

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))

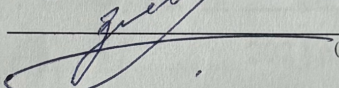
Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

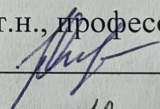
МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

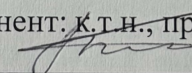
«Інформаційна технологія вирішення задачі
нейромережевого визначення якості вина»

Виконав: студент 2-го курсу, групи 1КН-23м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)


Закревський А.О.
(прізвище та ініціали)

Керівник: к.т.н., професор каф. КН

Колесницький О.К.
(прізвище та ініціали)

« 05 » 12 2024 р.

Опонент: к.т.н., професор каф. КСУ

Биков М. М.
(прізвище та ініціали)

« 05 » 12 2024 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А. 

(прізвище та ініціали)

« 06 » 12 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та
автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
Д.т.н., проф. Яровий А.А.

11. 10. 2024 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Закревському Артуру Олеговичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Інформаційна технологія нейромережевого визначення якості вина.
Керівник роботи: к.т.н., проф. кафедри КН Колесницький О. К.
затверджені наказом вищого навчального закладу від "17" 09 202_ року № 310

2. Строк подання студентом роботи 11. 11. 2024 року.

3. Вихідні дані до роботи:

Вхідні дані – кількість аналізованих параметрів вина – 11; обсяг навчальної вибірки – не менше 1000 записів; обсяг тестової вибірки – не менше 400 записів; кількість класів вина – не менше 2, вид класифікатора – нейронна мережа, використання об'єктно-орієнтованої мови програмування.

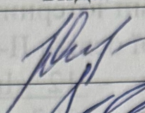
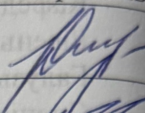
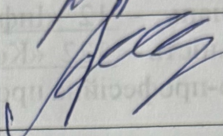
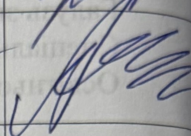
4. Зміст текстової частини:

Вступ, аналіз предметної області інформаційних технологій визначення якості вина, розробка інформаційної технології визначення якості вина, програмна реалізація інформаційної технології визначення якості вина, тестування та аналіз результатів роботи програми визначення якості вина, економічна частина, висновки, перелік використаних джерел, додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень):

Алгоритм роботи програми, UML діаграма класів програми, робочі вікна програми, результати тестування розробленої програми та програми-аналога.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-4	Колесницький О.К., к.т.н., проф. каф. КН		
5	Лесько О. Й., к. е. н., проф. зав. каф. ЕПВМ		

7. Дата видачі завдання 02. 09. 2024 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Прізвище
1	Аналіз сучасного рівня інформаційних технологій визначення якості вина. Постановка задач дослідження	02.09.24 – 16.09.24	
2	Побудова моделей нейромережевого визначення якості вина	17.09.24 – 26.09.24	
3	Практичне застосування та оцінка ефективності розроблених моделей	27.09.24 – 04.10.24	
4	Підготовка економічної частини	05.10.24 – 19.10.24	
5	Апробація та/або впровадження результатів дослідження	20.10.24 – 27.10.24	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	28.10.24 – 11.11.24	

Студент

Закревський А.О.

(підпис)

Керівник роботи

Колесницький О.К.

(підпис)

АНОТАЦІЯ

УДК 004.8

Закревський А. О. Інформаційна технологія вирішення задачі неймережевого визначення якості вина. Магістерська кваліфікаційна робота зі спеціальності 122 – «Комп'ютерні науки», освітня програма – «Системи штучного інтелекту». Вінниця: ВНТУ, 2024. 91 с.

На укр. мові. Бібліогр.: 21 назв; рис.: 18; табл. 8.

Дана магістерська кваліфікаційна робота присвячена розробці програмного забезпечення для інформаційної технології неймережевого визначення якості вина. У роботі було обґрунтовано вибір нейронної мережі багатошаровий персептрон для визначення якості вина, яка має 11 входів, 2 прихованих шари по 8 нейронів та вихідний шар із одного нейрона. У прихованих шарах обрано функцію активації ReLU та функцію активації Sigmoid у вихідному шарі. Для навчання цієї неймережі використовується модифікація ADAM методу стохастичного градієнтного спуску. Для створення програми було використано мову програмування Python та спеціалізовані бібліотеки NumPy, Pandas та Matplotlib. Навчання нейронної мережі відбувалось з використанням бази даних для червоних вин, яка налічує 1599 записів. Набір даних було поділено на навчальну (75%) та тестову (25%) вибірки: обсяг навчальної вибірки склав 1199 записів, а тестової - 400 записів. Розроблена програма має достовірність класифікації (визначення якості вина) на тестовій вибірці 79,2%, а найкращий із 3-х методів-аналогів має достовірність класифікації на тестовій вибірці 73,5%, тобто достовірність класифікації збільшилась на 5,7%.

Графічна частина складається з 6 плакатів.

У економічному розділі доведено доцільність проведення науково-дослідної роботи за даною темою. Термін окупності становить 4,5 р., що свідчить про комерційну привабливість науково-технічної розробки.

Ключові слова: класифікація, машинне навчання, нейронна мережа, двофакторний аналіз.

ABSTRACT

Zakrevskyi A. O. Information technology for solving the problem of neural network determination of wine quality. Master's thesis in the specialty 122 - «Computer sciences», educational program – «Artificial intelligence systems». Vinnytsia: VNTU, 2024. 91 p.

In Ukrainian language. Bibliogr. : 21 titles; fig. 18; table 8.

This master's qualification work is devoted to the development of software for information technology of neural network determination of wine quality. The work justified the choice of a multilayer perceptron neural network for determining wine quality, which has 11 inputs, 2 hidden layers of 8 neurons each and an output layer of one neuron. The ReLU activation function and the Sigmoid activation function in the output layer were selected in the hidden layers. To train this neural network, a modification of the ADAM stochastic gradient descent method was used. The Python programming language and specialized libraries NumPy, Pandas and Matplotlib were used to create the module. The module was trained using a database for red wines, which has 1599 records. The data set was divided into a training (75%) and a test (25%) sample: the volume of the training sample was 1199 records, and the test - 400 records. The developed non-network module has a classification accuracy (determination of wine quality) on the test sample of 79.2%, and the best of the 3 analogue methods has a classification accuracy on the test sample of 73.5%, i.e. the classification accuracy has increased by 5.7%.

The graphic part consists of 6 posters.

In the economic section, the expediency of conducting research work on this topic is proven. The payback period is 4,2 years, which indicates the commercial attractiveness of scientific and technical development.

Keywords: classification, machine learning, neural network, two-factor analysis.

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИЗНАЧЕННЯ ЯКОСТІ ВИНА.....	8
1.1 Постановка задачі	8
1.2 Огляд відомих методів класифікації в машинному навчанні	9
1.3 Обґрунтування вибору способу виявлення переваг розробленої інформаційної технології визначення якості вина	14
1.4 Висновок до розділу 1	15
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НЕЙРОМЕРЕЖЕВОГО ВИЗНАЧЕННЯ ЯКОСТІ ВИНА.....	16
2.1 Обґрунтування вибору типу нейронної мережі для визначення якості вина.....	16
2.2 Архітектура нейронної мережі багатошаровий персептрон для інформаційної технології визначення якості вина	23
2.3 Розробка алгоритму роботи інформаційної технології визначення якості вина	29
2.4 Висновок до розділу 2	33
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НЕЙРОМЕРЕЖЕВОГО ВИЗНАЧЕННЯ ЯКОСТІ ВИНА.....	34
3.1 Обґрунтування вибору мови програмування та спеціалізованих бібліотек.....	34
3.2 Опис набору даних параметрів вина для навчання та тестування нейромережі.....	38
3.3 Програмна реалізація інформаційної технології визначення якості вина на основі нейромережі.....	40
3.4 Висновок до розділу 3	49

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ НЕЙРОМЕРЕЖЕВОГО ВИЗНАЧЕННЯ ЯКОСТІ ВИНА.....	50
4.1 Тестування програми нейромережевого визначення якості вина ...	50
4.2 Аналіз результатів роботи програми нейромережевого визначення якості вина	59
4.3 Висновок до розділу 4	60
5 ЕКОНОМІЧНА ЧАСТИНА	61
5.1 Проведення комерційного та технологічного аудиту інформаційної технології вирішення задачі нейромережевого визначення якості вина	61
5.2 Розрахунок витрат на здійснення науково-дослідної роботи	62
5.3 Розрахунок економічної ефективності науково-технічної розробки інформаційної технології вирішення задачі нейромережевого визначення якості вина за її можливої комерціалізації потенційним інвестором.....	68
5.4 Висновок до розділу 5	73
ВИСНОВКИ	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	76
Додаток А (обов'язковий) ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ	79
Додаток Б (обов'язковий) Лістинг програми	80
Додаток В (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА	85

ВСТУП

Актуальність. Кожен виробник вина стикається з необхідністю його оцінки та сертифікації. Сертифікація дозволяє гарантувати, що вино відповідає стандартам і є безпечним для вживання продуктом. Оцінка вина дозволяє визначити його якість з погляду споживача, тобто наскільки гарний його смак, запах і т. д. Її результати можуть бути використані для покращення процесу виробництва, для розподілу вин за рейтингом та визначення його ціни. Оцінка інколи є частиною процесу сертифікації.

Існує два основних методи, що використовуються при сертифікації вина. Перший ґрунтується на фізико-хімічному аналізі. При цьому перевіряються, зокрема, такі параметри, як колір вина, наявність домішок, кількість алкоголю, цукрів і різних кислот, що містяться. Другим способом є сенсорний аналіз. Він заснований на використанні як оціночного приладу спеціально навченої людини – дегустатора. У процесі роботи, дегустатор повинен визначити якість вина, ґрунтуючись на своїх знаннях та використовуючи свої органи чуття.

Залежність між фізико-хімічними властивостями вина та результатом дегустації є вкрай складною та заплутаною, і в даний час вона до кінця не визначена. З цього випливає, що за допомогою фізико-хімічного аналізу складно дати відповідь на запитання, наскільки добре буде вино з погляду покупця, а отже, не можна віднести вино до певного бренду. З іншого боку, дегустатор визначає якість вина виходячи зі своїх суб'єктивних понять про «хороше» і «погане» вина. Це призводить до того, що вино, яке виграло певний конкурс, може не отримати жодних нагород в іншому змаганні, а оцінки різних дегустаторів, дані одному і тому ж вину, можуть відрізнятися. Подібні події підривають довіру до існуючої системи оцінювання вин, від чого можуть постраждати як виробники, які не зможуть розділити свою продукцію за рівнем якості та відповідно ціною. Так і споживачі, які

втрачати можливість вибирати вино за якістю, і, по суті, купуватимуть «кота в мішку». Отже, у сфері виробництва винної продукції існує потреба у дослідженні можливості створення деякого класифікатора, який міг би визначати якість вина за його хімічним складом, керуючись лише математичними алгоритмами [1].

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка прикладних інтелектуальних інформаційних технологій та систем» та плану наукової та навчально-методичної роботи кафедри.

Мета і завдання досліджень. Метою магістерської кваліфікаційної роботи є підвищення достовірності визначення якості вина за рахунок використання нейронної мережі.

Для досягнення мети розробки необхідно виконати такі задачі:

- провести аналіз предметної області визначення якості вина;
- розглянути існуючі методи побудови систем визначення якості вина та обрати й обґрунтувати вибір методу, який задовольняє мету даної магістерської кваліфікаційної роботи;
- обґрунтувати вибір типу нейромережі;
- розробити структуру нейронної мережі для інформаційної технології визначення якості вина
- сформулювати стадії інформаційної технології, розробити структуру та алгоритм роботи програмного засобу;
- виконати програмну реалізацію запропонованої інформаційної технології;
- провести тестування програмного продукту та виконати аналіз отриманих результатів.

Об'єкт дослідження – процес комп'ютеризованого визначення якості вина за допомогою нейронної мережі.

Предмет дослідження – інформаційна технологія та програмні засоби для визначення якості вина за допомогою нейронної мережі та достовірність їх роботи.

Методи дослідження. У роботі використані наступні методи наукових досліджень: системного аналізу, теорії штучних нейронних мереж, методи математичної статистики для розробки процесу розв'язання задачі та обрахунків результатів експериментів із програмним засобом, об'єктно-орієнтованого програмування.

Наукова новизна одержаних результатів.

1. Набула подальшого розвитку інформаційна технологія визначення якості вина, яка відрізняється використанням штучної нейронної мережі, що дозволило підвищити достовірність програмних засобів визначення якості вина.

Практичне значення одержаних результатів полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення нейромережевого визначення якості вина.

Запропонована інформаційна технологія сприяє підвищенню достовірності програмних засобів визначення якості вина, зокрема:

- розроблено алгоритм роботи програмного забезпечення визначення якості вина;
- розроблено програмні засоби для визначення якості вина.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю використання математичного апарату методів дослідження, експериментальними дослідженнями тестування програмної реалізації інформаційної технології визначення якості вина. Адекватність розроблених математичних моделей підтверджується результатами експериментальних досліджень.

Особистий внесок здобувача. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У працях, написаних у співавторстві, здобувачу належать: аналіз методів визначення якості вина та методів підвищення достовірності програмних засобів визначення якості вина [1].

Апробація результатів роботи. Результати роботи були апробовані на Міжнародній науково-практичній Інтернет-конференції студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2025)», Вінниця, 15 жовтня 2024 року - 15 червня 2025 року» [1].

Публікації. За результатами досліджень опубліковано тези доповіді на науково-технічній конференції [1].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВИЗНАЧЕННЯ ЯКОСТІ ВИНА

1.1 Постановка задачі

У цій роботі досліджується програмна реалізація класифікатора, заснованого на нейронній мережі та здатного визначати якість вина. Як дані для аналізу використовується інформація про склад вин та оцінки дегустаторів. Набір даних охоплює вина, створені в Португалії в провінції Мінью з травня 2004 року по лютий 2007 року [2].

Вихідний набір даних містить 4898 прикладів для білих вин та 1599 прикладів для червоних вин. Кожен приклад складається з 11 атрибутів та номера класу якості вина. Клас якості вина задається числами від 3 до 9. Атрибути характеризують такі хімічні параметри вина:

1. Фіксована кислотність (Fixed acidity): Рівень кислотності у вині.
2. Летюча кислотність (Volatile acidity): кількість оцтової кислоти, присутньої у вині.
3. Лимонна кислота (Citric acid): кількість лимонної кислоти у вині.
4. Залишковий цукор (Residual sugar): кількість цукру після бродіння.
5. Хлориди (Chlorides): кількість солей у вині.
6. Вільний діоксид сірки (Free sulfur dioxide): вільна форма SO_2 .
7. Загальний діоксид сірки (Total sulfur dioxide): кількість вільної та зв'язаної форм SO_2 .
8. Щільність (Density): Щільність вина (маса/об'єм).
9. Кислотність pH (pH): pH вина в діапазоні від 0 до 14.
10. Сульфати (Sulphates): рівень діоксиду сірки (SO_2) у вині.
11. Алкоголь (Alcohol): кількість алкоголю у вині.

Цільовим параметром є чисельно виражена якість вина, яка для відомих прикладів набуває значення від 3 до 9 включно.

Необхідно розробити нейромережевий класифікатор, що дозволяє за набором вхідних ознак визначити клас якості вина.

1.2 Огляд відомих методів класифікації в машинному навчанні

Завдання класифікації полягає у визначенні, якому класу належать об'єкти, що спостерігаються, ґрунтуючись на деякому попередньо відомому наборі пар, що складаються з опису об'єкта і класу, якому цей об'єкт належить. Опис об'єкта є набір вимірних характеристик.

Математична постановка задачі виглядає наступним чином: є множина класів Y , що не перетинаються, і множина описів об'єктів X .

$$g: X \rightarrow Y. \quad (1.1)$$

існує за
$$g(x_i) = y_i \quad (1.2)$$

де
$$(x_i, y_i) \quad x_i \in X, y_i \in Y \quad (1.3)$$

Існує багато різних методів класифікації об'єктів [3]. Вибір методу на вирішення поставленої задачі необхідно здійснювати виходячи з особливостей предметної області.

1.2.1 Логістична регресія (Logistic Regression)

Логістична регресія виводить прогнози щодо точок у бінарному масштабі — нульовому чи одиничному. Якщо значення чогось дорівнює чи

більше 0.5, то об'єкт класифікується у бік (до одиниці). Якщо значення менше 0.5 - у меншу (на нуль).

У кожної ознаки є своя мітка, що дорівнює лише 0 або лише 1. Логістична регресія є лінійним класифікатором і тому використовується, коли в даних простежується якась лінійна залежність.

1.2.2 Метод опорних векторів (Support Vector Machine)

Для вирішення завдань класифікації у 1963 році В. Вапник та А. Червоненкіс запропонували метод, названий ними методом опорних векторів. Метод заснований на пошуку гіперплощини у просторі ознак, що оптимально розділяє об'єкти на класи. Набір ознак, що характеризує певний об'єкт, називається вектором. Припустимо, ми маємо набір пар, зображений на рис. 1.1.

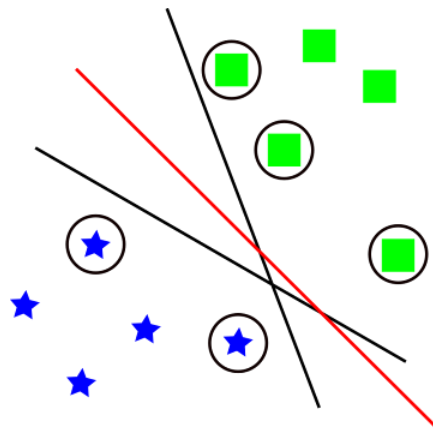


Рисунок 1.1 – Приклад гіперплощини, що розділяє класи

Гіперплощиною у певному просторі називається підпростір з розмірністю, меншою на 1, ніж розмірність самого простору. Тоді у двовимірному просторі гіперплощиною буде пряма. Можна побудувати кілька прямих, які розділяють два класи, як показано на рис.1.1. Ідея методу полягає в тому, що для вирішення задачі класифікації найкраще вибрати таку гіперплощину (а в даному випадку – пряму), щоб відстань від неї до

найближчих елементів класів була максимальною. Приклад подібної гіперплощини, що оптимально розділяє, можна побачити на рис. 1.1. Це пряма, намальована червоним кольором. Вектори, які розташовані найближче до роздільної гіперплощини, називають опорними векторами. На малюнку вони позначені за допомогою кіл [3].

Не завжди набори векторів є лінійно-роздільні. На рис. 1.2 показаний приклад, в якому неможливо знайти пряму лінію, яка розділяла б два класи. У такому випадку, можна використовувати перетворення, яке переводило б вектора в простір більшої розмірності. Тоді, згідно з теоремою Ковера, після такого перетворення, з високою ймовірністю можна отримати лінійно-розділені класи [4,5].

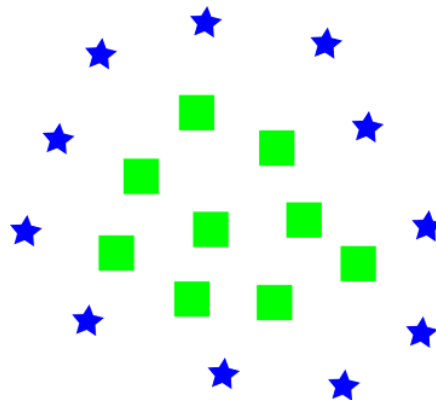


Рисунок 1.2 – Приклад лінійно-нероздільних класів

1.2.3 Метод k-найближчих сусідів (K-Nearest Neighbors)

Цей метод працює за допомогою пошуку найкоротшої дистанції між об'єктом, що класифікується, і найближчими до нього класифікованими об'єктами з навчального набору. Об'єкт, що класифікується, буде відноситися до того класу, до якого належить найближчий об'єкт набору (Рис. 1.3).

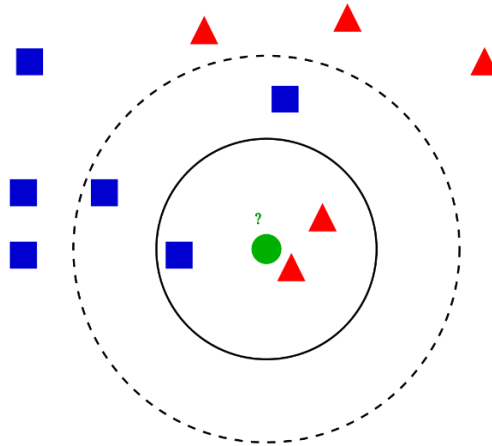


Рисунок 1.3 – Метод k-найближчих сусідів

1.2.4 Класифікатор дерева рішень (Decision Tree Classifier)

Цей класифікатор розбиває дані на все менші та менші підмножини на основі різних критеріїв, тобто у кожної підмножини своя категорія, що сортує. З кожним поділом кількість об'єктів певного критерію зменшується.

Класифікація добігає кінця, коли мережа дійде до підмножини лише з одним об'єктом. Якщо поєднати кілька подібних дерев рішень, то вийде так званий випадковий ліс (Random Forest).

1.2.5 Байєсівський класифікатор

Це набір імовірнісних методів класифікації. Вони засновані на теорії, що якщо відомі густини розподілу класів, то можна у явному вигляді виписати алгоритм класифікації. Причому при використанні цього алгоритму ймовірність помилки буде мінімальною. Принцип дії байєсовського класифікатора складається з наступних кроків:

- 1 Обчислити для об'єкта функцію правдоподібності кожного класу;

2 Обчислити апостеріорні ймовірності;

3 Віднести об'єкт до того класу, апостеріорна ймовірність якого виявилася найвищою.

Якщо під час вирішення завдання невідомі щільності розподілу класів, їх необхідно обчислити по навчальній вибірці. Є безліч методів відновлення густини розподілу, і, внаслідок цього, існує безліч різних байєсівських алгоритмів. Якщо припустити, що ознаки, які описують об'єкт, є незалежними, можна спростити побудову класифікатора. Але, оскільки на практиці ознаки рідко бувають незалежними, це призведе до зменшення точності класифікації об'єктів.

1.2.6 Нейронні мережі

Нейронна мережа є математичною моделлю біологічної нейронної мережі. Вона складається з багатьох простих пристроїв – нейронів, з'єднаних між собою. Кожен нейрон приймає сигнал, перетворює його, а потім посиляє перетворений сигнал іншим нейронам.

Відмінною особливістю нейронних мереж є те, що вони навчаються для вирішення певного завдання. З математичної погляду, під час навчання відбувається завдання коефіцієнтів зв'язків між нейронами. Нейронна мережа здатна у процесі навчання знаходити складні взаємозв'язки між елементами вхідних та вихідних даних. Також, нейронні мережі здатні до узагальнення, тобто видавати правильну відповідь на вхідних даних, які не брали участь у навчанні.

Метод опорних векторів має велику обчислювальну складність, і для його використання бажано мати більший простір ознак, ніж є в даній задачі.

Від початку відсутня статистична інформація про вихідні дані, що ускладнює використання байєсівського класифікатора.

Нейронні мережі здатні вирішувати завдання класифікації, набори даних у яких є лінійно-нероздільними. До того ж нейронні мережі показали здатність до апроксимації довільної функції з будь-якою бажаною точністю [6,7]. Внаслідок цього використання нейронних мереж для вирішення даної задачі є кращим варіантом.

1.3 Обґрунтування вибору способу виявлення переваг розробленої інформаційної технології визначення якості вина

Взагалі, існує декілька відомих програмних додатків, що здатні здійснювати визначення якості вина. Однак, у більшості випадків їх важко порівнювати між собою через те, що вони використовують для навчання класифікаційних алгоритмів різні набори даних. А всім відомо, що від якості підготовки навчального набору даних дуже залежить якість роботи методів та засобів класифікації на основі машинного навчання. Якість підготовки набору даних для навчання визначається тим, наскільки різноманітні дані до нього входять, наскільки повно ці дані представляють все можливе розмаїття випадків класифікації.

Від обсягу набору даних також багато залежить. Треба прагнути створювати набори даних якомога більших розмірів, з рівномірним представництвом різних класів та з всеохопленням різних випадків і нюансів даних для різних представників одного класу. Навіть при використанні одного набору даних, різні дослідники по-різному розбивають їх на навчальну та тестову вибірку. А це також впливає на результат класифікації, тому що ті приклади, що у одних знаходяться у навчальній вибірці, у інших знаходяться у тестовій вибірці і навпаки. Якщо в навчальну вибірку потраплять більш складні та представницькі приклади, а до тестової вибірки потраплять більш прості та типові приклади, то достовірність класифікації буде вищою, ніж у зворотному випадку.

Тому у даній роботі було вирішено в одній програмі промодельовати різні методи машинного навчання для вирішення задачі визначення якості вина., а потім порівняти результати. Причому моделювання цих методів буде відбуватись на одному наборі даних та при однаковому розбитті цього набору даних на навчальну та тестову вибірки. Моделюватись будуть такі методи машинного навчання:

- метод логістичної регресії,
- метод машини опорних векторів,
- метод дерев рішень,
- неймережевий метод на основі багат шарового персептрона.

Головним показником якості програм класифікації є достовірність класифікації, тому порівнювати різні методи машинного навчання будемо у розділі 4 саме за параметром достовірності визначення якості вина.

1.4 Висновок до розділу 1

У розділі описано детальну постановку задачі визначення якості вина. Також розглядаються різні методи розв'язання задачі класифікації і оцінюється їх застосовність до завдання класифікації продукції виноробства. Серед таких методів машинного навчання як метод логістичної регресії, метод машини опорних векторів, метод k-найближчих сусідів, метод дерев рішень, Байєсівський класифікатор та неймережевий метод як найбільш перспективний і застосовний до даної задачі було обрано неймережевий метод. Крім цього, було обгрунтовано вибір способу виявлення переваг розробленої інформаційної технології шляхом паралельного моделювання різних методів визначення якості вина.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НЕЙРОМЕРЕЖЕВОГО ВИЗНАЧЕННЯ ЯКОСТІ ВИНА

2.1 Обґрунтування вибору типу нейронної мережі для визначення якості вина

У пункті 1.2 було показано, що для задачі визначення якості вина варто обрати метод на основі нейронних мереж, так як він має переваги перед іншими методами класифікації. Тепер необхідно визначити який тип нейромережі треба обрати з значної кількості їх різновидів [8,9]. Розглянемо деякі з них та обґрунтуємо вибір тієї, що найбільше підходить для вирішення поставленої задачі.

2.1.1 Одношаровий персептрон. Одношаровий персептрон [8,9] є моделлю системи з навчанням. Він складається з матриці світлочутливих елементів (S-елементів), асоціативних елементів (А-елементів) і реагуючих елементів (R-елементів). Тобто персептрон містить один шар штучних нейронів, що з'єднані ваговими коефіцієнтами з множиною входів. На рис. 2.1 подано структуру одношарового персептрона.

В 60-ті роки ХХ століття персептрони привернули до себе велику увагу. Розенблаттом було доведено теорему про навчання персептрона й тим самим показано, що персептрон можна навчити всьому, що він здатен представляти. Уідроу показав ряд переконливих демонстрацій систем на основі персептронів. Дослідження можливостей цих систем виявили, що персептрони не спроможні навчитися деяким простим завданням. Мінський детально проаналізував цю проблему і довів, що є жорсткі обмеження на те, що можуть виконувати одношарові персептрони, а, отже, і на те, на що вони можуть навчатись [9].

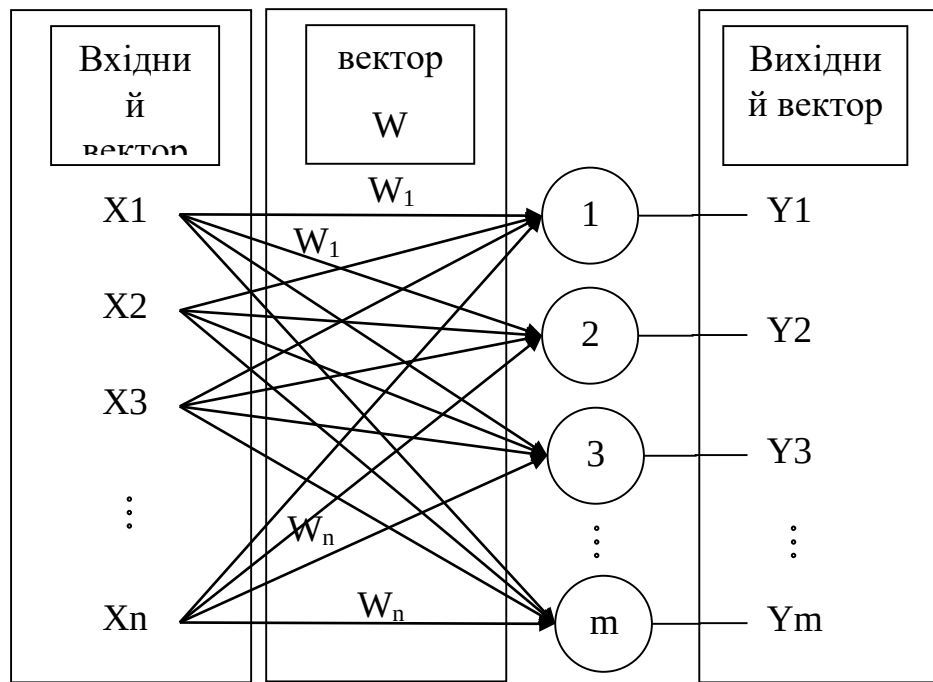


Рисунок 2.1 – Одношарова нейронна мережа

Один із песимістичних результатів Мінського довів, що одношаровий перцептрон не може виконувати таку просту функцію як ВИКЛЮЧНЕ АБО. Це логічна функція від двох змінних, кожна з яких може бути або нулем або одиницею. Є великий клас функцій (поряд з функцією ВИКЛЮЧНЕ АБО), які є лінійно нероздільними, а тому не можуть бути реалізовані одношаровим перцептроном. Лінійна роздільність обмежує одношарові перцептрони у виконанні завдань класифікації. Як показано в [8], імовірність випадково обраної функції виявитися лінійно роздільною, досить мала. Так як лінійна роздільність обмежує можливості одношарових перцептронів, то вони на практиці обмежені у використанні тільки для розв'язку простих завдань.

2.1.2 Багатошаровий перцептрон. Суттєве обмеження функціоналу одношарови нейромереж можна подолати додаванням додаткових шарів. Багатошарові нейромережі можна отримати при каскадному з'єднанні одношарових мереж, де виходи одного шару є входами для наступного

шару, причому така нейромережа може мати збільшену обчислювальну потужність тільки у тому випадку, коли активаційну функцію між шарами обрати нелінійною. Багат шарові нейромережі спроможні здійснювати загальні класифікації, відокремлюючи ті об'єкти, які утримуються в опуклих областях або необмежених областях [10].

Багат шаровий персептрон – це багат шарова нейромережа прямого поширення. Як правило, вона складається з множини сенсорних елементів, які складають вхідний шар, одного чи кількох прихованих шарів нейронів і одного вихідного шару нейронів. Вхідні сигнали поширюються нейромережею у прямому напрямку, від одного шару до іншого [8].

Для навчання багат шарових персептронів часто використовується алгоритм зворотного поширення похибки, який виконується за два проходи по всіх шарах мережі: прямий прохід і зворотний. При прямому проході вхідний образ (вектор) подається на сенсорні нейрони мережі, після чого поширюється на подальші шари. У результаті видається сукупність вихідних сигналів, які є реакцією нейромережі на цей вхідний вектор. При прямому проході всі синаптичні ваги мережі є фіксованими. При зворотному проході всі синаптичні ваги налаштовуються згідно правила корекції помилок, а саме: фактичний вихід мережі віднімається від цільового виходу, в результаті чого отримується сигнал помилки. Цей сигнал помилки далі поширюється нейромережею у зворотному напрямку до синаптичних зв'язків. Ваги синаптичних зв'язків налаштовуються у напрямку максимального наближення вихідного сигналу нейромережі до бажаного [9].

Багат шарові персептрони мають три характерні ознаки:

1. Кожен нейрон мережі має функцію активації, яка є нелінійною. Така нелінійна функція є гладкою (тобто диференційованою всюди), на відміну від «жорсткої сходинок», що використовується в персептроні

Розенблатта. Найпоширенішою функцією, що задовільняє цим вимогам, є сигмоїдальна:

$$y_i = \frac{1}{1 + \exp(-v_j)}, \quad (2.1)$$

де v_j – зважена сума всіх синаптичних входів мінус порогове значення нейрона j ; y_j – вихідний сигнал нейрона.

2. Нейромережа має один чи кілька прихованих шарів нейронів. Ці нейрони послідовно «витягають» найважливіші ознаки вхідного образу та дають можливість нейромережі навчатися розв'язувати складні завдання.

3. Нейромережа має високий ступінь зв'язності завдяки синаптичним зв'язкам. Зміна рівня зв'язності відбувається при зміні множини синаптичних зв'язків або їхніх вагових коефіцієнтів.

Комбінація цих властивостей у поєднанні зі здатністю до навчання на власному досвіді забезпечує високу обчислювальну потужність багат шарового персептрону. На рис. 2.2 зображено архітектуру багат шарового персептрона, що містить два прихованих шари та один вихідний шар. Ця нейромережа є повнозв'язною, тобто кожен нейрон у будь-якому шарі нейромережі зв'язаний з усіма нейронами попереднього шару.

Тришарова мережа є більш загальним випадком. Її можливості класифікації обмежуються лише кількістю нейронів і ваг. Відсутні обмеження на опуклість. Тепер нейрон третього шару сприймає на вхід набір опуклих багатокутників, а їх логічна комбінація може бути неопуклою. При додаванні нейронів і синаптичних ваг кількість сторін багатокутника може зростати необмежено. Це дозволяє апроксимувати область довільної форми з бажаною точністю. Крім цього, не всі вихідні області другого шару мають перетинатись. Таким чином, можливо поєднувати області різного типу, опуклі та неопуклі, виводячи на вихіді одиницю щоразу, коли вхідний вектор належить одній з цих областей.

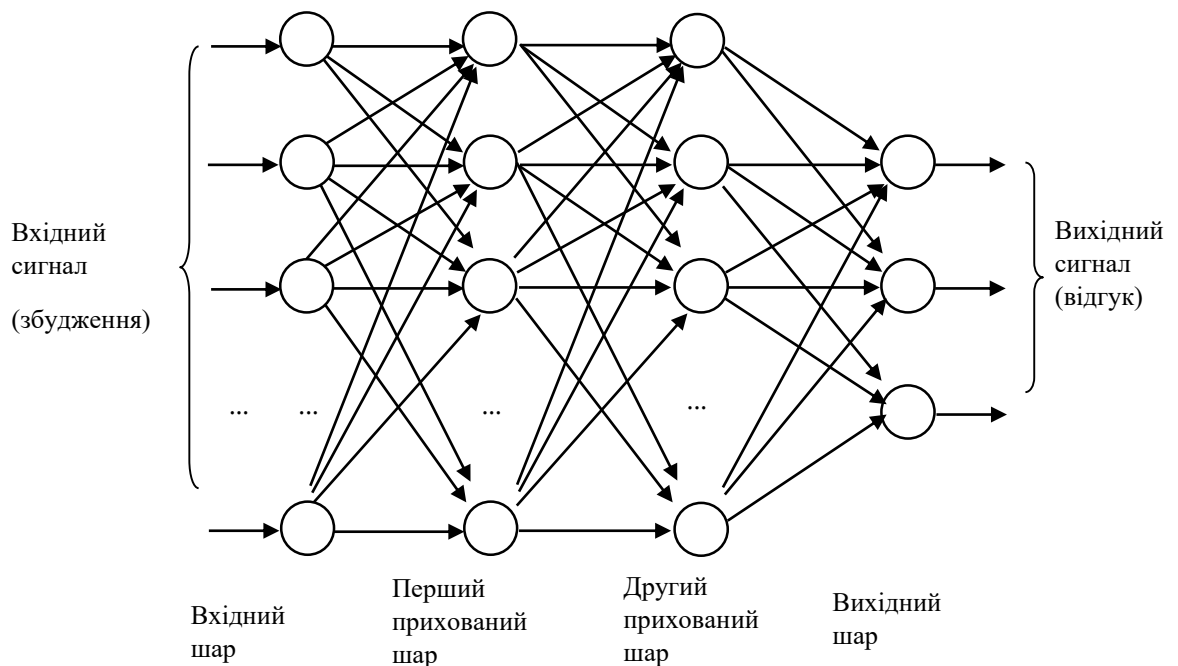


Рисунок 2.2 - Архітектура багатошарового персептрона
з двома прихованими шарами

Навчання багатошарового персептрона за алгоритмом зворотного поширення помилки є популярним з кількох причин:

- метод змін синаптичних ваг і порогів у багатошаровому персептроні є локальним.
- метод обчислення всіх окремих похідних функції вартості за вільними параметрами є ефективним.

2.1.3 Нейронні мережі на основі радіальних базисних функцій.

Базова архітектура нейромережі RBF передбачає наявність трьох шарів нейронів, які виконують різні функції (рис. 2.3).

Вхідний шар складено із сенсорних елементів, що пов'язують нейромережу із зовнішнім середовищем. Другий шар є єдиним прихованим шаром, що часто має значно більшу кількість нейронів, ніж вхідний шар.

Кожний нейрон прихованого шару використовує функцію активації Гауса (2.2).

$$h(\vec{x}) = \exp\left(-\frac{\|\vec{x} - \vec{c}\|^2}{r^2}\right) \quad (2.2)$$

де $\vec{x}$ – вхідний вектор, $\vec{c}$ – центр функції Гаусса, r – радіус функції Гаусса.

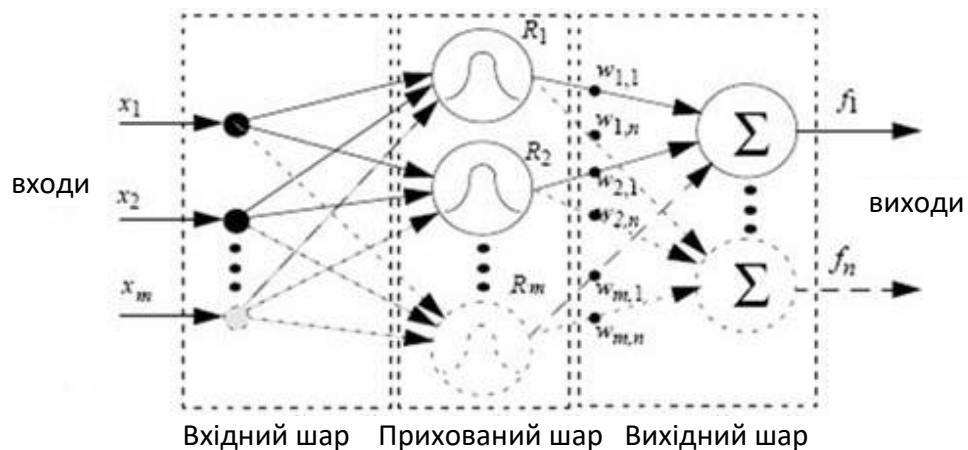


Рисунок 2.3 - Нейромережа на основі радіальних базисних функцій

У центрі радіальної базисної функції (функції ядра) ваговим вектором визначена точка, пов'язана із нейроном. Координати та ширина функції ядра мають бути визначені при навчанні на тренувальній вибірці. Як правило, число ядер є набагато меншим, ніж число навчальних прикладів.

Кожен вихідний нейрон обчислює лінійну комбінацію цих радіально-базисних функцій за (2.3).

$$f(\vec{x}) = \sum_{j=1}^m w_j h_j(\vec{x}) \quad (2.3)$$

З точки зору апроксимації приховані нейрони формують сукупність функцій, які утворюють базисну систему для представлення вхідних сигналів у побудованому на цій системі просторі.

Архітектура RBF ґрунтується на висновках із теореми Ковера про роздільність образів, відповідно до яких нелінійне перетворення складного завдання класифікації образів у простір більшої розмірності збільшує імовірність лінійної роздільності образів. Крім цього, розмірність прихованого шару пов'язана із можливістю нейромережі апроксимувати «гладке» відображення вхід–вихід. Чим більша розмірність прихованого шару, тим більша точність апроксимації [9,10].

Відомі різні алгоритми навчання нейромереж RBF. Базовий алгоритм використовує двокрокову стратегію навчання (або змішане навчання). Алгоритм «оцінює» позицію та ширину ядра з використанням методу кластеризації без вчителя, а потім застосовується алгоритм мінімізації середньоквадратичної похибки з учителем для визначення ваг між прихованим і вихідним шарами нейронів. Оскільки вихідні нейрони є лінійними, то застосовується неітераційний алгоритм. Після отримання цього початкового наближення застосовується алгоритм градієнтного спуску для уточнення параметрів нейромережі [9,11].

Такий змішаний алгоритм навчання нейромережі RBF сходиться набагато швидше, ніж алгоритм зворотного поширення помилки для навчання багат шарових персептронів, однак RBF-нейромережа часто містить дуже велике число прихованих елементів, а отже, функціонування RBF-нейромережі в такому випадку буде повільнішим за функціонування багат шарового персептрона. Через це ефективність (похибка у залежності від розміру нейромережі) мереж RBF та багат шарового персептрона залежить від виду розв'язуваної задачі [10].

Якщо порівнювати між собою багат шаровий персептрон та нейромережу радіальних базисних функцій, то можна зробити висновок, що багат шаровий персептрон виконує глобальну апроксимацію нелінійного відображення, а RBF-нейромережа за допомогою

локалізованих нелінійностей, які експоненційно зменшуються (функції Гауса) забезпечує локальну апроксимацію нелінійного відображення.

Багатошаровий персептрон (або багатошарова нейромережа прямого поширення) є більш прийнятним для даної задачі побудови інформаційної технології визначення якості вина. Процес визначення якості вина включає в себе процедуру попереднього навчання нейромережі, і результат буде максимально точним. Так як на входи подаються не складні образи, а ознаки, що визначаються параметрами хімічного складу вина, то класифікувати їх буде легше, що призведе до збільшення достовірності класифікації.

2.2 Архітектура нейронної мережі багатошаровий персептрон для інформаційної технології визначення якості вина

Штучна нейронна мережа – це математична модель, принцип роботи якої імітує роботу мережі біологічних нейронів.

Нейронні мережі набувають заслуженої популярності останнім часом все більше і більше. Одним із найпростіших типів нейронних мереж є персептрон. У цій роботі використовується реалізація багатошарового персептрона на Python з використанням надбудови для глибокого навчання Keras.

Розглянемо що таке багатошаровий персептрон і як він навчається.

Як відомо, найпростішим прикладом нейронної мережі є багатошаровий (в окремому випадку, одношаровий) персептрон. Розглянемо, що собою являє така модель.

Багатошаровий персептрон (рис. 2.2) складається з:

- вхідного вектора (вхідні дані),
- вихідного вектора (вихідні дані),
- вектора(ів) проміжного представлення (прихований(і) шар).

Елементи векторів прийнято ще називати нейронами. Обчислення в такій моделі поширюються від входу до виходу. Зв'язкам між нейронами на різних шарах відповідають деякі ваги. Тому така мережа є повнозв'язною.

У i -тий нейрон першого прихованого шару (рис. 2.4) надходять вхідні значення, наприклад, у нашому випадку 11 параметрів хімічного складу вина: $x_1, x_2, \dots, x_{11}$ (елементи вхідного вектора) з відповідними вагами $w_{1,i}, w_{2,i}, \dots, w_{11,i}$.

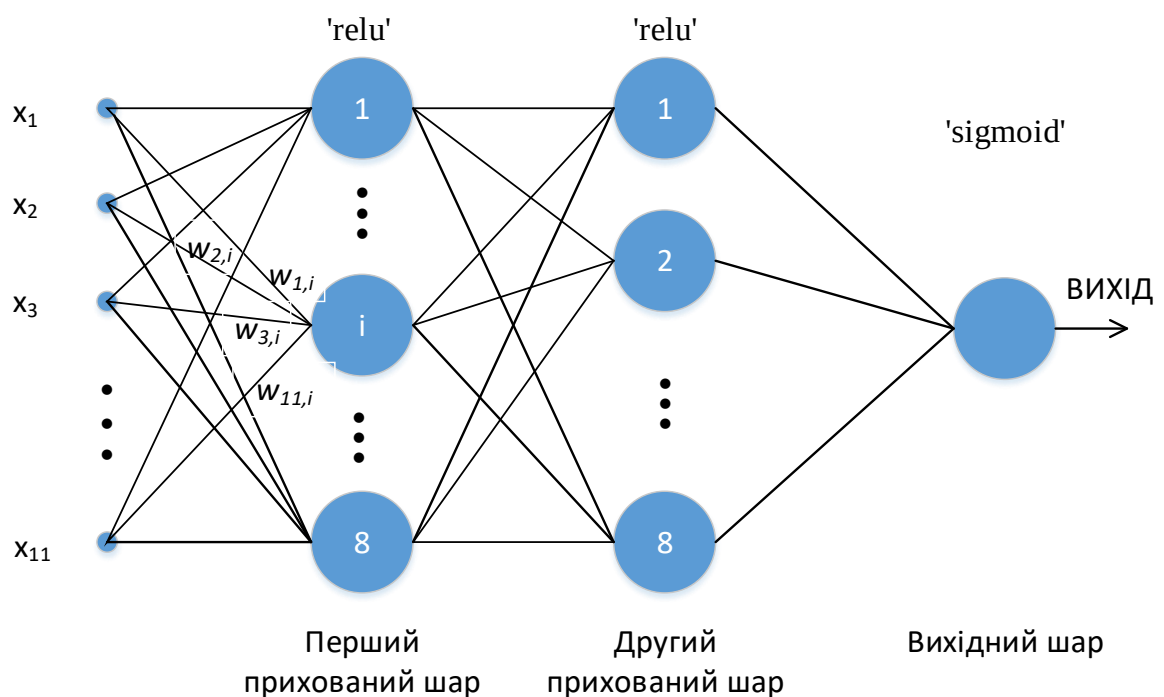


Рисунок 2.4 – Структура багатошарового персептрона для інформаційної технології визначення якості вина

Далі, всередині нейрона відбувається обчислення двох операцій, а саме, композиції лінійної та нелінійної функції:

– спочатку ми обчислюємо зважену суму вхідних значень і додаємо деякий параметр зміщення b ,

– далі, від отриманих на попередньому кроці значень, беремо нелінійну функцію f , яку прийнято називати функцією активації.

Таким чином, вихідне значення h з одного нейрона обчислимо за формулою:

$$(h_i = f(\sum_{j=1\dots 11} w_{j,i} x_j + b_i)) \quad (2.4)$$

Разом ці нейрони утворюють велику нейронну мережу.

Звернемо увагу на те, що одному шару в нейронній мережі відповідає вже ціла матриця ваг $\mathbf{W}_1$ і деякий вектор зміщень $\mathbf{B}$. Тому обчислення в одному шарі персептрона можна представити у вигляді композиції матричного множення, додавання вектора зміщень і поелементного взяття нелінійної функції

$$\mathbf{H} = f(\mathbf{W}_1 \cdot \mathbf{X} + \mathbf{B}_1), \quad (2.5)$$

де

$$\mathbf{H} = \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{pmatrix}, \mathbf{W}_1 = \begin{pmatrix} w_{11}^1 & w_{12}^1 & w_{13}^1 \\ w_{21}^1 & w_{22}^1 & w_{23}^1 \\ w_{31}^1 & w_{32}^1 & w_{33}^1 \\ w_{41}^1 & w_{42}^1 & w_{43}^1 \end{pmatrix}, \mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \mathbf{B}_1 = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}, \quad (2.6)$$

та

$$h_i = f\left(\sum_j w_{ij}^1 x_j + b_i\right) \quad (2.7)$$

Якщо ми візьмемо більше шарів, то отримаємо багат шарову нейронну мережу або багат шаровий персептрон, де вихід кожного попереднього шару є вхідними даними для наступного шару. Обчислення в такій мережі можна представити у вигляді такого рекурентного співвідношення:

$$X_k = f(W_k \cdot X_{k-1} + B_k), X_n = G_w(X_0), \quad (2.8)$$

де G_w - функція, що залежить від параметрів моделі (від ваг, що зв'язують нейрони і зміщення). Чим більше шарів у нашій нейронній мережі, тим складніше проміжне представлення вона має і тим складніші залежності між вхідними та вихідними даними вона здатна описати.

Яку функцію активації використовувати? Основна властивість функції активації - її нелінійність. Якби ми використовували лінійну функцію, то, по-перше, ми могли б розв'язувати тільки вузький клас задач, де залежність між вхідними і вихідними даними описується лінійною функцією, а, по-друге, збільшення числа прихованих шарів не підвищувало б ефективність нашої моделі, оскільки композиція лінійних функцій - це все ще лінійна функція.

Завдання функції активації - допомогти прийняти локальне рішення в кожному з нейронів. Наприклад, функція типу Sigmoid відображає значення на виході з нейрона у щось більше або менше нуля. Функція ReLU зануляє значення менші за нуль, а значення більші за нуль, залишає як є. У цієї функції дуже проста похідна, а саме похідна цієї функції братиме участь у процесі навчання і в алгоритмі зворотного поширення помилки. У структурі багатошарового персептрона для нашої задачі було використано функцію активації ReLU у обох прихованих шарах та функцію активації Sigmoid у вихідному шарі.

Як навчити нейронну мережу? Нейронні мережі навчаються за допомогою так званого навчання з учителем. Для цього необхідна навчальна вибірка – розмічені дані, які складаються з пар «вхідний об'єкт - вихідний об'єкт». Ми подаємо цю навчальну вибірку в процесі навчання, який полягає в тому, щоб знайти такі параметри (ваги) моделі W , щоб наша нейронна мережа передбачала правильно ті відповіді, які ми вже знаємо.

Таким чином, процес навчання зводиться до розв'язання задачі мінімізації або оптимізації функції помилки на тих прикладах, що є в нашій навчальній вибірці.

Функцію помилки можна записувати по-різному. Наприклад,

$$E = \|D - G_w(Z)\|^2, W^* = \operatorname{argmin}_w (E(W)), \quad (2.9)$$

де $(D - G_w(Z))$ є різницею між правильною відповіддю D і передбаченням мережі $G_w(Z)$. Від цієї різниці ми беремо норму, яку хочемо мінімізувати по всій навчальній вибірці. Потім ми шукаємо такі параметри моделі W^* які мінімізують цю помилку. Спочатку ваги моделі ініціалізуються випадковими значеннями.

Можна застосувати добре відомий у теорії оптимізації метод градієнтного спуску. Уявімо, що в нас є деяка функція, і ми хочемо знайти її мінімум. Ми починаємо з деякої точки і нам необхідно зрозуміти, куди нам потрібно рухатися, щоб наблизитися до мінімуму. Є вектор, який називається градієнт. І цей вектор спрямований у бік зростання функції. Тому антиградієнт – це градієнт зі знаком мінус спрямований у бік зменшення функції. Цей алгоритм пропонує рухатися в бік антиградієнта і таким чином наближатися до локального мінімуму.

У разі нейронних мереж використовується невелика модифікація – стохастичний градієнтний спуск. Відмінність від попереднього градієнтного спуску в тому, що ми не обчислюємо градієнт відразу на всіх зразках нашої вибірки, а тільки на одному зразку за одну ітерацію або на групі зразків (міні-батч).

Застосування багатошарового персептрона до задачі класифікації. Розглянемо приклад використання двошарового персептрона для задачі класифікації (рис.2.5).

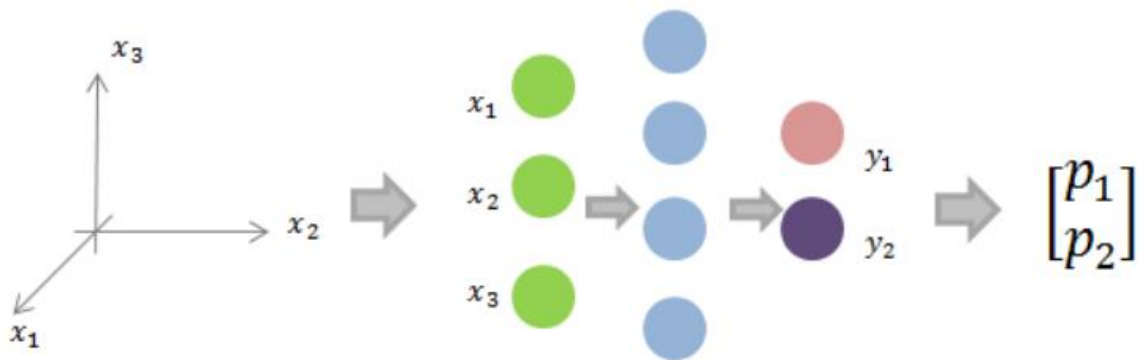


Рисунок 2.5 – Приклад використання двошарового персептрона для задачі класифікації

Припустимо, у нас є об'єкти, задані в тривимірному ознаковому просторі. Це означає, що кожен об'єкт подається вектором із трьох компонент. Побудуємо нейронну мережу, у якій є три вхідні нейрони, якраз пропорційно нашому вхідному об'єкту. І припустимо, що розв'язується задача бінарної класифікації, в якій необхідно визначити, до якого класу (рожевого чи фіолетового) належить наш об'єкт. Тому на виході ми маємо два вихідні нейрони. Припустимо, що ми вже навчили мережу. Розглянемо, яким чином її тепер використовувати?

Подаємо на вхід наш тривимірний вектор, що характеризує об'єкт, виконуємо пряме поширення згідно з формулами, описаними вище, і на виході отримуємо два числа. Ці числа вже відповідають за приналежність вхідного об'єкта до того чи іншого класу, але після спеціального перетворення ми отримуємо інші два числа p_1 і p_2 . Ці числа явно характеризують імовірність належності об'єкта до одного або другого класу. Таким чином, виходом із нейронної мережі у випадку задачі класифікації є розподіл імовірностей належності до того чи іншого класу.

Структура багатошарового персептрона для інформаційної технології визначення якості вина представлена на рис. 2.4 і містить 11 входів $x_1 \dots x_{11}$

по кількості параметрів хімічного складу вин із набору даних. Також вона містить 2 прихованих шари по 8 нейронів і 1 вихідний шар із одного нейрона. Якщо досліджуване вино відноситься до класу вин низької якості, то на виході буде число, близьке до 0, а якщо досліджуване вино відноситься до класу вин високої якості, то на виході буде число, близьке до 1.

Для навчання цієї нейромережі будемо використовувати не звичайний стохастичний градієнтний спуск, а його модифікацію ADAM – Adaptive Moment Estimation [12].

Адам можна розглядати як комбінацію RMSprop та Stochastic Gradient Descent з імпульсом. Він використовує квадратні градієнти для масштабування швидкості навчання, наприклад RMSprop, і використовує перевагу імпульсу, використовуючи середнє ковзне значення градієнта, а не сам градієнт, такий як SGD з імпульсом.

Adam – це метод адаптивної швидкості навчання, тобто він розраховує індивідуальні швидкості навчання для різних параметрів. Його назва походить від АДАптивного Моменту. І причина, за якою це так називається, полягає в тому, що Адам використовує оцінки першого та другого моментів градієнта, щоб адаптувати швидкість навчання для кожної ваги нейронної мережі. А n -й момент випадкової величини визначається як очікуване значення цієї змінної в ступені n .

2.3 Розробка алгоритму роботи інформаційної технології визначення якості вина

Відповідно до мети роботи та постановки задачі було розроблено алгоритм роботи інформаційної технології нейромережевого визначення якості вина, представлений на рис. 2.6.

Першим кроком в алгоритмі роботи інформаційної технології нейромережевого визначення якості вина буде завантаження набору даних (блок 1) із параметрами вин та їх показниками якості:

Далі переходимо до візуалізації результатів аналізу набору даних параметрів вин. Спочатку проводиться двофакторний аналіз (блок 2), який дозволить визначити ступінь впливу різних параметрів вина на його якість. Результати двофакторного аналізу у винах різної якості показано на рисунках 4.1-4.2 у 4 розділі.

Потім проводиться попередня обробка даних: видалення непотрібних стовпців із набору даних (блок 3). Видаляються стовпці з тими параметрами, які дуже мало пов'язані із залежною змінною якість. Тож навіть якщо ми видалимо ці стовпці, точність не постраждає.

Далі перетворюємо багатокласову оцінку якості вина на двокласову (блок 4) – вино добре або вино погане. Вина з оцінками від 2 до 6 будемо вважати поганими, а вина з оцінками від 7 до 8 будемо вважати добрими.

Потім аналізуємо кількість добрих і поганих вин у наборі даних (блок 5). Далі ділимо набір даних на залежні та незалежні змінні:

Набір даних розділяємо на навчальну (75%) та тестову (25%) вибірки (блок 6). Визначаємо обсяг навчальної (1199 записів) та тестової (400 записів) вибірок. Виконуємо стандартне масштабування (нормування) параметрів (блок 7).

Далі переходимо до вирішення задачі різними методами.

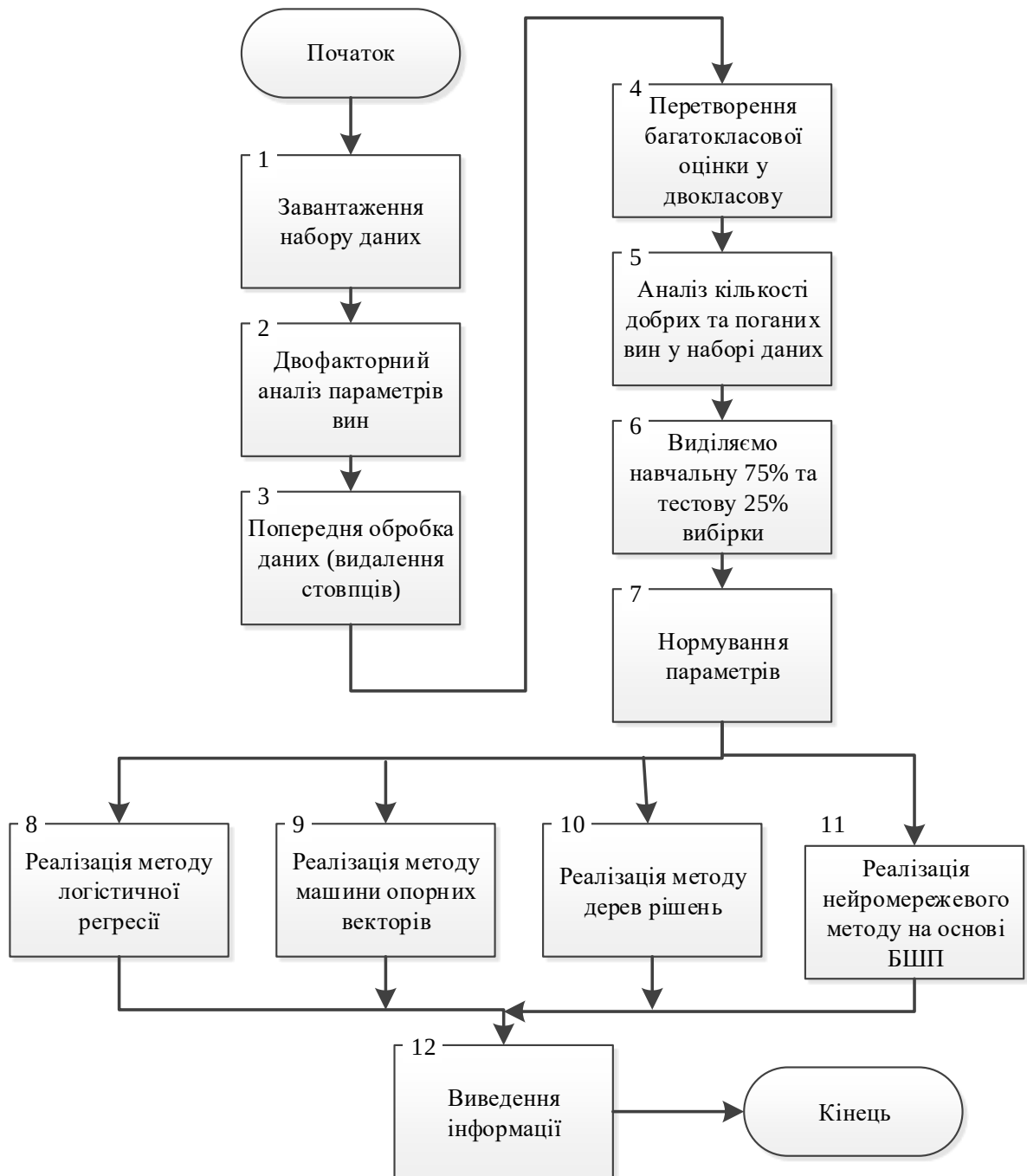


Рисунок 2.6 – Схема алгоритму роботи інформаційної технології нейромережевого визначення якості вина

Спочатку визначаємо якість вина методом логістичної регресії (Logistic Regression) - (блок 8).

Потім визначимо якість вина методом машини опорних векторів (Support Vector Machine) - (блок 9).

Далі визначимо якість вина методом дерев рішень (DecisionTree) - (блок 10).

І, нарешті, визначимо якість вина нейромережевим методом з використанням багатошарового перцептрона (Multi Layer Perceptron) - блок 11. Після цього виводиться інформація по достовірності визначення якості вина кожним з методів та похибки на навчальній та тестовій вибірках (блок 12).

UML-діаграма класів програми нейромережевого визначення якості вина представлена на рис.2.7.

З отриманої UML-діаграми класів програми нейромережевого визначення якості вина по рис. 2.7 видно, що вона містить головний клас (Main Class), який керує такими класами: Model Trainer для навчання нейронної мережі, Data Processor для обробки та аналізу даних та Data Vizualizer для візуалізації результатів та побудови діаграм.

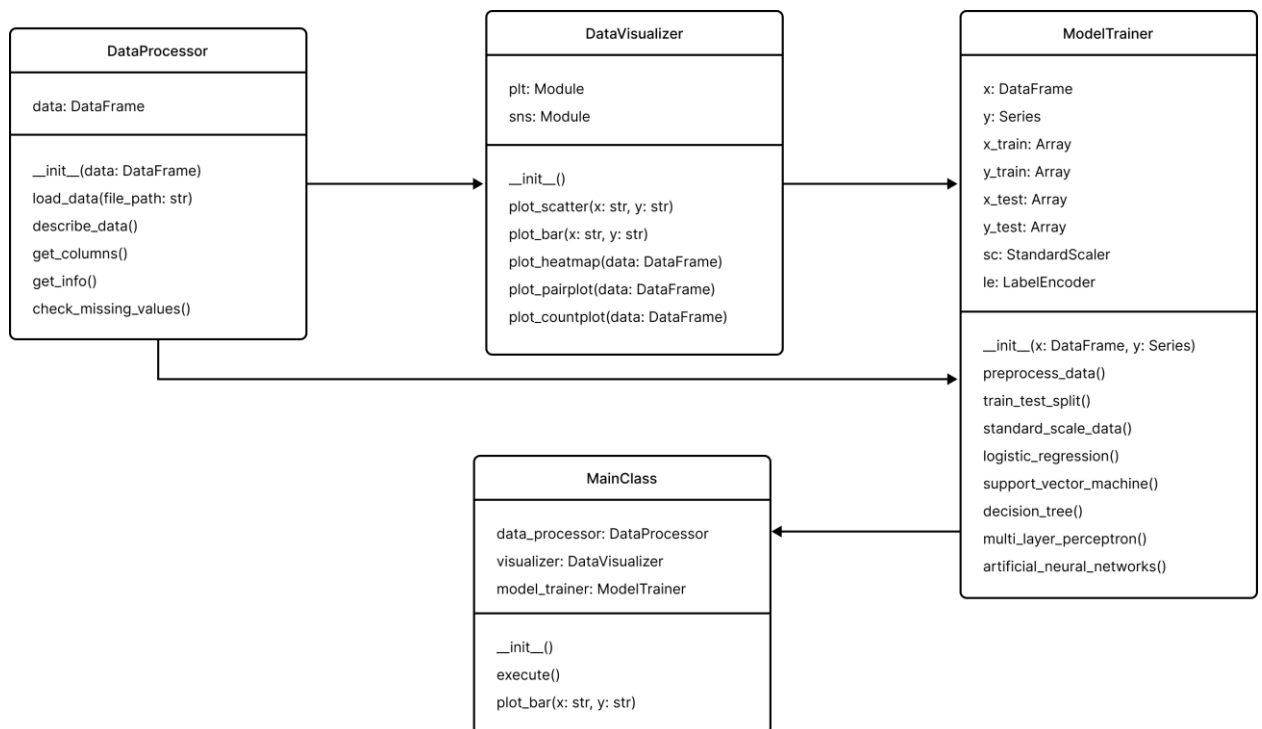


Рисунок 2.7 – UML-діаграма класів програми нейромережевого визначення якості вина

2.4 Висновок до розділу 2

У розділі було обґрунтовано вибір типу нейронної мережі для інформаційної технології визначення якості вина. Із трьох розглянутих нейромереж: одношаровий персептрон, багатошаровий персептрон та RBF-нейромережа було обрано архітектуру нейронної мережі багатошаровий персептрон. Розроблено архітектуру нейронної мережі багатошаровий персептрон для інформаційної технології визначення якості вина, яка має 11 входів, 2 прихованих шари по 8 нейронів та вихідний шар із одного нейрона. У прихованих шарах обрано функцію активації ReLU та функцію активації Sigmoid у вихідному шарі. Для навчання цієї нейромережі використовується модифікація ADAM метод стохастичного градієнтного спуска. Розроблено алгоритм роботи програми нейромережевого визначення якості вина та UML діаграму класів.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НЕЙРОМЕРЕЖЕВОГО ВИЗНАЧЕННЯ ЯКОСТІ ВИНА

3.1 Обґрунтування вибору мови програмування та спеціалізованих бібліотек

Для програмної реалізації інформаційної технології нейромережевого визначення якості вина обрано мову Python [13].

Python — інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання наявних компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

Отже, Python - це високорівнева мова програмування, яка привертає розробників своєю простотою, елегантністю та читабельністю. Завдяки своєму простому синтаксису, Python є прекрасним вибором для початківців, але він також надає потужні функції, які задовольняють потреби досвідчених розробників.

Основні переваги Python:

- Читабельний код: Синтаксис Python легко розуміти і легко читати. Код на Python виглядає природно і майже як псевдокод, що спрощує розробку, зрозумілість та співпрацю між розробниками.

- Легкість вивчення: Python має простий вступний поріг інтеграції. Новачки швидко можуть освоїти основи і почати писати функціональний код. Багато навчальних матеріалів і ресурсів доступні для допомоги початківцям.

- Розширюваність: Python дозволяє розширювати свої можливості за допомогою зовнішніх бібліотек та модулів, що значно розширює функціональність мови. Python має велику кількість сторонніх пакетів, які допомагають вирішувати різноманітні задачі.

- Кросплатформеність: Python підтримується на різних операційних системах, таких як Windows, macOS та Linux. Це дозволяє розробникам створювати програми, які працюють на різних платформах без необхідності значних змін в коді.

- Багатофункціональність: Python підтримує різні парадигми програмування, включаючи об'єктно-орієнтоване програмування, процедурне програмування та функціональне програмування. Це дає розробникам можливість вибрати найкращий підхід для вирішення конкретних завдань.

- Активна спільнота: Python має велику та активну спільноту розробників, яка внесла значний вклад у розширення мови та створення корисних інструментів. Це означає, що завжди є допомога, підтримка та велика кількість ресурсів для розробників Python.

Python використовується для різноманітних задач, включаючи веб-розробку, наукові дослідження, аналіз даних, штучний інтелект, робототехніку та багато іншого. Він є потужним і гнучким інструментом для розробників, які шукають простоту в поєднанні з функціональністю.

Оскільки нейронні мережі пов'язані з перемноженням векторів та матриць, то для цієї роботи знадобиться бібліотека NumPy [14]. NumPy

(скорочення від Numerical Python) — це бібліотека з відкритим кодом для мови програмування Python. Вона має такі можливості:

- підтримка багатомірних масивів (включно з матрицями);
- підтримка високорівневих математичних функцій, що призначені для роботи з багатомірними масивами.

Математичні алгоритми, що реалізовані інтерпретованими мовами (наприклад, Python), зазвичай працюють набагато повільніше за ті алгоритми, що реалізовані компільованими мовами (напр., Фортран, Сі, Java). Бібліотека NumPy надає реалізації обчислювальних алгоритмів (у вигляді функцій та операторів), оптимізованих для роботи з багатовимірними масивами. В результаті будь-який алгоритм, який може бути виражений у вигляді послідовності операцій над масивами (матрицями) та реалізований з використанням NumPy, працює так само швидко, як еквівалентний код, що виконується у MATLAB.

Оскільки при проектуванні програми нейромережевого визначення якості вина ми використовуємо набір даних параметрів вин для навчання та тестування нейронної мережі, то нам знадобиться бібліотека Pandas.

Pandas — програмна бібліотека [15], написана для мови програмування Python для маніпулювання даними та їхнього аналізу. Вона, зокрема, пропонує структури даних та операції для маніпулювання чисельними таблицями та часовими рядами. Pandas є вільним програмним забезпеченням, що випускається за трипунктовою ліцензією BSD. Ця назва походить від терміну «панельні дані» (англ. panel data), який в економетрії позначає багатовимірні структуровані набори даних.

Можливості бібліотеки:

- Інструменти для зчитування та записування даних між структурами даних у пам'яті та різними форматами файлів.
- Об'єкт DataFrame із вбудованим індексуванням для маніпулювання даними.

- Вирівнювання даних та вбудована підтримка пропущених даних.
- Отримання зрізів за мітками, індексування з розширеними можливостями та отримання піднаборів з великих наборів даних.
- Переформатовування для отримання зведених наборів даних.
- Вставляння та вилучення стовпчиків у структурах даних.
- Злиття та з'єднання наборів даних.
- Рушій групування, що дозволяє робити з наборами даних операції розділення-зміни-об'єднання (англ. split-apply-combine).
- Функціональність для часових рядів: породження діапазонів дат та перетворення частоти, статистики рухливого вікна, лінійні регресії рухливого вікна, зсування дат та запізнювання.
- Ієрархічне індексування осей для роботи з даними високої вимірності в структурі даних нижчої вимірності.

Ця бібліотека значно оптимізована за продуктивністю, критичні ланцюжки коду написано мовами програмування Cython та C.

Так як при проектуванні програми нейромережевого визначення якості вина ми виконуємо побудову різних діаграм та графіків, то нам знадобиться бібліотека Matplotlib [16].

Matplotlib — це комплексна бібліотека для створення статичних, анімованих та інтерактивних візуалізацій на Python. Matplotlib має такі функціональні можливості:

- Створення якісних сюжетів для публікації.
- Створення інтерактивних фігур, які можна масштабувати, панорамувати, оновлювати.
- Налаштування візуального стилю і макету.
- Експорт у багато форматів файлів.
- Інтегрування в JupyterLab і графічний інтерфейс користувача.
- Використання багатого набору сторонніх пакетів, побудованих на Matplotlib.

Matplotlib створює цифрової якості публікації в різноманітних форматах друкованих копій та інтерактивних середовищах на різних платформах. Matplotlib можна використовувати в сценаріях Python, оболонках Python/IPython, серверах веб-додатків і різноманітних наборах інструментів графічного інтерфейсу користувача.

3.2 Опис набору даних параметрів вина для навчання та тестування нейромережі

Набір даних охоплює вина, створені в Португалії в провінції Мінью з травня 2004 року по лютий 2007 року. Дані були надані дослідниками Paulo Cortez, Antonio Cerdeira, Fernando Almeida, Telmo Matos, Jose Reis у статті «Modeling wine preferences by data mining from physicochemical properties» [17].

Фрагмент записів набору даних показано на рис. 3.1.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	Sulfate	Alcohol	Quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Рисунок 3.1 – Фрагмент набору даних параметрів вина для навчання та тестування нейромережі

Набір даних містить параметри як червоного вина, так і білого вина. Набір даних червоного вина містить 1599 екземплярів, а набір даних білого вина містить 4898 екземплярів. Обидва набори даних мають 11 вхідних змінних (на основі фізико-хімічних тестів): фіксовану кислотність, летючу кислотність, лимонну кислоту, залишковий цукор, хлориди, вільний діоксид сірки, загальний діоксид сірки, щільність, pH, сульфати, алкоголь та 1 вихідну змінну (на основі сенсорних даних дегустаторів): якість. Сенсорні дані оцінюються за 11 класами якості від 0 до 10 (0 - дуже погано, 10 - дуже добре). У таблиці 3.1 подано опис атрибутів.

Таблиця 3.1 - Опис атрибутів вин.

Атрибути	Переклад	Діапазон зміни
Fixed acidity	Фіксована кислотність	3.8 ... 15.9
Volatile acidity	Летюча кислотність	0.1 ... 1.6
Citric acid	Лимонна кислота	0.0 ... 1.7
Residual sugar	Залишковий цукор	0.6 ... 65.8
Chlorides	Хлориди	0.01 ... 0.61
Free sulfur dioxide	Вільний діоксид сірки	1 ... 289
Total sulfur dioxide	Загальний діоксид сірки	6 ... 440
Density	Щільність	0.987 ... 1.039
pH	Кислотність pH	2.7 ... 4.0
Sulfates	Сульфати	0.2 ... 2.0
Alcohol	Алкогіль	8.0 ... 14.9

Quality (the output target)	Якість (вихідний параметр)	0 ... 10
-----------------------------	----------------------------	----------

Для ефективної класифікації важливо правильно провести вибір ознак. Вибір ознак – це метод відбору найкращої підмножини ознак, які будуть використовуватися для класифікації (Fauzi et al., 2017). У цій роботі для кращого розуміння параметрів вина і вивчення кореляції між ними будемо застосовувати коефіцієнт кореляції Пірсона, який розраховується для кожного атрибута у таблиці 3.1. Він показує парний коефіцієнт кореляції P , який розраховується за допомогою наведеної нижче формули (Dastmard, 2013).

$$P_{x,y} = \frac{\text{cov}(X,Y)}{\sigma_X, \sigma_Y} \quad (3.1)$$

де σ — стандартне відхилення ознак X і Y , cov — коваріація. Діапазон зміни коефіцієнта кореляції від -1 до 1. Значення коефіцієнта 1 передбачає лінійне рівняння, яке описує сильну позитивну кореляцію між X і Y , що означає, що всі точки даних лежать на лінії для збільшення Y із зростанням X . Значення коефіцієнта -1 вказує на сильну негативну кореляцію між точками даних. Усі точки даних лежать на лінії, на якій Y зменшується зі збільшенням X . А значення коефіцієнта 0 вказує на те, що між X і Y немає кореляції [18].

3.3 Програмна реалізація інформаційної технології визначення якості вина на основі нейромережі

Першим кроком в програмі нейромережевого визначення якості вина буде завантаження бібліотек:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Далі потрібно завантажити набір даних із параметрами вин та їх показниками якості:

```
data = pd.read_csv('drive/My Drive/Projects/wine
quality/winequality-red.csv')
```

```
data.head()
data.columns
data.describe()
```

```
data['quality'].value_counts()
```

Далі переходимо до візуалізації результатів аналізу набору даних параметрів вин.

Спочатку проведемо двофакторний аналіз. Наприклад, перевірка варіації фіксованої кислотності у винах різної якості.

```
# checking the variation of fixed acidity in the different
qualities of wine

plt.scatter(data['quality'], data['fixed acidity'], color =
'green')

plt.title('relation of fixed acidity with wine')

plt.xlabel('quality')

plt.ylabel('fixed acidity')

plt.legend()

plt.show()
```

Далі проведемо двофакторний аналіз варіації алкоголю у винах різної якості.

```
# checking the variation of alcohol in the different qualities of wine
```

```
plt.bar(data['quality'], data['alcohol'], color = 'maroon')
plt.title('relation of alcohol with wine')
plt.xlabel('quality')
plt.ylabel('alcohol')
plt.legend()
plt.show()
```

```
# Composition of citric acid go higher as we go higher in the quality of the wine
```

```
import seaborn as sns
```

```
fig = plt.figure(figsize = (10,6))
```

```
sns.barplot(x = 'quality', y = 'citric acid', data = data)
```

```
fig = plt.figure(figsize = (10,6))
```

```
sns.barplot(x = 'quality', y = 'residual sugar', data = data)
```

```
#Composition of chloride go down as we go higher in the quality of the wine
```

```
fig = plt.figure(figsize = (10,6))
```

```
sns.barplot(x = 'quality', y = 'chlorides', data = data)
```

```
fig = plt.figure(figsize = (10,6))
```

```
sns.barplot(x = 'quality', y = 'free sulfur dioxide', data = data)
```

#Sulphates level goes higher with the quality of wine

```
fig = plt.figure(figsize = (10,6))

sns.barplot(x = 'quality', y = 'sulphates', data = data)

f, ax = plt.subplots(figsize=(10, 8))

corr = data.corr()

sns.heatmap(corr, mask=np.zeros_like(corr, dtype=np.bool),
            cmap=sns.diverging_palette(220, 10, as_cmap=True),
            square=True, ax=ax)

sns.pairplot(data)
```

Далі проведемо попередню обробку даних: видалення непотрібних стовпців із набору даних. В процесі роботи програми (двофакторного аналізу) ми побачили, що летюча кислотність, загальний діоксид сірки, хлориди, щільність дуже мало пов'язані із залежною змінною якість. Тож навіть якщо ми видалимо ці стовпці, точність не постраждає.

```
#data = data.drop(['volatile acidity', 'total sulfur dioxide', 'chlorides',
'density'], axis = 1)
```

Перетворимо багатокласову оцінку якості вина на двокласову – вино добре або вино погане. Вина з оцінками від 2 до 6 будемо вважати поганими, а вина з оцінками від 7 до 8 будемо вважати добрими.

```
#names = ['bad', 'good']

#bins = (2, 6.5, 8)
```

```
#data['quality'] = pd.cut(data['quality'], bins = bins, labels
= names)

data['quality'] = data['quality'].map({3 : 'bad', 4 : 'bad', 5:
'bad',

6: 'good', 7: 'good', 8:
'good'})
```

Проаналізуємо кількість добрих і поганих вин у наборі даних

```
# analyzing the different values present in the dependent
variable(quality column)

data['quality'].value_counts()

good      855
bad       744
Name: quality, dtype: int64

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

data['quality'] = le.fit_transform(data['quality'])

data['quality'].value_counts

sns.countplot(data['quality'])
```

Поділимо набір даних на залежні та незалежні змінні:

```
# dividing the dataset into dependent and independent
variables

x = data.iloc[:, :11]

y = data.iloc[:, 11]

print(x.shape)

print(y.shape)
```


Поділимо набір даних на навчальну (75%) та тестову (25%) вибірки:

```
# dividing the dataset in training and testing set

from sklearn.cross_validation import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size = 0.25, random_state = 44)
```

Визначаємо обсяг навчальної (1199 записів) та тестової (400 записів) вибірок:

```
# determining the shapes of training and testing sets

print(x_train.shape)

print(y_train.shape)

print(x_test.shape)

print(y_test.shape)
```

Виконуємо стандартне масштабування (нормування) параметрів:

```
# standard scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

x_train = sc.fit_transform(x_train)

x_test = sc.fit_transform(x_test)
```

Далі переходимо до вирішення задачі різними методами.

Спочатку визначимо якість вина методом логістичної регресії (Logistic Regression):

```
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import classification_report,
confusion_matrix
```

```

from sklearn.model_selection import GridSearchCV,
cross_val_score

# creating the model

model = LogisticRegression()

# feeding the training set into the model

model.fit(x_train, y_train)

# predicting the results for the test set

y_pred = model.predict(x_test)

# calculating the training and testing accuracies

print("Training accuracy :", model.score(x_train, y_train))

print("Testing accuracy :", model.score(x_test, y_test))

# classification report

print(classification_report(y_test, y_pred))

# confusion matrix

print(confusion_matrix(y_test, y_pred))

```

Потім визначимо якість вина методом машини опорних векторів (Support Vector Machine):

```

from sklearn.svm import SVC

# creating the model

model = SVC()

# feeding the training set into the model

model.fit(x_train, y_train)

# predicting the results for the test set

y_pred = model.predict(x_test)

# calculating the training and testing accuracies

print("Training accuracy :", model.score(x_train, y_train))

```

```

print("Testing accuracy :", model.score(x_test, y_test))

# finding the best parameters for the SVC model

param = {

    'C': [0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4],

    'kernel': ['linear', 'rbf'],

    'gamma' : [0.1, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4]

}

grid_svc = GridSearchCV(model, param_grid = param, scoring =
'accuracy', cv = 10)

grid_svc.fit(x_train, y_train)

grid_svc.best_params_

# creating a new SVC model with these best parameters

model2 = SVC(C = 1.4, gamma = 0.1, kernel = 'rbf')

model2.fit(x_train, y_train)

y_pred = model2.predict(x_test)

print(classification_report(y_test, y_pred))

```

Далі визначимо якість вина методом дерев рішень (DecisionTree):

```

from sklearn.tree import DecisionTreeClassifier

# creating model

model = DecisionTreeClassifier()

# feeding the training set into the model

model.fit(x_train, y_train)

# predicting the results for the test set

y_pred = model.predict(x_test)

```

```

# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))

# classification report
print(classification_report(y_test, y_pred))

# confusion matrix
print(confusion_matrix(y_test, y_pred))

#Now lets try to do some evaluation for decision tree model
using cross validation.

model_eval = cross_val_score(estimator = model, X = x_train, y
= y_train, cv = 10)

model_eval.mean()

```

Далі визначимо якість вина нейромережевим методом з використанням багат шарового перцептрона (Multi Layer Perceptron):

```

from sklearn.neural_network import MLPClassifier

# feeding the training data to the model
model.fit(x_train, y_train)

# calculating the accuracies
print("training accuracy :", model.score(x_train, y_train))
print("testing accuracy :", model.score(x_test, y_test))

import keras

from keras.models import Sequential

from keras.layers import Dense

```

Using TensorFlow backend.

In [0]:

```

# creating the model

model = Sequential()

# first hidden layer

model.add(Dense(output_dim = 8, init = 'uniform', activation =
'relu', input_dim = 11))

# second hidden layer

model.add(Dense(output_dim = 1, init = 'uniform', activation =
'sigmoid'))

# Compiling the NN

# binary_crossentropy loss function used when a binary output
is expected

model.compile(optimizer = 'adam', loss =
'binary_crossentropy', metrics = ['accuracy'])

model.fit(x_train, y_train, batch_size = 10, nb_epoch = 100)

```

3.4 Висновок до розділу 3

У розділі було обґрунтовано вибір мови програмування Python та спеціалізованих бібліотек NumPy, Pandas та Matplotlib для програмної реалізації інформаційної технології нейромережевого визначення якості вина. Проведено аналіз набору даних параметрів вина для навчання та тестування нейронної мережі. Описано основні етапи програмної реалізації та функціонування засобів нейромережевого визначення якості вина, що складається з завантаження бібліотек, завантаження набору даних із параметрами вин та їх показниками якості, аналізу цих параметрів, їх попередньої обробки, вирішення задачі різними методами та виведення результатів.

4 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ ПРОГРАМИ НЕЙРОМЕРЕЖЕВОГО ВИЗНАЧЕННЯ ЯКОСТІ ВИНА

4.1 Тестування програми нейромережевого визначення якості вина

Результатом роботи є програмний застосунок, що визначає якість вина. Для того щоб впевнитися, що програма працює правильно, проведемо її тестування.

Після завантаження бібліотек та набору даних для навчання та тестування, програма виводить таблицю із статистичними показниками набору даних параметрів вин (середнє (mean), стандартне (std) мінімальне (min), максимальне (max), значення) – див. табл. 4.1:

Таблиця 4.1 – Статистичні показники набору даних параметрів вин

	fixed aci- dity	Vola- tile aci- dity	citric acid	Resi- dual sugar	Chlo- rides	free sulfur dio- xide	total sulfur dio- xide	Den- sity	pH	Sulph- ates	Alco- hol	Qua- lity
count	1599	1599	1599	1599	1599	1599	1599	1599	1599	1599	1599	1599
mean	8.319	0.527	0.27	2.538	0.087	15.87	46.46	0.996	3.311	0.658	10.42	5.636
std	1.741	0.179	0.19	1.409	0.047	10.46	32.89	0.001	0.154	0.169	1.065	0.807
min	4.600	0.120	0.00	0.900	0.012	1.000	6.000	0.990	2.740	0.330	8.400	3.000
max	15.90	1.580	1.00	15.50	0.611	72.00	289.0	1.003	4.010	2.000	14.90	8.000

Ми використовуємо тільки дані для червоних вин, а їх є 1599 зразків, причому вони розподілені по показниках якості так (показник якості – кількість зразків):

5	681
6	638
7	199
4	53
8	18
3	10

Далі проводимо візуалізацію результатів аналізу набору даних параметрів вин.

Спочатку проведеться двофакторний аналіз. Наприклад, перевірка варіації фіксованої кислотності у винах різної якості. Результат двофакторного аналізу варіації фіксованої кислотності у винах різної якості показано на рис. 4.1.

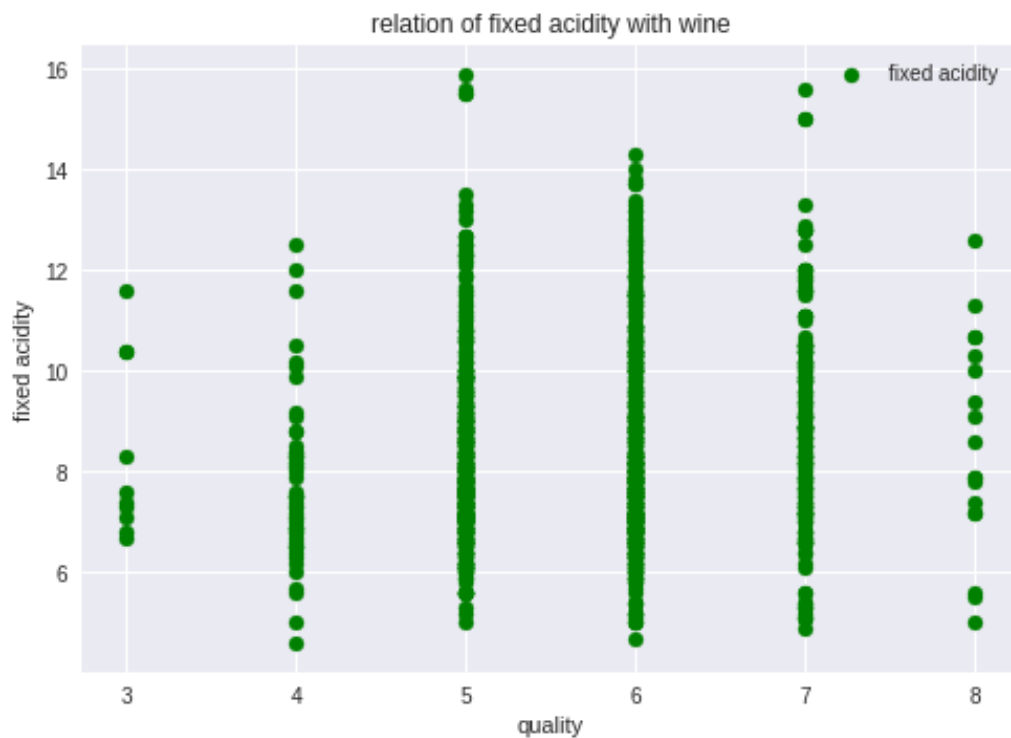


Рисунок 4.1 – Результат двофакторного аналізу варіації фіксованої кислотності у винах різної якості

Далі проводиться двофакторний аналіз варіації алкоголю у винах різної якості. Результат двофакторного аналізу варіації алкоголю у винах різної якості показано на рис. 4.2.

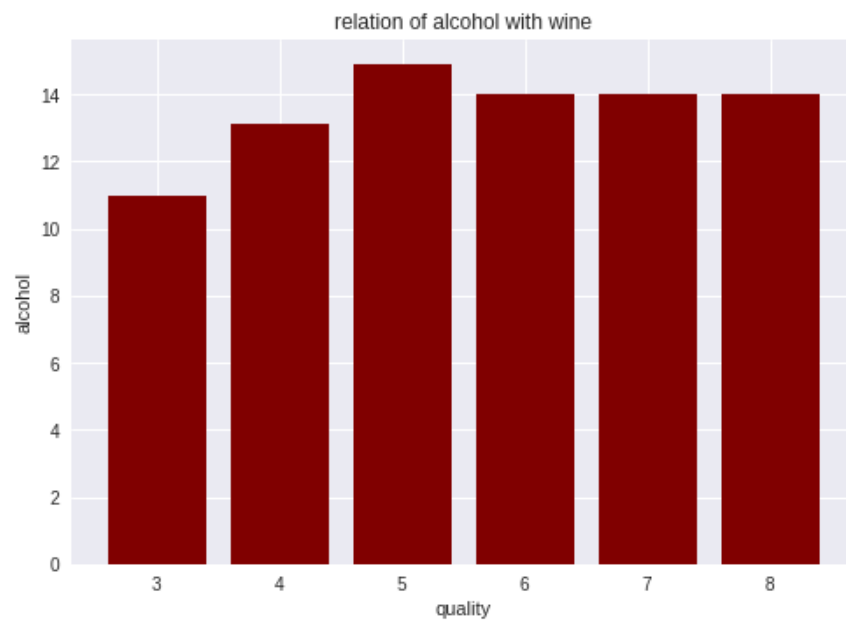


Рисунок 4.2 – Результат двофакторного аналізу варіації алкоголю у винах різної якості

Також двофакторний аналіз показує, що вміст лимонної кислоти збільшується з підвищенням якості вина (див. рис. 4.3).

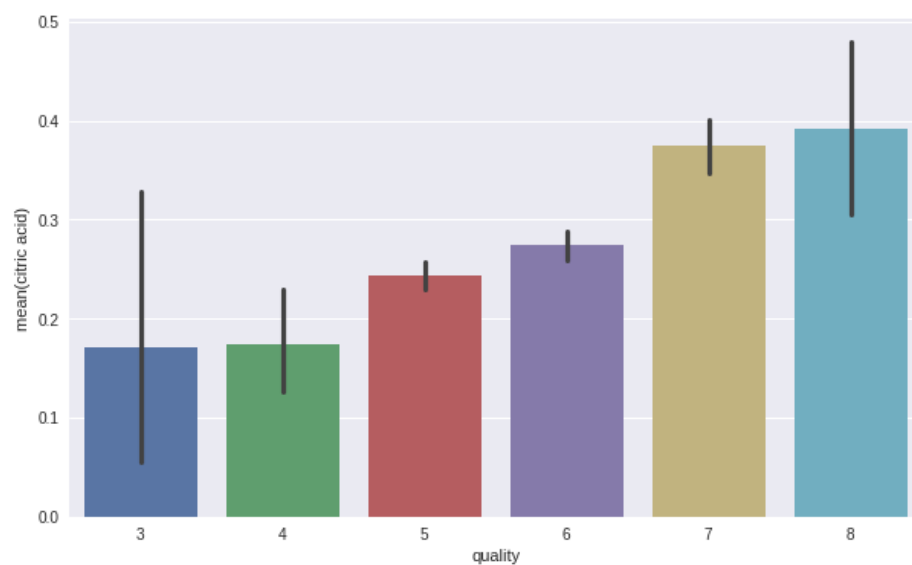


Рисунок 4.3 – Вміст лимонної кислоти збільшується з підвищенням якості вина

Вміст хлоридів знижується з підвищенням якості вина (див. рис. 4.4).

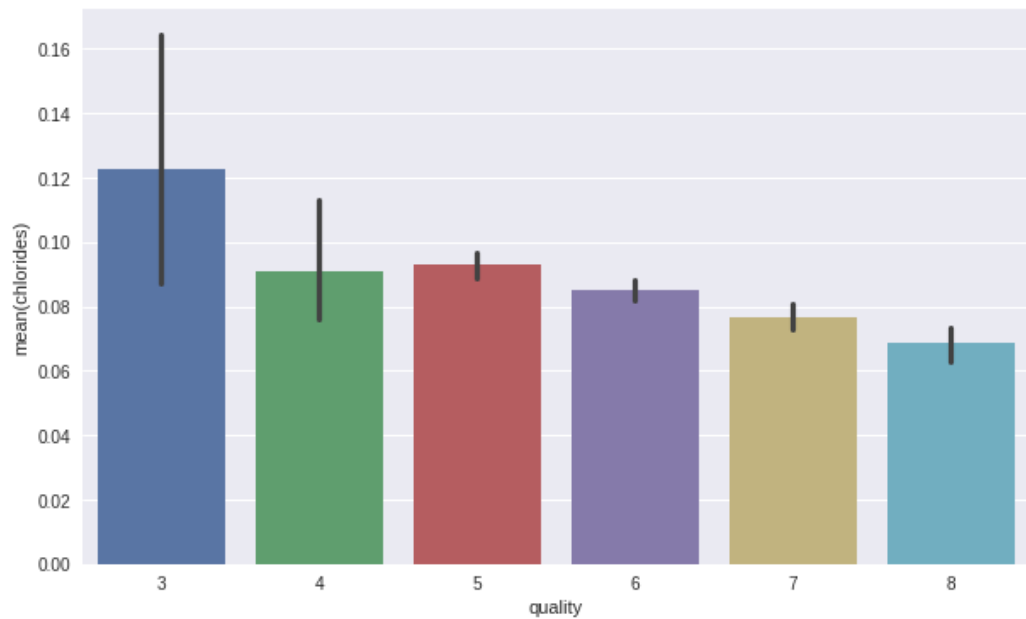


Рисунок 4.4 – Вміст хлоридів знижується з підвищенням якості вина

З підвищенням якості вина рівень сульфатів зростає (див. рис. 4.5).

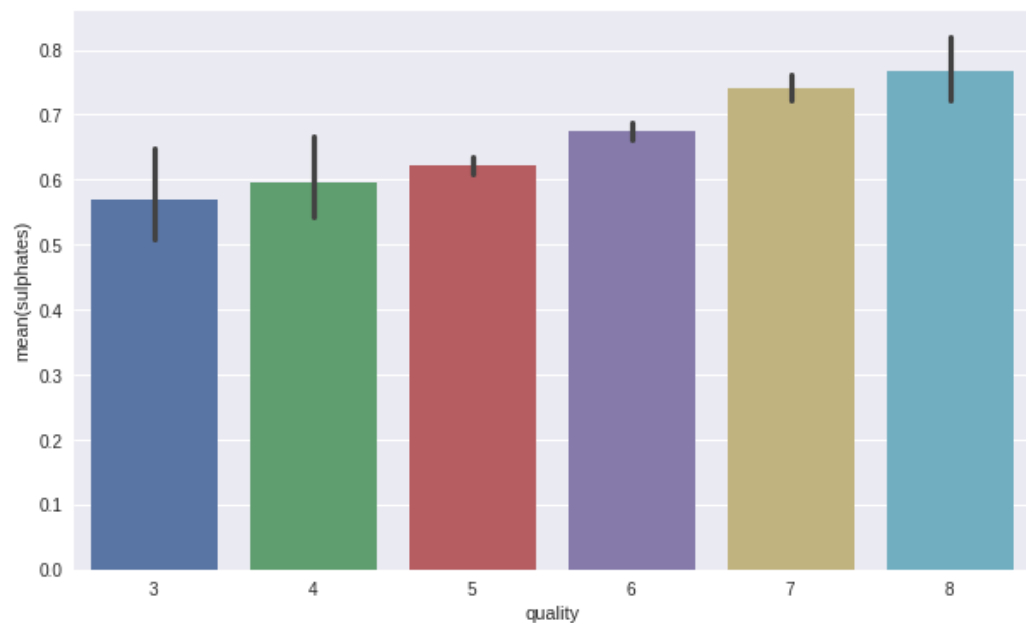


Рисунок 4.5 – Рівень сульфатів зростає з підвищенням якості вина

Оскільки ми бачимо, що два вищезазначені параметри (хлориди та сульфати) не мають сильного зв'язку із залежною змінною, то доцільним буде продемонструвати діаграму кореляції, щоб перевірити, які з параметрів більше пов'язані із залежною змінною, а які – менш пов'язані із залежними змінними. Діаграма кореляції представлена на рис. 4.6.

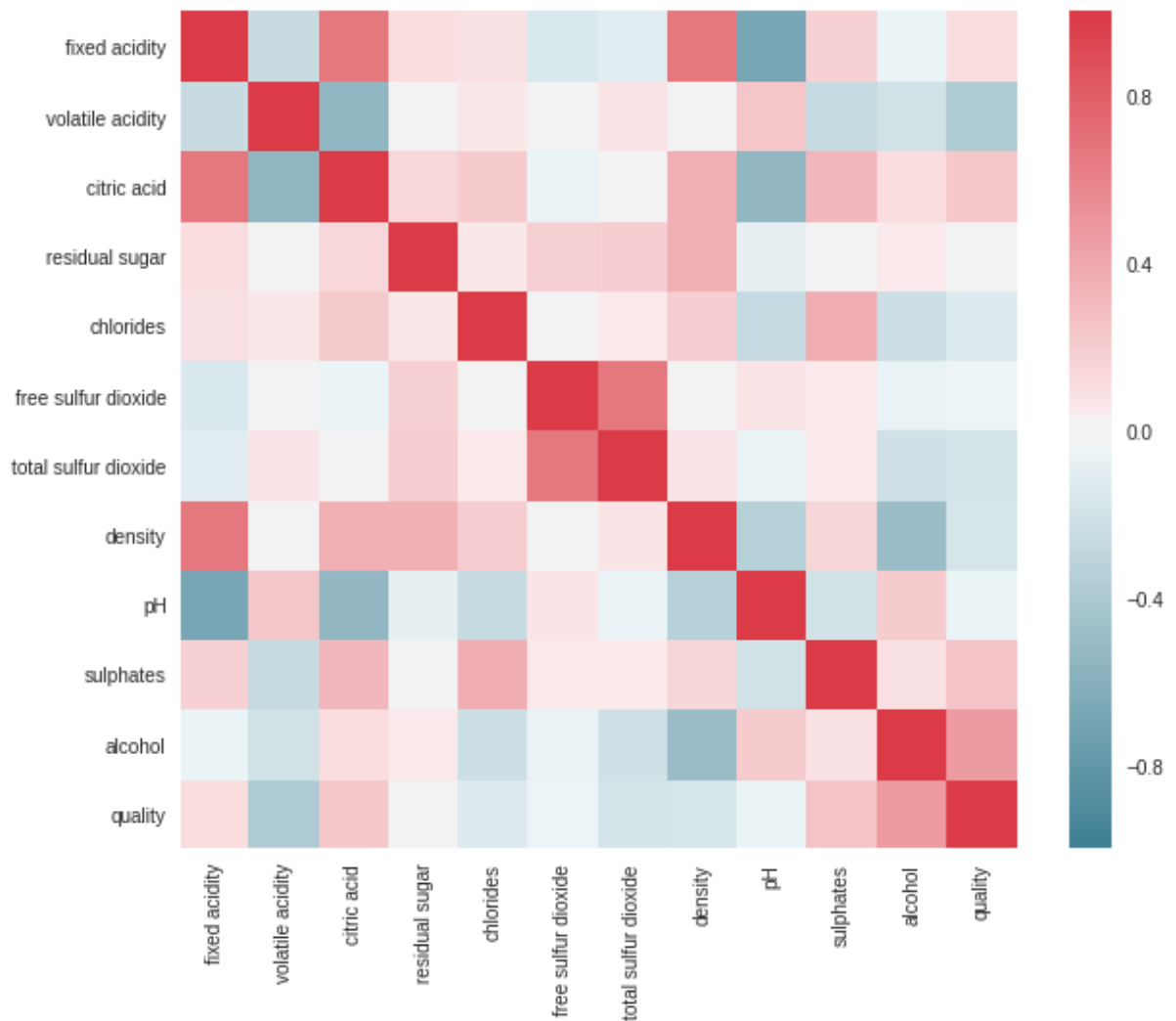


Рисунок 4.6 – Діаграма попарної кореляції параметрів вина

З наведеної вище діаграми попарної кореляції заданого набору даних для прогнозування якості вина ми можемо легко побачити, які параметри тісно пов'язані один з одним, а які – слабо. Наприклад:

Тісно пов'язані параметри:

1. Фіксована кислотність і лимонна кислота.
2. Вільний діоксид сірки та загальний діоксид сірки.
3. Фіксована кислотність і щільність.
4. Алкоголь і якість.

Отже, з вищезазначених пунктів можна зробити висновок, що алкоголь є найважливішою характеристикою для визначення якості вина.

Слабо корельовані параметри:

1. Лимонна кислота та летюча кислотність.
2. Фіксована кислотність і рН.
3. Щільність і алкоголь.

Також є параметри, які абсолютно не залежать один від одного.

Далі проводиться попередня обробка даних: видалення непотрібних стовпців із набору даних. В процесі роботи програми (двофакторного аналізу) ми побачили, що летюча кислотність, загальний діоксид сірки, хлориди, щільність дуже мало пов'язані із залежною змінною якість. Тож навіть якщо ми видалимо ці стовпці, точність не постраждає.

У розробленій програмі ми перетворили багатокласову оцінку якості вина на двокласову – вино добре або вино погане. Вина з оцінками від 2 до 6 вважаємо поганими, а вина з оцінками від 7 до 8 вважаємо добрими.

Аналіз кількості добрих і поганих вин у наборі даних показав, що із 1599 зразків хороших вин 855, а поганих – 744.

Набір даних було поділено на навчальну (75%) та тестову (25%) вибірки: обсяг навчальної вибірки склав 1199 записів, а тестової - 400 записів.

Перед введенням параметрів у класифікатор було виконано стандартне масштабування (нормування) параметрів.

Далі програма переходить до вирішення задачі класифікації якості вина різними методами.

Спочатку у програмі визначається якість вина методом **логістичної регресії** (Logistic Regression). Після обробки даних програма виводить точність класифікації якості вина на навчальній та тестовій вибірці, а також конфузійну матрицю:

Training accuracy : 0.7481234361968306

Testing accuracy : 0.7225

	precision	recall	f1-score	support
0	0.75	0.69	0.72	204
1	0.70	0.76	0.73	196
avg / total	0.72	0.72	0.72	400

```
[[141  63]
 [ 48 148]]
```

Потім у програмі визначається якість вина методом **машини опорних векторів** (Support Vector Machine). Після обробки даних програма також виводить точність класифікації якості вина на навчальній та тестовій вибірці, а також конфузійну матрицю:

Training accuracy : 0.8098415346121768

Testing accuracy : 0.73

	precision	recall	f1-score	support
0	0.75	0.73	0.74	204
1	0.72	0.74	0.73	196
avg / total	0.74	0.73	0.74	400

Далі у програмі визначається якість вина методом **дерев рішень** (DecisionTree). Після обробки даних програма також виводить точність класифікації якості вина на навчальній та тестовій вибірці, а також конфузійну матрицю:

Training accuracy : 1.0

Testing accuracy : 0.735

	precision	recall	f1-score	support
0	0.74	0.73	0.73	204
1	0.72	0.73	0.73	196
avg / total	0.73	0.73	0.73	400

```
[[148  56]
 [ 52 144]]
```

І, нарешті, у програмі визначається якість вина **нейромережевим методом** з використанням багатошарового персептрона (Multi Layer Perceptron). Після обробки даних програма також виводить точність класифікації якості вина на навчальній та тестовій вибірці:

training accuracy : 0.9374478732276897

testing accuracy : 0.7915

У структурі багатошарового персептрона перший та другий приховані шари мали 11 входів, 8 нейронів, функція активації - 'relu', вихідний шар мав 8 входів, 1 нейрон, функція активації - 'sigmoid'. Початкова присвоєння ваг здійснювалось випадковим чином за рівномірним законом розподілу. Використовувалось пакетне навчання (пакет по 10 зразків) протягом 100 епох. Фрагмент інформації про прогрес

процесу навчання по епохам, що виводиться програмою, представлено на рис. 4.7.

```

Epoch 1/100
1199/1199 [=====] - 1s 554us/step - loss: 0.6883 -
acc: 0.5505
Epoch 2/100
1199/1199 [=====] - 0s 102us/step - loss: 0.6188 -
acc: 0.6847
Epoch 3/100
1199/1199 [=====] - 0s 106us/step - loss: 0.5518 -
acc: 0.7448
Epoch 4/100
1199/1199 [=====] - 0s 102us/step - loss: 0.5275 -
acc: 0.7448
Epoch 5/100
1199/1199 [=====] - 0s 105us/step - loss: 0.5166 -
acc: 0.7440
Epoch 6/100
1199/1199 [=====] - 0s 103us/step - loss: 0.5120 -
acc: 0.7465

. . .

Epoch 96/100
1199/1199 [=====] - 0s 100us/step - loss: 0.4521 -
acc: 0.7873
Epoch 97/100
1199/1199 [=====] - 0s 97us/step - loss: 0.4514 -
acc: 0.7915
Epoch 98/100
1199/1199 [=====] - 0s 96us/step - loss: 0.4500 -
acc: 0.7973
Epoch 99/100
1199/1199 [=====] - 0s 95us/step - loss: 0.4516 -
acc: 0.7940
Epoch 100/100
1199/1199 [=====] - 0s 95us/step - loss: 0.4493 -
acc: 0.7915

```

Рисунок 4.7 – Фрагмент інформації про прогрес процесу навчання по епохам, що виводиться програмою

Із рис. 4.7 видно як від епохи до епохи зменшується похибка навчання (loss) від 0.6883 до 0,4493 та збільшується достовірність навчання (acc - accuracy) від 0,5505 до 0,7915 (від 55,05% до 79,15%).

4.2 Аналіз результатів роботи програми нейромережевого визначення якості вина

Чисельні результати роботи розробленого програмного забезпечення, отримані в результаті тестування та наведені вище, занесено до табл. 4.2 для наочності порівняння різних методів визначення якості вина (класифікації).

Таблиця 4.2 – Порівняння достовірності класифікації різних методів визначення якості вина

Метод класифікації	Кількість зображень у навчальній вибірці	Кількість зображень у тестовій вибірці	Достовірність класифікації на навчальній вибірці	Достовірність класифікації на тестовій вибірці
Метод логістичної регресії	1199	400	74,8%	72,3%
Метод машини опорних векторів	1199	400	81%	73%
Метод дерев рішень	1199	400	100%	73,5%
Нейромережевий метод	1199	400	93,7%	79,2%

Із табл. 4.2 видно, що розроблений програмний продукт має достовірність класифікації (визначення якості вина) на тестовій вибірці 79,2%, а найкращий із 3-х методів-аналогів має достовірність класифікації на тестовій вибірці 73,5%.

Таким чином, можна зробити висновок, що нейромережевий метод визначення якості вина має порівняно з методами-аналогами збільшену на 5,7% достовірність класифікації. Тобто мета роботи досягнута – достовірність визначення якості вина підвищена.

4.3 Висновок до розділу 4

У розділі в результаті тестування програми було доведено її працездатність та відповідність поставленому завданню. Навчання програми відбувалось з використанням бази даних для червоних вин, яка налічує 1599 записів. Набір даних було поділено на навчальну (75%) та тестову (25%) вибірки: обсяг навчальної вибірки склав 1199 записів, а тестової - 400 записів. Розроблена нейромережева програма має достовірність класифікації (визначення якості вина) на тестовій вибірці 79,2%, а найкращий із 3-х методів-аналогів має достовірність класифікації на тестовій вибірці 73,5%. Таким чином, розроблений нейромережевий метод визначення якості вина має порівняно з методами-аналогами збільшену на 5,7% достовірність класифікації. Тобто мета роботи досягнута – достовірність визначення якості вина підвищена.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Проведення комерційного та технологічного аудиту інформаційної технології вирішення задачі нейромережевого визначення якості вина

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту залучаємо 3-х незалежних експертів [19], якими є провідні викладачі випускової або спорідненої кафедри.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу здійснюємо із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, а результати зводимо до таблиці 5.1 [19,20].

Таблиця 5.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу інформаційної технології вирішення задачі нейромережевого визначення якості вина

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Технічна здійсненність концепції	3	2	3
Ринкові переваги (наявність аналогів)	2	3	2
Ринкові переваги (ціна продукту)	3	2	3
Ринкові переваги (технічні властивості)	3	2	3
Ринкові переваги (експлуатаційні витрати)	2	3	3
Ринкові перспективи (розмір ринку)	3	3	2
Ринкові перспективи (конкуренція)	2	3	3

Продовження таблиці 5.1

Критерії	Експерти		
	Експерт 1	Експерт 2	Експерт 3
	Бали, виставлені експертами		
Практична здійсненність (наявність фахівців)	3	2	3
Практична здійсненність (наявність фінансів)	2	3	3
Практична здійсненність (необхідність нових матеріалів)	3	3	3
Практична здійсненність (термін реалізації)	3	2	3
Практична здійсненність (розробка документів)	3	2	2
Сума балів	32	30	31
Середньоарифметична сума балів, СБ	31		

За результатами розрахунків, наведених в таблиці 5.1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал інформаційної технології вирішення задачі нейромережевого визначення якості вина – середній [20].

5.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати на оплату праці. Належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Основна заробітна плата дослідників. Витрати на основну заробітну плату дослідників ($З_0$) розраховують відповідно до посадових окладів працівників, за формулою:

$$З_0 = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.1)$$

де k – кількість посад дослідників, залучених до процесу дослідження; M_{ni} – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.; T_p – число робочих днів в місяці; приблизно $T_p = (21 \dots 23)$ дні; t_i – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 5.2.

Таблиця 5.2 – Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	10000	455	6	2730
Розробник	9000	409	12	4908
Консультанти	9000	409	7	2863
Всього:	10501			

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників ($З_p$) за відповідними найменуваннями робіт розраховують за формулою:

$$З_p = \sum_{i=1}^n C_i \cdot t_i, \quad (5.2)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год; t_i – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i і можна визначити за формулою:

$$C_i = \frac{M_m \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (5.3)$$

де M_m – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), у 2024 році $M_m = 8000$ грн; K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду; K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати; T_p – середня кількість робочих днів в місяці, приблизно $T_p = 21 \dots 23$ дні; $t_{зм}$ – тривалість зміни, год.

Таблиця 5.3 – Витрати на заробітну плату робітників

Найменування робіт	Трудовісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Аналіз предметної області	23	5	61,8	1,36	1421,4
Моделювання інформаційної технології	113	5	61,8	1,36	6983,4
Створення основної структури проєкту	18	6	65,9	1,45	1186,2
Створення та наповнення бази даних	79	6	65,9	1,45	5206,1
Тестування інформаційної технології	71	5	61,8	1,36	4387,8
Всього					18184,9

Додаткова заробітна плата. Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу

роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$З_d = 0,11 \cdot (З_o + З_p) = 0,11 \cdot (10501 + 18184,9) = 3155 \text{ грн.}$$

Відрахування на соціальні заходи. Нарахування на заробітну плату $H_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$\begin{aligned} H_{зп} &= \beta \cdot (З_o + З_p + З_d) = \\ &= 0,22 \cdot (10501 + 18184,9 + 3155) = 7005 \text{ грн.} \end{aligned}$$

де $З_o$ – основна заробітна плата розробників, грн.; $З_p$ – основна заробітна плата робітників, грн.; $З_d$ – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Амортизація обладнання. Амортизація обладнання, комп’ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування A в цілому бути розраховані за формулою:

$$A = \frac{Ц_6}{T_v} \cdot \frac{t}{12}, \quad (5.4)$$

де $Ц_6$ – загальна балансова вартість всього обладнання, комп’ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; t – термін використання основного фонду, місяці; T_v – термін корисного використання основного фонду, роки.

Таблиця 5.4 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук	35000	2	2	2917
Стіл	5500	5	2	183
Стілець	4700	5	2	157
Мишка	1200	2	2	100
Клавіатура	2500	2	2	208
Настільна лампа	600	2	2	50
Всього	3615			

Витрати на електроенергію для науково-виробничих цілей. Витрати на силову електроенергію $В_e$, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

Таблиця 5.5 – Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Ноутбук	0,26	330
Освітлення	0,033	150

$$\begin{aligned}
 В_e &= \sum \frac{W_i \cdot t_i \cdot Ц_e \cdot K_{впi}}{ККД} \\
 &= \frac{0,26 \cdot 330 \cdot 7,5 \cdot 0,85}{0,98} + \frac{0,033 \cdot 150 \cdot 7,5 \cdot 0,85}{0,98} \\
 &= 590 \text{ грн.},
 \end{aligned}$$

W_i – встановлена потужність обладнання, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год.; $Ц_e$ – вартість 1 кВт електроенергії,

грн.; $K_{\text{впi}}$ – коефіцієнт використання потужності; ККД – коефіцієнт корисної дії обладнання.

Інші витрати. До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (Z_o + Z_p) \cdot \frac{H_{\text{iв}}}{100\%} = (10501 + 18184,9) \cdot \frac{100}{100} = 28686 \text{ грн.},$$

де $H_{\text{iв}}$ – норма нарахування за статтею «Інші витрати».

Накладні (загальновиробничі) витрати. До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{\text{нзв}} = (Z_o + Z_p) \cdot \frac{H_{\text{нзв}}}{100\%} = (10501 + 18184,9) \cdot \frac{111}{100} = 31841 \text{ грн.},$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення науково-дослідної роботи. Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$\begin{aligned} B_{\text{заг}} &= Z_o + Z_p + Z_{\text{дод}} + Z_n + K_v + B_{\text{спец}} + B_{\text{прг}} + A_{\text{обл}} + B_e + \\ &\quad + I_v + B_{\text{нзв}} = \\ &= 10501 + 18184,9 + 3155 + 7005 + 3615 + 590 + 28686 + \\ &\quad 31841 = 100737 \text{ грн.} \end{aligned}$$

Загальні витрати. Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\eta} = \frac{100737}{0,9} = 111930 \text{ грн.,}$$

де η – коефіцієнт, що характеризує етап виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на стадії технічного проектування, то $\eta=0,2$.

5.3 Розрахунок економічної ефективності науково-технічної розробки інформаційної технології вирішення задачі нейромережевого визначення якості вина за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

В даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати: ΔN – збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу; N – кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки; Π_0 – вартість послуги у році до впровадження інформаційної системи; $\pm \Delta \Pi_0$ – зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi$ для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta \Pi = (\pm \Delta \Pi_0 \cdot N + \Pi_0 \cdot \Delta N_i)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.5)$$

де $\pm \Delta \Pi$ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки); $\pm \Delta \Pi_0$ може мати як додатне, так і від'ємне значення (від'ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни); N – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки; Π_0 – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році; Π_0 – основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження

результатів; ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки); λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2024 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,8333$; ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho = 0,2 \dots 0,5$; ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2024 році $\vartheta = 18\%$.

Очікуваний термін життєвого циклу розробки 1 рік, тому:

$$\Delta\P = ((1300 - 1000) \cdot 1000 - (1000 - 1300) \cdot 1300) \cdot 0,8333 \cdot 0,2 \cdot \left(1 - \frac{18}{100}\right) = 234807 \text{ грн.}$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\text{ПП} = \sum_{i=1}^T \frac{\Delta\P_i}{(1 + \tau)^t} = \frac{234807}{(1 + 0,1)^0} = 234807 \text{ грн.,}$$

де $\Delta\P$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.; T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо $T=1$ рік); τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$; t – період часу (в роках) від моменту початку впровадження науково-технічної

розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot ЗВ = 2 \cdot 111930 = 223860 \text{ грн.}$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}}=2\dots5$, але може бути і більшим; $ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV = 234807 - 223860 = 10947 \text{ грн.,}$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.; PV – теперішня вартість початкових інвестицій, грн.

Оскільки $E_{\text{абс}} > 0$, то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність E_v або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну

внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій E_v , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_v = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1 = \sqrt[1]{1 + \frac{10947}{223860}} - 1 = 0,24,$$

де $T_{\text{ж}}$ – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування $\tau_{\text{мін}}$, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{\text{мін}}$ визначається за формулою:

$$\tau_{\text{мін}} = d + f = 0,12 + 0,05 = 0,17,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = 0,9...0,12$; f – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина $f = 0,05...0,5$, але може бути і значно вищою.

Оскільки $E_v = 0,24 > \tau_{\text{мін}} = 0,17$, то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

Далі розраховуємо період окупності інвестицій T_o , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію

науково-технічної розробки інформаційної технології вирішення задачі нейромережевого визначення якості вина:

$$T_o = \frac{1}{E_b} = \frac{1}{0,24} = 4,2 \text{ року.}$$

Оскільки $T_o < 1 \dots 5$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки інформаційної технології вирішення задачі нейромережевого визначення якості вина і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

5.4 Висновок до розділу 5

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія вирішення задачі нейромережевого визначення якості вина» становить 31,0 балів, що вказує на комерційну вагомість проведення даних досліджень (рівень комерційного потенціалу розробки середній). Термін окупності становить 4,2 р., що менше 5-ти років, що говорить про комерційну привабливість науково-технічної розробки і може спонукати потенційних інвесторів на фінансування впровадження цієї розробки та виведення її на ринок. Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія вирішення задачі нейромережевого визначення якості вина».

ВИСНОВКИ

У роботі було розглянуто задачу визначення якості вина на основі нейромереж. В ході аналізу предметної області було розглянуто детальну постановку задачі визначення якості вина. Також описано різні методи розв'язання задачі класифікації і оцінено їх застосовність до завдання класифікації продукції виноробства. Серед таких методів машинного навчання як метод логістичної регресії, метод машини опорних векторів, метод k-найближчих сусідів, метод дерев рішень, Байєсівський класифікатор та нейромережевий метод як найбільш перспективний і застосовний до даної задачі було обрано нейромережевий метод. Крім цього, було обгрунтовано вибір способу виявлення переваг розробленої технології шляхом паралельного моделювання різних методів визначення якості вина.

В ході аналізу предметної області було обгрунтовано вибір типу нейронної мережі для інформаційної технології визначення якості вина. Із трьох розглянутих нейромереж: одношаровий персептрон, багатошаровий персептрон та RBF-нейромережа було обрано архітектуру нейронної мережі багатошаровий персептрон. Розроблено архітектуру нейронної мережі багатошаровий персептрон для інформаційної технології визначення якості вина, яка має 11 входів, 2 прихованих шари по 8 нейронів та вихідний шар із одного нейрона. У прихованих шарах обрано функцію активації ReLU та функцію активації Sigmoid у вихідному шарі. Для навчання цієї нейромережі використовується модифікація ADAM методу стохастичного градієнтного спуску. Розроблено алгоритм роботи програми нейромережевого визначення якості вина та UML діаграму класів.

Для програмної реалізації інформаційної технології нейромережевого визначення якості вина було обгрунтовано вибір мови програмування Python та спеціалізованих бібліотек NumPy, Pandas та Matplotlib. Проведено

аналіз набору даних параметрів вина для навчання та тестування програмних засобів. Описано основні етапи програмної реалізації та функціонування інформаційної технології нейромережевого визначення якості вина, що складається з завантаження бібліотек, завантаження набору даних із параметрами вин та їх показниками якості, аналізу цих параметрів, їх попередньої обробки, вирішення задачі різними методами та виведення результатів.

В результаті тестування програми було доведено її працездатність та відповідність поставленому завданню. Навчання нейромережі відбувалось з використанням бази даних для червоних вин, яка налічує 1599 записів. Набір даних було поділено на навчальну (75%) та тестову (25%) вибірки: обсяг навчальної вибірки склав 1199 записів, а тестової - 400 записів. Розроблена нейромережева програма має достовірність класифікації (визначення якості вина) на тестовій вибірці 79,2%, а найкращий із 3-х методів-аналогів має достовірність класифікації на тестовій вибірці 73,5%. Таким чином, розроблений нейромережевий метод визначення якості вина має порівняно з методами-аналогами збільшену на 5,7% достовірність класифікації. Тобто мета роботи досягнута – достовірність визначення якості вина підвищена.

В економічній частині було визначено рівень комерційного потенціалу розробки за темою «Інформаційна технологія вирішення задачі нейромережевого визначення якості вина», який становить 31,0 балів, що вказує на комерційну вагомість проведення даних досліджень (рівень комерційного потенціалу розробки середній). Термін окупності становить 4,2 р., що менше 5-ти років, що говорить про комерційну привабливість науково-технічної розробки і може спонукати потенційних інвесторів на фінансування впровадження цієї розробки та виведення її на ринок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. А. О. Закревський, О. К. Колесницький, Ю. М. Паночишин Інформаційна технологія вирішення задачі нейромережевого визначення якості вина / Матеріали конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2025)», Вінниця, 2024, [Електронний ресурс]. Режим доступу:
<https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/22814/18>
2. Wine Quality [Електронний ресурс] – Режим доступу:
<https://archive.ics.uci.edu/dataset/186/wine+quality>
3. Machine Learning with PyTorch and Scikit-Learn . Book Code Repository. Paperback: 770 pages. Publisher: Packt Publishing. Language: English. ISBN-10: 1801819319 I.SBN-13: 978-1801819312
4. Fauzi, M., Arifin, A.Z., Gosaria, S., Prabowo, I.S., 2017. Indonesian News Classification Using Naïve Bayes and Two-Phase Feature Selection Model. Indones. J. Electr. Eng. Comput. Sci. 8, 610–615. [Електронний ресурс] – Режим доступу: <https://doi.org/10.11591/ijeecs.v8.i3.pp610-615>
5. Інтелектуальний аналіз даних та машинне навчання. Частина 1. Базові методи та засоби аналізу даних (навчальний посібник) / Я. В. Іванчук, В. І. Месюра, А. А. Яровий, О. Д. Манжілевський – Вінниця : ВНТУ, 2021. – 69 с.
6. Кононова К. Ю. Машинне навчання: методи та моделі : підручник для бакалаврів, магістрів та докторів філософії спеціальності 051 «Економіка» / К. Ю. Кононова. Харків: ХНУ імені В. Н. Каразіна, 2020. 280 с. [Електронний ресурс]. Режим доступу:
https://karazin.ua/storage/staticcontent/source/documents/vydavnytstvo/2021/nachalnivydannia/Kononova_.pdf

7. Руденко О.В. Штучні нейронні мережі: Навчальний посібник / О.В.Руденко, Є.В.Бодянський. - Харків : ТОВ «Компанія СМІТ», 2006. — 404 с. - ISBN 966-8630-73-X.
8. Куссуль Н. М. Інтелектуальні обчислення: навчальний посібник (із грифом МОН України) / Н. М. Куссуль, А. Ю. Шелестов, А. Н. Лавренюк. - К.: «Наукова думка», 2006. — 186 с. ISBN 966-00-0592-X.
9. О.М. Колодчак Інтелектуальний аналіз даних Lviv Polytechnic National University Institutional Repository [Електронний ресурс] – Режим доступу: http://ena.lp.edu.ua/bitstream/ntb/26402/1/10_49-58.pdf
10. Kiwiel, Krzysztof C. (2001). Convergence and efficiency of subgradient methods for quasiconvex minimization. Mathematical Programming (Series A) 90 (1) (Berlin, Heidelberg: Springer). с. 1–25. ISSN 0025-5610. MR 1819784. doi:10.1007/PL00011414
11. Любунь З. М. Основи теорії нейромереж: текст лекцій / З. М. Любунь. – Львів : Видавничий центр ЛНУ ім. Івана Франка, 2006.
12. Python [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Python>
13. NumPy [Електронний ресурс] – Режим доступу: <https://numpy.org/>
14. Pandas - Python Data Analysis Library [Електронний ресурс] – Режим доступу: <https://pandas.pydata.org/>
15. Matplotlib: Visualization with Python [Електронний ресурс] – Режим доступу: <https://matplotlib.org/>
16. P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.
17. Dastmard, Benjamin. (2013). A statistical analysis of the connection between test results and field claims for ECUs in vehicles.

18. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

19. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

20. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. К. Колесницький. – Вінниця : ВНТУ, 2023. – (58 с.)

Додаток А (обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА
НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬНазва роботи: Інформаційна технологія вирішення задачі
нейромережевого визначення якості винаТип роботи: магістерська кваліфікаційна робота
(БДР, МКР)Підрозділ кафедра комп'ютерних наук, ФІІТА
(кафедра, факультет)

Показники звіту подібності Turnitin

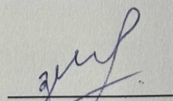
Оригінальність 81,00% Схожість 19,00%

Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

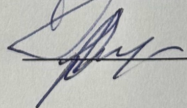
Ознайомлені з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи



Закревський А.О.

Керівник роботи

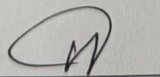


Колесницький О.К.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv('drive/My Drive/Projects/wine quality/winequality-
red.csv')

data.head()
data.columns
data.describe()
data.info()
data.isnull().sum()

# checking the variation of fixed acidity in the different qualities of wine

plt.scatter(data['quality'], data['fixed acidity'], color = 'green')
plt.title('relation of fixed acidity with wine')
plt.xlabel('quality')
plt.ylabel('fixed acidity')
plt.legend()
plt.show()

# checking the variation of fixed acidity in the different qualities of wine

plt.bar(data['quality'], data['alcohol'], color = 'maroon')
plt.title('relation of alcohol with wine')
plt.xlabel('quality')
plt.ylabel('alcohol')
plt.legend()
plt.show()

# Composition of citric acid go higher as we go higher in the quality of the
wine

import seaborn as sns

fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'citric acid', data = data)

fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'residual sugar', data = data)

#Composition of chloride go down as we go higher in the quality of the wine

fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'chlorides', data = data)

fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'free sulfur dioxide', data = data)

#Sulphates level goes higher with the quality of wine

fig = plt.figure(figsize = (10,6))
```

```

sns.barplot(x = 'quality', y = 'sulphates', data = data)

f, ax = plt.subplots(figsize=(10, 8))
corr = data.corr()
sns.heatmap(corr, mask=np.zeros_like(corr, dtype=np.bool),
            cmap=sns.diverging_palette(220, 10, as_cmap=True),
            square=True, ax=ax)

sns.pairplot(data)

# Removing Unnecassary columns from the dataset
# As we saw that volatile acidity, total sulphur dioxide, chlorides,
density are very less related to the dependent variable
# quality so even if we remove these columns the accuracy won't be
affected that much.

data = data.drop(['volatile acidity', 'total sulfur dioxide',
                  'chlorides', 'density'], axis = 1)

# checking the shape of the dataset
#print(data.shape)

data.columns

# converting the response variables(3-7) as binary response variables that is
either good or bad

names = ['bad', 'good']
bins = (2, 6.5, 8)

data['quality'] = pd.cut(data['quality'], bins = bins, labels = names)

data['quality'] = data['quality'].map({3 : 'bad', 4 : 'bad', 5 : 'bad',
                                       6 : 'good', 7 : 'good', 8 : 'good'})

# analyzing the different values present in the dependent variable(quality
column)
data['quality'].value_counts()

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

data['quality'] = le.fit_transform(data['quality'])

data['quality'].value_counts

sns.countplot(data['quality'])

# dividing the dataset into dependent and independent variables

x = data.iloc[:, :11]
y = data.iloc[:, 11]

# determining the shape of x and y.
print(x.shape)
print(y.shape)

```

```

# dividing the dataset in training and testing set

from sklearn.cross_validation import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.25,
random_state = 44)

# determining the shapes of training and testing sets
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)

# standard scaling
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.fit_transform(x_test)

```

Logistic Regression

```

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

from sklearn.model_selection import GridSearchCV, cross_val_score

# creating the model
model = LogisticRegression()

# feeding the training set into the model
model.fit(x_train, y_train)

# predicting the results for the test set
y_pred = model.predict(x_test)

# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))

# classification report
print(classification_report(y_test, y_pred))

# confusion matrix
print(confusion_matrix(y_test, y_pred))

```

Support Vector Machine

```

from sklearn.svm import SVC

# creating the model
model = SVC()

# feeding the training set into the model
model.fit(x_train, y_train)

```

```

# predicting the results for the test set
y_pred = model.predict(x_test)

# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))

# finding the best parameters for the SVC model
param = {
    'C': [0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4],
    'kernel': ['linear', 'rbf'],
    'gamma' : [0.1, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4]
}
grid_svc = GridSearchCV(model, param_grid = param, scoring = 'accuracy', cv =
10)

grid_svc.fit(x_train, y_train)

grid_svc.best_params_

# creating a new SVC model with these best parameters

model2 = SVC(C = 1.4, gamma = 0.1, kernel = 'rbf')
model2.fit(x_train, y_train)
y_pred = model2.predict(x_test)

print(classification_report(y_test, y_pred))

```

Decision Tree

```

from sklearn.tree import DecisionTreeClassifier

# creating model
model = DecisionTreeClassifier()

# feeding the training set into the model
model.fit(x_train, y_train)

# predicting the results for the test set
y_pred = model.predict(x_test)

# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))

# classification report
print(classification_report(y_test, y_pred))

# confusion matrix
print(confusion_matrix(y_test, y_pred))

#Now lets try to do some evaluation for decision tree model using cross
validation.

model_eval = cross_val_score(estimator = model, X = x_train, y = y_train, cv
= 10)
model_eval.mean()

```

Multi Layer Perceptron

```
from sklearn.neural_network import MLPClassifier

# creating the model
model = MLPClassifier(hidden_layer_sizes = (100, 100), max_iter = 150)

# feeding the training data to the model
model.fit(x_train, y_train)

# calculating the accuracies
print("training accuracy :", model.score(x_train, y_train))
print("testing accuracy :", model.score(x_test, y_test))
```

Artificial Neural Networks

```
import keras
from keras.models import Sequential
from keras.layers import Dense

# creating the model
model = Sequential()

# first hidden layer
model.add(Dense(output_dim = 8, init = 'uniform', activation = 'relu',
input_dim = 11))

# second hidden layer
model.add(Dense(output_dim = 8, init = 'uniform', activation = 'relu'))

# output layer
model.add(Dense(output_dim = 1, init = 'uniform', activation = 'sigmoid'))

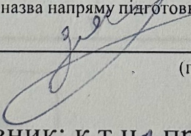
# Compiling the NN
# binary_crossentropy loss function used when a binary output is expected
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics =
['accuracy'])

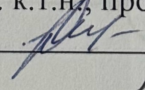
model.fit(x_train, y_train, batch_size = 10, nb_epoch = 100)
```


ІЛЮСТРАТИВНА ЧАСТИНА

**ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВИРІШЕННЯ ЗАДАЧІ
НЕЙРОМЕРЕЖЕВОГО ВИЗНАЧЕННЯ ЯКОСТІ ВИНА**

Виконав: студент 2-го курсу,
групи 1КН-23м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)


Закревський А.О.
(прізвище та ініціали)

Керівник: к.т.н., професор каф. КН

Колесницький О.К.
(прізвище та ініціали)

« 05 » 12 2024 р.

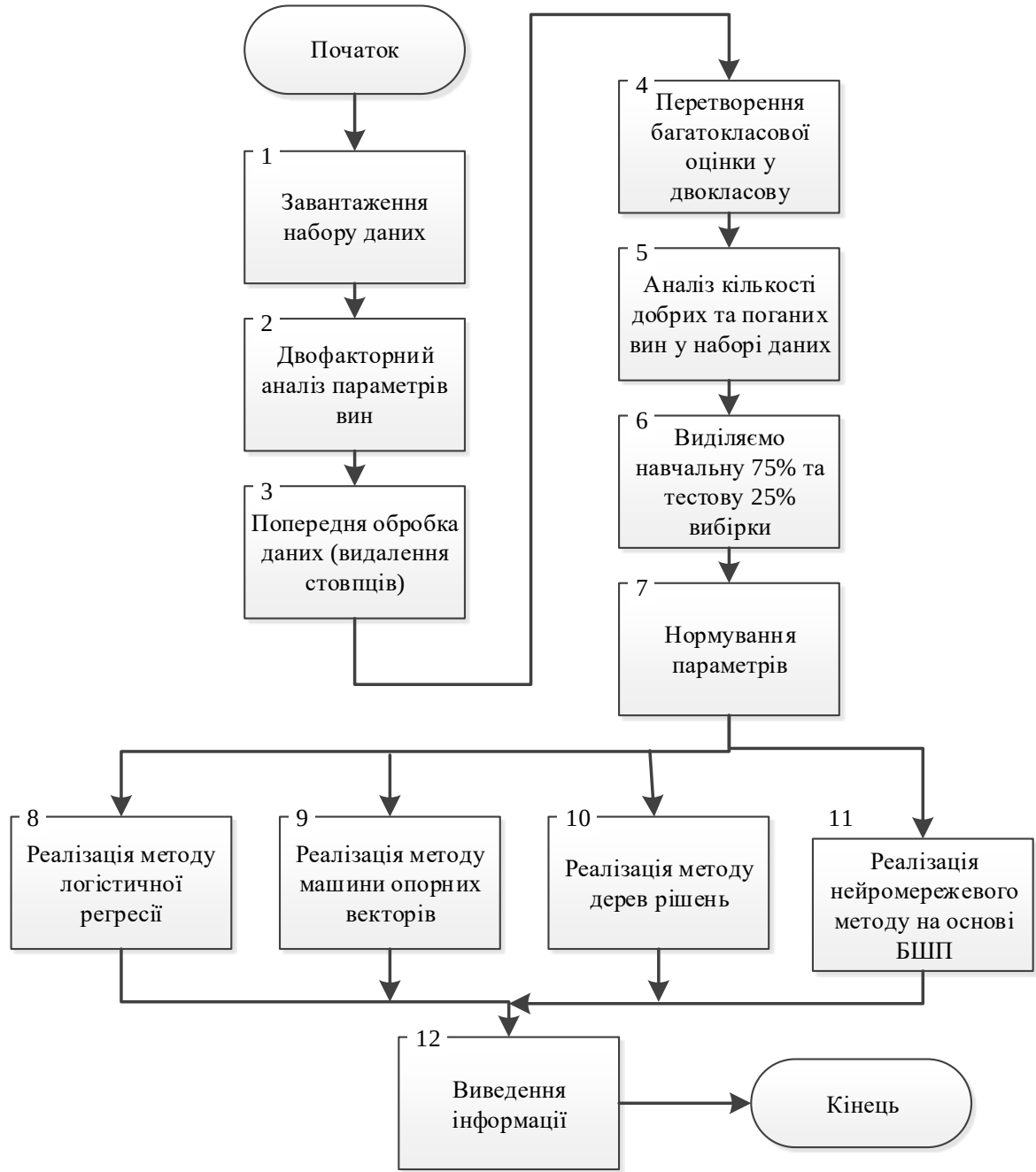


Рисунок В.1– Схема алгоритму роботи програми нейромережевого визначення якості вина

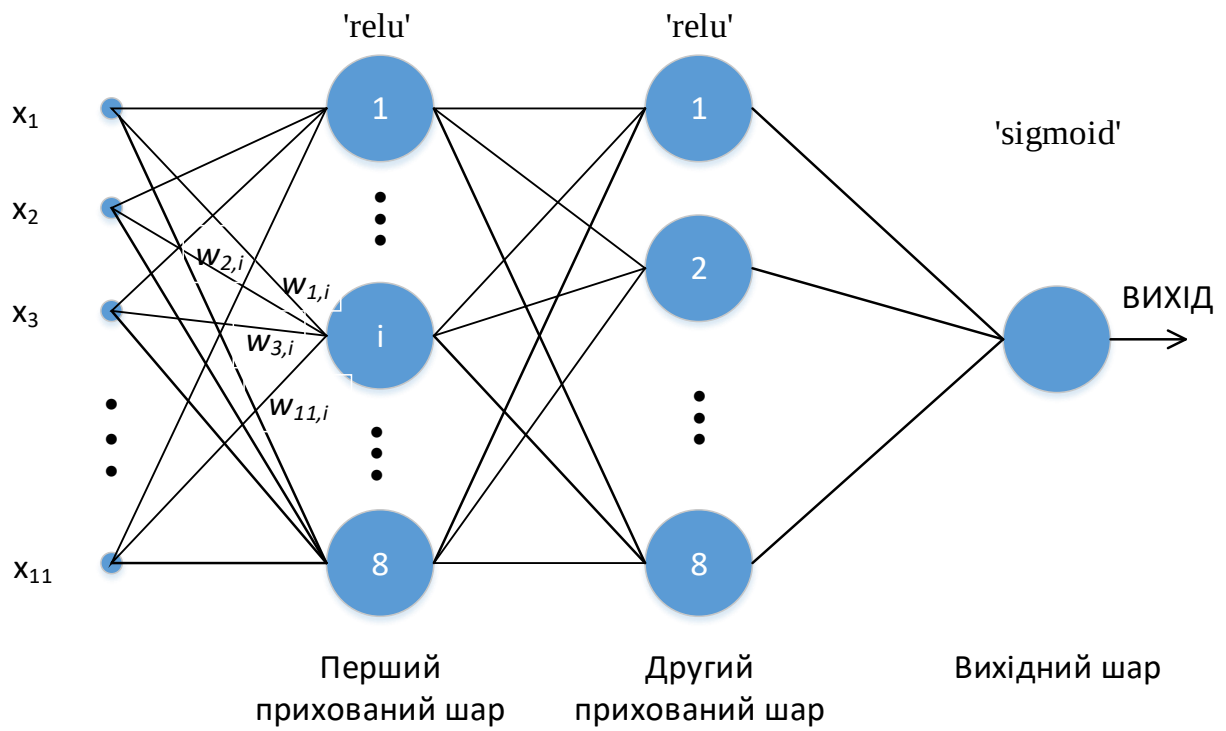


Рисунок В.2– Структура багатошарового персептрона для інформаційної технології визначення якості вина

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	Sulp- hates	Alco- hol	Qua- lity
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Рисунок В.3– Фрагмент набору даних параметрів вина для навчання та тестування нейромережі

Атрибути	Переклад	Діапазон зміни
Fixed acidity	Фіксована кислотність	3.8 ... 15.9
Volatile acidity	Летюча кислотність	0.1 ... 1.6
Citric acid	Лимонна кислота	0.0 ... 1.7
Residual sugar	Залишковий цукор	0.6 ... 65.8
Chlorides	Хлориди	0.01 ... 0.61
Free sulfur dioxide	Вільний діоксид сірки	1 ... 289
Total sulfur dioxide	Загальний діоксид сірки	6 ... 440
Density	Щільність	0.987 ... 1.039
pH	Кислотність pH	2.7 ... 4.0
Sulfates	Сульфати	0.2 ... 2.0
Alcohol	Алкоголь	8.0 ... 14.9
Quality (the output target)	Якість (вихідний параметр)	0 ... 10

Рисунок В.4– - Опис атрибутів вин.

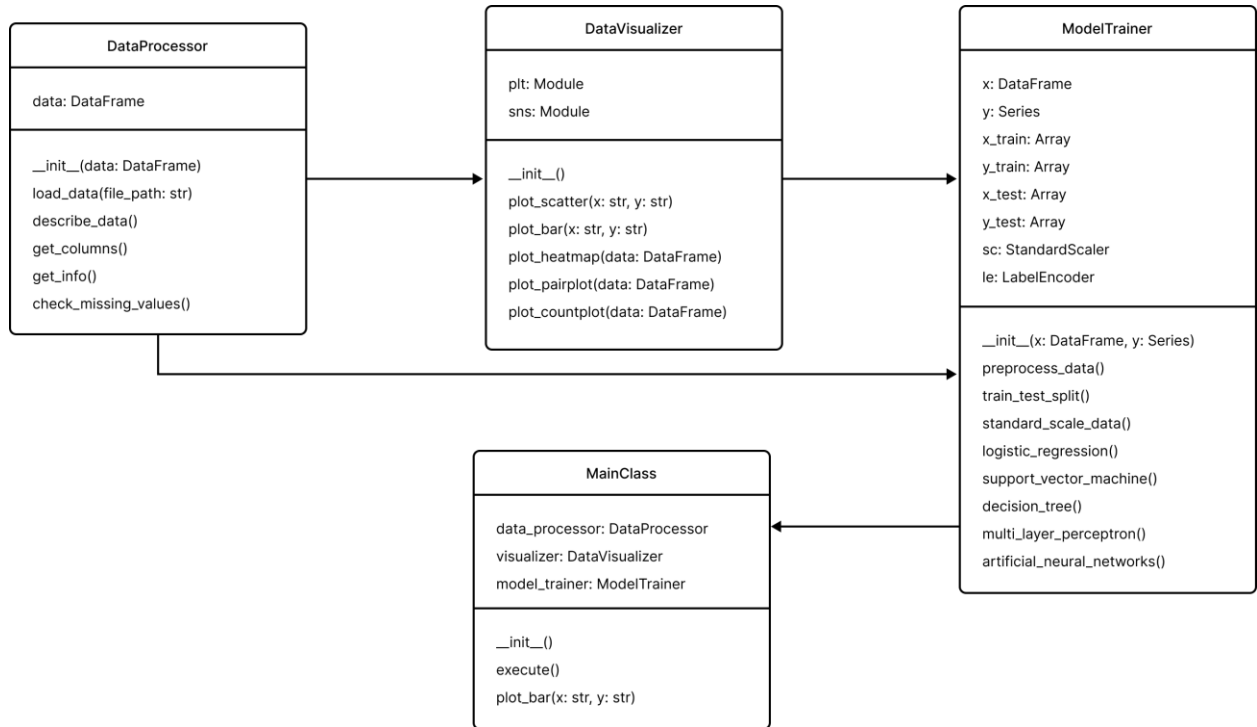


Рисунок В.5 – UML-діаграма класів програми визначення якості вина

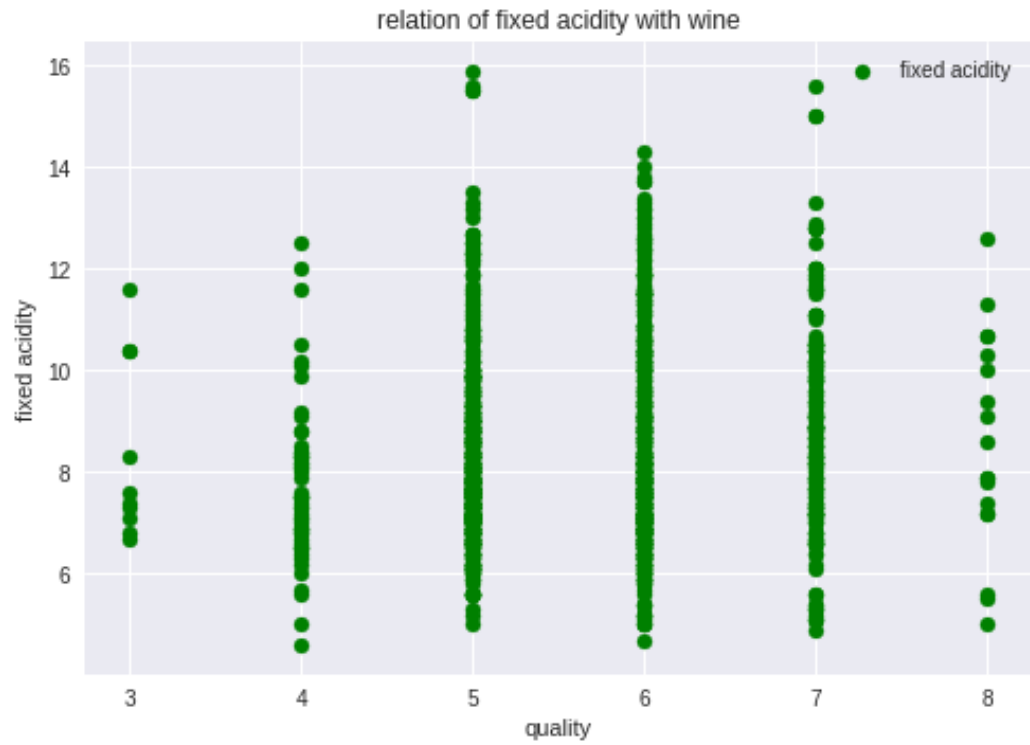


Рисунок В.6 – Результат двофакторного аналізу варіації фіксованої кислотності у винах різної якості

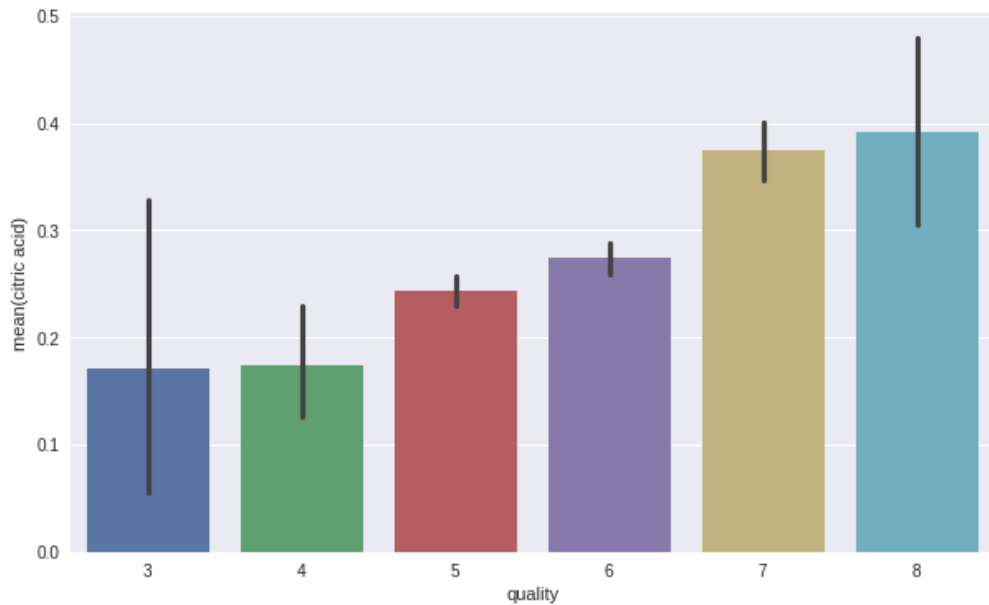


Рисунок В.7 – Вміст лимонної кислоти збільшується з підвищенням якості вина

```

Epoch 1/100
1199/1199 [=====] - 1s 554us/step - loss: 0.6883 - acc:
0.5505
Epoch 2/100
1199/1199 [=====] - 0s 102us/step - loss: 0.6188 - acc:
0.6847
Epoch 3/100
1199/1199 [=====] - 0s 106us/step - loss: 0.5518 - acc:
0.7448
Epoch 4/100
1199/1199 [=====] - 0s 102us/step - loss: 0.5275 - acc:
0.7448
Epoch 5/100
1199/1199 [=====] - 0s 105us/step - loss: 0.5166 - acc:
0.7440
Epoch 6/100
1199/1199 [=====] - 0s 103us/step - loss: 0.5120 - acc:
0.7465

. . .

Epoch 96/100
1199/1199 [=====] - 0s 100us/step - loss: 0.4521 - acc:
0.7873
Epoch 97/100
1199/1199 [=====] - 0s 97us/step - loss: 0.4514 - acc:
0.7915
Epoch 98/100
1199/1199 [=====] - 0s 96us/step - loss: 0.4500 - acc:
0.7973
Epoch 99/100
1199/1199 [=====] - 0s 95us/step - loss: 0.4516 - acc:
0.7940
Epoch 100/100
1199/1199 [=====] - 0s 95us/step - loss: 0.4493 - acc:
0.7915

```

**Рисунок В.8 – Фрагмент інформації про прогрес процесу навчання по епохам,
що виводиться програмою**