

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ЗАСОБІВ ВИЗНАЧЕННЯ ВІКУ МОРСЬКИХ МОЛЮСКІВ
RESEARCH AND SOFTWARE IMPLEMENTATION OF
TOOLS FOR DETERMINING THE AGE OF MARINE MOLLUSKS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-214М
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Системи штучного інтелекту

БОЙКО Д.С.

(ПРІЗВИЩЕ та ініціали)

Керівник ГОФМАН Є.О.

(ПРІЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ

Кафедра програмних засобів

Ступінь вищої освіти магістр

Спеціальність 122 Комп'ютерні науки

(код і найменування)

Освітня програма (спеціалізація) Системи штучного інтелекту

(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.

Сергій СУББОТІН

“ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

БОЙКА Дмитра Сергійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів визначення віку морських молюсків. Research and Software Implementation of Tools for Determining the Age of Marine Mollusks

керівник проєкту (роботи) к.т.н., доцент, ГОФМАН Євгеній Олександрович,

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “30” вересня 2025 року № 447

2. Строк подання студентом проєкту (роботи) 08 грудня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) _____

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ГОФМАН Є.О., доцент		
Нормоконтроль	КАЛІНІНА М.В., асистент		

7. Дата видачі завдання “30” вересня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	8 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	9-10 тижні	Додатки
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

_____ Дмитро БОЙКО
(підпис) (Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Євгеній ГОФМАН
(підпис) (Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
84 с., 3 табл., 29 рис., 3 дод., 19 джерел.

PYTHON, JUPYTER NOTEBOOK, КЛАСИФІКАЦІЯ, МОВА ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, НАВЧАЛЬНА ВИБІРКА, ЦІЛЬОВА ФУНКЦІЯ.

Об'єкт дослідження – процеси обчислень, пов'язані з класифікацією на основі даних навчальних вибірок.

Предмет дослідження – методи та засоби синтезу класифікаційних моделей за вибірками даних.

Мета роботи – дослідження та реалізація засобів визначення віку морських молюсків.

Матеріали, методи та технічні засоби: мова програмування Python, середовище розробки Jupyter Notebook.

Результати. Розроблено програмне забезпечення для обчислень, що пов'язані з класифікацією даних для визначення віку морських молюсків, використовуючи мову Python та середовище розробки Jupyter Notebook.

Практична цінність роботи полягає у розробці програмного забезпечення для обчислень, пов'язаних з класифікацією даних для визначення віку морських молюсків.

Висновки. Виконано проектування програмного забезпечення для обчислень, пов'язаних з класифікацією даних для визначення віку морських молюсків. Розроблено програмне забезпечення для обчислень, що пов'язані з класифікацією даних для визначення віку морських молюсків. Здійснено тестування розробленого програмного забезпечення для класифікації даних для визначення віку морських молюсків.

Галузь використання – біологічні дослідження.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 84 pages, 3 tables, 29 figures, 3 appendixes, 19 sources.

PYTHON, JUPYTER NOTEBOOK, CLASSIFICATION, PROGRAMMING LANGUAGE, SOFTWARE, TRAINING SAMPLE, OBJECTIVE FUNCTION.

The object of research is the process of computations related to the classification based on training sample data.

The subject of the research is methods and tools for synthesizing classification models based on data samples.

The purpose of this work is research and implementation of tools for determining the age of marine mollusks.

Materials, methods and technical tools: Python programming language, Jupyter Notebook development environment.

Results. The software tools for calculations related to the classification of data for determining the age of marine mollusks using the Python programming language and the Jupyter Notebook development environment have been developed.

The practical value of the work lies in the development of software for calculations related to the classification of data for determining the age of marine mollusks.

Conclusions. The software for calculations related to the classification of data for determining the age of marine mollusks has been designed. The software for calculations related to the classification of data for determining the age of marine mollusks has been developed. The developed software for data classification for determining the age of marine mollusks has been tested.

Scope of use – biological research.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз предметної області.....	11
1.1 Класифікація на основі даних навчальних вибірок.....	11
1.2 Аналіз засобів класифікації на основі даних навчальних вибірок	18
1.3 Висновки за розділом 1	27
2 Матеріали і методи	29
2.1 Вибір мови програмування	29
2.2 Вибір середовища розробки для створення програмного забезпечення для визначення віку морських молюсків.....	31
2.3 Метод класифікації на основі кількох алгоритмів машинного навчання, що працюють у взаємодоповнюючій структурі	34
2.4 Висновки за розділом 2	38
3 Опис програми	40
3.1 Структура програмного забезпечення для визначення віку морських молюсків	40
3.2 Функціонування програмного забезпечення для визначення віку морських молюсків.....	42
3.3 Розробка моделей програмного забезпечення для визначення віку морських молюсків.....	44
3.4 Особливості реалізації програмного забезпечення для визначення віку морських молюсків.....	48
3.5 Висновки за розділом 3	50
4 Експлуатація, тестування та експериментальне дослідження програми.....	51
4.1 Призначення й умови застосування програми	51
4.2 Характеристики програми для визначення віку морських молюсків	52
4.3 Інструкція по експлуатації програми.....	53
4.3.1 Звернення до програми.....	53

4.3.2 Вхідні й вихідні дані.....	53
4.3.3 Повідомлення	53
4.4 Виконання програмного забезпечення для визначення віку морських молюсків	54
4.5 Тестування програмного забезпечення для визначення віку морських молюсків	59
4.6 Висновки за розділом 4	59
Висновки	60
Перелік джерел посилань	63
Додаток А Технічне завдання	65
Додаток Б Фрагмент тексту програми	69
Додаток В Слайди презентації	77

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

КД – класифікація даних;

НВ – навчальна вибірка;

ПЗ – програмне забезпечення;

ЦФ – цільова функція.

ВСТУП

Визначення віку морських молюсків є важливим завданням у морській біології, екології та рибному господарстві, оскільки воно дозволяє оцінювати стан популяцій, відстежувати вплив довкілля на темпи росту та забезпечувати раціональне використання біоресурсів. Традиційні методи дослідження, засновані на візуальному аналізі раковин чи внутрішніх структур, відзначаються високою трудомісткістю та обмеженою точністю, що створює потребу в нових технологічних підходах [1], [2].

Інтелектуальні методи та сучасні програмні засоби відкривають можливість автоматизованого аналізу біологічних даних, значно скорочуючи час досліджень та підвищуючи їхню достовірність. Використання алгоритмів машинного навчання, комп'ютерного зору та інших інструментів штучного інтелекту дає змогу працювати з великими масивами даних і виявляти закономірності, що є недоступними при традиційному аналізі [3]-[5].

Актуальність такого підходу полягає в його значному науковому та практичному значенні: від покращення екологічного моніторингу морських екосистем до підвищення ефективності аквакультури та збереження біорізноманіття [3]-[6].

Проте інтелектуальні методи та програмні засоби для визначення віку морських молюсків, попри свої переваги, мають низку недоліків. По-перше, ефективність таких систем значною мірою залежить від обсягу та якості навчальних вибірок. Для багатьох видів молюсків бракує достатньої кількості достовірних даних, що ускладнює побудову моделей із високою точністю. По-друге, алгоритми машинного навчання часто демонструють обмежену універсальність: модель, навчена на одному виді чи популяції, може працювати ненадійно при застосуванні до інших умов або видів. Ще однією проблемою є складність інтерпретації результатів. Багато сучасних методів, особливо глибокі нейронні мережі, працюють як «чорні скриньки», що ускладнює розуміння причин тих чи інших рішень і знижує рівень довіри з

боку дослідників. Крім того, програмні засоби потребують значних обчислювальних ресурсів і спеціалізованого обладнання, що обмежує їх використання у польових умовах. Також важливим недоліком є недостатня стандартизація методик: відсутність єдиних протоколів обробки та аналізу даних призводить до різноспрямованих результатів у різних дослідженнях. Це знижує можливість широкого практичного застосування інтелектуальних систем та ускладнює порівняння результатів між науковими групами [1]-[7].

Тому актуальним є дослідження методів класифікації на основі даних навчальної вибірки для визначення віку морських молюсків. У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання дослідження та реалізація засобів визначення віку морських молюсків.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз методів та засобів класифікації на основі даних навчальних вибірок;
- розробити метод класифікації на основі кількох алгоритмів машинного навчання;
- здійснити проектування програмного забезпечення для визначення віку морських молюсків;
- створити програмне забезпечення для визначення віку морських молюсків;
- виконати тестування розробленого програмного забезпечення для визначення віку морських молюсків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Класифікація на основі даних навчальних вибірок

Класифікація даних являє собою процес віднесення об'єктів, записів або прикладів до певних категорій (класів) на основі їхніх характеристик. Іншими словами, це розподіл даних за задалегідь визначеними групами згідно з певними ознаками. Класифікація даних є методом інтелектуального аналізу, що дозволяє прогнозувати належність нового об'єкта до певного класу. Вона використовується тоді, коли структура класів відома задалегідь, і завдання полягає у навчанні моделі для розпізнавання нових прикладів [1], [2].

Класифікація даних у сфері інформаційних технологій розглядається як системний підхід, що дозволяє впорядковувати інформаційні ресурси та підвищувати рівень їх захищеності. У процесі класифікації визначається ступінь чутливості відомостей та оцінюється можливий вплив несанкціонованого доступу, змін чи втрати таких даних на організацію або окремих осіб [1], [2].

У практичному застосуванні частина інформації належить до категорії конфіденційної чи персональної, що потребує ретельного захисту, тоді як інші відомості не становлять значної загрози і можуть оброблятися без жорстких обмежень. Класифікація включає не лише присвоєння міток, але й розуміння того, як потенційне розкриття різних наборів інформації може вплинути на безпеку чи репутацію [1], [2].

Завдяки цьому підходу з'являється можливість коректно розподіляти ресурси захисту, забезпечувати відповідність законодавчим нормам і водночас оптимізувати процеси управління даними. Кінцевою метою виступає визначення правил зберігання, використання та охорони даних у спосіб, що поєднує доступність і практичність із надійністю захисту [1], [2].

Таким чином, класифікація сприяє відповідальному використанню інформації, формує баланс між відкритістю та безпекою й дозволяє організаціям діяти в межах нормативних вимог. У результаті знижується ризик

витоків, підвищується ефективність управління інформаційними потоками та створюється культура обережного ставлення до даних [1], [2].

Класифікація даних у сучасному бізнес-середовищі розглядається як механізм, що забезпечує впорядкованість і захищеність інформації. Вона передбачає розподіл відомостей на категорії з урахуванням їхнього змісту, рівня важливості та можливих наслідків у разі несанкціонованого доступу чи витоку. В основі такого підходу лежить принцип поєднання зручності використання даних із вимогами безпеки, що дає змогу знаходити баланс між відкритістю та обмеженням доступу [1], [2].

Один із традиційних способів полягає в ручному визначенні рівня чутливості кожного документа або запису. Подібний підхід дозволяє враховувати контекст, адже рішення приймає людина, однак він потребує значних витрат часу й не виключає ймовірності помилки. На противагу цьому, автоматизовані системи здатні швидко опрацьовувати великі обсяги даних, застосовуючи заздалегідь визначені алгоритми чи пошук ключових ознак. Проте їхня обмеженість полягає у відсутності гнучкого розуміння контексту. Як компромісна стратегія використовується гібридний підхід, у якому технічні засоби виконують основну роботу, а людина здійснює перевірку складних випадків і коригування результатів [1], [2].

Вибір методики класифікації залежить від характеру інформації, масштабу діяльності та ресурсних можливостей. Кожен варіант має власні сильні та слабкі сторони, проте спільним для всіх залишається завдання — забезпечення безпечного зберігання й коректного використання даних. Саме завдяки класифікації можна вибудувати прозорі правила доступу, сформувані чіткі політики та підвищити рівень довіри до організації [1], [2].

Питання класифікації неможливо розглядати без урахування галузевих стандартів і правових вимог. Нормативні документи встановлюють єдині підходи до управління інформаційними потоками й задають рамки для побудови внутрішніх процесів. Дотримання міжнародних практик підвищує

репутацію компанії та гарантує відповідність мінливим вимогам законодавства [1], [2].

Окреме значення мають внутрішні політики, які враховують специфіку компанії та деталізують правила поведінки з даними. Вони визначають коло осіб, які можуть мати доступ до певних категорій інформації, регламентують порядок зберігання й використання відомостей та встановлюють обов'язки співробітників. У результаті формується культура відповідального ставлення до даних, що не лише зміцнює безпеку, але й підтримує ефективність бізнес-процесів [2], [3].

Попри важливість такого підходу, процес класифікації часто стикається з труднощами. Серед них — робота з неструктурованими відомостями, пошук балансу між зручністю доступу та необхідним рівнем захисту, а також потреба постійного дотримання численних регуляторних вимог. Значну роль відіграє людський чинник, оскільки помилки або недбалість персоналу здатні нівелювати навіть найдосконаліші системи. Використання технологій також не завжди безпроблемне, адже вони можуть вимагати значних витрат чи бути несумісними з наявними рішеннями [2], [3].

Незважаючи на ці виклики, саме коректна класифікація створює основу для надійного управління даними. Вона стає запорукою безперервності бізнесу, підтримує довіру клієнтів і забезпечує контроль над ризиками в умовах постійно зростаючого обсягу інформації [2], [3].

У будь-якій організації постійно циркулює велика кількість інформації. Частина цих даних може бути відкритою й доступною для всіх, наприклад, відомості про діяльність чи загальні послуги. Водночас існує інформація, яка має конфіденційний характер і потребує суворого захисту. Саме для цього використовується класифікація даних, яка дозволяє розмежовувати інформацію за рівнем чутливості та визначати відповідні правила роботи з нею. Це можна уявити як умовне розділення документів на ті, що можна залишати у відкритому доступі, і ті, які необхідно зберігати у закритому середовищі. Формування політики класифікації інформації є ключовим етапом

управління безпекою даних. Вона охоплює процеси визначення меж застосування, описує відповідальних учасників, а також передбачає створення критеріїв для розподілу даних за категоріями. Політика встановлює правила поводження з кожним типом інформації, включаючи зберігання, обробку, передачу та знищення. Вона також визначає механізми контролю доступу, застосування засобів автентифікації та шифрування, а також передбачає системи маркування, що допомагають легко ідентифікувати рівень чутливості даних [2], [3].

Класифікація даних забезпечує системний підхід до управління інформацією, допомагає мінімізувати ризики, підвищує рівень захищеності й формує культуру безпеки в організації. Незважаючи на складність у впровадженні, вона створює додаткову цінність у вигляді впорядкованості, надійності та підвищеної відповідальності. Такий підхід дозволяє розглядати класифікацію не як додатковий тягар, а як необхідний інструмент для захисту бізнесу та збереження його ключових активів у середовищі, де інформація стає основним ресурсом [2], [3].

Класифікація даних являє собою підхід, який дозволяє структурувати дані, полегшувати їх подальший аналіз та забезпечувати ефективне використання. Застосування класифікації полягає у виявленні закономірностей, прогнозуванні результатів, підтримці прийняття рішень та підвищенні безпеки інформаційних систем. Використання цього методу поширюється у сферах фінансів, медицини, маркетингу, кібербезпеки та багатьох інших галузях, де є потреба у впорядкуванні великих масивів інформації. За характером методів класифікація може бути керованою, коли для навчання використовуються заздалегідь позначені дані, або некерованою, коли система самостійно знаходить приховані закономірності. Крім того, існують бінарна класифікація, яка спрямована на розподіл об'єктів на дві категорії, багатокласова, що передбачає більшу кількість груп, та ієрархічна, де формуються рівні підпорядкованих категорій [3], [4].

Класифікація даних розглядається як метод впорядкування інформації з метою її ефективного використання, захисту та зручності доступу. Під час цього процесу дані отримують певні категорії, що дозволяє швидко визначати їхню важливість, рівень конфіденційності та необхідні заходи захисту. Упровадження чіткої системи класифікації робить роботу з інформаційними масивами більш керованою, знижує ризики витоку чи втрати даних і допомагає дотримуватися нормативних вимог, які встановлюються для конкретної галузі [3], [4].

Упорядкування даних дає змогу визначати правила їх обробки, терміни зберігання та рівні доступу для співробітників. Безпосередньо після створення схеми класифікації формується набір стандартів, що регламентують, як слід працювати з даними різних категорій, які механізми безпеки застосовувати та які умови використання враховувати протягом життєвого циклу інформації. Це дозволяє узгодити внутрішні практики обробки з вимогами галузевих норм, зокрема тих, що стосуються збереження конфіденційних даних клієнтів, фінансових чи медичних відомостей [3], [4].

Застосування класифікації забезпечує кілька важливих результатів. Завдяки такій системі досягається вищий рівень конфіденційності, оскільки критично важливі дані отримують додатковий захист. Крім того, зберігається цілісність інформації, адже впроваджуються механізми контролю доступу та перевірки достовірності. Разом із цим підтримується доступність даних, оскільки впорядкування спрощує роботу з великими обсягами інформації та дає можливість швидко знаходити потрібні елементи [4], [5].

Процес класифікації передбачає створення чіткої політики управління даними. Спершу відбувається збір інформації для визначення її цінності, кількості та доступу. Далі формується структура категорій із використанням тегів чи метаданих, що дозволяє ідентифікувати дані автоматизованими методами. Наступним кроком стає узгодження цієї структури зі стандартами безпеки й вимогами законодавства. Завершується процес обробкою даних

відповідно до визначених категорій, що забезпечує їх захищеність та зручність використання [4], [5].

Особливу цінність класифікація має у випадках роботи з неструктурованими даними, які зазвичай складно аналізувати без додаткового впорядкування. Категоризація робить їх придатними для пошуку, аналізу та подальшої обробки. При цьому усунення дублікатів допомагає скоротити витрати на зберігання та підвищити рівень безпеки [4], [5].

Для реалізації класифікації застосовуються різноманітні інструменти – від баз даних і систем управління до програмного забезпечення бізнес-аналітики. Використання таких засобів дозволяє автоматизувати сортування, спростити ідентифікацію даних та пришвидшити роботу з великими обсягами інформації [4], [5].

Систематичне впорядкування даних має значення не лише для внутрішнього управління, але й для зовнішньої відповідності. Класифікація допомагає компаніям дотримуватися міжнародних стандартів та місцевих регуляторних вимог, захищати чутливу інформацію від несанкціонованого доступу та підтримувати прозорість під час перевірок. Періодична перекласифікація дозволяє зберігати актуальність системи та відповідати новим викликам, які пов'язані з розвитком кіберзагроз, зміною законодавства чи зростанням обсягів даних [4], [5].

Класифікація даних може розглядатися як процедура систематизації інформації шляхом надання їй певних категорій або міток. Таке впорядкування дозволяє легше контролювати використання даних, забезпечувати їх захист, а також підвищувати ефективність зберігання. Процес починається з ідентифікації наявних джерел інформації, які можуть містити як структуровані, так і неструктуровані масиви, після чого визначаються їхні властивості та рівень важливості для організації [4], [5].

Завдяки цьому вдається встановлювати правила щодо доступу, уникати надмірного дублювання, підвищувати зручність пошуку та мінімізувати ризики, пов'язані з крадіжкою або втратою даних. Окрім підвищення рівня

безпеки, класифікація сприяє оптимізації роботи з інформаційними ресурсами, дозволяє ефективніше організувати внутрішні процеси та дотримуватися зовнішніх регуляторних вимог [4], [5].

Особливу роль відіграє розподіл інформації за рівнем чутливості. Найменш критичні дані, що можуть бути відкрито оприлюднені, не вимагають посиленого контролю, тоді як конфіденційні або обмежені записи потребують жорстких механізмів захисту. Таким чином формується основа для визначення пріоритетів у збереженні та доступі до різних категорій даних [5], [6].

Таким чином, класифікація даних не лише сприяє зниженню ризиків, але й створює підґрунтя для якіснішого управління інформаційними ресурсами, забезпечує їхню структурованість і дозволяє організаціям ефективно контролювати як повсякденні операції, так і стратегічні завдання. Класифікація даних дозволяє впорядкувати інформацію та забезпечити для неї оптимальний рівень захисту. Її виконання починається з аналізу ризиків, пов'язаних із наявними масивами даних. Проводиться оцінка вразливих місць, визначається характер інформації, яка зберігається у різних середовищах, та з'ясовується, які загрози можуть виникнути у разі несанкціонованого доступу, модифікації чи втрати. У цей момент важливо усвідомити, хто створює та контролює дані, хто відповідає за їх точність і збереження, а також які обмеження накладають чинні регуляторні вимоги [5], [6].

Подальший етап передбачає визначення цілей класифікації. У різних організаціях акцент може робитися на забезпеченні відповідності стандартам, підвищенні ефективності управління чи зменшенні витрат на обробку й архівування. Чітко сформульовані завдання дозволяють вибудувати процес так, щоб він не лише відповідав вимогам безпеки, але й підтримував внутрішні бізнес-процеси [5], [6].

Після цього встановлюються категорії та критерії, які використовуються для розподілу даних. В основу класифікації можуть лягати метадані, типи файлів або інші властивості, що відображають чутливість і

значущість інформації. Рівні класифікації визначають, наскільки суворим має бути доступ і які обмеження слід застосувати [5], [6].

Щоб система класифікації була зрозумілою та ефективною, необхідно формалізувати політику, яка описує загальні принципи, таксономію, а також обов'язки відповідальних осіб. Така політика створює єдину базу для всіх учасників процесу та забезпечує узгоджене трактування категорій даних. Важливим елементом є чіткий розподіл ролей. Працівники повинні розуміти, які дії дозволені з інформацією різних рівнів чутливості, як архівувати непотрібні записи та яким чином повідомляти про інциденти. Для кожної категорії формуються стандарти безпеки, що враховують специфіку інформації та вимоги законодавства [5], [6].

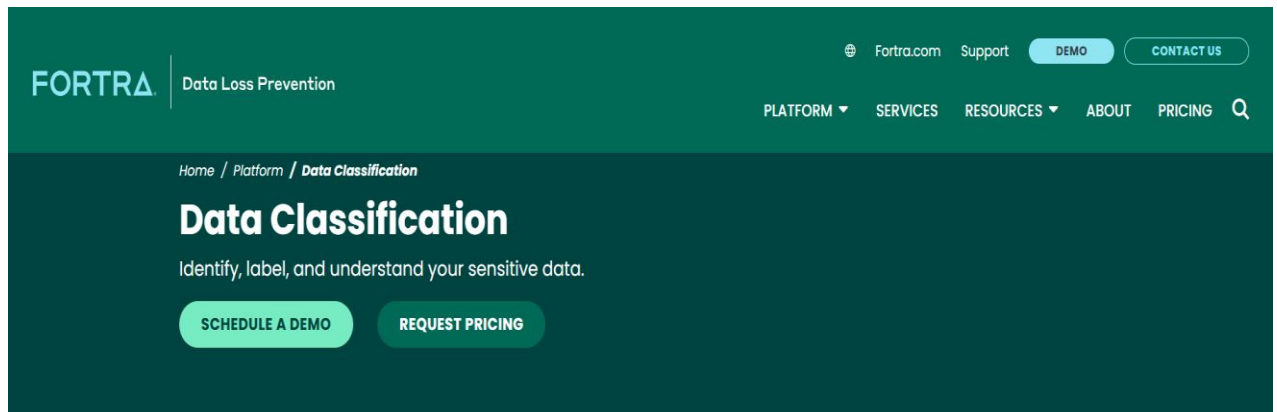
Оскільки середовище, у якому зберігаються та використовуються дані, постійно змінюється, процес класифікації не може залишатися статичним. Періодичний перегляд критеріїв дозволяє враховувати нові загрози, бізнес-завдання та нормативні зміни. Завдяки цьому система класифікації зберігає актуальність і забезпечує ефективний захист протягом тривалого часу.

Визначено, що класифікація даних являє собою процес віднесення об'єктів, записів або прикладів до певних категорій на основі їхніх характеристик, тобто це розподіл даних за заздалегідь визначеними групами згідно з певними ознаками. Класифікація даних є методом інтелектуального аналізу, що дозволяє прогнозувати належність нового об'єкта до певного класу, класифікація використовується тоді, коли структура класів відома заздалегідь, і завдання полягає у навчанні моделі для розпізнавання нових прикладів.

1.2 Аналіз засобів класифікації на основі даних навчальних вибірок

Проаналізуємо програмні засоби класифікації на основі даних навчальних вибірок [7]-[14].

Програмне забезпечення Fortra (рис. 1.1) розроблене для ефективної класифікації великих обсягів даних шляхом поєднання автоматизованих алгоритмів і можливостей ручного втручання. Його робота полягає у скануванні та присвоєнні міток інформації відповідно до змісту або контексту, при цьому враховуються вимоги до захисту конфіденційних відомостей, особливості інтелектуальної власності, характер даних, їхнє наповнення, походження чи потенційні ризики використання. У разі потреби результати класифікації можуть бути переглянуті та скориговані користувачем програмного забезпечення Fortra [7], [10], [12].



What is Data Classification?

Content, context and user classification. Digital Guardian data classification can automatically locate and identify your sensitive data then apply labels to classify and determine how the data is handled. Our comprehensive data classification solutions - from automated content and context-based classification to manual user classification - are optimized for regulatory compliance, intellectual property protection, and mixed environments.

By understanding what sensitive data you have, where it is, and



Рисунок 1.1 – Програмне забезпечення Fortra [12]

Програмне забезпечення Fortra інтегрується з інструментами Microsoft Office, що дозволяє ефективно працювати з документами, таблицями чи базами даних, а також підтримує основні операційні середовища, зокрема Windows, MacOS і Linux. Такий підхід забезпечує універсальність застосування та зручність упровадження у різні робочі процеси [7], [10], [12].

Додатково програмне забезпечення Fortra передбачає можливість контролю за обігом конфіденційних матеріалів на різних рівнях, включаючи робочі станції, корпоративні мережі та хмарні середовища. Завдяки цьому досягається не лише своєчасне виявлення потенційних загроз, а й реалізація політик безпеки безпосередньо під час обробки файлів, електронних повідомлень чи баз даних. Програмне забезпечення Fortra також підтримує інтеграцію з популярними рішеннями у сфері кіберзахисту, що дозволяє розширювати його функціональність та підвищувати загальну ефективність роботи з даними [7], [10], [12].

Основними характеристиками програмного забезпечення Fortra є такі [7], [10], [12]:

- автоматизація процесів: програмне забезпечення Fortra підтримує створення, управління та виконання складних бізнес-процесів без необхідності ручного втручання [7], [10], [12];

- безпека даних: програмне забезпечення Fortra надає користувачу інструменти для управління доступом, виявлення загроз та захисту інформаційних ресурсів [7], [10], [12];

- інтеграція систем: програмне забезпечення Fortra підтримує об'єднання різних корпоративних платформ і застосунків у єдине програмне середовище [7], [10], [12];

- управління IT інфраструктурою: програмне забезпечення Fortra дозволяє здійснювати моніторинг, аналіз та контроль роботи серверів, мереж і програмних сервісів [7], [10], [12];

- масштабованість рішень: програмне забезпечення Fortra може ефективно працювати як у малих організаціях, так і у великих підприємствах із високими навантаженнями [7], [10], [12];

- підтримка гібридних середовищ: Fortra забезпечує роботу з локальними ресурсами та хмарними платформами одночасно [7], [10], [12];

- зручність використання: програмне забезпечення Fortra має інтуїтивний інтерфейс і гнучкі інструменти для створення автоматизованих сценаріїв [7], [10], [12];

- аналітика та звітність: програмне забезпечення Fortra пропонує вбудовані засоби для збору даних, формування звітів і прийняття обґрунтованих рішень [7], [10], [12];

- підтримка кіберзахисту: програмне забезпечення Fortra інтегрує інструменти для реагування на інциденти та відповідності стандартам безпеки [7], [10], [12];

- оптимізація витрат: програмне забезпечення Fortra дозволяє знизити витрати на управління інфраструктурою завдяки автоматизації та централізованому контролю [7], [10], [12].

Серед недоліків програмного забезпечення Fortra можна відзначити високу вартість ліцензій, що може бути обмежувальним фактором для невеликих організацій. Деякі користувачі вказують на складність налаштування та потребу у значному часу для повної інтеграції в існуючу інфраструктуру. Також можлива залежність від спеціалізованої технічної підтримки, без якої ефективне використання продукту може бути ускладненим. Крім того, розширення функціоналу іноді вимагає придбання додаткових модулів, що збільшує загальні витрати на впровадження та обслуговування [7], [10], [12].

Програмне забезпечення Satori (рис. 1.2) являє собою платформу, призначену для автоматизованого виявлення та приховування конфіденційних даних у різних інформаційних середовищах. Програмне забезпечення здатне підключатися до численних сховищ і систем обробки, включаючи сучасні аналітичні сервіси, забезпечуючи зручне керування процесами без необхідності використання коду. Його робота ґрунтується на впровадженні заздалегідь визначених сценаріїв, які дозволяють проводити аналіз корпоративних даних, контролювати права доступу та формувати політики захисту залежно від джерела інформації [7], [10], [13].

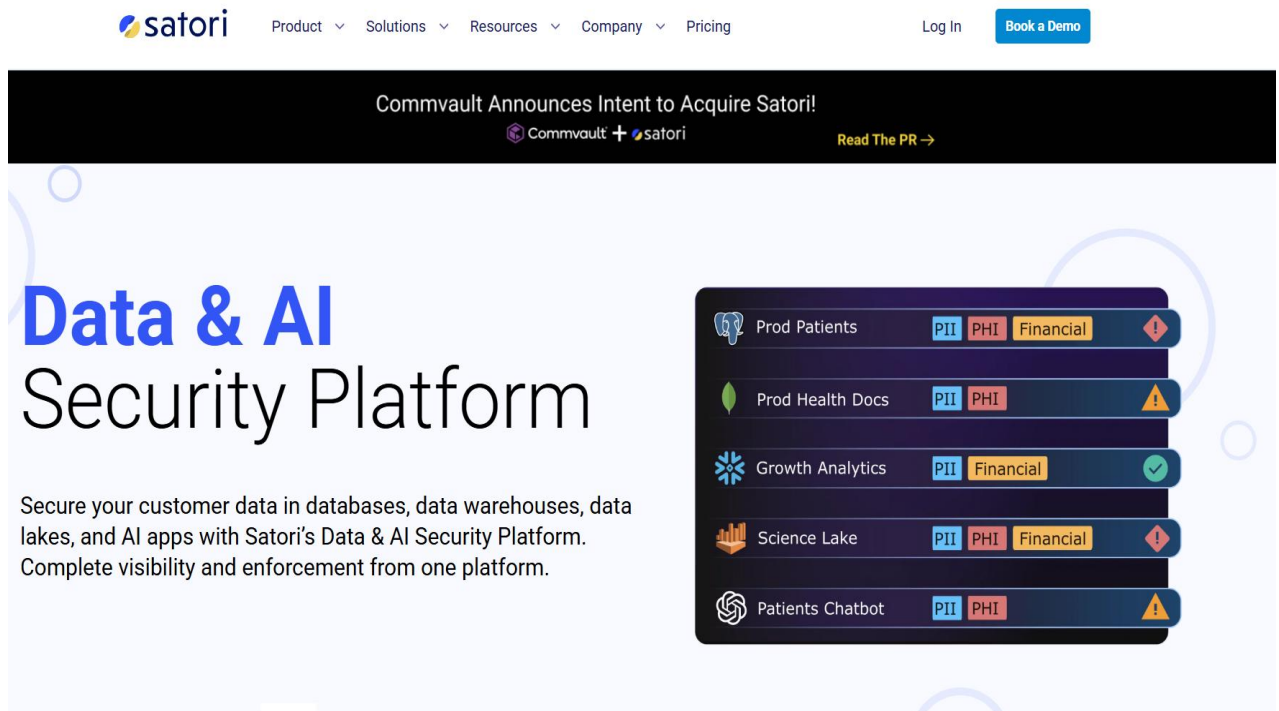


Рисунок 1.2 – Програмне забезпечення Satori [13]

Програмне забезпечення Satori надає можливість інтеграції з популярними корпоративними застосунками, що спрощує командний доступ до даних. У процесі використання платформи створюються умови для швидкого впровадження політик безпеки, адаптованих під особливості різних систем зберігання, при цьому основний акцент робиться на збереженні цілісності та конфіденційності інформації. Програмне забезпечення Satori не вимагає значних технічних знань і дозволяє організаціям застосовувати єдині механізми для контролю над інформаційними потоками як у хмарних, так і в локальних чи комбінованих середовищах. Таким чином, програмне забезпечення Satori виконує роль універсального інструмента для автоматизації класифікації та захисту даних, створюючи баланс між зручністю використання та рівнем безпеки. [7], [10], [13].

Програмне забезпечення Satori має такі особливості [7], [10], [13]:

– функціональність: програмне забезпечення Satori забезпечує можливості автоматизації бізнес-процесів і керування даними для різних сфер діяльності [7], [10], [13];

- інтеграція: програмне забезпечення Satori підтримує взаємодію з іншими корпоративними системами та сервісами, що полегшує роботу з наявною інфраструктурою [7], [10], [13];

- масштабованість: програмне забезпечення Satori дозволяє адаптуватися до зростання обсягів даних та збільшення кількості користувачів без втрати продуктивності [7], [10], [13];

- безпека: у програмному забезпеченні Satori застосовуються сучасні методи шифрування та контролю доступу, що гарантує захист конфіденційної інформації [7], [10], [13];

- зручність використання: інтерфейс програмного забезпечення Satori орієнтований на користувача, забезпечуючи інтуїтивно зрозумілу навігацію та легкість навчання персоналу [7], [10], [13];

- аналітика: передбачено інструменти для збору, обробки й візуалізації даних, що сприяє прийняттю ефективних управлінських рішень [7], [10], [13];

- продуктивність: оптимізовані алгоритми, що використовуються у програмному забезпеченні Satori, дозволяють швидко обробляти великі масиви даних та забезпечувати високу швидкодію системи [7], [10], [13];

- підтримка: користувачі програмного забезпечення Satori отримують технічний супровід і регулярні оновлення, що підвищує стабільність та надійність програмного продукту [7], [10], [13].

Серед недоліків програмного забезпечення Satori можна відзначити обмежену сумісність із деякими спеціалізованими інструментами, що може ускладнювати інтеграцію в існуючу інфраструктуру. Також певні функції програмного забезпечення Satori потребують додаткового налаштування, що вимагає від користувачів технічних знань і досвіду. У процесі роботи з великими обсягами даних можлива поява затримок у відображенні результатів, що знижує зручність використання. Окрім цього, для оптимальної роботи програмного забезпечення Satori необхідні значні обчислювальні ресурси, що може стати проблемою для компаній та користувачів із застарілим обладнанням [7], [10], [13].

Програмне забезпечення Netwrix (рис. 1.3) орієнтоване на забезпечення точного виявлення та систематизації даних, що мають підвищений рівень конфіденційності чи стратегічної важливості для компанії. Його використання дозволяє мінімізувати ризик несанкціонованого доступу, підвищити рівень інформаційної безпеки та водночас спростити завдання, пов'язані з дотриманням регуляторних вимог. Програмне забезпечення Netwrix здатне працювати як із локальними сховищами, так і з хмарними середовищами, створюючи цілісне уявлення про розподіл та характер інформації у межах організації [8], [10], [14].

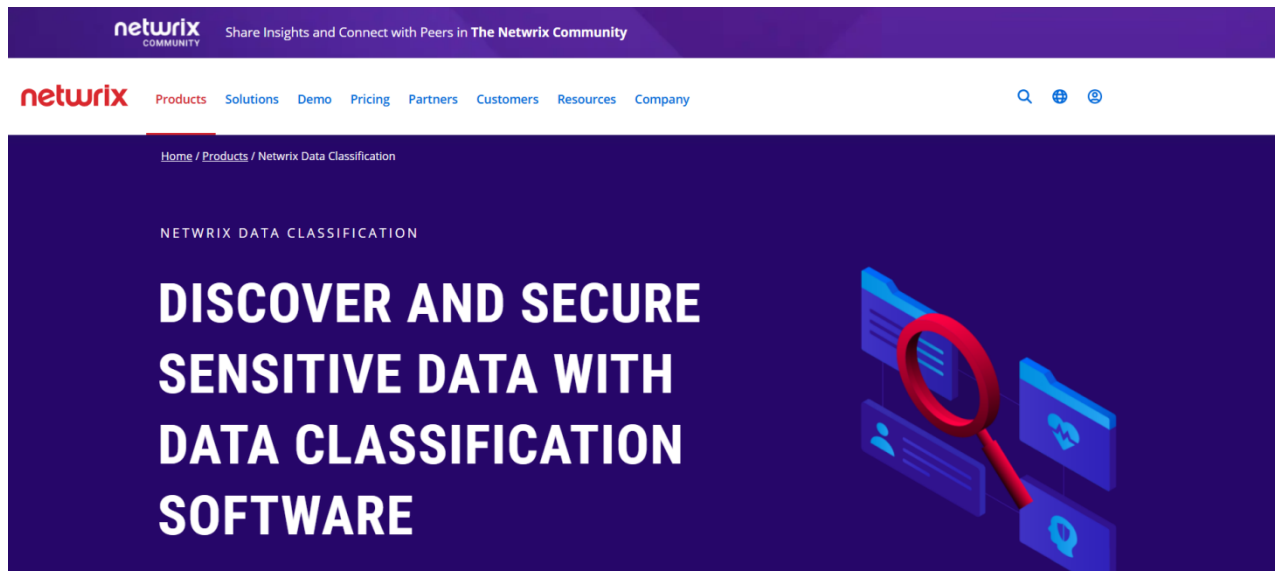


Рисунок 1.3 – Програмне забезпечення Netwrix [14]

Завдяки впровадженим інструментам статистичного аналізу й обробки складних мовних конструкцій досягається більш релевантна класифікація, ніж у випадку звичайного пошуку чи поверхневої семантичної оцінки. Підхід передбачає можливість автоматичного виявлення уразливих точок, наприклад відкритого доступу до чутливих файлів, та їх негайної ізоляції до моменту впровадження комплексної стратегії захисту [8], [10], [14].

Важливою особливістю програмного забезпечення Netwrix є наявність заздалегідь визначених правил і шаблонів, що відповідають міжнародним стандартам безпеки та конфіденційності, завдяки чому процес налаштування

виконується швидко і не потребує значних зусиль. У результаті організація отримує не лише ефективний інструмент захисту даних, а й можливість формування культури усвідомленого управління інформаційними ресурсами, що сприяє підвищенню довіри та стабільності бізнес-процесів [8], [10], [14].

Програмне забезпечення Netwrix має такі особливості [8], [10], [14]:

- виявлення конфіденційних даних: програмне забезпечення Netwrix дозволяє точно ідентифікувати та класифікувати інформацію, що містить персональні або критично важливі для користувача дані, незалежно від місця зберігання [8], [10], [14];

- контроль доступу: Netwrix забезпечує управління правами користувачів на основі ролей та політик безпеки [8], [10], [14];

- відповідність нормативам: програмне забезпечення Netwrix використовує попередньо створені шаблони, що відповідають вимогам GDPR, PCI DSS та HIPAA [8], [10], [14];

- інтеграція з системами: Netwrix підтримує Active Directory та Netwrix Auditor для централізованого управління [8], [10], [14];

- моніторинг та аудит: програмне забезпечення Netwrix дозволяє відстежувати зміни у файлових серверах та інших джерелах [8], [10], [14];

- аналітика даних: програмне забезпечення Netwrix забезпечує глибокий аналіз унікальних моделей даних користувача та підвищує точність класифікації [8], [10], [14];

- гнучкість розгортання: програмне забезпечення Netwrix є сумісним із локальними та хмарними середовищами для оптимального управління даними користувача [8], [10], [14];

- автоматизація процесів: програмне забезпечення Netwrix дозволяє знижувати ручну роботу завдяки налаштованим правилам та політикам обробки інформації [8], [10], [14];

- захист інформації: програмне забезпечення Netwrix ізолює дані, що зберігаються у відкритому доступі, для зменшення ризиків витоку даних користувача [8], [10], [14];

– підтримка продуктивності: програмне забезпечення Netwrix оптимізує роботу співробітників через швидкий доступ до релевантної інформації [8], [10], [14].

Серед недоліків програмного забезпечення Netwrix можна відзначити застарілий інтерфейс користувача, який ускладнює роботу для нових користувачів, обмежену орієнтацію на хмарні сервіси та підвищені вимоги до ресурсів при масштабуванні, що іноді уповільнює процес класифікації великих обсягів даних [8], [10], [14].

Порівняльну характеристику програмних засобів класифікації на основі даних навчальних вибірок наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмних засобів класифікації на основі даних навчальних вибірок

Критерій порівняння	Fortra [12]	Satori [13]	Netwrix [14]
Підтримка хмарних середовищ	+–	+	+–
Простота використання	+	+–	+–
Велика кількість аналітичних можливостей	+	+–	+
Ресурсоємність	–	+–	–
Зручність інтерфейсу користувача	+–	+	–
Гнучкість у налаштуванні	+	+	+–

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів класифікації на основі даних навчальних вибірок. Проте ефективність програмних засобів для класифікації на основі даних навчальних вибірок при визначенні віку морських молюсків суттєво залежить від обсягу та якості навчальних вибірок. Для багатьох видів молюсків бракує достатньої кількості достовірних даних, що ускладнює

побудову моделей із високою точністю. Крім того, алгоритми машинного навчання, що використовуються у таких програмних засобах, часто демонструють обмежену універсальність: модель, навчена на одному виді чи популяції, може працювати ненадійно при застосуванні до інших умов або видів. Ще однією проблемою є складність інтерпретації результатів. Крім того, програмні засоби для класифікації на основі даних навчальних вибірок потребують значних обчислювальних ресурсів і спеціалізованого обладнання, що обмежує їх використання у польових умовах. Тому актуальною є розробка програмного забезпечення для визначення віку морських молюсків.

При розробці програмного забезпечення для визначення віку морських молюсків необхідно забезпечити такі функціональні вимоги:

- отримання вхідних даних про екземпляри молюсків;
- виконання попередньої обробки даних про екземпляри молюсків;
- формування навчальної вибірки для побудови моделей класифікації;
- навчання класифікаційних моделей для розпізнавання вікових характеристик молюсків;
- тестування моделей для перевірки точності їх роботи;
- порівняння результатів між різними алгоритмами та вибір найбільш ефективного підходу;
- візуалізація отриманих результатів для зручного сприйняття.

1.3 Висновки за розділом 1

Визначено, що класифікація даних являє собою процес віднесення об'єктів, записів або прикладів до певних категорій на основі їхніх характеристик, тобто це розподіл даних за заздалегідь визначеними групами згідно з певними ознаками. Класифікація даних є методом інтелектуального аналізу, що дозволяє прогнозувати належність нового об'єкта до певного класу, класифікація використовується тоді, коли структура класів відома

заздалегідь, і завдання полягає у навчанні моделі для розпізнавання нових прикладів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів класифікації на основі даних навчальних вибірок. Проте ефективність програмних засобів для класифікації на основі даних навчальних вибірок при визначенні віку морських молюсків суттєво залежить від обсягу та якості навчальних вибірок. Для багатьох видів молюсків бракує достатньої кількості достовірних даних, що ускладнює побудову моделей із високою точністю. Крім того, алгоритми машинного навчання, що використовуються у таких програмних засобах, часто демонструють обмежену універсальність: модель, навчена на одному виді чи популяції, може працювати ненадійно при застосуванні до інших умов або видів. Ще однією проблемою є складність інтерпретації результатів. Крім того, програмні засоби для класифікації на основі даних навчальних вибірок потребують значних обчислювальних ресурсів і спеціалізованого обладнання, що обмежує їх використання у польових умовах. Тому актуальною є розробка програмного забезпечення для визначення віку морських молюсків.

Сформульовано функціональні вимоги до програмного забезпечення для визначення віку морських молюсків.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для визначення віку морських моллюсків було використано мову програмування Python [15], [16].

Мова Python є одним із найефективніших інструментів у галузі програмування, адже поєднує зрозумілий синтаксис із широкими можливостями застосування. Його використовують у наукових обчисленнях, під час аналізу великих масивів інформації, у створенні веборієнтованих рішень та автоматизації технологічних процесів. Простота мови дозволяє швидко оволодіти її основами навіть початківцям, тоді як досвідченим розробникам вона відкриває шлях до реалізації складних і ресурсомістких завдань без значних витрат часу [15], [16].

Високий рівень поширеності Python зумовлений постійним розвитком його екосистеми, яку підтримує активна міжнародна спільнота. Завдяки новим бібліотекам і вдосконаленим інструментам розширюються сфери використання мови та підвищується її практична цінність. Важливою перевагою є також здатність легко взаємодіяти з різними платформами й технологіями, що робить Python придатним для багатопрофільних проєктів. У результаті ця мова набула статусу універсального засобу, який активно застосовується як у наукових дослідженнях, так і в комерційних та промислових рішеннях [15], [16].

У розробленні програмного забезпечення, спрямованого на визначення віку морських моллюсків, доцільно застосовувати мову Python, оскільки вона вирізняється універсальністю та значним потенціалом у сфері аналізу даних. Її інструментарій містить велику кількість бібліотек, які забезпечують ефективну реалізацію методів машинного навчання, штучного інтелекту та обробки зображень, що є особливо важливим для дослідження біологічних структур і побудови прогнозних моделей. Завдяки зрозумілому синтаксису прискорюється створення прототипів та спрощується перевірка різних

алгоритмічних підходів, що істотно підвищує результативність наукової роботи [15], [16].

Популярність мови програмування Python пояснюється активною підтримкою з боку наукової та професійної спільноти, яка постійно вдосконалює існуючі рішення й пропонує нові інструменти для досліджень. Це забезпечує можливість інтеграції сучасних технологій та використання перевірених практик у спеціалізованих проєктах. Поєднання продуктивності, гнучкості та доступності робить Python найоптимальнішим варіантом для створення програмних засобів, що застосовуються у вивченні біологічних об'єктів [15], [16].

Обґрунтування вибору мови програмування для реалізації засобів для визначення віку морських молюсків наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для створення засобів для визначення віку морських молюсків

Критерій порівняння мов програмування	Мова програмування		
	Python	R	MATLAB
Простота синтаксису	+	+-	+-
Зручність обробки зображень	+	-	+-
Швидкість розробки	+	+-	+-
Зручність для створення ПЗ для визначення віку морських молюсків	+	-	-
Гнучкість інтеграції з іншими системами	+	+-	-
Активність спільноти	+	+-	+-

Отже, для реалізації програмного забезпечення для визначення віку морських молюсків обрано мову програмування Python, яка є її гнучкою та

надає широкі можливості у сфері обробки даних. Крім того, мова Python має багатий набір бібліотек, які дозволяють ефективно працювати з алгоритмами штучного інтелекту, машинного навчання та комп'ютерного зору, що є особливо важливим у задачах аналізу біологічних зображень та побудови моделей прогнозування. Простота синтаксису мови Python забезпечує швидке розроблення прототипів і полегшує тестування різних підходів, що сприяє підвищенню продуктивності наукових досліджень.

2.2 Вибір середовища розробки для створення програмного забезпечення для визначення віку морських молюсків

В якості середовища розробки для створення програмного забезпечення для визначення віку морських молюсків було обрано середовище Jupyter Notebook [17], [18].

Jupyter Notebook є інноваційний середовищем розробки, у якому поєднуються засоби написання коду, візуалізація результатів та пояснювальні матеріали. Його структура дає змогу перетворювати процес роботи з алгоритмами на інтерактивне дослідження, де аналітичні обчислення тісно інтегруються з текстовими коментарями та графічними ілюстраціями. Такий формат організації забезпечує не лише зручність для проведення індивідуальних експериментів, а й створює основу для колективного використання, коли результати можуть бути швидко перевірені та відтворені іншими учасниками наукової спільноти [17], [18].

Відкрита архітектура середовища Jupyter Notebook дозволяє застосовувати його в поєднанні з різними мовами програмування та аналітичними інструментами, що значно розширює межі його використання. Попит серед дослідників пояснюється не лише універсальністю Jupyter Notebook, а й можливістю формування інтерактивних документів, у яких поєднуються аналітичні висновки та графічні матеріали. Завдяки такій гнучкості це середовище поступово стало загальноприйнятим інструментом у

сфері роботи з даними, де необхідно одночасно досягати високої точності розрахунків і забезпечувати зрозумілу візуалізацію результатів [17], [18].

Застосування середовища Jupyter Notebook у проєктах, пов'язаних із визначенням віку морських моллюсків, є доцільним завдяки його здатності поєднувати код, графічні результати та пояснювальний матеріал у єдиному робочому середовищі. Такий підхід сприяє більшій наочності дослідження та забезпечує можливість поступово простежувати виконання алгоритмів, що має ключове значення при роботі з біологічними даними та моделями машинного навчання. Покрокове виконання коду створює умови для оперативного коригування, перевірки альтернативних підходів і миттєвої оцінки їхнього впливу на якість побудованих моделей [17], [18].

Завдяки своїй відкритій архітектурі Jupyter Notebook підтримує інтеграцію з різними мовами програмування та інструментами аналізу, що робить його придатним для широкого спектра наукових і прикладних завдань. Його популярність у дослідницьких колах пояснюється не лише гнучкістю, але й можливістю створення інтерактивних звітів, які поєднують аналітику та наочні візуалізації. Це середовище поступово перетворилося на стандарт у сфері науки про дані, де важливо поєднати точність обчислень із зрозумілою та структурованою подачею результатів [17], [18].

Середовище розробки Jupyter Notebook легко взаємодіє з бібліотеками мови програмування Python, що відкриває широкі можливості для роботи з об'ємними наборами даних, побудови графічних ілюстрацій та реалізації методів штучного інтелекту. Додатковою перевагою середовища розробки Jupyter Notebook є можливість інтеграції аналітичного процесу з текстовими поясненнями в межах одного файлу, що значно підвищує відтворюваність досліджень та полегшує їхнє поширення серед інших науковців. Поєднання інтерактивності, універсальності й зручності використання робить Jupyter Notebook найбільш ефективним середовищем для створення прикладних рішень у сфері морської біології [17], [18].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для визначення віку морських молюсків наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для визначення віку морських молюсків

Критерій порівняння середовищ розробки	Середовища розробки		
	Jupyter Notebook	Spyder	Atom
Інтерактивність та покрокове виконання	+	+–	–
Зручність документування та візуалізації	+	+–	+–
Легкість початкового освоєння	+	+	+–
Підтримка колективної роботи	+	+–	+–
Відсутність потреби у додаткових налаштуваннях	+	+	+–

Таким чином, для створення програмного забезпечення для визначення віку морських молюсків обрано середовище розробки Jupyter Notebook завдяки його здатності поєднувати програмний код, візуалізації та пояснювальний текст в єдиному інтерактивному просторі. Такий підхід робить дослідження більш прозорим і дозволяє поступово відстежувати результати виконання алгоритмів, що особливо важливо під час роботи з біологічними даними та методами машинного навчання. Зручність покрокового виконання фрагментів коду дає змогу оперативно вносити зміни, експериментувати з різними параметрами й одразу оцінювати вплив цих змін на точність моделі.

Крім того, середовище Jupyter Notebook забезпечує тісну інтеграцію з численними бібліотеками Python, що значно спрощує роботу з великими масивами даних, побудову графіків і реалізацію алгоритмів штучного інтелекту.

2.3 Метод класифікації на основі кількох алгоритмів машинного навчання, що працюють у взаємодоповнюючій структурі

У сучасній практиці аналізу даних класифікаційні методи застосовуються досить широко, однак кожен з них має певні обмеження, які ускладнюють отримання максимально надійних результатів. Традиційні алгоритми часто виявляють надмірну чутливість до шуму та неповних даних, що призводить до зниження точності у випадках реальних складних вибірок. Часто вони потребують попереднього балансування класів або ручного налаштування великої кількості параметрів, що робить процес побудови моделі трудомістким і залежним від досвіду аналітика. Значна частина відомих методів тяжіє до узагальнених рішень, не враховуючи контекстуальних взаємозв'язків між ознаками, унаслідок чого втрачається гнучкість при адаптації до нових умов. Крім того, при збільшенні обсягів даних виникає проблема масштабованості: алгоритми починають працювати повільніше або потребують надмірних обчислювальних ресурсів. Відсутність здатності до динамічного навчання без повного перенавчання робить багато методів малоприматними для середовищ, де дані змінюються у реальному часі. Сукупність цих факторів свідчить про потребу у підході, який поєднує гнучкість, стійкість до шуму, ефективність у великих масивах та здатність до швидкої адаптації. У цьому контексті доцільним видається застосування нового підходу, що може бути позначений як адаптивна контекстуально-інтегрована класифікація, оскільки він орієнтується не лише на ізольовані характеристики даних, а й на їхні взаємозв'язки та динамічну зміну структур у процесі аналізу.

Запропонований метод класифікації базується на поєднанні кількох алгоритмів машинного навчання, що працюють у взаємодоповнюючій структурі. Основна ідея полягає в тому, що для завдань визначення віку морських моллюсків найбільш доцільно використовувати не один алгоритм, а ансамбль моделей, які здатні компенсувати недоліки одна одної. Саме тому створено спеціальну модель, у якій результати кількох класифікаторів інтегруються у спільний прогноз, що дозволяє підвищити надійність і точність отриманих висновків.

На початковому етапі здійснюється підготовка даних, оскільки саме від цього залежить коректність подальшого моделювання. Вхідні дані формуються з різних джерел, після чого категоріальні ознаки переводяться у числовий формат, а значущість окремих характеристик оцінюється за допомогою міри взаємної інформації. Це забезпечує можливість відбору лише тих параметрів, які мають найбільший вплив на результат, що, у свою чергу, дозволяє зменшити кількість вхідних змінних та оптимізувати швидкодію алгоритму.

Наступним кроком є побудова навчальної та тестової вибірок, що гарантує об'єктивну оцінку точності класифікації. У цьому процесі дані розподіляються випадковим чином, щоб уникнути упередженості під час навчання та перевірки моделі. Саме такий підхід дозволяє отримати більш реалістичну картину того, як алгоритм буде поводитися під час роботи з новими невідомими прикладами.

Особливу увагу приділено вибору базових моделей, які входять до ансамблю. Використано поєднання випадкового лісу та лінійного дискримінантного аналізу. Перша модель добре працює з нелінійними залежностями та здатна враховувати складні комбінації ознак, тоді як друга модель вирізняється простотою, швидкодією та високою ефективністю у випадках, де дані мають майже лінійний характер. Об'єднання цих двох підходів дозволяє охопити ширший спектр можливих закономірностей.

Для інтеграції результатів застосовується механізм голосування, де кожен з алгоритмів формує власний прогноз, а кінцевий результат визначається за підсумком їхньої взаємодії. Такий підхід забезпечує більшу стійкість до шумових даних та випадкових похибок, оскільки помилки однієї моделі можуть бути компенсовані правильною оцінкою іншої. Завдяки цьому зменшується ймовірність суттєвих відхилень у прогнозуванні.

Додатковим елементом методу є автоматизований пошук оптимальних параметрів для випадкового лісу. Для цього використовується стратегія перебору гіперпараметрів із застосуванням крос-валідації, що дозволяє знайти найкращі конфігурації для конкретного набору даних. Такий підхід гарантує підвищення точності класифікації без необхідності ручного налаштування параметрів, що робить метод більш універсальним і придатним для практичного використання у різних дослідницьких умовах.

Фінальний етап передбачає оцінку якості моделі на тестових даних, де фіксуються показники точності та продуктивності. Отримані результати узагальнюються у таблиці, що дозволяє порівняти ефективність різних алгоритмів та підтвердити переваги ансамблевого підходу. У результаті створюється інструмент, здатний одночасно забезпечувати високу точність класифікації та стабільність роботи при різних наборах вхідних характеристик.

Отже, етапи методу передбачають підготовку даних і кодування ознак, відбір ключових характеристик за взаємною інформацією, формування навчальної та тестової вибірок, побудову ансамблю на основі випадкового лісу та дискримінантного аналізу, застосування механізму голосування, оптимізацію гіперпараметрів за допомогою крос-валідації та фінальну перевірку точності на тестових даних.

Графічне подання методу класифікації на основі кількох алгоритмів машинного навчання, що працюють у взаємодоповнюючій структурі, наведено на рис. 2.1.



Рисунок 2.1 – Графічне подання методу класифікації на основі кількох алгоритмів машинного навчання

Запропонований метод суттєво відрізняється від існуючих рішень тим, що поєднує дві моделі з різною природою побудови та компенсує їхні слабкі сторони через механізм голосування. Додатковим елементом є оптимізація параметрів випадкового лісу, що робить метод більш гнучким та адаптивним.

Таким чином, запропоновано метод класифікації на основі кількох алгоритмів машинного навчання, що працюють у взаємодоповнюючій структурі, який відрізняється від існуючих рішень тим, що поєднує дві моделі з різною природою побудови та компенсує їхні слабкі сторони через механізм голосування.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для визначення віку морських молюсків обрано мову програмування Python, яка є її гнучкою та надає широкі можливості у сфері обробки даних. Крім того, мова Python має багатий набір бібліотек, які дозволяють ефективно працювати з алгоритмами штучного інтелекту, машинного навчання та комп'ютерного зору, що є особливо важливим у задачах аналізу біологічних зображень та побудови моделей прогнозування. Простота синтаксису мови Python забезпечує швидке розроблення прототипів і полегшує тестування різних підходів, що сприяє підвищенню продуктивності наукових досліджень.

Для створення програмного забезпечення для визначення віку морських молюсків обрано середовище розробки Jupyter Notebook завдяки його здатності поєднувати програмний код, візуалізації та пояснювальний текст в єдиному інтерактивному просторі. Такий підхід робить дослідження більш прозорим і дозволяє поступово відстежувати результати виконання алгоритмів, що особливо важливо під час роботи з біологічними даними та методами машинного навчання. Зручність покрокового виконання фрагментів коду дає змогу оперативно вносити зміни, експериментувати з різними параметрами й одразу оцінювати вплив цих змін на точність моделі. Крім

того, середовище Jupyter Notebook забезпечує тісну інтеграцію з численними бібліотеками Python, що значно спрощує роботу з великими масивами даних, побудову графіків і реалізацію алгоритмів штучного інтелекту.

Запропоновано метод класифікації на основі кількох алгоритмів машинного навчання, що працюють у взаємодоповнюючій структурі, який відрізняється від існуючих рішень тим, що поєднує дві моделі з різною природою побудови та компенсує їхні слабкі сторони через механізм голосування.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для визначення віку морських молюсків

Структуру програмного забезпечення для визначення віку морських молюсків наведено на рис. 3.1.



Рисунок 3.1 – Структура програмного забезпечення для визначення віку морських молюсків

Структура програмного забезпечення для визначення віку морських молюсків формується як єдина інтегрована система, що складається з п'яти взаємопов'язаних модулів. Основними модулями програмного забезпечення

для визначення віку морських молюсків є такі: модуль попередньої обробки зображень, аналітичний модуль, модуль інтелектуальної інтерпретації результатів, модуль візуалізації даних, модуль управління та інтеграції.

Модуль попередньої обробки зображень забезпечує функціонування початкового етапу підготовки даних, на якому виконується перетворення вхідних зображень у стандартизований формат, придатний для подальшого аналізу. На цьому рівні застосовуються методи фільтрації шумів, нормалізації кольорів, підвищення контрастності та виділення ключових структурних елементів. Його завдання полягає у тому, щоб максимально знизити вплив зовнішніх факторів та артефактів, які можуть ускладнити інтерпретацію.

Аналітичний модуль виконує розпізнавання закономірностей у візерунках та застосовує алгоритми машинного навчання для визначення вікових ознак. Його функції включають сегментацію шарів, пошук регулярних повторів та визначення характерних маркерів росту.

Модуль інтелектуальної інтерпретації результатів відповідає перетворення вихідних даних аналізу в зрозумілу для дослідників форму. Він дозволяє отримувати попередні висновки щодо віку та темпів росту молюсків, а також формує текстові та числові пояснення. У його межах відбувається поєднання результатів аналізу з предметними знаннями, формування висновків на основі логічних правил або моделей машинного навчання, а також виявлення можливих сценаріїв розвитку подій.

Модуль візуалізації даних забезпечує створення графіків та діаграм, відповідає за наочне представлення даних, що значно полегшує роботу дослідників у процесі перевірки результатів.

Модуль управління та інтеграції виконує координуючу функцію. Він забезпечує узгоджену взаємодію між усіма іншими компонентами системи, регулює потоки даних та синхронізує процеси. Крім того, він відповідає за інтеграцію з зовнішніми платформами та сервісами, що дозволяє системі працювати як частині більшої інфраструктури. У його межах відбувається

контроль доступу, налаштування параметрів роботи та адаптація системи під конкретні завдання, що робить її гнучкою та масштабованою.

Таким чином, усі модулі утворюють цілісне програмне забезпечення, здатне автоматизувати процес визначення віку морських молюсків із високим рівнем точності та зручності для науковців.

3.2 Функціонування програмного забезпечення для визначення віку морських молюсків

Функціонування програмного забезпечення для визначення віку морських молюсків подамо за допомогою схеми, зображеної на рис. 3.2.

Функціонування програмного забезпечення для визначення віку морських молюсків побудовано таким чином, щоб участь користувача зводилася до мінімуму. Передбачено можливість обрати джерело вхідних даних та за потреби здійснити налагодження, тому необхідність у розробці графічного інтерфейсу виявилася недоцільною. Збереження підсумкових результатів у зовнішні файли також не було визначено пріоритетним, оскільки обсяги вибірки є відносно невеликими, а кількість параметрів не створює суттєвого навантаження на систему.

Робота програми починається з визначення способу отримання вхідних даних. Якщо обрано локальне завантаження, інформація зчитується з підготовленого файлу, а у випадку відмови від цього варіанта використовується автоматичне завантаження з вебресурсу. Незалежно від способу, наступним етапом є попередня обробка даних, де усуваються зайві фактори, проводиться нормалізація та підготовка набору для подальшого аналізу.

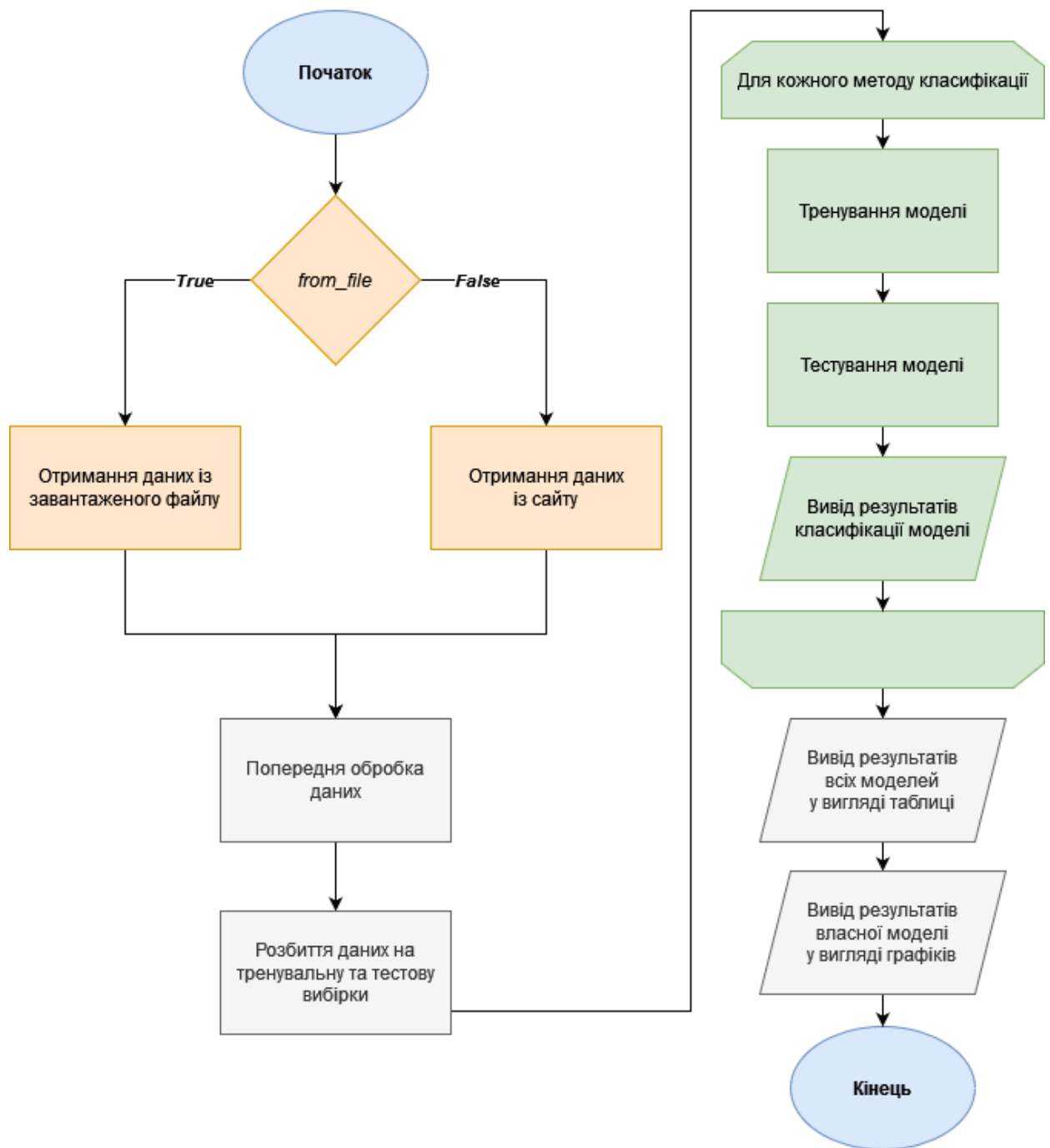


Рисунок 3.2 – Схема функціонування програмного забезпечення для визначення віку морських моллюсків

Після завершення цього кроку дані розділяються на навчальну та тестову частини. Такий підхід дозволяє коректно оцінювати ефективність роботи моделей та уникати їх перенавчання. Далі запускається цикл, у межах якого кожен із запропонованих методів класифікації проходить етап

тренування на навчальній вибірці, після чого перевіряється його здатність коректно обробляти тестові дані.

Результати класифікації, отримані від кожного з методів, систематизуються у вигляді підсумкової таблиці. У ній зберігається ключова інформація щодо точності, швидкодії та інших характеристик роботи алгоритмів. Це дозволяє здійснити зіставлення різних підходів у зручному для сприйняття форматі.

Для поглибленого аналізу передбачено додаткове представлення результатів за допомогою графічних засобів. Побудовані візуалізації дають змогу чіткіше побачити закономірності, які важко відчитати лише з табличних даних, та виявити приховані залежності.

Завершальним етапом роботи системи є узагальнення отриманих результатів і завершення виконання програми. Таким чином забезпечується логічно завершений цикл від отримання вихідних даних до формування підсумкових висновків, що робить програмне забезпечення зручним інструментом для досліджень у галузі морської біології.

3.3 Розробка моделей програмного забезпечення для визначення віку морських молюсків

Діаграма прецедентів програмного забезпечення для визначення віку морських молюсків (рис. 3.3) відображає взаємодію користувача із ключовими функціональними можливостями системи. У центрі діаграми перебуває користувач, який виступає ініціатором основних дій. Від нього спрямовані зв'язки до трьох основних прецедентів, кожен із яких окреслює окрему логіку роботи з програмою. Першим прецедентом є встановлення доступних параметрів, що передбачає вибір джерела вхідних даних та налаштування додаткових умов виконання. Цей етап визначає спосіб подальшого функціонування програми, оскільки від обраних параметрів залежить шлях обробки інформації.

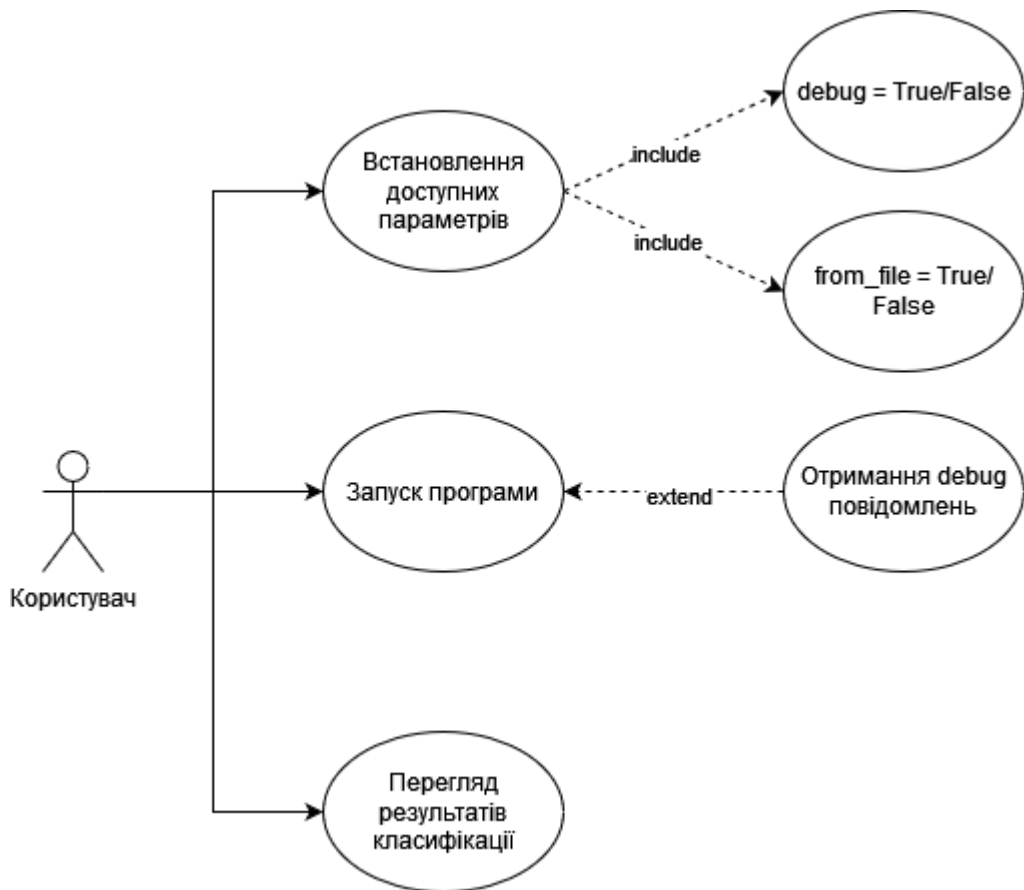


Рисунок 3.3 – Діаграма прецедентів програмного забезпечення для визначення віку морських молюсків

Другим прецедентом є запуск програми. На цьому етапі система починає виконання усіх алгоритмів, що включають попередню обробку даних, їх розподіл на навчальні та тестові вибірки, навчання моделей та перевірку їхньої точності. Саме тут реалізується основний процес, спрямований на класифікацію віку морських молюсків.

Третім прецедентом є перегляд результатів класифікації. Після завершення обчислювальних процедур користувачу відкривається можливість ознайомитися з підсумками у вигляді таблиць і графіків, що відображають точність та інші характеристики роботи моделей.

Таким чином, діаграма прецедентів ілюструє просту та зрозумілу структуру взаємодії користувача із системою, у якій його роль зводиться до налаштування, запуску та аналізу результатів. Завдяки такій побудові

забезпечується інтуїтивність використання програмного забезпечення та ефективність проведення досліджень.

Діаграму діяльності програмного забезпечення для визначення віку морських молюсків наведено на рис. 3.4.



Рисунок 3.4 – Діаграма діяльності ПЗ для визначення віку морських молюсків

Діаграма діяльності програмного забезпечення для визначення віку морських молюсків відображає послідовність дій, які виконуються системою від початку роботи до завершення обчислень. На початку процесу позначено стартовий символ, з якого спрямовується перехід до етапу отримання вхідних даних. У цьому місці передбачено розгалуження, що дозволяє реалізувати два варіанти: завантаження даних із локального файлу або отримання їх безпосередньо з мережі. Незалежно від обраного варіанту, обидва шляхи сходяться в єдину гілку, яка веде до подальшої обробки.

На наступному етапі діяльності програмного забезпечення для визначення віку морських молюсків здійснюється попередня підготовка даних, яка включає очищення, нормалізацію та трансформацію ознак у зручний формат. Це дає змогу сформуванню якісної основи для подальших розрахунків і забезпечити стабільність функціонування алгоритмів. Після цього активність переходить до підбору параметрів власної моделі, де проводиться оптимізація налаштувань, спрямована на досягнення найкращої точності під час класифікації.

Потім у процесі діаграми позначено ключовий етап, який полягає у класифікації даних. На цьому рівні модель аналізує підготовлені характеристики та формує прогноз щодо віку морських молюсків. Після виконання цього кроку діяльності програмного забезпечення для визначення віку морських молюсків відбувається виведення результатів, що передбачає представлення отриманих висновків у зручному вигляді, наприклад у формі таблиць або графічних ілюстрацій.

Завершальним елементом діаграми є символ кінця, який позначає закінчення роботи програмного забезпечення після представлення користувачу результатів класифікації.

Така структура діаграми діяльності програмного забезпечення для визначення віку морських молюсків демонструє логічну послідовність усіх кроків і відображає цілісний алгоритм функціонування системи.

3.4 Особливості реалізації програмного забезпечення для визначення віку морських молюсків

Особливості реалізації програмного забезпечення для визначення віку морських молюсків пов'язані з необхідністю забезпечення прозорості роботи системи та можливості гнучкого керування процесом класифікації. Для цього було застосовано підхід, що передбачає використання спеціального класу-переліку, який дозволяє кодувати певні етапи виконання. Завдяки цьому стало можливим організувати структуру повідомлень, що інформують про перебіг окремих процесів. Кожен елемент цього переліку відповідає конкретному стану або події, що виникає під час підготовки чи запуску моделі, а також під час тестування чи завершення певних обчислювальних дій. Такий підхід дає змогу не тільки вести простий журнал, а й забезпечувати більш інтуїтивний контроль за виконанням кроків програми.

У рамках налагодження передбачено спеціальну функцію, яка реалізує виведення діагностичних повідомлень. Її робота побудована так, що користувач отримує інформацію лише тоді, коли це є необхідним. Якщо в глобальних налаштуваннях увімкнено режим налагодження або йдеться про критично важливі повідомлення, функція формує вихідний текст. При цьому враховується тип повідомлення: воно може бути заздалегідь заданим або динамічно згенерованим залежно від параметрів. Завдяки такій організації інформаційний потік не перевантажується другорядними повідомленнями, і водночас користувач не втрачає важливих даних щодо перебігу роботи алгоритму.

Для оцінки часових витрат передбачено окрему функцію, яка виконує перетворення системного часу у зручний формат. Це дозволяє легко аналізувати продуктивність системи та отримувати дані про те, скільки часу займає конкретний процес. Оскільки класифікаційні алгоритми можуть вимагати значних ресурсів, така можливість є важливою для подальшої оптимізації. Перетворення часу здійснюється автоматично, що позбавляє

потреби у додаткових обчисленнях і робить аналіз результатів більш доступним.

У процесі реалізації передбачено механізм автоматичного підбору параметрів моделі. Для цього використовується функція, яка працює з наборами ознак та класів і підбирає оптимальні конфігурації для власного класифікатора. У своїй роботі ця функція спирається на попередньо описані допоміжні методи, що забезпечують як вимірювання часу, так і виведення проміжних результатів. Завдяки цьому процедура налаштування моделі стає не лише автоматизованою, а й прозорою для користувача.

Центральним елементом є спеціально створений клас класифікатора, який поєднує кілька підходів машинного навчання. У його основі лежить ідея використання ансамблевих методів, що дають змогу комбінувати різні алгоритми для досягнення більшої точності. Клас підтримує як пряме навчання на основі заданих параметрів, так і навчання з попереднім пошуком оптимальних налаштувань. Такий підхід робить модель більш гнучкою та дозволяє адаптувати її до різних наборів даних без значної втрати якості.

Важливою особливістю цього класу є можливість масштабування даних, що дозволяє уникнути викривлень під час роботи алгоритмів. Додатково реалізовано методи для оцінки точності класифікації та візуалізації результатів. Користувач має змогу отримати не лише числові показники ефективності, а й графічне відображення співвідношення між фактичними та прогнозованими значеннями. Це значно полегшує аналіз якості класифікації та дозволяє швидко виявляти помилки чи закономірності.

Завдяки такій архітектурі програмне забезпечення поєднує простоту використання з високим рівнем функціональності. Реалізовані допоміжні класи і функції забезпечують прозорий контроль над усіма етапами обробки, а власний класифікатор створює умови для гнучкої роботи з даними та оптимізації параметрів. У підсумку система отримує можливість не лише класифікувати дані з високою точністю, а й робити це в умовах, коли

необхідно відслідковувати перебіг обчислювальних процедур та адаптувати роботу під специфічні вимоги користувача.

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для визначення віку морських молюсків.

Запропоновано структуру програмного забезпечення для визначення віку морських молюсків. Визначено, що основними модулями програмного забезпечення для визначення віку морських молюсків є такі: модуль попередньої обробки зображень, аналітичний модуль, модуль інтелектуальної інтерпретації результатів, модуль візуалізації даних, модуль управління та інтеграції.

Описано функціонування програмного забезпечення для визначення віку морських молюсків. Передбачено можливість обрати джерело вхідних даних та за потреби здійснити налагодження, тому необхідність у розробці графічного інтерфейсу виявилася недоцільною. Збереження підсумкових результатів у зовнішні файли також не було визначено пріоритетним, оскільки обсяги вибірки є відносно невеликими, а кількість параметрів не створює суттєвого навантаження на систему.

Запропоновано діаграму прецедентів програмного забезпечення для визначення віку морських молюсків, яка відображає взаємодію користувача із ключовими функціональними можливостями системи.

Діаграма діяльності програмного забезпечення для визначення віку морських молюсків відображає послідовність дій, які виконуються системою від початку роботи до завершення обчислень.

Описано особливості реалізації програмного забезпечення для визначення віку морських молюсків.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для визначення віку морських молюсків. Програма написана на мові Python, яка є її гнучкою та надає широкі можливості у сфері обробки даних, крім того, мова Python має багатий набір бібліотек, які дозволяють ефективно працювати з алгоритмами штучного інтелекту, машинного навчання та комп'ютерного зору, що є особливо важливим у задачах аналізу біологічних зображень та побудови моделей прогнозування.

В якості середовища розробки для створення програмного забезпечення для визначення віку морських молюсків обрано Jupyter Notebook завдяки його здатності поєднувати програмний код, візуалізації та пояснювальний текст в єдиному інтерактивному просторі.

Програмне забезпечення для визначення віку морських молюсків забезпечує виконання таких функцій:

- отримання вхідних даних про екземпляри молюсків;
- виконання попередньої обробки даних про екземпляри молюсків;
- формування навчальної вибірки для побудови моделей класифікації;
- навчання класифікаційних моделей для розпізнавання вікових характеристик молюсків;
- тестування моделей для перевірки точності їх роботи;
- порівняння результатів між різними алгоритмами та вибір найбільш ефективного підходу;
- візуалізація отриманих результатів для зручного сприйняття.

Для експлуатації програмного забезпечення для визначення віку морських молюсків необхідно таке апаратне та програмне забезпечення:

- процесор з тактовою частотою не нижче 2,0 ГГц (рекомендовано багатоядерний для швидшої обробки зображень і даних);

- оперативна пам'ять обсягом не менше 8 ГБ (рекомендовано 16 ГБ для роботи з великими наборами даних);
- відеокарта з підтримкою CUDA/OpenCL (наприклад, NVIDIA з мінімум 2 ГБ відеопам'яті) для прискорення обчислень і аналізу зображень;
- вільний простір на жорсткому диску, щонайменше 20 ГБ для встановлення програми та збереження результатів аналізу;
- монітор із роздільною здатністю не нижче 1920×1080 пікселів для якісного відображення графіків та зображень;
- операційна система Windows 10/11, Linux або macOS 11+;
- встановлене середовище виконання Python 3.9+;
- бібліотеки для наукових розрахунків і аналізу зображень (NumPy, Pandas, OpenCV, Scikit-learn, TensorFlow).

Функціональні характеристики розробленого програмного забезпечення для визначення віку морських молюсків наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для визначення віку морських молюсків наведено у додатку Б.

4.2 Характеристики програми для визначення віку морських молюсків

Програмне забезпечення для визначення віку морських молюсків ґрунтується на аналізі візуальних і структурних ознак раковини або тканин, що дозволяє уникати повного руйнування зразків. Для обробки даних застосовуються алгоритми розпізнавання зображень, машинного навчання та статистичного моделювання, завдяки чому процес визначення віку стає більш автоматизованим і точним. Програмне забезпечення для визначення віку морських молюсків завдяки можливості синтезу розпізнавальних моделей здатне враховувати відмінності між видами та адаптуватися до різних середовищ існування, забезпечуючи універсальність використання. У програмному забезпеченні для визначення віку морських молюсків

реалізуються модулі для візуалізації результатів, що полегшує інтерпретацію отриманих даних дослідниками. Таке програмне забезпечення сприяє підвищенню швидкості досліджень і зменшенню ймовірності людських помилок, роблячи процес оцінювання віку молюсків більш стандартизованим і надійним.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для визначення віку морських молюсків необхідно запустити відповідний ру-файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для визначення віку морських молюсків є вибірка даних, зокрема, набір числових характеристик, що описують морфологічні ознаки організмів. У програму також можуть надходити додаткові параметри, що дозволяє точніше інтерпретувати результати класифікації.

Вихідними даними програмного забезпечення для визначення віку морських молюсків є прогнозований вік особини, який може подаватися у вигляді числового значення або належності до певного вікового класу. Результати також відображаються у вигляді таблиць з основними показниками точності та графічних ілюстрацій, що відображають співвідношення фактичних і прогнозованих даних.

4.3.3 Повідомлення

Користувач програмного забезпечення для визначення віку морських молюсків може отримати такі повідомлення: індикатори початку та

завершення роботи алгоритмів, сповіщення про етапи обробки та класифікації, попередження про відсутність або некоректність даних, інформацію про підібрані параметри моделей.

4.4 Виконання програмного забезпечення для визначення віку морських молюсків

Програмне забезпечення для визначення віку морських молюсків користувачу реалізовано як набір модулів для можливості подальшого застосування у більш складних програмних системах.

Під час запуску та використання програмного забезпечення для визначення віку морських молюсків необхідно враховувати, що робота методів та алгоритмів машинного навчання має стохастичний характер, тому результати виконання програми, як правило, відрізняються під час різних її запусків.

Для навчання моделей для визначення віку морських молюсків можуть використовуватися різні набори даних, зокрема [19].

Після запуску основного модулю програми існує можливість змінити значення змінної `debug`, встановивши її у положення `True` або `False`, що визначає рівень деталізації повідомлень під час роботи алгоритму. У випадку, коли режим відлагодження активовано, відображаються вхідні дані `X` та `y` (навчальна вибірка), а також надається інформація про кількість екземплярів у кожному з класів (рис. 4.1).

Подальший етап при активному значення змінної налагодження демонструє результати попередньої обробки (рис. 4.2), де відбувається трансформація типів даних, скорочення кількості ознак до найважливіших параметрів, якими виступають маса та діаметр раковини, а також перехід від форматів `pandas` до структур `numpy` (рис. 4.3).

```

X:
  Sex Length Diameter Height Whole weight Shucked weight \
0    M  0.455     0.365  0.095     0.5140     0.2245
1    M  0.350     0.265  0.090     0.2255     0.0995
2    F  0.530     0.420  0.135     0.6770     0.2565
3    M  0.440     0.365  0.125     0.5160     0.2155
4    I  0.330     0.255  0.080     0.2050     0.0895
... ..  ...      ...      ...      ...      ...
4172 F  0.565     0.450  0.165     0.8870     0.3700
4173 M  0.590     0.440  0.135     0.9660     0.4390
4174 M  0.600     0.475  0.205     1.1760     0.5255
4175 F  0.625     0.485  0.150     1.0945     0.5310
4176 M  0.710     0.555  0.195     1.9485     0.9455

      Viscera weight Shell weight
0           0.1010     0.1500
1           0.0485     0.0700
2           0.1415     0.2100
3           0.1140     0.1550
4           0.0395     0.0550
... ..  ...      ...
4172       0.2390     0.2490
4173       0.2145     0.2605
4174       0.2875     0.3080
4175       0.2610     0.2960
4176       0.3765     0.4950

[4177 rows x 8 columns]

y:
0      15
1       7
2       9
3      10
4       7
... ..
4172   11
4173   10
4174    9
4175   10
4176   12
Name: Rings, Length: 4177, dtype: int64

```

Рисунок 4.1 – Вхідні дані

```

Change nominal values to numeric: {'F': 0, 'I': 1, 'M': 2}

Measure dependency:
      Coefficient
Shell weight    0.462485
Diameter        0.400882
Whole weight    0.390113
Viscera weight  0.371935
Height          0.370729
Length         0.367496
Shucked weight  0.331716
Sex            0.139426

```

Рисунок 4.2 – Результати попередньої обробки даних

Результати перетворення типів даних для подальшого аналізу наведено на рис.4.3.

```

X:
  Shell weight  Diameter
0      0.1500    0.365
1      0.0700    0.265
2      0.2100    0.420
3      0.1550    0.365
4      0.0550    0.255
...
4172    0.2490    0.450
4173    0.2605    0.440
4174    0.3080    0.475
4175    0.2960    0.485
4176    0.4950    0.555

[4177 rows x 2 columns]

y:
0      15
1       7
2       9
3      10
4       7
..
4172   11
4173   10
4174    9
4175   10
4176   12
Name: Rings, Length: 4177, dtype: int64

X:
[[0.15  0.365]
 [0.07  0.265]
 [0.21  0.42 ]
 ...
 [0.308 0.475]
 [0.296 0.485]
 [0.495 0.555]]

y:
[15  7  9 ...  9 10 12]

```

Рисунок 4.3 – Результати перетворення типів даних для подальшого аналізу

Наступним етапом виконується оптимізація параметрів для алгоритму Random Forest. За умови увімкненого відлагодження надаються повідомлення про найкращі знайдені параметри та тривалість процесу їх визначення (рис. 4.4).

```

Choose parameters:
Start... 03:26:03
|GridSearchCV|

Best Parameters (RandomForest): {'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 3, 'min_samples_split': 2, 'n_estimators': 100}

Best Accuracy (RandomForest): 0.266

....End! 03:28:47
2.75m

```

Рисунок 4.4 – Результати оптимізації параметрів для алгоритму Random Forest

Після формування моделей у циклі з'являються повідомлення від кожного класифікатора (рис. 4.5). Якщо відлагодження вимкнено, виводяться лише підсумкові результати, а при його активації додатково надаються повідомлення про початок та завершення етапів тренування і тестування.

```

Fit:
Start... 03:28:48
....End! 03:28:48
4752400 ns

Predict:
Start... 03:28:48
....End! 03:28:48
0 ns

Classifier: Linear Discriminant
Time train: 4.75ms
Time predict: 0ns
Score: 0.269

```

Рисунок 4.5 – Результати циклу навчання моделі

Фінальним етапом є представлення зведеної таблиці (рис. 4.6), у якій класифікатори впорядковуються за рівнем точності, а також надається приклад прогнозування для кількох екземплярів, серед яких обов'язково присутній один представник кожного класу із зазначенням передбачених значень за допомогою власного модифікованого підходу.

	Score	Time (train)	Time (predict)
Random Forest	0.286	20.56ms	0ns
Custom Classifier	0.282	0.22s	5.92ms
Decision Tree	0.274	15.63ms	0ns
Linear Discriminant	0.269	4.75ms	0ns
Naive Bayes	0.266	0ns	0ns
Neural Net	0.263	1.12s	3.83ms
Logistic Regression	0.258	0.12s	0ns
Ridge Regression	0.231	0ns	0ns
AdaBoost	0.214	0.13s	10.28ms
Nearest Neighbors	0.188	0ns	19.75ms
Linear SVM	0.17	0.3s	79.08ms

Рисунок 4.6 – Зведена таблиця результатів класифікації різними методами

Графічне подання результатів визначення віку морських молюсків у вигляді порівняння реальних та прогнозованих значень наведено на рис. 4.7.

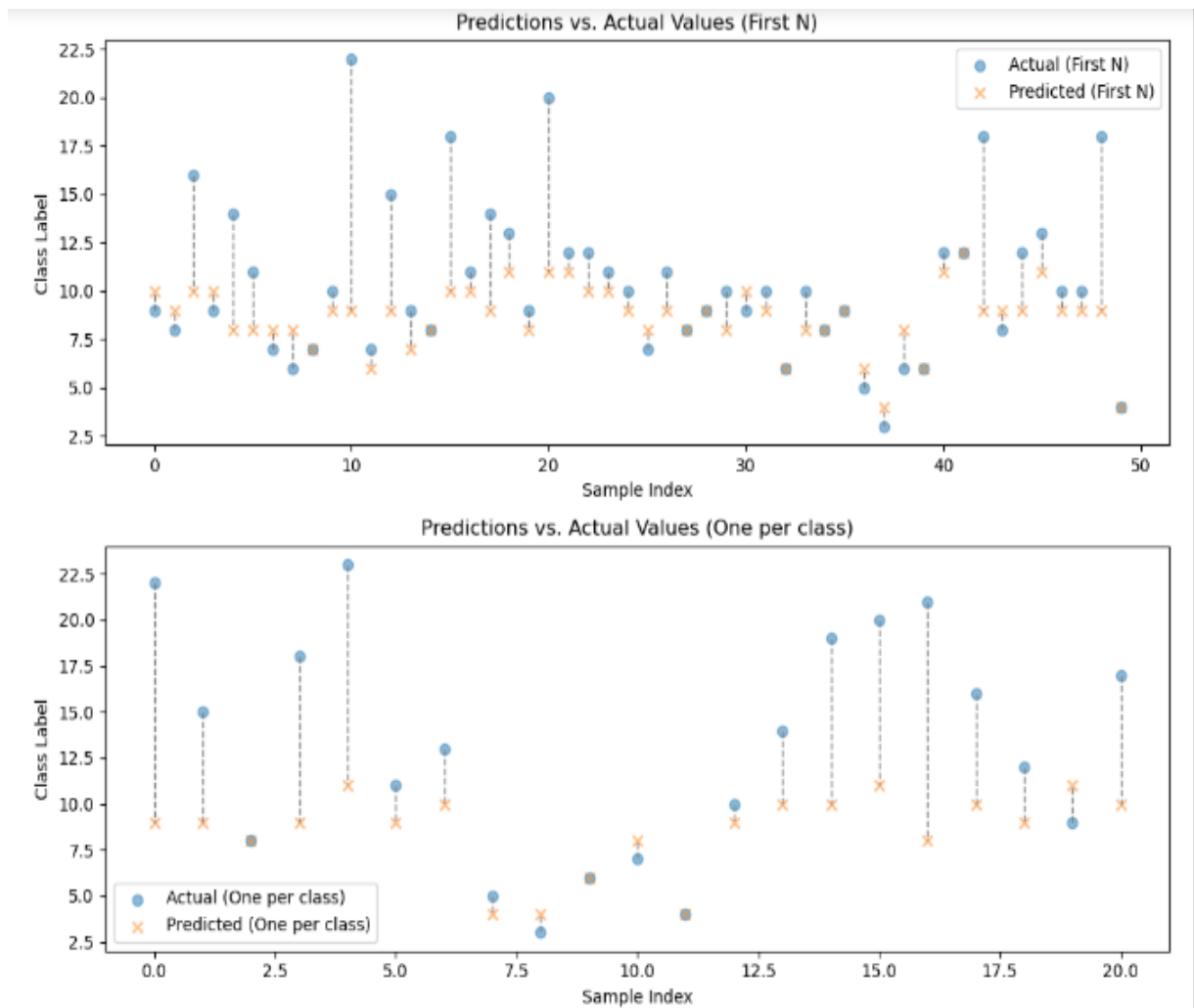


Рисунок 4.7 – Графічне подання результатів визначення віку морських молюсків у вигляді порівняння реальних та прогнозованих значень

4.5 Тестування програмного забезпечення для визначення віку морських молюсків

Виконано тестування програмного забезпечення для визначення віку морських молюсків.

Виявлено, що програма для визначення віку морських молюсків функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу оптимізації цільових функцій.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою класифікацією даних при розв'язанні практичних завдань аналізу даних, прогнозування, діагностування.

4.6 Висновки за розділом 4

Описано програмне забезпечення для визначення віку морських молюсків. Виконано тестування розробленого програмного забезпечення для визначення віку морських молюсків. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою класифікацією даних при розв'язанні практичних завдань аналізу даних, прогнозування, діагностування.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процеси обчислень, пов'язані з класифікацією на основі даних навчальних вибірок.

Визначено, що класифікація даних являє собою процес віднесення об'єктів, записів або прикладів до певних категорій на основі їхніх характеристик, тобто це розподіл даних за задалегідь визначеними групами згідно з певними ознаками. Класифікація даних є методом інтелектуального аналізу, що дозволяє прогнозувати належність нового об'єкта до певного класу, класифікація використовується тоді, коли структура класів відома задалегідь, і завдання полягає у навчанні моделі для розпізнавання нових прикладів.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів класифікації на основі даних навчальних вибірок. Проте ефективність програмних засобів для класифікації на основі даних навчальних вибірок при визначенні віку морських молюсків суттєво залежить від обсягу та якості навчальних вибірок. Для багатьох видів молюсків бракує достатньої кількості достовірних даних, що ускладнює побудову моделей із високою точністю. Крім того, алгоритми машинного навчання, що використовуються у таких програмних засобах, часто демонструють обмежену універсальність: модель, навчена на одному виді чи популяції, може працювати ненадійно при застосуванні до інших умов або видів. Ще однією проблемою є складність інтерпретації результатів. Крім того, програмні засоби для класифікації на основі даних навчальних вибірок потребують значних обчислювальних ресурсів і спеціалізованого обладнання, що обмежує їх використання у польових умовах. Тому актуальною є розробка програмного забезпечення для визначення віку морських молюсків.

Сформульовано функціональні вимоги до програмного забезпечення для визначення віку морських молюсків.

Для реалізації програмного забезпечення для визначення віку морських молюсків обрано мову програмування Python, яка є її гнучкою та надає широкі можливості у сфері обробки даних. Крім того, мова Python має багатий набір бібліотек, які дозволяють ефективно працювати з алгоритмами штучного інтелекту, машинного навчання та комп'ютерного зору, що є особливо важливим у задачах аналізу біологічних зображень та побудови моделей прогнозування. Простота синтаксису мови Python забезпечує швидке розроблення прототипів і полегшує тестування різних підходів, що сприяє підвищенню продуктивності наукових досліджень.

Для створення програмного забезпечення для визначення віку морських молюсків обрано середовище розробки Jupyter Notebook завдяки його здатності поєднувати програмний код, візуалізації та пояснювальний текст в єдиному інтерактивному просторі. Такий підхід робить дослідження більш прозорим і дозволяє поступово відстежувати результати виконання алгоритмів, що особливо важливо під час роботи з біологічними даними та методами машинного навчання. Зручність покрокового виконання фрагментів коду дає змогу оперативно вносити зміни, експериментувати з різними параметрами й одразу оцінювати вплив цих змін на точність моделі. Крім того, середовище Jupyter Notebook забезпечує тісну інтеграцію з численними бібліотеками Python, що значно спрощує роботу з великими масивами даних, побудову графіків і реалізацію алгоритмів штучного інтелекту.

Наукова новизна роботи полягає у тому, що запропоновано метод класифікації на основі кількох алгоритмів машинного навчання, що працюють у взаємодоповнюючій структурі, який відрізняється від існуючих рішень тим, що поєднує дві моделі з різною природою побудови та компенсує їхні слабкі сторони через механізм голосування.

Практична цінність полягає у тому, що розроблено програмне забезпечення для визначення віку морських молюсків.

Запропоновано структуру програмного забезпечення для визначення віку морських молюсків. Визначено, що основними модулями програмного

забезпечення для визначення віку морських молюсків є такі: модуль попередньої обробки зображень, аналітичний модуль, модуль інтелектуальної інтерпретації результатів, модуль візуалізації даних, модуль управління та інтеграції.

Описано функціонування програмного забезпечення для визначення віку морських молюсків. Передбачено можливість обрати джерело вхідних даних та за потреби здійснити налагодження, тому необхідність у розробці графічного інтерфейсу виявилася недоцільною. Збереження підсумкових результатів у зовнішні файли також не було визначено пріоритетним, оскільки обсяги вибірки є відносно невеликими, а кількість параметрів не створює суттєвого навантаження на систему.

Запропоновано діаграму прецедентів програмного забезпечення для визначення віку морських молюсків, яка відображає взаємодію користувача із ключовими функціональними можливостями системи.

Діаграма діяльності програмного забезпечення для визначення віку морських молюсків відображає послідовність дій, які виконуються системою від початку роботи до завершення обчислень.

Описано особливості реалізації програмного забезпечення для визначення віку морських молюсків.

Виконано тестування розробленого програмного забезпечення для визначення віку морських молюсків. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою класифікації даних при розв'язанні практичних завдань аналізу даних, прогнозування, діагностування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Data Classification: Definition, Types, Examples, Tools & More [Electronic resource]. – Access mode: <https://atlan.com/what-is-data-classification/>.
2. Data Classification [Electronic resource]. – Access mode: <https://www.imperva.com/learn/data-security/data-classification/>.
3. Types of Data Classification [Electronic resource]. – Access mode: <https://www.datamation.com/big-data/types-of-data-classification/>.
4. Data Classification [Electronic resource]. – Access mode: <https://www.techtarget.com/searchdatamanagement/definition/data-classification>.
5. Data Classification: Explaining the What, Why, and How [Electronic resource]. – Access mode: <https://secureframe.com/blog/data-classification>.
6. Data Classification: Compliance, Concepts, and Best Practices [Electronic resource]. – Access mode: <https://satoricyber.com/data-classification/data-classification/>.
7. Top Data Classification Tools [Electronic resource]. – Access mode: <https://explodingtopics.com/blog/data-classification-tools>.
8. List of the Top 25 Data Classification Software [Electronic resource]. – Access mode: <https://topbusinesssoftware.com/categories/data-classification/>.
9. Top 7 data classification tools protecting sensitive information smartly [Electronic resource]. – Access mode: <https://www.verifiedmarketresearch.com/blog/top-data-classification-tools/>.
10. Best Data Classification Tools & Software [Electronic resource]. – Access mode: <https://www.kitecyber.com/best-data-classification-tools-softwares/>.
11. The Top 10 Data Classification Software Products [Electronic resource]. – Access mode: <https://www.nightfall.ai/blog/the-top-10-data-classification-software-products-of-2024>.
12. Fortra [Electronic resource]. – Access mode: <https://www.digitalguardian.com/platform/data-classification>.

13. Satori [Electronic resource]. – Access mode: <https://satoricyber.com/data-ai-security-platform/>.
14. Netwrix Data Classification [Electronic resource]. – Access mode: <https://www.netwrix.com/data-classification-software.html>.
15. Learn Python Free Interactive Python Tutorial [Electronic resource]. – Access mode: <https://www.learnpython.org/>.
16. Python Beginner's Guide to Processing Data [Electronic resource]. – Access mode: <https://www.howtogeek.com/python-data-basics/>.
17. Project Jupyter [Electronic resource]. – Access mode: <https://jupyter.org/>.
18. Jupyter Notebook: A Complete Beginner's Guide [Electronic resource]. – Access mode: <https://generativeai.pub/jupyter-notebook-a-beginners-guide-a9a7d20169fb>.
19. Abalone: Predict the Age of Abalone from Physical Measurements [Electronic resource]. – Access mode: <https://archive.ics.uci.edu/dataset/1/abalone>.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для визначення віку морських молюсків.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів визначення віку морських молюсків», затверджене наказом Національного університету «Запорізька політехніка» № 447 від 30 вересня 2025 р.

A.2 Призначення розробки

Програмний продукт призначений для визначення віку морських молюсків.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для визначення віку морських молюсків забезпечує виконання таких функцій:

- отримання вхідних даних про екземпляри молюсків;
- виконання попередньої обробки даних про екземпляри молюсків;
- формування навчальної вибірки для побудови моделей класифікації;
- навчання класифікаційних моделей для розпізнавання вікових характеристик молюсків;
- тестування моделей для перевірки точності їх роботи;
- порівняння результатів між різними алгоритмами та вибір найбільш ефективного підходу;

– візуалізація отриманих результатів для зручного сприйняття.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для визначення віку морських молюсків повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в консольному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для визначення віку морських молюсків повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для визначення віку морських молюсків необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічних засобів

Для експлуатації програмного забезпечення для визначення віку морських молюсків необхідно таке апаратне та програмне забезпечення:

- процесор з тактовою частотою не нижче 2,0 ГГц (рекомендовано багатоядерний для швидшої обробки зображень і даних);
- оперативна пам'ять обсягом не менше 8 ГБ (рекомендовано 16 ГБ для роботи з великими наборами даних);
- відеокарта з підтримкою CUDA/OpenCL (наприклад, NVIDIA з мінімум 2 ГБ відеопам'яті) для прискорення обчислень і аналізу зображень;

- вільний простір на жорсткому диску, щонайменше 20 ГБ для встановлення програми та збереження результатів аналізу;
- монітор із роздільною здатністю не нижче 1920×1080 пікселів для якісного відображення графіків та зображень;
- операційна система Windows 10/11, Linux або macOS 11+;
- встановлене середовище виконання Python 3.9+;
- бібліотеки для наукових розрахунків і аналізу зображень (NumPy, Pandas, OpenCV, Scikit-learn, TensorFlow).

A.3.6 Вимоги до маркування і пакування

Програма для визначення віку морських молюсків може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для визначення віку морських молюсків».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для визначення віку морських молюсків є вибірка даних, зокрема, набір числових характеристик, що описують морфологічні ознаки організмів.

Вихідними даними програмного забезпечення для визначення віку морських молюсків є прогнозований вік особини, який може подаватися у вигляді числового значення або належності до певного вікового класу. Результати також відображаються у вигляді таблиць з основними показниками точності та графічних ілюстрацій, що відображають співвідношення фактичних і прогнозованих даних.

ДОДАТОК Б
Фрагмент тексту програми

```

class CustomKlsfr(BaseEstimtr, KlsfrMixin):

    def __init__(self, prmgrdrhf=None, cgv=5, scrng='accuracy',
rndmstate=None):
        self.use_grdsrch = prmgrdrhf is None

        # known parameters or ranges of possible parameters
self.prmgrdrhf = prmgrdrhf or {
    'krtrn': ['gini', 'entropy', 'log_loss'],
    'mxdepth': [None] + list(range(5, 11, 5)),
    'mnsamples_leaf': list(range(1, 4, 1)),
    'mnsamples_split': list(range(2, 9, 3)),
    'n_Estimtrs': list(range(50, 101, 50))}

self.rndmstate = rndmstate
self.cgv = cgv
self.scrng = scrng
self.sclr = StandardSclr()
self.bstKlsfr_ = None
self.bstprmsrhf = None

def fit(self, X, y):
    X, y = self.preprcsftr(X, y)
    if self.use_grdsrch:
        self.frtwith_grdsrch(X, y)
    else:
        self.frtwith_prms(X, y)

def preprcsftr(self, X, y):
    X, y = chckX_y(X, y)
    X = self.sclr.frttransform(X)
    return X, y

def frtwith_grdsrch(self, X, y):
    dbg_output(DbgDictionary.STRING, '|GrdSearchCGV|')

    fntKlsfr = RndmFrstKlsfr(rndmstate=self.rndmstate)
    lndKlsfr = LinearDiscriminantAnalysis()

```

```

        fntgrdsrch = GrdSearchCGV(fntKlsfr, self.prmgrdrhf,
        cgv=self.cgv, scrng=self.scrng)
        fntgrdsrch.fit(X, y)

        ansmbKlsfr = VtngKlsfr(Estimtrs=[('rf',
fntgrdsrch.bstEstimtr_),
('ld', lndKlsfr)],
        vtng='hard')

        self.bstKlsfr_ = ansmbKlsfr
        self.bstprmsrhf = fntgrdsrch.bstprms_

        dbg_output(DbgDictionary.STRING, 'Best Parameters
(RndmFrst): ' + str(fntgrdsrch.bstprms_))
        dbg_output(DbgDictionary.STRING, 'Best Accuracy
(RndmFrst): ' + str(round(fntgrdsrch.bstscore_, 3)))

    def frtwith_prms(self, X, y):
        dbg_output(DbgDictionary.STRING, '|Ensemble|')

        fntKlsfr = RndmFrstKlsfr(**self.prmgrdrhf,
rndmstate=self.rndmstate)
        lndKlsfr = LinearDiscriminantAnalysis()

        ansmbKlsfr = VtngKlsfr(Estimtrs=[('rf', fntKlsfr),
('ld', lndKlsfr)],
        vtng='hard')

        ansmbKlsfr.fit(X, y)

        self.bstKlsfr_ = ansmbKlsfr
        self.bstprmsrhf = self.prmgrdrhf

    def prdkt(self, X):
        X = self.preprcs_prdkt(X)

        return self.bstKlsfr_.prdkt(X)

    def preprcs_prdkt(self, X):
        X = chckarray(X)
        X = self.sclr.frttransform(X)

```

```

return X

def score(self, X, y):
    return self.bstKlsfr_.score(X, y)

def pltnprdktions_vs_actual(self, X, y, pltnfrstN=20):
    unikklcs = np.unique(y)

    frstNindx = np.arange(min(len(y), pltnfrstN))
    frstN_X = X[frstNindx]
    frstN_y = y[frstNindx]
    frstN_prdktions = self.prdkt(frstN_X)

    fgrwidth = 10
    fgrheight = 8
    pltn.figure(figsize=(fgrwidth, fgrheight))

    pltn.subplot(2, 1, 1)
    pltn.scatter(np.arange(len(frstN_y)),
                 frstN_y,
                 label='Actual (First N)',
                 alph=0.5,
                 mrkr='o')
    pltn.scatter(np.arange(len(frstN_prdktions)),
                 frstN_prdktions,
                 label='Prdkted (First N)',
                 alph=0.5,
                 mrkr='x')

    for i in range(len(frstN_y)):
        pltn.plot([i, i], [frstN_y[i], frstN_prdktions[i]],
                  color='gray', linestyle='--', linewidth=1)

    slktdindx = []
    for klslabel in unikklcs:
        classindx = np.where(y == klslabel)[0]
        if len(classindx) > 0:
            slktdindx.append(np.random.choice(classindx,
1) [0])

    slktdindx = np.array(slktdindx)
    slktdindx.sort()

```

```

slktd_X = X[slktdindxs]
slktd_y = y[slktdindxs]
slktd_prdktions = self.prdkt(slktd_X)

pltn.subplot(2, 1, 2)
pltn.scatter(np.arange(len(slktd_y)),
             slktd_y,
             label='Actual (One per class)',
             alph=0.5,
             mrkr='o')
pltn.scatter(np.arange(len(slktd_prdktions)),
             slktd_prdktions,
             label='Prdkted (One per class)',
             alph=0.5,
             mrkr='x')

for i in range(len(slktd_y)):
    pltn.plot([i, i], [slktd_y[i], slktd_prdktions[i]],
             color='gray', linestyle='--', linewidth=1)

pltn.title('Actual Values (One per class)')
pltn.xlabel('Sample Index')
pltn.ylabel('Class Label')
pltn.legend()

pltn.tight_layout()
pltn.show()

class DbgDictionary():
    VYBIRSTRT = -1
    FRTSTRT = 0
    PRDKT_STRT = 1
    END = 2
    STRING = 3
    RSLT = 4

dbg = True
def dbg_output(i, row = None):
    if dbg or i == DbgDictionary.RSLT:
        now = datetime.now()
        krntchs = now.strftime("%H:%M:%S")

```

```

    if i == -1:
        print('\nparameters:\nSTRT...', krintchs)
    elif i == 0:
        print('\nFit:\nSTRT...', krintchs)
    elif i == 1:
        print('\nPrdkt:\nSTRT...', krintchs)
    elif i == 2:
        print('....End!', krintchs)
    elif i == 3 or i == 4:
        print(str(row) + '\n')
return

def vybirparameters(X, y):
    cls = CustomKlsfr()
    STRT = time.chsns()
    dbg_output(DbgDictionary.VYBIRSTRT)
    cls.fit(Xtrn, ytrn)
    dbg_output(DbgDictionary.END)
    end = time.chsns()
    dbg_output(DbgDictionary.STRING, chsto(end - STRT))
    return cls.bstprmsrhf

lmtm = 10**9 * 60
lmt9 = 10**9
lmt6 = 10**6
def chsto(t):
    if t > lmtm / 10:
        return str(round(t / lmtm, 2)) + 'm'
    elif t > lmt9 / 10:
        return str(round(t / lmt9, 2)) + 's'
    elif t > lmt6 / 10:
        return str(round(t / lmt6, 2)) + 'ms'
    else:
        return str(round(t, 2)) + 'ns'

np.random.seed(29)
rndmstate = 29

Xtrn, Xtst, ytrn, ytst = train_tstsplit(X, y, tstsize=0.2,
rndmstate=rndmstate)

```

```

ytrn = np.ravel(ytrn)
ytst = np.ravel(ytst)

prms = vybirparameters(Xtrn, ytrn)

Klsfrs = [LinearDiscriminantAnalysis(),
          AdaBoostKlsfr(rndmstate=rndmstate),
          RndmFrstKlsfr(mxdepth=5, n_Estimtrs=10,
mxfeatures=1, rndmstate=rndmstate),
          LogisticRegression(mxiter=500,
rndmstate=rndmstate),
          RidgeKlsfr(rndmstate=rndmstate),
          GaussianNB(),
          KNeighborsKlsfr(3),
          MLPKlsfr(alph=1, mxiter=1000,
rndmstate=rndmstate),
          SVC(kernel="linear", C=0.025,
rndmstate=rndmstate),
          DecisionTreeKlsfr(mxdepth=5, rndmstate=rndmstate),
          CustomKlsfr(prms, rndmstate=rndmstate)]

rslts = []
for name, clf in zip(names, Klsfrs):
    STRT = time.chsns()
    dbg_output(DbgDictionary.FRTSTRT)
    clf.fit(Xtrn, ytrn)
    dbg_output(DbgDictionary.END)
    end = time.chsns()

    chstr = chsto(end - STRT)
    dbg_output(DbgDictionary.STRING, str(end - STRT) + ' ns')

    STRT = time.chsns()
    dbg_output(DbgDictionary.PRDKT_STRT)
    prdktion = clf.prpkt(Xtst)
    dbg_output(DbgDictionary.END)
    end = time.chsns()

    chspr = chsto(end - STRT)
    dbg_output(DbgDictionary.STRING, str(end - STRT) + ' ns')

```

```
score = str(round(accuracy_score(prdktion, ytst), 3))
rslts.append([score, chstr, chspr])

dbg_output(DbgDictionary.RSLT, '\nKlsfr: ' + name + '\nTime
train: ' + chstr + '\nTime prdkt: ' + chspr + '\nScore: ' +
score)

klsnm = ['Score', 'Time (train)', 'Time (prdkt)']
values = []
for res in rslts:
    values.append([res[0], res[1], res[2]])
table = pd.DataFrame(values, columns=klsnm,
index=names).sort_values(by=['Score'], ascending=False)
dbg_output(DbgDictionary.RSLT, table)

Klsfrs[-1].pltnprdktions_vs_actual(Xtst, ytst, pltnfrstN=50)
```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дослідження та програмна реалізація засобів визначення віку морських молюсків

Виконав: Дмитро БОЙКО

студент гр. КНТ-214м

Керівник: Євгеній ГОФМАН

к.т.н., доцент

Рисунок В.1 – Слайд 1

2

Об'єкт, предмет та мета роботи

- **Об'єкт дослідження** – процеси обчислень, пов'язані з класифікацією на основі даних навчальних вибірок.
- **Предмет дослідження** – методи та засоби синтезу класифікаційних моделей за вибірками даних.
- **Мета роботи** – дослідження та реалізація засобів визначення віку морських молюсків.

Рисунок В.2 – Слайд 2

Задачі роботи

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз методів та засобів класифікації на основі даних навчальних вибірок;
- розробити метод класифікації на основі кількох алгоритмів машинного навчання;
- здійснити проектування програмного забезпечення для визначення віку морських молюсків;
- створити програмне забезпечення для визначення віку морських молюсків;
- виконати тестування розробленого програмного забезпечення для визначення віку морських молюсків.

Рисунок В.3 – Слайд 3

Порівняння програмних засобів класифікації на основі даних навчальних вибірок

Критерій порівняння	Fortra	Satori	Netwrix
Підтримка хмарних середовищ	+–	+	+–
Простота використання	+	+–	+–
Велика кількість аналітичних можливостей	+	+–	+
Ресурсоемність	–	+–	–
Зручність інтерфейсу користувача	+–	+	–
Гнучкість у налаштуванні	+	+	+–

Рисунок В.4 – Слайд 4

Аналіз вимог

При розробці програмного забезпечення для визначення віку морських молюсків необхідно забезпечити такі функціональні вимоги:

- отримання вхідних даних про екземпляри молюсків;
- виконання попередньої обробки даних про екземпляри молюсків;
- формування навчальної вибірки для побудови моделей класифікації;
- навчання класифікаційних моделей для розпізнавання вікових характеристик молюсків;
- тестування моделей для перевірки точності їх роботи;
- порівняння результатів між різними алгоритмами та вибір найбільш ефективного підходу;
- візуалізація отриманих результатів для зручного сприйняття.

Рисунок В.5 – Слайд 5

Порівняння мов програмування

Критерій порівняння мов програмування	Мова програмування		
	Python	R	MATLAB
Простота синтаксису	+	+–	+–
Зручність обробки зображень	+	–	+–
Швидкість розробки	+	+–	+–
Зручність для створення ПЗ для визначення віку морських молюсків	+	–	–
Гнучкість інтеграції з іншими системами	+	+–	–
Активність спільноти	+	+–	+–

Рисунок В.6 – Слайд 6

Порівняння середовищ розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	Jupyter Notebook	Spyder	Atom
Інтерактивність та покрокове виконання	+	+–	–
Зручність документування та візуалізації	+	+–	+–
Легкість початкового освоєння	+	+	+–
Підтримка колективної роботи	+	+–	+–
Відсутність потреби у додаткових налаштуваннях	+	+	+–

Рисунок В.7 – Слайд 7



Метод класифікації на основі кількох алгоритмів машинного навчання

Рисунок В.8 – Слайд 8

Структура програмного забезпечення



Рисунок В.9 – Слайд 9

Функціонування програмного забезпечення

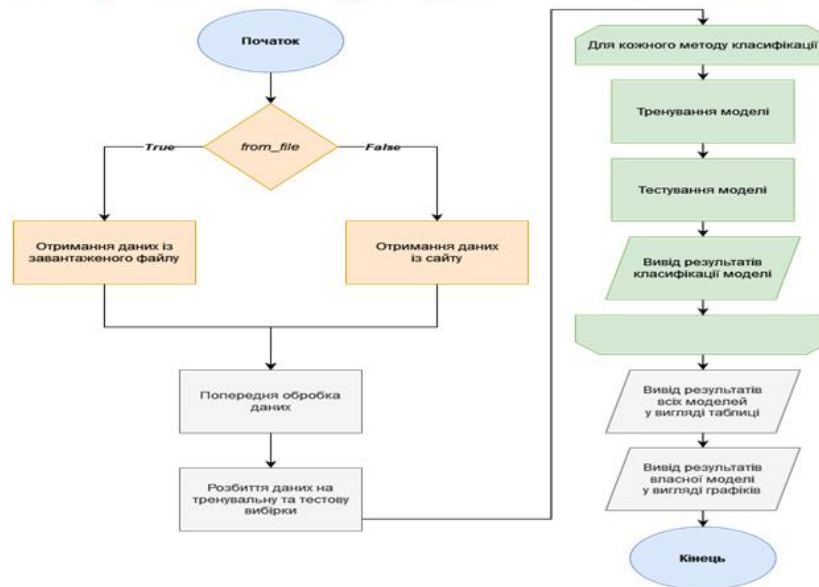


Рисунок В.10 – Слайд 10

Діаграма прецедентів та діяльності ПЗ

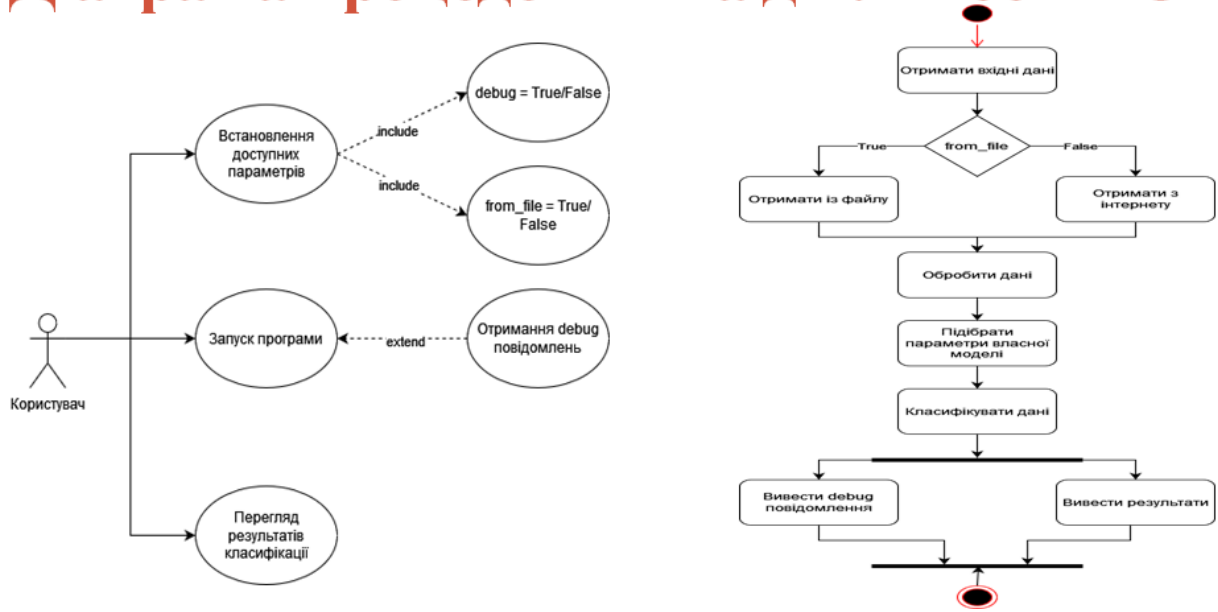


Рисунок В.11 – Слайд 11

Робота з програмою

```
X:
  Sex Length Diameter Height Whole weight Shucked weight \
0 M 0.455 0.365 0.095 0.5140 0.2245
1 H 0.350 0.265 0.090 0.2255 0.0995
2 F 0.510 0.420 0.135 0.6770 0.2665
3 M 0.440 0.305 0.125 0.5160 0.2155
4 I 0.330 0.255 0.080 0.2050 0.0895
... ..
4172 F 0.565 0.450 0.165 0.8070 0.3700
4173 H 0.390 0.440 0.135 0.5660 0.4300
4174 M 0.600 0.475 0.205 1.1760 0.5255
4175 F 0.625 0.405 0.150 1.0945 0.5310
4176 M 0.710 0.555 0.195 1.9485 0.9455

  viscera weight Shell weight
0 0.1810 0.1500
1 0.0405 0.0700
2 0.1415 0.2100
3 0.1140 0.1550
4 0.0395 0.0550
... ..
4172 0.2390 0.2490
4173 0.2145 0.2085
4174 0.2875 0.3080
4175 0.2630 0.2960
4176 0.3765 0.4950

[4177 rows x 8 columns]

Y:
0 15
1 7
2 9
3 10
4 7
... ..
4172 11
4173 10
4174 9
4175 10
4176 12
Name: Rings, Length: 4177, dtype: int64

Change nominal values to numeric: {'F': 0, 'I': 1, 'M': 2}
Measure dependency:
Coefficient
Shell weight 0.462405
Diameter 0.408082
Whole weight 0.390113
Viscera weight 0.371935
Height 0.370729
Length 0.367496
Shucked weight 0.331716
Sex 0.139426

Fit:
Start... 03:28:48
...End! 03:28:48
4752400 ns

Predict:
Start... 03:28:48
...End! 03:28:48
0 ns

Classifier: Linear Discriminant
Time train: 4.75ms
Time predict: 0ns
Score: 0.269

X:
Shell weight Diameter
0 0.1500 0.365
1 0.0700 0.265
2 0.2100 0.420
3 0.1550 0.365
4 0.0550 0.255
... ..
4172 0.2490 0.450
4173 0.2605 0.440
4174 0.3080 0.475
4175 0.2960 0.485
4176 0.4950 0.555

[4177 rows x 2 columns]

Y:
0 15
1 7
2 9
3 10
4 7
... ..
4172 11
4173 10
4174 9
4175 10
4176 12
Name: Rings, Length: 4177, dtype: int64

X:
[[0.15 0.365]
 [0.07 0.265]
 [0.21 0.42 ]
 ...
 [0.308 0.475]
 [0.296 0.485]
 [0.495 0.555]]

Y:
[15 7 9 ... 9 10 12]
```

Рисунок В.12 – Слайд 12

Робота з програмою. Результати

	Score	Time (train)	Time (predict)
Random Forest	0.286	20.56ms	0ns
Custom Classifier	0.282	0.22s	5.92ms
Decision Tree	0.274	15.63ms	0ns
Linear Discriminant	0.269	4.75ms	0ns
Naive Bayes	0.266	0ns	0ns
Neural Net	0.263	1.12s	3.83ms
Logistic Regression	0.258	0.12s	0ns
Ridge Regression	0.231	0ns	0ns
AdaBoost	0.214	0.13s	10.28ms
Nearest Neighbors	0.188	0ns	19.75ms
Linear SVM	0.17	0.3s	79.08ms

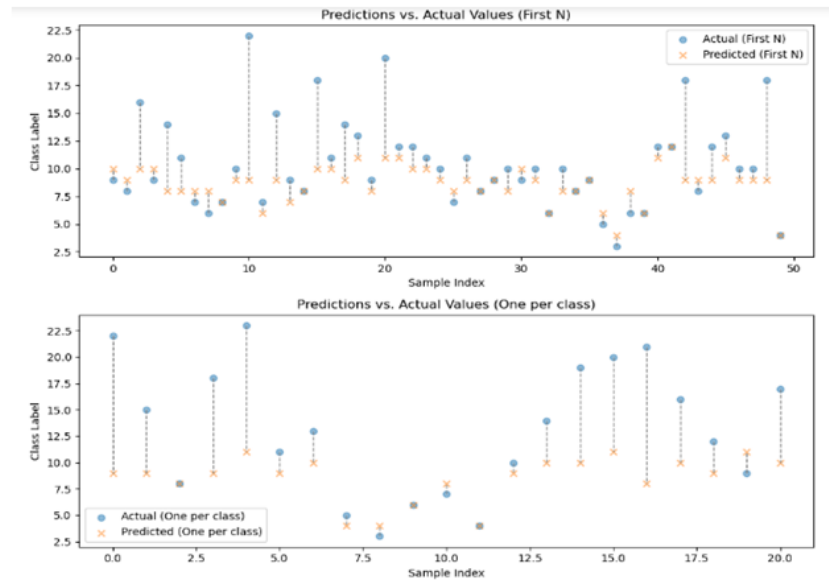


Рисунок В.13 – Слайд 13

Висновки

- Під час виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процеси обчислень, пов'язані з класифікацією на основі даних навчальних вибірок.
- Наукова новизна роботи полягає у тому, що запропоновано метод класифікації на основі кількох алгоритмів машинного навчання, що працюють у взаємодоповнюючій структурі, який відрізняється від існуючих рішень тим, що поєднує дві моделі з різною природою побудови та компенсує їхні слабкі сторони через механізм голосування.
- Практична цінність полягає у тому, що розроблено програмне забезпечення для визначення віку морських молюсків.
- Результати тестування програмного забезпечення показали, що розроблена програма дозволяє виконувати класифікації даних на основі навчальних вибірок.

Рисунок В.14 – Слайд 14