МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____Ю. П. Кондратенко
«____»_____2022 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ БІОЛОГІЧНОЇ КЛАСИФІКАЦІЇ

Спеціальність 122 «Комп'ютерні науки»

122 – МКР – 601.21610207

Студент _____ І.О. Івченко
«__» _____ 2022 р.

Консультант _____ І.О. Калініна
к.т.н., доцент
«__» _____ 2022 р.

Миколаїв – 2022

Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Освітньо-кваліфікаційний рівень **магістр**
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри інтелектуальних інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
«____» _____ 2022 р.

З А В Д А Н Н Я

на магістерську кваліфікаційну роботу

ІВЧЕНКУ Івану Олександровичу

1. Тема магістерської кваліфікаційної роботи «Застосування методів машинного навчання для вирішення задачі біологічної класифікації».

Керівник роботи Калініна Ірина Олександрівна, к.т.н., доцент б.в.з..

Затв. наказом Ректора ЧНУ ім. Петра Могили від «20» жовтня 2021 р. № 288

2. Строк подання студентом роботи 16 лютого 2022 р.

3. Вхідні (початкові) дані до роботи: набір докладних біометричних даних молюсків; методи класифікації. Очікуваний результат: система прогнозування віку молюсків, що заснована на методах машинного навчання, що вирішують задачу класифікації.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

– аналіз сучасного стану вирішення задачі прогнозування віку біологічних істот;

– огляд існуючих методів машинного навчання класифікації;

– оцінка складності побудови системи прогнозування віку молюсків методами машинного навчання з учителем;

– порівняльний аналіз результатів застосування обраних методів класифікації для розв'язання задачі біологічної класифікації.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: Оцінка умов праці та забезпечення безпеки в умовах надзвичайних ситуацій персоналу ТОВ «PINFC» .

7. Консультанти розділів роботи

Розділ	**Прізвище, ініціали та посада консультанта**	**Підпис**
Спеціальна частина з охорони праці	Щербак Ю.Г. к.т.н., доцент	
Методична частина	Калініна І.О. к.т.н., доцент	

Керівник роботи __к.т.н., доцент б.в.з.. Калініна І. О._____
(наук. ступінь, вчене звання, прізвище та ініціали)

_____
(підпис)

Завдання прийнято до виконання ___Івченко І. О._____
(прізвище та ініціали)

_____
(підпис)

Дата видачі завдання «_____» _____ 202_ р.

КАЛЕНДАРНИЙ ПЛАН

Виконання магістерської кваліфікаційної роботи

Тема: Застосування методів машинного навчання для вирішення задачі біологічної класифікації

№	Найменування роботи	Початок	Закінчення	Примітки
1	Визначення керівника і теми МКР. Подання заяви на затвердження теми МКР	01.09.2021	01.10.2021	
2	Отримання завдання на виконання МКР	19.10.2021	22.10.2021	
3	Складання календарного плану на період виконання МКР	23.10.2021	26.10.2021	
4	Огляд літератури за темою дослідження	27.10.2021	10.11.2021	
5	Проходження переддипломної практики, збір та аналіз матеріалів до МКР	22.11.2021	11.12.2021	
6	Аналіз предметної області та розробка технічного завдання. Моделювання результатів	16.12.2021	12.01.2022	
7	Опис фахової частини МКР, зокрема дослідження публікацій щодо методів прогнозування віку молюска, огляд існуючих методів класифікації, на основі машинного навчання для вирішення поставленої задачі, реалізація системи прогнозування з аналізом отриманих результатів	13.01.2022	25.01.2022	
8	Розробка спеціальної частини з охорони праці та методичної частини	26.01.2022	30.01.2022	
9	Попередній захист МКР на засіданні комісії кафедри	31.01.2022	31.01.2022	
10	Корегування роботи за результатами попереднього захисту	01.02.2022	03.02.2022	
11	Остаточне оформлення пояснювальної записки та слайдів доповіді для захисту	04.02.2022	06.02.2022	
12	Подання МКР рецензенту	09.02.2022	10.02.2022	
13	Рецензування МКР	11.02.2022	12.02.2022	
14	Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	16.02.2022	16.02.2022	
15	Захист МКР перед екзаменаційною комісією (ЕК)	21.02.2022	22.02.2022	

Розробив студент _____Івченко І.О._____ _____ _____
(прізвище та ініціали) (*підпис*)

Керівник роботи _____к.т.н., доцент Калініна І. О._____ _____
(*наук. ступінь, вчене звання, прізвище та ініціали)* *(підпис)*

«__» _____ 2021 р.

АНОТАЦІЯ

до магістерської кваліфікаційної роботи
студента групи 601 ЧНУ ім. Петра Могили

Івченка Івана Олександровича

на тему: **“ ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ БІОЛОГІЧНОЇ КЛАСИФІКАЦІЇ ”**

Актуальність даного дослідження полягає у необхідності підвищення точності підрахунку кількості мікроскопічних кілець на мушлі молюсків, яка відповідає віку молюска. Однак, база інших біометричних та суміжних даних у поєднанні з методами машинного навчання можуть вирішити цю проблему.

Об’єктом дослідження є процес класифікації набору біометричних даних молюсків.

Предметом дослідження є методи машинного навчання для вирішення задачі класифікації.

Метою дослідження є засобів та технологій класифікації, та створення інформаційної системи, яка дозволить автоматизувати процес аналізу даних.

В результаті виконання роботи було досліджено три методи машинного навчання класифікації (метод опорних векторів, лінійній дискримінантний аналіз, метод випадкового лісу), проаналізовано вплив якості даних на роботу алгоритмів, визначені основні їх переваги та недоліки, а також розроблено програмне забезпечення, в якому реалізовані відповідні методи.

Дана робота складається із п’яти розділів. Кожен розділ відповідно присвячений: аналізу предметної області, математичним моделям і методам, використаним у магістерській роботі, моделюванню і проєктуванню системи прогнозування віку молюсків, аналізу отриманих результатів, охороні праці, методичній частині магістерської роботи.

Загальний обсяг роботи – __ сторінок. Магістерська кваліфікаційна робота містить __ додаток, __ рисунків, __ таблицю і посилання на __ літературних джерел.

Ключові слова: машинне навчання, методи класифікації, прогнозування віку морського вушка, LDA, Random forest, SVM.

ABSTRACT

to the master's qualification work by the student of the group 601 of Petro Mohyla Black Sea National University

Ivchenko Ivan

"APPLICATION OF MACHINE LEARNING METHODS FOR SOLVING THE PROBLEM OF BIOLOGICAL CLASSIFICATION"

The relevance of this study is the need to increase the speed of counting the number of microscopic rings on the shell of abalone, which corresponds to the age of the abalone. However, other biometric and related databases combined with machine learning techniques may solve this problem.

The object of research is the process of classification of a set of biometric data of mollusks.

The subject of research is the methods of machine learning to solve the problem of classification.

The purpose of the study is the means and technologies of classification, and the creation of a system that will automate the process of data analysis.

As a result of the work, three methods of machine learning classification (reference vectors method, linear discriminant analysis, random forest method) were investigated, the influence of data quality on the operation of algorithms was analyzed, their main advantages and disadvantages were identified, and software was developed methods.

This work consists of five sections. Each section is devoted to: analysis of the subject area, mathematical models and methods used in the master's thesis, modeling and design of the system for predicting the age of mollusks, labor protection, methodological part of the master's thesis.

The overall scope of the work is __ pages. Thesis contains _ application, __ figures, __ tables and __ sources in it.

Key words: machine learning, classification methods, predicting abalone age, LDA, Random forest, SVM.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ШІ – штучний інтелект

ІС – інформаційна система

LDA – Linear Discriminant Analysis

SVM – Support Vector Machine

ПЗ – програмне забезпечення

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему:

«ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ БІОЛОГІЧНОЇ КЛАСИФІКАЦІЇ»

Спеціальність 122 «Комп'ютерні науки»

122 – МКР – 601.21610207

Студент _____ І.О. Івченко
«__» _____ 2022 р.

Консультант _____ І.О. Калініна
к.т.н., доцент
«__» _____ 20___ р.

Миколаїв – 2022

ВСТУП

Актуальність. Попит на використання сучасних методів машинного навчання не є випадковим. Оскільки, задачі які можна вирішити класичними методами ймовірно вже вирішені, може потребуватись свіжий погляд. Зі збільшенням обчислювальної потужності та використанням методів машинного навчання з'явилися нові методи вирішення багатьох проблем, які раніше було складно або неможливо розв'язати.

Щоб запустити процес машинного навчання, для початку необхідно завантажити датасет, на яких алгоритм буде вчитися обробляти запити. Наприклад, це можуть бути дані про масу різних частин тіла молюсків, розміри, стать та мітки, наприклад вік. Після процесу навчання, програма вже сама зможе розпізнавати вік молюсків на нових датасетах без вмісту міток. Процес навчання триває і після виданих прогнозів, чим більше даних ми проаналізували програмою, тим більше точно вона розпізнає потрібні зображення.

Для дослідження методів регресії був обраний датасет з біометричними параметрами молюсків. Програмне середовище для виконання роботи – Rstudio, та мова програмування R, через її переваги над іншими мовами у питаннях класифікації та задачах лінійної регресії.

Це обумовило **мету дослідження**, яка полягає у створенні найбільш точного класифікатору для передбачення віку молюсків, дослідження впливу параметрів налаштування на навчання класифікаторів та результуючу точність.

Відповідно до поставленої мети було сформульовано **завдання дослідження**:

1) Здійснити аналіз сучасного стану вирішення задачі прогнозування віку морського вушка;

2) Зробити огляд існуючих методів класифікації, що використовують засоби машинного навчання;

3) Оцінити складність побудови системи прогнозування віку молюсків методами машинного навчання з учителем;

4) Здійснити порівняльний аналіз результатів застосування обраних методів класифікації для розв'язання задачі біологічної класифікації.

Об'єкт дослідження – процес класифікації набору біометричних даних молюсків.

Предмет дослідження – методи машинного навчання для вирішення задачі класифікації.

Методологічною основою дослідження є загальнонаукові, статистично-аналітичні методи та методи машинного навчання, які дозволили комплексно вивчити предмет та об'єкт дослідження, дослідити нові сфери використання, напрями та шляхи оптимізації доступу до інформації, агрегованої на персональній сторінці користувача з метою здійснення її аналізу та використання.

Практичне значення *отриманих результатів полягає в тому, що* використання розробленої моделі дозволить підвищити точність аналізу агрегованої біологічної інформації про так зване морське вушко та прогнозувати їх вік.

Структура дипломної роботи. Відповідно до мети, завдань і предмета дослідження, дипломна робота містить основну та спеціальну частини. Основна частина роботи складається із вступу, трьох розділів, методичної частини, спеціальної частини з охорони праці, висновку, списку використаних джерел та 1 додатку. Загальний обсяг роботи – __ сторінок, із них основного тексту основної частини – __ сторінок, спеціальної – __ сторінок. Кількість використаних джерел – __.

.

1 ТЕОРЕТИЧНІ ЗАСАДИ ВИКОРИСТАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ

1.1 Розвиток методів машинного навчання

Як наукова діяльність, машинне навчання виросло з пошуку штучного інтелекту. На початку розвитку ШІ як академічної дисципліни, деякі дослідники хотіли зробити так, щоб машини навчалися на даних. Вони намагалися підійти до проблеми різними символічними методами. Хоча деякі з них тоді називалися «нейронними мережами», здебільшого це були перцептрони та інші моделі, які пізніше, як виявилося, були лише перевинайденими узагальненими лінійними моделями статистики. Ймовірнісні підходи також використовувалися, особливо в автоматизованій медичній діагностиці.

Проте, зростаюча популярність заснованому на знаннях підходу, спричинила розподіл між штучним інтелектом та машинним навчанням. Ймовірнісні системи страждали теоретичними та практичними проблемами збору та представлення даних. Вже до 1980 року експертні системи стали домінувати в області штучного інтелекту ШІ, а статистика відповідно стала використовуватись менше. Робота над навчанням на основі знань продовжувалася в рамках ШІ, що призвело до індуктивного логічного програмування, а статистичний напрямок досліджень тепер був витіснений штучним інтелектом, у розпізнаванні образів та пошуку інформації.

Дослідження нейронних мереж були залишені ШІ та інформатикою приблизно в той же час. Їх вивчення було продовжено за межами областей ШІ та комп’ютерних наук, у підході «коннекціонізму», дослідниками з інших дисциплін, включаючи Хопфілда, Румельхарта та Хінтона. Їхній головний успіх прийшовся в середині 1980-х років із відкриттям методу зворотного поширення помилки.

Машинне навчання, реорганізоване як окрема галузь, почало процвітати в

1990-х роках. Сфера змінила свою мету від досягнення штучного інтелекту до вирішення вирішуваних проблем практичного характеру. Додатковий розвиток сфери відбувся від збільшення доступності цифрованої інформації та можливості поширювати її через Інтернет. Машинне навчання та інтелектуальний аналіз даних часто використовують одні й ті ж методи і значно перекриваються. Приблизно їх можна розрізнити наступним чином:

– Машинне навчання зосереджується на передбаченні, заснованому на відомих властивостях, отриманих з даних навчання;

– Інтелектуальний аналіз даних фокусується на виявленні (раніше) невідомих властивостей у даних. Це етап аналізу пошуку знань у базах даних.

Ці дві області багато в чому перекриваються: інтелектуальний аналіз даних використовує багато методів машинного навчання, але часто мають на увазі дещо іншу мету. З іншого боку, машинне навчання також використовує методи аналізу даних як «навчання без нагляду» або як етап попередньої обробки для підвищення точності учнів. Велика частина плутанини між цими двома дослідницькими спільнотами (які часто мають окремі конференції та окремі журнали, ECML PKDD є основним винятком) походить від основних припущень, з якими вони працюють: у машинному навчанні продуктивність зазвичай оцінюється щодо здатності відтворювати відомі знання, тоді як у Knowledge Discovery and Data Mining (KDD) ключовим завданням є виявлення раніше невідомих знань. Оцінений з огляду на відомі знання, неінформований (неконтрольований) метод буде легко перевершити контрольовані методи, тоді як у типовому завданні KDD контрольовані методи не можуть бути використані через недоступність навчальних даних. Машинне навчання також має тісний зв’язок з оптимізацією: багато завдань навчання формулюються як мінімізація деякої функції втрат на навчальному наборі прикладів. Функції втрат виражають невідповідність між передбаченнями моделі, яка навчається, і фактичними екземплярами проблеми (наприклад, у класифікації потрібно призначити мітку екземплярам, а моделі навчаються правильно передбачати

попередньо призначені мітки набору прикладів). Різниця між двома полями виникає з метою узагальнення: в той час як алгоритми оптимізації можуть мінімізувати втрати на навчальному наборі, машинне навчання спрямоване на мінімізацію втрат на невидимих вибірках.

1.2 Огляд біологічної задачі

Морське ушко – це загальна назва морських молюсків. Систематика відносить їх до родини Haliotidae, яка містить лише один рід Haliotis, який колись містив шість підродів. Кількість видів, визнаних у всьому світі, коливається від 30 до 130 з описаними понад 230 таксонами на рівні видів. Найповнішою репрезентативною системою сімейства вважається та, що виділяє 56 дійсних видів з 18 додатковими підвидами [1].

Раковини вушка мають низьку відкриту спіральну структуру і характеризуються кількома відкритими дихальними порами біля зовнішнього краю раковини. Товстий внутрішній шар мушлі складається з перламутру, який у багатьох видів дуже переливається, створюючи цілий ряд сильних, мінливих кольорів, що робить їх мушлі привабливими для людини як матеріал предметів декору, ювелірних виробів, і як джерело колоритного перламутру [3].

М’якоть морського вушка широко вважається бажаною їжею і вживається в сирому або приготованому вигляді в різних культурах усього світу.

Зазвичай вушка варуються за розміром від 20 мм (Haliotis pulcherrima) до 200 мм, в той час як найбільший представник роду Haliotis rufescens становить за розміром 30 см [6].

Шкаралупа вушка опукла, варується від округлої до овальної форми, може бути сильно дугоподібною або дуже сплюснутою. Раковина більшості видів має невеликий плоский шпиль і два-три витки. Останній завиток, відомий як мутовка тіла, має вушну форму, що означає, що раковина нагадує вухо, що дало загальну назву «вушна раковина». Haliotis asinina має дещо іншу форму, так як він більш витягнутий і розтягнутий. Незвичайним є й панцир Haliotis

cracherodii cracherodii, який має яйцеподібну форму, недірчастий, має виступаючий шпиль, колючі ребра [4].

Мантійна щілина в раковині вражає борозенку в раковині, в якій розташований ряд отворів, характерних для роду. Ці отвори є дихальними отворами для випуску води із зябер і для випуску сперми та яйцеклітин у товщу води. Вони утворюють так званий селенізон, який утворюється в міру зростання оболонки. Ця серія отворів, кількість яких варується від 8 до 18, знаходиться біля переднього краю. Загалом для дихання відкрито лише декілька з них. Старі отвори поступово закриваються в міру зростання оболонки і формування нових. Кожен вид має типову кількість відкритих отворів, від чотирьох до десяти.

Зовнішній вигляд раковини смугастий і матовий. Колір раковини дуже варується від виду до виду, що може частково прогнозувати раціон тварини. Переливчастий перламутр, який вистилає внутрішню частину раковини (рис. 1.1), варіюється за кольором від сріблясто-білого, до рожевого, від червоного і зелено-червоного до насиченого синього, від зеленого до фіолетового [2].

Рисунок 1.1 – Внутрішня частина раковини

Тварина має фімбровані частки голови та бічні фімбровані частини. Радула має невеликі серединні зубці, а бічні поодинокі і балчасті. У них близько 70 унцінів із зубчастими гачками, перші чотири дуже великі. Округла стопа дуже велика в порівнянні з більшістю молюсків. М'яке тіло обвивається навколо колумелярного м'яза, і його вставка, замість того, щоб бути на колумелі, знаходиться на середині внутрішньої стінки раковини. Зябра симетричні, добре розвинені [7].

Ці молюски міцно чіпляються своєю широкою мускулистою ногою за скелясті поверхні на глибині субліторалі, хоча деякі види, такі як Haliotis cracherodii, раніше були поширені в припливній зоні. Вушка досягають зрілості при відносно невеликих розмірах. Плодючість їх висока і збільшується разом з розміром, відкладаючи від 10 000 до 11 мільйонів яєць за раз. Сперматозоїди ниткоподібні і загострені на одному кінці, а передній кінець являє собою округлу головку [7].

Дорослі особини більше не допомагають личинкам, і вони описуються як лецитотрофні. Дорослі особини є травоїдними і харчуються макро-водоростями використовуючи радулу, віддаючи перевагу червоним або бурим водоростям.

Сімейство галіотидних має поширення по всьому світу, уздовж прибережних вод усіх континентів, крім тихоокеанського узбережжя Південної Америки, Атлантичного узбережжя Північної Америки, Арктики та Антарктиди. Більшість видів морського вушка зустрічаються в холодних водах, наприклад, біля берегів Нової Зеландії, Південної Африки, Австралії, Західної Північної Америки та Японії [8].

Панцир вушка надзвичайно міцний і виготовлений з мікроскопічних плиток з карбонату кальцію, складених, як цегла. Між шарами плитки знаходиться клейка білкова речовина. Коли раковина вушка вражена, плитки ковзають, а не розбиваються, а білок розтягується, щоб поглинути енергію удару. Матеріалознавці всього світу вивчають цю плиткову структуру, щоб

зрозуміти як зробити керамічні вироби більш міцними, такі як бронежилет. Пил, що утворюється при подрібненні та різанні раковини вушка, небезпечний; повинні бути вжиті відповідні запобіжні заходи для захисту людей від вдихання цих частинок [9].

Морські вушки довгий час були цінним джерелом їжі для людей у всіх районах світу, де є багато видів. М'ясо цього молюска вважається делікатесом у деяких частинах Латинської Америки, Франції, Нової Зеландії, Східної та Південно-Східної Азії. У регіоні Великого Китаю вушка широко відомі як бао ю, а іноді є частиною китайського банкету. Подібно до супу з плавників акули або супу з пташиного гнізда, вушка вважається предметом розкоші і традиційно використовується для особливих випадків, таких як весілля та інші урочистості. Однак доступність вушка, вирощеного в комерційних цілях, дозволила ширше вживати цей колись рідкісний делікатес [10].

Морське вушко було визначено як один із багатьох класів організмів, яким загрожує вимирання через надмірний вилов риби та підкислення океанів від антропогенного вуглекислого газу, оскільки знижений pH руйнує їх панцирі. Передбачається, що вушко вимирають у дикій природі протягом 200 років при нинішніх темпах виробництва вуглекислого газу. В даний час біле, рожеве та зелене вушка входить до списку видів, які перебувають під загрозою зникнення, а можливі місця для відновлення їх популяції знаходяться у процесі розробки. Була також запропонована можливість вирощування вушних вушок для реінтродукції в дику природу, причому ці вушки повинні мати спеціальні мітки для відстеження популяції [11].

Зазвичай прогнозування віку вушка проводиться за допомогою фізичних вимірювань. Вік морського вушка визначають шляхом розрізання шкаралупи через конус, фарбування його та підрахунку кількості кілець за допомогою мікроскопа - нудне і трудомістке завдання.

1.3 Аналіз досліджень та публікацій

Було проведено дослідження Runze Guo et al 2021 J. Phys.: Conf. Ser. 1744 042181, де використовували методи мульти-лінійної регрессії (рис 1.1). Крім того, у цій статті спробували використати всі відомі біологічні атрибути. Інші вимірювання, використовували для прогнозування віку за допомогою штучної нейронної мережі (ANN).

У порівнянні з попередніми моделями, де не була враховано висота, оскільки вона була єдиною у взаємодії діаграми, яка зменшувалася зі збільшенням кілець. Результати показують, що, хоча відхилення удосконалено в аналізі залишків, змінна довжина все ще не відповідає умові 0,1 у аналіз значущості. Результати такі:

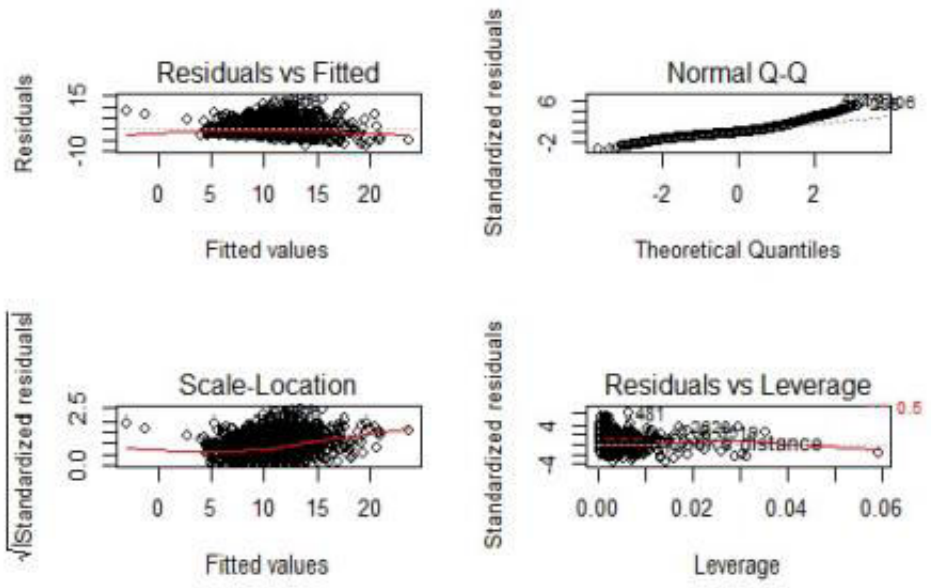

Рисунок 1.2 – QQ-plot за результатами лінійної регресії

У іншому дослідженні (A New Method of Measuring the Age of Abalone Based on Data Visualization Analysis) була представлена модель штучної нейронної мережі (рис. 1.2) для прогнозування віку морського вушка за фізичними вимірюваннями за допомогою функцій, отриманих із репозиторію машинного навчання UCI. У моделі використовувався алгоритм зворотного поширення прямої трансляції для навчання запропонованої моделі ШНС за

допомогою інструменту JNN. Фактори для моделі були отримані з набору даних, який представляє особливості вушка. Модель пройшла тестування, і показник точності склав 92,22%. Це дослідження показало, що штучна нейронна мережа здатна точно визначити вік морського вушка за фізичними вимірюваннями.

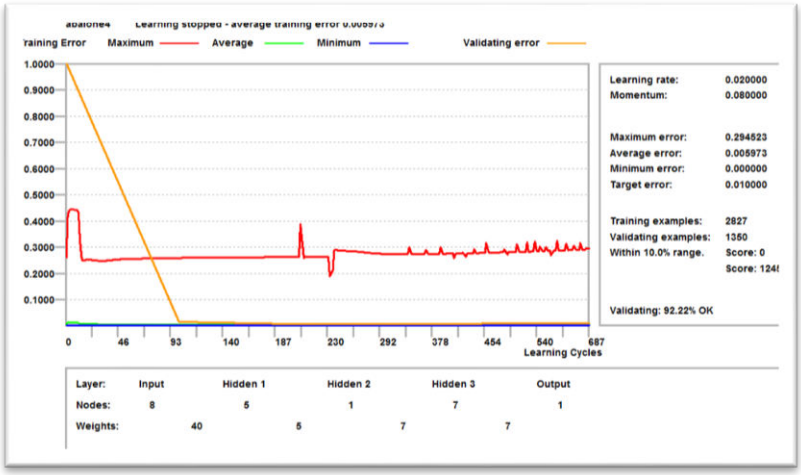

Рисунок 1.3 – Приклад тренування штучної нейронної мережі

Отже, можна зробити висновок, що дослідження проводились, але не всі методи було розглянуто.

Висновки до розділу 1

За результатами аналізу матеріалів, було прийнято рішення дослідити задачу прогнозування віку молюска на основі його біологічних властивостей. Оскільки вік морського вушка прямо залежить від кількості кілець на його панцирі, то задачу прогнозування віку можна представити у вигляді прогнозування кількості кілець на панцирі в цілих числах, а отже вирішувати задачу класифікації, а не лінійної регресії.

Щоб запустити процес машинного навчання, для початку необхідно завантажити датасет, на яких алгоритм буде вчитися обробляти запити. Наприклад, це можуть бути дані про масу різних частин тіла молюсків, розміри, стать та мітки, наприклад вік. Після процесу навчання, програма вже сама зможе розпізнавати вік молюсків на нових датасетах без вмісту міток. Процес

навчання триває і після виданих прогнозів, чим більше даних ми проаналізували програмою, тим більше точно вона розпізнає потрібні зображення.

2 МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ КЛАССИФІКАЦІЇ

2.1 Аналіз існуючих інтелектуальних методів

Протягом останніх років, інтелектуальні алгоритми все частіше використовуються компаніями для вирішення комерційних та наукових задач. Звісно від задачі залежить, який алгоритм буде використовуватись. Кожний з них потребує зусиль з його налаштування та навчання.

Усі методи поділяються на три основних групи:

- Навчання з вчителем;
- Навчання без вчителя;
- Навчання з підкріпленням.

У першому випадку машинне навчання відбувається за допомогою «наставника» - алгоритму, що заздалегідь відмічає потрібні дані, та конкретні приклади. Так, відбувається навчання з розрізнення даних, що може вирішувати задачі регресії та класифікації. З учителем машина вчиться набагато краще й швидше, тому для вирішення практичних завдань такі алгоритми використовують частіше.

Навчання без вчителя вимагає серед різних даних знайти закономірності. Їх поки що використовують рідше, наприклад як, методи аналізу та підготовки даних, тому що дійсно добре розрізнені дані зібрані в реальних умовах є рідкістю. Іноді, замість таких алгоритмів використовують людей з дешевою робочою силою для ручного сортування даних. До таких алгоритмів належать задачі кластеризації, зменшення розмірності і пошуку правил.

Навчання з підкріпленням менше схоже на попередні види, бо нагадує швидше той штучний інтелект, яким його уявляли письменники-фантасти. Такі алгоритми не використовують в задачах, де потрібно проаналізувати дані, а там, де потрібно вирішити задачу в реальному середовищі.

Середовищем може бути будь-що: як обстановка на дорозі, ігри, інші прояви реального життя. Знання усіх тонкощів усього світу для вирішення

задач не обов'язкове, оскільки, завдання таких методів – не розрахувати всі можливі варіації кінцевого результату, а мінімізувати помилки або максимізувати вигоду. Навчання з підкріпленням дуже схоже на реальне навчання людей – машину карають за помилки і заохочують за правильні вчинки.

Не варто забувати, що існують також різні типи задач машинного навчання, серед яких виділяють такі, як:

- Класифікація;
- Регресія;
- Кластеризація;
- Прогнозування;
- Зменшення розмірності;
- Виявлення аномалії (викидів);
- Пошук правил.

Для вирішення задачі класифікації, необхідно розділити об'єкти відповідно до зазначених заздалегідь класів, наприклад розділити щось за кольорами, жанром, біологічним видом, тощо.

Щоб класифікація спрацювала, потрібно мати розмічені дані з категоріями і ознаками, на яких машина буде навчатися диференціювати дані за класами. Залежно від певних ознак алгоритм визначає, до якого з класів можна віднести об'єкт. Деякі з ознак, можуть бути не потрібними, а деякі будуть впливати на результат найбільше.

Найпростішим завданням класифікації є бінарна класифікація. Тут потрібно розподілити об’єкти лише між двома класами. У багатокласовій класифікації кількість класів може досягати декількох тисяч, і рішення стає значно складнішим. Не виключено існування класів, що перетинаються. Тоді об'єкт може одночасно належати до декількох класів. Наостанок, існують нечіткі класи – коли належність до того чи іншого класу визначається ступенем від 0 до 1.

2.2 Linear Discriminant Analysis

Методи зменшення розмірності є важливими в багатьох програмах, пов’язаних з машинним навчанням, інтелектуальним аналізом даних, біоінформатикою та пошуком інформації.

Основна мета методів зменшення розмірності полягає в тому, щоб зменшити розміри шляхом видалення зайвих і залежних об’єктів шляхом перетворення об’єктів із простору вищої розмірності до нижчої. Існує два основних підходи до методів зменшення розмірності, а саме: неконтрольований і контрольований. При неконтрольованому підході немає потреби позначати класи даних. А у контрольованому підході методи зменшення розмірності враховують мітки класів.

Існує багато неконтрольованих методів зменшення розмірності, таких як незалежний компонентний аналіз (ICA) і факторізація невід’ємної матриці (NMF), але найвідомішою технікою цього підходу є лінійний дискримінантний аналіз (LDA). Ця категорія методів зменшення розмірності використовується в біометрії, біоінформатиці та хімії.

Техніка LDA розроблена для перетворення ознак у простір нижнього виміру, який максимізує відношення міжкласової дисперсії до дисперсії всередині класу (рис. 2.1), гарантуючи тим самим максимальну роздільність класів. Існує два типи техніки LDA для роботи з класами: залежна від класу та незалежна від класу. У залежному від класу LDA для кожного класу розраховується один окремий простір нижньої вимірності, щоб проєктувати його дані, тоді як у незалежній від класу LDA кожен клас буде розглядатися як окремий клас проти інших класів. У цьому типі є лише один простір нижнього виміру для всіх класів, щоб проєктувати на нього свої дані.

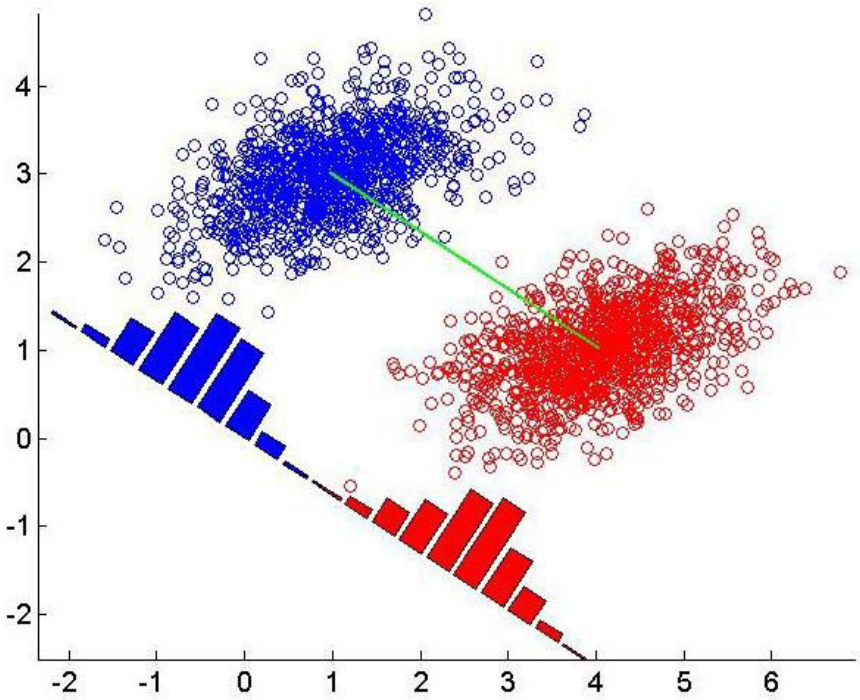

Рис. 2.1. Демонстрація принципу роботи методу LDA

Хоча методика LDA вважається найбільш часто використовуваною технікою скорочення даних, вона страждає від ряду проблем. У першій задачі LDA не вдається знайти нижній розмірний простір, якщо розміри набагато вищі за кількість вибірок у матриці даних. Таким чином, матриця всередині класу стає сингулярною, що відоме як *проблема малої вибірки*. Існують різні підходи, які пропонують вирішити цю проблему, але в актуальній задачі ця проблема не виникає, через велику кількість навчаючих даних.

Перший підхід полягає у видаленні нульового простору всередині класової матриці. Другий підхід використовував проміжний підпростір для перетворення матриці всередині класу в матрицю повного рангу; таким чином, його можна перевернути. Третій підхід, добре відоме рішення, полягає у використанні методу регуляризації для вирішення сингулярних лінійних систем.

У другій задачі, проблемі лінійності, якщо різні класи нелінійно роздільні, LDA не може розрізняти ці класи.

У випадку, коли існує більше двох класів, аналіз, використаний для виведення дискримінанта Фішера, можна розширити, щоб знайти підпростір, який, здається, містить усю мінливість класу.

Ці власні вектори в основному використовуються для зменшення ознак, як у PCA. Власні вектори, що відповідають меншим власним значенням, мають тенденцію бути дуже чутливими до точного вибору навчальних даних, і часто необхідно використовувати регуляризацію, як описано в наступному розділі. Якщо потрібна класифікація, замість зменшення розмірів існує ряд альтернативних методів. Наприклад, класи можуть бути розділені, а стандартний дискримінант Фішера або LDA використовуватимуться для класифікації кожного розділу. Типовим прикладом цього є «один проти решти», коли бали з одного класу поміщаються в одну групу, а все інше – в іншу, а потім застосовується LDA. Це призведе до класифікаторів C, результати яких об’єднані. Іншим поширеним методом є попарна класифікація, коли для кожної пари класів створюється новий класифікатор (загалом дає $C(C-1)/2$ класифікаторів), а окремі класифікатори об’єднуються для отримання остаточної класифікації.

На практиці середні класи та коваріації не відомі. Однак їх можна оцінити з навчального набору. Замість точного значення у наведених вище рівняннях можна використовувати або оцінку максимальної правдоподібності, або максимальну апостеріорну оцінку. Хоча оцінки коваріації можна вважати оптимальними в певному сенсі, це не означає, що результуючий дискримінант, отриманий підстановкою цих значень, є оптимальним у будь-якому сенсі, навіть якщо припущення про нормально розподілені класи є правильним.

Інша складність у застосуванні дискримінанта LDA та Фішера до реальних даних виникає, коли кількість вимірювань кожного зразка перевищує кількість зразків у кожному класі. У цьому випадку коваріаційні оцінки не мають повного рангу, і тому не можуть бути інвертовані. Існує кілька способів боротьби з цим.

Одним з них є використання псевдоінверсної замість звичайної оберненої матриці у наведених вище формулах. Однак кращої чисельної стабільності можна досягти, спочатку спроєктувавши проблему на підпростір .

Інша стратегія боротьби з невеликим розміром вибірки полягає в тому, щоб використовувати оцінку згортання коваріаційної матриці, яка може бути виражена математично.

Це веде до рамки регуляризованого дискримінантного аналізу або дискримінантного аналізу згортання. Крім того, у багатьох практичних випадках лінійні дискримінанти не підходять. Дискримінант LDA і Фішера можна розширити для використання в нелінійній класифікації за допомогою трюку ядра. Тут оригінальні спостереження ефективно відображаються у нелінійному простірі вищих розмірів. Лінійна класифікація в цьому нелінійному просторі тоді еквівалентна нелінійній класифікації в вихідному просторі.

Найбільш часто використовуваним прикладом цього є дискримінант ядра Фішера. LDA можна узагальнити до множинного дискримінантного аналізу, де стає категоріальною змінною з N можливими станами, а не лише з двома.. Ці прогнози можна знайти, розв’язавши узагальнену задачу на власні значення, де чисельником є коваріаційна матриця, утворена шляхом обробки середніх як вибірок, а знаменник — це спільна коваріаційна матриця.

Таким чином, можна зробити висновок, що метод LDA є класичним методом класифікації, а отже буде цікаво його перевірити на реальних даних.

2.2 SVM

SVM засновані на статистичній теорії навчання і мають на меті визначення розташування кордонів прийняття рішень, які створюють оптимальне розділення класів. У задачі розпізнавання шаблонів з двома класами, де класи лінійно роздільні, SVM вибирають одну лінійну межу рішення, яка залишає найбільший запас між двома класами. Запас визначається

як сума відстаней до гіперплощини від найближчих точок двох класів. Цю проблему максимізації маржі можна вирішити за допомогою стандартних методів оптимізації квадратичного програмування (QP). Точки даних, найближчі до гіперплощини, використовуються для вимірювання запасу. Тому ці точки даних називаються «векторами опори» кількість яких завжди мала. Якщо ці два класи не є лінійно розділеними, SVM намагаються знайти гіперплощину, яка максимізує запас, в той же час мінімізуючи кількість, пропорційну кількості помилок неправильної класифікації. Компроміс між маржею та помилкою неправильної класифікації контролюється позитивним, визначеним користувачем параметром.

SVM також можна розширити для обробки нелінійних поверхонь рішень. У 1992 було запропоновано метод проєктування вхідних даних у просторі об'єктів високої розмірності за допомогою нелінійного відображення та формулювання проблеми лінійної класифікації в цьому просторі ознак. Функції ядра використовуються для зменшення обчислювальних витрат, пов’язаних з великорозмірним простором ознак. SVM спочатку були розроблені для бінарних задач. При роботі з кількома класами потрібен відповідний мультикласовий метод. Такі прийоми, як «один проти одного» та «один проти решти», часто використовуються для вирішення багатокласових.

SVM спочатку були розроблені для класифікації і були розширені для регресії і навчання за перевагами (або рангом). Початкова форма SVM – це двійковий класифікатор, де вихід вивченої функції є позитивним або негативним.

Мультикласова класифікація може бути реалізована шляхом об'єднання кількох бінарних класифікаторів за допомогою методу попарного сполучення Цей розділ пояснює мотивацію та формалізацію SVM як бінарного класифікатора, а також дві ключові властивості – максимізацію маржі та хитрість ядра.

Бінарні SVM — це класифікатори, які розрізняють точки даних двох категорій. Кожен об’єкт даних (або точка даних) представлений n-вимірним вектором. Кожна з цих точок даних належить лише до одного з двох класів. Лінійний класифікатор розділяє їх гіперплощиною. Наприклад, Рис. 2.2 показує дві групи даних і роздільні гіперплощини, які є лініями у двовимірному просторі. Існує багато лінійних класифікаторів, які правильно класифікують (або поділяють) дві групи даних, такі як L1, L2 і L3 (рис. 2.2.).

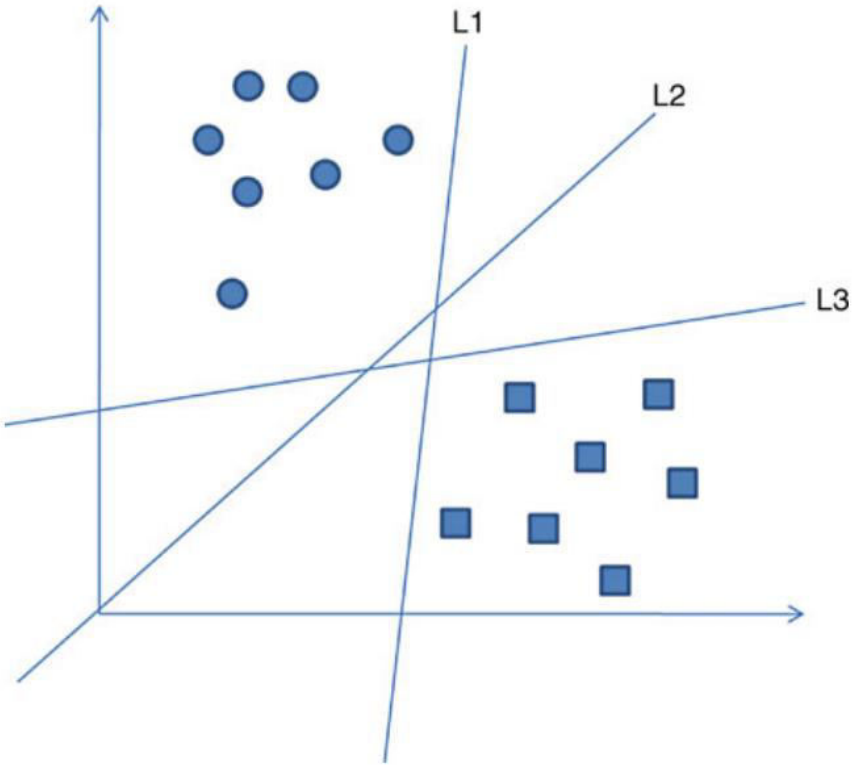

Рис. 2.2. Демонстрація принципу роботи методу SVM

Щоб досягти максимального поділу між двома класами, SVM вибирає гіперплощину, яка має найбільша маржа. Запас — це підсумок найкоротшої відстані від роздільної гіперплощини до найближчої точки даних обох категорій. Така гіперплощина, швидше за все, краще узагальнюватиме, що означає, що гіперплощина може правильно класифікувати «невидимі» або тестові точки даних. SVM виконують відображення з вхідного простору в

простір ознак, щоб підтримати точне формулювання функції відображення, яка може ввести випадок проблеми прокляття розмірності.

Це робить лінійну класифікацію в новому просторі (або просторі ознак) еквівалентною нелінійній класифікації у вихідному просторі (або просторі введення). SVM роблять це шляхом відображення вхідних векторів у більший вимірний простір (або простір ознак), де будується максимальна поділяюча гіперплощина.

2.3 Random Forest

Метод випадкових дерев, що використовують для класифікації, регресії та інших завдань, працює за допомогою побудови численних дерев прийняття рішень під час тренування моделі й продукує моду для класів (класифікацій) або усереднений прогноз (регресія) побудованих дерев.

Класифікатор випадкового лісу складається з комбінації деревних класифікаторів, де кожен класифікатор генерується з використанням випадкового вектора, вибірки незалежно від вхідного вектора, і кожне дерево віддає одиничний голос за найпопулярніший клас для класифікації вхідного вектора.

Класифікатор випадкових лісів, використаний для цього дослідження, складається з використання випадково вибраних ознак або комбінації ознак у кожному вузлі для вирощування дерева. Для кожної вибраної комбінації функції/функції використовувався метод генерування навчального набору даних шляхом випадкового малювання із заміною N прикладів, де N — розмір вихідного навчального набору. Будь-які приклади класифікуються шляхом взяття найпопулярнішого класу з усіх провісників дерев у лісі (рис. 2.3).

Проєктування дерева рішень вимагало вибору міри вибору атрибута та методу обрізання. Існує багато підходів до вибору атрибутів, що використовуються для індукції дерева рішень, і більшість підходів призначають показник якості безпосередньо атрибуту. Найбільш часто використовуваними

мірами вибору атрибутів при індукції дерева рішень є критерій коефіцієнта отримання інформації та індекс Джіні.

Класифікатор випадкових лісів використовує індекс Джіні як міру вибору атрибутів, яка вимірює домішку атрибута по відношенню до класів. Для навчального набору, вибираючи випадковим чином один випадок (піксель) і кажучи, що він належить до деякого класу, індекс Джіні можна записати як: ймовірність того, що обраний випадок належить до класу Сі. Кожного разу дерево вирощується на максимальну глибину на нових навчальних даних за допомогою комбінації функцій. Ці повністю дорослі дерева не обрізають. Це одна з головних переваг класифікатора випадкових лісів перед іншими методами дерева рішень, як-от запропонований Квінланом. Дослідження показують, що вибір методів обрізання, а не заходів вибору атрибутів, впливає на ефективність класифікаторів на основі дерев. Брейман припускає, що при збільшенні кількості дерев помилка узагальнення завжди збігається навіть без обрізки дерева, і переобладнання не є проблемою через сильний закон великих чисел.

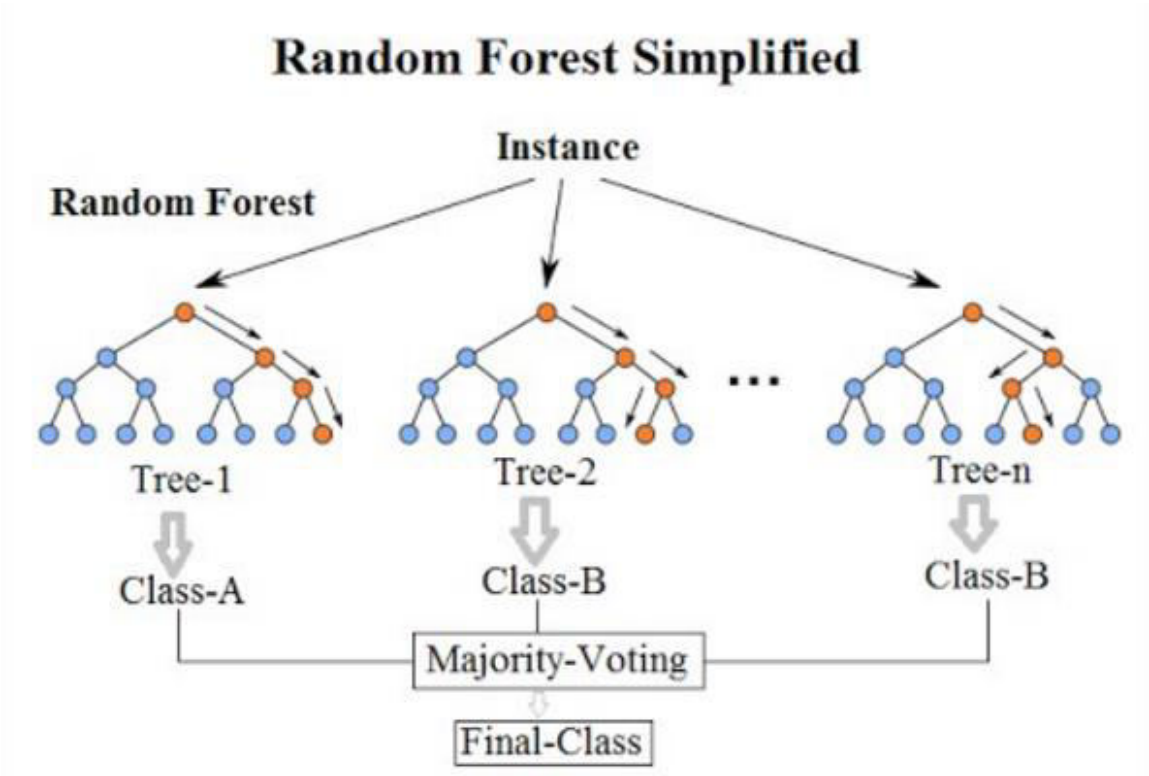

Рис. 2.3. Демонстрація принципу роботи методу Random Forest

Кількість функцій, які використовуються в кожному вузлі для створення дерева, і кількість дерев, які будуть вирощені, є двома визначеними користувачем параметрами, необхідними для створення класифікатора випадкових лісів. У кожному вузлі лише вибрані об’єкти шукаються для найкращого розбиття. Таким чином, класифікатор випадкових лісів складається з N дерев, де N – кількість дерев, які будуть вирощені, яке може бути будь-яким значенням, визначеним користувачем. Щоб класифікувати новий набір даних, кожен випадок наборів даних передається до кожного з N дерев. Ліс обирає клас, який має найбільше з N голосів для цього випадку.

Таким чином, аналіз випадкових лісів ідеально підходить для класифікації, видалення відсутніх значень і, хоча і в меншій мірі, як процедура відбору змінних, але вони не підходять для висновків про зв’язки провісників і залежних змінних.

2.4 Метод k-найближчих сусідів (kNN)

Також було використано метод k-найближчих сусідів. Це непараметричний метод, який використовується для класифікації та регресії. Це одна з найпростіших технік ML. Це модель лінивого навчання з локальним наближенням.

Основна логіка KNN полягає в тому, щоб дослідити околиці деякої точки, припустити, що точки даних подібні до них, і отримати вихідні дані. У KNN ми шукаємо k сусідів і складаємо прогноз.

У випадку класифікації KNN, більшість голосів використовується для k найближчих точок даних, тоді як у регресії KNN середнє значення k найближчих точок даних обчислюється як вихід. Як правило, ми вибираємо непарні числа як k. KNN — це модель відкладеного навчання, де обчислення відбуваються лише під час виконання.

На наведеній діаграмі (рис. 2.4.) жовті та фіолетові точки відповідають класам A та B у даних про т ренування. Червона зірка вказує на дані тесту, які

підлягають класифікації. коли k = 3, ми прогнозуємо клас B як вихід, а коли K=6, ми прогнозуємо клас A як вихід.

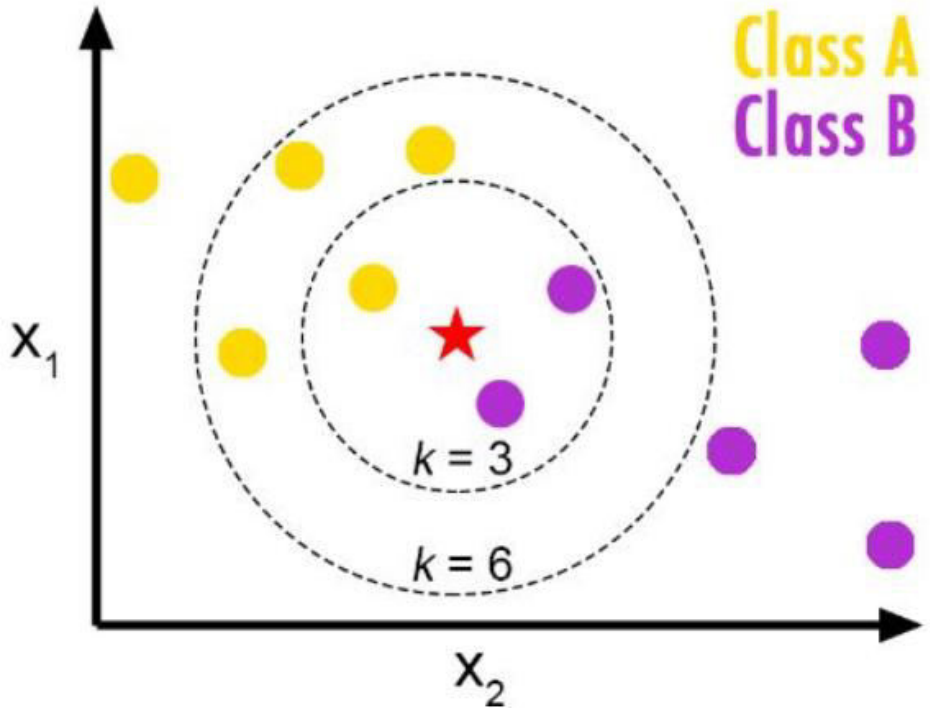

Рис. 2.4. Демонстрація принципу роботи методу Random Forest

До переваг відносять такі властивості. Це легка та проста модель машинного навчання. Має достатньо гіперпараметрів для налаштування моделі. Проте недоліки теж є. KNN слід користуватись з розумом, через те що можуть відбуватись великі витрати комп'ютерної потужності на обчислення під час виконання, якщо розмір вибірки великий. Також, необхідно забезпечити належне масштабування для справедливого ставлення між функціями.

2.5 Дерево рішень

Додатково було використано алгоритм дерева рішень, на випадок, якщо це спрацює. Дерево рішень — це алгоритм на основі дерева, який використовується для вирішення проблем класифікації. Для отримання результату формується перевернуте дерево, яке розгалужується від однорідного розподіленого по ймовірності кореневого вузла до дуже неоднорідних листкових вузлів. Дерева використовуються для залежної змінної з

безперервними значеннями, а дерева класифікації використовуються для залежної змінної з дискретними значеннями.

Дерево рішень є похідним від незалежних змінних, причому кожен вузол має умову над ознакою. Вузли вирішують, до якого вузла слід рухатися далі на основі умови. Після досягнення листового вузла прогнозується вихід.

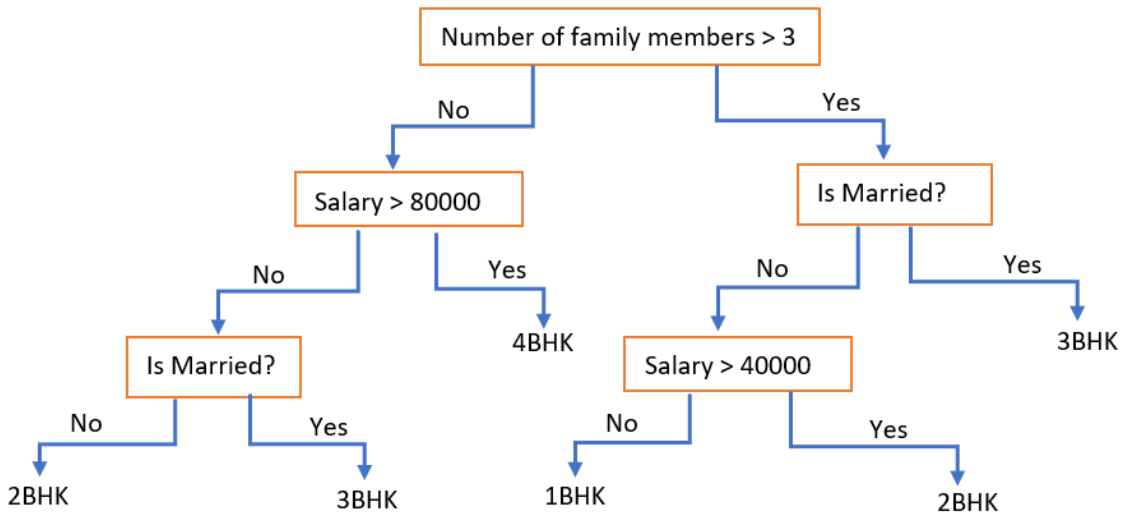

Рис. 2.5. Демонстрація принципу роботи методу Decision Tree

Правильна послідовність умов робить дерево ефективним. ентропія/інформаційний приріст використовуються як критерії вибору умов у вузлах. Для отримання деревоподібної структури використовується рекурсивний жадібний алгоритм.

2.6 Підготовка даних до роботи методів

Системи класифікації відіграють важливу роль в задачах дослідження складних систем та процесів, класифікуючи доступну інформацію на основі деяких критеріїв. В цьому дослідженні необхідно оцінити відносну ефективність деяких добре відомих методів класифікації на складному наборі даних. Досліджено методи класифікації, засновані на статистичних методах і методах штучного інтелекту.

Для вирішення завдання класифікації була розроблена методологія використання методів машинного навчання, яка представлена у вигляді

послідовності етапів на рис. 1.Вирішення завдання класифікації складається з п’яти етапів.

На першому етапі здійснюється збирання даних, аналіз та їх інтерпретація. Завантажується вхідний набір даних, аналізується структура набору даних, визначаються ознаки та їх типи, а у разі потреби відбувається перекодування цих ознак. В результаті такого попереднього опрацювання первинного набору даних маємо підготовлений набір до наступного етапу – розвідувального аналізу даних.

На другому етапі здійснюються процедури розвідувального аналізу даних. Визначається тип описової статистики, виявляються нечислові та відсутні значення, здійснюється відбір ознак для подальшого моделювання, визначається взаємозв’язок між змінними, генерується матриця ознак і масив міток та відбувається нормалізація числових даних. У результаті цих перетворень отримуємо набір даних, які підготовлені для моделювання.

Третій етап – етап моделювання складається з двох частин: підготовка та вибір моделі. При підготовці відбувається поділення основного набору даних на навчальну (тренувальну) та тестову вибірку, визначається функція втрат і створюються набори для крос-перевірки. При виборі моделі перевіряються алгоритми моделювання на тренувальній вибірці та вибирається найкращий за певними критеріями.

Четвертий етап – визначення параметрів ефективності моделі. Ефективність моделей визначається за допомогою матриці неточності, кривої «точність – повнота», F-міри, Каппа, значення робочої характеристики (ROC) та частоти помилок, вимірюваних середньою абсолютною помилкою (MAE) і середньоквадратичною помилкою (RMSE).На п’ятому етапі виконуються процедури з метою підвищення ефективності вибраної моделі класифікації. Залежно від вибраного на третьому етапі методу моделювання засобами покращення якості можуть бути: замість нормування числових значень стандартизація по z-оцінках, дослідження декількох варіантів K (при обранні

KNN-моделі), додавання адаптивного підсилення, використання матриці штрафів або введення правил класифікації. Процедури і методи, перелічені в етапах розробленої методології, безпосередньо зв'язані з процесом візуалізації. За допомогою візуалізації на кожному етапі є можливість швидко прийняти рішення з корегування послідовності дій та повернення на попередні етапи.

Важливо, що кожний з наведених етапів має певні особливості, які враховуються залежно від структури даних початкового набору, особливостей предметної галузі, для якої вирішуються завдання класифікації та засобу його реалізації.

Результати прогнозування були досліджені з точки зору точності, як-от: відгук, F-міра, Каппа, значення робочої характеристики (ROC) і частоти помилок, вимірюваних середньою абсолютною помилкою (MAE) і середньоквадратичною помилкою (RMSE).

Точність – це міра, яка являє собою співвідношення між кількістю правильно прогнозованих значень і загальною кількістю прогнозованих значень (як правильно, так і неправильно). Відгук – це міра повноти, яка являє собою співвідношення між кількістю правильно прогнозованих значень і загальною кількістю релевантних значень. Вони розраховуються з використанням значень істинно позитивної швидкості, непозитивної швидкості і помилково негативної швидкості в результатах прогнозування. Робоча характеристика приймача (ROC) може використовуватися як ще одна метрика, яка також включає частоту справжніх позитивних і помилкових спрацьовувань для оцінки якості вихідних даних класифікатора. Якщо TP, FP і FN позначають справжні спрацьовування, помилкові спрацьовування і помилкові заперечення, то формальне визначення точності і відкликання буде [14]: $\tau = TP\ TP+FP$, □ $= TP\ TP+FN$, де τ – точність; □ – відгук. F-міра – це показник, який об'єднує точність і чутливість в єдиній оцінці, яка являє собою гармонічне середнє значення точності і чутливості. Каппа – це показник, який порівнює спостережувану точність з очікуваною точністю (випадковий шанс). Формальне визначення F-заходи і

Каппи [14]: $Kmeasure = 2 * \tau * \square \ \tau + \square$, $Kappa = (OA - EA)\ (1 - EA)$, Управління розвитком складних систем (46 – 2021) ISSN 2219-5300 179 де OA (observed accuracy) – спостережувана точність; EA (expected accuracy) – очікувана точність. Середня абсолютна помилка (MAE) і середньоквадратична помилка (RMSE) використовуються для розрахунку частоти помилок кожної моделі на основі класифікатора, яка являє точність прогнозування в завданнях машинного навчання [15; 16]. Якщо прогнозовані значення в тестових примірниках дорівнюють $p1$, $p2$, … , pn, а фактичні значення $a1$, $a2$, … , an, для n точок даних MAE і RMSE формально визначені, як показано нижче. Для кожної моделі використано один і той самий набір даних, щоб точно порівняти методи. Для оцінювання розробленої методології було використано десятикратну перехресну перевірку для вихідного набору даних. Метод 5-кратної перехресної перевірки розділяє підготовлений набір на 5 наборів розміром N/5. Після цього кожну модель навчають на чотирьох наборах і тестують на останньому наборі. Згідно з процедурою перехресної перевірки це повторюється п’ять разів. Як результат беруться середні результати прогнозу для кожної моделі. У табл. 2 наведено результати прогнозування кожної моделі на основі класичної оцінки ефективності класифікатора з точки зору швидкості CCI (правильно класифіковані екземпляри), швидкості ICI (неправильно класифіковані екземпляри), швидкості середньої абсолютної помилки (MAE), середньоквадратичної помилки (RMSE) та значення робочої характеристики приймача (ROC). Результати обчислювального експерименту в порівнянні методів прогнозування, наведені в табл. 2, свідчать про те, що правильно класифіковані екземпляри, отримані на основі моделі випадкового лісу (RF) і дерева рішень (DT), становлять 85,37% і 82,66% відповідно, що вище, ніж у інших моделей. Значення ROC, які представляють справжню позитивну частоту порівняно з помилковою позитивною частотою цих моделей класифікації на основі дерева, також дають результати, які кращі за інші моделі на основі класифікатора, наведені в табл. 2. На додаток до цих вимірів, деревоподібні

моделі також дають нижчу частоту помилок з точки зору значень ICI, MAE і RMSE в задачах прогнозування аеродинамічних характеристик матеріалів.

Висновки до розділу 2

У розділі були розглянуті методи машинного навчання, серед яких більш детально було роз

Результати, наведені в цьому розділі, свідчать про те, що класифікатор випадкових лісів може досягти точності класифікації, порівнянної з точністю, досягнутою SVM. Ще одна перевага класифікатора випадкових лісів полягає в тому, що він вимагає встановлення лише двох параметрів, тоді як SVM вимагає ряду параметрів, визначених користувачем.

Класифікатор випадкових лісів може обробляти категоріальні дані, незбалансовані дані, а також дані з відсутніми значеннями, що все ще неможливо з SVM. Цей класифікатор також забезпечує відносну важливість різних ознак під час процесу класифікації, що може бути корисно при виборі ознак. Крім того, класифікатор випадкових лісів забезпечує спосіб виявлення викидів за допомогою аналізу близькості і може використовуватися для навчання без нагляду. Наразі ще тривають дослідження з метою подальшої оцінки ефективності класифікатора випадкових лісів для виявлення викидів, вибору ознак і групування з даними дистанційного зондування.

3 ПРАКТИЧНА ЧАСТИНА

3.1 Підготовка даних

Dataset для машинного навчання – це оброблена і структурована інформація в табличному вигляді. Рядки такої таблиці називаються об'єктами, а стовпці - ознаками. Розрізняють 2 види ознак:

- незалежні змінні - предиктори;
- залежні змінні - цільові ознаки, які обчислюються на основі одного або декількох предикторів.

Ознака (фіча, feature) - це змінна, яка описує окрему характеристику об'єкта. У табличному вигляді вибірки ознаки - це стовпці таблиці, а об'єкти - рядки. Вхідні, незалежні, змінні для моделі машинного навчання називаються предикторами, а вихідні, залежні, - цільовими ознаками. Всі ознаки можуть бути наступних видів:

- бінарні, які приймають два значення, наприклад, {true, false}, {0,1}, {1,1}, { «так», «ні»} і т.д .;
- номінальні (фактори), які мають кінцеве кількість рівнів, наприклад, фактор «день тижня» має іменованих 7 рівнів: понеділок, вівторок і т. Д. Фактори можуть бути впорядкованими та неупорядкованими. Наприклад, фактор «час доби» має 24 рівня і він впорядкований. Фактор «район міста» з 32 рівнями не впорядкований, оскільки всі рівні мають рівну значущість. Якщо фактор впорядкований, це варто явно вказати при його оголошенні.
- кількісні (числові) значення в діапазоні від мінус нескінченності до плюс нескінченності.

Для прогнозування використовуємо набір біометричних даних молюсків. Цей набір містить 4177 екземплярів, 8 атрибутів+1 мітка.

Атрибути і мітки перерахуємо в тому порядку, в якому вони представлені в наборі даних.

1. Sex (numerical): стать;

2. Length (continuous): Найбільший розмір мушлі;

3. Diameter (continuous): перпендикулярний довжині розмір;

4. Height (continuous): висота всього молюска;

5. Whole weight (continuous): загальна маса;

6. Shucked weight (continuous): маса молюска без мушлі;

7. Viscera weight (continuous): маса обезкровлених внутрішньостей;

8. Shell weight (continuous): маса мушлі після висушування;

9. Rings (integer): кількість кілець, що відповідає віку.

Цей набір даних можна отримати зі сховища даних для машинного навчання UCI.

Після завантаження даних необхідно підключити файл «abalone.data» в RStudio виконавши наступну команду.

read.csv("D:/MKR/abalone.data", header=FALSE, sep = ",",as.is = TRUE)

Для перегляду завантажених даних можна переглянути «abalone» в середовищі проєкту.

	V1	V2	V3	V4	V5	V6	V7	V8	V9
1	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.1500	15
2	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.0700	7
3	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.2100	9
4	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.1550	10
5	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.0550	7
6	I	0.425	0.300	0.095	0.3515	0.1410	0.0775	0.1200	8
7	F	0.530	0.415	0.150	0.7775	0.2370	0.1415	0.3300	20
8	F	0.545	0.425	0.125	0.7680	0.2940	0.1495	0.2600	16
9	M	0.475	0.370	0.125	0.5095	0.2165	0.1125	0.1650	9
10	F	0.550	0.440	0.150	0.8945	0.3145	0.1510	0.3200	19
11	F	0.525	0.380	0.140	0.6065	0.1940	0.1475	0.2100	14

Рис. 3.1. Дані на панелі джерел даних

Скористуємось функцією head для відтворення перших рядків даних.

```
> head(abalone)
  V1    V2    V3    V4     V5     V6     V7    V8 V9
1  M 0.455 0.365 0.095 0.5140 0.2245 0.1010 0.150 15
2  M 0.350 0.265 0.090 0.2255 0.0995 0.0485 0.070  7
3  F 0.530 0.420 0.135 0.6770 0.2565 0.1415 0.210  9
4  M 0.440 0.365 0.125 0.5160 0.2155 0.1140 0.155 10
5  I 0.330 0.255 0.080 0.2050 0.0895 0.0395 0.055  7
6  I 0.425 0.300 0.095 0.3515 0.1410 0.0775 0.120  8
```

Рис. 3.2. Перші рядки набору даних

Функція summary дасть базову статистику по кожному колонку даних.

```
> summary(abalone)
 sex          length         diameter          height        whole weight
 F:1307   Min.   :0.075   Min.   :0.0550   Min.   :0.0000   Min.   :0.0020
 I:1342   1st Qu.:0.450   1st Qu.:0.3500   1st Qu.:0.1150   1st Qu.:0.4415
 M:1528   Median :0.545   Median :0.4250   Median :0.1400   Median :0.7995
          Mean   :0.524   Mean   :0.4079   Mean   :0.1395   Mean   :0.8287
          3rd Qu.:0.615   3rd Qu.:0.4800   3rd Qu.:0.1650   3rd Qu.:1.1530
          Max.   :0.815   Max.   :0.6500   Max.   :1.1300   Max.   :2.8255
 shucked weight   viscera weight    shell weight        rings
 Min.   :0.0010   Min.   :0.0005   Min.   :0.0015   Min.   : 1.000
 1st Qu.:0.1860   1st Qu.:0.0935   1st Qu.:0.1300   1st Qu.: 8.000
 Median :0.3360   Median :0.1710   Median :0.2340   Median : 9.000
 Mean   :0.3594   Mean   :0.1806   Mean   :0.2388   Mean   : 9.934
 3rd Qu.:0.5020   3rd Qu.:0.2530   3rd Qu.:0.3290   3rd Qu.:11.000
 Max.   :1.4880   Max.   :0.7600   Max.   :1.0050   Max.   :29.000
```

Рис. 3.3. Результат summary

В цій частині роботи застосуємо наступні методи класифікації для розв'язання поставленої задачі.

Спочатку потрібно побудувати діаграми розподілу. Функція автоматично конвертує логічний та факторний тип даних в числовий, для відображення даних в вигляді діаграм.

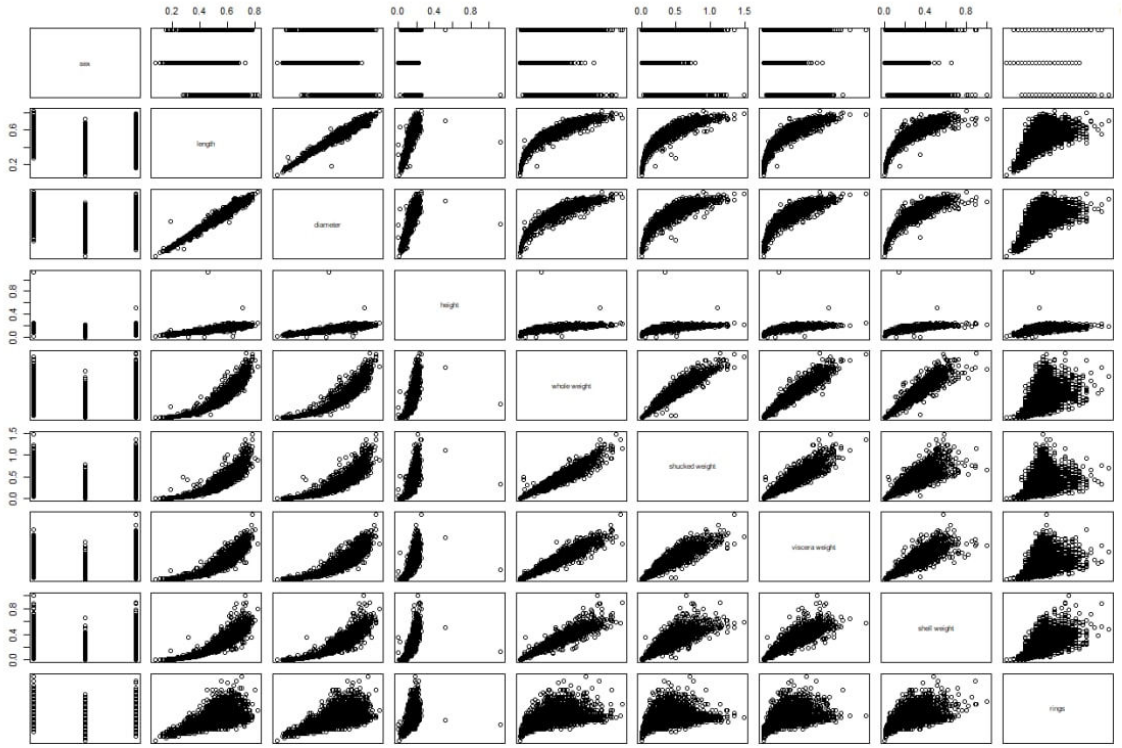

Рис. 3.4. Діаграми розсіяння

Можна побачити, що серед різних атрибутів простежується деяка кореляція. Кореляція – процес знаходження взаємозалежностей між даними. Перевіримо це, за допомогою наступної команди.

```
cor(abalone[, setdiff(names(abalone), c("sex", "var01"))])
```

```
                length  diameter    height whole weight shucked weight viscera weight shell weight     rings
length         1.0000000 0.9868116 0.8275536    0.9252612       0.8979137      0.9030177    0.8977056 0.5567196
diameter       0.9868116 1.0000000 0.8336837    0.9254521       0.8931625      0.8997244    0.9053298 0.5746599
height         0.8275536 0.8336837 1.0000000    0.8192208       0.7749723      0.7983193    0.8173380 0.5574673
whole weight   0.9252612 0.9254521 0.8192208    1.0000000       0.9694055      0.9663751    0.9553554 0.5403897
shucked weight 0.8979137 0.8931625 0.7749723    0.9694055       1.0000000      0.9319613    0.8826171 0.4208837
viscera weight 0.9030177 0.8997244 0.7983193    0.9663751       0.9319613      1.0000000    0.9076563 0.5038192
shell weight   0.8977056 0.9053298 0.8173380    0.9553554       0.8826171      0.9076563    1.0000000 0.6275740
rings          0.5567196 0.5746599 0.5574673    0.5403897       0.4208837      0.5038192    0.6275740 1.0000000
```

Рис. 3.5. Кореляція даних

Дійсно, згідно дослідження параметри мають між собою велику кореляцію. Зв’язки між Ймовірно, це відбувається через те, що маса, розміри та вік нерозривно залежать одне від одного. Для подальшої роботи було обрано атрибут length. Оскільки довжина мушлі має значний вплив на кількість місця, де можуть розташовуватися кільця. Перевіримо чи є структура, в цьому атрибуті.

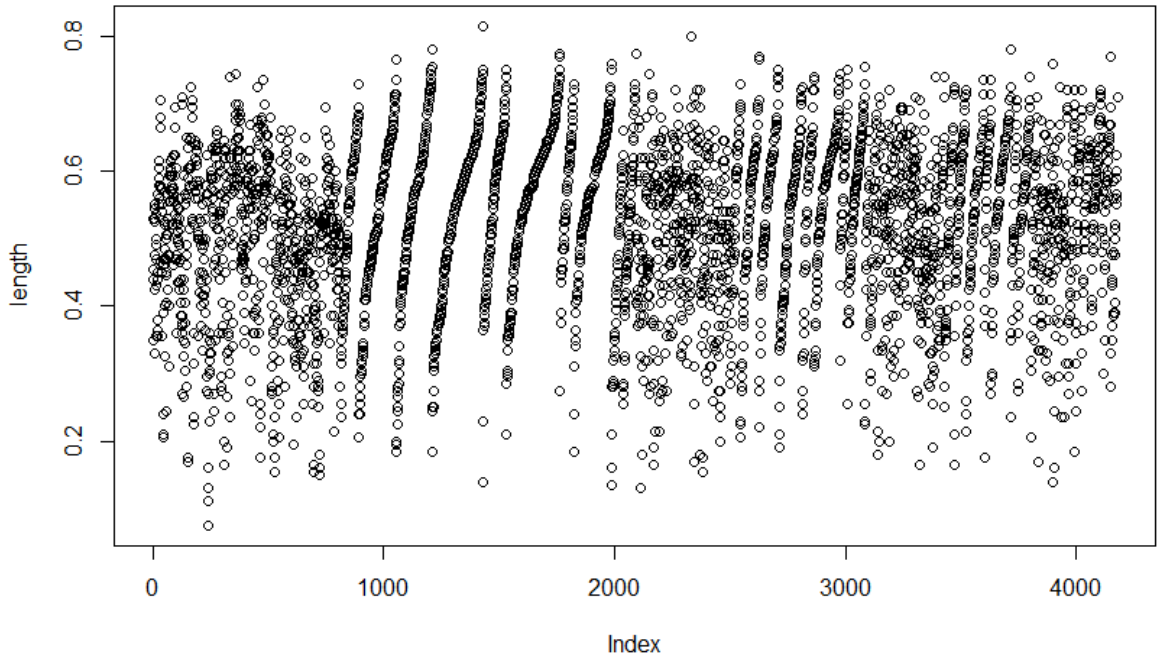

Рис.3.6. Розподілення параметру length

Виявлено, що дані мають залежність від індексу. Це може свідчити про те, що дані сортувались та оброблювались. Перевіримо скільки значень різних класів присутні в датасеті.

```
table(clone$rings)

  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  29
  1   1  15  57 115 259 391 568 689 634 487 267 203 126 103  67  58  42  32  26  14   6   9   2   1   1   2   1
```

Рис.3.7. Таблиця відповідності кількості екземплярів класу

Наглядно видно, що кількість екземплярів молюсків з кількістю кілець в класах 1, 2, 3 20-29, замала для навчання моделі. Тому прийнято рішення видалити екземпляри, що не вплинуть на подальші результати.

```
table(clone$rings)
clone$rings <- as.factor(clone$rings)
clone <- clone[!clone$rings == 1,]
clone <- clone[!clone$rings == 2,]
clone <- clone[!clone$rings == 3,]
clone <- clone[!clone$rings == 18,]
clone <- clone[!clone$rings == 19,]
clone <- clone[!clone$rings == 20,]
clone <- clone[!clone$rings == 21,]
clone <- clone[!clone$rings == 22,]
clone <- clone[!clone$rings == 23,]
clone <- clone[!clone$rings == 24,]
clone <- clone[!clone$rings == 25,]
clone <- clone[!clone$rings == 26,]
clone <- clone[!clone$rings == 27,]
clone <- clone[!clone$rings == 28,]
clone <- clone[!clone$rings == 29,]
row.names(clone) <- 1:nrow(clone)
```

```
> table(clone$rings)

  4   5   6   7   8   9  10  11  12  13  14  15  16  17
 57 115 259 391 568 689 634 487 267 203 126 103  67  58
```

Рис.3.8. Відредагована вибірка екземплярів

3.2 Лінійна модель

Побудуємо лінійну модель та отримуємо коефіцієнти моделі.

```
abalone$var01<-ifelse(median(abalone$rings)<abalone$rings,1,0)
plot (rings, var01)
data.frame(abalone, var01<-ifelse(median(abalone$rings)<abalone$rings,1,0))

pairs(abalone)

cor(abalone[, setdiff(names(abalone), c("sex", "rings"))])
cor(abalone[, setdiff(names(abalone), c("sex", "var01"))])
 attach(abalone)
plot(`length`)
```

```
Call:
glm(formula = var01 ~ sex + diameter + length + height + `whole weight` +
    `shucked weight` + `viscera weight` + `shell weight`, family = binomial,
    data = abalone)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.1114  -0.6613  -0.2284   0.7009   2.4369

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       -2.93703    0.45063  -6.518 7.14e-11 ***
sexI              -0.74692    0.11601  -6.438 1.21e-10 ***
sexM               0.04202    0.09606   0.437   0.6618
diameter           5.81848    2.66227   2.186   0.0288 *
length            -4.75165    2.16814  -2.192   0.0284 *
height             6.10928    2.64049   2.314   0.0207 *
`whole weight`     8.53916    1.17996   7.237 4.59e-13 ***
`shucked weight` -16.41302    1.32575 -12.380  < 2e-16 ***
`viscera weight`  -5.64109    1.81630  -3.106   0.0019 **
`shell weight`    10.10485    1.71152   5.904 3.55e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5790.5  on 4176  degrees of freedom
Residual deviance: 3690.1  on 4167  degrees of freedom
AIC: 3710.1

Number of Fisher Scoring iterations: 5
```

Рис. 3.9. Лінійні моделі

```
(Intercept)        sexI        sexM    diameter      length      height `whole weight` `shucked weight` `viscera weight` `shell weight`
 -2.93703148 -0.74691610  0.04202476  5.81848145 -4.75165462  6.10928400    8.53915912    -16.41302328     -5.64109202    10.10484631
```

Рис. 3.10. Коефіцієнти моделі

На основі отриманих коефіцієнтів моделі, робимо передбачення.

glm_probs1=predict(glm_cl_fit, type="response")

glm_probs1[1:10]

```
        1         2         3         4         5         6         7         8         9        10
0.3312579 0.1492975 0.6699548 0.4241006 0.0650685 0.1284426 0.9593536 0.7731868 0.3999712 0.9418166
```

Рис. 3.11. Передбачення

Далі маючи лінійну модель, можемо виконати прогнозування. Будуємо таблицю неточностей та дослідимо скільки значень було передбачено неправильно.

Код:

```
glm_predict1 = rep('Worst', nrow(dataset))
glm_predict1[glm_probs1 > 0.5] = "Best"
table(glm_predict1, dataset$area_01)
```

```
glm_predict1    0    1
        Best   420 1618
        Worst 1676  463
```

Рис. 3.12. Результат навчання

Згідно даних точність мережі: 0.2113. Це **21.1%** правильно передбачених даних. Нажаль дана модель є малоефективною, але порівнюючи з випадковим розподілом відсоток правильних рішень становить саме 21.1% проти 7,14% відповідно.

3.3 LDA-модель

Далі проведемо класифікацію за допомогою лінійно дискримінантного аналізу. Для цього будемо використовувати оптимізовану вибірку даних, і на ній проведемо LDA. Але спочатку потрібно розподілити датасет на дві частини (рис. 3.8.) – ту що навчає, і та на якій тестується модель. Вибірку було розподілено класично 80% на 20% :

```
#Dividing in two groups
library(caret)
set.seed(1)
TrainingIndex <- createDataPartition(clone$rings, p=0.8, list = FALSE)
TrainingSet <- clone[TrainingIndex,] # Training Set
TestingSet <- clone[-TrainingIndex,] # Test Set
```

Рис. 3.13 Розподіл даних

Далі проведемо класифікацію за допомогою лінійно дискримінантного аналізу. Для цього будемо використовувати оптимізовану вибірку даних, і все на ній проведемо LDA. Але спочатку потрібно розподілити датасет на дві

частини – ту що навчає, і та на якій тестується модель. Вибірку було розподілено класично 80% на 20% :

```
Call:
lda(var01 ~ sex + diameter + length + height + `whole weight` +
    `shucked weight` + `viscera weight` + `shell weight`, data = abalone)

Prior probabilities of groups:
        0         1
0.5017955 0.4982045

Group means:
       sexI      sexM  diameter    length    height `whole weight` `shucked weight` `viscera weight` `shell weight`
0 0.5224237 0.2752863 0.3554437 0.4623688 0.1184709      0.5703185        0.2605701        0.1244034      0.1592352
1 0.1186929 0.4569918 0.4606968 0.5860596 0.1607136      1.0890286        0.4588770        0.2371889      0.3190002

Coefficients of linear discriminants:
                        LD1
sexI              -0.66498551
sexM               0.01244621
diameter           6.39377542
length            -3.50473395
height             4.14710321
`whole weight`     3.06015105
`shucked weight` -7.22052632
`viscera weight`   0.35160979
`shell weight`     4.41709007
```

Рис. 3.14. Результат навчання

За результатами LDA видно, що $\hat{\pi}_1 = 0.501$, $\hat{\pi}_2 = 0.499$, тобто 49% навчальних спостережень відповідають позитивним очікуванням. В результаті також представленні групові середні кожного предиктора для кожного класу LDA. Результати прогнозування наведені нижче.

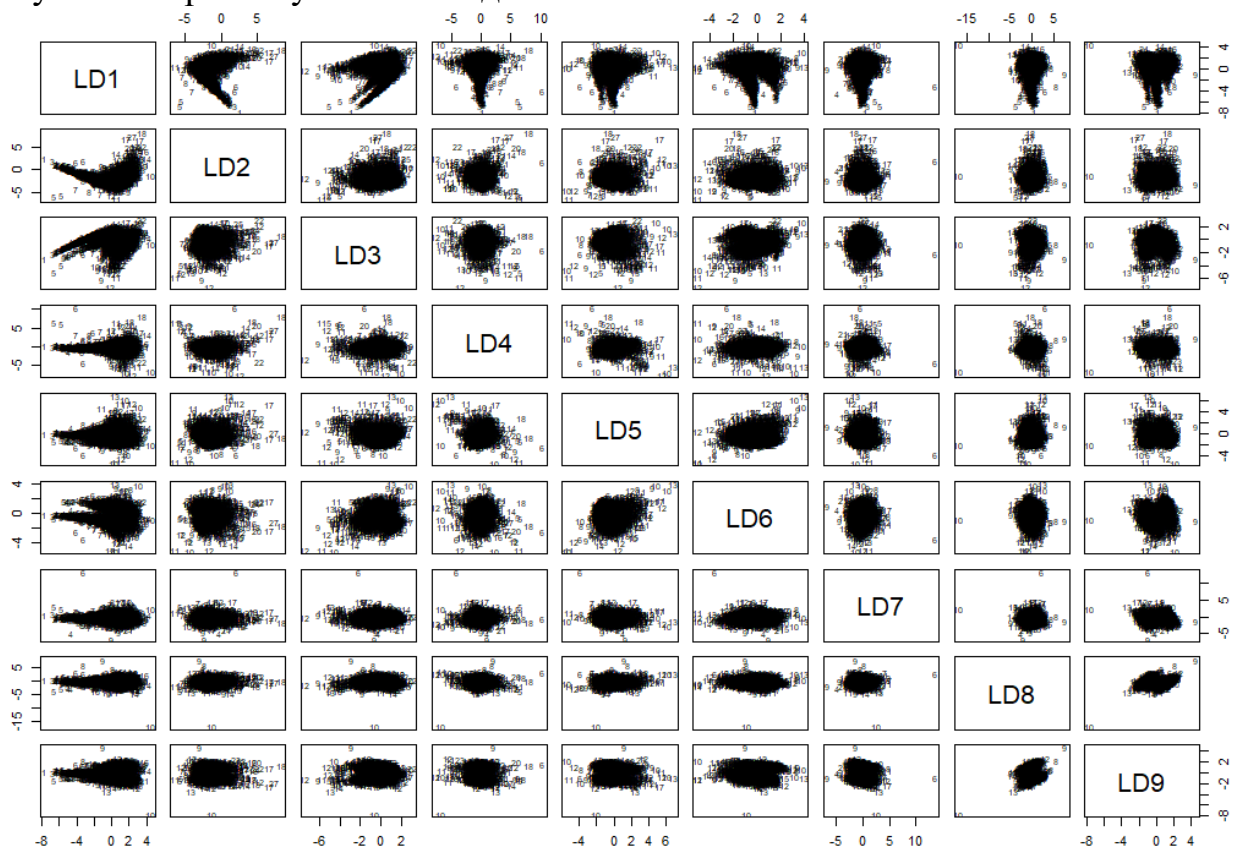

Рис. 3.15. Діаграми розсіяння за моделлю LDA

На малюнку показані графіки лінійних дискримінант, отриманих шляхом такого обчислення для кожного навчального спостереження. Виконаємо прогноз на тренувальній вибірці даних.

Досліджуємо точність мережі.

Код:

```
> tab
    Act
Pred   4   5   6   7   8   9  10  11  12  13  14  15  16  17
  4   28  18   5   0   0   0   0   0   0   0   0   0   0   0
  5   14  40  33  16   4   4   2   0   0   0   0   0   0   0
  6    4  24  70  59  20  10   4   2   1   0   0   0   0   0
  7    0   9  73 108  75  33  21   2   7   3   2   0   1   0
  8    0   1  13  63 134 127  57  44  13  10   8   6   2   1
  9    0   0  14  62 167 204 166  82  61  24  14  14   6   5
  10   0   0   0   4  46 126 146 119  52  47  30  20  12  12
  11   0   0   0   1   4  36  77 106  43  28   9   8   5   3
  12   0   0   0   0   1   6   5   6   2   7   1   1   2   1
  13   0   0   0   0   2   4  10  14  17  21  19  12   5   7
  14   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  15   0   0   0   0   0   0   4   1   1   3   1   1   1   1
  16   0   0   0   0   1   0   7   4   8   9   7  13  15   6
  17   0   0   0   0   1   2   9  10   9  11  10   8   5  11
> sum(diag(tab))/sum(tab)
[1] 0.2746435
```

Рис. 3.16 Результати LDA на тренувальній вибірці

Перевіримо ефективність моделі на тестувальній вибірці.

```
    Act
Pred  4  5  6  7  8  9 10 11 12 13 14 15 16 17
  4   5  6  1  1  0  0  0  0  0  0  0  0  0  0
  5   6  9  8  3  1  0  0  0  0  0  0  0  0  0
  6   0  4 17 18  4  2  1  4  0  1  0  0  0  0
  7   0  4 17 28 13  7  5  5  1  2  0  0  0  0
  8   0  0  2 18 50 26 14 10  2  3  0  2  0  1
  9   0  0  5  9 31 60 32 23 13  4  2  3  1  0
  10  0  0  0  1 12 33 45 27 14  9  9  4  2  2
  11  0  0  0  0  0  5 17 21 14  8  6  3  0  1
  12  0  0  0  0  1  1  2  1  0  1  1  1  1  0
  13  0  0  0  0  0  2  6  4  5  6  2  3  1  2
  14  0  0  0  0  0  0  0  0  0  0  0  0  1  0
  15  0  0  0  0  0  1  1  1  0  2  0  1  0  0
  16  0  0  0  0  1  0  2  0  2  1  3  0  4  3
  17  0  0  1  0  0  0  1  1  2  3  2  3  3  2
> sum(diag(tab2))/sum(tab2)
[1] 0.3107769
```

Рис. 3.17. Результати LDA на тестувальній вибірці

Згідно результату точність становить 27,46% та 31,07%, що значно більше ніж при випадковому розподілені.

3.4 SVM-модель

Проведемо аналогічні розрахунки над нашими даними за допомогою методу опорних векторів. Для побудови моделі, як і в попередній раз – використаймо усі атрибути:

```
Model <- train(rings ~ ., data = TrainingSet,
        method = "svmPoly",
        na.action = na.omit,

        trControl= trainControl(method="none"),
        tuneGrid = data.frame(degree=1,scale=1,C=1)
```

```
          Reference
Prediction   4   5   6   7   8   9  10  11  12  13  14  15  16  17
        4   25  13   3   0   0   0   0   0   0   0   0   0   0   0
        5   15  28  21   9   2   0   0   0   0   0   0   0   0   0
        6    4  37  55  37  14   8   1   0   0   0   0   0   0   0
        7    2  14 100 132  74  41  28   6   6   2   1   0   0   0
        8    0   0  15  61 139 109  43  33  10   8   6   4   2   0
        9    0   0  14  71 189 250 203 112  84  32  18  20   9   8
        10   0   0   0   3  37 135 199 189  77  93  63  43  27  25
        11   0   0   0   0   0   9  34  50  37  28  13  16  16  14
        12   0   0   0   0   0   0   0   0   0   0   0   0   0   0
        13   0   0   0   0   0   0   0   0   0   0   0   0   0   0
        14   0   0   0   0   0   0   0   0   0   0   0   0   0   0
        15   0   0   0   0   0   0   0   0   0   0   0   0   0   0
        16   0   0   0   0   0   0   0   0   0   0   0   0   0   0
        17   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

```
Overall Statistics

               Accuracy : 0.2722
                 95% CI : (0.2569, 0.2879)
    No Information Rate : 0.1711
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.1535

 Mcnemar's Test P-Value : NA
```

Рис. 3.18 Результати SVM

Результати моделювання продемонстрували точність передбачення на рівні 27,22%, що співставно з іншими розглянутими вище методами класифікації.

Підрахувавши число правильно прогнозованих відповідей, Отже модель навчилася розпізнавати з невеликою точністю, що є майже однаковим результатом у порівнянні з лінійним дискримінантним аналізом.

3.5 Метод К найближчих сусідів

Наступним для дослідження було надано метод К найближчих сусідів. Для проведення класифікації, виконаємо наступні підготовчі дії:

1. Матрицю з предикторами навчального набору означено – train_scale.
2. Матрицю з предикторами тестових даних – test_scale.
3. Вектор з мітками класів навчаюх спостережень – train_cl.
4. Використоване класифікатором число найближчих сусідів – К.

```
classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                      cl = train_cl$Rings,
                      k = 3)
classifier_knn
```

Виконавши процес моделювання виявилось, що модель не може бути побудована, через відсутність необхідної кількості і якості даних.

```
Error in knn(train = prc_train, test = prc_test, cl = prc_trainl, k = 20) :
  NA/NaN/Inf in foreign function call (arg 6)
In addition: Warning messages:
1: In knn(train = prc_train, test = prc_test, cl = prc_trainl, k = 20) :
  NAs introduced by coercion
2: In knn(train = prc_train, test = prc_test, cl = prc_trainl, k = 20) :
  NAs introduced by coercion
>
```

Рис. 3.19 Спроба прогнозування методом К найближчих сусідів

Таким чином, серед усіх побудованих моделей, ніякий результат не продемонструвала модель KNN.

3.6 Classification tree

Серед різних класифікаційних моделей виділяється дерево рішень. Його побудова теоретично зможе продемонструвати алгоритм чи залежність віднесення екземпляру до свого класу. Було перевірено і такий метод.

```
classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                      cl = train_cl$Rings,
                      k = 3)
classifier_knn
```

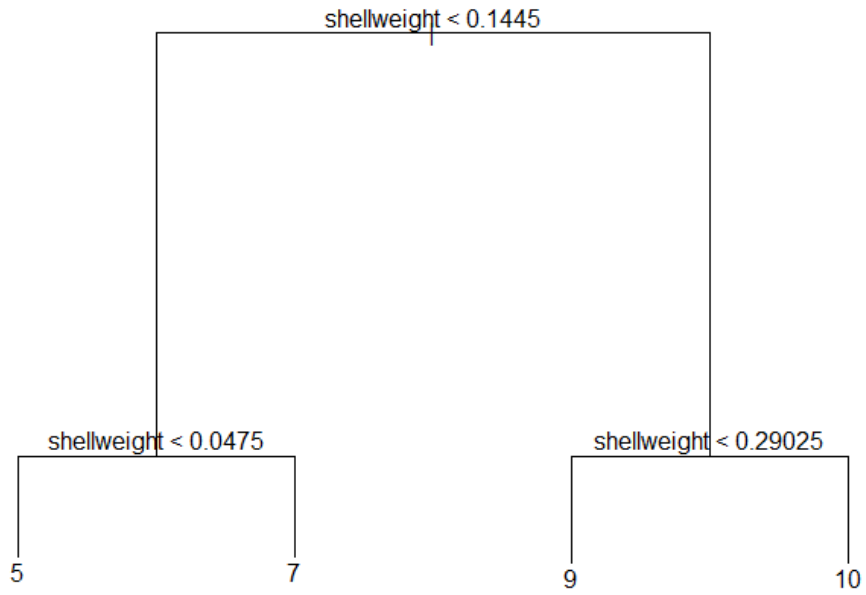

Рис. 3.20. Дерево класифікації

Незважаючи на сподівання, модель не змогла бути побудована. Дерево рішень (рис. 3.1) не демонструє алгоритм розподілу серед всього датасету і навіть не охоплює усі перелічені класи.

3.7 Random forest

Ще один метод, який набирає все більшої популярності, теж ідеологічно близький до дерев класифікації. Він називається "Random Forest", оскільки основою методу є виробництво великої кількості класифікаційних "дерев".

Зазвичай очікується значно більш висока ефективність цього порівняно з лінійним дискримінантним аналізом. Крім того, Random Forest дозволяє

з'ясувати значущість («importance») кожної ознаки, а також дистанції між усіма об'єктами тренувальної вибірки («proximity»), які потім можна використовувати для кластеризації або багатомірного шкалювання.

```
clone.rfp  4  5  6  7  8  9 10 11 12 13 14 15 16 17
        4  5  4  1  1  0  0  0  0  0  0  0  0  0  0
        5  4 11  5  1  1  0  0  0  0  0  0  0  0  0
        6  2  4 13 16  5  2  0  0  1  1  0  0  0  0
        7  0  4 18 24 14  5  4  5  0  3  1  0  0  0
        8  0  0  8 29 42 36 20  6  7  4  0  1  0  1
        9  0  0  3  4 35 38 27 20  5  1  2  2  1  0
        10 0  0  2  2  9 38 45 30 18 11  4  6  3  2
        11 0  0  0  0  4 12 18 24 12  9  7  4  2  1
        12 0  0  0  1  2  1  7  3  2  0  1  4  1  2
        13 0  0  1  0  1  3  2  4  4  8  5  2  3  2
        14 0  0  0  0  0  1  1  3  2  2  3  1  0  1
        15 0  0  0  0  0  0  1  0  1  0  1  0  1  1
        16 0  0  0  0  0  0  1  1  1  0  1  0  2  1
        17 0  0  0  0  0  1  0  1  0  1  0  0  0  0
> sum(diag(table(clone.rfp,TestingSet[,9])))/sum(table
[1] 0.2719298
```

Рис. 3.21 Random Forest прогнозування

Нарешті, цей метод дозволяє робити «чисту візуалізацію» даних, тобто може працювати як метод класифікації без навчання (рис. 3.22):

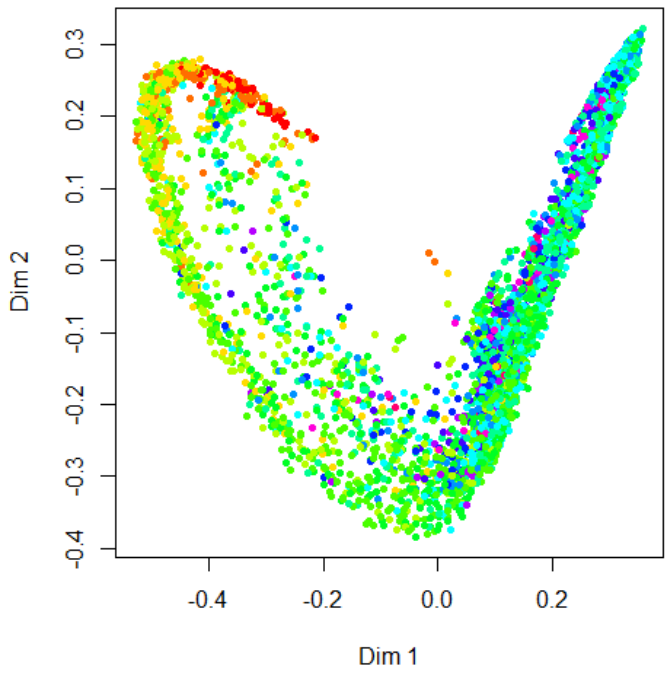

Рис. 3.22 Візуалізація Random Forest прогнозування

Нарешті, цей метод дозволяє робити «чисту візуалізацію» даних, тобто може працювати як метод класифікації без навчання (рис. 3.22):

Очевидно, що результат навчання досягає 27,19%, що знову є співставно з іншими методами класифікації.

3.8 Аналіз отриманих результатів

Таблиця 3.1.

Зведені результати класифікації

Назва методу	Точність Тренувальна	Точність Тестова
Матриця неточностей	21.1%	
LDA	**27.46%**	**31.07%**
SVM	27.22%	
Метод випадкового лісу	27.19%	
Метод К найближчих сусідів	-	
Метод бінарного дерева	-	
Випадковий розподіл	7,14%	

Аналізуючи отримані результати можна зробити декілька висновків.

По-перше, хоч деякі методи машинного навчання демонстрували високі показники точності прогнозів, вони належать до інших класів вирішення задач.

По-друге, методи класифікації не є ефективними для використання у реальному житті, незважаючи на те, що вони демонструють вищу ефективність від випадкової.

По-третє, вирішення задачі регресії методами класифікації не мали успіху, можливо і через те, що у датасеті дані були розподілені нерівномірно. Таким чином, модель могла перенавчатись у розпізнанні одних і недонавчатись у розпізнанні інших класів.

Висновки до розділу 3

У розділі було розглянуто застосування методів класифікації на основі машинного навчання. Побудовано діаграми розподілу. Досліджено ступінь кореляції між полями датасету. Створено різні класифікаційні моделі та було досягнуто таких результатів:

- загальні лінійні моделі (**21.1%**);
- лінійний дискримінантний аналіз (**27.46%, 31,07%**);
- метод опорних векторів (**27.22%**);
- метод випадкового лісу (**27.19%**)
- метод найближчих сусідів (не ефективний)
- метод класифікаційних дерев (не ефективний)
- метод випадкового розподілу (**7.14%**).

Згідно цих методів для даної задачі найкращим виявився лінійно-дискримінантний аналіз **31,07%**. Не набагато гіршими виявилися метод випадкового лісу та метод опорних векторів з результатом 27.19%, 27,22% відповідно. Найгіршим результатом є метод найближчих сусідів через неможливість свого побудування, та метод класифікаційних дерев. Можна сказати, що перевага за точність між найкращими результатами умовна.

ВИСНОВКИ

Хоча методів штучного інтелекту існує дуже багато і для різних задач потрібні різні підходи, ці методи надають надзвичайні можливості в сфері навчання на даних та передбачені результатів. Ця сфера є і буде актуальною ще багато років.

З теоретичних завдань виконано під час написання магістерської кваліфікаційної роботи було проведено аналіз датасету для прогнозування віку молюсків. А також:

- Проаналізовано сучасний стан задачі;
- Досліджено математичну модель, на якій побудовано основні методи машинного навчання;
- Застосовано порівняння різних підходів для заданої задачі та обрано найкращі, які і були використані на практиці.

Під час практичної частини налаштовано скрипти мовою програмування R для роботи з набором даних abalone. А також:

- Проведено аналіз усіх змінних для дослідження найбільш впливових, та найменш впливових.
- Проведено навчання та тестування класифікаторів.
- Проведено порівняльний аналіз результатів.
- Досліджено результати навчання моделей з різними вхідними параметрами налаштування.

Але найголовніше, що створено моделі, що можуть з певною точністю прогнозувати реальний вік молюсків, але на жаль не можуть бути використані в реальному світі.

Поставлені завдання виконано повністю, однак є проблеми, які потребують подальшої розробки: налаштування більш глибокого динамічного аналізу зібраної інформації.

Створено різні класифікаційні моделі та було досягнуто таких результатів:

- загальні лінійні моделі (**21.1%**);
- лінійний дискримінантний аналіз (**27.46%, 31,07%**);
- метод опорних векторів (**27.22%**);
- метод випадкового лісу (**27.19%**)
- метод найближчих сусідів (не ефективний)
- метод класифікаційних дерев (не ефективний)
- метод випадкового розподілу (**7.14%**).

Згідно цих методів для даної задачі найкращим виявився лінійно-дискримінантний аналіз **31,07%**. Не набагато гіршими виявилися метод випадкового лісу та метод опорних векторів з результатом 27.19%, 27,22% відповідно. Найгіршим результатом є метод найближчих сусідів через неможливість свого побудування, та метод класифікаційних дерев. Можна сказати, що перевага за точність між найкращими результатами умовна.

Використано основні методи для вирішення поставленої задачі. Отже згідно з поставленим завданням та метою виконано усі пункти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gofas, Serge; Tran, Bastien; Bouchet, Phillippe (2014). "WoRms Taxon Details: Haliotis Linnaeus, 1758". WoRMS (World Register of Marine Species). Retrieved16 August 2020.

2. Beesley, P. L.; Ross, G. J. B.; Wells, A. (1998). Mollusca: The Southern Synthesis: An Essential Reference. Melbourne, Australia: CSIRO Publishing. pp. 667–669.

3. Dauphin, Y.; Cuif, J. P.; Mutvei, H.; Denis, A. (1989). "Mineralogy, Chemistry and Ultrastructure of the External Shell-layer in Ten Species of Haliotis With Reference to Haliotis tuberculata (Mollusca, Archaeogastropoda)". Bulletin of the Geological Institutions of the University of Uppsala. 15: 7–38.

4. Cox, Keith W. (1962). "California abalone, family Haliotidae". The Resources Agency of California Department of Fish and Game: Fish Bulletin. 118.ISSN 6306-2593.

5. Geiger, Daniel L.; Owen, Buzz (2012). Abalone: Worldwide Haliotidae. Hackenheim, Germany: Conchbooks.

6. Hoiberg, Dale H., ed. (1993). Encyclopædia Britannica. 1: A-ak Bayes (15th ed.). Chicago, IL: Encyclopædia Britannica, Inc.

7. Tryon, Jr., George W. (1880). Manual of Conchology; Structural and Systematic With Illustrations of the Species (PDF). II: Muricinæ, Purpurinæ. Philadelphia, PA: Academy of Natural Sciences.

8. Anon (2014g). "Distribution Map: Haliotis". Ocean Biogeographic Information System. Retrieved 22 August 2020.

9. Leatherman, Stephen (2012). National Geographic Field Guide to the Water's Edge. National Geographic Field Guides. National Geographic. p. 93. ISBN 978-1426208683.

10. "Hypersensitivity Pneumonitis". www.clevelandclinicmeded.com. Retrieved 17 January 2020.

11. Witten, I. H., & Frank, E. (2005). Data mining: practical machine learning tools and techniques.

12. Machine Learning Repository. URL: http://archive.ics.uci.edu/ml/datasets/Airfoil+Self-Noise.

13. Aha, D. W., Kibler, D. & Albert, M. K. (1991). Instance-based learning algorithms. Mach Lear., 6(1), 37–66

14. Rogers-Bennett, Laura; et al. (2002). "Using Spatially Explicit Data to Evaluate Marine Protected Areas for Abalone in Southern California". Conservation Biology. 16 (5): 1308–1317.

15. A. Lofty and A. Benetton, “Using Probabilistic Neural Networks for Handwritten Digit Recognition”, Journal of Artificial Intelligence, vol. 4, no. 4, (2011).

16. P. Khanate and S. Chitins, “Handwritten Devanagari Character Recognition using Artificial Neural Network”, Journal of Artificial Intelligence, vol. 4, no. 1, (2011).

17. P. Erika and R. Udegbunam, “Application of neural network in evaluating prices of housing units in Nigeria: A preliminary investigation”, J. of Artificial Intelligence, vol. 3, no. 1, (2010).

18. H. Martin and D. Howard, “Neural Network Design”, 2nd Edition, Martin Hagan (2014).

19. Askarzadeh, A., and A. Rezazadeh. 2013. “Artificial Neural Network Training Using a New Efficient Optimization Algorithm.” Applied Soft Computing, 13(2): 1206– 1213.

20. UCI Machine Learning repository (https://archive.ics.uci.edu/ml/datasets.html)

21. Фісун М. Т., Журавська І. М. Методичні вказівки до оформлення звітної текстової документації та кваліфікаційних робіт з дисциплін, закріплених за факультетом комп’ютерних наук. Миколаїв : Вид-во ЧДУ ім. Петра Могили, 2012. 126 с.

ДОДАТОК А

Лістинг програми

```
par("mar")
par(mar=c(1,1,1,1))

library(caret)
library(randomForest)
install.packageS(AUC)
library(MASS)

abalone <- read.csv("D:/kursova/abalone.data", header=FALSE, sep = ",",as.is = TRUE)
clone$rings <- as.numeric(clone$rings)

head(abalone)
tail(abalone)
summary(abalone)
View(abalone)

names(abalone)[names(abalone) == 'V1'] <- 'sex'
names(abalone)[names(abalone) == 'V2'] <- 'length'
names(abalone)[names(abalone) == 'V3'] <- 'diameter'
names(abalone)[names(abalone) == 'V4'] <- 'height'
names(abalone)[names(abalone) == 'V5'] <- 'wholeweight'
names(abalone)[names(abalone) == 'V6'] <- 'shuckedweight'
names(abalone)[names(abalone) == 'V7'] <- 'viscera weight'
names(abalone)[names(abalone) == 'V8'] <- 'shellweight'
names(abalone)[names(abalone) == 'V9'] <- 'rings'

clone <- abalone

table(clone$rings)
clone$rings <- as.factor(clone$rings)
clone <- clone[!clone$rings == 1,]
clone <- clone[!clone$rings == 2,]
clone <- clone[!clone$rings == 3,]
clone <- clone[!clone$rings == 18,]
clone <- clone[!clone$rings == 19,]
clone <- clone[!clone$rings == 20,]
clone <- clone[!clone$rings == 21,]
clone <- clone[!clone$rings == 22,]
clone <- clone[!clone$rings == 23,]
clone <- clone[!clone$rings == 24,]
clone <- clone[!clone$rings == 25,]
clone <- clone[!clone$rings == 26,]
clone <- clone[!clone$rings == 27,]
clone <- clone[!clone$rings == 28,]
clone <- clone[!clone$rings == 29,]
row.names(clone) <- 1:nrow(clone)


#Dividing in two groups
library(caret)
set.seed(1)
TrainingIndex <- createDataPartition(clone$rings, p=0.8, list = FALSE)
TrainingSet <- clone[TrainingIndex,] # Training Set
TestingSet <- clone[-TrainingIndex,] # Test Set

#Linear Discriminant Analysis
```

```
library(MASS)
lda_model <- lda(rings~., TrainingSet)
plot(lda_model)

attributes(lda_model)
p <- predict(lda_model, TrainingSet)

  #lda training prediction
p1 <- predict(lda_model, TrainingSet)$class
tab <- table(Pred = p1, Act = TrainingSet$rings)
tab
s(tab)
accu
sum(diag(tab))/sum(tab)

  #testing prediction
p2 <- predict(lda_model, TestingSet)$class
tab2 <- table(Pred = p2, Act = TestingSet$rings)
tab2
sum(diag(tab2))/sum(tab2)

  #LDA graphs
ldahist(data = p$x[,1], g = TrainingSet$rings)
roc.perf = performance(tab2, measure = "tpr", x.measure = "fpr")
plot(roc.perf)

library(tree)

clone.tree <- tree(clone[,9] ~ ., clone[,-9])
plot(clone.tree)
text(clone.tree)

#RANDOM FOREST
library(randomForest)

iris.plot <- randomForest(clone[,-9])
MDSplot(iris.plot, clone[,9])

clone.rf <- randomForest(TrainingSet[,9] ~ ., TrainingSet[,1:8])
clone.rfp <- predict(clone.rf, TestingSet[,1:8])
table(clone.rfp,TestingSet[,9])
sum(diag(table(clone.rfp,TestingSet[,9])))/sum(table(clone.rfp,TestingSet[,9]))

plot(clone.rf)

#knn
# Installing Packages
install.packages("e1071")
install.packages("caTools")
install.packages("class")

# Loading package
library(e1071)
library(caTools)
library(class)


# Splitting data into train
# and test data
split <- sample.split(clone, SplitRatio = 0.7)
train_cl <- subset(clone, split == "TRUE")
```

```
test_cl <- subset(clone, split == "FALSE")

# Feature Scaling
train_scale <- scale(train_cl[, 1:9])
test_scale <- scale(test_cl[, 1:9])

# Fitting KNN Model
# to training dataset
classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                      cl = train_cl$Rings,
                      k = 1)
classifier_knn

# Confusion Matrix
cm <- table(test_cl$Species, classifier_knn)
cm

# Model Evaluation - Choosing K
# Calculate out of Sample error
misClassError <- mean(classifier_knn != test_cl$Species)
print(paste('Accuracy =', 1-misClassError))

# K = 3
classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                      cl = train_cl$Species,
                      k = 3)
misClassError <- mean(classifier_knn != test_cl$Species)
print(paste('Accuracy =', 1-misClassError))


install.packages("class")
library(class)
```