ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ, УПРАВЛІННЯ, ПРАВА ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

Пояснювальна записка

до кваліфікаційної роботи на здобуття ступеня вищої освіти магістр

на тему: **«Методи, технології та інструментальні засоби інтелектуальної обробки даних»**

Виконав: здобувач вищої освіти
за освітньо-професійною програмою
Інформаційні управляючі системи та технології спеціальності
126 Інформаційні системи та технології
ступеня вищої освіти магістр
групи 126ІСТ_мд_22
Тищенко А. В.
Керівник: Одарущенко О. М.
Рецензент: Яхін С.В.

Полтава – 2023 року

ВСТУП

Актуальність теми дослідження. Інтелектуальна обробка даних, або, як її ще називають, Data Mining, Data Analysis, або Data Science, стала однією з найбільш актуальних галузей сучасних інформаційних технологій. З кожним роком об'єми інформації, що генерується і зберігається, зростають експоненційно. Відомо, що інформація в сучасному світі є найціннішим ресурсом, і вміння аналізувати та використовувати її стає все більш критичним для підприємств, наукових установ, урядових організацій та інших сфер життя суспільства [1].

Методи, технології та інструментальні засоби інтелектуальної обробки даних являють собою набір інноваційних підходів, технологій та програмних засобів, які допомагають виявляти закономірності, тренди та цінну інформацію в великих обсягах даних. Ця галузь зростає не тільки завдяки розвитку обчислювальної потужності комп'ютерів, але і завдяки вдосконаленню алгоритмів та методів аналізу даних. Методи інтелектуальної обробки даних включають в себе статистичний аналіз, машинне навчання, глибоке навчання, обробку природної мови, графічні аналітичні методи, кластеризацію, класифікацію та багато іншого. Інструментальні засоби, такі як програмні платформи для аналізу даних і великі обчислювальні кластери, роблять ці методи доступними для широкого кола фахівців [2]. Інтелектуальна обробка даних є справжнім каталізатором інновацій та змін в сучасному світі. Вона допомагає впроваджувати наукові дослідження, оптимізувати виробництво, покращувати якість медичних діагнозів, прогнозувати зміни клімату, а також робити більше для підвищення якості життя нашого суспільства. Застосування інтелектуальної обробки даних розповсюджується в різних сферах: від бізнесу та маркетингу до медицини та науки. Компанії використовують її для прогнозування попиту, підвищення ефективності процесів, виявлення шахрайства та багато іншого. У науці інтелектуальна обробка даних допомагає відкривати нові знання і розуміння у великих наборах даних [3]. Методи, технології та інструментальні засоби інтелектуальної обробки даних є ключовими компонентами сучасного інформаційного суспільства. Їхнє застосування вже

принесло значні перемоги і досягнення, але також поставило перед нами нові виклики і питання, які потребують уваги та обговорення.

Мета дослідження – розробка навчальних матеріалів для вирішення завдань класифікації, кластеризації і регресії.

Завданнями кваліфікаційної роботи є:

1. Проведення аналізу методів та технологій інтелектуальної обробки даних.

2. Розгляд та вибір ефективного програмного забезпечення.

3. Розробка навчальних матеріалів для вирішення завдань інтелектуального аналізу.

Об'єкт дослідження – процеси дослідження математичних методів і їх реалізації для вирішення завдань класифікації, кластеризації і регресії в ході обробки великих даних.

Предмет дослідження – методи вирішення завдань класифікації, кластеризації і регресії.

Методи дослідження. У роботі використані різноманітні методи дослідження для збору, аналізу та інтерпретації даних, такі як літературний огляд, експериментальні дослідження, методи машинного навчання, моделювання, статистичний аналіз.

Інформаційна база кваліфікаційної роботи складається з результатів наукових досліджень, включаючи фахові статті, монографії та аналітичні звіти, які стосуються експлуатації критичних технічних систем. Ці матеріали були розроблені науково-дослідними та дисертаційними роботами в рамках визначеної тематики і отримані в результаті експертного аналізу державних та міжнародних експертних груп.

Елементи наукової новизни полягають в наступних аспектах роботи:

1. Використання новітніх методів;

2. Інтеграція технік глибокого навчання;

3. Використання оновлених прикладів та датасетів;

4. Адаптація до застосувань в реальному світі.

Практична значущість роботи полягає у розробці навчальних матеріалів для вирішення задач класифікації, регресії та кластеризації в Weka для створення ефективного та зрозумілого засобу для навчання та розвитку в області аналізу даних і машинного навчання.

Апробація результатів дослідження відбувалася шляхом оприлюднення доповідей на міжнародній та студентській конференціях.

Публікації. За результатами проведеного дослідження опубліковані тези: «Методи, технології та інструментальні засоби інтелектуальної обробки даних», матеріали щорічної студентської наукової конференції Полтавського державного аграрного університету, 28 вересня 2023 р. Полтава: ПДАУ, 2023.

Зв'язок роботи з науковими програмами, планами, темами: дослідження, проведені в кваліфікаційній роботі виконувалися в рамках/інтересах науково-дослідної роботи «Розвиток підприємництва: управлінські, економічні, інноваційна та правові аспекти» відповідно до договору №9 від 15.05.2023 р. між ТОВ «ПАФ Гарант» та Полтавським державним аграрним університетом (розділ «Обгрунтування показників оцінювання гарантоздатності розподілених інформаційних систем»).

Структура та обсяг роботи. Робота складається з вступу, трьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 67 сторінок; робота містить 26 рисунків; 2 таблиці; список використаних джерел включає 45 найменувань.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ ТА ТЕХНОЛОГІЙ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ДАНИХ

1.1 Основи обробки даних

Основи обробки даних представляють собою базовий набір концепцій, методів і процесів, які використовуються для збору, обробки, аналізу та інтерпретації інформації, що міститься в наборах даних. Ці основи є фундаментальними для роботи з даними, незалежно від галузі застосування, і вони лежать в основі інтелектуальної обробки даних (Data Processing). Далі розглянемо аспекти основ обробки даних.

Першим аспектом розглянемо збір даних. Збір даних є однією з фундаментальних складових обробки даних і відіграє важливу роль в процесі перетворення інформації на корисну та відповідну для подальшого аналізу та використання [4]. Розглянемо докладніше, що включає в себе основа обробки даних.

Початковим кроком є визначення джерел даних. Цей етап збору даних несе за собою ідентифікацію джерел, з яких можна отримати інформацію. Ці джерела можуть бути різними: датчики, бази даних, опитування, соціальні мережі, лог-файли, веб-сторінки, документи тощо. Другим етапом є планування та дизайн збору даних: на цьому етапі визначається стратегія та методи збору даних, вирішується, як будуть збиратися дані, скільки даних потрібно зібрати, і які інструменти і техніки використовуватимуться для цього. Також розробляється план опитування (якщо це відомість) та формат даних. Наступним кроком є безпосередньо збір даних, тобто фактичний процес отримання інформації з обраних джерел. Наприклад, це може включати запуск датчиків, опитування респондентів, вилучення інформації з баз даних, сканування документів тощо. Важливо забезпечити точність і повноту даних під час їх збору. Далі йде перевірка та очищення даних: після збору даних вони можуть містити помилки, некоректні

значення, дублікати. На цьому етапі проводиться перевірка та очищення даних для того, щоб вони були придатними для аналізу. Це може включати видалення аномальних даних, виправлення помилок і заповнення пропусків. І останнім кроком можна назвати збереження та організацію даних: очищені дані зазвичай зберігаються у відповідних форматах та структурах, таких як бази даних або файли. Важливо правильно організувати дані, щоб вони були легкодоступними для подальшого аналізу та обробки [5].

Другим аспектом варто зазначити аналіз даних. Аналіз даних є однією з ключових складових обробки даних, і він включає в себе ряд методів, інструментів і технік для вивчення, розуміння і вибору цінної інформації з набору даних. Аналіз даних допомагає виявляти закономірності, тренди, аномалії та інші важливі відомості, які можуть бути використані для прийняття рішень, створення прогнозів і вирішення конкретних завдань в різних галузях. Аналіз даних є критичним етапом в процесі обробки даних, оскільки він допомагає зрозуміти структуру даних, знайти складні зв'язки і зробити інформацію корисною для прийняття рішень. Від правильно проведеного аналізу даних залежить ефективність подальших кроків у вирішенні завдань та досягненні [6].

Третім аспектом є моделювання і прогнозування. Моделювання і прогнозування є ключовою складовою обробки даних і включають в себе процес створення математичних або статистичних моделей для передбачення майбутніх подій або розуміння залежностей в даних. Ця складова допомагає перетворити набір даних на цінну інформацію та використовувати її для прийняття обґрунтованих рішень. Моделювання і прогнозування можуть використовуватися в різних галузях, включаючи бізнес, науку, медицину, фінанси, інженерію та багато інших. На основі аналізу даних, проведеного на попередніх етапах обробки, моделювання і прогнозування включають такі дії [7]:

1. Обрання моделі: вибір математичної моделі або алгоритму, який найкраще підходить для розглянутого завдання і типу даних.

2. Побудова моделі: розробка моделі на основі обраних алгоритмів та підготовлених даних. Це може включати в себе підгонку параметрів моделі до тренувальних даних.

3. Перевірка моделі: оцінка якості моделі за допомогою тестових даних або перехресної перевірки (крос-валідації) для визначення її точності і надійності.

4. Прогнозування: використання натренованої моделі для прогнозування майбутніх значень або класифікації нових даних.

5. Оцінка результатів: аналіз і оцінка результатів прогнозування для зрозуміння, наскільки добре модель справляється зі своєю задачею.

Моделювання і прогнозування можуть використовуватися для різних цілей, таких як прогнозування продажів, визначення ризиків у фінансовому портфелі, класифікація текстової інформації, виявлення аномалій у виробництві, прогнозування погоди та багато інших застосувань. Ці процеси дозволяють використовувати великі обсяги даних для виявлення корисних взаємозв'язків і підтримки прийняття рішень на основі даних.

Четвертим кроком є інтерпретація та висновки. Інтерпретація та висновки в інтелектуальній обробці даних є ключовим етапом аналізу і дозволяють зробити значущі висновки з отриманих результатів. Цей процес включає в себе розуміння та пояснення знайдених закономірностей та прийняття обґрунтованих рішень на основі аналізу даних. Алгоритм інтерпритації може влючати такі складові як визначення важливості, виявлення патернів і залежностей, розробки висновків, прийняття рішень, комунікація результатів, відстеження та вдосконалення. Інтерпретація і висновки в інтелектуальній обробці даних допомагають використовувати дані для досягнення конкретних цілей і покращення прийняття рішень в різних сферах діяльності [8].

Наступним кроком варто зазначити візуалізацію результатів. Візуалізація результатів інтелектуальної обробки даних грає важливу роль у розумінні, комунікації та використанні інформації, отриманої під час аналізу та обробки даних. Графічні зображення можуть надати інсайти, зробити дані більш доступними та сприяти прийняттю обґрунтованих рішень. Ось деякі підходи до

візуалізації результатів інтелектуальної обробки даних: графіки та графіки розподілу, графіки розсіювання, теплові карти, лінійні графіки та часові ряди, географічні карти, дерева прийняття рішень та графи, словники і хмари слів, анімація, дашборди, інфографіка. Візуалізація є потужним інструментом для розуміння та комунікації результатів інтелектуальної обробки даних, і вона допомагає зробити дані більш доступними та корисними для прийняття рішень [9].

Останнім аспектом є збереження і передача даних. Завершальний етап, який включає в себе збереження та архівування даних для майбутнього використання, а також передачу результатів аналізу стейкхолдерам або іншим системам. Збереження і передача даних в інтелектуальній обробці даних є важливими аспектами, оскільки ці дані мають бути доступними, безпечними та ефективними для обробки та аналізу [10].

Структурна схема основних аспектів інтелектуальної обробки даних зображені на рисунку 1.1.

Рисунок 1.1 – Основні аспекти інтелектуальної обробки даних

Основи обробки даних є необхідними для роботи в багатьох сферах, включаючи бізнес, науку, медицину, соціальні науки та багато інших. Зрозуміння і вміння використовувати ці основи дозволяють ефективно аналізувати інформацію, приймати обґрунтовані рішення і досягати бажаних результатів в різних галузях діяльності.

1.2 Огляд алгоритмів машинного навчання та підходів до обробки та аналізу даних.

Машинне навчання (Machine Learning, ML) відіграє ключову роль в інтелектуальній обробці даних, сприяючи автоматизації процесів аналізу та прийняття рішень на основі великих обсягів інформації. Проведення огляду алгоритмів машинного навчання для виняткової обробки даних є важливим етапом у використанні машинного навчання для рішення певних завдань. Вибір правильного алгоритму великою мірою залежить від типу даних, які ви маєте, і конкретного завдання, яке ви намагаєтеся вирішити. Нижче наведено огляд деяких типових алгоритмів машинного навчання [11]:

1. Огляд алгоритмів. Машинне навчання використовує алгоритми для навчання моделей на основі даних. Декілька популярних алгоритмів машинного навчання включають в себе: лінійну регресію, дерева рішень, метод опорних векторів (SVM), нейронні мережі, k-найближчих сусідів (k-NN), навчання з підкріпленням та багато інших. Розглянемо детальніше деякі математичні аспекти [12].

Модель простої лінійної регресії використовується для прогнозування числової змінної (зазвичай позначеної як Y) на основі однієї незалежної змінної (позначеної як X). Модель має наступний вигляд:

$$Y = \beta_0 + \beta_1 X + \varepsilon \tag{1.1}$$

де: Y – прогнозована змінна.

β_0 – зсув (intercept), який показує значення Y, коли $X = 0$;

β_1 – коефіцієнт нахилу (slope), який визначає, наскільки змінюється Y при зміні X на одиницю;

ε – помилка моделі, яка враховує невідомі фактори або шум у даних.

Ця модель може бути використана для прогнозування значення Y на основі значення X.

Логістична регресія є важливим методом в інтелектуальній обробці даних, особливо для бінарної та багатокласової класифікації. Формула для логістичної регресії в інтелектуальній обробці даних наведена нижче:

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \quad (1.2)$$

де p – ймовірність віднесення до класу «1»;

$1 - p$ – ймовірність віднесення до класу «0»;

ln – натуральний логарифм.

$\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ – коефіцієнти регресії, які модель навчається визначати під час тренування.

x_1, x_2, x_n – вхідні ознаки (пояснюючі змінні), які використовуються для передбачення.

Ця формула дозволяє моделі логістичної регресії оцінювати ймовірності віднесення спостережень до певного класу. У відповідності з пороговим значенням (зазвичай 0.5), модель приймає рішення про приналежність до класу «1» або «0». Логістична регресія широко використовується в задачах класифікації, таких як виявлення спаму в електронних листах, медична діагностика, аналіз тексту, рекомендаційні системи, та інші завдання в інтелектуальній обробці даних.

Метод опорних векторів (Support Vector Machines, SVM) – це алгоритм у машинному навчанні, який використовується для класифікації та регресії. Ядро SVM – це пошук гіперплощини (у випадку багаторозмірних даних) або лінії (у випадку дворозмірних даних), яка найкраще розділяє дані на класи або моделює залежність в регресії [13]. Для класифікації задач SVM використовують наступну формулу:

$$f(X) = w * X + b \quad (1.3)$$

де x – вектор ознак прикладу;

w – вектор вагових коефіцієнтів, які потрібно знайти;

b – зсув (скаляр), який також потрібно знайти.

Дані мають бути поділені на два класи, позначені як «1» та «–1». Мета полягає у пошуку гіперплощини, яка найкраще розділяє ці класи. У випадку

класифікації, SVM спробує знайти гіперплощину так, щоб всі приклади одного класу мали значення f(X) більше 1, а всі приклади іншого класу мали значення менше –1. Іншими словами:

Якщо $y=1$, то $f(X)\geq 1$;

Якщо $y=-1$, то $f(X)\leq -1$;

y – класова мітка прикладу, де $y=1$ або $y=-1$.

Нижче наведено рисунок на якому зображена максимально розділова гіперплощина та межі для ОВМ, натренованої зразками з двох класів. Зразки на межах називаються опорними векторами.

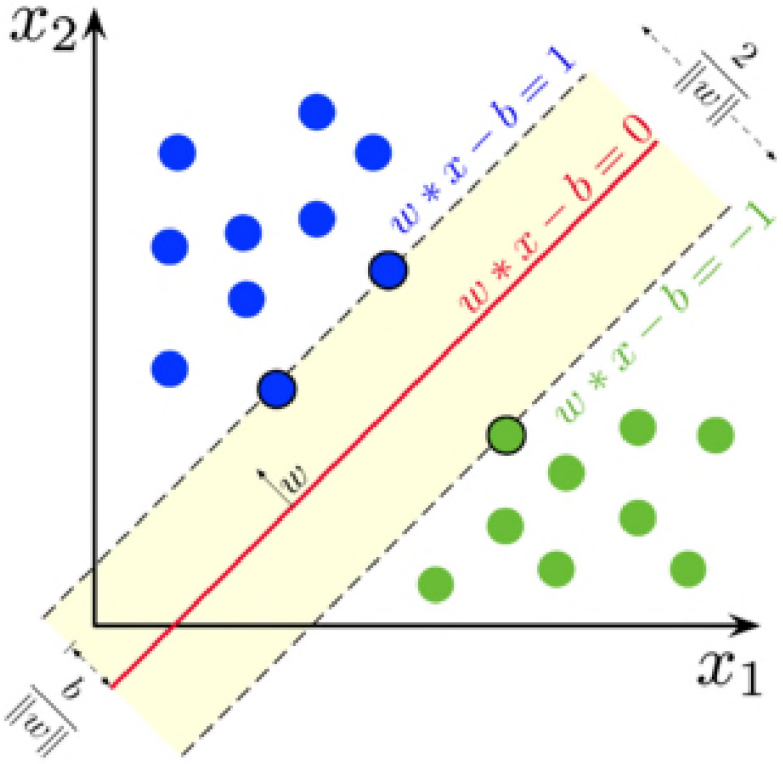

Рисунок 1.2 – Візуалізація методу опорних векторів

Ці обмеження можна представити в формі математичних нерівностей. Ідея полягає у виборі гіперплощини (вектора w та скаляра b), яка максимізує відстань між найближчими прикладами обох класів (це відомо як ширина коридору або «маржа»). Це оптимізаційне завдання може бути сформульоване як задача квадратичного програмування (Quadratic Programming, QP), де метою є знайти вектор w та скаляр b, щоб максимізувати маржу, а також задовольнити обмеження

нерівностей. Зазвичай, SVM використовується з ядровими функціями для вирішення нелінійних проблем, але базова ідея формули залишається такою ж.

Метод k-найближчих сусідів (k-NN) – це простий алгоритм у машинному навчанні, який використовується для класифікації та регресії на основі найближчих сусідів у просторі ознак. Розглянемо обидві формули [14].

Якщо алгоритм використовується для класифікації, то суть полягає в знаходженні k найближчих прикладів тренувального набору до тестового прикладу, використовуючи відстань (зазвичай Евклідову відстань) а також у визначенні класу, який найчастіше зустрічається серед цих k найближчих сусідів [15]. Це стає передбаченням для тестового прикладу Формула *k-NN* для класифікації може бути подана наступним чином:

$$\hat{y} = arg_{max}\left(\sum_{i=1}^{k} I(y_i = j)\right) \tag{1.4}$$

де $\hat{y}$ – передбачене значення класу для тестового прикладу;

y_i – клас i–го найближчого сусіда;

j – клас, який має найбільшу кількість представників серед k найближчих сусідів;

$I()$ – функція індикатора, яка повертає 1, якщо умова виконується, і 0 в іншому випадку.

Якщо алгоритм використовується для регресії, то суть полягає в знаходженні k найближчих прикладів тренувального набору до тестового прикладу, використовуючи відстань та обчислення середнього значення цільової змінної для цих k найближчих сусідів. Це стає передбаченням для тестового прикладу. Формула *k-NN* для регресії виглядає так:

$$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i \tag{1.5}$$

де $\hat{y}$ – передбачене значення цільової змінної для тестового прикладу;

y_i – значення цільової змінної i-го найближчого сусіда;

k – кількість найближчих сусідів.

Загалом, метод *k-NN* використовує простий принцип голосування або усереднення, в залежності від того, чи ми вирішуємо задачу класифікації чи регресії, для визначення передбачення на основі k найближчих сусідів в просторі ознак.

Навчання з підкріпленням (Reinforcement Learning, RL) – це галузь машинного навчання, в якій агент взаємодіє з динамічною середовищем і навчається вибирати дії, що максимізують нагороду (або мінімізують витрати) з часом [16]. Для прикладу, формула Беллмана для оновлення функції цінності в навчанні з підкріпленням може виглядати наступним чином:

$$V(s_t) = max_a(r_t + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a)V(s_{t+1})) \quad (1.6)$$

де $V(s_t)$ – функція цінності стану s_t;

r_t – нагорода, отримана за дію a_t у стані s_t;

γ – дисконтний фактор, який визначає важливість майбутніх нагород;

$P(s_{t+1}|s_t, a)$ – функція переходу, що вказує ймовірність переходу в стан s_{t+1} після виконання дії a у стані s_t.

Ця формула вказує на те, що функція цінності в кожному стані повинна бути оновлена на основі отриманої нагороди та очікуваної майбутньої нагороди.

2. Види задач. Машинне навчання використовується для класифікації, регресії, кластеризації, виявлення аномалій та інших завдань. Кожна з цих задач вимагає використання відповідного алгоритму.

Класифікація – це одна з основних задач в машинному навчанні, де модель навчається призначати кожному вхідному прикладу одну з попередньо визначених міток або класів на основі вхідних ознак. У задачах класифікації, іншими словами, ми намагаємося побудувати модель, яка може розділити дані на класи, таким чином, щоб вона могла правильно класифікувати нові, раніше невидані приклади [17]. Задача класифікації має велику кількість застосувань, включаючи виявлення шахрайства, класифікацію тексту, розпізнавання об’єктів у зображеннях та багато інших. Різні алгоритми які були розглянуті вище, такі як логістична регресія,

дерева рішень, метод опорних векторів (SVM), та багато інших, використовуються для розв'язання задач класифікації залежно від конкретного завдання.

Кластеризація – це задача машинного навчання, в якій модель намагається групувати схожі об'єкти разом у відповідних кластерах чи групах, без заздалегідь визначених міток чи категорій. Цей процес може бути корисним для виявлення природних структур у наборі даних, аналізу подібностей та різниць між об'єктами, а також для інших завдань, таких як рекомендація товарів та скорочення розмірності даних. Кластеризація може бути важливим етапом в аналізі даних, дослідженні споживачів та багатьох інших областях. Вона дозволяє виділити природні групи об'єктів і розкрити структуру в даних, що може бути важливою інформацією для прийняття рішень.

Виявлення аномалій в машинному навчанні – це задача ідентифікації несподіваних, надзвичайних або виокремлюваних подій або об'єктів, які відрізняються від більшості даних. Ця задача також іноді називається виявленням викидів або виявленням аномалій. Виявлення аномалій застосовується в різних областях, включаючи кібербезпеку, фінанси, медицину, виробництво та багато інших. Приклади алгоритмів для виявлення аномалій включають в себе методи на основі відстані (наприклад, k-найближчі сусіди), гаусівські моделі, метод опорних векторів (SVM), LOF та багато інших. Вибір алгоритму залежить від конкретної задачі та характеру даних.

3. Оцінка та підбір моделей. Важливо оцінювати та підбирати найкращі моделі, використовуючи метрики, такі як точність, відзив, точність, F1–показник та інші. Це допомагає визначити якість моделі. Оцінка та підбір моделей є важливими етапами в машинному навчанні для забезпечення створення найкращих можливих моделей для конкретного завдання. Розглянемо кілька основних кроків та понять, пов'язаних із цими процесами:

Для того щоб провести оцінку моделі можна здійснити такі дії, як наприклад метрика якості. Для цього потрібно вибрати відповідні метрики для оцінки моделі, в залежності від типу завдання (наприклад, середньоквадратична помилка для задач регресії, точність для задач класифікації). Також можна використати

розділення даних. Потрібно розділити набір даних на тренувальний, перевірочний і тестовий набори. Тренувальний набір використовується для навчання моделі, а перевірочний для налаштування гіперпараметрів та оцінки внутрішньої продуктивності моделі, тестовий – для оцінки зовнішньої продуктивності моделі. Перехресна перевірка (Cross-Validation). Використовується метод перехресної перевірки для об'єктивної оцінки моделі. Це включає в себе використання кількох розбиттів на навчання та перевірку для зменшення перенавчання. Графіки та візуалізація. Використовують графіки та візуалізацію для аналізу результатів моделі та визначення можливих проблем [18].

Для того щоб здійснити підбір моделі можна провести такі дії як вибір алгоритмів: потрібно вибрати алгоритми, які відповідають типу завдання (регресія, класифікація, кластеризація тощо). Налаштування гіперпараметрів: потрібно налаштувати гіперпараметри моделі, такі як швидкість навчання, кількість шарів та нейронів у нейронних мережах, кількість дерев у випадкових лісах тощо. Пошук гіперпараметрів: полягає у використанні методів пошуку гіперпараметрів, таких як решта, пошук по сітці, оптимізація гіперпараметрів для знаходження оптимальних значень.

Машинне навчання в інтелектуальній обробці даних допомагає розуміти, аналізувати та використовувати велику кількість інформації для вирішення різноманітних завдань та прийняття рішень в багатьох галузях, включаючи бізнес, науку, медицину, фінанси та інші.

1.3 Огляд методів статистичного аналізу для виняткової обробки даних

Статистичний аналіз важливий для обробки та розуміння даних. Це допомагає виявити статистичні залежності, патерни та важливі властивості даних. Нижче наведені деякі методи статистичного аналізу.

Описова статистика являє собою гілку статистичного аналізу, яка використовується для підсумовування та опису важливих характеристик набору

даних. Ці характеристики допомагають отримати загальне уявлення про розподіл даних і роблять можливими порівняння та виявлення патернів в даних. До описової статистики можна включити середнє значення, медіану, моду, розкид, квантилі та квартилі, діапазони, діаграми розподілу, графіки розсіювання. Цей метод є першим кроком у аналізі даних та допомагає зрозуміти їхню структуру та характеристики. Вона допомагає виявити основні патерни, незвичайності та генерувати гіпотези для подальшого статистичного аналізу [19].

Інференційна статистика – це галузь статистичного аналізу, яка вивчає процес висновків або узагальнень на основі обмеженої вибірки даних з метою робити висновки про загальну популяцію. Ця галузь статистики використовується для формулювання та перевірки гіпотез, прийняття рішень та прогнозування на основі вибіркових даних [20]. Основні концепції та методи інференційної статистики включають:

1. Спростування гіпотез. Цей процес включає формулювання нульової гіпотези (наприклад, немає різниці між групами) та альтернативної гіпотези (наприклад, є різниця між групами). Потім застосовується статистичний тест для прийняття чи спростування нульової гіпотези на основі вибіркових даних.

2. Інтервали надійності. Це інтервали, які надають діапазон можливих значень параметра популяції з певною ймовірністю. Вони допомагають оцінити не відомий параметр на основі вибіркових даних.

3. Аналіз дисперсії. ANOVA використовується для порівняння середніх значень трьох або більше груп і визначення, чи є статистично значущі відмінності між ними.

4. Лінійна регресія допомагає вивчити взаємозв'язок між залежною та незалежними змінними, дозволяючи прогнозувати значення залежної змінної на основі незалежних змінних.

5. Байєсівська статистика, що базується на теоремі Байєса і використовує апріорні імовірності для оцінки апостеріорних імовірностей. Вона корисна для розв'язання проблем з невеликими вибірками та враховує попередні знання.

6. Перевірка нормальності. Вона використовується для визначення, чи розподіл даних наближений до нормального розподілу, що є важливим при використанні багатьох статистичних методів.

7. Логістична регресія. Цей метод використовується для бінарної та багатокласової класифікації та робить прогнози ймовірностей на основі незалежних змінних.

8. Випадкові вибірки та розподіли, засновані на випадкових вибірках та розподілах вони допомагають оцінити параметри популяції.

9. Інференційна статистика допомагає вченим, дослідникам та приймальникам рішень робити узагальнення та висновки на основі обмеженої інформації, забезпечуючи статистично обґрунтовані результати та оцінки ризиків.

Наступним методом є кореляція та регресія. Це методи що використовуються для визначення взаємозв'язків між змінними. Коефіцієнти кореляції вказують на силу та напрямок зв'язку, тоді як регресія дозволяє прогнозувати значення однієї змінної.

Кореляція вказує на ступінь зв'язку між двома змінними. Вона вимірює, наскільки змінна «*X*» і змінна «*Y*» змінюються разом. Якщо вони змінюються в тому ж напрямку, кореляція позитивна; якщо вони змінюються в протилежному напрямку, кореляція негативна. Для вимірювання кореляції використовуються різні коефіцієнти, найпоширеніший з яких – це коефіцієнт кореляції Пірсона. Інші включають кореляцію Спірмена та Кендалла. Кореляція допомагає визначити, чи існує статистично значущий зв'язок між двома змінними. Вона використовується в економіці, науці, соціології, медицині та інших галузях для вивчення залежностей між факторами.

Регресія використовується для вивчення взаємозв'язку між залежною змінною (відгуком) і однією або декількома незалежними змінними (пояснюючими факторами). Вона допомагає побудувати модель для прогнозування значень відгуку на основі значень пояснюючих факторів. Існують різні види регресії, включаючи лінійну регресію, логістичну регресію, поліноміальну регресію та інші. Лінійна регресія найпоширеніша та використовується для моделювання лінійних

залежностей між змінними. Цей метод застосовується в багатьох галузях, включаючи економіку (економетрія), медицину, біологію, інженерію, соціальні науки і багато інших. Вона дозволяє проводити прогнози та аналізувати вплив різних факторів на величину відгуку.

1.4 Огляд глибоких нейронних мереж та їх використання для обробки даних

Глибоке навчання (Deep Learning) – це підгалузь машинного навчання, яка спрямована на створення та тренування глибоких нейронних мереж для вирішення завдань обробки даних. Глибокі нейронні мережі складаються з багатьох шарів нейронів, які намагаються моделювати складні залежності в даних. На рисунку 1.3 зображено як глибоке навчання є підмножиною машинного навчання і як машинне навчання є підмножиною штучного інтелекту [21].

Рисунок 1.3 – Глибоке навчання як підмножина машинного навчання

Розглянемо основні аспекти дослідження глибокого навчання та їх використання.

1. Глибоке навчання використовує нейронні мережі, такі як згорткові нейронні мережі (CNN), рекурентні нейронні мережі (RNN) та рекурентні нейронні мережі з довготривалим короткостроковим запам'ятовуванням (LSTM), для моделювання різних типів даних, включаючи зображення, текст, звуки та часові послідовності.

Згорткові нейронні мережі є особливим типом нейронних мереж, спеціально призначеними для обробки зображень та відео, вони стали основним інструментом у галузі комп'ютерного бачення і вирішенні завдань, пов'язаних з обробкою зображень, від розпізнавання облич до автоматичного опису зображень. Згорткові нейронні мережі можна розглянути з точки зору їх складових. Першою складовою є згорткові шари, вони використовуються для виявлення різних ознак та шаблонів у вхідних зображеннях, ці шари використовують фільтри (які називаються ядрами) для проведення операції згортки, що допомагає виділяти локальні зразки у зображенні. Наступною складовою є шари пулінгу, вони використовуються для зменшення розмірів репрезентації, зберігаючи при цьому важливі інформаційні ознаки, зазвичай використовується операція підсумовування (MaxPooling або AveragePooling). Повнозв'язані шари: після згорткових та пулінг-шарів зазвичай слідують повнозв'язані шари, які використовуються для класифікації або регресії, вони приймають репрезентацію з згорткових шарів та генерують вихідну інформацію. Згорткові шари зазвичай включають функції активації, такі як ReLU (Rectified Linear Unit), для введення нелінійності в модель та покращення її здатності до навчання складних залежностей. Щоб уникнути перенавчання, до згорткових нейронних мереж можна додати методи регуляризації, такі як випадковий вимкнення (Dropout) або обмеження ваг (Weight Regularization). Згорткові нейронні мережі демонструють вражаючі результати в багатьох завданнях комп'ютерного бачення. Вони також знайшли застосування у галузі обробки природної мови для аналізу тексту та у галузі рекомендаційних систем для покращення рекомендацій на основі зображень.

Рекурентні нейронні мережі – це клас нейронних мереж, які призначені для обробки послідовних даних, де поточний вихід залежить від попередніх вхідних даних та попереднього вихідного стану. Вони здатні моделювати залежності в послідовних даних та використовуються в широкому спектрі завдань, таких як обробка природної мови, машинний переклад, генерація тексту, аналіз часових рядів та багато інших. Рекурентні мережі мають рекурентні шари, які здійснюють повторні взаємозв'язки між часовими кроками, кожен рекурентний шар приймає вхід від поточного часового кроку та вихід від попереднього часового кроку. Кожен рекурентний шар має прихований стан, який зберігає інформацію про попередні часові кроки та використовує її для розуміння контексту в послідовних даних, для створення нелінійності в мережі в рекурентних шарах використовуються функції активації, такі як гіперболічний тангенс (tanh) або логістика (sigmoid). Деякі покращені рекурентні шари, такі як LSTM і GRU, були розроблені для уникнення проблем з надмірним затуханням градієнту, які можуть виникнути під час навчання на довгих послідовностях. Рекурентні мережі можуть мати багатошарову структуру, що дозволяє виразити складні залежності в даних. Рекурентні нейронні мережі зазвичай використовуються для послідовних даних, таких як текст, мовлення, часові ряди та послідовності даних. Вони можуть здійснювати завдання, такі як розпізнавання мовлення, машинний переклад, генерація тексту та прогнозування часових рядів [22].

2. Автоматичне навчання – це аспект глибокого навчання, який спрямований на автоматизацію процесу створення та навчання глибоких нейронних мереж та інших моделей машинного навчання. Головна мета автоматичного навчання полягає в тому, щоб зробити процес розробки моделей машинного навчання доступним та ефективним для більш широкого кола користувачів, навіть тих, хто не є експертами в глибокому навчанні або машинному навчанні загалом. Системи автоматичного навчання можуть визначити найкращі моделі для конкретного завдання. Вони можуть розглядати різні архітектури нейронних мереж, включаючи CNNs, RNNs, LSTM, GRU, та інші, і вибирати ту, яка найкраще підходить для вирішення завдання. Гіперпараметри (наприклад, кількість шарів, розмір кроку

навчання, розмір пакету) можуть суттєво впливати на ефективність моделі. Автоматичне навчання може автоматично визначити оптимальні гіперпараметри для покращення результатів моделі. Вибір підходящих ознак або їх створення може покращити результати моделі. Автоматичне навчання допомагає автоматизувати цей процес. Автоматичне навчання може включати пошук оптимальних гіперпараметрів для задачі. Це дозволяє моделі налаштуватися на конкретний набір даних і завдання. Системи автоматичного навчання можуть відслідковувати ефективність моделі та автоматично вирішувати, коли потрібно зупинити процес навчання, коли результати стають насиченими. Після створення та навчання моделі системи автоматичного навчання можуть автоматично розгортати модель для використання в реальних застосуваннях. Автоматичне навчання значною мірою спрощує процес створення та оптимізації моделей машинного навчання, зменшуючи необхідність в глибокому розумінні технічних аспектів машинного навчання. Воно робить машинне навчання більш доступним та швидким, що особливо важливо в умовах зростання інтересу та популярності глибокого навчання.

3. Зображення та комп'ютерне бачення є важливими аспектами глибокого навчання, оскільки ця технологія дозволяє автоматизовано аналізувати та розуміти великі обсяги візуальної інформації, зокрема фотографії, відео, зображення з дронів та багато інших видів візуальних даних. Глибоке навчання використовується для обробки і розуміння зображень та відео, це включає в себе завдання, такі як розпізнавання облич, об'єктів, класифікація зображень, детекція об'єктів та сегментація зображень. Згорткові нейронні мережі є основою багатьох систем комп'ютерного бачення, вони використовуються для виявлення ознак та патернів у вхідних зображеннях та відео. Також використовуються рекурентні нейронні мережі для обробки послідовних візуальних даних, таких як часові ряди зображень або відео. Глибоке навчання може використовуватися для сегментації зображень, що означає виділення та класифікацію окремих об'єктів або областей на зображеннях. це включає обробку фотографій, зображень з веб-камер та інших джерел відео, застосування включають визначення рухів, фільтрацію шуму та

відстеження об'єктів. Глибоке навчання використовується для машинного перекладу тексту в зображення або навпаки, що дозволяє створювати автоматизовані системи опису зображень або генерації зображень на основі текстового опису. Зображення та комп'ютерне бачення в глибокому навчанні є важливими, оскільки дозволяють системам аналізувати навколишній світ, розуміти візуальні дані та вирішувати багато завдань у реальному часі.

4. Обробка природної мови – це галузь глибокого навчання та машинного навчання, спрямована на аналіз, розуміння та генерацію людської мови. Обробка природної мови включає аналіз текстової інформації, включаючи текстовий розпізнавання, синтаксичний та семантичний аналіз тексту. Іншим аспектом є мовне розпізнавання, яке полягає в перетворенні звукових сигналів, наприклад, мовлення, у текстову інформацію, глибоке навчання застосовується для покращення точності мовного розпізнавання. Даний метод включає в себе також визначення мови, на якій написаний текст, і може бути важливим для міжнародних програм та послуг, для визначення тональності тексту, наприклад, для визначення, чи є відгук позитивним, негативним чи нейтральним, для автоматичного перекладу тексту з однієї мови на іншу тобто він може використовувати глибоке навчання для покращення точності перекладу. Глибоке навчання дозволяє створювати системи, які генерують текст, включаючи автоматичні відповіді, текстові описи зображень та авторозширення тексту, це дозволяє наприклад створювати системи обробки природної мови що можуть відповідати на питання, задані людьми в текстовій формі, шляхом аналізу тексту та пошуку відповідей. В області медицини обробка природної мови використовується для аналізу медичних записів, виявлення патологій та іншого текстового контенту. Обробка природної мови є важливою для розуміння та взаємодії з людьми через текстові дані та мовлення. Вона знаходить застосування в багатьох сферах.

5. Рекомендаційні системи є важливим застосуванням глибокого навчання та інтелектуальної обробки даних. Вони допомагають рекомендувати користувачам вміст, такий як продукти, статті, фільми, музику та інші об'єкти, на основі їхніх інтересів та попередньої поведінки. Цей метод включає в себе використання

глибоких нейронних мереж для створення векторних представлень об'єктів та користувачів, що дозволяє відобразити їхні вподобання та властивості у векторному просторі, ці вбудовування використовуються для розрахунку рекомендацій. Для прогнозування рейтингів або ймовірності взаємодії користувачів з об'єктами створюються моделі ранжування, які використовують глибокі нейронні мережі і допомагають визначити порядок рекомендацій. В рекомендаційних системах для вмісту, що містить зображення або відео, використовуються глибокі згорткові мережі для аналізу візуальної інформації, в свою чергу рекурентні нейронні мережі використовуються для моделювання послідовностей в рекомендаційних системах, наприклад, для рекомендацій текстового контенту або музики. Також рекомендаційні системи використовують особисті дані користувачів, такі як історія переглядів або рейтинги, для створення персоналізованих рекомендацій. Застосування рекомендаційних систем в глибокому навчанні дозволяє покращити якість рекомендацій та забезпечує більш ефективну взаємодію користувачів з різноманітними видами вмісту. Рекомендаційні системи грають важливу роль в бізнесі та розвагах, забезпечуючи користувачів більш настроєним та персоналізованим досвідом.

6. Автокодування – це клас нейронних мереж в глибокому навчанні, які використовуються для зменшення розмірності даних шляхом стиснення та після цього відновлення вхідних даних з цього стиснутого представлення. Генерація даних в глибокому навчанні може використовувати автокодування для створення нових даних на основі вже існуючих. Автокодування складається з двох основних частин енкодера (encoder) та декодер (decoder), енкодер перетворює вхідні дані в стиснуте представлення (код), а декодер відновлює дані з цього коду. Стиснене представлення, також відоме як код, містить важливі ознаки або змінні, які представляють вхідні дані, цей код може бути використаний для реконструкції даних або генерації нових. Важливим аспектом є функція втрати, яка визначає, наскільки точно вихід моделі відповідає вхідним даним, тренування автокодування полягає в мінімізації цієї функції. Автокодування може бути використане для генерації нових даних шляхом генерації нових кодів та відновлення цих кодів у

вхідних даних, може бути використане для генерації нових зображень або аудіофайлів, нейронні мережі можуть навчитися генерувати зображення, які подібні до навчальних прикладів. Автокодування та їх різні варіації знаходять застосування в різних галузях, включаючи комп'ютерне бачення, обробку природної мови, генерацію тексту, музику та багато інших. Автокодування в глибокому навчанні є потужним інструментом для стискання та генерації даних. Вони знаходять застосування в різних сферах, де необхідно аналізувати та генерувати дані з ефективністю та точністю.

У підсумку можна сказати що аналіз методів та технологій інтелектуальної обробки даних показує, що ця галузь великої важливості та активно розвивається в останні роки. Інтелектуальна обробка даних включає в себе широкий спектр методів, включаючи машинне навчання, глибоке навчання, обробку природної мови, комп'ютерне бачення, обробку сигналів, кластеризацію, класифікацію, та багато інших. Це робить інтелектуальну обробку даних потужним інструментом для аналізу та вивчення даних. Методи інтелектуальної обробки даних знайшли застосування в багатьох галузях, включаючи медицину, фінанси, транспорт, маркетинг, наукові дослідження, інтернет-пошук та багато інших. Вони допомагають вирішувати важливі завдання та покращують прийняття рішень. З розвитком інтелектуальної обробки даних з'явилися потужні бібліотеки та фреймворки, які дозволяють дослідникам та розробникам швидко розробляти та навчати моделі. У загальному, інтелектуальна обробка даних є важливою галуззю, що активно розвивається та має великий потенціал для вирішення складних завдань та розуміння даних. Для успішного використання цих методів необхідно уважно обирати відповідні технології та враховувати етичні та правові аспекти.

Висновки до розділу 1

В ході аналізу літературних джерел та дослідження основних методів та алгоритмів інтелектуальної обробки даних було виявлено ряд ключових висновків.

По-перше, велика різноманітність доступних методів, таких як машинне навчання, глибоке навчання, обробка природної мови та кластеризація, дозволяє вибирати підходи, які найкраще відповідають конкретним завданням або галузям застосування. По-друге, важливим аспектом є тенденція розвитку інтелектуальної обробки даних в напрямку глибшого використання нейронних мереж та моделей глибокого навчання. Це відкриває нові можливості для вирішення завдань, які раніше були складні або навіть неможливі для традиційних методів. По-третє, важливим аспектом є проблеми, пов'язані з етикою та безпекою в інтелектуальній обробці даних. Забезпечення конфіденційності, запобігання прийняттю рішень на основі невірних або вкрай спрощених даних, а також врахування соціокультурних аспектів стають важливими завданнями для подальшого розвитку цієї галузі. В цілому, дослідження підтверджує актуальність використання інтелектуальної обробки даних та вказує на необхідність подальших досліджень у напрямку оптимізації алгоритмів, розширення областей їх застосування та вирішення етичних та безпекових питань.

Проаналізувавши сукупність методів, була поставлена задача, яка полягає у розробці навчальних матеріалів для вирішення завдань класифікації, кластеризації та регресії. Для цього завдання був обраний програмний засіб Weka, детальніше про який йдеться у наступному розділі. Для реалізації поставленого завдання було обрано метод машинного навчання із використанням розглянутих в цьому розділі алгоритмів.

РОЗДІЛ 2
ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ІНТЕЛЕКТУАЛЬНОЇ ОБРОБКИ ДАНИХ

2.1 Вибір програмного засобу для виконання поставленого завдання, обгрунтування вибору

Згідно з поставленим завданням у кінці попереднього розділу, було обрано програмний засіб який буде використаний як інструмент для інтелектуальної обробки даних. Weka – це бібліотека алгоритмів машинного навчання, призначена для вирішення завдань інтелектуального аналізу даних. Ця система дозволяє застосовувати алгоритми безпосередньо до вибірок даних і викликати їх з програм на мові Java. Акронім Weka розшифровується як «Waikato Environment for Knowledge Analysis» – середовище для аналізу знань, розроблене у Університеті Вайкато (Нова Зеландія) [23].

Метою створення цього програмного засобу є створення сучасного середовища для розробки методів машинного навчання і їх використання на реальних даних, а також зробити методи машинного навчання доступними для широкого застосування. Передбачається, що завдяки цьому середовищу фахівці в прикладних галузях зможуть витягти корисні знання безпосередньо з даних, можливо, дуже великого обсягу [24]. Користувачами Weka є дослідники в галузі машинного навчання і прикладних наук. Вона також широко використовується в навчальних цілях. Weka містить інструменти для попередньої обробки даних, класифікації, регресії, кластеризації, відбору ознак, пошуку асоціативних правил і візуалізації. Weka відмінно підходить для розробки нових підходів у машинному навчанні. Програма реалізована як відкрите програмне забезпечення, розроблене світовою науковою спільнотою та поширюється під ліцензією GNU GPL. Програмне забезпечення повністю написане на Java. Вихідні дані припускається, що представлені у вигляді матриці ознакових описів об’єктів. Weka надає доступ до SQL-баз даних через Java Database Connectivity (JDBC) і може брати дані з результатів SQL-запитів. Можливість обробки декількох пов’язаних таблиць не

підтримується, але існують інструменти для перетворення таких даних в одну таблицю, яку можна завантажити в Weka [25].

Weka – це потужний і дуже популярний програмний інструмент для аналізу даних та машинного навчання. Вона має кілька переваг, які роблять її привабливою для дослідників та професіоналів в галузі аналізу даних:

1. Безкоштовність та відкритий код. Weka є вільно розповсюджуваним програмним забезпеченням з відкритим вихідним кодом, що означає, що ви можете використовувати її безкоштовно та вносити зміни в код, якщо це потрібно.

2. Широкий вибір алгоритмів. Weka включає в себе велику кількість алгоритмів машинного навчання, від класичних до сучасних. Це дозволяє дослідникам та аналітикам вибирати найбільш підходящий алгоритм для своєї задачі.

3. Графічний інтерфейс. Weka має інтуїтивний графічний інтерфейс, що полегшує роботу з нею, особливо для тих, хто не має глибоких знань у програмуванні.

4. Підтримка для препроцесингу даних. Weka надає різні інструменти для обробки та підготовки даних перед аналізом, включаючи кодування категоріальних ознак, видалення аномалій, нормалізацію даних і багато інших.

5. Розширення та плагіни. Weka дозволяє користувачам додавати розширення та плагіни для розширення її функціональності.

6. Активна спільнота користувачів. Weka має активну спільноту користувачів, яка надає підтримку та ресурси для вирішення питань та проблем.

7. Підтримка різних операційних систем. Weka підтримує різні операційні системи, включаючи Windows, macOS та Linux.

8. Зручність для навчання та освоєння. Weka підходить для навчання та експериментування з алгоритмами машинного навчання без необхідності глибоких технічних знань. Крім того, Weka також підтримує командний рядок для автоматизації завдань машинного навчання [26].

2.2 Огляд основних можливостей програмного засобу Weka

Розглянемо наглядно деякі можливості, які надає програмний засіб для роботи з даними. Основні можливості програми Weka включають:

1. Завантаження даних. Weka дозволяє завантажувати дані з різних джерел, таких як файли CSV, ARFF (формат Weka), бази даних тощо, для прикладу візьмемо файл з що завантажується разом з програмою і знаходиться в директорії data, нехай це буде cpu.rff. Weka також надає інструменти для цих завдань в тому ж інтерфейсі Explorer. На рисунку 2.1 зображений інтерфейс програми та дії які потрібно виконати для завантаження даних.

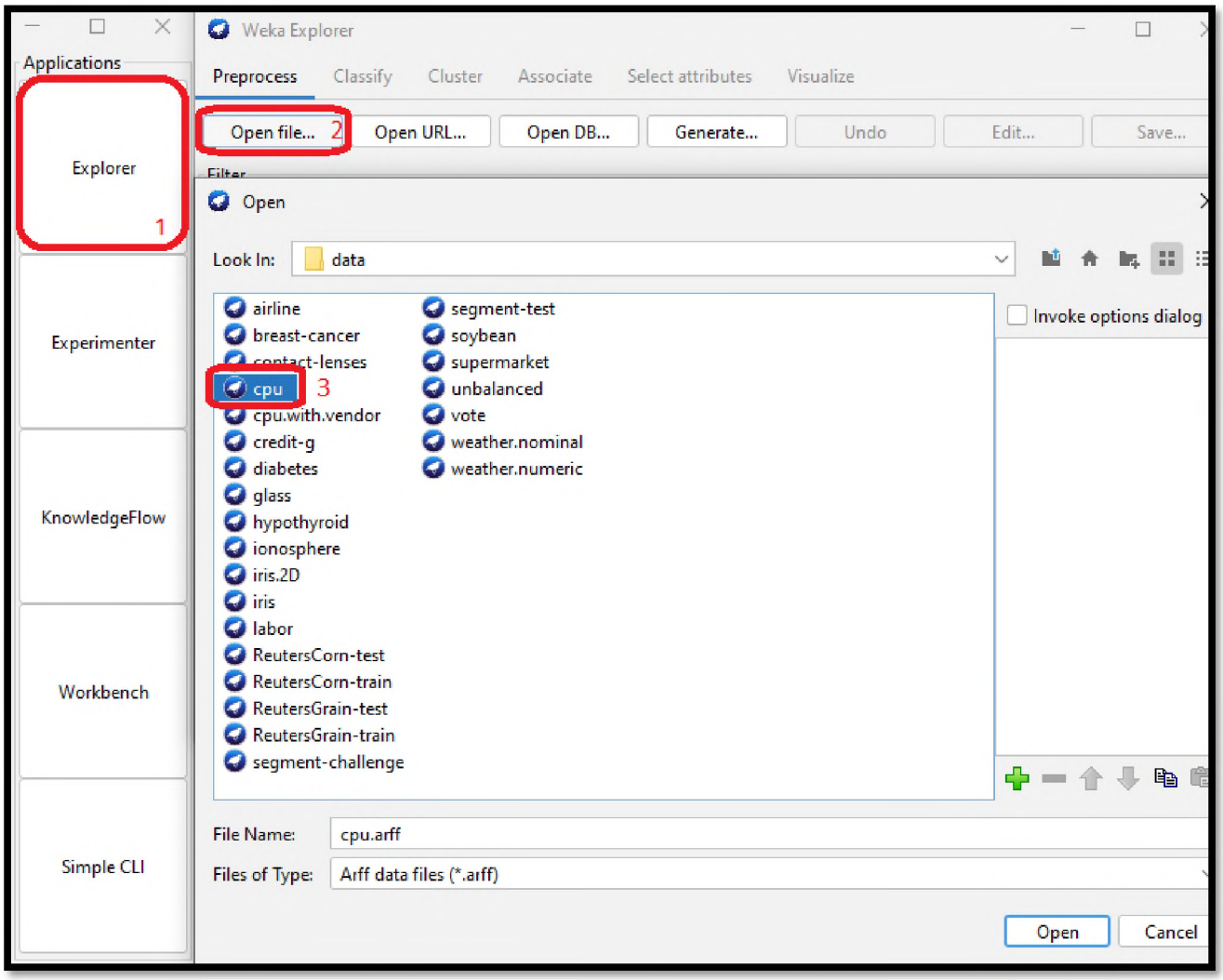

Рисунок 2.1 – Завантаження даних в програму Weka

2. Перегляд та аналіз даних. Після завантаження даних ви можете переглядати дані, вивчати їх статистичні характеристики та візуалізувати дані для отримання кращого розуміння. На рисунку 2.2 зображений інтерфейс програми на якому дані подані у вигляді списку, таблиці та діаграми, це дозволяє дуже легко перемикатись між різними елементами що знаходяться в файлі.

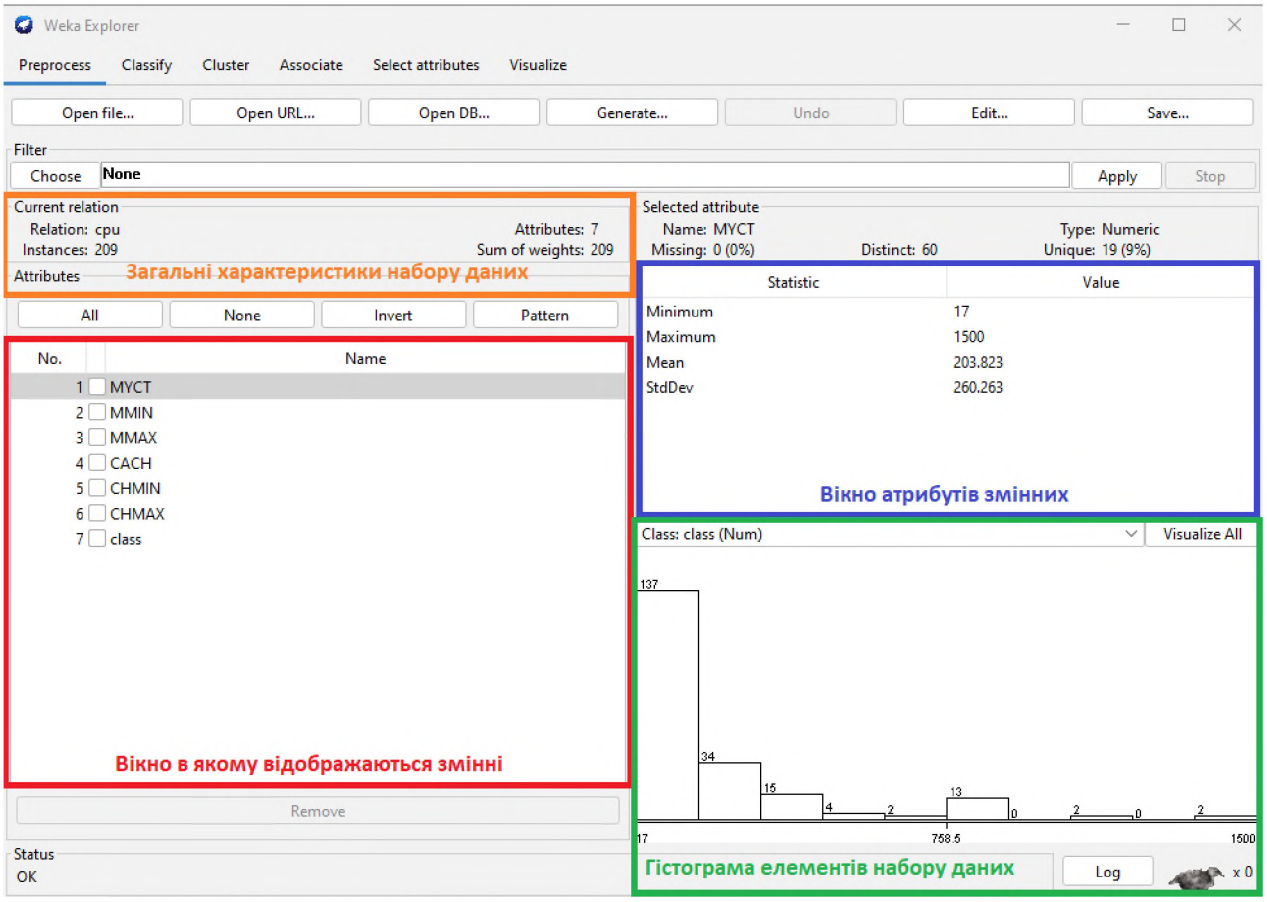

Рисунок 2.2 – Інтерфейс програми Weka для перегляду та аналізу даних

3. Підготовка даних. Програма дозволяє виконувати операції підготовки даних, такі як обрізка, фільтрація, перетворення категоріальних ознак, обробка відсутніх значень та інші операції. Для прикладу застосуємо нормалізацію для даних які ми завантажили Якщо ми поглянемо на інформацію про змінні, то побачимо що значення спільних атрибутів, таких як найменше, найбільше та середнє значення для кожної змінної є різними. Зазвичай для побудови таких моделей як модель регресії або модель класифікації необхідно виконати масштабування, для цього потрібно у вікні відкритого нами файлу у підрозділі

Filtres натиснути кнопку Choose і у випадаючому списку вибрати фільтр Normalaize, потім натиснути Apply. В результаті можемо побачити що значення для всіх змінних прирівнялись, максимальне стало рівним 1 а мінімальне 0. Після цих кроків дані будуть нормалізовані і готові для використання в алгоритмах машинного навчання в Weka. Нормалізація допоможе забезпечити, що значення ваших ознак мають однаковий масштаб і допоможе алгоритмам навчання працювати краще [27]. Результат виконаної нормалізації зображено на рисунку 2.3.

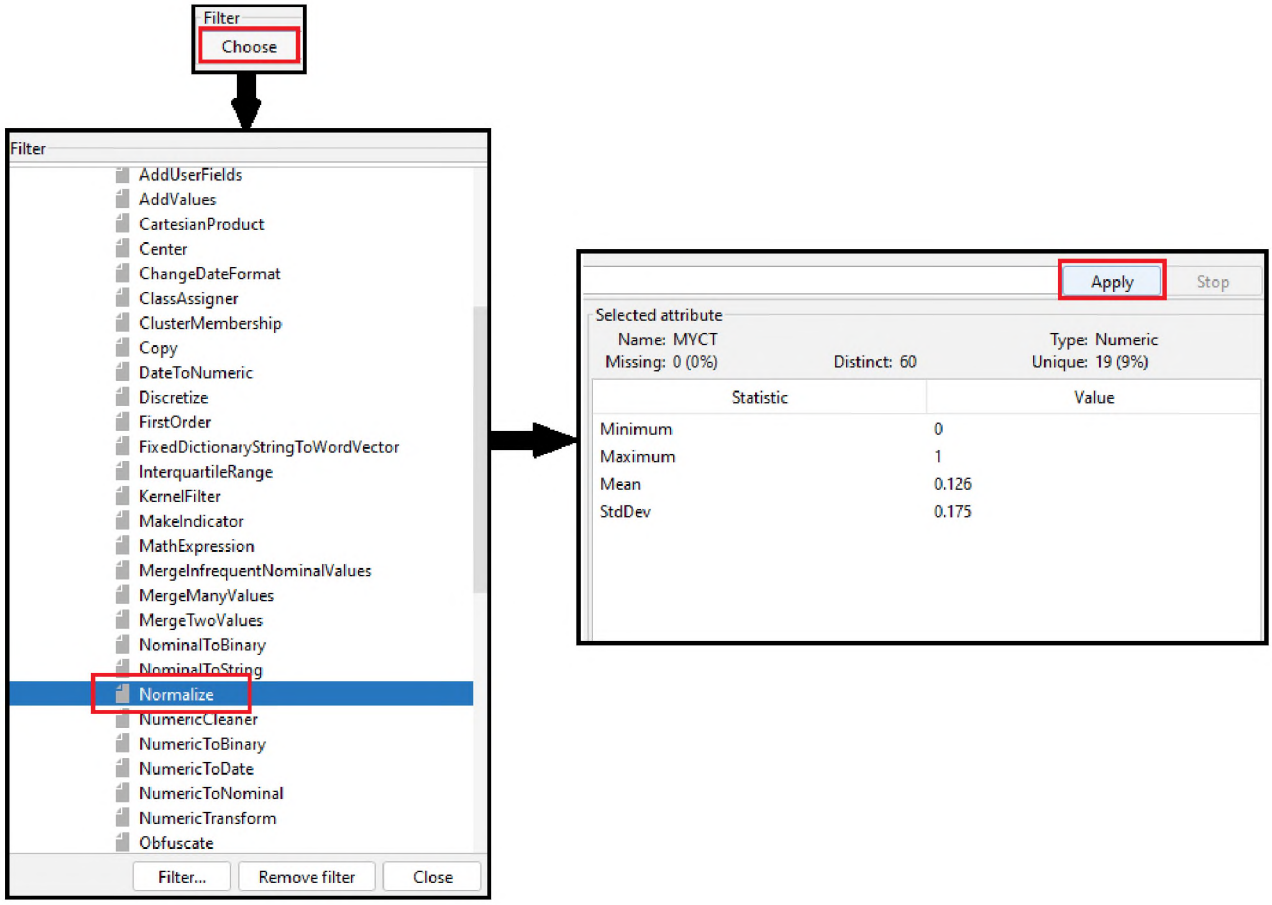

Рисунок 2.3 – Результат виконаної нормалізації

4. Вибір алгоритму машинного навчання, навчання моделей. Weka містить широкий набір алгоритмів машинного навчання для класифікації, регресії, кластеризації та інших завдань. Weka також надає можливість власноруч розширювати функціональність за допомогою плагінів та сторонніх розширень. Вибір алгоритму залежить від конкретної задачі та типу даних, над якими ви працюєте. Для того щоб обрати алгоритм машинного навчання потрібно перейти

на вкладку Classify і обрати бажаний алгоритм із випадаючого списку. Нехай це буде лінійна регресія (LinearRegression), вибравши алгоритм потрібно натиснути кнопку Start, в результаті буде створена модель лінійної регресії [28]. На рисунку 2.4 зображений інтерфейс програми.

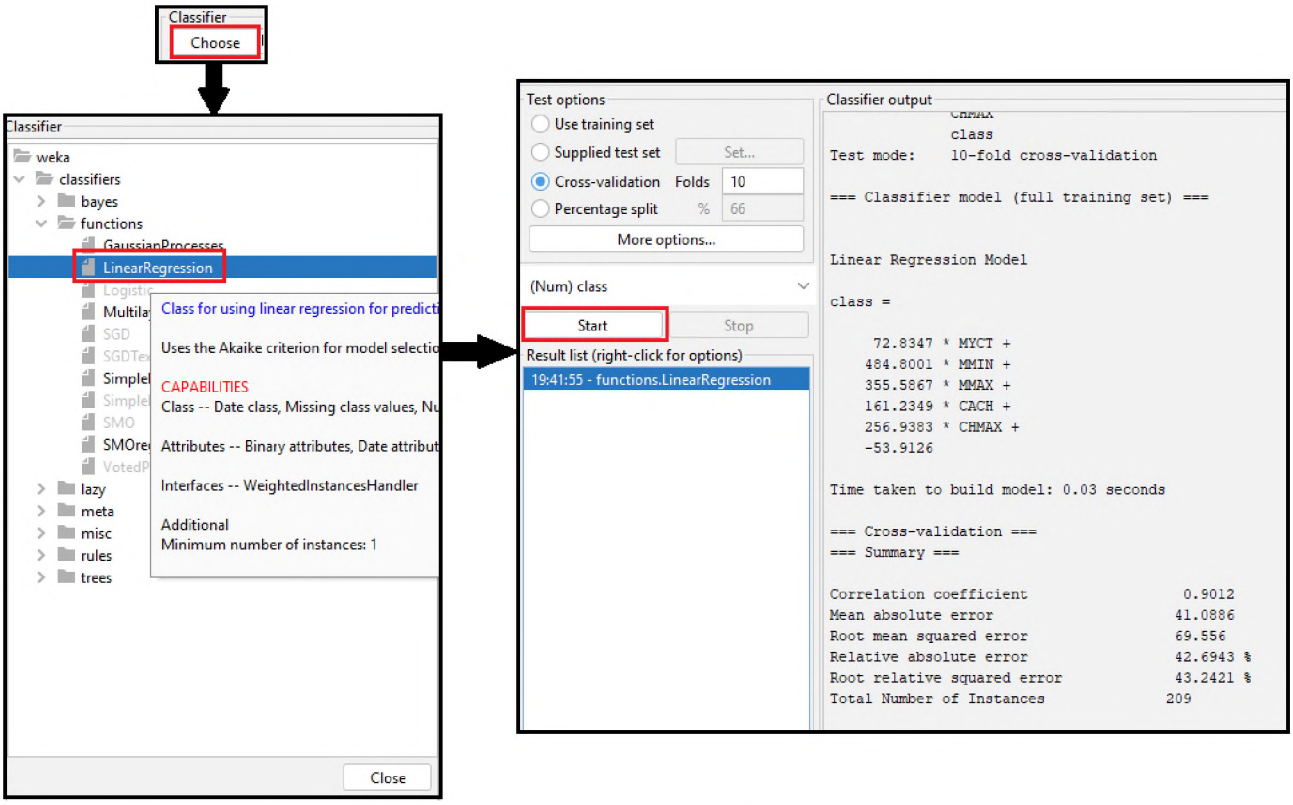

Рисунок 2.4 – Результат створення простої моделі лінійної регресії

5. Оцінка моделей. Оцінка результатів в програмі Weka включає в себе оцінку якості моделі машинного навчання та визначення того, наскільки вона ефективно вирішує вашу конкретну задачу. Для оцінки моделей в Weka важливо обирати відповідні інструменти та метрики, відповідно до вашої конкретної задачі та типу даних, над якими ви працюєте. Weka надає різні інструменти для оцінки результатів моделі. Після побудови моделі програма видасть вам результати навчання моделі, серед яких безпосередньо рівняння лінійної регресії, коефіцієнт кореляції, середньоквадратична помилка та інші. Можна використовувати різні алгоритми і аналізуючи отримані результати робити висновки про те, який алгоритм виявився найефективнішим. Оцінка моделей важлива для визначення

того, наскільки добре модель вирішує вашу конкретну задачу та як її можна покращити [29]. Зображення інтерфейсу програми наведено на рисунку 2.5.

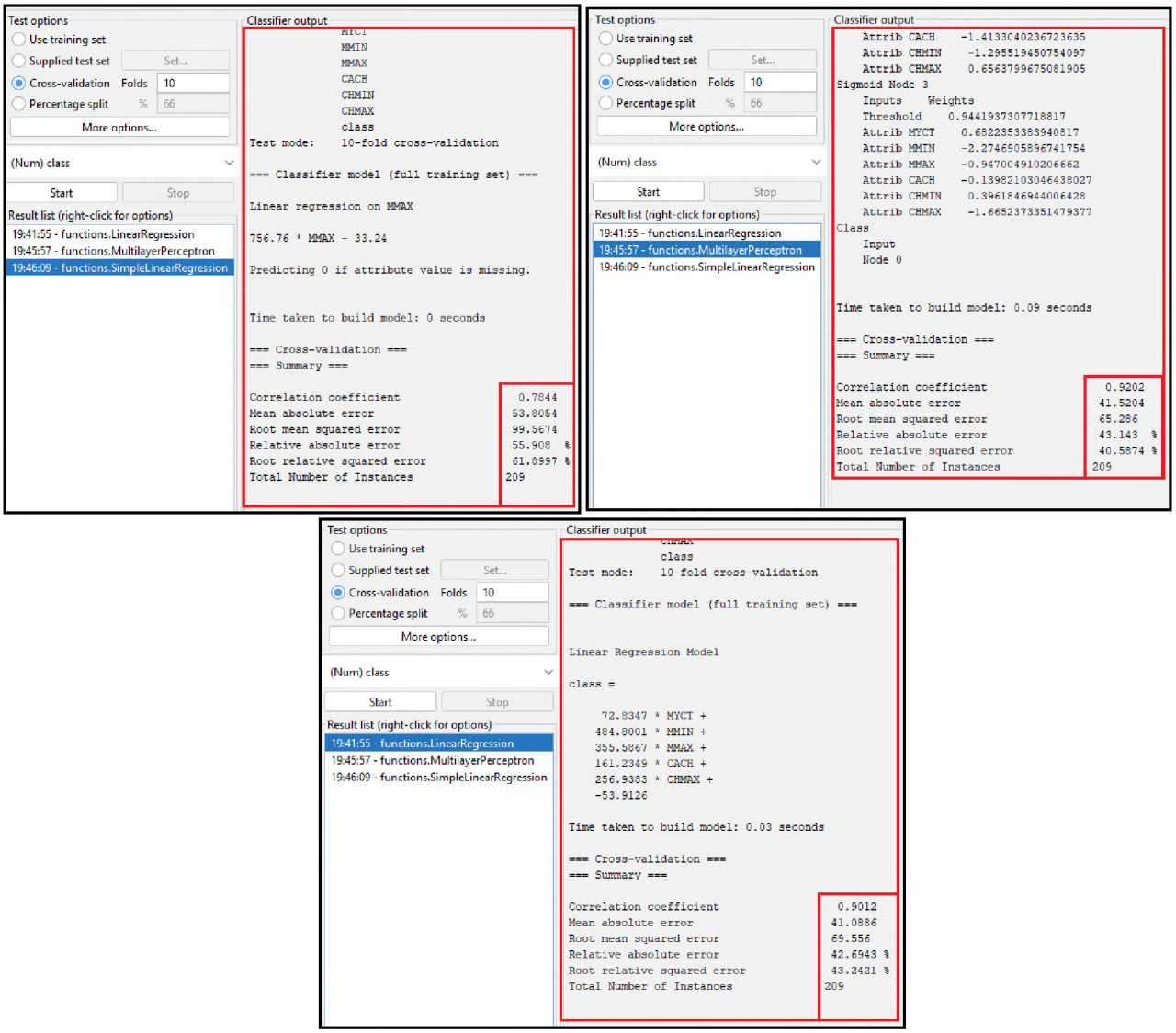

Рисунок 2.5 – Результати навчання моделі в Weka

6. Візуалізація результатів. Програма надає різні можливості візуалізації результатів аналізу даних та навчання моделей машинного навчання. Візуалізація може значно полегшити розуміння ваших даних та результатів моделі. Ось декілька способів візуалізації результатів в Weka:

Графіки розсіювання, які можна створити графіки розсіювання для візуалізації взаємозв'язків між атрибутами у вашому наборі даних. Виберіть підходящі пари атрибутів та створіть графіки розсіювання для їх відображення [30].

Гістограми використовуються для візуалізації розподілу числових атрибутів у наборі даних. Вони допомагають зрозуміти, як розподілені дані.

Weka надає можливість створювати різні графіки, такі як лінійні діаграми, гістограми, кругові діаграми і багато інших для візуалізації результатів.

Відображення дерев рішень використовується якщо ви використовуєте алгоритми, які будують дерева рішень (наприклад, J48 або C4.5), ви можете візуалізувати ці дерева, щоб краще розуміти рішення, які приймає ваша модель.

Представлення кластерів застосовують у разі задачі кластеризації ви можете візуалізувати кластери, які були сформовані внаслідок аналізу.

У нашому випадку ми побачимо діаграми розсіювання для всіх наших змінних, для цього потрібно перейти на вкладку Visualise. Інтерфейс програми зображено на рисунку 2.6.

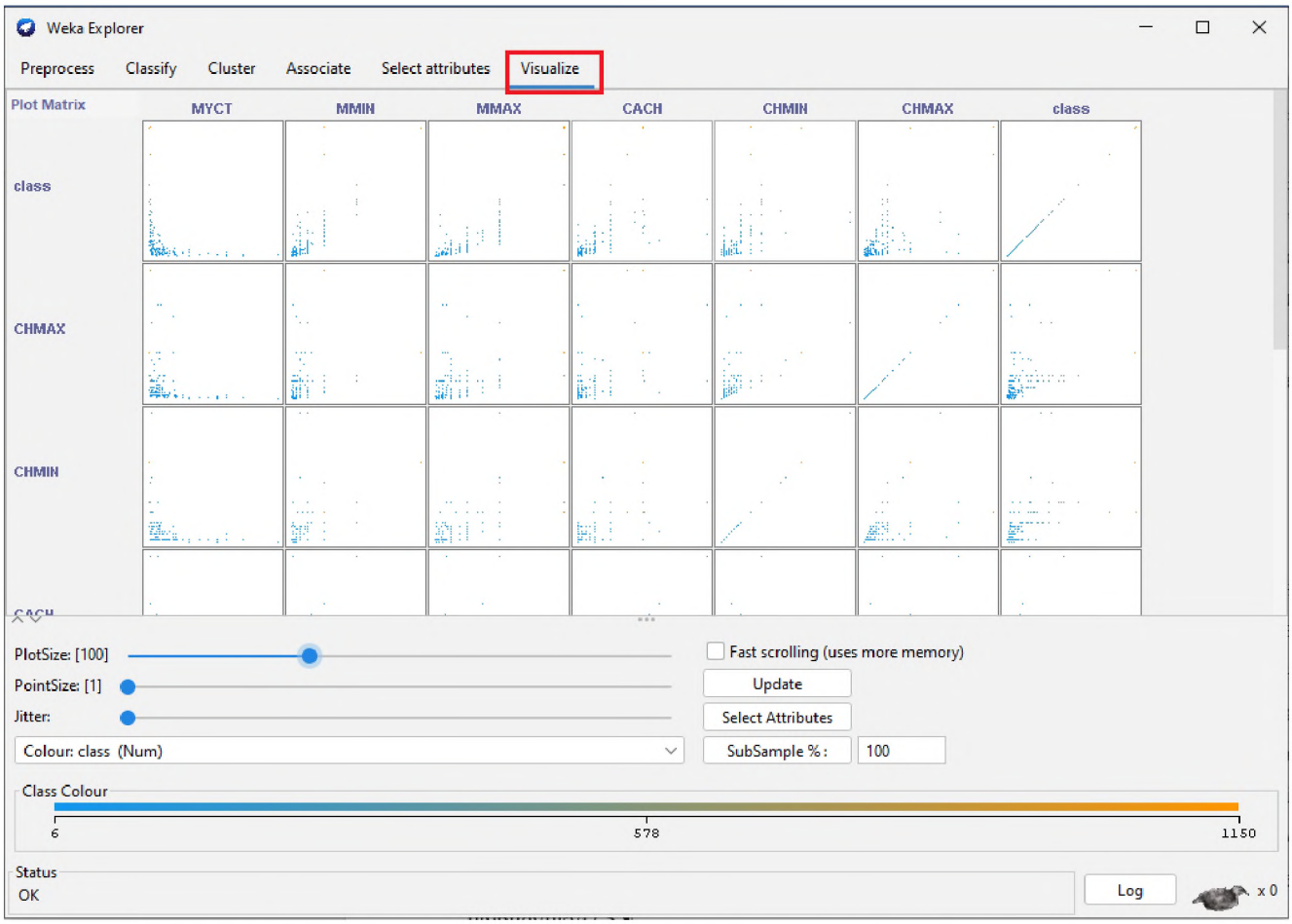

Рисунок 2.6 – Візуалізація результатів машинного навчання в Weka

7. Збереження та завантаження моделей. Якщо ви задоволені результатом навчання, у Weka є можливість зберегти навчену модель. Це корисно, коли ви хочете використовувати раніше навчену модель для класифікації, прогнозування або інших завдань без необхідності повторного навчання. Для цього на вкладці Preprocess потрібно вибрати пункт Save і обрати потрібне розширення файлу (зазвичай це ARFF або модель Weka). Потім можна швидко завантажити навчену модель. Інтерфейс програми зображений на рисунку 2.7.

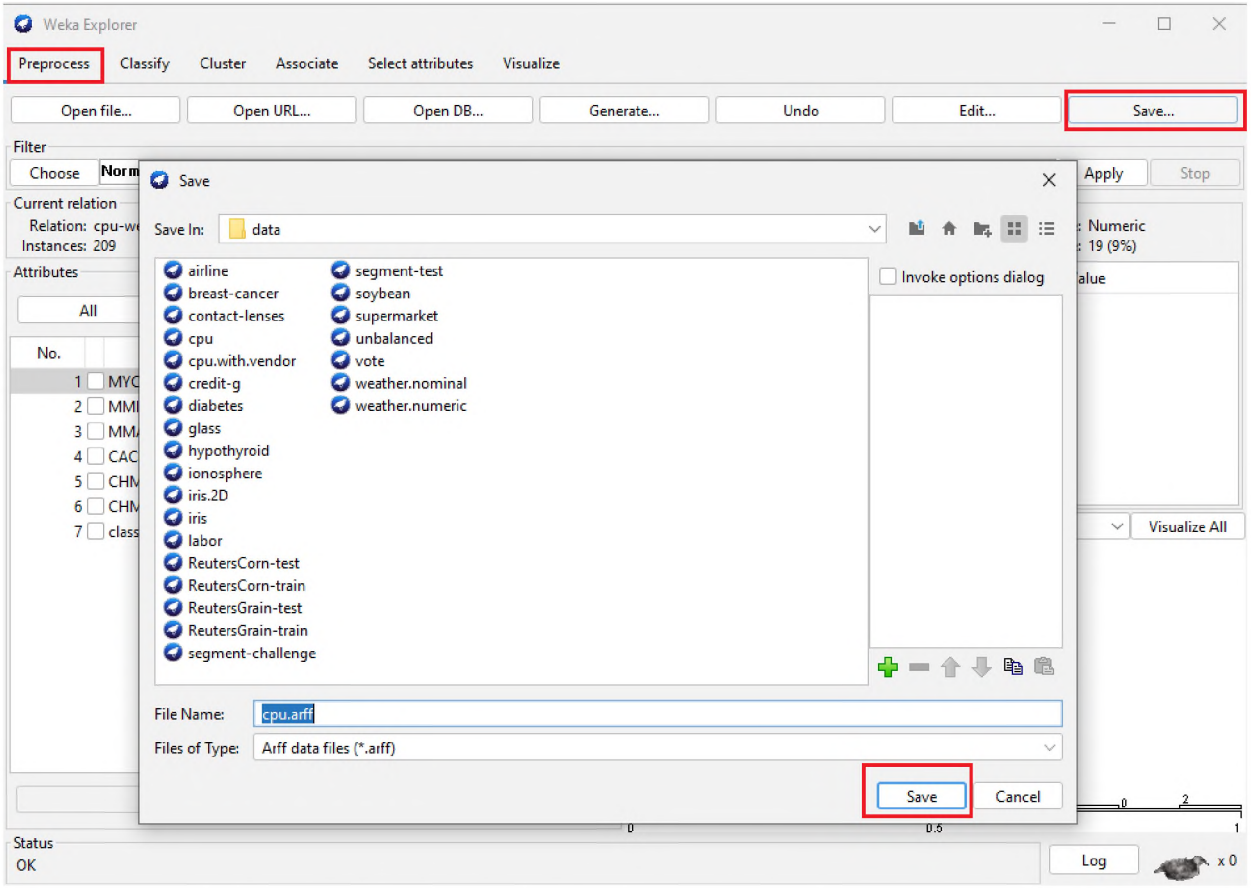

Рисунок 2.7 – Збереження навченої моделі в Weka

Це лише невелика частина можливостей які надає програмний засіб Weka, більше можливостей розглянуто у наступному розділі шляхом розробки лабораторних робіт які базуються на даному програмному забезпеченні. Загалом, Weka є потужним і зручним інструментом для аналізу даних і машинного навчання, який використовується в академічних дослідженнях та в промисловості для вирішення різних завдань аналізу даних.

Висновки до розділу 2

Загалом можна сказати що дослідження функціональних можливостей та можливостей програмного забезпечення Weka дозволило зробити кілька ключових висновків. Перш за все Weka виділяється своєю потужною інтеграцією із засобами машинного навчання та аналізу даних [31]. Інтерфейс користувача, орієнтований на спрощення процесу моделювання та експериментування, робить його доступним для використання як для новачків, так і для досвідчених фахівців. Також слід виділити багатий функціонал Weka який включає в себе широкий спектр алгоритмів машинного навчання, статистичних методів та інструментів для попередньої обробки даних. Це дозволяє користувачам ефективно використовувати різноманітні методи залежно від конкретних завдань та характеристик даних. Weka підтримує розширення за допомогою плагінів, що відкриває можливості для розширення функціональності та інтеграції нових методів та алгоритмів. Неспроможність Weka ефективно обробляти великі обсяги даних та деякі обмеження в інтерфейсі можуть бути визначені як потенційні недоліки. Проте, враховуючи відкритий код та активну спільноту користувачів, можливості Weka для вирішення цих питань шляхом розробки нових версій та розширень залишаються в перспективі. Загалом, Weka представляє собою важливий інструмент для дослідників та аналітиків даних, що шукають ефективні засоби розв’язання задач машинного навчання та аналізу даних.

РОЗДІЛ 3
РОЗРОБЛЕННЯ НАВЧАЛЬНИХ МАТЕРІАЛІВ ДЛЯ ВИВЧЕННЯ МЕТОДІВ РІШЕННЯ ЗАДАЧ КЛАСИФІКАЦІЇ, РЕГРЕСІЇ ТА КЛАСТЕРИЗАЦІЇ.

3.1 Вирішення задач класифікації за допомогою Weka

Класифікація в інтелектуальному аналізі використовується для автоматичного визначення класів або категорій, до яких можуть відноситися об'єкти на основі їх характеристик. Це одна з ключових задач машинного навчання, яка включає в себе навчання моделі на основі історичних даних і використання цієї моделі для прогнозування класів нових об'єктів. В інтелектуальному аналізі, класифікація може бути використана для різних завдань, таких як прогнозування, фільтрація, асоціаційне правило, та інші. Метод класифікації, також відомий як метод класифікаційних дерев або дерево прийняття рішень, представляє собою алгоритм аналізу даних, що визначає послідовний процес прийняття рішень в залежності від значень конкретних параметрів. Графічне зображення цього методу може бути виражене у вигляді дерева, де різні гілки та вузли представляють різні рішення, а кожен крок визначається врахуванням конкретних характеристик [32].

Програма Weka має широкі можливості для вирішення задач класифікації за допомогою різних алгоритмів машинного навчання. Можливості Weka у вирішенні задач класифікації наведені нижче.

Різноманітні алгоритми класифікації. Weka містить великий набір алгоритмів класифікації, таких як C 4.5, Naive Bayes, Random Forest, Support Vector Machines (SVM), k-Nearest Neighbors (k-NN) і багато інших. Ви можете експериментувати з різними алгоритмами, щоб знайти той, який найкраще підходить для вашого конкретного завдання.

Візуалізація результатів. Weka дозволяє візуалізувати результати класифікації за допомогою графіків та діаграм, що допомагає зрозуміти ефективність моделі та можливі проблеми.

Оптимізація параметрів. В Weka є можливість налаштовувати параметри класифікаторів для оптимізації їхньої продуктивності. Ви можете використовувати різні методи оптимізації, такі як перехресна перевірка (cross-validation), для визначення оптимальних значень параметрів [33].

Робота з різними типами даних. Weka підтримує роботу з різними типами даних, включаючи категоріальні та числові дані. Ви можете використовувати різні методи кодування та обробки даних для оптимізації використання алгоритмів класифікації.

Вибір атрибутів. Weka надає інструменти для вибору атрибутів, що дозволяє зменшити розмірність даних та покращити ефективність моделі.

Комплексний інтерфейс для дослідження. Інтерфейс Weka дозволяє вам виконувати експерименти, обирати різні конфігурації та швидко порівнювати результати різних моделей, має інтуїтивний інтерфейс користувача, який дозволяє легко завантажувати дані, вибирати атрибути та виконувати класифікацію без необхідності написання коду.

Підтримка власних моделей. Ви можете імпортувати власні моделі машинного навчання в Weka для їхнього тестування та порівняння з вбудованими алгоритмами.

Weka – це потужний інструмент для класифікації, і він дозволяє досліджувати та експериментувати з різними аспектами задач машинного навчання, ці можливості дозволяють вам ефективно вирішувати завдання класифікації в Weka навіть без глибоких знань у сфері машинного навчання. Нижче наведений приклад послідовних завдань метою якого є ознайомлення з можливостями програми та вирішення задачі класифікації.

Завдання 1: Завантаження файлу для побудови моделі класифікації

Для цього потрібно відкрити програму та натиснути кнопку Explorer і у новому вікні оберіть пункт Open file, потрібний файл має назву credit-g.atff, його можна знайти перейшовши по такому маршруту: Local disk C:→Program files→Weka–3–8–6→Data→ credit–g.atff. На рисунку 3.1 візуалізовано цей процес.

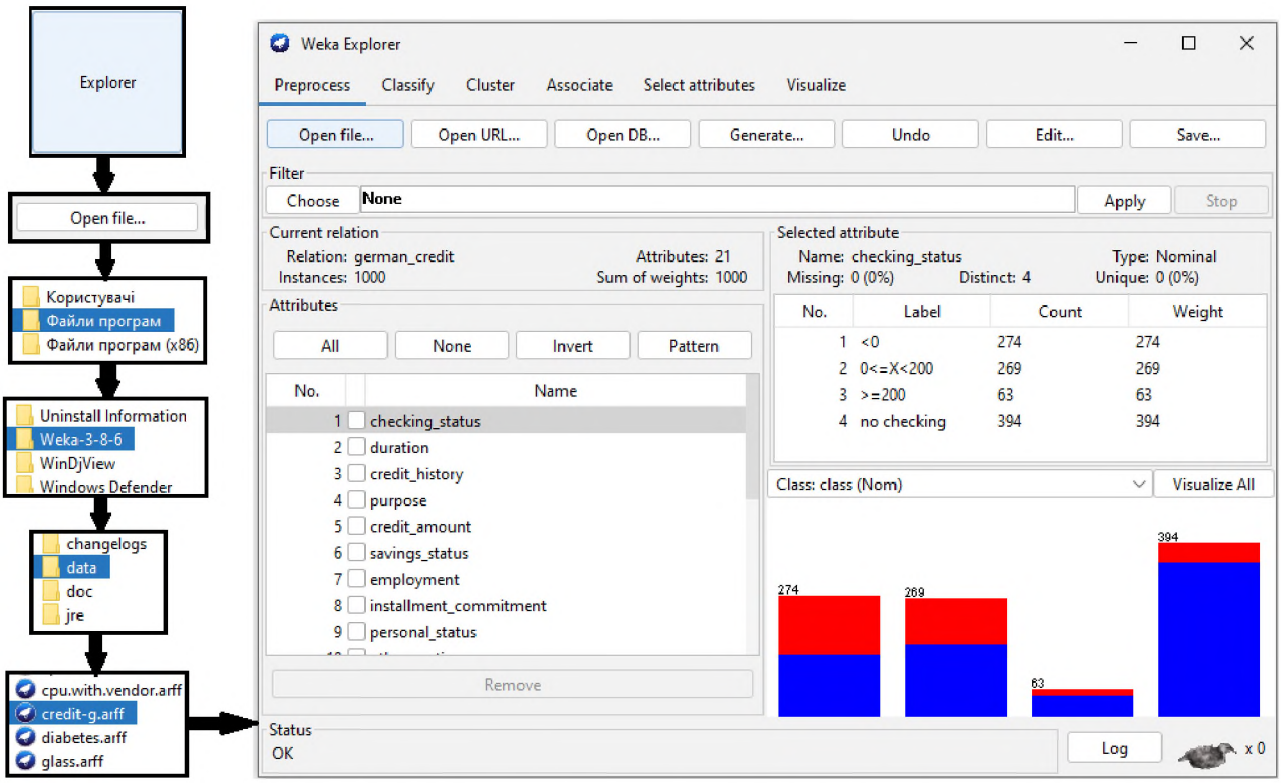

Рисунок 3.1 – Завантаження масиву даних у ПЗ Weka

Завдання 2: Аналіз набору даних у вкладці Preprocess.

Credit-G – це датасет, який часто використовується у сфері машинного навчання для вирішення завдань класифікації. Його зазвичай використовують для прогнозування того, чи буде особа надійним позичальником (тобто, чи вона відштовхнеться від своїх кредитних зобов'язань).

Загалом, датасет містить різні атрибути або характеристики для кожного клієнта, такі як вік, місце роботи, рівень доходу, наявність кредитів у минулому, а також інші фінансові та особисті дані. Кожен запис у датасеті має мітку (мітки класу), яка вказує, чи клієнт відповідає критеріям для видачі кредиту, чи ні (наприклад, «добрий» або «поганий» позичальник). Мета використання цього датасету полягає у тому, щоб навчити класифікатор передбачати, чи буде клієнт надійним для видачі кредиту на основі його особистих і фінансових характеристик.

Розглянемо детальніше інформацію яку нам надає інтерфейс програми. На першій палені з назвою Current relation відображаються параметри відкритого нами файлу, На рисунку 3.2 зображено інтерфейс програми.

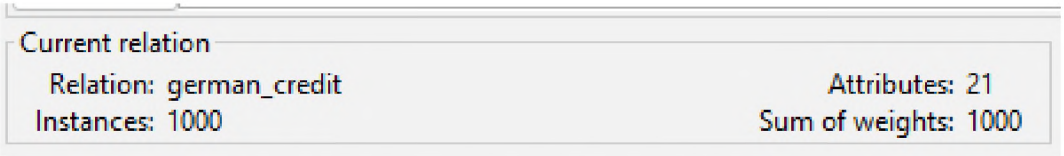

Рисунок 3.2 – Панель Current relation в програмі Weka

На рисунку зображено:

1. Ім’я зв’язку (Relation): ***german_credit*** – це зразок файлу;
2. Примірники (Instances): 1000 рядків даних у наборі даних;
3. Атрибути (Attributes): 21 атрибут у наборі даних.

На другій панелі з назвою Attributes відображається список змінних, серед яких 21 змінна, 20 з них незалежні, а остання що має назву ***class*** являє собою змінною класу, тобто атрибут по якому ми безпосередньо проводимо класифікацію. На рисунку 3.3 зображено інтерфейс програми.

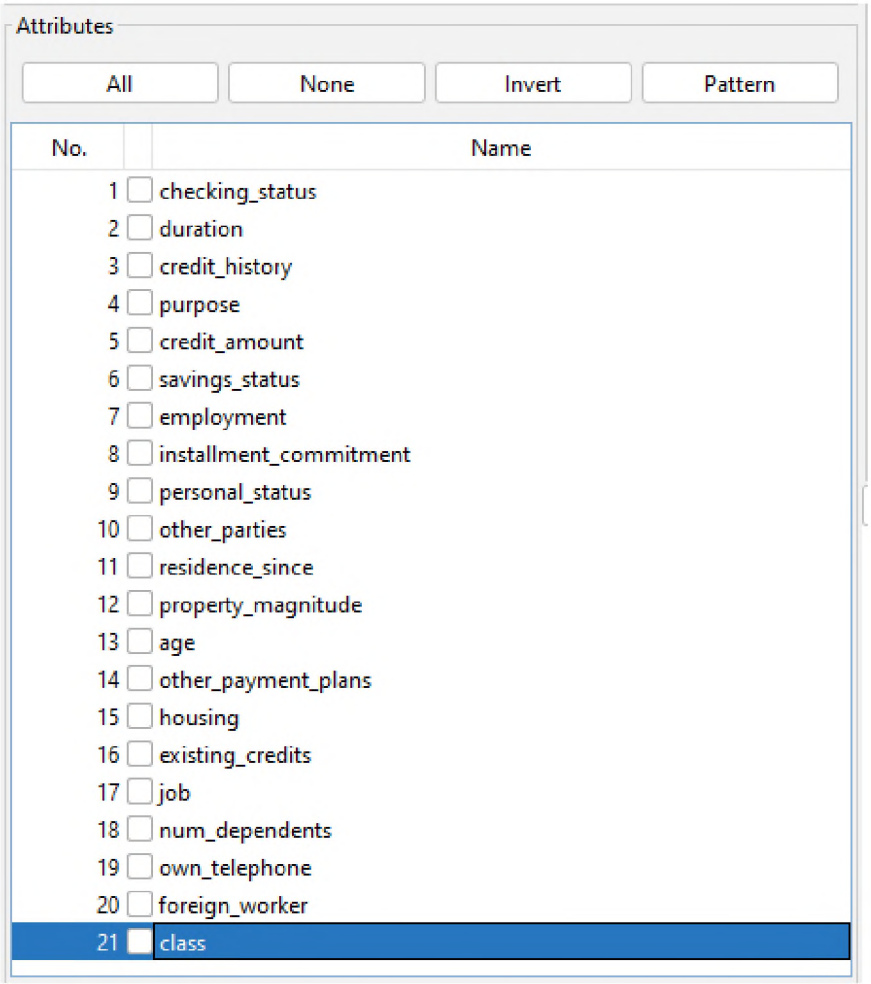

Рисунок 3.3 – Панель Attributes в програмі Weka

Щоб ознайомитися із третьою панеллю що має назву Selected attribute виділіть другу змінну ***duration***, на даній панелі відображається інформація про змінну. На рисунку 3.4 зображено інтерфейс програми.

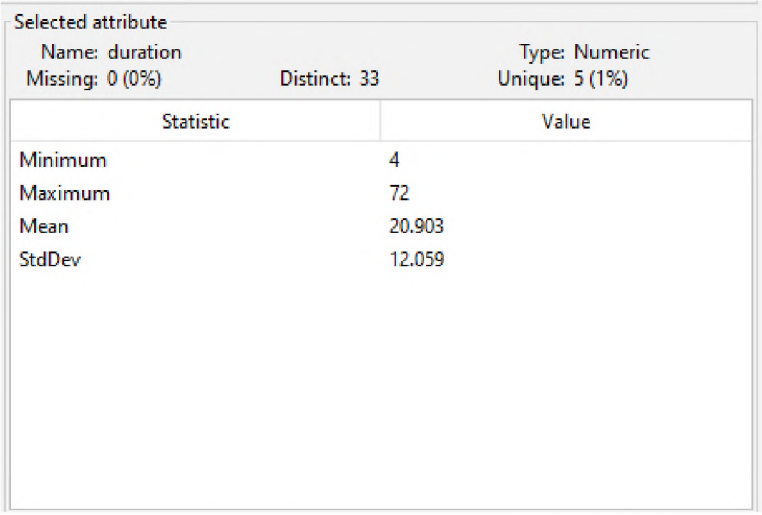

Рисунок 3.4 – Панель Selected attribute в програмі Weka

На рисунку зображено:

1. Ім’я (Name): це ім’я атрибута;
2. Тип (Type): тип атрибута є числовим;
3. Відсутнє значення (Missing): атрибут не має відсутнього значення;
4. Відмінний (Distinct): має 33 різні значення в 1000 екземплярах;
5. Унікальний (Unique): має 5 унікальних значень які не повторюються;
6. Мінімальне значення (Minimum): мінімальне значення атрибута – 4;
7. Максимальне значення (Maximum): максимальне значення атрибута – 72;
8. Середнє (Mean): середнє додає всі значення, поділені на екземпляри;
9. Стандартне відхилення (StdDev): відхилення тривалості атрибута.

Останнім елементом є гістограма яка відображає як при тривалості в 4 одиниці, максимальна кількість випадків трапляється для хорошого класу. Коли тривалість збільшується до 38 одиниць, кількість екземплярів для хороших міток класів зменшується. А коли тривалість досягає 72 одиниць, що має лише один випадок, який класифікує рішення як погане. Зображення інтерфейсу програми зображено на рисунку 3.5.

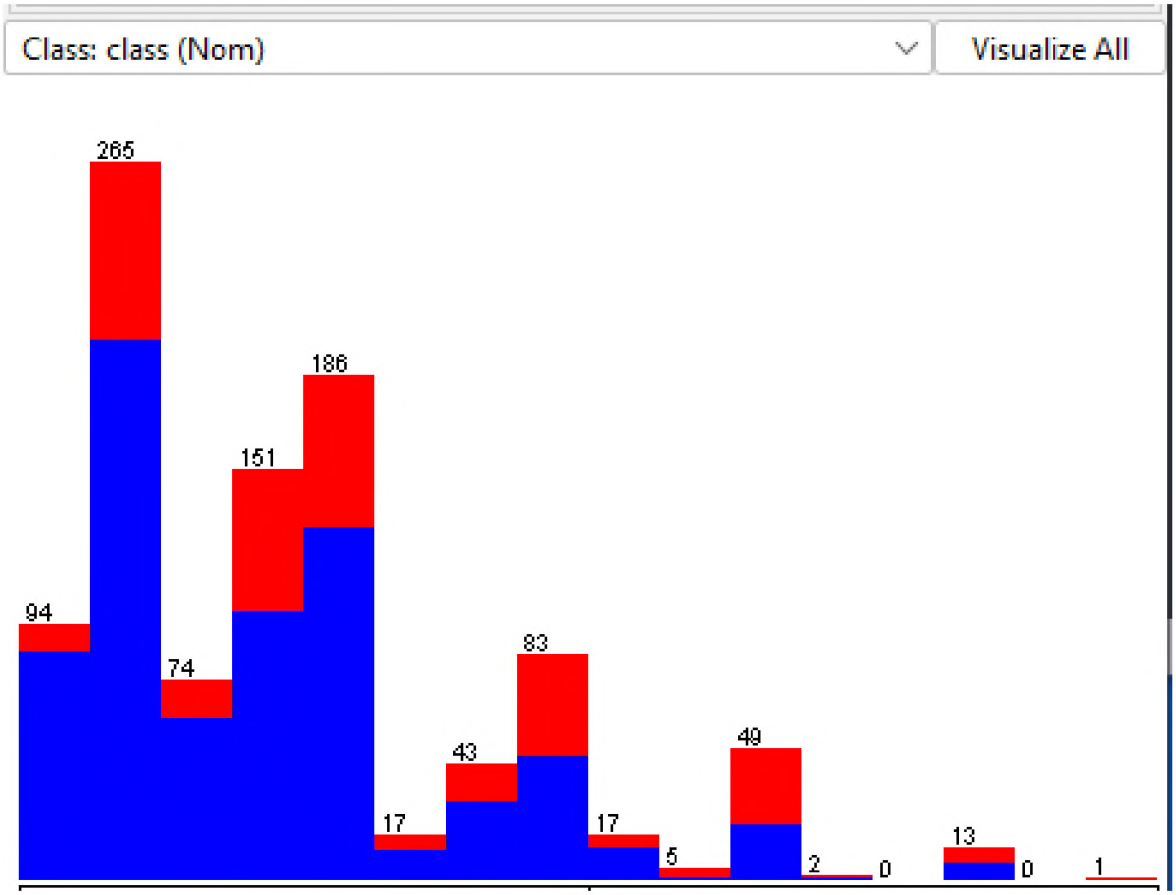

Рисунок 3.5 – Гістограма змінної Duration в Weka

Завдання 3. Огляд та застосування різних алгоритмів для класифікації.

Після попереднього аналізу інформації у Preprocess потрібно перейти на вкладку Classify, в Weka вона є центральним місцем для використання різноманітних алгоритмів машинного навчання для вирішення завдань класифікації. Тут ви можете вибирати, налаштовувати та оцінювати різні класифікатори на вашому наборі даних. Для прикладу використаємо декілька різних алгоритми що використовуються для вирішення подібних завдань, а саме J48, Naive Bayes, SVM, k-Nearest Neighbors. Розглянемо кожен алгоритм трохи детальніше.

Алгоритм J48 (C4.5) в Weka є реалізацією алгоритму побудови дерев рішень для задач класифікації. Основні етапи роботи алгоритму J48 в Weka:

1. Розбиття даних. Починаючи з усіх даних, алгоритм J48 обирає атрибут, який найкращим чином розділяє дані на підгрупи. Це розділення базується на критеріях, таких як індекс Джині, ентропія або інші міри невизначеності.

2. Побудова дерева. Після кроку розбиття, алгоритм J48 рекурсивно застосовує цей процес до кожної підгрупи, поки не буде досягнуто критерію зупинки, такого як глибина дерева чи кількість об’єктів у вузлі.

3. Вибір класу. Коли досягнуто критерію зупинки або виконано іншу умову, вузол дерева призначається конкретному класу (або значенню цільового атрибуту) для класифікації.

4. Порогові значення. Якщо атрибут числовий, J48 може вибирати порогове значення для розділення даних на дві підгрупи.

5. Використання дерева для класифікації. Після побудови дерева можна використовувати його для класифікації нових об’єктів, шляхом проходження вниз по дереву згідно з умовами розділень, поки не буде досягнуто листа дерева, що містить прогнозований клас.

Алгоритм Naive Bayes у вирішенні задач класифікації в Weka використовується для прогнозування класів об’єктів на основі ймовірностей, він ґрунтується на теоремі Байєса і вважає, що всі атрибути незалежні між собою при умові відомого класу об’єкта. Це припущення може бути досить сильним, але не дивлячись на це, Naive Bayes часто працює ефективно на реальних даних і особливо добре для великих наборів даних.

Алгоритм SVM (Support Vector Machine) у вирішенні задач класифікації в Weka використовується для розділення об’єктів на дві або більше класів за допомогою гіперплощини в просторі великої розмірності, він використовує ідею пошуку оптимальної гіперплощини, яка максимізує відстань між класами. Якщо класи неможливо розділити лінійно, SVM може використовувати ядрові функції для переведення даних в більш високорозмірний простір, де вони можуть бути розділені лінійно.

Для того щоб обрати алгоритм, потрібно натиснути кнопку Choose, першим алгоритмом використаємо J48, він знаходиться в категорії Trees. Шлях зображено на рисунку 3.6.

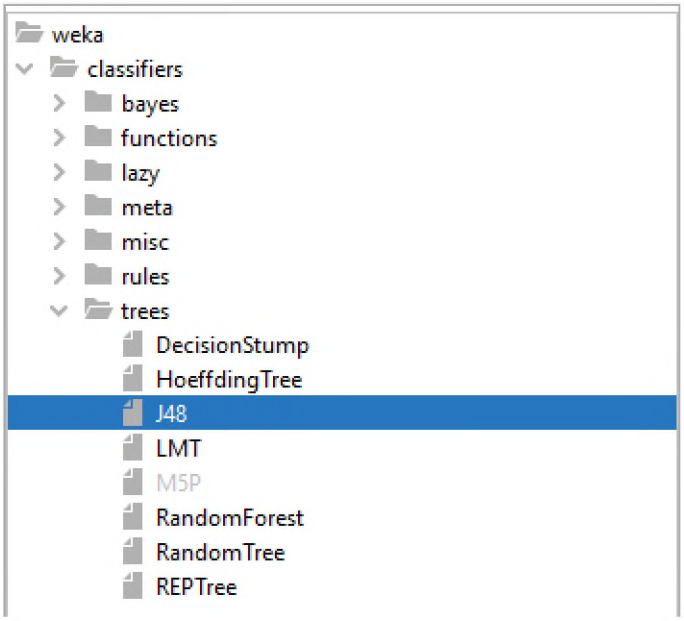

Рисунок 3. 6 – Вікно вибору алгоритму для класифікації в Weka

Після того як ми обрали алгоритм, потрібно звернути увагу на панель Test Options, яка зображена на рисунку 3.6.

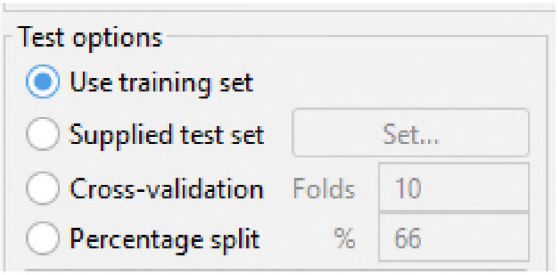

Рисунок 3.7 – Інструменти на панелі Test Options в Weka

Тут присутні такі інструменти як:

1. Use training set використовується для визначення того, чи бажаєте ви оцінювати ефективність моделі машинного навчання на тому ж наборі даних, що і використовувався для навчання. Це означає, що модель буде тестуватися на тому ж наборі даних, на якому вона була навчена.

2. Supplied test set використовується для вказівки власного тестового набору даних для оцінки ефективності вашої моделі класифікації. Замість того, щоб використовувати вбудований або автоматично розділений тестовий набір, ви можете вказати конкретний файл з тестовими даними.

3. Cross-validation дозволяє оцінити ефективність вашої моделі на декількох різних тестових наборах даних, які генеруються шляхом поділу вихідного набору даних на кілька піднаборів. Використовується k-крос-валідація, де весь набір даних розділяється на k піднаборів, а потім модель навчається і тестується k разів.

4. Percentage split використовується для визначення того, який відсоток ваших даних буде використовуватися для тренування моделі, а який – для тестування. Замість того, щоб використовувати крос-валідацію або надавати власний тестовий набір даних, ви можете вказати процент даних, які ви хочете використовувати для тестування.

Власний тестовий набір для використання Supplied test set відсутній, тому використаємо інші три інструменти з алгоритмом J48. Коли ми впевнились у тому що обраний пункт Use training set, потрібно натиснути на кнопку Start, після цього ми отримаємо результуючу модель яка виглядає так як показано на рисунку 3.7.

```
Size of the tree :      140


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances         855               85.5    %
Incorrectly Classified Instances       145               14.5    %
Kappa statistic                          0.6251
Mean absolute error                      0.2312
Root mean squared error                  0.34
Relative absolute error                 55.0377 %
Root relative squared error             74.2015 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,956    0,380    0,854      0,956    0,902      0,640    0,857     0,905     good
                 0,620    0,044    0,857      0,620    0,720      0,640    0,857     0,783     bad
Weighted Avg.    0,855    0,279    0,855      0,855    0,847      0,640    0,857     0,869

=== Confusion Matrix ===

   a   b   <-- classified as
 669  31 |   a = good
 114 186 |   b = bad
```

Рисунок 3.8 – Результуюча модель в Weka

Розберемося, що означають ці числа. Для того, щоб зрозуміти наскільки ефективна модель яку ми отримали, перш за все потрібно звернути увагу на:

1. Correctly Classified Instances (85.5%) вказує на кількість об'єктів чи екземплярів у вашому тестовому наборі, які були правильно класифіковані вашою моделлю машинного навчання.

2. Incorrectly Classified Instances (14.5%) вказує на кількість об'єктів чи екземплярів у вашому тестовому наборі, які були неправильно класифіковані вашою моделлю машинного навчання.

3. MCC (Matthews Correlation Coefficient) – це метрика, яка вимірює якість бінарного класифікатора і призначена для врахування неспівпадінь класів та дисбалансу між ними. MCC важлива з декількох причин:

Врахування дисбалансу класів. MCC враховує як правильно класифіковані позитивні і негативні екземпляри, так і помилково класифіковані позитивні і негативні екземпляри. Це особливо важливо в ситуаціях, коли класи неоднакові за розміром (дисбаланс).

Уникнення перекосів. MCC враховує всі чотири класифікаційні результати: True Positive (TP), True Negative (TN), False Positive (FP) і False Negative (FN). Це дозволяє уникнути перекосів, які можуть виникнути при використанні простих метрик, таких як точність, у випадках дисбалансу класів.

Узагальнення для різних завдань. MCC можна використовувати для різних бінарних класифікаційних задач і придатний для оцінки ефективності класифікатора незалежно від конкретного контексту задачі.

Забезпечення балансу. MCC набуває значення +1 у випадку ідеальної класифікації, -1 у випадку ідеальної зворотної класифікації, і 0 у випадку, коли класифікатор діє як випадковий.

Оскільки у нашому тестовому наборі 85.5% об'єктів були правильно класифіковані то модель можна назвати відносно успішною. Після побудови моделі, програма дає можливість візуалізувати класифікаційне дерево, для цього потрібно натиснути правою кнопкою миші по панелі результуючої моделі і обрати пункт Visualize tree. Візуалізоване дерево зображено на рисунку 3.8.

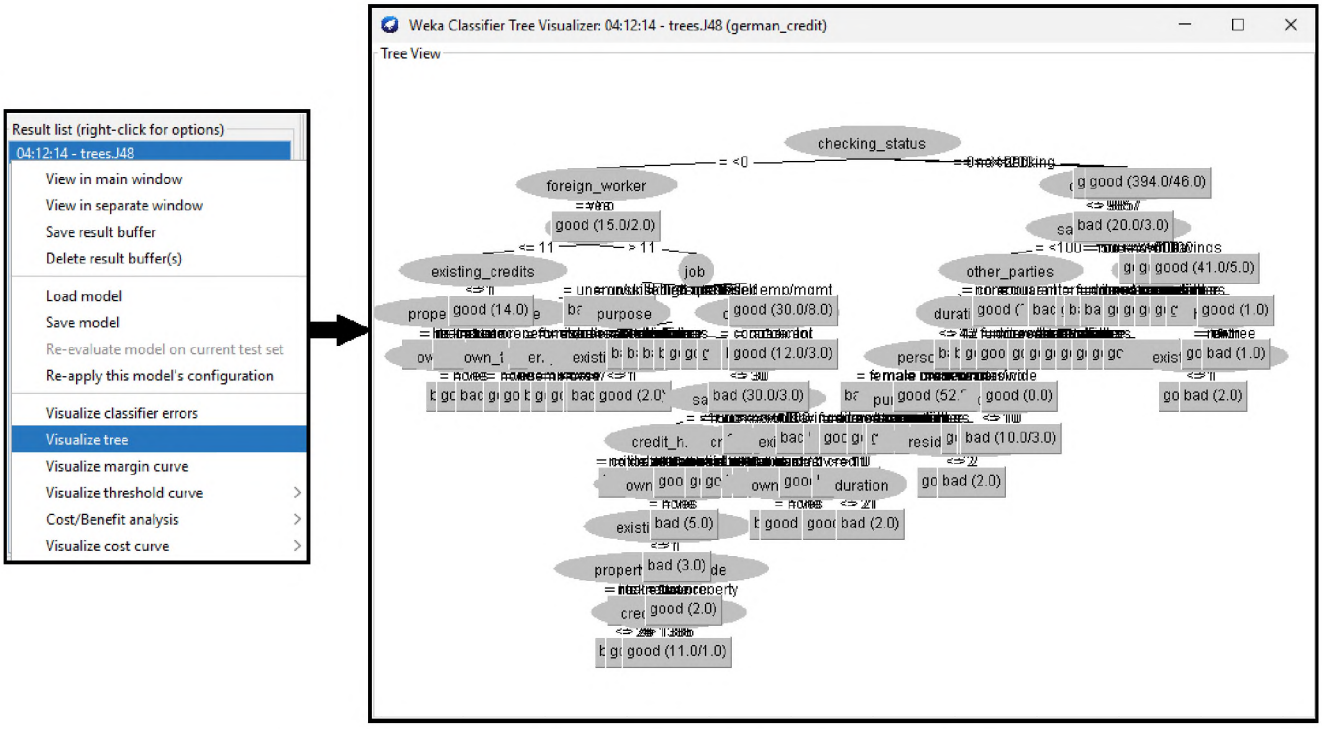

Рисунок 3.9 – Класифікаційне дерево навченої моделі в Weka

Після цього потрібно по черзі застосувати різні інструменти з панелі Test option та різні алгоритми, в результаті має бути 12 навчених моделей (4 різні алгоритми і 3 різних інструменти. Шляхи для обрання алгоритмів зображено на рисунку 3.9.

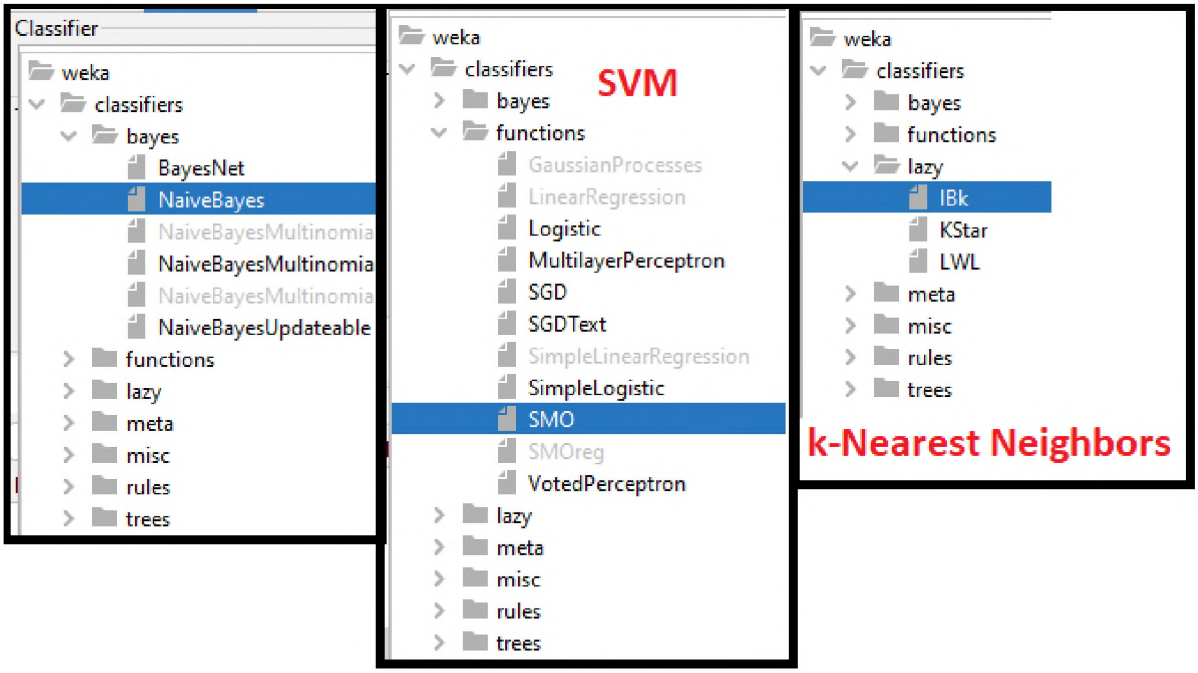

Рисунок 3.10 – Алгоритми для побудови моделі класифікації в Weka

Після виконання цих дій потрібно заповнити таблицю 3.1 і проаналізувати, які з алгоритмів виявилися найбільш ефективними. В чарунки потрібно вписати показники Correctly Classified Instances (CCI), Incorrectly Classified Instances (ICI) і MCC (Matthews Correlation Coefficient).

Таблиця 3.1 – результати застосування алгоритмів для побудови моделі класифікації в програмі Weka

	Use training set			Cross–validation			Percentage split		
	CCI	ICI	MCC	CCI	ICI	MCC	CCI	ICI	MCC
J48									
Naive Bayes									
SVM									
k-Nearest Neighbors									

Завдання 4. Дати відповідь на контрольні питання.

Для закріплення отриманої інформації потрібно дати відповіді на контрольні питання:

1. Що таке навчальний набір даних і як він використовується у вирішенні задачі класифікації в Weka?

2. Які основні етапи роботи з класифікатором в Weka?

3. Що таке атрибути та як вони визначаються в навчальному наборі даних?

4. Як вибрати та завантажити класифікатор в Weka?

5. Які є різновиди класифікаторів, доступних в Weka, і як вони відрізняються один від одного?

6. Як розділити дані на навчальний та тестовий набори у Weka?

7. Як визначити важливість атрибутів у результаті класифікації?

8. Що таке перехресна валідація і як вона допомагає оцінити ефективність класифікатора в Weka?

9. Що таке матриця плутанини та як вона використовується для оцінки ефективності класифікатора?

3.2 Вирішення задач кластеризації за допомогою Weka

Кластеризація – це потужний метод машинного навчання, що включає групування точок даних. За наявності різноманітних точок даних вчені можуть використовувати алгоритм кластеризації для класифікації або групування кожної точки даних в окрему групу. Теоретично точки даних, які належать до однієї групи, мають схожі характеристики або властивості. З іншого боку, точки даних, що належать до різних груп, мають дуже унікальні характеристики або властивості. Кластеризація – це метод навчання без нагляду, популярний серед дослідників у галузі обробки даних, який використовується для отримання статистичного аналізу даних в різних областях. Люди використовують кластерний аналіз в науці про дані, щоб отримати критичне уявлення. Вони аналізують групи, до яких потрапляє кожна точка даних при використанні алгоритмів кластеризації [34].

Алгоритми кластеризації потрібні для того, щоб дослідники даних виявляли вроджені групи серед немаркованих та маркованих наборів даних. Не існує конкретних критеріїв для виділення хорошої кластеризації. Це зводиться до індивідуальних уподобань, вимог і того, що використовує фахівець з даних для задоволення своїх потреб. Скажімо, наприклад, можна було б зацікавитися виявленням однорідних представників груп (зменшення даних) у природних кластерах та визначення їхніх невідомих властивостей. Деякі також хочуть знайти незвичайні об’єкти даних та інші відповідні групи. Так чи інакше, цей алгоритм робить кілька припущень, які стосуються подібності між різними точками. Крім того, кожне припущення створює нові, але однакові.

У сфері кластеризації Weka надає широкий спектр алгоритмів та можливостей для вирішення завдань, такі як k-середніх (K-Means), агломератна кластеризація, EM (Expectation-Maximization), DBSCAN та інші. Ви можете вибирати підходящий алгоритм відповідно до вашого конкретного завдання. Також Weka дозволяє оцінювати якість кластеризації за допомогою різних метрик та інструментів валідації. Це важливо для визначення ефективності вибраного алгоритму. Програма є зручним інструментом для роботи з завданнями

кластеризації, і його різноманітні можливості роблять його популярним серед дослідників та фахівців у галузі машинного навчання.

Завдання 1: Завантаження файлу для побудови моделі класифікації, аналіз набору даних у вкладці Preprocess.

По аналогії з підпунктом 3.2 потрібно завантажити файл що має ім'я . bank_data.arff і ознайомитися з інформацією у вкладці Preprocess.

Завдання 2. Огляд та застосування різних алгоритмів для кластеризації

Після попереднього аналізу інформації у Preprocess потрібно перейти на вкладку Cluster. У Weka, вкладка Cluster використовується для вирішення завдань кластеризації. Ця вкладка надає можливість вибрати різні алгоритми кластеризації, визначити параметри цих алгоритмів та виконати кластерний аналіз даних. Ця вкладка надає зручний інтерфейс для вибору, налаштування та виконання алгоритмів кластеризації в Weka. Важливо експериментувати з різними алгоритмами та параметрами, щоб отримати найкращі результати для вашого конкретного датасету. Для прикладу використаємо алгоритм що використовуються для вирішення подібних завдань, а саме k-Means.

Алгоритм k-Means є одним з найпоширеніших методів кластеризації і використовується для групування об'єктів в k кластерів. Основна ідея полягає в тому, щоб розділити дані на k груп так, щоб сума квадратів відстаней від кожної точки до центру її кластера була мінімальною. Кроки алгоритму k-Means:

1. Обирання кількості кластерів (k). Спочатку визначається, на скільки кластерів ви хочете розділити ваші дані.

2. Ініціалізація центроїдів. Випадковим чином обираються k точки з даних як початкові центроїди кластерів.

3. Присвоєння точок до кластерів. Кожна точка даних призначається тому кластеру, центроїд якого знаходиться найближче до неї.

4. Перерахунок центроїдів. Обчислюються нові центроїди для кожного кластера, взявши середнє значення всіх точок у кластері.

5. Повторяються кроки 3 і 4 до тих пір, поки центроїди не стабілізуються або досягнете фіксованої кількості ітерацій.

6. Коли центроїди стабільні, кластеризація завершена.

Алгоритм k-Means ефективний і простий, але важливо враховувати, що результати можуть залежати від початкового вибору центроїдів, і вони можуть застрягти в локальному мінімумі. Тому декілька запусків з різними початковими умовами можуть покращити результати. Для того щоб обрати алгоритм, потрібно натиснути кнопку Choose. Шлях зображено на рисунку 3.9.

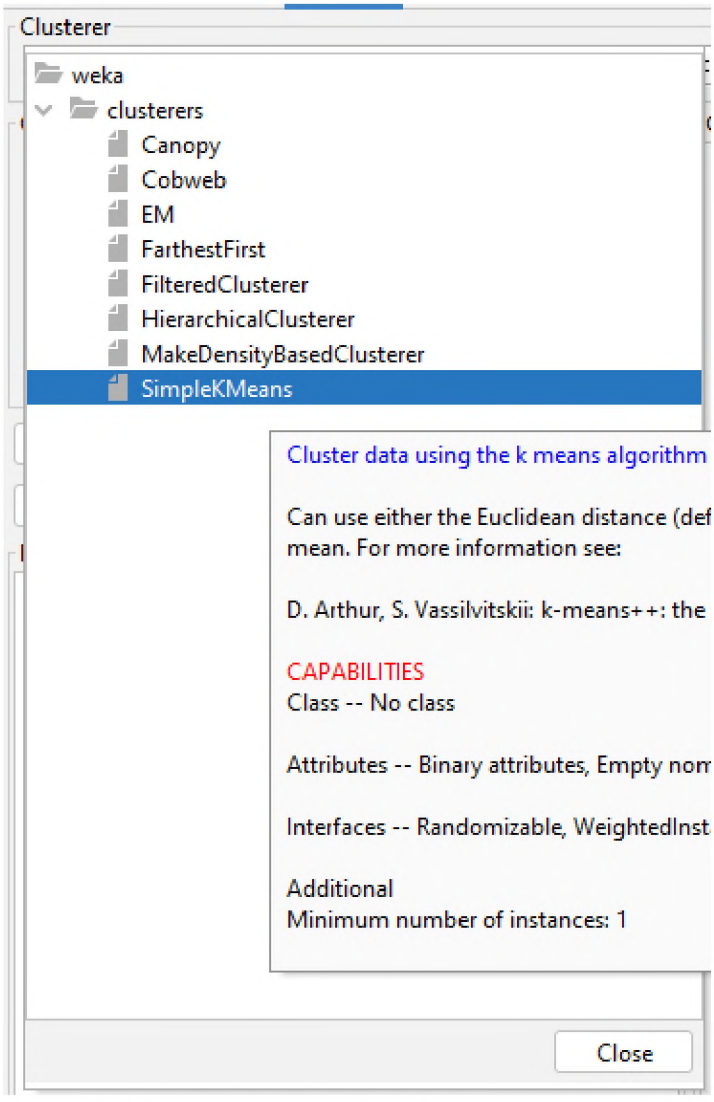

Рисунок 3.11 – Алгоритм k-Means в програмі Weka

Після того як ми обрали алгоритм, потрібно двічі натиснути на назву алгоритму щоб відкрити панель налаштувань. Тут можна змінювати, наприклад, кількість кластерів на які ми бажаємо розділити наші дані, для цього потрібно змінити значення в колонці numClusters, по стандарту стоїть значення 2, тобто дані

будуть розділені на 2 кластера, також важливою є колонка seed, вона задає початковий стан для генератора випадкових чисел (це впливає на початковий вибір центроїдів). Якщо ви встановите це значення, кластеризація буде більш детермінованою. Змінимо значення numClusters на 6 замість 2, значення seed залишимо без змін. Тепер, після перевірки того, що на панелі Cluster mode обраний варіант Use training set, потрібно натиснути на кнопку start. В результаті ми отримаємо результат як на зображенні 3.10.

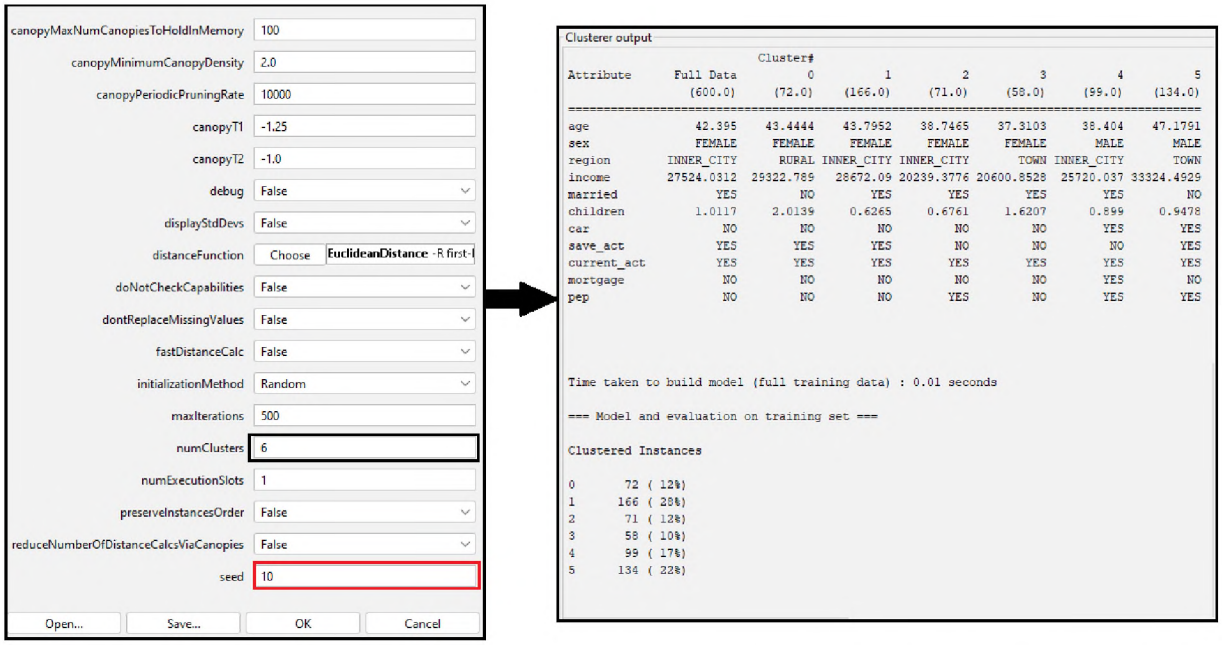

Рисунок 3.12 – Результат кластеризації за допомогою алгоритму k-Means

У даному випадку йдеться про людей що мають різні характеристики (вік, стать, середній дохід та інші) і досліджується чи буде їм цікавий продукт із назвою PEP. У вікні результатів можна побачити центроїди кожного кластера, а також статистичні дані щодо кількості та відсотка екземплярів, які були призначені різним кластерам. Центроїди кластерів є середніми векторами для кожного кластера, що означає, що кожне значення в центроїді представляє середнє значення для цього конкретного атрибута в кластері. За допомогою центроїдів можна оцінювати та характеризувати кожен кластер. Наприклад, центроїд для кластера 1 може вказувати, що це група випадків, представлених жінками середнього віку (приблизно 43 роки), які проживають в центрі міста з середнім доходом близько 28

500 доларів США, одружені та мають одну дитину і т. д. Крім того, ця група в середньому відповідає YES на продукт PEP. Іншим способом розуміння характеристик кожного кластера є використання візуалізації. Можна виконати це, натиснувши правою кнопкою миші на наборі результатів у лівій панелі Result list і обравши опцію Visualize cluster assignments. Ця дія викличе вікно візуалізації, подібне до того, яке показано на малюнку 3.11.

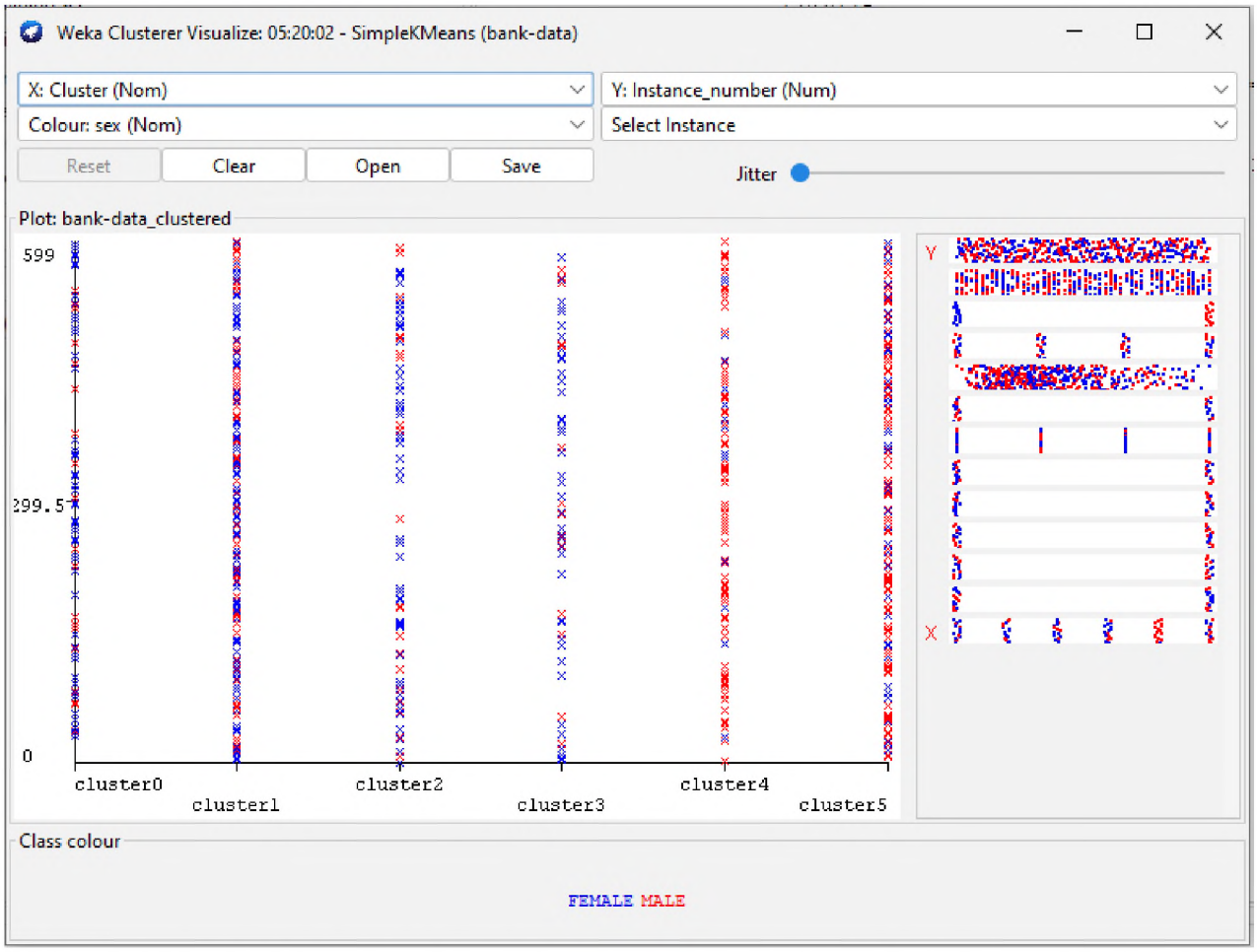

Рисунок 3.13 – Вікно візуалізації кластерного розподілу в програмі Weka

Ви можете обирати номер кластера та інший атрибут для трьох розмірів (вісь x, вісь y та колір). Різні комбінації цих варіантів відображають різні взаємозв'язки у кожному кластері. У даному прикладі ми обрали номер кластера для відображення по осі x, номер екземпляра (призначений WEKA) для осі y, та атрибут «стать» для визначення кольору. Це дає можливість візуально аналізувати розподіл чоловіків та жінок в кожному кластері. Наприклад, відзначається, що в кластерах 2

і 3 переважають чоловіки, в той час як у кластерах 4 і 5 – жінки. Змінивши атрибут кольору на інші характеристики, ми можемо вивчити їхній розподіл у кожному з кластерів.

Завершивши аналіз, важливо зберегти отриманий набір даних, що включає для кожного екземпляра його призначений кластер. Для цього слід натискати кнопку Save у вікні візуалізації і зберегти результат як файл bank-kmeans.arff. Частину цього файлу можна побачити на зображенні 3.12.

```
@relation bank-data_clustered

@attribute Instance_number numeric
@attribute age numeric
@attribute sex {FEMALE,MALE}
@attribute region {INNER_CITY,TOWN,RURAL,SUBURBAN}
@attribute income numeric
@attribute married {NO,YES}
@attribute children numeric
@attribute car {NO,YES}
@attribute save_act {NO,YES}
@attribute current_act {NO,YES}
@attribute mortgage {NO,YES}
@attribute pep {YES,NO}
@attribute Cluster {cluster0,cluster1,cluster2,cluster3,cluster4,cluster5}

@data
0,48,FEMALE,INNER_CITY,17546,NO,1,NO,NO,NO,NO,YES,cluster2
1,40,MALE,TOWN,30085.1,YES,3,YES,NO,YES,YES,NO,cluster4
2,51,FEMALE,INNER_CITY,16575.4,YES,0,YES,YES,YES,NO,NO,cluster1
3,23,FEMALE,TOWN,20375.4,YES,3,NO,NO,YES,NO,NO,cluster3
4,57,FEMALE,RURAL,50576.3,YES,0,NO,YES,NO,NO,NO,cluster1
5,57,FEMALE,TOWN,37869.6,YES,2,NO,YES,YES,NO,YES,cluster3
6,22,MALE,RURAL,8877.07,NO,0,NO,NO,YES,NO,YES,cluster2
7,58,MALE,TOWN,24946.6,YES,0,YES,YES,YES,NO,NO,cluster5
8,37,FEMALE,SUBURBAN,25304.3,YES,2,YES,NO,NO,NO,NO,cluster3
9,54,MALE,TOWN,24212.1,YES,2,YES,YES,YES,NO,NO,cluster5
10,66,FEMALE,TOWN,59803.9,YES,0,NO,YES,YES,NO,NO,cluster1
```

Рисунок 3.14 – Текстовий вигляд моделі кластеризації в текстовому редакторі

Важливо відзначити, що атрибут instance_number, WEKA автоматично додає ще один атрибут під назвою Cluster до вихідного набору даних. Тепер кожен екземпляр у частині даних має свій призначений кластер як останнє значення атрибута.

Завдання 3. Дати відповідь на контрольні питання.

1. Що таке кластеризація в контексті інтелектуального аналізу даних?

2. Які головні завдання можуть бути вирішені за допомогою кластеризації в інтелектуальному аналізі?

3. Які основні типи алгоритмів кластеризації використовуються в інтелектуальному аналізі?

4. Що таке k-Means і як він працює в контексті кластеризації?

5. Які можливі застосування кластеризації в маркетинговому аналізі?

6. Як можна визначити оптимальну кількість кластерів в алгоритмі k-Means?

7. Які переваги та недоліки використання кластеризації в інтелектуальному аналізі?

8. Які інші алгоритми кластеризації використовуються крім k-Means?

9. Як може візуалізація допомагати в розумінні результатів кластеризації?

10. Як можна використовувати кластеризацію для виявлення аномалій в даних?

3.3 Вирішення задач регресії за допомогою Weka

Регресія в інтелектуальній обробці даних – це метод аналізу, який використовується для прогнозування числових значень на основі інших змінних. Вона дозволяє розуміти взаємозв'язок між змінними та передбачувати значення змінної в залежності від інших факторів [35].

Інтелектуальна обробка даних, як правило, використовує різноманітні алгоритми регресії, такі як лінійна регресія, поліноміальна регресія, дерева регресії, метод опорних векторів (SVR), нейронні мережі тощо, для аналізу даних та прогнозування.

Основна мета регресії в інтелектуальній обробці даних полягає в тому, щоб побудувати модель, яка може передбачати значення цільової змінної на основі інших змінних, використовуючи вже наявні дані для навчання.

Процес регресії в інтелектуальній обробці даних включає наступні кроки:

1. Збір даних. Збір та підготовка набору даних, включаючи цільову змінну та фактори, які впливають на цю змінну.

2. Вибір моделі регресії. Вибір алгоритму регресії, який найкраще підходить для вашого набору даних та ваших потреб.

3. Навчання моделі. Використання навчального набору даних для навчання моделі регресії, визначення параметрів моделі.

4. Оцінка моделі. Використання тестового набору даних для оцінки ефективності моделі регресії, вимірюючи метрики точності, такі як середньоквадратична помилка (RMSE), коефіцієнт детермінації (R-squared), тощо.

5. Використання моделі. Застосування навченої моделі для прогнозування значень на нових даних або в реальному часі.

6. Оптимізація та підтримка моделі. Моделі регресії можуть потребувати оптимізації та постійного вдосконалення, особливо при використанні в реальному часі або зміні умов даних.

Регресія – це потужний інструмент в інтелектуальній обробці даних, який допомагає у вирішенні задач прогнозування та розумінні взаємозв'язків між змінними у наборі даних [36].

Завдання 1: Завантаження файлу для побудови моделі класифікації, аналіз набору даних у вкладці Preprocess.

По аналогії з підпунктом 3.2 потрібно завантажити файл що має ім'я cpu.arff і ознайомитися з інформацією у вкладці Preprocess.

Завдання 2. Нормалізація значень та побудова моделі регресії за допомогою алгоритмів.

Після завантаження файлу в програму, якщо розглядати інформацію про змінні, можна помітити, що значення загальних атрибутів, таких як мінімальне, максимальне та середнє значення для кожної змінної, є різними. Зазвичай для створення моделей, таких як регресійна чи класифікаційна, необхідно провести масштабування. Для цього відкрийте файл у вікні та оберіть розділ Filters. Натисніть кнопку Choose і виберіть фільтр Normalize зі списку, а потім натисніть

Apply. Після цього значення всіх змінних будуть нормалізовані, де максимальне буде рівним 1, а мінімальне – 0. Після цих кроків дані будуть готові для використання в алгоритмах машинного навчання в Weka. Інтерфейс програми зображено на рисунку 3.13.

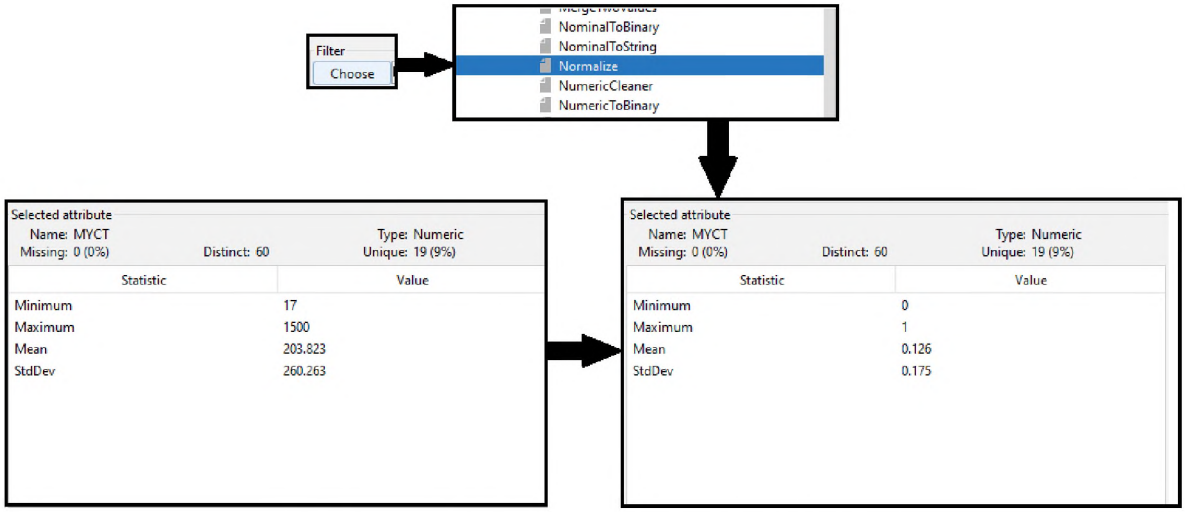

Рисунок 3.15 – Нормалізація в Weka

Щоб обрати алгоритм машинного навчання, перейдіть на вкладку Classify і виберіть потрібний алгоритм зі списку, який випадає. У нашому випадку будуть використані декілька алгоритмів, а саме LinearRegression, MultilayerPerceptron і RandomForest.

LinearRegression є одним із методів лінійної регресії, який широко використовується в інтелектуальній обробці даних для моделювання лінійних залежностей між залежною змінною (цільовою) та незалежними змінними (ознаками) [37].

Основні аспекти та характеристики LinearRegression в інтелектуальній обробці даних включають:

1. Лінійна модель. LinearRegression побудована на припущенні, що залежність між змінною відгуку і признаками є лінійною, тобто може бути виражена лінійною комбінацією признаків.

2. Множинна лінійна регресія. Метод може використовуватися для множинної лінійної регресії, коли є багато незалежних змінних.

3. Мінімізація помилок. Метою LinearRegression є мінімізація суми квадратів різниці між фактичними та передбаченими значеннями (метод найменших квадратів).

4. Коефіцієнти регресії. Алгоритм підбирає оптимальні значення коефіцієнтів регресії (інтерсепта та нахилу) для максимально точного відображення лінійної залежності.

5. Прогнозування. Після навчання модель може використовуватися для прогнозування значень цільової змінної на основі нових наборів признаків.

Важливо враховувати, що лінійна регресія може бути чутливою до викидів та інших аномалій в даних, тому попередній аналіз та підготовка даних є важливою частиною процесу моделювання [38].

Multilayer Perceptron (MLP) в інтелектуальній обробці даних використовується як алгоритм машинного навчання для рішення завдань класифікації та регресії. Ключові аспекти використання MLP у контексті обробки даних [39]:

1. Розпізнавання залежностей. MLP добре підходить для виявлення та моделювання складних залежностей в даних, які можуть бути нелінійними. Він може розпізнавати та вивчати взаємозв'язки між вхідними та вихідними змінними.

2. Різноманітні завдання. MLP може використовуватися як для задач класифікації, коли потрібно визначити категорію вихідної змінної, так і для регресії, коли потрібно прогнозувати числове значення.

3. Архітектура мережі. Вибір архітектури мережі, такої як кількість шарів і нейронів у кожному шарі, може впливати на продуктивність моделі. Правильний вибір цих параметрів важливий для досягнення оптимальної ефективності.

4. Навчання методом зворотного поширення помилки. MLP використовує метод зворотного поширення помилки для коригування ваг усіх зв'язків у мережі. Цей процес допомагає моделі вивчати та адаптуватися до вихідних даних.

5. Регуляризація та уникнення перенавчання. Для покращення загальної моделі можуть використовуватися техніки регуляризації, такі як dropout або L2 регуляризація.

6. Важливість попередньої обробки даних. Попередня обробка даних, така як нормалізація, важлива для досягнення стабільної та ефективної роботи моделі MLP.

Random Forest – це потужний алгоритм машинного навчання, який може використовуватися для класифікації та регресії в інтелектуальній обробці даних [40]. Ключові аспектів його використання:

1. Багатокласова класифікація і регресія. Random Forest може використовуватися для задач класифікації, де модель має прогнозувати категорії, або для задач регресії, де модель прогнозує числові значення.

2. Ансамбль дерев рішень. Random Forest базується на ідеї ансамблю дерев рішень. Він створює кілька різних дерев та комбінує їх прогнози для покращення стійкості та точності.

3. Бутстреп вибірка. Для кожного дерева в лісі випадковим чином вибирається підмножина навчальних даних методом бутстрепу (з поверненням).

4. Випадкові вибори функцій. Під час побудови кожного дерева лише певна частина признаків використовується для прийняття рішень. Це сприяє різноманітності дерев та запобігає перенавчанню.

5. Оцінка важливості признаків. Random Forest надає важливість кожного признака на основі того, як часто признак використовується для прийняття рішень в ансамблі.

6. Стійкість до перенавчання. завдяки випадковості в процесі навчання, Random Forest є стійким до перенавчання, що робить його ефективним для роботи з різноманітними даними.

Цей алгоритм зазвичай використовується для задач класифікації та регресії, особливо в тих випадках, коли важливо отримати стійку та точну модель.

Для наочності використаємо алгоритми LinearRegression і Random Forest, обравши спочатку Use training set а потім Cross-validation. Результат має бути таким як на рисунку 3.14.

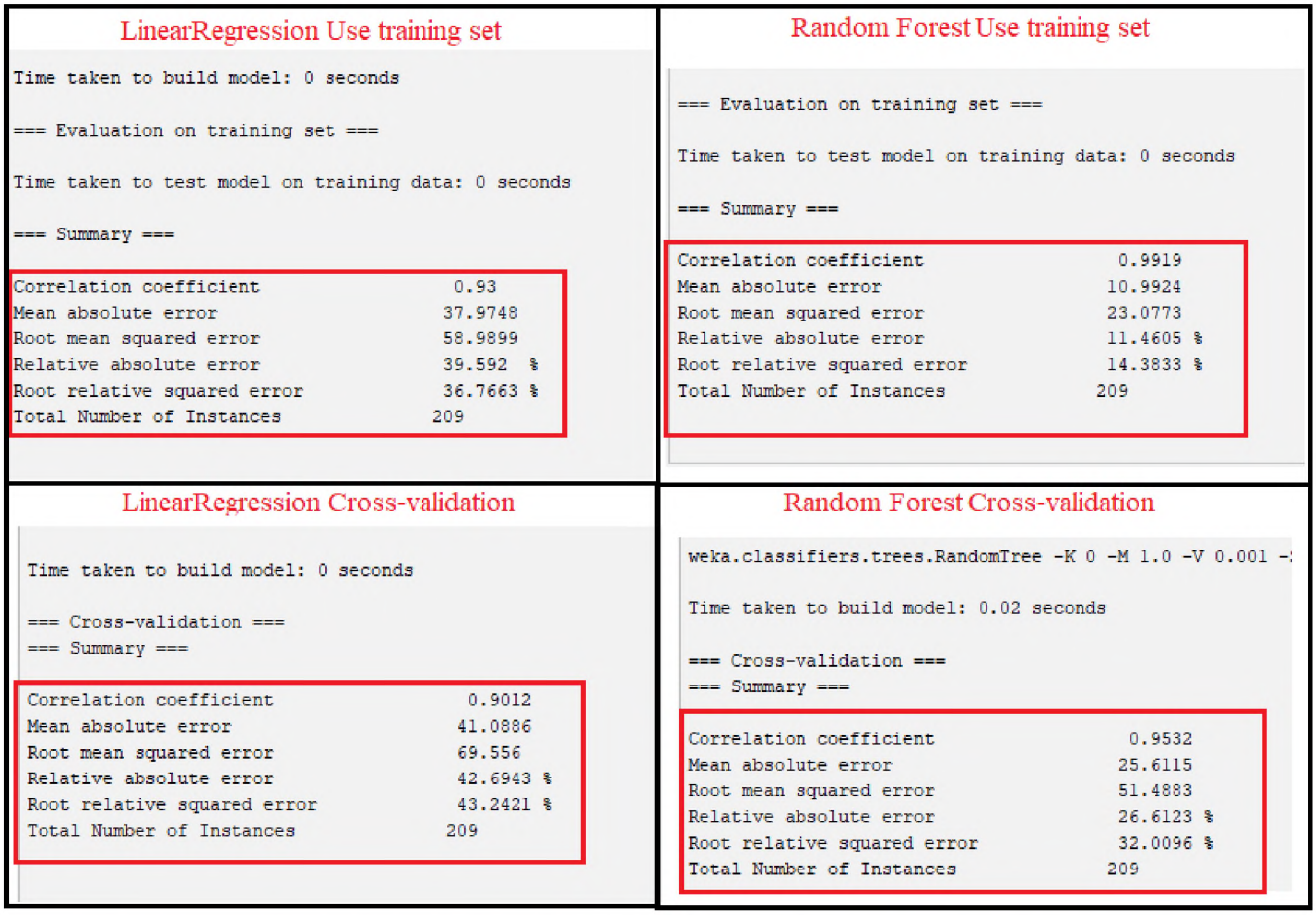

Рисунок 3.16 – Моделі регресії з використанням різних алгоритмів

У контексті регресії в Weka, результати вказують на те, наскільки добре побудована модель прогнозує числову (кількісну) цільову змінну для нових даних. Основні метри, які використовуються для оцінки результатів регресії, включають:

1. Correlation coefficient – у контексті задач регресії, коефіцієнт кореляції може бути використаний для вимірювання того, наскільки сильно прогнозовані значення моделі лінійно залежать від реальних значень. Зазвичай, коефіцієнт кореляції може мати значення від –1 до 1: Якщо кореляція близька до 1, це означає сильну позитивну лінійну залежність. Це вказує, що модель добре передбачає фактичні значення. Якщо кореляція близька до –1, це означає сильну від'ємну лінійну залежність. Також є добрим результатом, але вказує на те, що модель відхиляється у від'ємному напрямку. Кореляція близька до 0 означає відсутність лінійної залежності між прогнозованими і фактичними значеннями [41].

2. Середня абсолютна помилка (Mean Absolute Error, MAE) в задачах регресії в Weka вказує на середню абсолютну величину різниці між фактичними і прогнозованими значеннями цільової змінної. Це одна з метрик, яку можна використовувати для оцінки точності регресійної моделі. У випадку MAE, мета полягає в тому, щоб мінімізувати суму абсолютних значень різниць між прогнозованими і фактичними значеннями. Чим менше значення MAE, тим краще модель регресії пристосована до даних [42].

3. Корінь середньоквадратичної помилки (Root Mean Squared Error, RMSE) це ще одна метрика, яку використовують для оцінки точності моделей регресії в задачах прогнозування в Weka. RMSE обчислюється шляхом взяття квадратного кореня від середньої квадратичної помилки (MSE). Ця метрика покаже розмір середньої абсолютної відхиленості між фактичними і прогнозованими значеннями. У випадку RMSE, також, як і з іншими метриками помилок, менше значення вказує на кращу точність моделі [43].

Ці метри допомагають оцінити точність та ефективність регресійної моделі. У випадку, коли ви використовуєте Weka для регресійного аналізу, можна знайти ці метри в виведених результатів після навчання моделі. Об'єкт Result list в Weka містить ці показники, що дозволяє вам оцінити якість регресійної моделі на основі обраних метрик. Підсумовуючи, краща модель регресії зазвичай має менші значення MSE, MAE, RMSE, вищий R-квадрат та властивості, які підтверджують стабільність та ефективність моделі при роботі з новими даними. Також важливо обирати модель, яка найкраще відповідає особливостям вашого конкретного завдання. У нашому випадку можна сказати що модель побудована за допомогою алгоритму Random Forest з використанням Cross-Validation є найбільш ефективною [44].

Далі слід використати третій алгоритм Multilayer Perceptron а також спосіб Percentage split для того щоб отримати ще більше результатів. Потім варто проаналізувати отримані дані і зробити висновки про те, який алгоритм та спосіб виявився найбільш ефективними, тобто загалом має бути 9 результатів, їх необхідно внести до таблиці 3.2.

Таблиця 3.2 – результати застосування алгоритмів для побудови моделі регресії в програмі Weka

	Use training set			Cross-validation			Percentage split		
	CC	MAE	RMSE	CC	MAE	RMSE	CC	MAE	RMSE
Cross-validation									
Random Forest									
MultilayerPerceptron									

Завдання 3. Дати відповіді на контрольні питання:

1. Що таке задача регресії в контексті машинного навчання?

2. Як використовувати Weka для вирішення задач регресії?

3. Що таке лінійна регресія і які її ключові властивості?

4. Як працює метод Random Forest в контексті задач регресії?

5. Як визначити, яка модель регресії є більш ефективною, використовуючи метрики якості?

6. Що таке нормалізація даних і чому вона важлива при вирішенні задач регресії?

7. Як визначити важливість признаків у моделі регресії?

8. Що таке середня квадратична помилка (MSE), і як вона обчислюється?

3.4 Економічна ефективність

Розрахунок економічної ефективності від розробки навчальних матеріалів, може бути проведений за допомогою кількох ключових кроків. Спочатку потрібно розрахувати вартість розробки навчальних матеріалів, щоб зробити це, варто взяти до уваги те, що на розробку було витрачено 180 годин, при цьому до роботи були залучені 2 особи, це консультант та безпосередньо автор. Для реалізації машинного навчання було використано безкоштовне програмне забезпечення Weka. Згідно зі статистикою міністерства фінансів України, в 2022 році середня вартість однієї

години роботи складає в середньому 3 $. Тепер враховуючи цю інформацію ми можемо використати формулу 3.1 для розрахунку вартості розробки матеріалів.

$$CD = ST * CH \quad (3.1)$$

де CD – вартість розробки;

ST – кількість витраченого часу;

CH – середня вартість години.

Згідно з формулою 3.1:

$$CD = (180 * 2) * 3$$

$$CD = 1080$$

Вартість розробки складає 1080$. Наступним кроком варто порахувати економічну вигоду від зробленої роботи. Оскільки результатом є навчальні матеріали, розрахувати пряму вигоду немає можливості, але можна зробити припущення, що вигода буде полягати у двох складових:

1. Це ефективність робочого часу для викладача, тобто оцінка вартості часу, яку зекономить викладач, якщо йому не потрібно буде створювати навчальні матеріали. Щоб оцінити першу складову використаємо попередні дані про вартість однієї години роботи і припустимо, що викладач витратить на розробку також 180 годин. Тоді можемо порахувати вартість за формулою 3.2.

$$WTE = ST * CH \quad (3.2)$$

де WTE – ефективність робочого часу;

ST – кількість витраченого часу;

CH – вартість години роботи.

Згідно з формулою 3.2:

$$WTE = 180 * 3$$

$$WTE = 540$$

2. Другою складовою можна вважати додаткові вигоди, наприклад можливість залучення нових студентів, які після ознайомлення з результатами роботи проявлять бажання пройти навчальну дисципліну, в яку будуть включені розроблені матеріали. Припустимо що буде один такий учень. Взявши до уваги, що

ціна за рік навчання 10000 грн, тобто 270$, а тривалість навчання складає 4 роки, можемо порахувати другий аспект економічної вигоди за допомогою формули 3.3.

$$AB = CE * DS \tag{3.3}$$

де AB – додаткова вигода;

CE – вартість навчання;

DS – тривалість навчання.

Згідно з формулою 3.3:

$$AB = 270 * 4$$

$$AB = 1080$$

Тепер, коли ми розрахували обидві складові економічної вигоди (EB), можемо визначити її за формуло 3.4.

$$EB = WTE + AB \tag{3.4}$$

Згідно з формулою 3.4:

$$EB = 540 + 1080$$

$$EB = 1620$$

Тепер, коли відомо, що економічна вигода складає 1620$, а витрати на розробку дорівнюють 1080$ можна використати формулу ROI під номером 3.5.

$$ROI = \left(\frac{EB - CD}{CD}\right) * 100\% \tag{3.5}$$

За цією формулою визначимо прибутковість чи збитковість інвестицій.

$$ROI = \left(\frac{1620 - 1080}{1080}\right) * 100\%$$

$$ROI = 50\%$$

В результаті було отримано позитивне значення, а отже проведені інвестиції виявилися прибутковими. Звичайно, варто брати до уваги що показник економічної вигоди є умовним, але навіть припустивши, що розробка навчальних матеріалів виявиться невигідною в економічному плані, варто враховувати численні вигоди в аспекті освіти та професійного розвитку

Висновки до розділу 3

У висновку можна сказати що розроблені навчальні матеріали на тему вирішення задач регресії, класифікації та кластеризації в Weka дозволяють студентам і професіоналам ознайомитися з різноманітними алгоритмами машинного навчання, це допомагає розширити їхні знання та розуміння у використанні таких інструментів для розв'язання реальних завдань. Розроблені матеріали надають студентам можливість отримати практичний досвід роботи з Weka, вони можуть навчитися завантажувати дані, виконувати їх попередню обробку та використовувати різні алгоритми для вирішення конкретних завдань. Вивчення задач регресії, класифікації та кластеризації також включає в себе аналіз результатів, тобто учасники можуть дізнатися, як правильно оцінювати ефективність моделей та розуміти метрики, такі як точність, чутливість, специфічність тощо. Матеріали демонструють, як вирішувати реальні проблеми за допомогою алгоритмів машинного навчання, це включає в себе вибір оптимального алгоритму для конкретного типу завдання та оптимізацію параметрів моделей. Ці навички при правильному опануванні можуть призвести до розвитку професійної кар'єри та розширення розуміння учасників у цій сфері.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи на тему «Методи, технології і інструментальні засоби інтелектуальної обробки даних» була проведена об'ємна робота яка охоплює широкий спектр питань, від загальних методів інтелектуальної обробки даних до конкретних прикладів виконання поставлених завдань.

Перш за все були розглянуті загальні методи і алгоритми які використовуються у цих методах. Під час дослідження була встановлена важливість інтелектуальної обробки даних, а саме було підтверджено важливість у сучасному світі, де обсяги даних стрімко зростають. ІОД стає ключовим інструментом для вилучення цінних інсайтів з великих обсягів інформації, а її загальні методи в свою чергу дозволяють автоматизувати аналіз даних, що робить процес більш ефективним та швидким. Важливе місце займають методи машинного навчання, які є важливим елементом ІОД, вони дозволяють адаптуватися до змін в даних та виявляти нові патерни без необхідності явного програмування, допомагають виявляти складні патерни та взаємозв'язки. Методи прогнозування, що використовує метод роблять можливим передбачення майбутніх тенденцій та подій, це сприяє більш обдуманому прийняттю стратегічних та оперативних рішень. Вивчення підтверджує роль ІОД у виявленні аномалій, шаблонів шахрайства та злочинних дій, це особливо актуально у фінансовій та кібербезпеці. Також були підтверджені думки про широкий спектр застосувань ІОД, включаючи бізнес, медицину, науку, екологію та інші сфери, тобто можна сказати що метод є універсальним інструментом для вирішення різноманітних завдань. На рахунок перспектив розвитку можна сказати що загальні методи обробки постійно розвиваються, особливо за рахунок швидкого прогресу в галузі штучного інтелекту та машинного навчання, тож прогнозується ще більше інновацій та застосувань у майбутньому [45].

Наступним кроком був обраний програмний засіб Weka – це інструмент для машинного навчання та аналізу даних, було встановлено основні переваги та характеристики Weka, що визначають його ефективність, вони включають:

1. Weka надає графічний інтерфейс, що робить його доступним навіть для користувачів без глибоких знань у сфері програмування., це сприяє швидкому навчанню та використанню.

2. Weka має широкий спектр алгоритмів для класифікації, кластеризації, регресії, виділення асоціацій та інші завдання машинного навчання.

3. Weka написаний на Java, і він може використовуватися як незалежний інструмент, так і бути вбудованим у Java-проекти, це дозволяє використовувати його в різних відомих середовищах розробки.

4. У Weka існують різноманітні додаткові інструменти, такі як Experimenter для проведення експериментів з алгоритмами, KnowledgeFlow для візуального програмування тощо.

5. Weka активно використовується в навчальних цілях, інструмент надає велику кількість навчальних ресурсів, прикладів і документації, що дозволяє користувачам швидко освоїти його функціонал.

Останнім кроком була розробка навчальних матеріалів що демонструють можливості Weka і направлені на ознайомлення з методами вирішення завдань з класифікації, кластеризації та регресії. Для демонстрації кожного із завдань спочатку розглядалися різні алгоритми а також був обраний та описаний набір даних який направлений на конкретну задачу. Потім було розглянуто та продемонстровано покроковий алгоритм вирішення задачі в процесі якого був описаний інтерфейс програмного забезпечення і безпосередньо результати, тобто побудовані моделі класифікації, кластеризації та регресії. Завершальним етапом була побудова декількох моделей для вирішення однієї задачі, на основі отриманої інформації були зроблені висновки про ефективність кожного з алгоритмів.

Тож в підсумку можна сказати що була проведена об’ємна робота під час якої було проаналізовано багато аспектів інтелектуальної обробки даних включаючи методи, технології та інструменти. Результатом роботи стали розроблені навчальні матеріали для вирішення різних завдань в цій галузі.