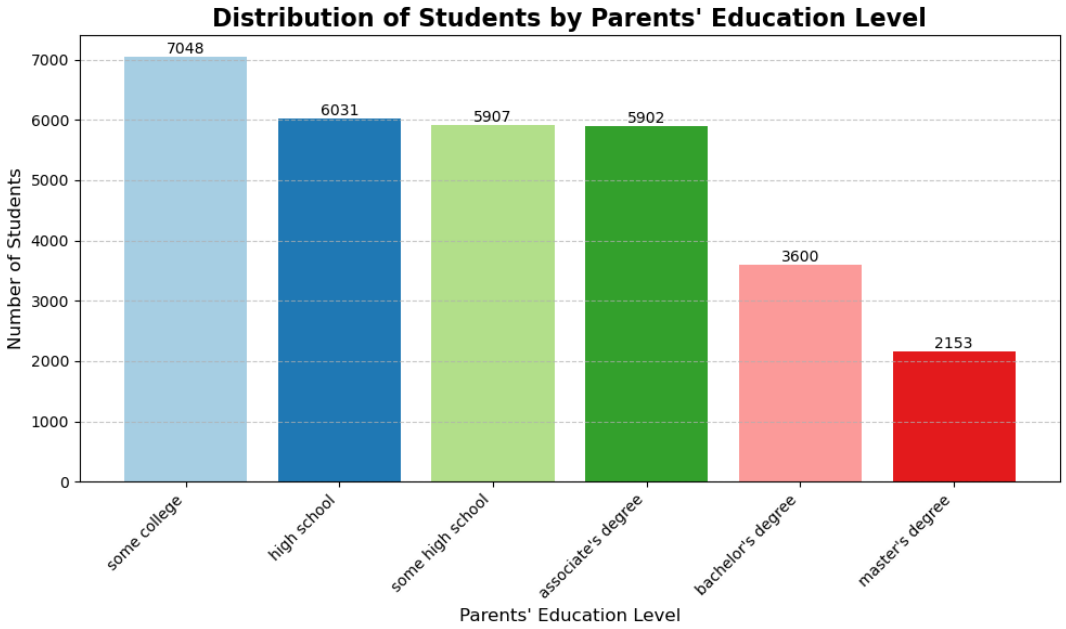

Рисунок 2.5 – Розподіл студентів за рівнем освіти батьків

Стовпець «ParentEduc» у наборі даних містить інформацію про найвищий рівень освіти, досягнутий батьками студентів у наборі даних. Аналіз цього стовпця за допомогою гістограми відкриває розуміння рівня освіти батьків

учнів та його потенційного впливу на успішність учнів. Бачимо, що менша частина батьків студентів мала вищу освіту.

Далі в процесі аналізу даних додаються 2 поля, які допомагають краще охарактеризувати загальну успішність студентів. Перше поле – загальний бал студента за всі 3 іспити, друге поле – відсоток загального балу, набраного від максимального. Максимальне значення для кожного іспиту становить 100. Варто зазначити, що набір даних має майже рівний розподіл студентів за статтю: 15424 жінки та 15217 чоловіки. Таким чином, ми можемо оцінити загальну середню успішність в залежності від різних величин, наприклад, статі.

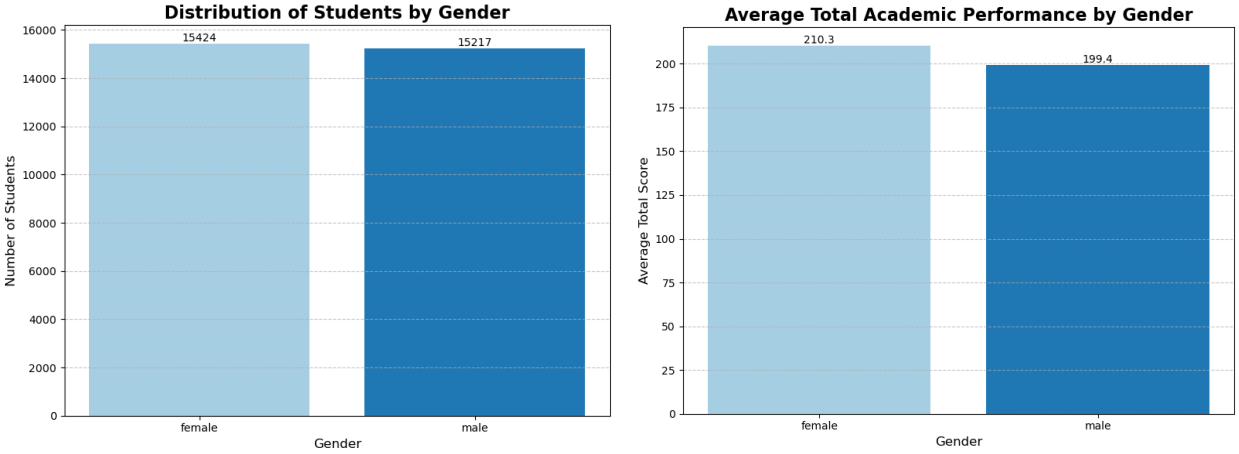

Рисунок 2.6 – Середня загальна академічна успішність залежно від статі

Використовуючи кореляційну матрицю, можна визначити, наскільки сильно чисельні змінні корелюють одна з одною. На основі матриці на рисунку 2.7 можна встановити, що всі типи тестів мають сильну позитивну кореляцію відносно один одного. Наприклад, це означає, що студенти, які добре склали письмовий іспит, також отримали хороший бал з математики. Від’ємні значення в матриці означають зворотний зв’язок двох змінних. У цих даних немає великих значень. Значення, близькі до 0, означають, що змінні погано корелюють один з одним, між ними немає прямого або зворотного зв’язку. Загальний бал і загальний відсоток, по суті, є одним показником, необхідним для аналізу даних, тому їх співвідношення дорівнює 1.

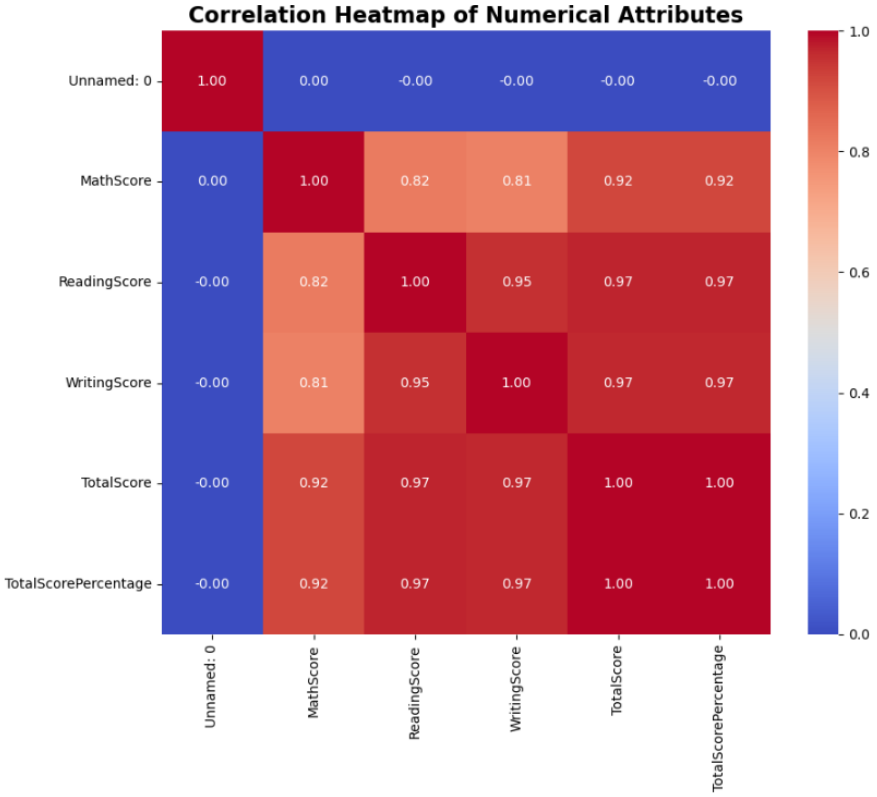

Рисунок 2.7 – Теплова карта кореляції

Далі в процесі аналізу даних також виявились деякі закономірності, які можуть вплинути на методи вирішення проблеми. Середній загальний бал серед усіх студентів склав 204 бали. Учні групи «E» були більш успішними з усіх видів тестів. Вони набрали в середньому на 18 балів більше з усіх предметів разом узятих. Рівень освіти батьків також впливає на успішність учнів. У середньому на 20 балів більше отримали студенти, батьки яких були магістрами. Студенти, які віддали перевагу стандартному обідньому набору, набрали на 29 балів більше, ніж студенти з безкоштовним обідом. Учні, які закінчили підготовчі курси, отримали на 20 балів більше за інших. Крім того, студенти, які в середньому навчалися більше годин на тиждень, мали дещо вищу успішність порівняно з іншими. Інші характеристики не мали такого великого впливу на успішність учнів.

Графічне представлення цих шаблонів показано на рисунку 2.8.

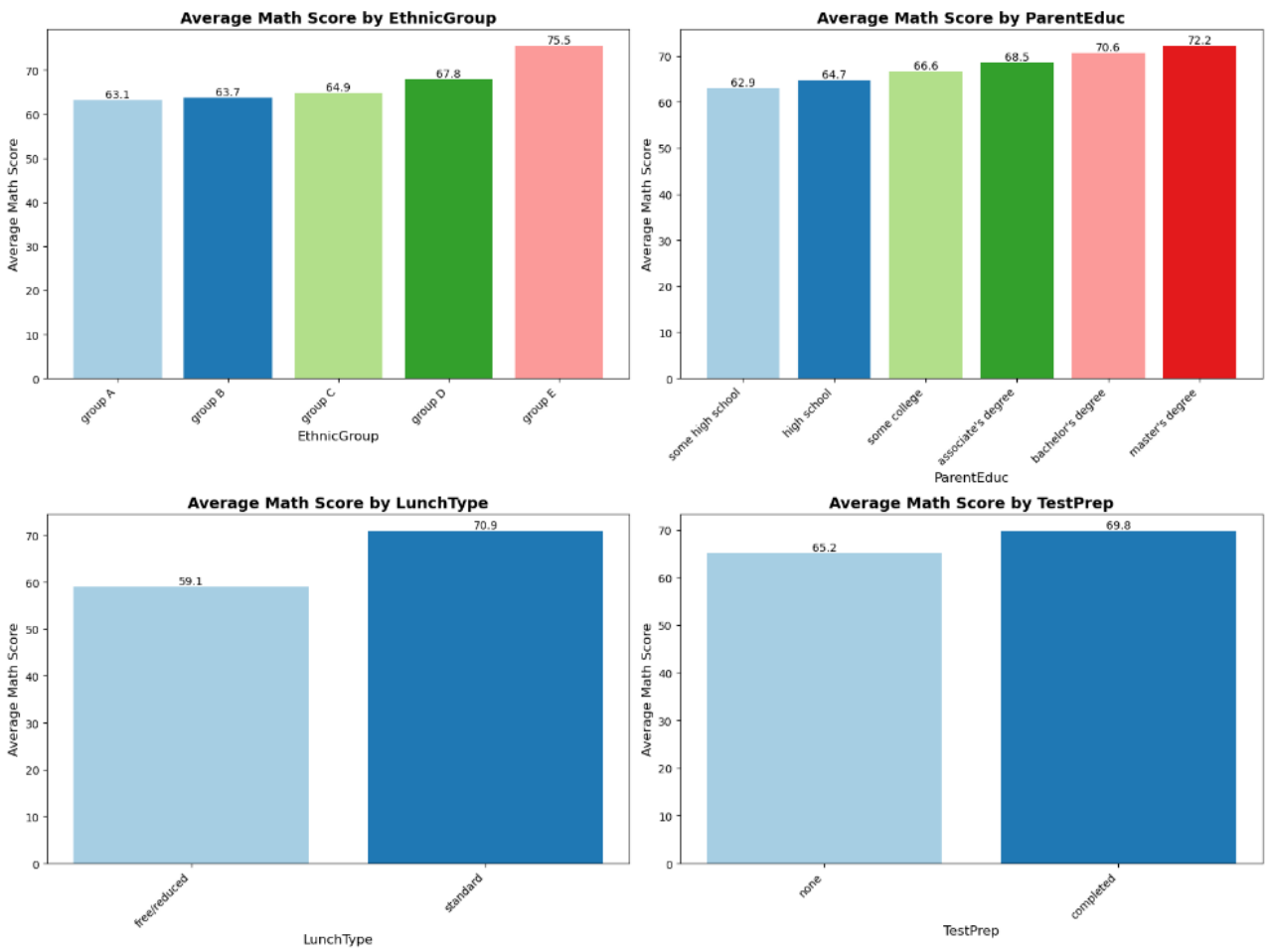

Рисунок 2.8 – Графіки змінних, які мали найбільший вплив на оцінку з математики

Аналізуючи гістограми [22] на рисунку 2.9 нижче, ми можемо помітити значну різницю в успішності чоловіків і жінок з різних предметів. Виходячи з інформації, можна зробити висновок, що чоловіки показують найкращі показники з математики, а жінки – на іспитах з письма та читання.

Далі необхідно оцінити якість наявних даних. Для цього необхідно перевірити дублювання даних, відсутні значення та проаналізувати викиди в даних. Оскільки ми не маємо унікального ідентифікатора студента, нам потрібно перевірити дані на наявність повторюваних рядків у всіх стовпцях. Дублювання даних не виявлено.

Далі, використовуючи інтерквартильний діапазон, було перевірено викиди в даних серед числових змінних. Для інтерквартильного діапазону була використана формула:

$$IQR = Q_3 - Q_1, \qquad (2.9)$$

де Q_3 – верхній (75-й) квартиль, що представляє значення, вище якого падає 25% даних, Q_1 – нижній (25-й) квартиль, що представляє значення, вище якого падає 75% даних.$Q_3 Q_1$

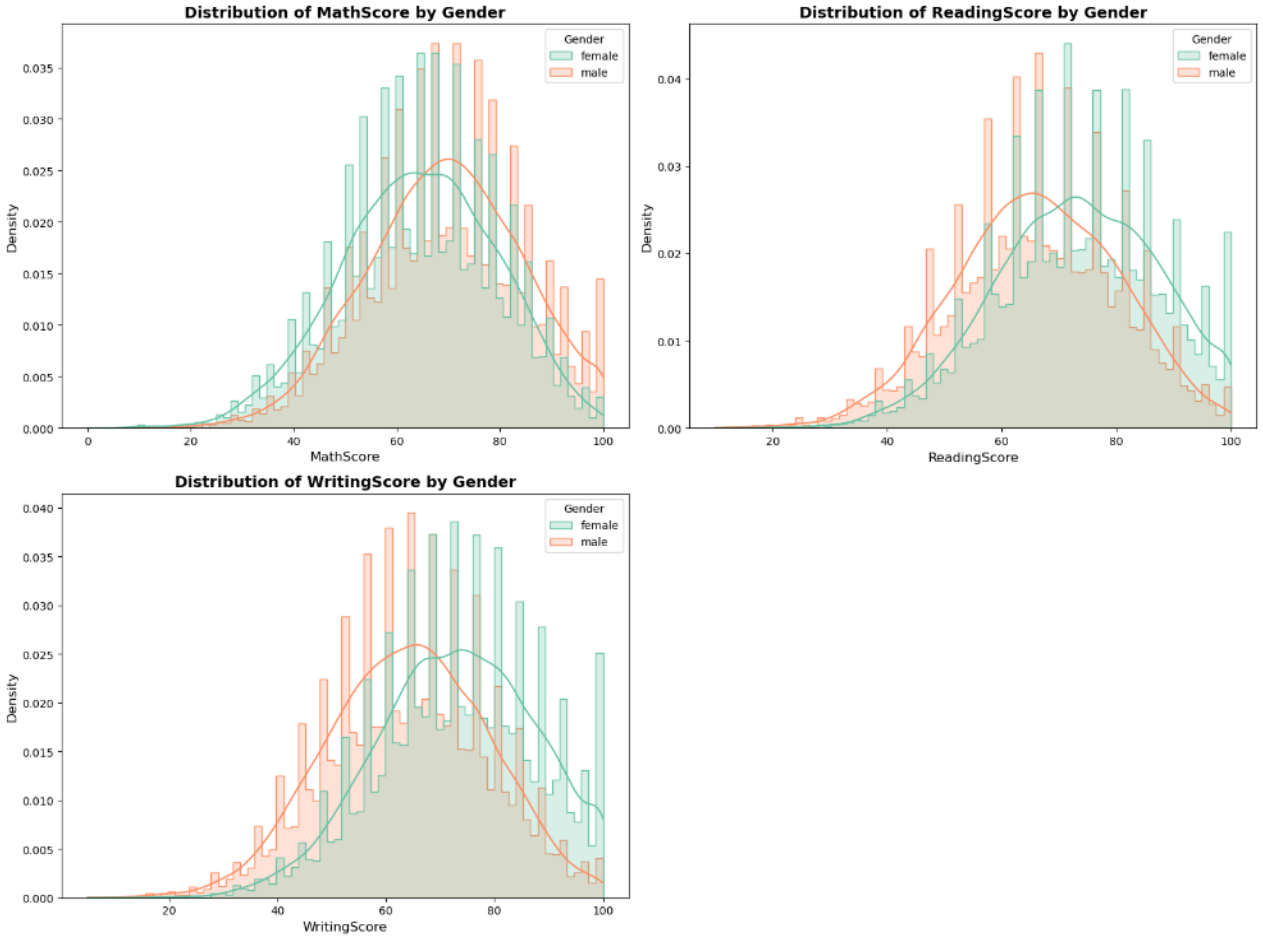

Рисунок 2.9 – Порівняння балів чоловіків і жінок на різних іспитах

Як правило, значення поза діапазоном $(Q_1 - 1{,}5 \cdot IQR,\ Q_3 + 1{,}5 \cdot IQR)$ вважаються викидами. Кількість викидів виявилася незначною – 90 рядків для оцінки читання та 106 рядків для оцінки письма. Також їх не можна назвати викидами, оскільки вони є результатами успішності студентів на цих видах іспитів.

У наборі даних усі дані були заповнені. Тобто пропущених даних не спостерігається (рис. 2.10).

```
[22]: # Стовпці для перевірки
columns_to_check = [
    'Gender', 'EthnicGroup', 'ParentEduc', 'LunchType', 'TestPrep',
    'MathScore', 'ReadingScore', 'WritingScore', 'TotalScore', 'TotalScorePercentage'
]

# Перевірка на наявність нульових значень у зазначених стовпцях
null_values = df[columns_to_check].isnull().sum()

# Створення таблиці з результатами
null_values_df = null_values.reset_index()
null_values_df.columns = ['Column', 'Null Values']

# Виведення таблиці
print(null_values_df)
```

```
                Column  Null Values
0               Gender            0
1          EthnicGroup            0
2           ParentEduc            0
3            LunchType            0
4             TestPrep            0
5            MathScore            0
6         ReadingScore            0
7         WritingScore            0
8           TotalScore            0
9  TotalScorePercentage            0
```

Рисунок 2.10 – Виявлення нульових значень за стовпцями df

Далі в процесі підготовки даних необхідно перевести категоріальні змінні в числову форму. Для цього кожній категорії ми присвоїли унікальне числове значення. Наприклад, поле зі статтю студента вказується так –1 - чоловіча, 2 - жіноча. Інші поля оброблені таким же чином (рис. 2.11).

```
[23]: from sklearn.preprocessing import LabelEncoder

# Стовпці для кодування
columns_to_encode = ['Gender', 'EthnicGroup', 'ParentEduc', 'LunchType', 'TestPrep']

# Перетворення категоріальних змінних у числові
label_encoders = {}
for column in columns_to_encode:
    label_encoders[column] = LabelEncoder()
    df[column] = label_encoders[column].fit_transform(df[column])

# Перегляд оновленого DataFrame
df.head()
```

[23]:

	Unnamed: 0	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	MathScore	ReadingScore	WritingScore	TotalScore	TotalScorePercentage
0	0	0	1	1	1	1	72	72	74	218	72
1	1	0	2	4	1	0	69	90	88	247	82
2	2	0	1	3	1	1	90	95	93	278	92
3	3	1	0	0	0	1	47	57	44	148	49
4	4	1	2	4	1	1	76	78	75	229	76

Рисунок 2.11 – Результати переведення категоріальних змінних df в числову форму

Таким чином, у процесі аналізу даних за допомогою інструментів візуалізації вдалося виявити цікаві та корисні закономірності, а також проблеми в даних, які в майбутньому можуть допомогти у вирішенні певних завдань з цим набором даних.

РОЗДІЛ 3.
РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ВПЛИВУ АЛГОРИТМІВ ПОПЕРЕДНЬОЇ ОБРОБКИ ДАНИХ НА РЕЗУЛЬТАТИ СТВОРЕННЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ

3.1. Формування експерименту дослідження впливу алгоритмів попередньої обробки даних на результати створення моделей прогнозування

Для вирішення задачі прогнозування оцінок студентів з математики на основі наявних у них даних було відібрано окремі алгоритми машинного навчання із учителем. Таким чином, у процесі навчання моделей та оцінки їх результатів можна буде порівняти вплив різних методів обробки на точність різних моделей прогнозування. Це дослідження не приділяло великої уваги вибору параметрів моделі. Використовувалися параметри за замовчуванням, значення яких можна знайти в документації для цих алгоритмів. Перелік алгоритмів, що використовуються для аналізу даних та оцінки результатів:

- ✓ LGBMRRegressor [28];
- ✓ XGBRegressor [50];
- ✓ GradientBoostingRegressor [36];
- ✓ RandomForestRegressor [37].

Потім вхідний набір даних скопіювали кілька разів. На кожній із копій були проведені певні маніпуляції з даними, щоб потім оцінити ступінь впливу різних методів обробки даних на кінцевий результат. Оскільки у нас немає пропущених значень, то немає потреба заповнити ці значення.

У подальшому дослідженні пропонується розглядати наступні варіанти підготовки даних до навчання моделей прогнозуванню оцінки студентів з математики:

Набір даних 1 (df_no_outliers) – видалення рядків з аномаліями;

Набір даних 2 (df_replaced_outliers) – заміна аномалій середнім значенням;

Набір даних 3 (df_standardized) – стандартизація значень;

Набір даних 4 (df_standardized) – видалено стовпці «Gender», «ParentEduc», «TestPrep», які за результатами аналізу не мали сильного впливу на результати студентів на іспиті з математики.

Далі набори даних були розділені на набори для навчання та прогнозування у співвідношенні 80% до 20%. Усі типи моделей були навчені на цих наборах, а результати були введені в загальну таблицю для подальшого аналізу.

Для оцінки впливу методів обробки даних на зміну навантаження на обчислювальну систему під час навчання моделей було виміряно навантаження на центральний процесор (CPU). Бібліотека psutil дозволяє вимірювати поточне використання CPU, пам’яті та інших системних ресурсів. Це ефективний спосіб для моніторингу обчислювальної системи під час виконання коду. Навчальний процес проводився на локальній машині з процесором Intel(R) Xeon(R) CPU E3-1230 V2 з тактовою частотою 3.30 ГГц, який має 4 фізичних ядра та підтримує технологію Hyper-Threading, що дозволяє використовувати 8 потоків.

Для оцінки точності прогнозів використовувалися наступні показники. Середня абсолютна похибка (MAE) [21] – вимірює середню абсолютну різницю між фактичними та прогнозованими значеннями:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \qquad (3.1)$$

де y_i – фактичне значення; $\hat{y}_i$ – передбачене значення; n – кількість спостережень.

Квадратний корінь середньої квадратичної помилки (RMSE) [21] – також вимірює різницю між фактичними та прогнозованими значеннями, але більші похибки штрафуються більше, ніж MAE:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \qquad (3.2)$$

де y_i – фактичне значення; $\hat{y}_i$ – передбачене значення; n – кількість спостережень.

Коефіцієнт детермінації (R^2) [34] – вимірює частку дисперсії в залежній змінній, що пояснюється моделлю. Він коливається від 0 до 1, де 1 означає, що модель ідеально відповідає даним, а 0 означає, що модель нічого не пояснює:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \quad (3.3)$$

де y_i – фактичне значення; $\hat{y}_i$ – передбачене значення; $\bar{y}$ – середнє значення фактичних даних; n – кількість спостережень.

3.2. Попередня обробка даних для машинного навчання моделей

Насамперед нами створено код, який виконує перетворення категоріальних змінних у числові за допомогою методу LabelEncoder з бібліотеки sklearn (рис. 3.1).

```
from sklearn.preprocessing import LabelEncoder

# Стовпці для кодування
columns_to_encode = ['Gender', 'EthnicGroup', 'ParentEduc', 'LunchType', 'TestPrep']

# Перетворення категоріальних змінних у числові
label_encoders = {}
for column in columns_to_encode:
    label_encoders[column] = LabelEncoder()
    df[column] = label_encoders[column].fit_transform(df[column])

# Перегляд оновленого DataFrame
df.head()
```

Рисунок 3.1 – Фрагмент коду перетворення категоріальних змінних у числові

Цей процес дозволяє використовувати такі змінні у моделях машинного навчання, які працюють виключно з числовими даними.

При цьому було перетворено наступні стовпці:

- ✓ Gender – стать студента;
- ✓ EthnicGroup – етнічна приналежність студента;
- ✓ ParentEduc – рівень освіти батьків;
- ✓ LunchType – тип обіду;
- ✓ TestPrep – інформація про проходження курсу підготовки до тесту.

Для кожного стовпця створюється окремий енкодер, який прив'язується до назви стовпця у словнику label_encoders. Для кожного значення в стовпці категорії перетворюються на унікальні числові мітки. Кожен зазначений стовпець замінюється його закодованими значеннями.

У DataFrame (df) тепер замість текстових категорій у зазначених стовпцях представлені числові значення (рис. 3.2).

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	MathScore	ReadingScore	WritingScore
0	0	1	1	1	1	72	72	74
1	0	2	4	1	0	69	90	88
2	0	1	3	1	1	90	95	93
3	1	0	0	0	1	47	57	44
4	1	2	4	1	1	76	78	75

Рисунок 3.2 – Фрагмент створеного df після перетворення категоріальних змінних у числові

У стовпці Gender категорія female закодована як 0, а male – як 1. У стовпці EthnicGroup категорії, наприклад, group A, group B, group C відповідно закодовані як 0, 1, 2. Більшість алгоритмів машинного навчання працюють лише з числовими даними. Закодовані дані легко порівнювати й аналізувати. Це перетворення є важливим кроком у попередній обробці даних перед навчанням моделей.

У подальшому нами створено код, що виконує виявлення аномалій у числових колонках DataFrame (df) з використанням методу міжквартильного розмаху (IQR) (рис. 3.3). Аномалії визначаються як значення, що знаходяться за

межами діапазону, визначеного нижньою та верхньою межами, розрахованими на основі IQR.

```
# Виявлення аномалій за допомогою методу IQR для числових колонок
numerical_columns = df.select_dtypes(include=['number']).columns
outliers = {}

for column in numerical_columns:
    Q1 = df[column].quantile(0.25)  # Перший квартиль
    Q3 = df[column].quantile(0.75)  # Третій квартиль
    IQR = Q3 - Q1  # Міжквартильний розмах

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Виявлення аномалій у стовпці
    outliers[column] = df[(df[column] < lower_bound) | (df[column] > upper_bound)]

# Об'єднання всіх аномалій в один DataFrame для перегляду
outliers_df = pd.concat(outliers.values(), keys=outliers.keys())

# Перегляд виявлених аномалій
outliers_df
```

Рисунок 3.3 – Фрагмент коду виявлення аномалій за допомогою методу IQR для числових колонок

Насамперед було вибрано всі числові колонки в DataFrame для аналізу. Назви цих колонок зберігаються у змінній numerical_columns. Після цього передбачено розрахунок IQR для кожної колонки. Зокрема, для кожної колонки обчислюється:

- ✓ перший квартиль (Q1) – значення, нижче якого знаходиться 25% даних;
- ✓ третій квартиль (Q3) – значення, нижче якого знаходиться 75% даних;
- ✓ міжквартильний розмах (IQR) – різниця між Q3 та Q1.

Після цього виконується розрахунок меж для визначення аномалій. Нижня межа – значення менші за цю межу вважаються аномаліями, верхня межа – значення більші за цю межу також вважаються аномаліями. Для кожної числової колонки аномалії зберігаються у словнику outliers, де ключем є назва колонки, а значенням – DataFrame із відповідними аномаліями.

Наступний крок забезпечує об’єднання аномалій у загальний DataFrame під назвою outliers_df (рис. 3.4).

		Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	MathScore	ReadingScore	WritingScore
MathScore	**17**	0	1	5	0	1	18	32	28
	59	0	2	5	0	1	0	17	10
	145	0	2	4	0	1	22	39	33
	787	0	1	4	1	1	19	38	32
	980	0	1	2	0	1	8	24	23
...	...	...	...	...	...	...	...	...	...
WritingScore	**28704**	1	2	5	0	1	28	10	13
	29283	1	1	4	1	1	42	24	21
	30267	1	2	0	0	1	27	26	23
	30406	1	1	0	0	1	43	28	24
	30589	1	0	4	0	1	22	19	22

295 rows × 8 columns

Рисунок 3.4 – Фрагмент створеного outliers_df після виявлення аномалій за допомогою методу IQR для числових колонок

Всі виявлені аномалії з різних колонок об’єднуються у новий DataFrame outliers_df, який дозволяє переглядати аномальні значення для всіх колонок одночасно. Кінцевий DataFrame outliers_df містить усі аномалії, виявлені у числових колонках, із зазначенням їхніх відповідних колонок і значень.

Переваги методу стосуються простоти обчислень із використанням IQR-методу, що дозволяє легко та ефективно виявляти аномалії. Метод підходить для даних, які не обов’язково мають нормальний розподіл. Цей підхід забезпечує ідентифікацію аномальних значень, які можуть впливати на результати моделювання, дозволяючи їх подальшу обробку.

У подальшому нами створено код, що дозволяє обробляти аномалії у числових даних двома різними способами:

- ✓ видалення рядків із аномаліями;
- ✓ заміна аномалій середнім значенням відповідної колонки.

При цьому проводиться розрахунок меж для визначення аномалій. Видалення рядків із аномаліями забезпечує у DataFrame збереження df_no_outliers тільки тих рядків, значення колонок яких знаходяться в межах від lower_bound до upper_bound.

```python
# Видалення та заміна аномалій
for column in numerical_columns:
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Видалення рядків з аномаліями
    df_no_outliers = df_no_outliers[(df_no_outliers[column] >= lower_bound) & (df_no_outliers[column] <= upper_bound)]

    # Заміна аномалій середнім значенням
    mean_value = df[column].mean()

    # Перетворення середнього значення до типу колонки
    if pd.api.types.is_integer_dtype(df[column]):
        mean_value = int(mean_value)

    df_replaced_outliers.loc[(df[column] < lower_bound) | (df[column] > upper_bound), column] = mean_value

# Перегляд результатів
print("DataFrame Without Outliers:")
df_no_outliers.head()

print("\nDataFrame with Outliers Replaced by Mean:")
df_replaced_outliers.head()
```

Рисунок 3.5 – Фрагмент коду видалення та заміна аномалій

Заміна аномалій середнім значенням передбачає із DataFrame створення df_replaced_outliers, де аномальні значення замінюються середнім значенням колонки.

Результати двох методів обробки аномалій можна переглянути за допомогою функції head() і вони представлені на рис. 3.6.

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	MathScore	ReadingScore	WritingScore
0	0	1	1	1	1	72	72	74
1	0	2	4	1	0	69	90	88
2	0	1	3	1	1	90	95	93
3	1	0	0	0	1	47	57	44
4	1	2	4	1	1	76	78	75

Рисунок 3.6 – Фрагмент створеного масиву даних після видалення та заміни аномалій

Код враховує типи даних колонок, щоб зберігати цілісність DataFrame. Цей підхід забезпечує ефективну обробку аномалій для покращення якості даних перед моделюванням.

Нами написано код, що виконує стандартизацію числових колонок у DataFrame (df) за допомогою методу StandardScaler із бібліотеки sklearn (рис. 3.7). Стандартизація змінює розподіл даних таким чином, що кожна змінна матиме середнє значення 0 та стандартне відхилення 1. Це покращує ефективність моделей машинного навчання, які чутливі до масштабу ознак.

```
from sklearn.preprocessing import StandardScaler

# Вибір числових колонок для стандартизації
numerical_columns = df.select_dtypes(include=['number']).columns

# Створення об'єкта стандартизатора
scaler_standardization = StandardScaler()

# Копія DataFrame для збереження результатів
df_standardized = df.copy()

# Стандартизація
df_standardized[numerical_columns] = scaler_standardization.fit_transform(df[numerical_columns])

# Перегляд результатів
print("Standardized DataFrame:")
df_standardized.head()
```

Рисунок 3.7 – Фрагмент коду виконання стандартизації

Насамперед виконується вибір числових колонок. Де всі колонки числового типу вибираються у DataFrame для подальшої стандартизації. Назви таких колонок зберігаються у змінній numerical_columns.

Об’єкт StandardScaler забезпечує обчислення середнього значення та стандартного відхилення для кожної колонки, а також їх трансформацію у стандартизовані значення.

У подальшому створюється копія DataFrame для збереження результатів стандартизації, щоб вихідний DataFrame залишився незмінним. Виконання стандартизації здійснюється завдяки fit_transform, де fit обчислює середнє значення та стандартне відхилення для кожної числової колонки, а transform застосовує ці обчислення, перетворюючи значення в колонках таким чином, щоб кожна змінна мала середнє значення 0 і стандартне відхилення 1.

У результати отримуємо фрагмент створеного масиву даних після виконання стандартизації (рис. 3.8).

	Gender	EthnicGroup	ParentEduc	LunchType	TestPrep	MathScore	ReadingScore	WritingScore
0	-0.993267	-1.040275	-0.891327	0.734413	0.725850	0.345305	0.161882	0.361369
1	-0.993267	-0.162005	0.773750	0.734413	-1.377695	0.148012	1.388764	1.275949
2	-0.993267	-1.040275	0.218724	0.734413	0.725850	1.529064	1.729565	1.602585
3	1.006779	-1.918545	-1.446353	-1.361632	0.725850	-1.298804	-0.860520	-1.598447
4	1.006779	-0.162005	0.773750	0.734413	0.725850	0.608363	0.570843	0.426696

Рисунок 3.8 – Фрагмент створеного масиву даних після виконання стандартизації

Оновлений DataFrame df_standardized містить стандартизовані числові значення. Наприклад, значення в колонці MathScore до стандартизації могли виглядати так:

Стандартизація запобігає домінуванню змінних із великим масштабом у моделюванні. Алгоритми, такі як логістична регресія чи градієнтний бустинг, працюють краще зі стандартизованими даними. Стандартизовані дані підходять для більшості алгоритмів машинного навчання. Написаний код є важливим етапом попередньої обробки даних, який дозволяє забезпечити ефективність і точність моделей машинного навчання.

3.3. Результати створення моделей машинного навчання за використання різних алгоритмів попередньої обробки даних

Нами написано код спрямований на порівняння продуктивності різних моделей регресії для прогнозування оцінок студентів із математики (MathScore) на різних наборах даних. Він також оцінює вплив методів попередньої обробки даних на точність, ефективність та навантаження на обчислювальну систему.

Насамперед було здійснено підготовку наборів даних. Для аналізу були використані чотири набори даних, які оброблялися різними способами:

1. df_no_outliers – дані, з яких видалено рядки з аномаліями.
2. df_replaced_outliers – дані, в яких аномалії замінено середнім значенням.
3. df_standardized – дані зі стандартизованими значеннями.
4. Reduced Columns – стандартизовані дані без колонок Gender, ParentEduc, TestPrep.

Наведений нижче фрагмент коду (рис. 3.1) створює словник із назвами наборів даних та їх відповідними об’єктами.

```
# Datasets
datasets = {
    "No Outliers": df_no_outliers,
    "Replaced Outliers": df_replaced_outliers,
    "Standardized": df_standardized,
    "Reduced Columns": df_standardized.drop(columns=["Gender", "ParentEduc", "TestPrep"], errors="ignore")
}
```

Рисунок 3.9 – Фрагмент коду створення словника із назвами наборів даних та їх відповідними об’єктами

Цей фрагмент створює словник із назвами наборів даних та їх відповідними об’єктами. Ключі визначають тип підготовки, який застосовувався до кожного набору.

У подальшому виконується навчання моделей. Цільовий стовпець у кожному наборі даних – MathScore. Дані були поділені на навчальний і тестовий набори в співвідношенні 80% до 20% (рис. 3.10).

```
# Loop through datasets and models
for dataset_name, dataset in datasets.items():
    X = dataset.drop(columns=[target], errors="ignore")
    y = dataset[target]

    # Split into train and test sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Рисунок 3.10 – Фрагмент коду поділу df на навчальний і тестовий набори даних

Цей фрагмент (рис. 3.10) створює матриці ознак (X) та цільовий стовпець (y) для навчання та тестування моделей.

Було використано чотири моделі для навчання:

- ✓ LGBMRegressor – із параметром force_row_wise=True для явного вибору схеми потоків;
- ✓ XGBRegressor – із метрикою rmse;
- ✓ GradientBoostingRegressor – класична модель градієнтного бустингу;
- ✓ RandomForestRegressor – ансамблева модель на основі дерев рішень.

```
# Models
models = {
    "LGBMRegressor": LGBMRegressor(force_row_wise=True),  # Explicit row-wise training
    "XGBRegressor": XGBRegressor(eval_metric="rmse"),
    "GradientBoostingRegressor": GradientBoostingRegressor(),
    "RandomForestRegressor": RandomForestRegressor()
}
```

Рисунок 3.11 – Фрагмент коду опису вибраних моделей для навчання

Представлений код (рис. 3.11) визначає перелік моделей, які навчаються для кожного набору даних.

У подальшому виконується оцінка обчислювальних витрат. Кожна модель навчається з фіксацією часу навчання та навантаження на CPU (рис. 3.12).

```
for model_name, model in models.items():
    # Record CPU usage
    cpu_start = psutil.cpu_percent(interval=None)
    start_time = time.time()

    # Train the model
    model.fit(X_train, y_train)

    # Record CPU usage and time after training
    cpu_end = psutil.cpu_percent(interval=None)
    training_time = time.time() - start_time
    cpu_usage = cpu_end - cpu_start

    # Predict
    y_pred = model.predict(X_test)
```

Рисунок 3.12 – Фрагмент коду для фіксації часу навчання та навантаження на CPU

У написаному коді використано:

- ✓ psutil.cpu_percent – вимірює завантаження CPU;
- ✓ time.time() – визначає час початку та завершення навчання.

Результати зберігаються як training_time (секунди) та cpu_usage (відсотки).

У подальшому виконується оцінка точності моделей. Метрики точності обчислювалися на тестовій вибірці:

- ✓ MAE – середня абсолютна похибка;
- ✓ RMSE – квадратний корінь середньої квадратичної похибки;
- ✓ R^2 – коефіцієнт детермінації.

```
# Metrics
mae = mean_absolute_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)  # Updated for compatibility
r2 = r2_score(y_test, y_pred)

# Store results
results.append({
    "Dataset": dataset_name,
    "Model": model_name,
    "MAE": mae,
    "RMSE": rmse,
    "R2": r2,
    "CPU Usage (%)": cpu_usage,
    "Training Time (s)": training_time
})
```

Рисунок 3.13 – Фрагмент коду для оцінки точності моделей

Цей фрагмент коду (рис. 3.13) обчислює метрики точності. У подальшому отримані результати зберігаються у вигляді словника, який конвертується у DataFrame. Кожен запис у словнику містить:

- ✓ назву набору даних;
- ✓ назву моделі;
- ✓ метрики точності (MAE, RMSE, R^2);
- ✓ завантаження CPU;
- ✓ час навчання.

Написаний код дозволяє автоматизувати навчання моделей та оцінку їх продуктивності на різних наборах даних. Результати, такі як точність моделей, обчислювальні витрати та ефективність методів обробки даних, можна використовувати для порівняння та аналізу.

3.4. Результати навчання моделей прогнозування оцінок студентів з математики за використання різних алгоритмів попередньої обробки даних

Нами отримано результати навчання моделей машинного навчання для прогнозування оцінки студентів з математики (MathScore) із використанням різних алгоритмів обробки даних. Навчання виконувалося на чотирьох наборах даних, підготовлених за різними підходами, що описано вище.

Нами виконано графічне представлення результатів у вигляді діаграми зміни коефіцієнта детермінації (R^2) для кожної моделі та набору даних (рис. 3.14).

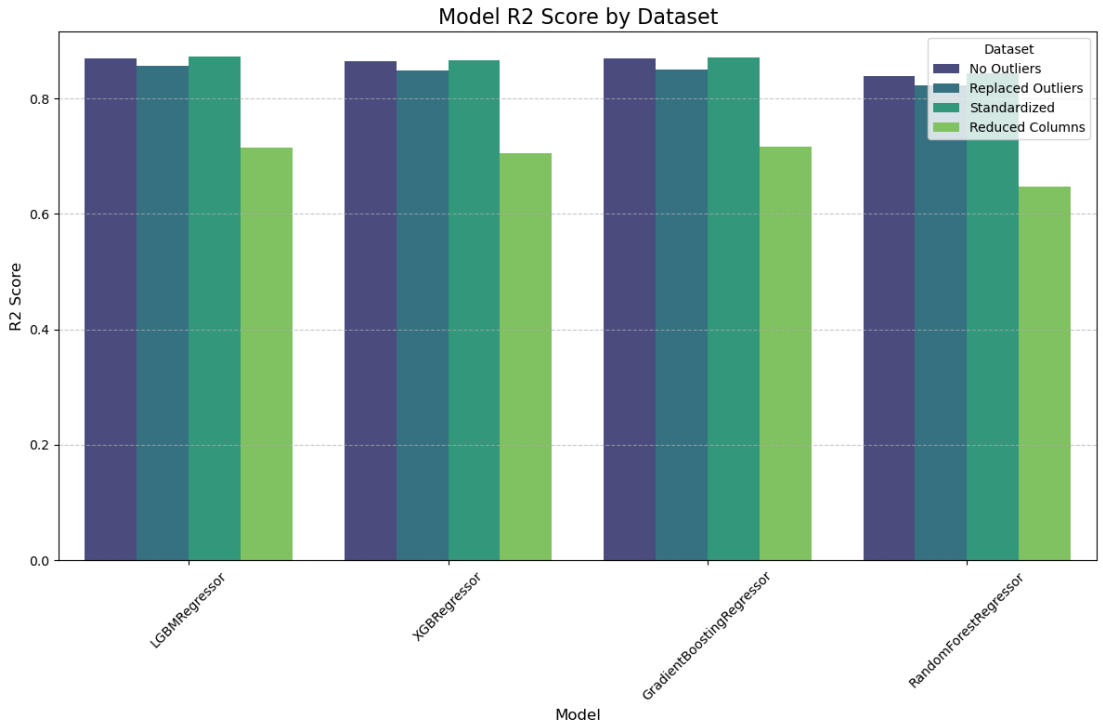

Рисунок 3.14 – Діаграма зміни коефіцієнта детермінації (R^2) для кожної моделі та набору даних

На рис. 3.14 зображено коефіцієнт детермінації (R^2) для кожної моделі та набору даних. Як видно з графіку, модель LightGBM показала найвищі значення R^2 на всіх наборах даних, а найменш продуктивною виявилася RandomForestRegressor. Серед наборів даних, стандартизація (Standardized) сприяла покращенню точності всіх моделей, а видалення маловпливових колонок (Reduced Columns) дало змогу скоротити час навчання без значного зниження точності.

На рисунку 3.15 показано навантаження на центральний процесор (CPU) під час навчання моделей.

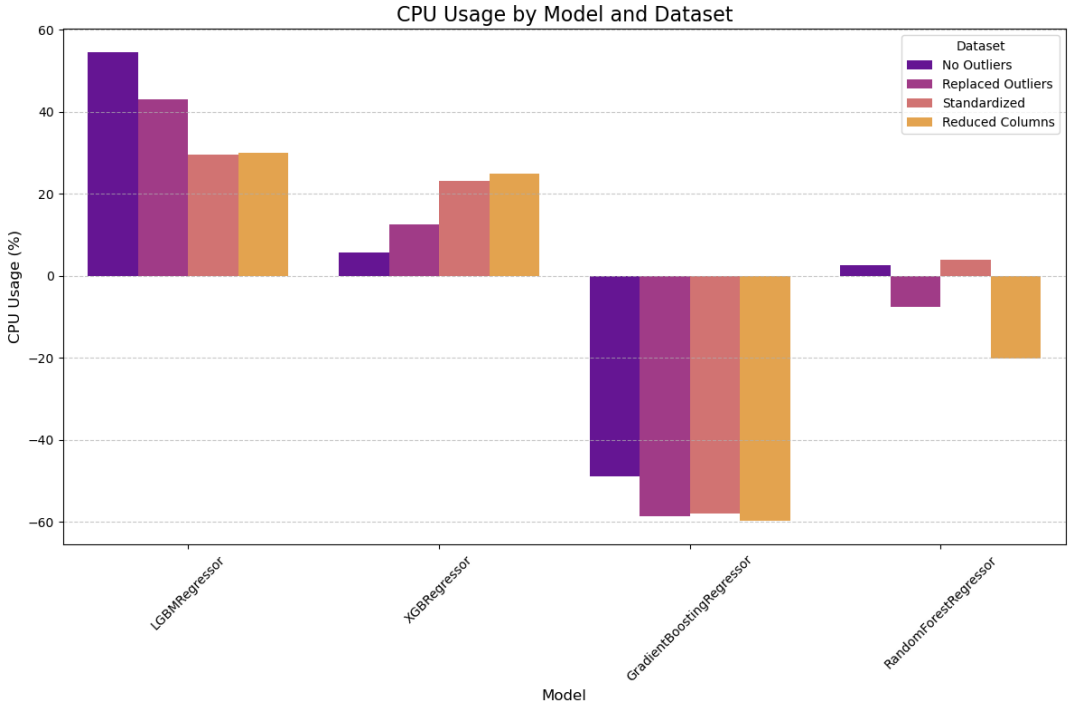

Рисунок 3.15 – Діаграма зміни навантаження на центральний процесор (CPU) під час навчання моделей

Алгоритм LGBMRRegressor продемонстрував найбільше завантаження CPU, тоді як моделі XGBRegressor та RandomForestRegressor були більш оптимальними у використанні обчислювальних ресурсів. Значення CPU Usage (%) є як позитивними, так і негативними, що вказує на динамічну зміну навантаження системи. Позитивне значення CPU Usage (%) означає, що під час навчання моделі завантаження процесора зросло порівняно з початковим станом. Це свідчить про активне використання ресурсів CPU моделлю.

Негативне значення CPU Usage (%) вказує, що завантаження процесора зменшилося під час виконання задачі порівняно з початковим станом. Це може статися через зниження активності інших процесів у системі.

На рис. 3.16 представлено час навчання моделей. Алгоритм LGBMRRegressor показав найшвидше навчання, навіть на великих наборах даних. Виключення маловпливових колонок суттєво скоротило час навчання для всіх моделей.

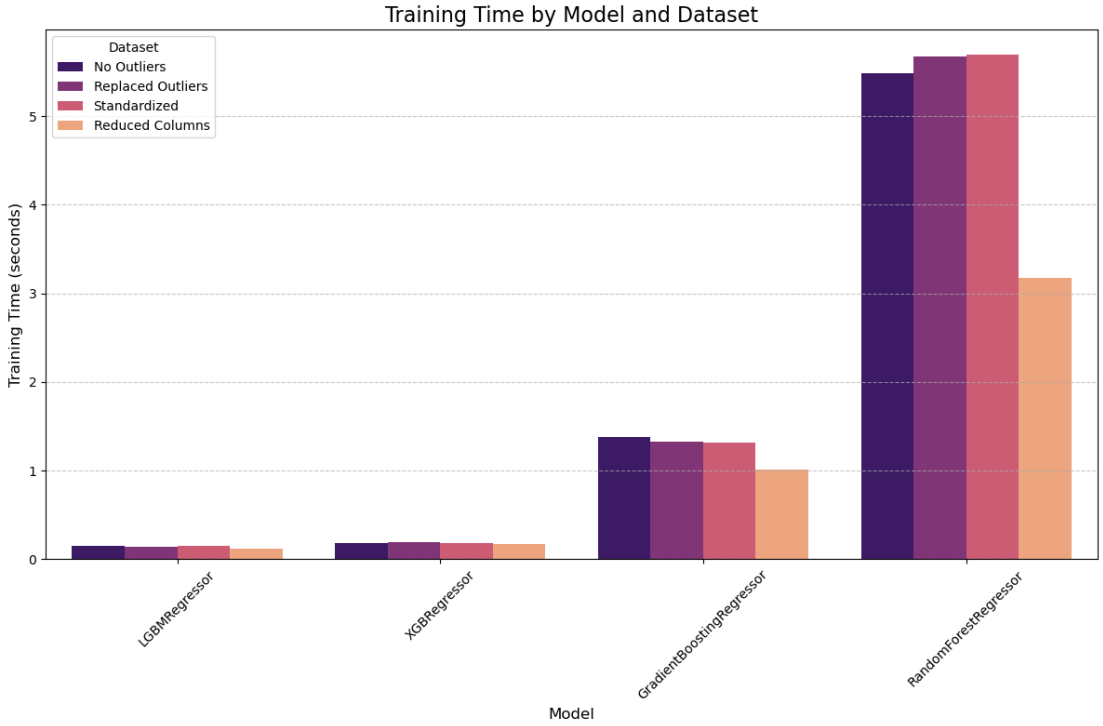

Рисунок 3.16 – Діаграма зміни часу навчання моделей

Результати дослідження показали, що методи попередньої обробки даних мають суттєвий вплив на точність та ефективність моделей. Найкращі результати точності були досягнуті на стандартизованих даних, тоді як видалення колонок із малим впливом сприяло значному зменшенню часу навчання без значного зниження точності прогнозу. Алгоритм LGBMRRegressor виявився найбільш ефективним за всіма критеріями, забезпечуючи високу точність, швидке навчання та низьке навантаження на обчислювальну систему.

РОЗДІЛ 4.
ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Аналіз небезпечних чинників під час виконання машинного навчання моделей

Процес машинного навчання моделей передбачає виконання значної кількості обчислень, які можуть вплинути на стабільність роботи обчислювальної системи та спричинити небезпеки для цілісності даних, безпеки результатів і стабільності системи. Нами проаналізовано основні небезпечні чинники, які виникають під час виконання машинного навчання, та запропоновано способи їхньої мінімізації.

Основні небезпечні чинники представлено на рис. 4.1.

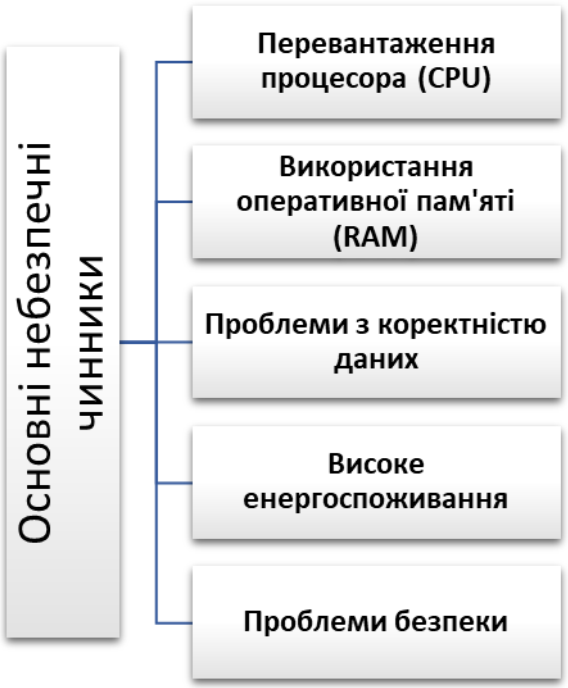

Рисунок 4.1 – Основні небезпечні чинники під час виконання машинного навчання моделей

Під час навчання моделей, особливо на великих наборах даних, спостерігається значне завантаження CPU. Це може призвести до зниження

Таблиця 4.1 – Небезпечні чинники під час машинного навчання моделей з позиції охорони праці

№	Чинник	Опис	Можливі наслідки	Рекомендації
1	Перевантаження CPU	Тривала робота CPU на максимальному завантаженні.	Перегрів обладнання, ризик займання, вихід із ладу.	Забезпечення охолодження, моніторинг температури.
2	Використання великого обсягу RAM	Надмірне використання оперативної пам’яті.	Збої в роботі системи, підвищене енергоспоживання.	Оптимізація задач, використання серверних рішень.
3	Нестабільність електропостачання	Інтенсивна робота обладнання збільшує навантаження на електромережу.	Перепади напруги, коротке замикання, аварійні ситуації.	Використання стабілізаторів і джерел безперебійного живлення.
4	Шкідливий вплив тепла	Підвищення температури у приміщенні через інтенсивне використання серверів.	Перегрів повітря, зниження працездатності персоналу.	Налаштування вентиляції та кондиціонування.
5	Шумове забруднення	Підвищений рівень шуму від серверного обладнання.	Втома, зниження концентрації працівників.	Використання шумоізоляції, окремих серверних кімнат.
6	Тривале сидіння за ПК	Робота з великими наборами даних потребує багато часу за комп’ютером.	М’язова втома, біль у спині, погіршення зору.	Дотримання режиму перерв, ергономіка робочого місця.
7	Високе енергоспоживання	Тривала робота обладнання значно збільшує витрати на електроенергію.	Високі фінансові витрати, ризик перевантаження мережі.	Використання енергоефективного обладнання.
8	Цифровий стрес	Робота з великими обсягами даних та високі вимоги до результатів.	Емоційне виснаження, втрата концентрації.	Організація регулярного відпочинку, психологічна підтримка.

швидкості інших процесів та можливого перегріву CPU за умов недостатнього охолодження.

Великі набори даних або складні моделі вимагають значного обсягу пам’яті, що може викликати затримки у виконанні задач та аварійне завершення роботи через нестачу пам’яті.

Відсутність попередньої обробки або аномалії в даних можуть спричинити помилки під час навчання та зниження точності моделі.

Інтенсивні обчислення збільшують енергоспоживання, що може бути критичним для мобільних пристроїв або пристроїв із обмеженим джерелом енергії.

Зберігання великих обсягів даних або доступ до обчислювальних ресурсів можуть бути уразливими до атак через витік даних та несанкціонований доступ до обчислювальної системи.

У таблиці 4.1 розглянуто небезпечні чинники з позиції охорони праці під час виконання задач машинного навчання. Вказані ризики можуть впливати як на персонал, так і на стабільність роботи обладнання, тому для їх мінімізації рекомендовано вживати відповідні заходи.

4.2. Розробка заходів із покращення умов праці виконавців

Сьогодні програмування є однією з ключових професій, яка пов’язана з високими інтелектуальними навантаженнями, тривалим перебуванням за комп’ютером та роботою в умовах підвищених вимог до продуктивності. Для забезпечення ефективності праці програмістів та збереження їхнього здоров’я важливо створювати комфортні та безпечні умови праці. У цьому розділі розглянуто основні ризики для програмістів та запропоновано заходи для їхнього усунення або зменшення.

Основні проблеми умов праці програмістів представлені на рис. 4.2.

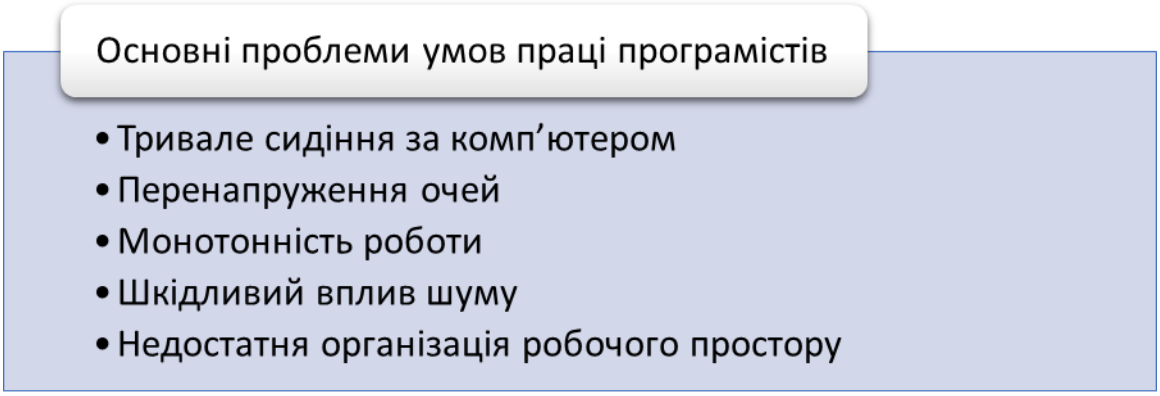

Рисунок 4.2 – Основні проблеми умов праці програмістів

Постійна фіксована поза призводить до проблем із хребтом, напругою м’язів та суглобів, що може спричинити розвиток остеохондрозу, болю в спині та шиї. Довготривале перебування перед монітором викликає втомлюваність очей, зниження гостроти зору та синдром «сухого ока». Постійна концентрація на одному виді діяльності призводить до емоційного виснаження, стресу та зниження продуктивності.

Робота у спільних приміщеннях або поруч із шумним обладнанням знижує концентрацію уваги та спричиняє психологічний дискомфорт. Неправильне налаштування робочого місця (висота стільця, освітлення, розташування монітора) впливає на ергономіку та комфорт праці.

Заходи щодо покращення умов праці програмістів представлені на рис. 4.3.

Ергономіка робочого місця

Використання регульованих стільців із підтримкою спини. Забезпечення правильного положення монітора: на рівні очей і на відстані 50–70 см. Забезпечення комфортного освітлення, уникання відблисків на екрані.

Регулярні перерви та фізична активність

Введення правил «20-20-20»: кожні 20 хвилин робити перерву на 20 секунд і дивитися на об’єкти на відстані 20 футів (~6 м). Рекомендація вставати кожну годину, виконувати легкі вправи для спини, рук та очей.

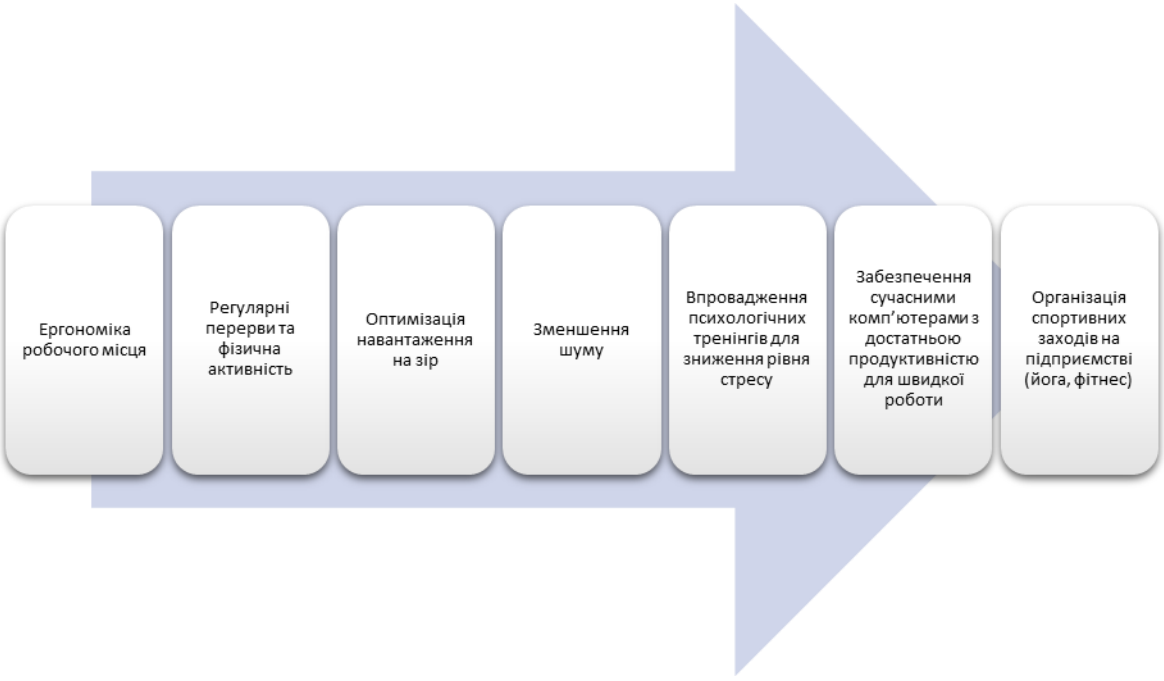

Рисунок 4.3 – Заходи щодо покращення умов праці програмістів

Оптимізація навантаження на зір

Використання програм, що регулюють яскравість екрана залежно від часу доби (наприклад, f.lux). Забезпечення наявності спеціальних моніторів із низьким рівнем шкідливого випромінювання. Регулярне відвідування офтальмолога для профілактики порушень зору.

Зменшення шуму

Організація окремих зон для тихої роботи. Використання шумоізоляційних матеріалів у приміщенні. Надання працівникам шумопоглинаючих навушників. Психологічна підтримка та організація праці

Впровадження психологічних тренінгів для зниження рівня стресу

Створення сприятливого мікроклімату в колективі. Розподіл задач для уникнення надмірного навантаження на працівників. Технічне оснащення робочих місць

Забезпечення сучасними комп'ютерами з достатньою продуктивністю для швидкої роботи

Використання якісних клавіатур та мишок для зниження напруги рук. Впровадження політики здорового способу життя. Окрім покращення умов

праці на робочому місці, необхідно заохочувати програмістів до ведення активного способу життя:

Організація спортивних заходів на підприємстві (йога, фітнес)

Забезпечення доступу до здорової їжі (зони харчування, корисні перекуси). Надання можливостей для прогулянок під час робочого дня.

Впровадження запропонованих заходів спрямоване на зниження негативного впливу умов праці на здоров’я програмістів, підвищення їхньої продуктивності та створення комфортного робочого середовища. Дотримання цих рекомендацій дозволить уникнути професійних захворювань, знизити рівень стресу та покращити загальний добробут працівників.

4.3. Розробка заходів із забезпечення безпеки під час надзвичайних ситуацій

Умови роботи програмістів, особливо в офісах і серверних кімнатах, можуть бути ускладнені ризиками, пов’язаними з надзвичайними ситуаціями, такими як пожежі, відключення електропостачання, витік води, природні катаклізми або кіберзагрози. Розробка та впровадження комплексних заходів безпеки є необхідними для збереження здоров'я працівників, цілісності даних та безперебійної роботи обладнання.

Пожежа може бути викликана перегрівом обладнання, коротким замиканням або порушенням пожежної безпеки. Відключення електроенергії призводить до втрати незбережених даних, простою в роботі та ризику пошкодження серверів. Витік води або підтоплення загрожує працездатності електронного обладнання та серверів. Природні катаклізми, землетруси, сильний вітер чи повені, можуть пошкодити приміщення та обладнання. Кіберзагрози, такі як хакерські атаки чи несанкціонований доступ до серверів, можуть призвести до витоку даних або блокування роботи систем.

Слід розробляти та реалізовувати комплекс заходів із забезпечення безпеки працівників. Захист від пожежі можна забезпечити завдяки встановленню автоматичних систем пожежогасіння, що не пошкоджують електроніку (наприклад, газові системи). Розміщення вогнегасників у легкодоступних місцях. Регулярна перевірка електромережі та обмеження використання несертифікованих пристроїв. Проведення тренінгів для персоналу щодо дій у разі пожежі.

Використання джерел безперебійного живлення (UPS) для підтримки роботи серверів та збереження даних під час відключення електроенергії. Встановлення резервних генераторів для забезпечення довготривалого енергопостачання. Слід здійснювати постійний моніторинг стану електромережі.

Для захисту від витоків води слід встановлювати системи виявлення витоків води у серверних кімнатах. Розміщення серверів на підвищеннях для уникнення пошкодження у разі підтоплення. Забезпечення належного функціонування дренажних систем у приміщеннях.

Захист від природних катаклізмів виконується завдяки створенню планів евакуації персоналу та збереження обладнання у разі надзвичайних ситуацій. Посилення конструкцій приміщень для захисту від землетрусів чи ураганів.

Використання захищених серверних шаф забезпечує протидію кіберзагрозам. Встановлення міжмережевих екранів (firewall) та антивірусних програм для захисту даних. Регулярне оновлення програмного забезпечення та паролів доступу. Організація резервного копіювання даних на захищених хмарних сервісах. Проведення навчання персоналу щодо правил кібербезпеки.

Організація навчання персоналу виконується завдяки регулярному проведенню тренувань із евакуації в разі пожежі чи інших надзвичайних ситуацій. Слід проводити інструктажі з використання засобів пожежогасіння та першої допомоги. Навчання з управління кіберзагрозами та дій у разі виявлення підозрілої активності в мережі.

Розробка та затвердження планів дій у разі надзвичайних ситуацій є основою забезпечення безпеки під час їх виникнення. Слід передбачити призначення відповідальних осіб за евакуацію персоналу та безпеку обладнання. Створення резерву необхідних матеріалів: вогнегасників, аптечок, додаткових батарей та паливних запасів для генераторів.

Запропоновані заходи дозволяють значно знизити ризики, пов’язані з надзвичайними ситуаціями, та забезпечити безперервну роботу програмістів і обладнання. Впровадження систем захисту, навчання персоналу та організація резервних рішень сприяє підвищенню безпеки та ефективності роботи в умовах можливих надзвичайних ситуацій.

РОЗДІЛ 5.
ВИЗНАЧЕННЯ ПОКАЗНИКІВ ЕФЕКТИВНОСТІ ВІД ВИКОРИСТАННЯ АЛГОРИТМІВ ПОПЕРЕДНЬОЇ ОБРОБКИ ДАНИХ НА РЕЗУЛЬТАТИ СТВОРЕННЯ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ

Для визначення економічної ефективності використання алгоритмів попередньої обробки даних враховувались:

1) час навчання моделі – тривалість навчання, яка прямо впливає на витрати електроенергії;

2) завантаження CPU – інтенсивність використання процесора, що визначає рівень енергоспоживання;

3) енергоспоживання: Вартість обчислювальних ресурсів розраховувалася за формулою:

$$C_{resources} = PCPU \cdot t_{train} \cdot C_{kWh}\ , \qquad (5.1)$$

де $PCPU$ – середнє енергоспоживання CPU. $PCPU$ =65 Вт; t_{train} – час навчання моделі в годинах; C_{kWh} – вартість 1 кВт·год, C_{kWh} =4.32 грн/кВт.

Результати розрахунків показників економічної ефективності алгоритмів попередньої обробки даних подано у табл. 5.1.

Таблиця 5.1 – Економічна ефективність алгоритмів попередньої обробки даних

Набір даних	Модель	Час навчання, с	Завантаження CPU, %	Вартість, грн.
No Outliers	LGBMRegressor	0.1459	54.5	0.0273
No Outliers	XGBRegressor	0.1779	5.6	0.0334
Standardized	LGBMRegressor	0.1499	29.5	0.0279
Standardized	GradientBoosting Regressor	1.3152	-58.0	0.3693
Reduced Columns	LGBMRegressor	0.1159	30.1	0.0216
Reduced Columns	RandomForestRegressor	3.1760	-20.1	0.8910

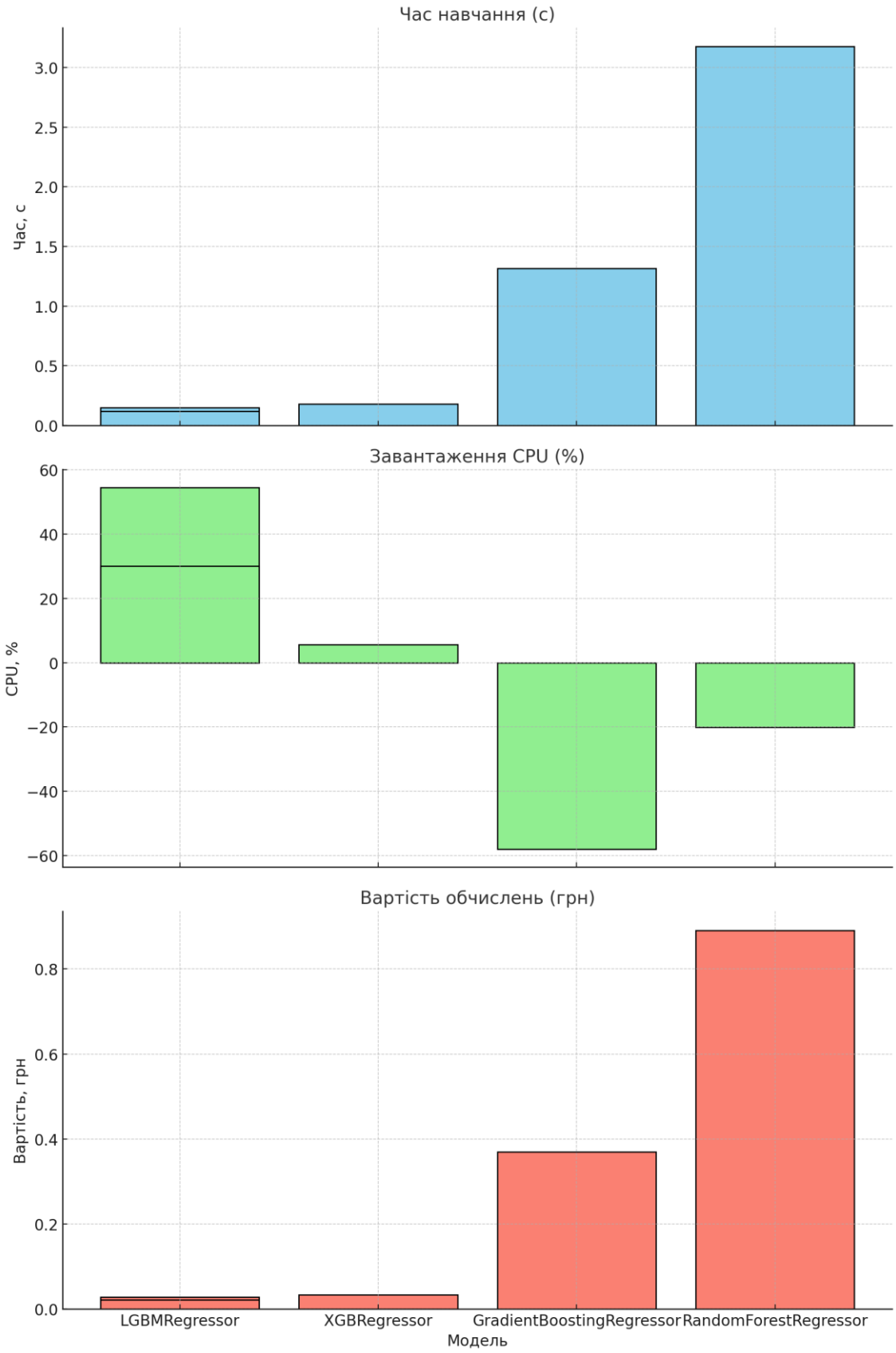

Рисунок 5.1 – Тенденції зміни показників економічної ефективності алгоритмів попередньої обробки даних подано

Найнижчі витрати на електроенергію спостерігаються для моделі LGBMRegressor із набором Reduced Columns (0.0216 грн). Найвищі витрати (0.8910 грн) спостерігались для моделі RandomForestRegressor, що обумовлено тривалим часом навчання.

Набір даних із видаленими колонками (Reduced Columns) забезпечив найкращі результати з точки зору економії ресурсів. Стандартизація також показала високу ефективність, особливо для моделей LGBMRegressor. Видалення аномалій та стандартизація суттєво скорочують час навчання моделей, знижуючи витрати на електроенергію.

Використання алгоритмів попередньої обробки даних, таких як видалення колонок та стандартизація, суттєво підвищує економічну ефективність машинного навчання. Найкращі результати отримано для моделі LGBMRegressor із набором даних Reduced Columns, яка забезпечила витрати на електроенергію лише 0.0216 грн за час навчання 0.1159 с. Аналогічно, для стандартизованого набору даних ця ж модель мала витрати 0.0279 грн при часу навчання 0.1499 с.

Рекомендовано застосовувати стандартизацію або зменшення кількості колонок для досягнення оптимального співвідношення точності та економічної ефективності. Ці результати підтверджують доцільність використання алгоритмів попередньої обробки для зниження енергетичних витрат та підвищення продуктивності моделей.

ВИСНОВКИ І ПРОПОЗИЦІЇ

Актуальність теми кваліфікаційної роботи зумовлена швидким зростанням обсягів даних та цінністю їх ефективного використання. У різних сферах якість обробки даних прямо впливають на точність прогнозів і рішень, які приймаються на основі моделей машинного навчання.

Аналіз стану використання та попередньої обробки даних засвідчив, що якість вхідних даних є ключовим фактором, який визначає ефективність і точність моделей машинного навчання.

Аналіз етапів попередньої обробки даних показав, що цей процес є багатоступеневим і включає кілька ключових компонентів. Основними етапами є виявлення та усунення пропущених значень, обробка аномалій, нормалізація та стандартизація числових даних, кодування категоріальних змінних і зменшення розмірності набору даних.

Аналіз стану дослідження алгоритмів попередньої обробки даних показав, що ефективність машинного навчання значною мірою залежить від якості підготовки даних. Дослідження також підтверджують, що вибір алгоритму попередньої обробки має бути адаптований до специфіки задачі та характеристик даних, оскільки некоректна обробка може призводити до втрати важливої інформації або появи систематичних помилок.

Основою машинного навчання є виконання попередньої обробки даних, оскільки якість даних напряму впливає на ефективність моделей. Нами проаналізовано алгоритми машинного навчання, які представлено на рис. 2.1. Процес побудови моделей машинного навчання включає кілька етапів, представлених на рисунку 2.2. Попередня обробка даних є ключовим етапом у процесі машинного навчання.

Інтеграція даних є першим етапом попередньої обробки, спрямованої на об’єднання даних із різними джерелами в єдину структуру для подальшого аналізу. Процес інтеграції забезпечує придатність даних, усуває дублікати та вирішує проблеми різних форматів і структур даних.

Відомо, що нормалізація даних суттєво покращує точність моделей, які чутливі до масштабу ознаки. Для SVM точність зросла на 15%, а для логістичної регресії – на 12%.

Досліджено вплив на заповнення пропущених значень на моделі дерев рішень та Random Forest. Автори використовували такі методи заповнення: заповнення середнім значенням; інтерполяція; видалення записів із пропусками. Найкращі результати показали заповнення середнім значенням. Точність Random Forest зросла з 80% до 88%.

Досліджено вплив видалення аномалій на точність моделі лінійної регресії. Після видалення аномалії середня абсолютна похибка (MAE) зменшилася на 20%.

Проаналізовано методи вибору ознак, такі як методи головних компонентів (PCA) та відбір на основі ознак (важливість ознак) у Random Forest. Встановлено, що скорочення розмірності набору даних від 50 до 10 знаків за допомогою PCA скоротило час навчання моделей на 40%, зберігаючи точність на рівнях 92%.

Для дослідження впливу алгоритмів попередньої обробки даних на результати створення моделей машинного навчання нами використано набори даних із платформи Kaggle. Цей набір даних використовується для навчання моделей прогнозуванню оцінок студентів. Розміри набору даних становлять 30641 рядок на 14 стовпців. Цільовою змінною в цьому дослідженні є результат іспиту студентів з математики.

Встановлено, що за расою студенти розподіляються таким чином, як представлено на рис. 2.4. Атрибут «ParentEduc» у наборі даних містить інформацію про найвищий рівень освіти, досягнутий батьками студентів у наборі даних. Аналіз цього стовпця за допомогою гістограми відкриває розуміння рівня освіти батьків учнів та його потенційного впливу на успішність учнів. Бачимо, що менша частина батьків студентів мала вищу освіту.

Максимальне значення для кожного іспиту становить 100. Варто зазначити, що набір даних має майже рівний розподіл студентів за статтю:

15424 жінки та 15217 чоловіки. Таким чином, ми можемо оцінити загальну середню успішність в залежності від різних величин, наприклад, статі.

Використовуючи кореляційну матрицю, можна визначити, наскільки сильно чисельні змінні корелюють одна з одною. На основі матриці на рисунку 2.7 можна встановити, що всі атрибути даних що мають сильну позитивну кореляцію відносно один одного.

Далі в процесі аналізу даних також виявились деякі закономірності, які можуть вплинути на методи вирішення проблеми. Аналізуючи гістограми на рисунку 2.9 нижче, ми можемо помітити значну різницю в успішності чоловіків і жінок з різних предметів. Виходячи з інформації, можна зробити висновок, що чоловіки показують найкращі показники з математики, а жінки – на іспитах з мови та літератури. У наборі даних усі дані були заповнені. Тобто пропущених даних не спостерігається (рис. 2.10).

Для вирішення задачі прогнозування оцінок студентів з математики на основі наявних у них даних було відібрано окремі алгоритми машинного навчання із учителем: LGBMRRegressor; XGBRegressor; GradientBoostingRegressor; RandomForestRegressor.

У подальшому дослідженні розглядали такі варіанти підготовки даних: 1) видалення рядків з аномаліями; 2) заміна аномалій середнім значенням; 3) стандартизація значень; 4) видалено стовпці «Gender», «ParentEduc», «TestPrep», які за результатами аналізу не мали сильного впливу на результати студентів на іспиті з математики.

Насамперед нами створено код, який виконує перетворення категоріальних змінних у числові за допомогою методу LabelEncoder з бібліотеки sklearn (рис. 3.1). У DataFrame (df) тепер замість текстових категорій у зазначених стовпцях представлені числові значення (рис. 3.2).

У подальшому нами створено код, що виконує виявлення аномалій у числових колонках DataFrame (df) з використанням методу міжквартильного розмаху (IQR) (рис. 3.3). Аномалії визначаються як значення, що знаходяться за межами діапазону, визначеного нижньою та верхньою межами, розрахованими

на основі IQR. Наступний крок забезпечує об’єднання аномалій у загальний DataFrame під назвою outliers_df (рис. 3.4).

У подальшому нами створено код, що дозволяє обробляти аномалії у числових даних двома різними способами: 1) видалення рядків із аномаліями; 2) заміна аномалій середнім значенням відповідної колонки (рис. 3.5). Результати двох методів обробки аномалій можна переглянути за допомогою функції head() і вони представлені на рис. 3.6.

Нами написано код, що виконує стандартизацію числових колонок у DataFrame (df) за допомогою методу StandardScaler із бібліотеки sklearn (рис. 3.7). Стандартизація змінює розподіл даних таким чином, що кожна змінна матиме середнє значення 0 та стандартне відхилення 1. У результати отримуємо фрагмент створеного масиву даних після виконання стандартизації (рис. 3.8).

Нами написано код спрямований на порівняння продуктивності різних моделей регресії для прогнозування оцінок студентів із математики (MathScore) на різних наборах даних. Він також оцінює вплив методів попередньої обробки даних на точність, ефективність та навантаження на обчислювальну систему. Написаний код дозволяє автоматизувати навчання моделей та оцінку їх продуктивності на різних наборах даних. Результати, такі як точність моделей, обчислювальні витрати та ефективність методів обробки даних, можна використовувати для порівняння та аналізу.

Результати дослідження показали, що методи попередньої обробки даних мають суттєвий вплив на точність та ефективність моделей. Найкращі результати точності були досягнуті на стандартизованих даних, тоді як видалення колонок із малим впливом сприяло значному зменшенню часу навчання без значного зниження точності прогнозу. Алгоритм LGBMRRegressor виявився найбільш ефективним за всіма критеріями, забезпечуючи високу точність, швидке навчання та низьке навантаження на обчислювальну систему.

Нами розроблено заходи із охорони праці під час виконання машинного навчання моделей, дотримання яких забезпечить створення безпечних умов праці.

Використання алгоритмів попередньої обробки даних, таких як видалення колонок та стандартизація, суттєво підвищує економічну ефективність машинного навчання. Найкращі результати отримано для моделі LGBMRegressor із набором даних Reduced Columns, яка забезпечила витрати на електроенергію лише 0.0216 грн за час навчання 0.1159 с. Аналогічно, для стандартизованого набору даних ця ж модель мала витрати 0.0279 грн при часу навчання 0.1499 с.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Введення в машинне навчання за допомогою Python и Scikit-Learn. URL: https://habr.com/ua/company/mlclass/blog/247751/ (дата звернення: 14.09.2024).

2. Жидецький В.Ц., Джигирей В.С., Мельников О.В. Основи охорони праці. Підручник. Вид. 5-е, доповнене. Львів: Афіша, 2012. 350с.

3. Класифікація в Python з Scikit-Learn та Pandas. URL: https://stackabuse.com/classification-in-pythonwith-scikit-learn-and-pandas/ (дата звернення: 18.10.2024).

4. Лехман С.Д., Рублев В.І., Рябцев Б.І. Запобігання аварійності і травматизму у сільському господарстві. К.: Урожай, 1993. 267 с.

5. Павлиш В. А. ,. Гліненко Л. К, Шаховська Н. Б. Основи інформаційних технологій і систем: підручник. Львів: Львівська політехніка, 2018. 620 с.

6. Ситнік Б.Т. Основи інформаційних систем і технологій: навч. посіб. Харків: УкрДУЗТ, 2018. 130 с.

7. Tryhuba, A., Boyarchuk, V., Tryhuba, I., Ftoma, O., Padyuka, R., Rudynets, M. Forecasting the Risk of the Resource Demand for Dairy Farms Basing on Machine Learning. Proceedings of the 2nd International Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT+DS 2020). 2020; I, 327-340.

8. Aksangur I., Eren B., Erden C. Evaluation of data preprocessing and feature selection process for prediction of hourly PM10 concentration using long short-term memory models, Environmental Pollution 311 (2022). doi: 10.1016/j.envpol.2022.119973.

9. Balaji K.V., What is Data Visualization and Why Is It Important?, 2020. URL: https://medium.com/analytics-vidhya/what-is-data-visualization-and-why-is-it-important-

3c2ccb108945#:~:text=Data%20visualization%20is%20the%20visual,communicate%20it%20to%20your%20peers.

10. Botelho B., Bigelow S. J. Big Data, 2022. URL: https://www.techtarget.com/searchdatamanagement/definition/bigdata#:~:text=Big%20data%20is%20a%20combination,and%20other%20advanced%20analytics%20applications.

11. Cabello-Solorzano, K., Ortigosa de Araujo, I., Peña, M., Correia, L., J. Tallón-Ballesteros, A. (2023). The Impact of Data Normalization on the Accuracy of Machine Learning Algorithms: A Comparative Analysis. In: García Bringas, P., et al. 18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023). SOCO 2023. Lecture Notes in Networks and Systems, vol 750. Springer, Cham. https://doi.org/10.1007/978-3-031-42536-3_33

12. Conejero J.M., Preciado J.C., Fernandez-Garcia A.J., Prieto A.E. Rodriguez-Echeverria R., Towards the use of Data Engineering, Advanced Visualization techniques and Association Rules to support knowledge discovery for public policies. Expert Systems with Applications, 170 (2021). Doi: 10.1016/j.eswa.2020.114509.

13. Crone S. F., Lessmann S., Stahlbock R. The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing, European Journal of Operational Research 173 (2006) 781-800. doi: 10.1016/j.ejor.2005.07.023.

14. Davaasambuu B., Yu K., Sato T. Self-optimization of handover parameters for long-term evolution with dual wireless mobile relay nodes. Future Internet, 7 (2) (2015), pp. 196-213.

15. Davis J.F. Process data analysis and interpretation. Adv. Chem. Eng., 2022. 25.

16. Dealing with Outliers Using Three Robust Linear Regression Models. URL: https://developer.nvidia.com/blog/dealing-with-outliers-using-three-robust-linear-regression-models/

17. Farhangi F. Investigating the role of data preprocessing, hyperparameters tuning, and type of machine learning algorithm in the improvement of drowsy EEG signal modeling, Intelligent Systems with Applications 15 (2022). doi: 10.1016/j.iswa.2022.200100.

18. Fu Y, Liao H, Lv L. A Comparative Study of Various Methods for Handling Missing Data in UNSODA. Agriculture. 2021; 11(8):727. https://doi.org/10.3390/agriculture11080727.

19. Great Learning Team, 2020. URL: https://medium.com/@mygreatlearning/what-is-artificial-intelligence-how-does-ai-work-and-future-of-it-d6b113fce9be.

20. Gupta S., Gupta A., Gupta S. Dealing with noise problem in machine learning data-sets: a systematic review. Procedia Comput. Sci., 161 (2019), pp. 466-474.

21. Hodson T. O., Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not, Geosci. Model Dev., 15 (2022): 5481–5487, doi: 10.5194/gmd-15-5481-2022.

22. Huang J., Li Y. F., Xie M. An empirical analysis of data preprocessing for machine learning-based software cost estimation, 67(2015) 108-127. doi: 10.1016/j.infsof.2015.07.004.

23. Kaggle Team, Students Exam Scores: Extended Dataset, 2023. URL: https://www.kaggle.com/datasets/desalegngeb/students-exam-scores/data

24. Kashina M., Lenivtceva I. D., Kopanitsa G. D. Preprocessing of unstructured medical data: the impact of each preprocessing stage on classification, Procedia Computer Science 178 (2020) 284-290. doi: 10.1016/j.procs.2020.11.030.

25. Kimmons R., Exam Scores dataset, 2012. URL: http://roycekimmons.com/tools/generated_data/exams

26. Koval N., Tryhuba A., Kondysiuk I., Tryhuba I., Boiarchuk O., Rudynets M., Grabovets V., Onyshchuk V. Forecasting the Fund of Time for Performance of Works in Hybrid Projects Using Machine Training Technologies. Proceedings of the 3nd International Workshop on Modern Machine Learning

Technologies and Data Science Workshop. Proc. 3rd International Workshop (MoMLeT&DS 2021). 2021; I, 96-206.

27. Lao R. Beginnner's A. Guide to Machine Learning, 2018. URL: https://medium.com/@randylaosat/a-beginners-guide-to-machine-learning-dfadc19f6caf.

28. LightGBM Core Team, LGBMRegressor, 2024. URL: https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRegressor.html

29. Maharana K., Mondal S., Nemade B. A review: Data pre-processing and data augmentation techniques, Global Transitions, 3 (2022) 91-99. doi: 10.1016/j.gltp.2022.04.020.

30. Malanchuk, O., Tryhuba, A., Tryhuba, I., Sholudko, R., Pankiv, O. A Neural Network Model-based Decision Support System for Time Management in Pediatric Diabetes Care Projects. International Scientific and Technical Conference on Computer Sciences and Information Technologies, 2023.

31. Olisah C.C., Smith L., Smith M. Diabetes mellitus prediction and diagnosis from a data preprocessing and machine learning perspective, Computer Methods and Programs in Biomedicine 220 (2022). doi: 10.1016/j.cmpb.2022.106773.

32. Rajendran G.B., Kumarasamy U.M., Zarro C., Divakarachari P.B., Ullo S.L. Land-use and land-cover classification using a human group-based particle swarm optimization algorithm with an LSTM Classifier on hybrid pre-processing remote-sensing images. Remote Sens., 12 (24) (2020), p. 4135

33. Ram T., What is Data Quality in Machine Learning, 2023. URL: https://www.analyticsvidhya.com/blog/2023/01/the-role-of-data-quality-in-machine-learning/

34. Rowe W., Mean Square Error & R2 Score Clearly Explained, 2018. URL: https://www.bmc.com/blogs/mean-squared-error-r2-and-variance-in-regression-analysis/

35. Schoot M., Kapper C., Kollenburg G. H., Postma G. J., Kessel G., Buydens L. M., Jansen J. J. Investigating the need for preprocessing of near-infrared

spectroscopic data as a function of sample size, Chemometrics and Intelligent Laboratory Systems 204 (2020). doi: 10.1016/j.chemolab.2020.104105.

36. SKLearn Core Team, Gradient boosted trees, 2024. URL: https://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting

37. SKLearn Core Team, Random forests and other randomized tree ensembles, 2024. URL: https://scikit-learn.org/stable/modules/ensemble.html#forest

38. Soni N., Sharma E. K., Singh N., Kapoor A. Artificial Intelligence in Business: From Research and Innovation to Market Deployment, Procedia Computer Science (2020): 2200-2210. doi: 10.1016/j.procs.2020.03.272.

39. Sustainable Communication Networks and Application. Springer Scince and Business Media LLC. 2022. P. 263.

40. Tryhuba, A., Boyarchuk, V., Tryhuba, I., Ftoma, O., Padyuka, R. & Rudynets, M. Forecasting the risk of the resource demand for dairy farms basing on machine learning. CEUR Workshop Proceedings. https://www.scopus.com/authid/detail.uri?authorId=57205225539. 2021; 2631: 327–340.

41. Tryhuba, A., Kondysiuk, I., Tryhuba, I., Boiarchuk, O., Tatomyr, A. Intellectual information system for formation of portfolio projects of motor transport enterprises. CEUR Workshop Proceedings. 2022; 3109, 44–52.

42. Tryhuba, A., Tryhuba, I., Ftoma, O. & Boyarchuk, O. Method of quantitative evaluation of the risk of benefits for investors of fodder-producing cooperatives. International Scientific and Technical Conference on Computer Sciences and Information Technologies, https://www.scopus.com/authid/detail.uri?authorId=57205225539. 2019; 3: 55–58.

43. Tryhuba, A., Tryhuba, I., Malanchuk, O., Marmulyak, A. A deep neural network model for predicting the competitive score of social projects for community development. CEUR Workshop Proceedings, 2024, 3711, pp. 55–74.

44. Tryhuba, A., Vovk, M., Batyuk, B., Holomsha, O., Sava, A. Improving the quality of management in the system of forecasting milk procurement in

communities usage the technology of neutron networks. Journal of Hygienic Engineering and Design, 2022, 40, pp. 201–209

45. Tryhuba, A., Koval, N., Tryhuba, I. & Boiarchuk, O. Application of sarima models in information systems forecasting seasonal volumes of food raw materials of procurement on the territory of communities. CEUR Workshop Proceedings. 2022; 3295: 64–75. https://www.scopus.com/authid/detail.uri?authorId=57205225539.

46. Tryhuba, A., Malanchuk, O., Tryhuba, I. Prediction of the Duration of Inpatient Treatment of Diabetes in Children Based on Neural Networks. CEUR Workshop Proceedings. 2023; 3426, 122–135.

47. Tryhuba, I., Hutsol, T., Tryhuba, A., Tulej, W., Sojak, M. An Approach to Assessing the State of Organic Waste Generation in Community Households Based on Associative Learning. Sustainability (Switzerland), 2023, 15(22), 15922.

48. Tryhuba, I., Tryhuba, A., Hutsol, T., Tulej, W., Sojak, M. Prediction of Biogas Production Volumes from Household Organic Waste Based on Machine Learning. Energies, 2024, 17(7), 1786.

49. Understanding Feature Selection Techniques in Machine Learning. URL: https://medium.com/@nirajan_DataAnalyst/understanding-feature-selection-techniques-in-machine-learning-02e2642ef63e

50. XGBoost Core Team, Python API Reference, 2024. URL: https://xgboost.readthedocs.io/en/stable/python/python_api.html

51. Zhou Q., Ooka R. Influence of data preprocessing on neural network performance for reproducing CFD simulations of non-isothermal indoor airflow distribution, Energy and Buildings 230 (2021). doi: 10.1016/j.enbuild.2020.110525.

52. Ziegler P., Dittrich K.R., Krogstie J., Opdahl A.L., Brinkkemper S. Data integration – problems, approaches, and perspectives. Conceptual Modelling in Information Systems Engineering, Springer, Berlin, Heidelberg. 2007, 10.1007/978-3-540-72677-7_3

ДОДАТКИ

Додаток А

Код для навчання моделей прогнозування оцінок студентів з математики за використання різних алгоритмів попередньої обробки даних

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import time
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from lightgbm import LGBMRegressor
from xgboost import XGBRegressor
from sklearn.ensemble import GradientBoostingRegressor, RandomForestRegressor
import psutil

# Results storage
results = []

# Datasets
datasets = {
    "No Outliers": df_no_outliers,
    "Replaced Outliers": df_replaced_outliers,
    "Standardized": df_standardized,
    "Reduced Columns": df_standardized.drop(columns=["Gender", "ParentEduc", "TestPrep"],
errors="ignore")
}

# Target and Features
target = "MathScore"

# Loop through datasets and models
for dataset_name, dataset in datasets.items():
    X = dataset.drop(columns=[target], errors="ignore")
    y = dataset[target]

    # Split into train and test sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Models
    models = {
        "LGBMRegressor": LGBMRegressor(force_row_wise=True),  # Explicit row-wise training
        "XGBRegressor": XGBRegressor(eval_metric="rmse"),
        "GradientBoostingRegressor": GradientBoostingRegressor(),
        "RandomForestRegressor": RandomForestRegressor()
    }

    for model_name, model in models.items():
        # Record CPU usage
```

```
        cpu_start = psutil.cpu_percent(interval=None)
        start_time = time.time()

        # Train the model
        model.fit(X_train, y_train)

        # Record CPU usage and time after training
        cpu_end = psutil.cpu_percent(interval=None)
        training_time = time.time() - start_time
        cpu_usage = cpu_end - cpu_start

        # Predict
        y_pred = model.predict(X_test)

        # Metrics
        mae = mean_absolute_error(y_test, y_pred)
        rmse = mean_squared_error(y_test, y_pred, squared=False)  # Updated for compatibility
        r2 = r2_score(y_test, y_pred)

        # Store results
        results.append({
            "Dataset": dataset_name,
            "Model": model_name,
            "MAE": mae,
            "RMSE": rmse,
            "R2": r2,
            "CPU Usage (%)": cpu_usage,
            "Training Time (s)": training_time
        })

# Convert results to DataFrame
results_df = pd.DataFrame(results)

# Plotting
# 1. Bar plot for model accuracy
plt.figure(figsize=(12, 8))
sns.barplot(data=results_df, x="Model", y="R2", hue="Dataset", palette="viridis")
plt.title("Model R2 Score by Dataset", fontsize=16)
plt.ylabel("R2 Score", fontsize=12)
plt.xlabel("Model", fontsize=12)
plt.xticks(rotation=45, fontsize=10)
plt.legend(title="Dataset", fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

# 2. CPU Usage Histogram
plt.figure(figsize=(12, 8))
sns.barplot(data=results_df, x="Model", y="CPU Usage (%)", hue="Dataset", palette="plasma")
plt.title("CPU Usage by Model and Dataset", fontsize=16)
plt.ylabel("CPU Usage (%)", fontsize=12)
plt.xlabel("Model", fontsize=12)
```

```
plt.xticks(rotation=45, fontsize=10)
plt.legend(title="Dataset", fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

# 3. Training Time Histogram
plt.figure(figsize=(12, 8))
sns.barplot(data=results_df, x="Model", y="Training Time (s)", hue="Dataset", palette="magma")
plt.title("Training Time by Model and Dataset", fontsize=16)
plt.ylabel("Training Time (seconds)", fontsize=12)
plt.xlabel("Model", fontsize=12)
plt.xticks(rotation=45, fontsize=10)
plt.legend(title="Dataset", fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

# Display results table
print("Model Performance Results:")
print(results_df)

# Save results to CSV
results_df.to_csv("model_performance_results.csv", index=False)
```