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 підтримки прийняття рішень щодо вибору ноутбуків. Частина 2. Нечітка логіка.»

Виконав: студенти 2-го курсу, групи
3КН-22м спеціальності 122
«Комп'ютерні науки»
(шифр і назва напряму підготовки, спеціальності)
Мельник І.С.
(прізвище та ініціали)

Керівник: д.т.н., проф. кафедри КН
Яровий А.А.
(прізвище та ініціали)
«07» 12 2023 р.

Опонент: д.т.н., проф. кафедри КСУ
Юхимчук М.С
(прізвище та ініціали)
«07» 12 2023 р.

Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А. А.
(прізвище та ініціали)
«08» 12 2023 р.

Вінниця ВНТУ - 2023 рік

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

ЗАТВЕРДЖУЮ
Директор ФОП «СЦ СЕРВІС+»
Гуменюк С.В.
29.08. 2023 року

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
д.т.н., проф. Яровий А.А.
29.08. 2023 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Мельнику Івану Сергійовичу

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра зі спеціальності: 122 – «Комп'ютерні науки»

керівник роботи д.т.н., професор кафедри КН Яровий А.А.
затверджені наказом вищого навчального закладу від "18" 09 2023 року №247.

1. Строк подання студентом роботи 13.11 2023 року

2. Вихідні дані до роботи:
   необхідна наявність хмарного доступу, кількість вхідних параметрів – не менше 7. Мінімальна потужність бази знань – не менше 30 правил. Спосіб введення даних – слайдер з мінімальним кроком 1. Об'єктно-орієнтована мова програмування.

3. Зміст текстової частини:
   вступ, аналіз предметної області вибору ноутбуків, моделювання та проектування інформаційної технології підтримки прийняття рішень щодо вибору ноутбуків, програмна реалізація нечіткої інформаційної технології, економічна частина, висновки, перелік використаних джерел, додатки.

4. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)
   Фрагменти нечітких термів, логічні ланцюги правил та контролера UML діаграма визначення методу, UML діаграма взаємодії компонентів технології

5. Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	викона прийн
1-3	Яровий А.А., д.т.н., проф. каф. КН	29.08.23	10.11.23
4	Кавецький В.В., к.е.н., доц. каф. ЕПВМ	21.10.23	29.10.23

6. Дата видачі завдання 29.08 2023 року

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Обґрунтування доцільності розробки інформаційної технології підтримки прийняття рішень щодо вибору	01.09.23	07.09.23	Розділ 1
2	Методи та моделі нечіткої логіки при проектуванні інформаційної технологія підтримки прийняття рішень щодо вибору ноутбуків	08.09.23	24.09.23	Розділ 2
3	Програмна реалізація інформаційної технології з використанням нечіткої логіки	25.09.23	20.10.23	Розділ 3
4	Підготовка економічної частини	21.10.23	29.10.23	Розділ 4
5	Апробація результатів дослідження	30.10.23	05.11.23	тези доповідей/акт впровадження
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23	10.11.23	Пояснювальна записка, графічний матеріал, презентація

Студент (підпис) Мельник І.С.

Керівник роботи (підпис) Яровий А.А.

АНОТАЦІЯ

УДК 004.891.2

Мельник І.С. Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків. Частина 2. Нечітка логіка. Магістерська кваліфікаційна робота зі спеціальності 122 – Комп’ютерні науки, освітня програма – Системи штучного інтелекту. Вінниця: ВНТУ, 2023. 115 с.

На укр. мові. Бібліогр.: 20 назв; рис.: 25; табл. 17.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології підтримки прийняття рішень щодо вибору ноутбуків. Було обгрунтовано доцільність розробки. Для рішення завдання відбору ноутбуків із великою кількістю користувацьких характеристик була використана нечітка логіка. Порівняно моделі поданя знань, обрано продукційну модель. Здійснене комп’ютерне моделювання процесу рекомендації вибору ноутбуків. Спроектована інформаційна технологія підтримки прийняття рішень для рекомендацій.

Обрано хмарне середовище Google Colab та мову програмування Python. Розроблений алгоритм роботи і проведене тестування програмного забезпечення та визначена мінімальна кількість вхідних параметрів становить 5. Мінімальна потужність не менше 30 правил. В ході порівняння показника якості надання рекомендацій запропонованої розробки та аналогу визначено, що підвищення якості надання рекомендації становить 7.18% у порівняні з аналогом.

У економічному розділі розраховано суму витрат на розробку та виготовлення нового технічного рішення, яка складає 536 926 гривень, спрогнозовано орієнтовану величину витрат по кожній з статей витрат, розраховано чистий прибуток, термін окупності витрат для виробника 1,56 роки.

Ключові слова: інформаційні технології, рекомендаційні технології, експертні технології, нечітка логіка, споживчі властивості, ноутбук.

ABSTRACT

Melnyk I.S. Information technology of decision support for the choice of laptops. Part 2. Fuzzy logic. Master's qualification work in specialty 122- Computer Science, educational program - Artificial intelligence systems. Vinnytsia: VNTU, 2023. 115 c.

In Ukrainian. Bibliography: 20 titles; Figures: 25; Table 17.

This master's qualification work is devoted to the development of information technology to support decision-making on the choice of laptops. The feasibility of the development was substantiated. To solve the problem of selecting laptops with a large number of user characteristics, fuzzy logic was used. The knowledge representation models were compared and the product model was chosen. The computer modeling of the process of recommending the choice of laptops was carried out. The information technology for decision support for recommendations is designed.

The cloud environment Google Colab and the programming language Python were chosen. An algorithm was developed and the software was tested, and the minimum number of input parameters was determined to be 5. The minimum capacity is at least 30 rules. In comparing the quality of recommendations of the proposed development and the analog, it was determined that the improvement in the quality of recommendations is 7.18% compared to the analog.

The economic section calculates the amount of costs for the development and manufacture of a new technical solution, which is 536,926 UAH, predicts the estimated cost for each of the cost items, calculates the net profit, and the payback period for the manufacturer is 1.56 years.

Keywords: information technologies, recommender technologies, expert technologies, fuzzy logic, consumer properties, laptop.

ЗМІСТ

ВСТУП

Актуальність теми. Магістерська робота присвячена розв'язанню актуальної науково-практичної задачі в галузі інформаційних технології для підтримки прийняття рішень щодо вибору ноутбуків з використанням нечіткої логіки. У сучасному світі ноутбуки є важливими пристроями, і правильний вибір ноутбуків може суттєво вплинути на ефективність та комфорт користувача. З урахуванням широкого асортименту ноутбуків та різноманітних технічних характеристик, виникає необхідність в розробці інформаційної технології, яка забезпечуватиме користувачів об'єктивними рекомендаціями щодо вибору найбільш підходящого ноутбуків для їхніх потреб.

Зв’язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна 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.

Метою та завдання досліджень: Метою магістерської кваліфікаційної роботи є підвищення якості надання рекомендацій при виборі ноутбуків із використанням нечіткої логіки, враховуючи індивідуальні потреби та вимоги користувачів.

Для досягнення поставленої мети необхідно виконати такі завдання:

- здійснити аналіз доцільності застосування інтелектуальних технологій для підтримки прийняття рішень щодо вибору ноутбуків;
- розробити базу знань з використанням нечіткої логіки;
- розробити діаграми нечітких модулів, розробити діаграми взаємодій класів;
- здійснити програмну реалізацію інтелектуальної технології для підтримки прийняття рішень щодо вибору ноутбуків;

- провести тестування програмного застосунку та проаналізувати результати.

Об’єктом дослідження є процес прийняття рішень щодо вибору ноутбуків на основі нечіткої логіки.

Предметом дослідження є програмні засоби прийняття рішень щодо вибору ноутбуків на основі нечіткої логіки.

Методи дослідження. у роботі використано такі методи наукових досліджень: теорія експертних систем; методи нечіткої логіки; методи об’єктно-орієнтованого програмування.

Наукова новизна отриманих результатів. Удосконалено інформаційну модель прийняття рішень щодо вибору ноутбуків, яка відрізняється від аналогічних використанням комбінованої функції належності із організацією нечіткого логічного виведення, що дозволило підвищити якість надання рекомендацій.

Практичне значення отриманих полягає в тому, що на основі проведених досліджень розроблено програмне забезпечення зокрема:

- розроблено нечітку базу знань та спроектовано інформаційну технологію підтримки прийняття рішень щодо вибору ноутбуків;
- реалізовано інформаційну технологію підтримки прийняття рішень щодо вибору ноутбуків із використанням хмарного середовища Google Colab для покращення роботи програмного продукту.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується коректністю постановки завдання, коректністю роботи програмних модулів, тестуванням програмної реалізації інформаційної технології підтримки прийняття рішень щодо вибору ноутбуків. Адекватність розробленої технології підтверджується результатами тестування та експериментальних досліджень.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У роботах, опублікованих у співавторстві, автору належать такі результати: [1] - розробка моделі вибору

ноутбуків на основі нечіткої логіки; [2] - реалізація інформаційної технології для прийняття рішень щодо вибору ноутбуків; проведення тестування і оцінка результатів.

Апробація. Результати дослідження були представлені та обговорені на LII науково-технічній конференції підрозділів ВНТУ (2023) та IV Міжнародній науково-практичній конференції ”Global Science: Prospects And Innovations”, м. Ліверпуль, Великобританія (2023).

Публікації За результатами дослідження опубліковано 1 тези [1] та 1 статтю у збірнику матеріалів міжнародної науково-технічної конференції [2], а також подано до реєстрації в Український національний офіс інтелектуальної власності та інновацій свідоцтво про реєстрацію авторського права на твір (комп’ютерна програма), реєстраційний номер Вх-44681/2023 від 14 листопада 2023.

1. ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВИБОРУ НОУТБУКІВ

1.1 Аналіз доцільності застосування інтелектуальних технологій для підтримки прийняття рішень щодо вибору ноутбуків

Використання інтелектуальних технологій, зокрема нечіткої логіки та експертних систем, для підтримки прийняття рішень має багато концептуальних переваг і може бути доцільним[1]. Розглянемо кілька основних аспектів:

1. Зменшення інформаційного шуму: На ринку існує велика кількість різних моделей ноутбуків з численними технічними параметрами та функціями. Інтелектуальна система може аналізувати цю інформацію та фільтрувати найважливіші аспекти, що спрощує процес вибору для користувача.

2. Прийняття рішень на основі особистих уподобань: Система може враховувати індивідуальні потреби та пріоритети кожного користувача. Наприклад, одному може бути важливою продуктивність для геймерів, іншому – легкість і портативність для мандрівок, а третьому – робоча потужність для професійного використання.

3. Оптимізація вартості та витрат: Інтелектуальна система може допомогти користувачам знайти найкращий баланс між функціональністю ноутбуків та його ціною. Це дозволяє заощадити кошти та обрати оптимальний варіант, який задовольнить потреби користувача.

4. Зменшення помилок і ризиків: Інтелектуальна система може допомогти уникнути неправильних виборів або недоліків у технічних характеристиках, які можуть з'явитися при недостатньому розумінні технологій. Це сприяє збільшенню задоволеності користувачів та підвищенню якості їхнього досвіду.

5. Заощадження часу та зусиль: Процес вибору ноутбуків може бути дуже часом і ресурсами витратним. Інтелектуальна система може значно прискорити цей процес, зменшивши необхідність в довгих дослідженнях та порівняннях.

Використання інтелектуальних технологій для підтримки прийняття рішень щодо вибору ноутбуків може значно полегшити цей процес для користувачів, забезпечуючи їм індивідуальний та оптимальний вибір, що відповідає їхнім потребам і вимогам. Такий підхід також сприяє підвищенню якості та об'єктивності прийнятих рішень.

1.2 Обґрунтування вибору нечіткої логіки при реалізації інформаційної технології підтримки прийняття рішень щодо вибору ноутбуків.

Використання нечіткої логіки в інформаційних технологіях має велике значення, особливо у сферах, де існує суб'єктивність або неоднозначність даних:

1. Нечіткість у визначенні параметрів: Часто терміни, які використовуються для опису характеристик продуктів, є нечіткими, наприклад, "висока продуктивність" або "продовжений час роботи батареї". Нечітка логіка дозволяє краще інтерпретувати такі поняття, забезпечуючи більш точні рекомендації.

2. Гнучкість у прийнятті рішень: В умовах нечітких та змінюваних вимог, як це часто буває при виборі технічних пристроїв, нечітка логіка може допомогти знайти оптимальний баланс між різними параметрами, адаптуючись до змінних умов і переваг користувачів.

3. Ефективність у ситуаціях неоднозначності: У сценаріях, де є множинність варіантів вибору і велика кількість факторів для розгляду, нечітка логіка допомагає обробляти і інтегрувати різноманітну інформацію, що покращує якість рішень.

4. Підтримка складних систем рекомендацій: Нечітка логіка ідеально підходить для систем рекомендацій, де потрібно балансувати між різними критеріями вибору та індивідуальними вподобаннями користувачів. Це робить процес вибору більш гнучким та відповідним до конкретних потреб.

5. Моделювання людської нечіткості у сприйнятті: Людське сприйняття часто характеризується нечіткістю – ми рідко мислимо в строго бінарних термінах. Нечітка логіка дозволяє моделювати цю нечіткість, роблячи взаємодію з технологіями більш природною та інтуїтивно зрозумілою.

Загалом, нечітка логіка надає можливість більш точно моделювати та враховувати людські особливості, суб'єктивність та неоднозначність у процесі прийняття рішень, особливо у сфері вибору споживчих товарів, таких як ноутбуки.

1.3 Відбір та визначення нечітких даних

В контекст вибору параметрів було зручно спростити та узагальнити процес аналізу вхідних змінних та побудувати UML, що допомагає візуалізувати, специфікувати, конструювати та документувати аспекти технології

Визначаємо вхідні змінні, які впливають на прийняття рішень щодо вибору ноутбуків. Кожна вхідна змінна представляє певну характеристику ноутбуків[2-3], яка може бути важливою для користувача (Рисунок 1.1) при виборі. З використанням UML, ми можемо створити діаграму класів, де кожен клас представляє певну категорію характеристик, наприклад, "Процесор", "Оперативна пам'ять", "Вага" і т.д. Це дозволить чітко структурувати та систематизувати інформацію, зробивши її більш доступною та зрозумілою.

Рисунок 1.1 — UML діаграма процесу вибору ноутбуків

Інтегруючи UML в процес вибору ноутбуків, можна забезпечити більшу ясність та структурованість процесу, що допоможе користувачам у прийнятті обґрунтованих та інформованих рішень.

Розглянемо список вхідних змінних та їх параметрів, що буде використаний в даній розробці:

1. RAM (Оперативна пам'ять):
   - Діапазон значень: від 4 до 64 ГБ
   - Назва: 'RAM'

2. SSD (Твердотільний накопичувач):
   - Діапазон значень: від 128 до 2048 ГБ
   - Назва: 'SSD'

3. HDD (Жорсткий диск):
   - Діапазон значень: від 512 до 4096 ГБ
   - Назва: 'HDD'

4. Diagonal (Діагональ екрану):
   - Можливі значення: 14, 15, 17 дюймів
   - Назва: 'Diagonal'

5. Matrix Type (Тип матриці екрану):
   - Можливі значення: 0 (TN), 1 (IPS), 2 (VA), 3 (OLED)
   - Назва: 'Matrix Type'

6. Keyboard (Клавіатура):
   - Діапазон значень: від 0 до 10 (де 0 - найгірше, 10 - найкраще)
   - Назва: 'Keyboard'

7. USB Ports (Кількість USB-портів):
   - Діапазон значень: від 0 до 10 (де 0 - мінімальна кількість, 10 - максимальна кількість)
   - Назва: 'USB Ports'

8. Camera (Камера):
   - Можливі значення: 0 (відсутня), 1 (присутня)
   - Назва: 'Camera'

Під час створення технології для підтримки прийняття рішень, важливим етапом є визначення категорій для кожної вхідної змінної. Це дозволяє системі більш точно відображати реальні умови та потреби користувачів. Наприклад, для вхідної змінної, такої як оперативна пам'ять (RAM) у ноутбуків, можна визначити декілька категорій, зазвичай класифікованих як низький, средній, та високий.

- Низький RAM: Ця категорія може відповідати ноутбуків з невеликою кількістю оперативної пам'яті, яка підходить для базових завдань, таких як веб-серфінг або офісні додатки.

- Середній RAM: Середній рівень оперативної пам'яті, який забезпечує достатню продуктивність для більшості завдань, включаючи деякі ігри та мультимедійні програми.

- Високий RAM: Висока кількість оперативної пам'яті, що є ідеальною для інтенсивних завдань, таких як професійний дизайн, розробка програмного забезпечення, або використання складних програм.

Для кожної з цих категорій можна визначити функції належності, які описують, наскільки конкретна кількість RAM відповідає кожній категорії. Наприклад, 4 ГБ оперативної пам'яті може мати високу ступінь належності до категорії 'low', але низьку - до 'medium' і 'high'.

Після визначення категорій та функцій належності, наступним кроком є створення технології правил, які вказують, як взаємодіють вхідні змінні. Наприклад, правило може вказувати, що якщо RAM є 'high', а процесор - 'high', то рекомендований тип ноутбуків може бути 'для геймерів' або 'для професійного використання'. Ці правила допомагають системі приймати більш точні та відповідні рішення на основі вхідних даних:

```
        # Правило: Якщо RAM низька і SSD низька, то Laptop Name = Brand1
        rule1 = ctrl.Rule(RAM['low'] & SSD['low'], laptop_name['Brand1'])
        # Правило: Якщо RAM висока і SSD висока, то Laptop Name = Brand7
        rule2      =      ctrl.Rule(RAM['high']      &      SSD['high'],
laptop_name['Brand7'])
```

Після обчислення вихідних значень можна вивести рекомендації або висновки щодо вибору ноутбуків на основі цих значень здійснити обчислення.

1.4 Аналіз об’єкту проектування

1. Моделювання нечіткості даних: Не завжди можливо оцінити характеристики ноутбуків чітко. Наприклад, характеристика "потужність процесора" може бути важко оцінити як "висока" або "низька", оскільки ця оцінка може залежати від конкретних потреб користувача. За допомогою нечіткої логіки можна ефективно враховувати таку нечіткість.

2. Суб'єктивність вибору: Вибір ноутбуків часто суб'єктивний, оскільки він базується на індивідуальних вподобаннях та потребах користувача. Нечітка логіка дозволяє враховувати ці суб'єктивні фактори та надає можливість користувачам налаштовувати вагомість різних параметрів у виборі[6].

3. Адаптивність технології: Нечітка логіка дозволяє створювати адаптивні технології, які можуть пристосовуватися до змінних умов і нових моделей ноутбуків на ринку. Система може легко оновлювати свої правила та враховувати нові дані.

4. Робота з багатьма параметрами: Вибір ноутбуків зазвичай враховує багато параметрів, таких як ціна, продуктивність, розмір екрану, вага і багато інших. Нечітка логіка дозволяє ефективно обробляти і враховувати цей мультикритеріальний аспект.

5. Зменшення помилок прийняття рішень: Система на основі нечіткої логіки допомагає користувачам уникнути помилкових рішень та зробити більш об'єктивний та інформований вибір.

6. Гнучкість та індивідуалізація: Нечітка логіка дозволяє створювати індивідуальні рекомендації для кожного користувача, враховуючи його унікальні потреби та пріоритети.

7. Покращення якості обслуговування клієнтів: Система на основі нечіткої логіки може забезпечити покращення якості обслуговування клієнтів, допомагаючи їм знайти найбільш підходящий ноутбук для їхніх потреб. Розглянемо конкретніше деякі параметри:

Вхідні параметри (Антецеденти):

1. RAM (Random Access Memory):

- Діапазон значень: від 4 ГБ до 64 ГБ.

- Функції належності: "low", "medium", "high".

- RAM визначає обсяг оперативної пам'яті ноутбуків. Функції належності "low", "medium" та "high" визначають, наскільки RAM належить до кожного з цих класів.

2. SSD (Solid State Drive):

- Діапазон значень: від 128 ГБ до 2048 ГБ.

- Функції належності: "low", "medium", "high".

- SSD вказує на обсяг SSD-пам'яті в ноутбуку. Функції належності допомагають визначити, які значення SSD належать до кожної категорії.

3. HDD (Hard Disk Drive):

- Діапазон значень: від 512 ГБ до 4096 ГБ.

- Функції належності: "low", "medium", "high".

- HDD вказує на обсяг HDD-пам'яті в ноутбуку. Функції належності розподіляють значення HDD між категоріями.

4. Diagonal (Діагональ екрану):

- Можливі значення: 14 дюймів, 15 дюймів, 17 дюймів.

- Функції належності: "small", "medium", "large".

- Діагональ визначає розмір екрану ноутбуків. Функції належності вказують, які значення діагоналі відносяться до кожної категорії.

5. Matrix Type (Тип матриці екрану):

- Можливі значення: TN, IPS, VA, OLED.

- Функції належності: "TN", "IPS", "VA", "OLED".

- Тип матриці вказує на характеристику екрану. Функції належності визначають належність значень до різних типів матриць.

6. Keyboard (Клавіатура):

- Діапазон значень: від 0 до 10 (релевантність).

- Функції належності: "low", "medium", "high".

- Якість та зручність клавіатури оцінюються на основі введених даних. Функції належності допомагають визначити належність значень до категорій.

7. USB Ports (Порти USB):

- Діапазон значень: від 0 до 10 (кількість портів).

- Функції належності: "low", "medium", "high".

- Кількість USB-портів в ноутбуку визначається на основі введених даних. Функції належності допомагають визначити належність значень до кожної категорії.

8. Camera (Камера):

- Можливі значення: відсутня (0), присутня (1).

- Функції належності: "absent", "present".

- Наявність камери в ноутбуку визначається на основі введених даних. Функції належності допомагають визначити, чи є камера в ноутбуці.

Вихідні параметри (Консиквенти):

1. Laptop Name (Назва ноутбуків):

- Діапазон значень: від 0 до 10 (індекси для назв ноутбуків).

- Функції належності: "Brand1" до "Brand10".

- Цей вихідний параметр визначає вибір назви ноутбуків зі списку брендів на основі введених даних та нечітких функцій належності.

2. Processor Name (Назва процесора):

- Діапазон значень: від 0 до 10 (індекси для назв процесорів).

- Функції належності: Назви процесорів.

- Цей вихідний параметр визначає вибір назви процесора зі списку на основі введених даних та нечітких функцій належності.

3. Graphics Card Name (Назва відеокарти):

- Діапазон значень: від 0 до 10 (індекси для назв відеокарт).

- Функції належності: Назви відеокарт.

- Цей вихідний параметр визначає вибір назви відеокарти зі списку на основі введених даних та нечітких функцій належності.

За допомогою цих даних можна реалізувати нечітку логічну модель, яка дозволить визначити оптимальний ноутбук на основі заданих характеристик.

Загальна UML діаграма наведена на рисунку 1.2

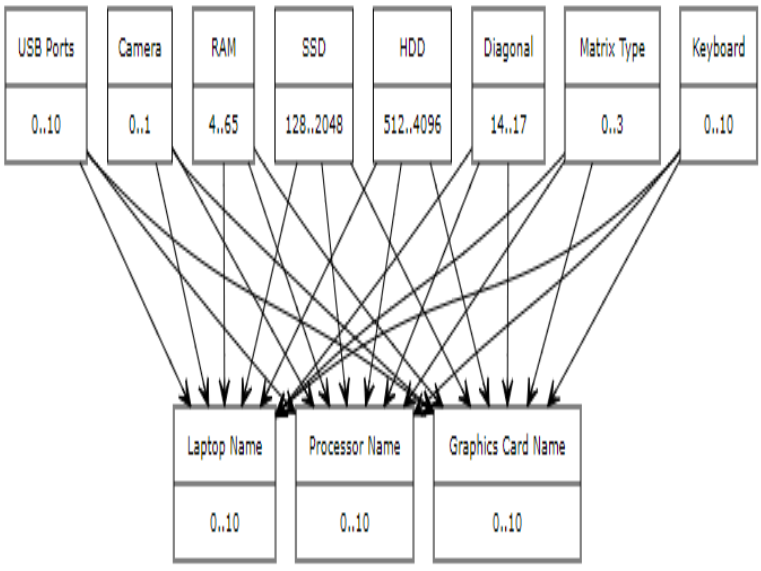


Рисунок 1.2— Загальна UML діаграма обраних параметрів

Для повного функціонування технології потрібно додати правила нечіткого виведення, які визначать, який ноутбук, процесор та відеокарта

найкраще відповідають введеним характеристикам, а також процедуру дефазифікації для отримання кінцевих значень.

1.5 Висновок до розділу 1

Вибір нечіткої логіки як основи для реалізації технології є обґрунтованим, оскільки ця методологія дозволяє ефективно враховувати не лише конкретні значення технічних характеристик ноутбуків, але й особисті вподобання та потреби користувачів. Аналіз об'єкту проектування показав, що завдання вибору ноутбуків є складним завданням, яке вимагає врахування різних параметрів та відомостей про різні моделі та бренди. Доцільність застосування інтелектуальних технологій, таких як нечітка логіка, є очевидною, оскільки вони дозволяють підтримувати користувачів у прийнятті обґрунтованих рішень на основі невизначеності та нечіткості даних.

Відбір та визначення нечітких даних є важливою складовою розробки технології, і він вимагає уважного аналізу та вибору параметрів, які будуть використовуватися для прийняття рішень. Загалом, інформаційна технологія, заснована на нечіткій логіці, має потенціал спростити та оптимізувати процес вибору ноутбуків для користувачів, забезпечуючи індивідуальні рекомендації та об'єктивний підхід до цього завдання.

2. МЕТОДИ ТА МОДЕЛІ НЕЧІТКОЇ ЛОГІКИ ПРИ ПРОЕКТУВАННІ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВИБОРУ НОУТБУКІВ.

2.1 Класифікація методів нечіткої логіки

Спершу розглянемо основні підходи що можуть бути доцільними для вирішення для вирішення поставленої задачі, а саме методи нечіткої логіки є важливим інструментом в сучасних інформаційних технологіях, особливо при розробці систем підтримки прийняття рішень. Ці методи дозволяють моделювати та аналізувати нечіткі технології, де інформація може бути невизначеною або нечіткою.

1. Методи базовані на множинах[8]:

- Нечіткі множини: це множини, де кожен елемент має ступінь належності до множини, який може бути в діапазоні від 0 до 1. Це дозволяє представляти нечіткість та невизначеність в даних.

- Операції над нечіткими множинами: такі як об'єднання, перетин та доповнення, дозволяють обробляти нечіткі множини та отримувати нові нечіткі множини.

2. Методи базовані на правилах:

- Нечіткі правила: це правила, які використовують нечіткі множини для представлення умов та висновків. Наприклад, "якщо температура є 'високою', то швидкість обертання вентилятора є 'швидкою'".

- Нечіткі технології висновків: це технології, які використовують набір нечітких правил для отримання висновків на основі вхідних даних.

3. Методи оптимізації:

- Нечітка логіка Такагі-Сугено: це метод, який використовує лінійні функції для представлення висновків в нечітких правилах.

- Нечітка логіка Мамдані: це метод, який використовує нечіткі множини для представлення висновків в нечітких правилах.

4. Методи класифікації та регресії:

- Нечіткі нейронні мережі: це нейронні мережі, які використовують нечіткі множини для представлення ваг та активаційних функцій.

- Нечіткі кластери: це методи, які дозволяють групувати дані на основі їх нечіткої схожості.

5. Методи ідентифікації систем:

- Нечітка ідентифікація: це метод, який дозволяє ідентифікувати параметри нечіткої технології на основі вхідних та вихідних даних.

- Нечітка модельована система: це система, яка використовує нечіткі правила для моделювання динаміки технології.

На основі вищенаведеного перспективним варто зазначити при реалізації нечіткої бази знань найбільш поширеною є продукційна модель подання знань . Вона дозволяє представляти знання у вигляді правил "якщо-то", де "якщо" є умовою, а "то" є висновком. Продукційна модель подання знань є однією з основних моделей в області штучного інтелекту та експертних систем. Вона базується на використанні продукційних правил, які мають форму "якщо-то".

- Якщо (умова): Ця частина правила визначає умову або набір умов, які повинні бути виконані. Умови можуть бути простими (наприклад, "температура вища за 30 градусів") або складними, комбінуючи декілька умов за допомогою логічних операторів[9].

- То (висновок): Ця частина правила визначає дію або набір дій, які слід виконати, якщо умова виконана. Висновок може включати в себе виведення нової інформації, зміну стану технології або виконання певних дій.

Продукційні моделі дозволяють ефективно представляти знання експертів у структурованому вигляді(правила, евристики) та використовувати ці знання для прийняття рішень в автоматизований спосіб.

Методи нечіткої логіки є спеціальними підходами до обробки нечітких даних та прийняття рішень на основі нечітких правил. Ці методи можна класифікувати за різними критеріями. Ось декілька основних категорій для класифікації методів нечіткої логіки:

1. Методи нечіткого визначення (Fuzzy Set Theory):

- Класичні нечіткі множини: Використовуються для опису нечітких змінних та їх приналежності до певних категорій.
- Нечіткі числа (Fuzzy Numbers): Використовуються для опису нечітких числових даних і відносин між ними.

2. Методи нечіткого виведення (Fuzzy Inference Systems):

- Технології нечіткого виведення (Fuzzy Inference Systems, FIS): Використовуються для прийняття рішень на основі нечітких правил та виводять нечіткі висновки.

3. Методи агрегації нечітких даних:

- Максимінний метод (Max-Min): Використовує максимінне правило для агрегації нечітких значень.
- Максимаксний метод (Max-Max): Використовує максимаксне правило для вибору найвищого нечіткого значення.
- Середньоквадратичний метод (Mean of Maxima): Використовує середньоквадратичне правило для агрегації нечітких значень.

4. Методи розв'язання оптимізаційних завдань:

- Метод лінійного програмування з нечіткими обмеженнями: Використовує нечіткі обмеження у задачах лінійного програмування.
- Метод нечіткої оптимізації: Використовує нечіткі функції цілі та обмеження для оптимізаційних завдань.

5. Методи вибору та ранжування альтернатив:

- Методи агрегації нечітких балів: Використовуються для обчислення загального бала для альтернатив.

- Методи порівняння нечітких чисел: Використовуються для порівняння нечітких числових даних.

6. Методи моделювання нечітких відносин:

- Методи нечітких відносин (Fuzzy Relation Methods): Використовуються для моделювання нечітких відносин між об'єктами або подіями.

7. Методи дефазифікації (Defuzzification):

- Центр тяжіння (Centroid): Використовується для перетворення нечітких висновків у деяку конкретну точку в просторі значень.

8. Методи нечіткої класифікації:

- Технології класифікації з використанням нечітких правил: Використовуються для класифікації об'єктів або подій на основі нечітких правил.

9. Методи роботи з великими обсягами даних:

- Методи нечіткої кластеризації: Використовуються для групування схожих даних в кластери з використанням нечітких відносин(Рисунок 2.1).

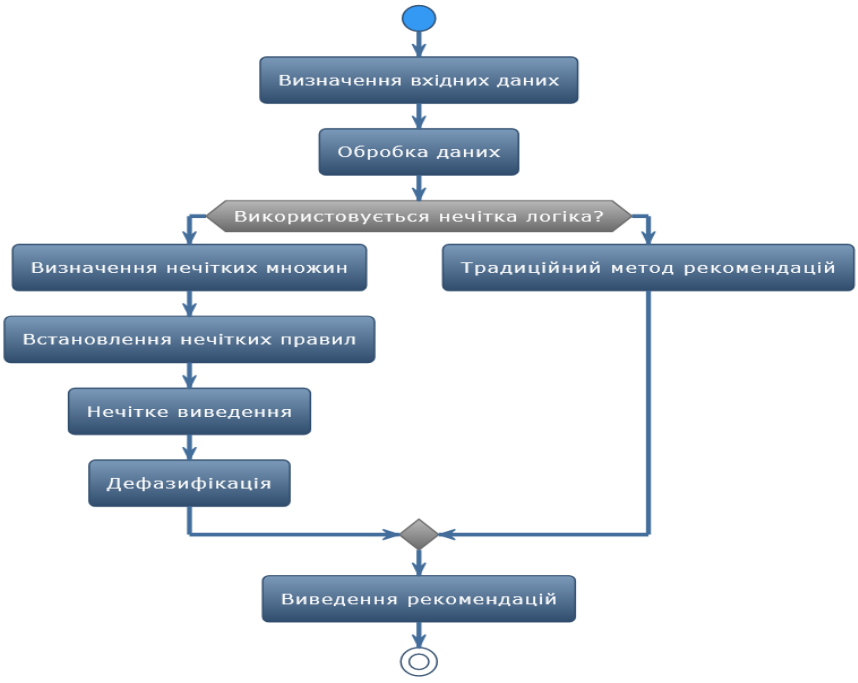


Рисунок 2.1— UML діаграма процесу надання рекомендацій

Ця класифікація допомагає розуміти різні аспекти та застосування методів нечіткої логіки в інформаційних технологіях, включаючи прийняття рішень, обробку даних та моделювання нечітких відносин[10].

2.2 Аналіз та обґрунтування вибору методів нечіткої логіки

У сфері штучного інтелекту існує багато методів подання та обробки знань. Однією з ключових технологій, яка дозволяє ефективно моделювати реальний світ і адаптуватися до змін, є нечітка логіка. UML наведена на рисунку 2.2.

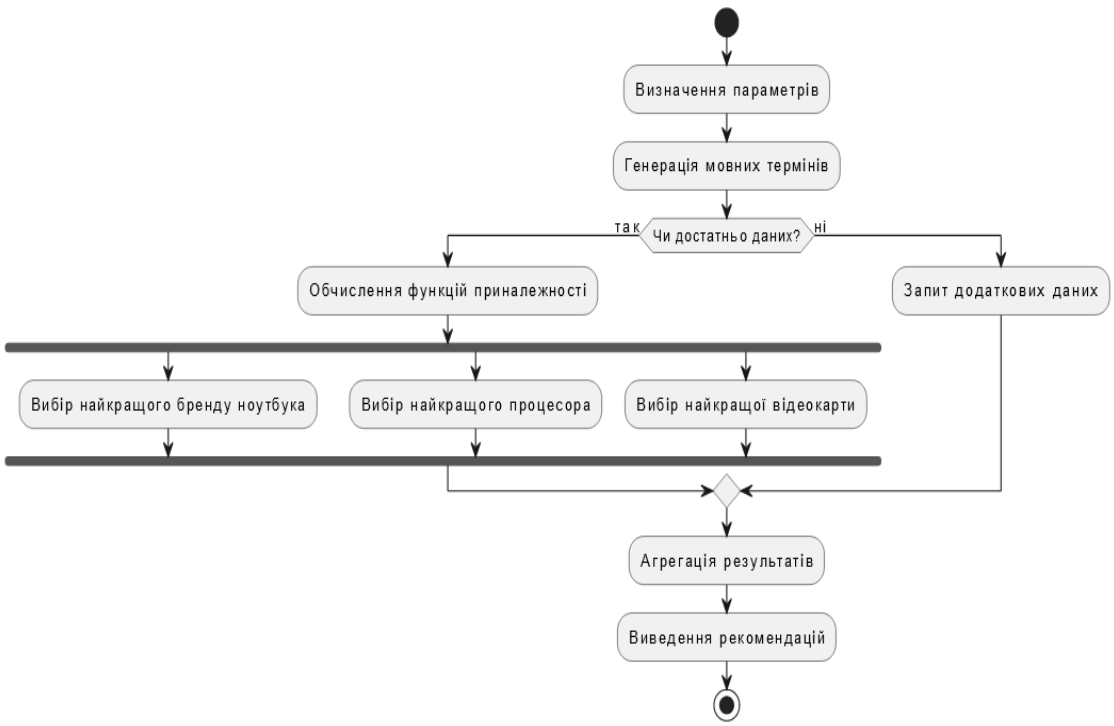


Рисунок 2.2— UML діаграма процесу прийняття рішень на основі нечіткої логіки

Поширеною моделлю подання що використовується в нечіткій логіці є продуційна модель падання знань. Вона базується на використанні правил у форматі "якщо-то", де "якщо" представляє умову, а "то" - дію або висновок. Основна перевага цієї моделі полягає в її гнучкості та здатності легко

інтегруватися з іншими системами. У контексті нечіткої логіки, продукційна модель дозволяє ефективно представляти та обробляти нечіткі правил[11-14].

На основі вищезазначеного розглянемо комбінований підхід, що об'єднує переваги різних підходів до нечіткої логіки. Він дозволяє використовувати продукційну модель для представлення знань та інші методи для обробки цих знань, такі як методи нечіткої інференції, нейронні мережі та інші техніки машинного навчання.

Методи агрегації нечітких даних:

Ці методи дозволяють комбінувати нечіткі значення характеристик для отримання загальної оцінки, що спрощує процес прийняття рішень.

В межах комбінованого підходу доцільно розглянути методи кластеризації та класифікації зокрема:

Методи нечіткої кластеризації та класифікації:

Ці методи можуть бути корисними для групування схожих ноутбуків на основі нечітких характеристик або для класифікації ноутбуків за певними критеріями.

Вибір методів нечіткої логіки для конкретної задачі залежить від багатьох факторів, включаючи характер задачі, доступні дані, рівень нечіткості та вимоги до точності. Комбінація різних методів може забезпечити високу точність та надійність в технологіях підтримки прийняття рішень. На основі вищенаведеного перспективною варто відзначити, що при реалізації нечіткої бази знань найбільш поширеною є продукційна модель подання знань. Ця модель базується на використанні правил у формі "якщо-то", що дозволяє ефективно моделювати експертні знання та враховувати різноманітні умови та варіанти розвитку ситуацій. Поширеною моделлю подання знань щодо вибору ноутбуків використання в нечіткій логіці, а саме нечітка логіка Продукційна модель забезпечує гнучкість у визначенні правил та легкість їх модифікації, що є важливим для адаптації технології до змінних умов та вимог.

2.3 Комп’ютерне моделювання процесу надання рекомендацій на основі нечіткої логіки

Комп'ютерне моделювання є важливим процесом створення нечіткої інтелектуальної технології, яка враховуватиме нечіткість та суб'єктивність даних користувача для надання персоналізованих рекомендацій. Ось кроки, котрі були використані при створенні даної технології:

1. Визначення задачі.

2. Визначення вхідних змінних: Визначте вхідні змінні, які впливають на прийняття рішення. Наприклад, для задачі вибору ноутбуків це можуть бути характеристики, такі як ціна, продуктивність, розмір екрану тощо.

3. Створення нечітких множин: Для кожної вхідної змінної визначте нечіткі множини та функції приналежності. Наприклад, для ціни можуть бути створені нечіткі множини "дуже низька", "низька", "середня", "висока", "дуже висока".

4. Визначення правил: Встановіть правила, які вказують, як взаємодіють вхідні змінні для прийняття рішення. Наприклад, "Якщо ціна низька і продуктивність висока, то рекомендувати цей ноутбук".

5. Агрегація правил: Використовуйте агрегацію правил для виведення нечітких висновків на основі вхідних даних та правил.

6. Дефазифікація: Перетворіть нечіткі висновки у конкретні числові значення або рекомендації.

7. Персоналізація рекомендацій: Враховуйте індивідуальні вподобання та ступінь важливості кожної вхідної змінної для кожного користувача. Це може здійснюватися через налаштування параметрів технології для кожного користувача.

8.Тестування та валідація: Перевірте та протестуйте систему на реальних даних та з використанням сценаріїв вибору, щоб переконатися, що вона надає адекватні та корисні рекомендації.

9.Оновлення та підтримка: Забезпечте систему механізмами оновлення та підтримки, оскільки вимоги користувачів та характеристики товарів можуть змінюватися з часом.

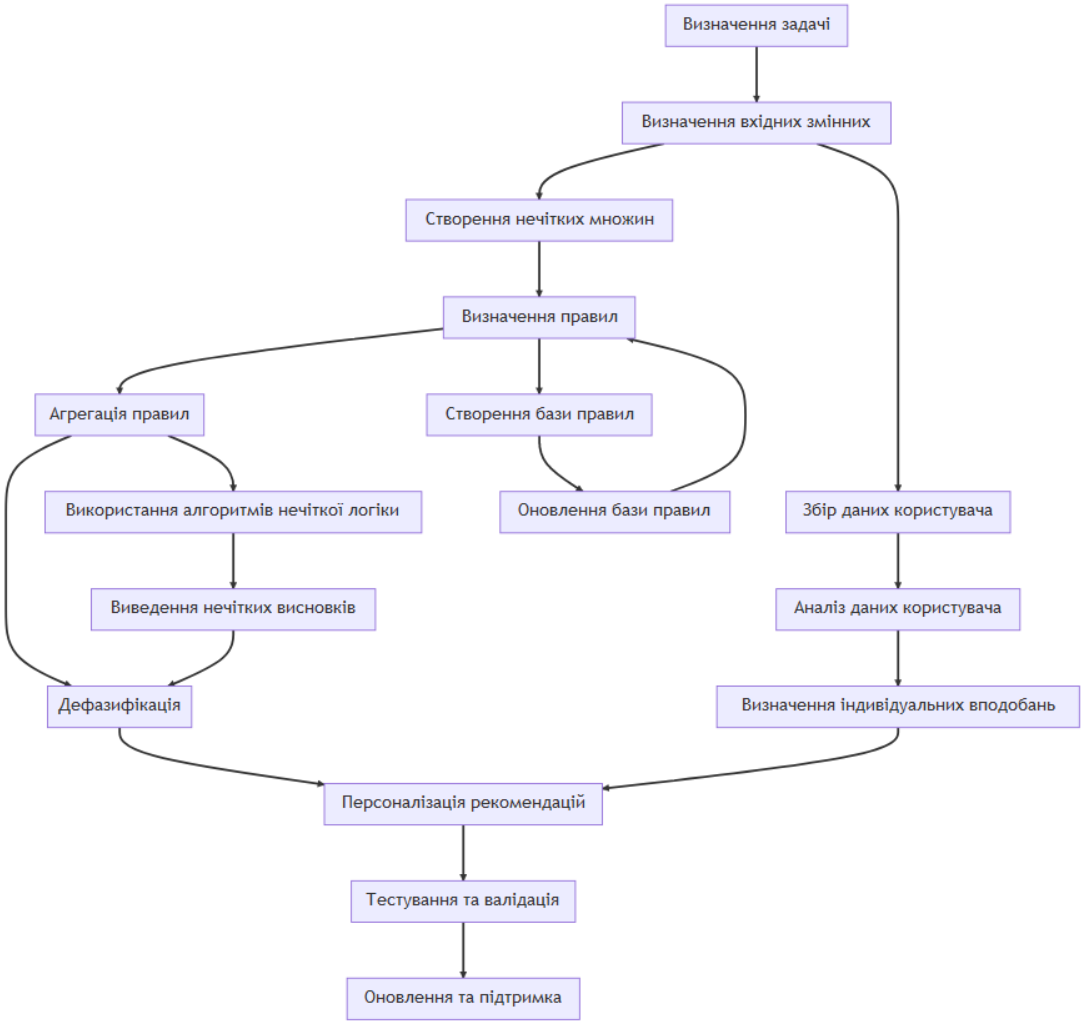


Рисунок 2.3 — UML діаграма моделювання процесу надання рекомендацій

Така модель на основі нечіткої логіки дозволить надавати персоналізовані рекомендації на основі суб'єктивних вимог і характеристик користувача, що спростить процес прийняття рішень в ситуаціях, де дані нечіткі або непередбачувані.

2.3.1 Моделювання бази знань в середовищі Google Colab

Google Colab (скорочено від Google Colaboratory) - це безкоштовний інтерактивний сервіс для обчислень в хмарі, який надає можливість створювати та виконувати код на мові програмування Python у віртуальному середовищі, не потребуючи встановлення та налаштування середовища на власному комп'ютері. Він широко використовується спільнотою програмістів, дослідників та студентів, оскільки надає безліч переваг[6]:

1. Безкоштовність: Використання Google Colab є абсолютно безкоштовним. Ви можете створювати та виконувати свій код без витрат на апаратне забезпечення або плату за послуги.

2. Віддалений доступ: Colab працює в хмарному середовищі Google, що означає, що ви можете отримувати доступ до своїх проектів з будь-якого пристрою з Інтернетом, не турбуючись про локальні обмеження обладнання.

3. Підтримка GPU: Google Colab надає можливість використовувати графічні обчислювальні прискорювачі (GPU) для обчислень у великих обсягах даних та глибокого навчання (Deep Learning). Ви можете отримати доступ до GPU безкоштовно.

4. Зберігання та обмін проектами: Ви можете зберігати свої проекти в Google Drive або на GitHub і легко ділитися ними з іншими користувачами або колегами.

5. Підтримка Jupyter Notebook: Google Colab інтегрується з Jupyter Notebook, що дозволяє створювати і виконувати ноутбуки з кодом, графіками, текстом та іншими елементами в інтерактивному режимі.

6. Бібліотеки та завантаження даних: Ви можете встановлювати різні бібліотеки Python та завантажувати дані з різних джерел, щоб працювати над своїми проектами.

7. Обмеження ресурсів: Незважаючи на безкоштовну доступність, Google Colab має обмеження на обчислювальні ресурси та час виконання, але це може бути покращено за допомогою Google Colab Pro - платної версії з більшими ресурсами.

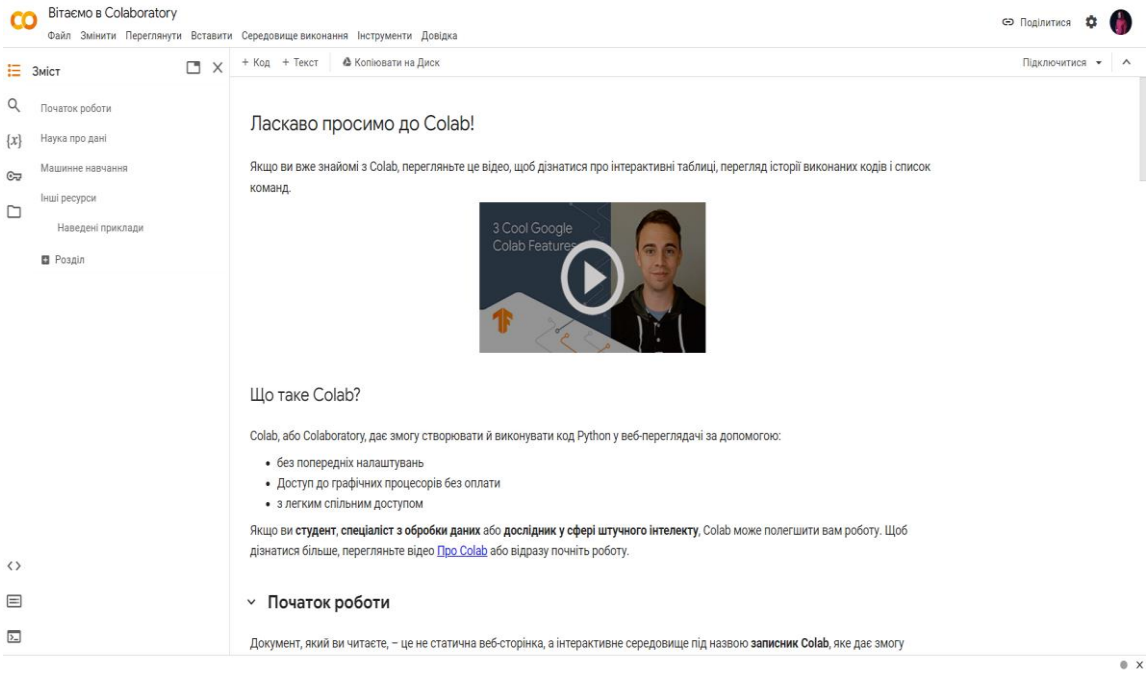


Рисунок 2.4— Частина інтерфейсу початкової сторінки Google Colab

Загалом, Google Colab є потужним інструментом для розробки, тестування та спільного користування проектами з використанням Python, особливо в областях машинного навчання, обробки даних, наукових досліджень та багатьох інших[6].

Моделювання бази знань в середовищі Google Colab - це важлива частина створення технології підтримки прийняття рішень щодо вибору ноутбуків на основі нечіткої логіки. Для зручності моделювання та обробки бази знань, можна використовувати бібліотеки, такі як NetworkX та scikit-fuzzy[6-7].

Спочатку, потрібно встановити ці бібліотеки, використовуючи наступні команди:

```
!pip install networkx
!pip install scikit-fuzzy
```

Після встановлення бібліотек, можна створити базу знань та моделювати нечітку логіку для визначення оптимального вибору ноутбуків на основі характеристик.

Google Colab надає можливість працювати з Python у віртуальному середовищі та легко виконувати різні обчислення, включаючи моделювання бази знань для технології підтримки прийняття рішень.

Після встановлення бібліотек NetworkX та scikit-fuzzy, можна розпочати створення бази знань та моделювання нечіткої логіки для технології підтримки прийняття рішень щодо вибору ноутбукі[6].

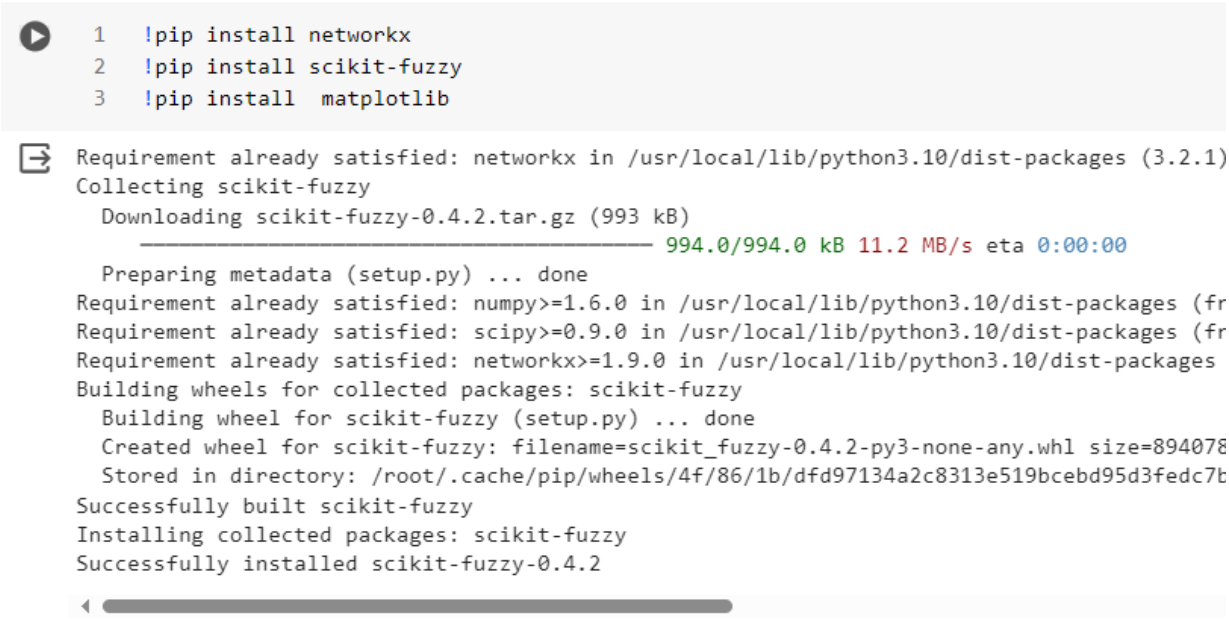


Рисунок 2.5— Встановлення бібліотек

1. Створення бази знань: Розробка бази знань передбачає визначення та організацію інформації про різні моделі ноутбуків, їх технічні характеристики та властивості. Використовуючи NetworkX, що створює графову структуру бази

знань, де вузли представляють ноутбуки, а ребра - зв'язки між ними на основі спільних характеристик.

2. Моделювання нечіткої логіки: Для визначення оптимального вибору ноутбуків використовуючи нечітку логіку.З scikit-fuzzy можна створити нечіткі функції належності для антецедентів та консиквентів, а потім визначити правила вибору на основі цих функцій.

3. Визначення оптимального вибору: Після створення бази знань та правил нечіткої логіки, можна подавати вхідні дані про характеристики користувача та використовувати цю систему для визначення оптимального ноутбуків, його процесора та відеокарти.

Рисунок 2.6 — Інтерфейсна складова розробки Google Colab

Використовуючи Google Colab, можна легко виконувати ці кроки, розробляти та тестувати систему . Вона покликана допомоготи користувачам

зробити інформований та оптимальний вибір серед широкого спектру ноутбуків, враховуючи їхні потреби та вимоги[6].

2.3.2 Розробка бази знань з використанням нечіткої логіки

Розробка бази знань з використанням нечіткої логіки є корисною для моделювання та роботи з нечіткими або суб'єктивними даними. Ця база знань може використовуватися для прийняття рішень, надання рекомендацій або аналізу нечітких відносин. Ось кроки, які можна виконати для розробки бази знань з використанням нечіткої логіки:

1. Визначення мети бази знань: Спочатку визначте, для яких цілей ви створюєте базу знань. Наприклад, це може бути система рекомендацій, система прийняття рішень або інструмент для аналізу нечітких даних.

Таблиця 2.1- Вхідні та вихідні змінні

Об’єкт	**Діапазон**	**Лінгвістичні терми**
RAM	4 to 64 GB	Very Low, Low, Medium, Good, Very Good
SSD	128 to 3072 GB	Very Low, Low, Medium, Good, Very Good
HDD	512 to 4000 GB	Very Low, Low, Medium, Good, Very Good
Keyboard	1 to 10	Very Bad, Bad, Average, Good, Very Good
Matrix Type	0 to 3	TN, IPS, OLED, Retina
USB Ports	0 to 5	None, Few, Average, Many, Very Many
Screen	1 to 4	Small, Medium, Large, Extra Large
Price	$200 to $3999	Very Low, Low, Medium, Good, Very Good
Laptop Name	0 to 14	Asus, Dell, Acer, MSI, HP, Lenovo, Apple, Xiaomi, Gygabite, Microsoft
Processor Name	0 to 30	Intel i3, i5, i7, i9; AMD Ryzen 3, 5, 7, 9
Graphics Card Name	0 to 30	Various NVIDIA GeForce and AMD Radeon models, Intel Graphics

2. Визначення вхідних та вихідних змінних: Визначте вхідні та вихідні змінні, які будуть використовуватися в базі знань. Вхідні змінні можуть бути характеристиками об'єктів або ситуацій, а вихідні - рекомендаціями, висновками або аналізом.

3. Створення нечітких множин та функцій приналежності: Для кожної вхідної змінної створіть нечіткі множини та функції приналежності. Наприклад, якщо ви моделюєте температуру, то створіть нечіткі множини для "низька", "середня", "висока" температура і визначте, які значення відповідають кожній з них.

1. `import pandas as pd`: Цей рядок імпортує бібліотеку Pandas і надає їй псевдонім "pd", що дозволяє використовувати її в коді.

2. `data`: Це словник, який містить дані для створення DataFrame. Кожен ключ (наприклад, "Component", "Range" та "Linguistic Terms / Specific Models") має своє значення, яке представляє собою список відповідних даних.

3. Створення DataFrame: Наступні рядки коду використовують бібліотеку Pandas для створення DataFrame на основі даних зі словника `data`. Кожен ключ у словнику стає назвою стовпця в DataFrame, а відповідний список значень стає даними у цьому стовпці.

4. `df = pd.DataFrame(data)`: Цей рядок створює DataFrame з ім'ям "df" на основі даних зі словника `data`.

5. `df`: Цей рядок виводить створений DataFrame, що дозволяє вам побачити його вміст у виведенні програми.

Результатом виконання цього коду буде таблиця даних (DataFrame), де кожен стовпець представляє одну з характеристик компонентів (Component), їх діапазон (Range) та лінгвістичні терміни.

```python
import pandas as pd

# Data for Antecedents and Consequents
data = {
    "Component": ["RAM", "SSD", "HDD", "Keyboard", "Matrix Type", "USB Ports", "Screen", "Price", "Laptop Name", "Processor Name", "Graphics Card Name"],
    "Range": ["4 to 64 GB", "128 to 3072 GB", "512 to 4000 GB", "1 to 10", "0 to 3", "0 to 5", "1 to 4", "$200 to $3999", "0 to 14", "0 to 30", "0 to 30"],
    "Linguistic Terms / Specific Models": [
        "Very Low, Low, Medium, Good, Very Good",
        "Very Low, Low, Medium, Good, Very Good",
        "Very Low, Low, Medium, Good, Very Good",
        "Very Bad, Bad, Average, Good, Very Good",
        "TN, IPS, OLED, Retina",
        "None, Few, Average, Many, Very Many",
        "Small, Medium, Large, Extra Large",
        "Very Low, Low, Medium, Good, Very Good",
        "Asus, Dell, Acer, MSI, HP, Lenovo, Apple, Xiaomi, Gygabite, Microsoft",
        "Intel i3, i5, i7, i9; AMD Ryzen 3, 5, 7, 9",
        "Various NVIDIA GeForce and AMD Radeon models, Intel Graphics"
    ]
}

# Create DataFrame
df = pd.DataFrame(data)
df

```

Рисунок 2.7 — Формування кодової структури відображення DataFrame

	Component	Range	Linguistic Terms / Specific Models
0	RAM	4 to 64 GB	Very Low, Low, Medium, Good, Very Good
1	SSD	128 to 3072 GB	Very Low, Low, Medium, Good, Very Good
2	HDD	512 to 4000 GB	Very Low, Low, Medium, Good, Very Good
3	Keyboard	1 to 10	Very Bad, Bad, Average, Good, Very Good
4	Matrix Type	0 to 3	TN, IPS, OLED, Retina
5	USB Ports	0 to 5	None, Few, Average, Many, Very Many
6	Screen	1 to 4	Small, Medium, Large, Extra Large
7	Price	$200 to $3999	Very Low, Low, Medium, Good, Very Good
8	Laptop Name	0 to 14	Asus, Dell, Acer, MSI, HP, Lenovo, Apple, Xiao...
9	Processor Name	0 to 30	Intel i3, i5, i7, i9; AMD Ryzen 3, 5, 7, 9
10	Graphics Card Name	0 to 30	Various NVIDIA GeForce and AMD Radeon models, ...

Рисунок 2.8 — Виведення даних, що використовуються в програмному застосунку

4. Створення правил нечіткої логіки: Визначте правила, які вказують, як взаємодіють вхідні змінні для отримання вихідних результатів. Наприклад, "Якщо температура середня і вологість висока, то рекомендувати включити кондиціонер".

```python
import pandas as pd

laptop_rules_data = {
    "Тип Ноутбука": ["Базовий", "Середній клас", "Ігровий", "Преміум", "Бюджетний", "Офісний", "Мультимедійний", "Тестове правило"],
    "RAM": ["дуже низька", "середня", "дуже висока", "дуже висока", "низька", "середня", "висока", "дуже низька"],
    "SSD": ["дуже низька", "середня", "хороша", "дуже висока", "низька", "хороша", "дуже висока", "дуже низька"],
    "HDD": ["дуже низька", "середня", "хороша", "дуже висока", "низька", "середня", "хороша", "дуже низька"],
    "Клавіатура": ["дуже погана", "середня", "хороша", "дуже хороша", "погана", "середня", "хороша", "дуже погана"],
    "Тип матриці": ["TN", "IPS", "OLED", "Retina", "TN", "IPS", "OLED", "TN"],
    "USB порти": ["жодного", "середнє", "багато", "дуже багато", "мало", "середнє", "багато", "мало"],
    "Екран": ["малий", "середній", "великий", "дуже великий", "малий", "середній", "великий", "малий"],
    "Ціна": ["дуже низька", "хороша", "хороша", "дуже хороша", "низька", "середня", "хороша", "дуже низька"],
    "Назва Ноутбука": ["Asus", "MSI", "MSI", "Asus", "Acer", "Hp", "Lenovo", "Asus"],
    "Назва Процесора": ["Intel i3", "Intel i5", "AMD Ryzen 7", "Intel i9", "Intel i3", "AMD Ryzen 3", "AMD Ryzen 5", "Intel i3"],
    "Назва Відеокарти": ["Intel UHD Graphics", "NVIDIA GeForce GTX 1650", "NVIDIA GeForce RTX 2060", "NVIDIA GeForce RTX 3070", "Intel UHD Graphics",
}

# Create DataFrame
df_laptop_rules = pd.DataFrame(laptop_rules_data)
df_laptop_rules
```

Рисунок 2.9 — Формування кодової структури відображення правил DataFrame

1 to 8 of 8 entries Filter

index	Тип Ноутбука	RAM	SSD	HDD	Клавіатура	Тип матриці	USB порти	Екран	Ціна	Назва Ноутбука	Назва Процесора	Назва Відеокарти
0	Базовий	дуже низька	дуже низька	дуже низька	дуже погана	TN	жодного	малий	дуже низька	Asus	Intel i3	Intel UHD Graphics
1	Середній клас	середня	середня	середня	середня	IPS	середнє	середній	хороша	MSI	Intel i5	NVIDIA GeForce GTX 1650
2	Ігровий	дуже висока	хороша	хороша	хороша	OLED	багато	великий	хороша	MSI	AMD Ryzen 7	NVIDIA GeForce RTX 2060
3	Преміум	дуже висока	дуже висока	дуже висока	дуже хороша	Retina	дуже багато	дуже великий	дуже хороша	Asus	Intel i9	NVIDIA GeForce RTX 3070
4	Бюджетний	низька	низька	низька	погана	TN	мало	малий	низька	Acer	Intel i3	Intel UHD Graphics
5	Офісний	середня	хороша	середня	середня	IPS	середнє	середній	середня	Hp	AMD Ryzen 3	Radeon Graphics
6	Мультимедійний	висока	дуже висока	хороша	хороша	OLED	багато	великий	хороша	Lenovo	AMD Ryzen 5	NVIDIA GeForce GTX 1650
7	Тестове правило	дуже низька	дуже низька	дуже низька	дуже погана	TN	мало	малий	дуже низька	Asus	Intel i3	Intel UHD Graphics

Show 25 per page

Рисунок 2.10 — Виведення результатів

1. `import pandas as pd`: Цей рядок імпортує бібліотеку Pandas та присвоює їй псевдонім "pd", щоб мати можливість використовувати її функції та об'єкти в коді.

2. `laptop_rules_data`: Це словник, який містить дані щодо правил для різних типів ноутбуків. Кожен ключ (наприклад, "Тип Ноутбуків", "RAM" та інші) має своє значення, яке представляє собою список відповідних даних для цього ключа.

3. Створення DataFrame: Наступний рядок коду використовує бібліотеку Pandas для створення DataFrame з назвою "df_laptop_rules" на основі даних зі

словника `laptop_rules_data`. Кожен ключ у словнику стає назвою стовпця в DataFrame, а відповідний список значень стає даними у цьому стовпці.

4. `df_laptop_rules`: Цей рядок виводить створений DataFrame, що дозволяє вам побачити його вміст у виведенні програми.

Результатом виконання цього коду буде таблиця даних (DataFrame), де кожен стовпець представляє одну з характеристик ноутбуків (наприклад, RAM, SSD, HDD і т. д.) і відповідні правила або оцінки для цих характеристик.

5. Агрегація правил: Використовуйте агрегацію правил для об'єднання нечітких висновків з різних правил.

6. Дефазифікація: Перетворіть нечіткі висновки у конкретні числові значення або текстові результати.

7. Збереження бази знань: Збережіть базу знань для подальшого використання в системі прийняття рішень або аналітичному інструменті.

8. Тестування та валідація: Перевірте та протестуйте базу знань на реальних даних або сценаріях, щоб переконатися, що вона працює правильно та надає корисні результати.

9. Оновлення та підтримка: Забезпечте механізми оновлення та підтримки бази знань, оскільки вимоги можуть змінюватися з часом.

Ці кроки допоможуть вам розробити базу знань з використанням нечіткої логіки, яка дозволить ефективно моделювати та аналізувати нечіткі дані та приймати рішення на їх основі.

2.4 Проектування модулів нечіткої логіки

Модулі нечіткої логіки допомагають враховувати нечіткість та суб'єктивність даних у процесі аналізу та прийняття рішень. Основні модулі програмного застосунку наведені на рисунках 2.11-2.12

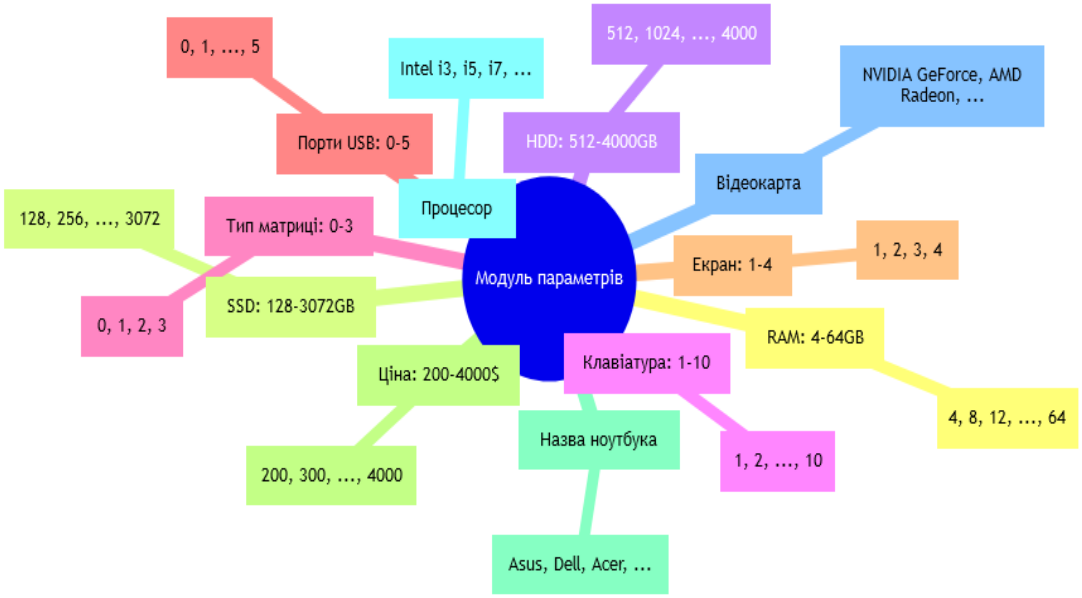


Рисунок 2.11— Модуль параметрів

Рисунок 2.12— Головний модуль програми

Модуль складається з наступних компонентів:

1. Антецеденти:

Антецеденти - це вхідні змінні, які впливають на вивід технології:

- `RAM`: діапазон від 4 до 64 з кроком 1.

- `SSD`: діапазон від 128 до 2048 з кроком 1.
- `HDD`: діапазон від 512 до 4096 з кроком 1.
- `diagonal`: масив зі значеннями [14, 15, 17].
- `matrix_type`: масив зі значеннями [0, 1, 2, 3].
- `keyboard`: діапазон від 0 до 10 з кроком 1.
- `usb_ports`: діапазон від 0 до 10 з кроком 1.
- `camera`: масив зі значеннями [0, 1].

2. Консеквенти:

Консеквенти - це вихідні змінні технології:

- `laptop_name`: діапазон від 0 до 10 з кроком 1.
- `processor_name`: діапазон від 0 до 10 з кроком 1.
- `graphics_card_name`: діапазон від 0 до 10 з кроком 1.

3. Функції належності для антецедентів:

Для кожного антецедента визначено декілька функцій належності:

- Для `RAM` є три функції належності: `low`, `medium` та `high`.
- Для `SSD` є три функції належності: `low`, `medium` та `high`.

4. Функції належності для консеквентів:

Для кожного консеквента визначено декілька функцій належності:

- Для `laptop_name` є функції належності для різних брендів ноутбуків.
- Для `processor_name` є функції належності для різних моделей процесорів.

Цей модуль представляє собою основу для технології нечіткої логіки, яка може бути використана для рекомендацій ноутбуків на основі вхідних параметрів. Функції належності допомагають визначити, наскільки вхідні

параметри відповідають певним категоріям, і на цій основі роблять висновки про вихідні параметри.

Модуль: `laptop_selection_ctrl`

Цей модуль, зосереджений на виборі ноутбуків на основі різних вхідних параметрів. Він використовує нечітку систему керування для прийняття рішень. Ось докладний опис:

1. Ініціалізація технології керування:

- `laptop_selection_ctrl = ctrl.ControlSystem([...])`: Цей рядок ініціалізує систему керування з набором правил. Є 30 правил (від `rule1` до `rule30`), які використовуються для визначення поведінки технології на основі різних вхідних умов.

2. Створення симуляції: - `laptop_selection = ctrl.ControlSystemSimulation(laptop_selection_ctrl)`: Створюється симуляція технології керування. Ця симуляція буде використовуватися для введення значень та отримання виводу на основі визначених правил[11-14]

.

3. Вхідні параметри:

- Це параметри, які користувач може встановити, щоб вказати свої уподобання щодо ноутбуків:

- `RAM`: Об'єм RAM в GB. Діапазон від 4GB до 64GB.

- `SSD`: Об'єм пам'яті SSD в GB. Діапазон від 128GB до 2048GB.

- `HDD`: Об'єм пам'яті HDD в GB. Діапазон від 512GB до 4048GB.

- `Diagonal`: Діагональ екрана в дюймах. Діапазон від 14 до 17 дюймів.

- `Matrix_Type`: Тип матриці дисплея. Це значення слайдера в діапазоні від 0 до 3, але точні типи, що відповідають цим числам, не надані.

- `Keyboard`: Якість або тип клавіатури. Це значення слайдера в діапазоні від 1 до 10.

- `USB_Ports`: Кількість USB портів. Діапазон від 0 до 10.

- `Camera`: Наявність камери. Це бінарне значення (0 або 1).

4. Встановлення вхідних значень:

- Вхідні значення для симуляції встановлюються за допомогою методу `laptop_selection.input`. Наприклад, `laptop_selection.input['RAM'] = RAM` встановлює значення RAM для симуляції.

5. Обчислення:

- `laptop_selection.compute()`: Цей рядок обчислює вивід на основі наданих вхідних значень та правил, визначених у системі керування.

6. Отримання результатів:

- Результати обчислення отримуються за допомогою методу `laptop_selection.output`. Модуль, , зацікавлений у трьох виводах:

- `Laptop Name`: Назва або модель рекомендованого ноутбуків.

- `Processor Name`: Назва або модель рекомендованого процесора для ноутбуків.

- `Graphics Card Name`: Назва або модель рекомендованої відеокарти для ноутбуків.

Модуль призначений для рекомендації конфігурації ноутбуків на основі уподобань користувача за допомогою нечіткої логічної технології керування. Користувач надає свої уподобання щодо різних функцій ноутбуків, і система обчислює рекомендацію на основі попередньо визначених правил.

1. Антецеденти (Antecedents):

Для кожного з вхідних параметрів, таких як RAM, SSD, HDD, Diagonal, Matrix Type, Keyboard, USB Ports та Camera, створено антецеденти,змінні, які представляють вхідні параметри і визначають їхні можливі значення.

Наприклад, для RAM антецедент визначається у такий спосіб:

```
RAM = ctrl.Antecedent(np.arange(4, 65, 1), 'RAM')
RAM['low'] = fuzz.trimf(RAM.universe, [4, 4, 32])
RAM['medium'] = fuzz.trimf(RAM.universe, [8, 32, 64])
RAM['high'] = fuzz.trimf(RAM.universe, [32, 64, 64])
```

У цьому випадку антецедент RAM приймає значення в діапазоні від 4 до 64, і він має три можливі категорії членства: "low", "medium" і "high". Функція trimf() визначає форму функції належності для кожної категорії.

Після досить коректного визначення антецедентів необхідно визначити вихідні параметри а саме консиквенти .

Для кожного з вихідних параметрів, таких як Laptop Name, Processor Name та Graphics Card Name та Price також створюються консиквенти. Консиквенти визначають вихідні параметри та їхні можливі значення. Наприклад, для Laptop Name консиквент визначається у такий спосіб:

```
laptop_name = ctrl.Consequent(np.arange(0, 15, 1), 'Laptop Name')
laptop_name['Asus'] = fuzz.trimf(laptop_name.universe, [0, 0, 1])
laptop_name['Dell'] = fuzz.trimf(laptop_name.universe, [0, 1, 2])
laptop_name['Acer'] = fuzz.trimf(laptop_name.universe, [1, 2, 3])
laptop_name['MSI'] = fuzz.trimf(laptop_name.universe, [2, 3, 4])
laptop_name['Hp'] = fuzz.trimf(laptop_name.universe, [3, 4, 5])
laptop_name['Lenovo'] = fuzz.trimf(laptop_name.universe, [4, 5, 6])
laptop_name['Aplle'] = fuzz.trimf(laptop_name.universe, [5, 6, 7])
laptop_name['Xiaomi'] = fuzz.trimf(laptop_name.universe, [6, 7, 8])
laptop_name['Gygabite'] = fuzz.trimf(laptop_name.universe, [7, 8,
9])
laptop_name['Microsoft'] = fuzz.trimf(laptop_name.universe, [8, 9,
10])
```

У цьому випадку консиквент Laptop Name приймає значення в діапазоні від 0 до 10, і він має категорії членства для кожного можливого бренду ноутбуків. Після того як було визначено вхідні та вихідні параметри потрібно створити ланцюги за допомогою яких система буде робити вибір, для спрощеня побудови ланцюгів в обраній бібліотеці передбачені правила.

3. Правила (Rules):

Правила визначаються, встановлючи логіку вибору вихідних параметрів на основі вхідних параметрів. Наприклад, ось фрагмент одного із правил:

```
   rules = [

ctrl.Rule(RAM['very low'] & SSD['very low'] & HDD['very low'] &
keyboard['very bad'] & matrix_type['TN'] & usb_ports['none'] &
screen['small'],
                  (prise['very low'], laptop_name['Asus'],
processor_name['Intel i3'], graphics_card_name['Intel UHD Graphics'])),
…
]
```

4. Створення контрольної технології (Control System):

За допомогою бібліотеки створюється контрольна система, яка об'єднує всі правила:

laptop_selection_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, ..., rule30])

Усі правила додаються до контрольної технології,абож ми просто підключаємо сиписок з параметрами rules=[] що містить усі правила.

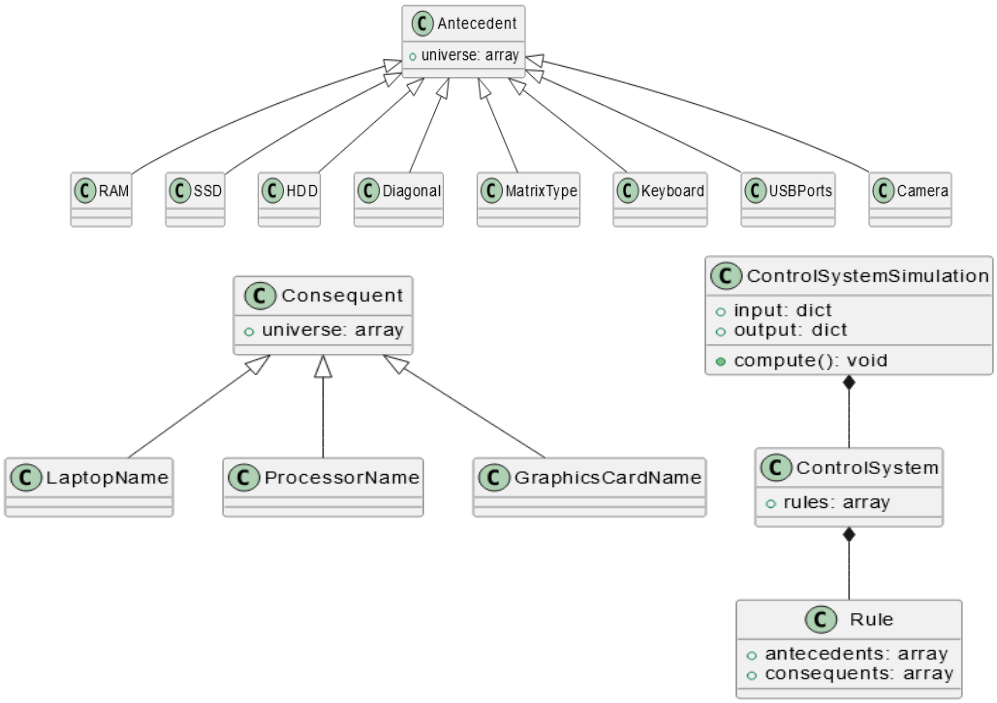


Рисунок 2.13— Фрагменти розробленої класової структури та її взаємозв’язки

5. Встановлення вхідних значень та обчислення результатів:

За допомогою симулятора контрольної технології встановлюються вхідні значення для антецедентів (наприклад, RAM, SSD, тощо), і потім обчислюються вихідні параметри на основі заданих вхідних параметрів та логіки правил:

```
laptop_selection.input['RAM'] = RAM
laptop_selection.input['SSD'] = SSD
laptop_selection.input['HDD'] = HDD
laptop_selection.compute()
Після виконання обчислень можна отримати результати:
print(laptop_selection.output['Laptop Name'])
print(laptop_selection.output['Processor Name'])
print(laptop_selection.output['Graphics Card Name'])
```

Ці результати покажуть, який ноутбук, процесор та відеокарта найбільше підходять на основі введених характеристик.

2.5 Висновок до розділу 2

Нечітка логіка виявляється надзвичайно корисною при розробці інформаційних технологій, призначених для вибору ноутбуків[12-14]. Основні висновки щодо цього включають:

1. Складність вибору ноутбуків: У сучасному світі, де існує велика кількість брендів, моделей та технічних характеристик, вибір ноутбуків може бути важливою задачею для користувачів.

2. Актуальність нечіткої логіки: Розробка нечітких інтелектуальних систем для вибору ноутбуків стає актуальною, оскільки вони дозволяють враховувати не лише конкретні значення характеристик ноутбуків, але й особисті вподобання користувача, його пріоритети та ступінь важливості кожної характеристики.

3. Ефективність прийняття рішень: Інформаційні технології на основі нечіткої логіки дозволяють ефективно підтримувати користувачів у процесі прийняття рішень, спрощуючи вибір ноутбуків, зменшуючи витрати часу і грошей на невідповідні моделі та забезпечуючи більш об'єктивний та раціональний підхід до вибору.

4. Значення для сучасних користувачів: Розробка систем на основі нечіткої логіки має велике значення для сучасних користувачів, які прагнуть знайти оптимальне рішення у світі швидкого технологічного прогресу.

У підсумку, використання нечіткої логіки допомагає користувачам зробити більш обдуманий і об'єктивний вибір, що відповідає їхнім потребам і вимогам в умовах сучасного ринку технологій.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ З ВИКОРИСТАННЯМ НЕЧІТКОЇ ЛОГІКИ

3.1 Обґрунтування вибору мови та середовища програмування для модулів нечіткої логіки

Вибір правильних інструментів значно спрощує процес розробки програмного продукту. декілька мов програмування, що підходять для роботи із нечіткою логікою зображено на Таблиці 3.1

Таблиця 3.1 – Порівняльна оцінка мов програмування

Мова програмування	Плюси	Мінуси
Python	Велика кількість бібліотек для штучного інтелекту та нечіткої логіки, легкість інтеграції	Повільніша продуктивність порівняно з компільованими мовами
C++	Висока продуктивність, можливість розробки ефективних алгоритмів	Складність управління пам'яттю, висока вимогливість до розробника
MATLAB	Вбудовані інструменти для наукових обчислень та нечіткої логіки, візуалізація даних	Дорожче в використанні, менш гнучка ніж загальні мови програмування
Java	Портативність, використовується у великих масштабних проектах	Менше фокусу на нечіткій логіці у стандартних бібліотеках
R	Сильна підтримка статистичних обчислень та аналізу даних	Використовується переважно у наукових та дослідницьких сферах, менш популярна для загальної розробки

На основі вищенаведеного таблиці можна підсумувати що обґрунтування вибору мови та середовища програмування для розробки модулів нечіткої логіки включає в себе розгляд декількох ключових аспектів, таких як вимоги до

продуктивності, доступні бібліотеки та інструменти, легкість використання та підтримка спільноти. Серед низки параметрів слід відокремити, такі параметри як:

Продуктивність

- C++ часто вибирають для високопродуктивних застосунків, оскільки він дозволяє точно управляти ресурсами та пам'яттю.

- Java також використовується у великих масштабних системах, забезпечуючи достатній баланс між продуктивністю та легкістю управління ресурсами.

Доступні Бібліотеки та Інструменти

- Python вирізняється завдяки широкому вибору бібліотек для нечіткої логіки, таких як Scikit-Fuzzy.

- MATLAB пропонує вбудовані інструменти для наукових обчислень і часто використовується в академічних та дослідницьких сферах.

Легкість Використання

- Python вважається однією з найлегших мов для вивчення, завдяки читабельному синтаксису і простоті використання.

- Ruby також є досить інтуїтивно зрозумілою і легкою для вивчення, хоча вона менш популярна в області нечіткої логіки.

Спільнота та Підтримка

- Python та Java мають великі, активні спільноти, що забезпечує хорошу підтримку та регулярні оновлення.

Обґрунтування Вибору Python

1. Широкий Вибір Бібліотек: Python має численні бібліотеки, спеціалізовані на нечіткій логіці та штучному інтелекті.

2. Легкість Інтеграції: Python легко інтегрувати з іншими інструментами та системами.

3. Спільнота та Підтримка: Активна спільнота та доступність ресурсів для навчання та підтримки.

4. Легкість Вивчення: Ідеальний для новачків завдяки простоті та читабельності.

5. Гнучкість: Підходить для різних типів проектів, від досліджень до виробничих систем.

Враховуючи ці фактори, Python часто є відмінним вибором для розробки модулів нечіткої логіки, особливо коли потрібно швидко розробити прототип або реалізувати комплексну систему з використанням доступних бібліотек і інструментів. Ще одним важливим аспектом є інтеграція з сервісами, такими як Google Colab.- Низький Поріг Входження: Google Colab дозволяє розробникам використовувати Python без необхідності налаштування локального середовища, що значно спрощує вступ до програмування та досліджень.- Безкоштовний Доступ до Обчислювальних Ресурсів: Google Colab надає безкоштовний доступ до обчислювальних ресурсів, включаючи GPU та TPU, що особливо корисно при роботі з великими даними та складними алгоритмами.- Сумісність з Jupyter Notebooks: Colab повністю сумісний з Jupyter Notebooks, що є популярним інструментом серед науковців та інженерів для створення та спільного використання документів, які містять код, рівняння, візуалізації та описовий текст.

- Легкість Спільної Роботи та Інтеграції з Іншими Сервісами Google: Google Colab спрощує спільну роботу та інтеграцію з іншими продуктами Google, такими як Google Drive та Google Sheets.Ці особливості роблять Google Colab чудовим вибором для освітніх, дослідницьких та прототипних проектів, особливо тих, які використовують Python і нечітку логіку. Він дозволяє швидко почати роботу, надаючи міцну основу для подальшої розробки та досліджень без необхідності інвестування у складне або дороге програмне забезпечення.

Також слід зазначити ,що обрана мова програмування має бути сумісною із завданнями. Python став популярною мовою для реалізації наукових та математичних досліджень завдяки своєму зручному синтаксису та великій кількості наукових бібліотек. В даному випадку, використання Python реалізувати відносну хорошу програму на основі нечіткої логіки.

Середовище Google Colab було обрано з міркувань доступності та зручності. Google Colab надає безкоштовний доступ до обчислювальних ресурсів, включаючи графічні процесори (GPU) та процесори Tensor Processing Unit (TPU), які можуть суттєво прискорити обчислення, особливо при роботі з великими об'ємами даних та складними обчисленнями.

Також варто відзначити важливість використання відповідних бібліотек у розробці нечітких систем. Бібліотека `numpy` була використана для роботи з числовими масивами та математичними операціями, що є важливим у процесі обчислення належностей та правил нечіткої логіки. `scikit-fuzzy` та `control` бібліотека та компоненти якої надають інструменти для створення та обчислення нечітких систем керування.

У підсумку, вибір мови програмування та середовища розробки був зроблений з урахуванням потреби у зручності, швидкості та ефективності розробки нечітких модулів. Використання Python та Google Colab разом з відповідними бібліотеками відіграло важливу роль у досягненні успішних результатів.

3.2 Основні етапи роботи програмного продукту на основі нечіткої логіки

Потреби користувачів стають все більш специфічними та різноманітними, важливою стає задача розробки гнучких та адаптивних програмних продуктів. Одним із підходів до створення таких систем є використання нечіткої логіки, яка дозволяє ефективно обробляти неоднозначність та невизначеність у вхідних даних. Нечітка логіка, відмінна від традиційної бінарної логіки, дозволяє системі приймати рішення на основі нечітких вхідних параметрів(Рисунок 3.2), що робить її оптимальною для моделювання складних систем, таких як вибір ноутбуків, де потрібно враховувати множину змінних та їх взаємодію.

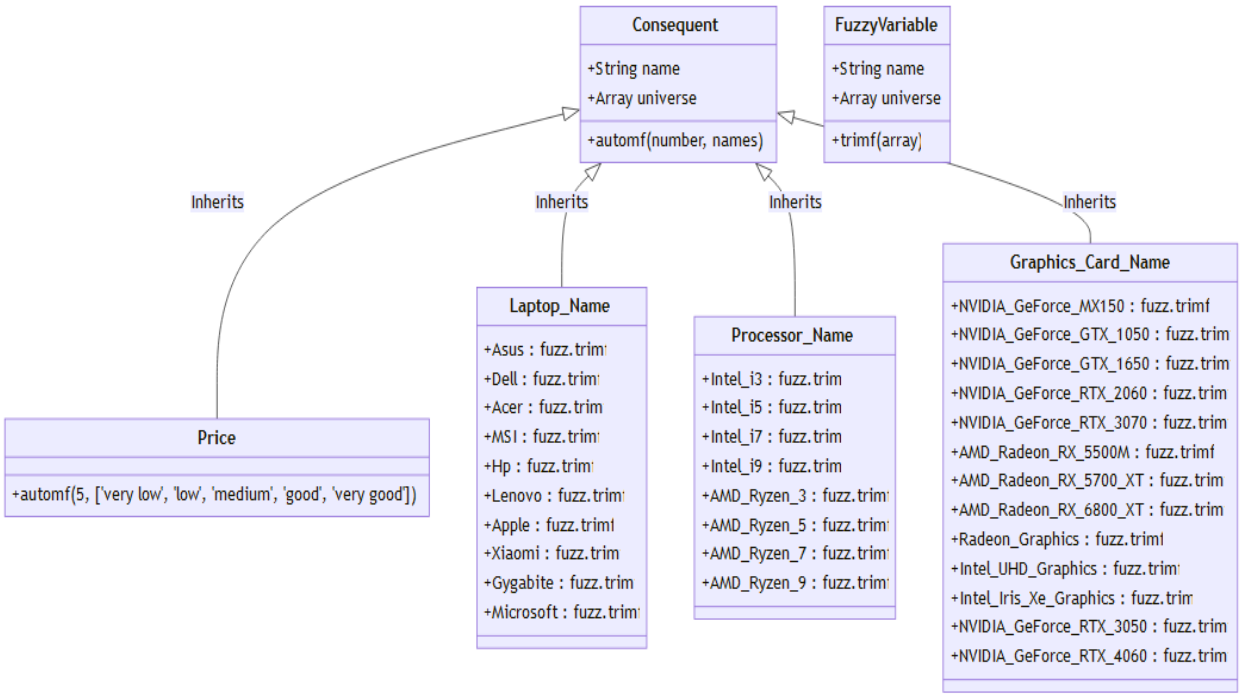


Рисунок 3.1 — UML діаграма надання рекомендацій на основі нечіткої логіки

Діаграма класів UML представляє систему нечіткої логіки для оцінки ноутбуків. Діаграма структурована на кілька класів, кожен з яких відповідає різним компонентам технології:

1. Класи Antecedent та Consequent: Це два основні класи, які представляють входи (Antecedents) та виходи (Consequents) технології нечіткої логіки. Вони мають атрибути, такі як `name` та `universe`, та метод `automf` (автоматична функція належності).

RAM, SSD, HDD, Keyboard, Matrix_Type, USB_Ports, Screen (Підкласи Antecedent): Ці підкласи успадковують від класу Antecedent. Кожен підклас представляє конкретний вхідний параметр для технології оцінки ноутбуків, такий як RAM, об'єм SSD тощо. Вони використовують метод `automf` для автоматичного створення лінгвістичних термінів (наприклад, 'дуже низький', 'низький', 'середній' тощо) для їх відповідних універсумів.

Price, Laptop_Name, Processor_Name, Graphics_Card_Name (Підкласи Consequent): Ці підкласи успадковують від класу Consequent. Вони представляють вихідні параметри технології, такі як ціновий діапазон, марка

ноутбуків, тип процесора та графічної карти. Підклас `Price`, подібно до деяких підкласів Antecedent, використовує `automf` для створення лінгвістичних термінів.

2. Клас FuzzyVariable: Цей клас, представляє загальну структуру для визначення нечітких змінних. Він включає атрибути для назви та універсуму змінної та метод `trimf` для визначення трикутних функцій належності.

Laptop_Name, Processor_Name, Graphics_Card_Name (Підкласи FuzzyVariable): Ці підкласи успадковують від класу FuzzyVariable.
Вони спеціалізуються на визначенні конкретних нечітких змінних, таких як марки ноутбуків, типи процесорів та моделі графічних карт. Кожен підклас має кілька атрибутів, що відповідають різним маркам або моделям, кожен з яких визначений трикутною функцією належності (`fuzz.trimf`).

Діаграма ефективно відображає структуру технології нечіткої логіки для оцінки ноутбуків, показуючи взаємозв'язки між різними входами та виходами та спосіб їх обробки для прийняття рішення. Використання успадкування ілюструє спільні характеристики між різними типами входів (Antecedents) та виходів (Consequents), тоді як конкретні підкласи деталізують унікальні аспекти кожного параметра в системі.

1. Діаграма Класів

Ця діаграма відображає структуру класів, які використовуються в системі, та їх взаємозв'язки.

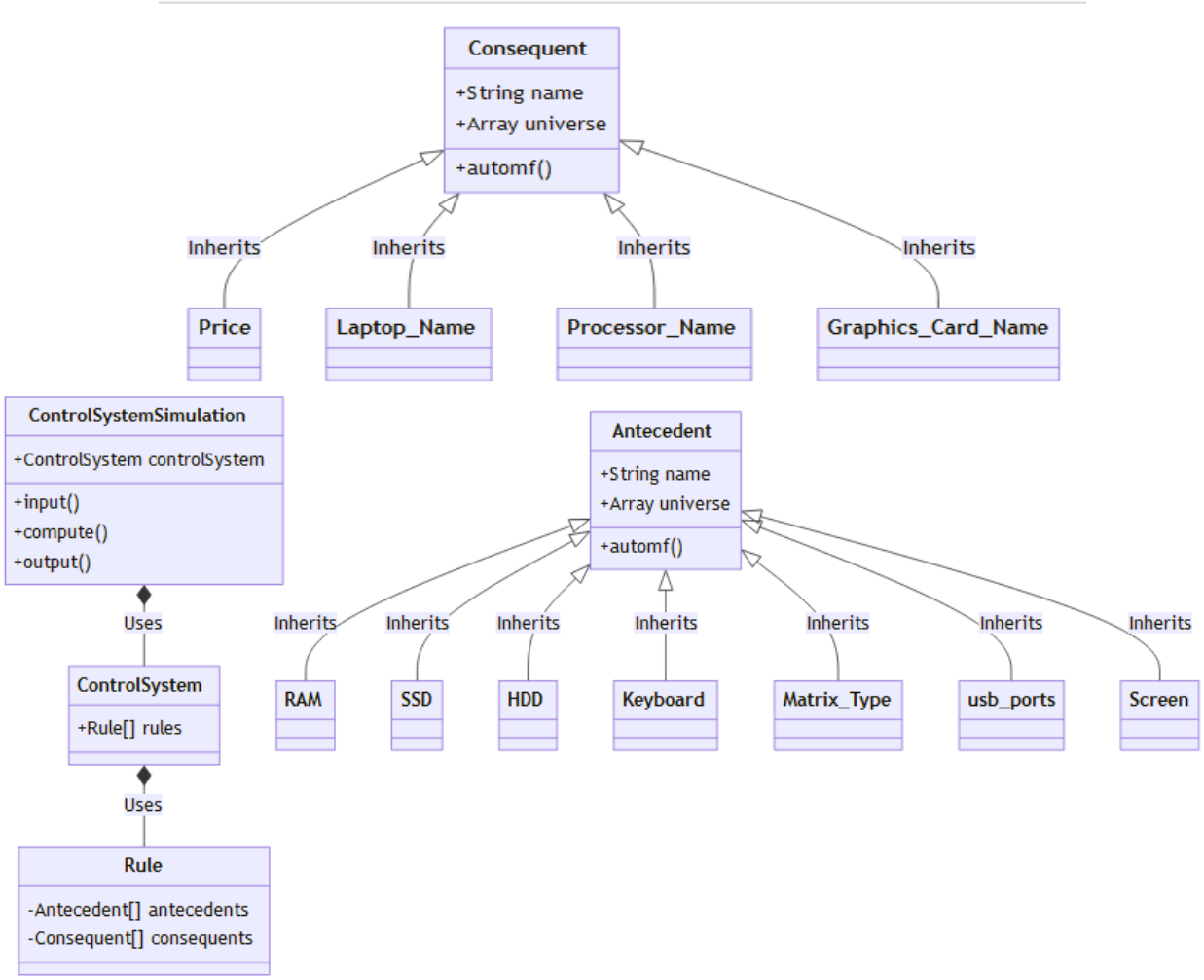


Рисунок 3.2— Фрагмент взаємодії технології обчислень

Класи та їх атрибути:

1. Antecedent:
   - `name`: String - Назва антецеденту.
   - `universe`: Array - Масив можливих значень для антецеденту.
   - Метод `automf()` - Автоматично генерує мовні терміни.

2. Consequent:
   - `name`: String - Назва консеквенту.
   - `universe`: Array - Масив можливих значень для консеквенту.
   - Метод `automf()` - Автоматично генерує мовні терміни.

3. Rule:
   - `antecedents`: Antecedent[] - Масив антецедентів.
   - `consequents`: Consequent[] - Масив консеквентів.

4. ControlSystem:
   - `rules`: Rule[] - Масив правил.

5. ControlSystemSimulation:
   - `controlSystem`: ControlSystem - Система контролю.
   - Методи `input()`, `compute()`, `output()` - Для введення даних, обчислення та отримання результатів.

Взаємозв'язки:

- Класи `RAM`, `SSD`, `HDD`, `Keyboard`, `Matrix_Type`, `usb_ports`, `Screen` є нащадками класу `Antecedent`.
- Класи `Price`, `Laptop_Name`, `Processor_Name`, `Graphics_Card_Name` є нащадками класу `Consequent`.
- Клас `ControlSystem` використовує клас `Rule`.
- Клас `ControlSystemSimulation` використовує клас `ControlSystem`.

2. Діаграма Послідовності (Sequence Diagram)

Ця діаграма відображає взаємодію між об'єктами в певному порядку часу.

Елементи Діаграми:

- Об'єкти: Представлені різними класами (наприклад, `RAM`, `SSD`).
- Лінії Життя: Вертикальні лінії, що представляють час існування об'єкта.

- Повідомлення: Горизонтальні стрілки, що показують взаємодію між об'єктами.
Процес Взаємодії:

1. Об'єкти вводять дані (наприклад, значення `RAM`).
2. Виконується обчислення (наприклад, в `ControlSystemSimulation`).
3. Генеруються результати (наприклад, вивід `Laptop Name`).

3. Діаграма Використання (Use Case Diagram)

Ця діаграма показує функціональність технології з точки зору користувача.

Елементи Діаграми:

- Актори: Зовнішні суб'єкти, які взаємодіють з системою (наприклад, користувач).
- Випадки Використання: Конкретні функції або завдання, які система може виконувати.

Процес Взаємодії:

1. Користувач вводить параметри (наприклад, `RAM`, `SSD`).
2. Система обробляє ці дані.
3. Користувач отримує рекомендації (наприклад, `Laptop Name`, `Price`).

Кожна з цих діаграм відображає різні аспекти технології: структуру класів, взаємодію між об'єктами та функціональність з точки зору користувача.

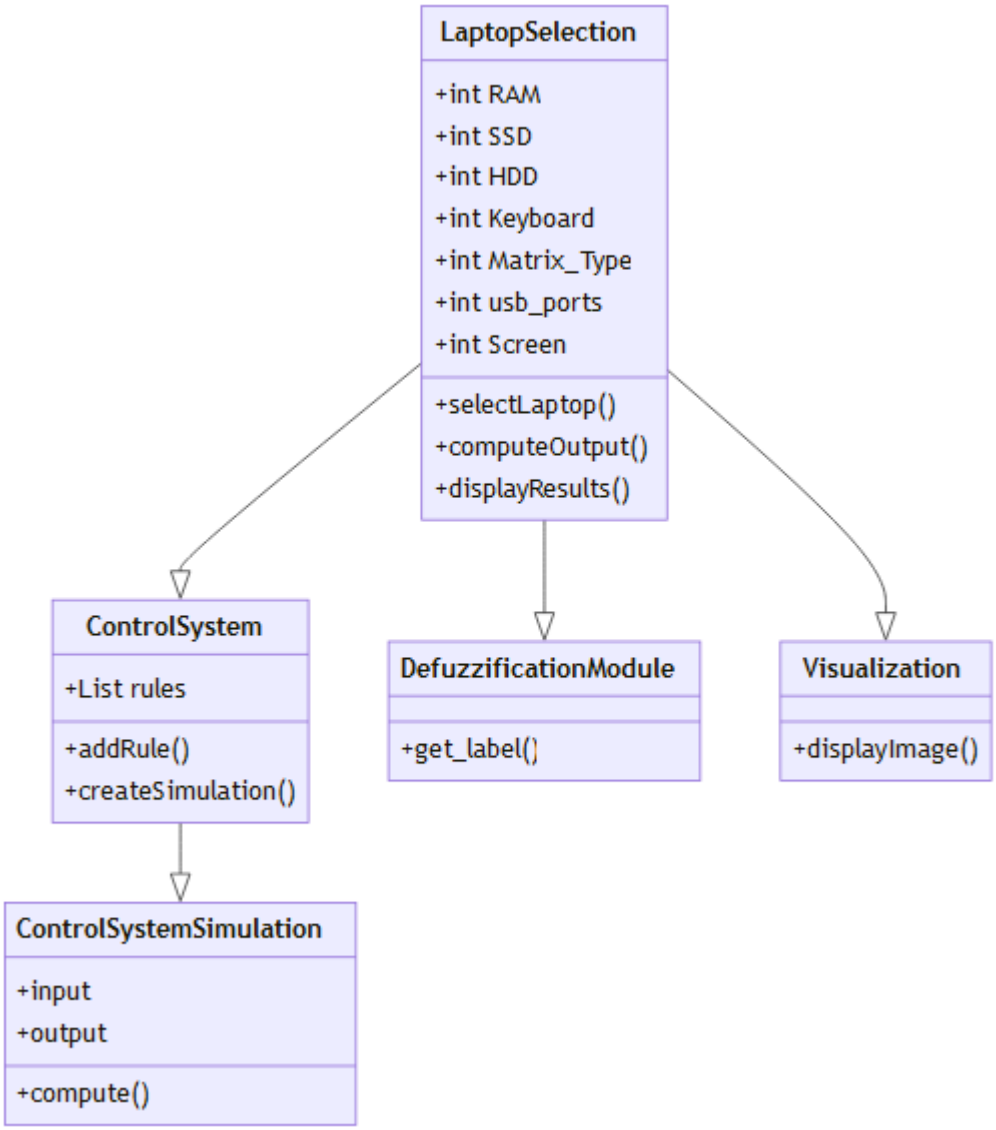


Рисунок 3.3— Фрагмент взаємодії консиквентів та технології обчислень

1. Клас `LaptopSelection` (Вибір ноутбуків):
   - Атрибути:
     - `RAM`, `SSD`, `HDD`, `Keyboard`, `Matrix_Type`, `usb_ports`, `Screen` - цілі числа, що представляють специфікації ноутбуків.
   - Методи:
     - `selectLaptop()`: Метод для вибору ноутбуків на основі заданих критеріїв.
     - `computeOutput()`: Обчислює вихідні дані на основі критеріїв вибору.
     - `displayResults()`: Відображає результати вибору ноутбуків.

2. Клас `ControlSystem` (Система контролю):

- Атрибут:
  - `rules`: Список правил для технології контролю.
- Методи:
  - `addRule()`: Додає нове правило до технології контролю.
  - `createSimulation()`: Створює симуляцію на основі правил технології контролю.

3. Клас `ControlSystemSimulation` (Симуляція технології контролю):
- Атрибути:
  - `input`: Вхідні значення для симуляції.
  - `output`: Вихідні значення симуляції.
- Метод:
  - `compute()`: Обчислює симуляцію з заданими вхідними даними.

4. Клас `DefuzzificationModule` (Модуль дефазифікації):
- Метод:
  - `get_label()`: Метод для отримання мітки для чіткого значення в системі нечіткої логіки.

5. Клас `Visualization` (Візуалізація):
- Метод:
  - `displayImage()`: Відображає зображення на основі вибору ноутбуків.

Відносини між класами:

- Наслідування (`--|>`):
  - `LaptopSelection` наслідує від `ControlSystem`.
  - `ControlSystem` наслідує від `ControlSystemSimulation`.
  - `LaptopSelection` наслідує від `DefuzzificationModule`.

- `LaptopSelection` наслідує від `Visualization`.

Ця діаграма представляє структуру технології вибору ноутбуків, показуючи, як різні класи пов'язані між собою та які атрибути та методи вони містять.

3.3 Тестування та аналіз результатів

Тепер для детального аналізу роботи розробленого застосунку встановимо необхідні пакети

1. `!pip install networkx`:
   - Ця команда встановлює бібліотеку `networkx`.
   - `networkx` - це Python бібліотека, яка використовується для створення, маніпулювання та вивчення структури, динаміки та функцій складних мереж.
   - Ця бібліотека широко використовується в областях, які вимагають аналізу графів, таких як соціальні мережі, аналіз мережевого трафіку, біологічні мережі тощо.

2. `!pip install scikit-fuzzy`:
   - Ця команда встановлює бібліотеку `scikit-fuzzy`.
   - `scikit-fuzzy` - це бібліотека, яка надає інструменти для роботи з нечіткою логікою та нечіткими системами в мові програмування Python.
   - Вона використовується для розробки систем, які потребують нечіткого управління або прийняття рішень, наприклад, в системах управління, аналізі даних, штучному інтелекті тощо.

3. `!pip install matplotlib`:
   - Ця команда встановлює бібліотеку `matplotlib`.
   - `matplotlib` - це бібліотека для створення статичних, анімованих та інтерактивних візуалізацій в Python.

- Вона широко використовується для візуалізації даних у різноманітних областях, включаючи науку про дані, інженерію, фінанси та багато інших.

Знак оклику (`!`) на початку кожної команди зазвичай використовується в Jupyter Notebook або інших інтерактивних середовищах Python для виконання команд оболонки замість Python коду.

Рисунок 3.4— Фрагмент встановлення необхідних компонентів

Код, представлений у запиті, описує процес вибору ноутбуків за допомогою технології контролю.

1. Ініціалізація параметрів: Код починається з ініціалізації різних параметрів ноутбуків, таких як RAM, SSD, HDD, тип клавіатури, тип матриці, кількість USB портів та тип екрану. Кожен параметр має певний діапазон значень, який може бути обраний користувачем через слайдери.

2. Створення технології контролю: Далі код створює систему контролю (`laptop_selection_ctrl`), використовуючи набір правил (змінна `rules`, яка не визначена в наданому фрагменті коду).

3. Створення симуляції: Після створення технології контролю, код ініціює симуляцію цієї технології (`laptop_selection_sim`).

4. Введення значень: Код вводить раніше визначені параметри (RAM, SSD, HDD і т.д.) у симуляцію технології контролю.

5. Обчислення виводу: Система контролю обчислює вивід на основі введених даних.

6. Виведення результатів: Нарешті, код виводить рекомендації щодо ноутбуків, включаючи рекомендований бренд, процесор, відеокарту та діапазон цін. Ці дані отримуються з виводу симуляції технології контролю.

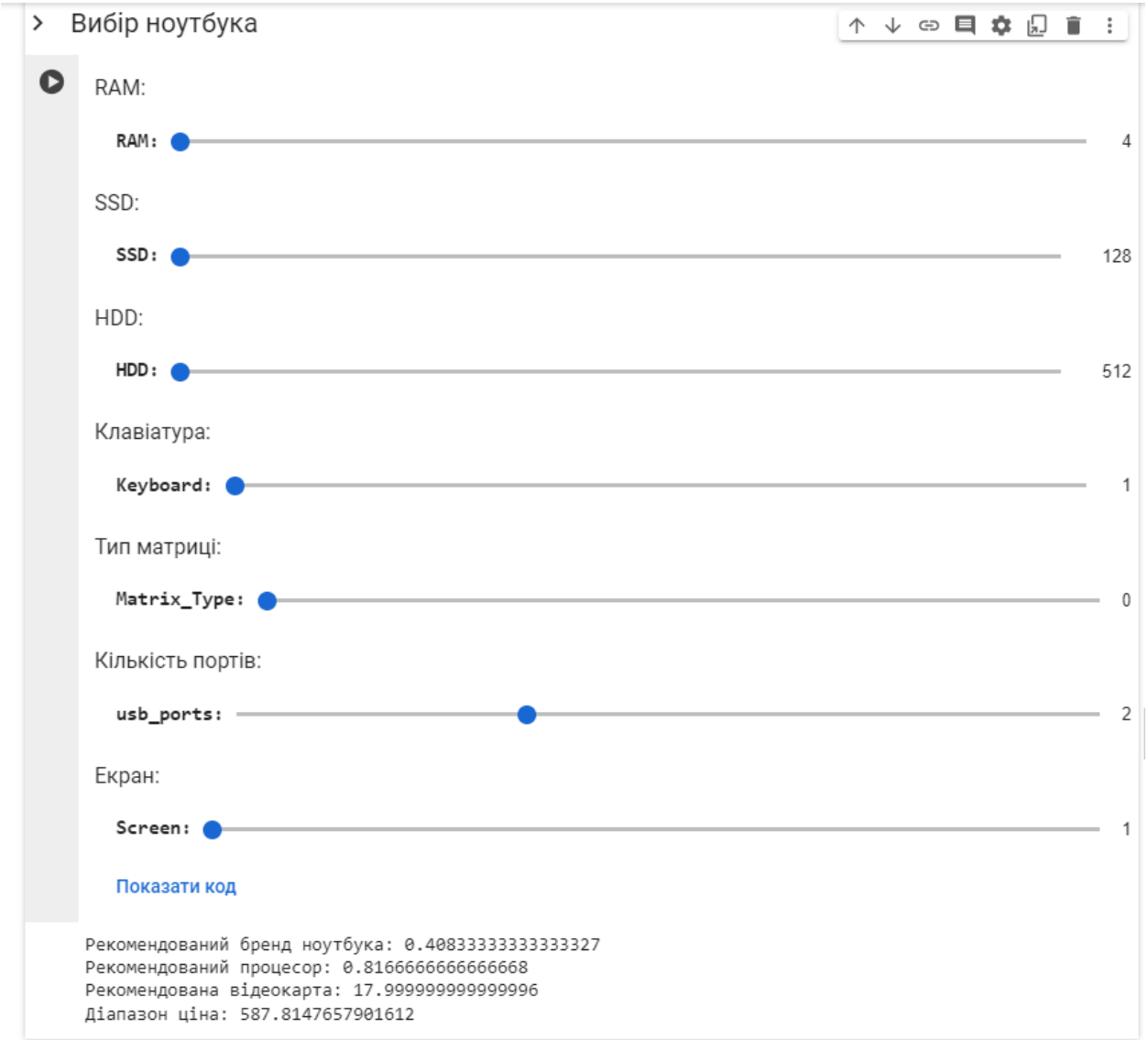


Рисунок 3.5— Виведення результатів в машинному форматі

Після чого функція `get_label`:

- Ця функція призначена для визначення найближчої мітки (label) до заданого числового значення (`crisp_value`) на основі членства в нечіткій множині (`antecedent`).

- Вона перебирає всі терміни (мітки) в антецеденті, обчислює відстань між значенням членства та `crisp_value`, і вибирає термін з найменшою відстанню.

3. Використання функції `get_label`:

- Функція `get_label` використовується для отримання міток для різних характеристик ноутбуків (бренд, процесор, відеокарта, ціновий діапазон) на основі результатів симуляції технології контролю.

4. Виведення результатів:

- Код виводить рекомендовані характеристики ноутбуків, використовуючи отримані мітки.наведено на рисунку 3.6

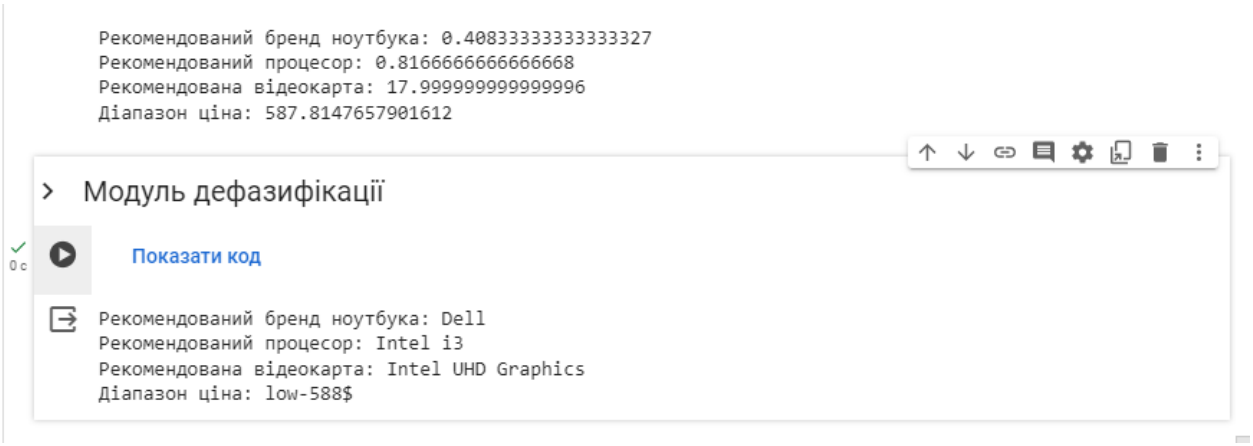


Рисунок 3.6 — Результат денацифікації даних

5. Словник зображень ноутбуків:

- Створюється словник `laptop_images`, який містить шляхи до зображень для різних брендів ноутбуків.Потрібин для виведеня конкретного ноутбуків

6. Відображення зображення вибраного ноутбуків:

- Код визначає, чи існує зображення для вибраного бренду ноутбуків.
- Якщо так, він відображає зображення за допомогою `matplotlib`.
- Якщо зображення відсутнє, виводиться повідомлення про його відсутність.

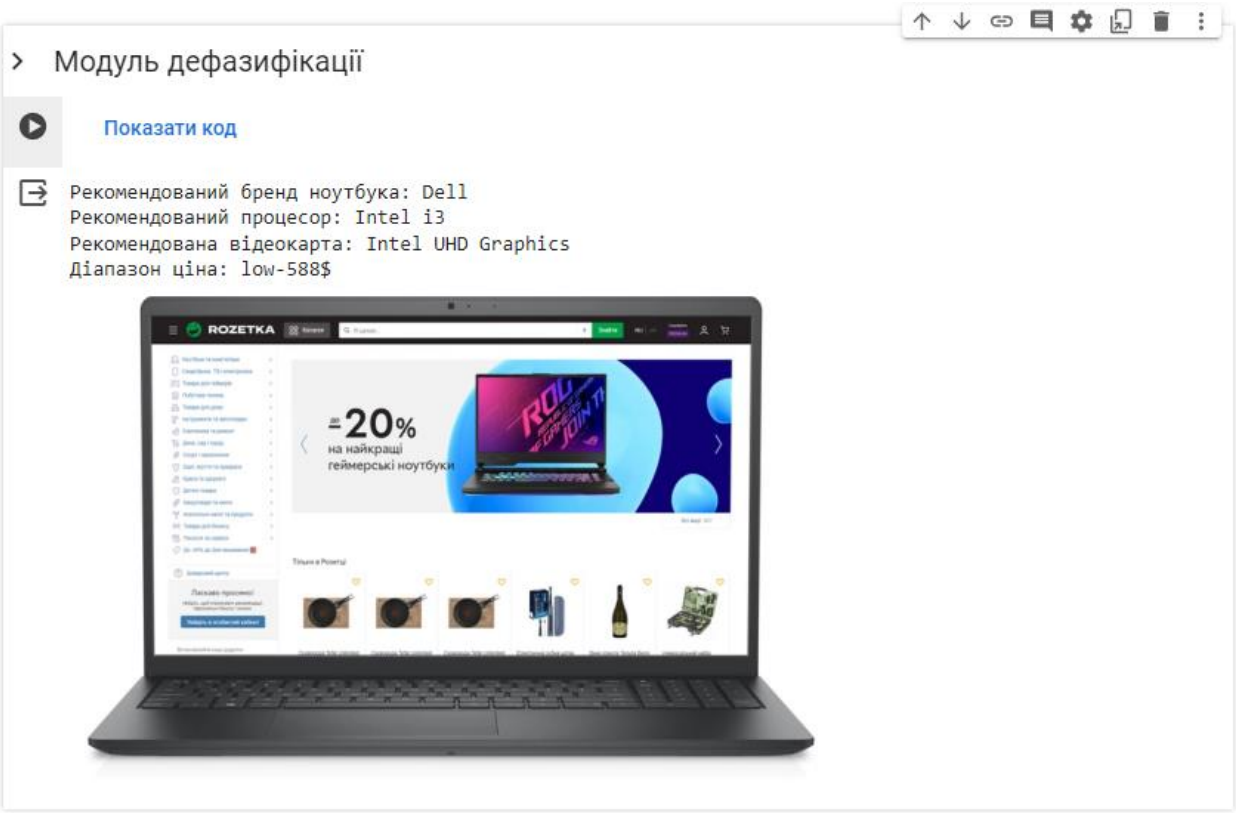


Рисунок 3.7 — Графічна складова виводу результату

Програма працює коректо та допомагає обрати підходяший ноутбук для більш глибокого тестування за допомогою експертів визначимо ефективність технології .Таблиця із результатами експетного тестування наведена в додатку В .

На базі експертних оцінок за тест кейсами можна визначити критерій ефективності нашої технології для цього ми побудуєм структуру та результат роботи Рисунок 3.10:

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

data = {
    'Тест кейс': ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11'],
    'Експерт1': {'Аналог': [8, 7, 8, 8, 8, 9, 9, 9, 8, 4, 9],
```

```
                        'Розробка': [9, 8, 9, 8, 8, 7, 9, 8, 7, 6, 8]},
            'Експерт2': {'Аналог': [6, 7, 9, 8, 7, 8, 8, 7, 7, 8, 6],
                        'Розробка': [5, 7, 8, 7, 8, 8, 9, 7, 8, 9, 8]},
        }
        df = pd.DataFrame(data['Експерт1'])
        df.set_index(pd.Index(data['Тест кейс']), inplace=True)
        fig, ax = plt.subplots(figsize=(14, 7))
        width = 0.35
        ind = pd.np.arange(len(df))
        p1 = ax.bar(ind - width/2, df['Аналог'], width, label='Аналог',
color='skyblue')
        p2 = ax.bar(ind + width/2, df['Розробка'], width, label='Розробка',
color='orange')
        ax.set_ylabel('Оцінки')
        ax.set_title('Різниця між аналогом та розробкою за тест кейсами')
        ax.set_xticks(ind)
        ax.set_xticklabels(df.index)
        ax.legend()

        for i in range(len(df)):
            diff = df['Розробка'][i] - df['Аналог'][i]
            ax.text(i, df['Аналог'][i] + diff/2, str(diff), ha='center',
va='bottom')
        plt.show()
```

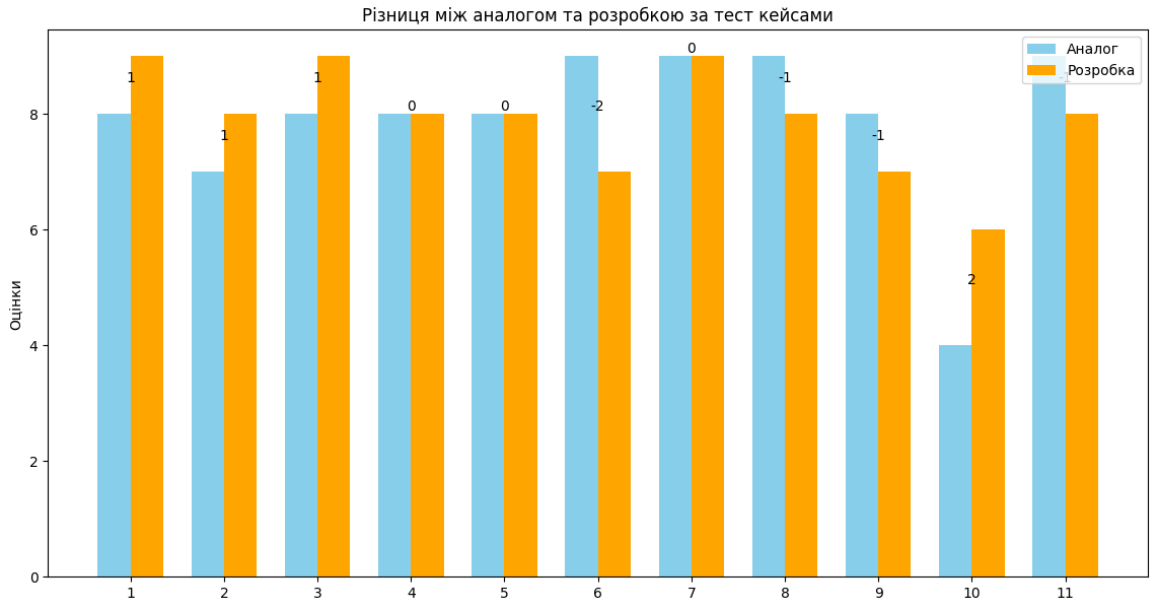


Рисунок 3.8 — Різниця між аналогом та розробкою за тест кейсами експерта1 та експерта2

Проаналізовавши отримані дані (Додаток В та Рисунок 3.8) визначено загальну кількість балів, а також середню кількість балів на по тест-кейсах, що дозволило визначити показник якості надання рекомендацій. В ході порівняння показника якості надання рекомендацій запропонованої розробки та аналогу визначено, що підвищення якості надання рекомендації становить 7.18% у порівняні з аналогом. Інформацію з детальним відображенням цих параметрів наведено у табл. 3.2.

Таблиця 3.2 – Таблиця визначення якості надання рекомендацій

Позначення	**Аналог**	**Розробка**
Сума усіх оцінок експертів	738	791
Середній бал за тест-кейс	82	87,888889
Середній бал оцінки експерта	7,454545	7,989899
Δ		7,18%

Також підчас тестування було визначені додаткові позитивні аспекти, що засвідчують перспективність розробленої інформаційної технології:

1. Гнучкість у прийнятті рішень: Система використовує нечітку логіку, що дозволяє обробляти неоднозначні та нечіткі вхідні дані, які часто зустрічаються при виборі ноутбуків (наприклад, переваги щодо бренду, ціни, характеристик). Це забезпечує більш гнучкий та адаптивний підхід до прийняття рішень.

2. Персоналізовані рекомендації: Система може адаптуватися до індивідуальних потреб користувачів, надаючи персоналізовані рекомендації на основі введених параметрів. Це підвищує задоволеність користувачів та ефективність вибору.

3. Точність та надійність: Використання продукційних правил та модуля дефазифікації забезпечує точність та надійність технології. Вона здатна точно інтерпретувати вхідні дані та надавати відповідні висновки.

4. Інтерактивність та зручність користування: Інтерфейс технології, який включає слайдери для вибору параметрів та візуалізацію результатів, робить процес вибору ноутбуків інтуїтивно зрозумілим та зручним для користувачів.

5. Масштабованість та адаптивність: Система легко адаптується до змін у ринкових умовах або користувацьких вимогах, що дозволяє їй залишатися актуальною та ефективною протягом тривалого часу.

6. Візуалізація даних**:** Використання графічних компонентів для відображення результатів робить інформацію більш доступною та зрозумілою для користувачів.

7. Швидкість обробки: Система здатна швидко обробляти вхідні дані та надавати результати, що забезпечує високу продуктивність та ефективність у виборі ноутбуків.

На основі проведеного тестування розробка демонструє достатню ефективність використання нечіткої логіки для розробки інтелектуальної технології, яка може адаптуватися до різноманітних вимог користувачів та надавати персоналізовані рекомендації. Система чітко працює згідно

розроблених продукційних ланок ,модуль дефазифікації робить дані краще для сприйняття. Система працює вірно та виконує поставлені перед нею задачі.

3.4 Висновок до розділу 3

Вибір мови програмування та середовища був зроблений з урахуванням їх здатності ефективно підтримувати реалізацію нечіткої логіки.Обрана мова програмування забезпечує гнучкість, широкий набір бібліотек та інструментів, що сприяє легкості розробки та інтеграції нечітких логічних модулів.Середовище програмування вибрано з огляду на його зручність, продуктивність та підтримку спільноти, що дозволяє ефективно розробляти, тестувати та впроваджувати нечіткі логічні технології.

Алгоритм розроблено з використанням принципів нечіткої логіки, що дозволяє системі ефективно обробляти неоднозначні та суб'єктивні дані.
Реалізація включає в себе визначення нечітких множин, правил та методів дефазифікації, що забезпечує точність та адекватність висновків технології.
Алгоритм демонструє гнучкість у прийнятті рішень, здатність адаптуватися до різних сценаріїв та вимог користувачів. Тестування показало достатню ефективність та надійність розробленої технології на основі нечіткої логіки.

Аналіз результатів демонструє достатню точність технології зокрема розробка демонструє вищу коректність надання рекомендацій ,що склала 7.18% у виборі оптимальних рішень на основі введених даних. Система досить ефективно впоралася з завданнями, демонструючи свою здатність до адаптації та персоналізації відповідно до потреб користувачів.

Реалізація демонструє реалізацію інформаційної технології з використанням нечіткої логіки. Від обґрунтованого вибору мови програмування до ефективного алгоритму та ретельного тестування, кожен аспект роботи підкреслює значення нечіткої логіки у розробці адаптивних та інтелектуальних систем.

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота на тему «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;

2) розраховано витрати на здійснення науково-технічної розробки;

3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної технології оцінювання за 12-ма критеріями, наведеними в табл. 4.1 [19].

Таблиця 4.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в	Технічні та споживчі влас-тивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в	Технічні та споживчі властивості продукту значно кращі, ніж в
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Продовження таблиці 4.1

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь- які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 4.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	5	5	5
2. Ринкові переваги (наявність аналогів)	3	3	3
3. Ринкові переваги (ціна продукту)	3	4	3
4. Ринкові переваги (технічні властивості)	3	3	3
5. Ринкові переваги (експлуатаційні витрати)	3	3	3
6. Ринкові перспективи (розмір ринку)	4	3	3
7. Ринкові перспективи (конкуренція)	3	2	2
8. Практична здійсненність (наявність фахівців)	5	5	5
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	4	5	5
11. Практична здійсненність (термін реалізації)	5	4	5
12. Практична здійсненність (розробка документів)	5	5	4
Сума балів	45	45	43
Середньоарифметична сума балів $СБ_c$	44,3		

За результатами розрахунків, наведених в таблиці 4.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 4.3 [19].

Таблиця 4.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41…48	Високий
31…40	Вище середнього
21…30	Середній
11…20	Нижче середнього
0…10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків» становить 44,3 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

4.2 Оцінювання рівня новизни розробки

Виводячи на ринок новинку виробник вважає, що тієї новизни, якою наділена нова розробка є достатньо для того, щоб вона була сприйнята споживачем як нова. Але це не завжди так, в силу того, що споживач і виробник неоднозначно визначать її рівень новизни. Тому доцільним є визначення рівня новизни розробки отриманої в результаті досліджень за темою «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків».

Саме визначення рівня і ступеня інтегральної новизни є найбільш актуальним, оскільки її рівень визначає ступінь однакового позитивного сприйняття новизни розробки як виробником, так і споживачем, а отже і ринком в цілому, а це, у свою чергу, є гарантією того, що новинка знайде своє місце на ринку, користуватиметься попитом у споживачів і забезпечить відшкодування витрат, зазнаних товаровиробником під час розроблення та виробництва технічної розробки [19,20].

Рівень новизни нової продукції розраховуємо експертним методом шляхом протиставлення нової продукції та її аналогів, що існують в даний час на ринку, за чинниками що визначають її значення, в системі «краще-гірше». Рівень новизни встановлюємо відносно рівня аналога (або продукту, що досить близький до аналога).

Для визначення *i*-го виду новизни, застосуємо чинники, які впливають на її рівень. Кожен чинник *i*-го виду новизни розраховуємо в балах. Більша кількість набраних балів свідчить про більший рівень новизни. Для оцінювання рівня новизни використаємо думки експертів, які встановлюють визначені бали відповідним чинникам. Бал відповідності проставляється в діапазоні від (-5 – значно гірше аналога до +5 – значно краще аналога). Результати попереднього оцінювання зведемо до відповідного листа оцінювання (таблиця 4.4).

Таблиця 4.4 – Лист оцінювання рівня новизни експертами

Види та чинники		Бали та експерти		
		Експерт 1	Експерт 2	Експерт 3
1		*2*	*3*	*4*
Споживча новизна	Питома вага 0,26	Максимальний бал $Б_{i\ MAX}$		25
1. Зміна поведінкових звичок споживача		2	2	2
2. Ступінь задоволення потреб і запитів		4	4	4
3. Спосіб задоволення потреби		2	2	2
4. Формування нової потреби		4	3	3
5. Формування нового споживача		0	0	0
Середній бал експертів $Б_{i\ отр}$		11		
Товарна новизна	Питома вага 0,21	Максимальний бал $Б_{i\ MAX}$		30
1. Параметричні зміни показників продукції				
1.1. Якісні		3	3	4
1.2. Технічні		3	4	3
1.3. Економічні		3	3	3
1.4. Сервісні		4	4	4
2. Якість продукції по відношенню до конкурентів		3	3	3
3. Функціональні зміни		4	4	4
Середній бал експертів $Б_{i\ отр}$		21		
Виробнича новизна	Питома вага 0,014	Максимальний бал $Б_{i\ MAX}$		25
1. Рівень унікальності товару для підприємства		5	5	5
2. Рівень унікальності для галузі		3	3	3
3. Рівень унікальності товару для країни		1	1	1
4. Зміна виробничої технології		4	4	4
5. Відносно існуючого асортименту		3	2	3
Середній бал експертів $Б_{i\ отр}$		16		
Прогресивна новизна	Питома вага 0,2	Максимальний бал $Б_{i\ MAX}$		25
1. Зміна технології виготовлення		4	4	4
2. Рівень застосування нових компонентів і матеріалів		0	0	0
3. Зміна технологічного принципу дії виробу		1	2	1
4. Зміна конструктивного виконання		3	3	3
5. Рівень застосування інновацій		2	2	2
Середній бал експертів $Б_{i\ отр}$		10		
Ринкова новизна	Питома вага 0,1	Максимальний бал $Б_{i\ MAX}$		20
1. Новий виріб на новому ринку		0	0	0
2. Новий виріб на відомому ринку		4	4	4
3. Модернізований виріб		3	3	3
4. Нова модель		1	1	1
Середній бал експертів $Б_{i\ отр}$		8		
Екологічна новизна	Питома вага 0,035	Максимальний бал $Б_{i\ MAX}$		20
1. Рівень екологічної чистоти технології виробництва		5	5	5

Продовження таблиці 4.4

2. Рівень впровадження мало- та безвідходних технологій		5	5	5
3. Рівень екологічно небезпечних режимів експлуатації продукції		5	5	5
4. Рівень забруднення навколишнього середовища		5	5	5
Середній бал експертів $Б_{i\ отр}$		20		
Соціальна новизна	Питома вага 0,036	Максимальний бал $Б_{i\ MAX}$		20
1. Використання нового товару приводить до покращення стану здоров'я нації		0	0	0
2. Використання нового товару приводить до зростання доходів населення		0	0	0
3. Виробництво нового товару приводить до збільшення (зменшення) кількості робочих місць на підприємстві		4	5	4
4. Виробництво нового товару приводить до підвищення кваліфікації персоналу		3	4	3
Середній бал експертів $Б_{i\ отр}$		8		
Маркетингова новизна	Питома вага 0,145	Максимальний бал $Б_{i\ MAX}$		20
1. Нові методи маркетингових досліджень		0	0	0
2. Вживання нових стратегій сегментації ринку		3	3	3
3. Вибір нової маркетингової стратегії обхвату і розвитку цільового сегмента		1	1	1
4. Побудова нових каналів збуту		2	1	1
Середній бал експертів $Б_{i\ отр}$		5		

Значення *i*-го виду новизни розрахуємо за формулою [20]:

$$I_i = \frac{Б_{i\,отр}}{Б_{i\,MAX}}, \tag{4.1}$$

де $Б_{i\ отр}$ – отримана кількість балів за шкалою оцінок чинників, що визначають *i*-й вид новизни;

$Б_{i\ MAX}$ – максимальна кількість балів, що може бути отримана за *i*-м видом новизни.

Загальний рівень інтегральної новизни розраховуємо шляхом перемноження отриманого значення *i*-го виду новизни на її вагомість, причому вагомість *i*-го виду новизни визначаємо експертним методом, за формулою [20]:

$$N_{інт} = \sum_{i}^{n} W_i \cdot I_i, \tag{4.2}$$

де $N_{інт}$ – рівень інтегральної (сукупної) новизни;

W_i – вагомість (питома вага) *i*-го виду новизни;

n – загальна кількість видів новизни.

$$N_{інт} = (0,26 \cdot 11/25)+(0,21 \cdot 21/30)+(0,014 \cdot 16/25)+(0,2 \cdot 10/25)+(0,1 \cdot 8/20)+ (0,035 \cdot 20/20)+(0,036 \cdot 8/20)+(0,145 \cdot 5/20) = 0,481.$$

Отримане значення інтегрального рівня новизни зіставляємо зі шкалою, що наведена в табл. 4.5 [19,20].

Таблиця 4.5 – Рівні новизни нового товару та їхня характеристика

Рівні новизни товару	Значення інтегральної новизни	Характеристика товару	Вид нового товару
Найвища	1,00	Абсолютно новий товар	Новий товар, що наділений ознаками інноваційності (інноваційний товар)
Висока	0,8…0,99	Товар, який не має аналогів	
Значуща	0,6…0,79	Принципова зміна споживчих властивостей товару	
Достатня	0,4…0,59	Принципова технологічна модифікація товару	
Незначна	0,2…0,39	Кардинальна зміна параметрів	Новий товар
Помилкова	0,00…0,19	Малоістотна модифікація	

Згідно таблиці 4.5 розробка відповідає рівню при значенні інтегральної новизни 0,481 - достатня новизна; за характеристикою: принципова технологічна модифікація товару; вид розробки - новий товар, що наділений ознаками інноваційності (інноваційний товар).

4.3 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення розрахуємо за формулою [20]:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i \,, \tag{4.3}$$

де k – кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i – коефіцієнт, який враховує питому вагу *i-го* технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і при цьому має виконуватись умова $\sum_{i=1}^{k} \alpha_i = 1$;

β_i – відносне значення *i-го* технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{нi}}{I_{ai}} \,, \tag{4.4}$$

де $I_{нi}$ та $I_{на}$ – чисельні значення конкретного *i*-го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{нi}} \,; \tag{4.5}$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 4.6.

Таблиця 4.6 – Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірю-вання	Аналог	Проектоване програмне забезпечення	Відношення параметрів нової розробки до аналога	Питома вага показника
Оперування стандартними даними	бал	7	9	1,28	0,35
Можливість працювати з нечіткими параметрами	бал	1	7	7	0,3
Ідеальний експерт	бал	2	8	4	0,1
Професійне надання рекомендацій необізнаному користувачеві	бал	2	9	4,5	0,1
Стабільний доступ в інтернет в інтернет	бал	9	9	1	0,15

Узагальнений коефіцієнт якості ($B_н$) для нового технічного рішення складе:

$$B_н = \sum_{i=1}^{k} \alpha_i \cdot \beta_i = 1{,}28 \cdot 0{,}35 + 7 \cdot 0{,}3 + 4 \cdot 0{,}1 + 4{,}5 \cdot 0{,}1 + 1 \cdot 0{,}15 = 3{,}55.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 3,55 рази.

4.4 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов’язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків. Частина 2. Нечітка логіка», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.4.1 Витрати на оплату праці

До статті «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.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників ($З_о$) розраховуємо у відповідності до посадових окладів працівників, за формулою [19]:

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p}, \qquad (4.6)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, T_p=22 дні.

$$З_о = 18250{,}00 \cdot 28 / 22 = 23227{,}27 \; грн.$$

Проведені розрахунки зведемо до таблиці.

Таблиця 4.7 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	18250,00	829,55	28	23227,27
Відповіданьний виконавець (інженер-програміст вищої категорії)	16800,00	763,64	28	21381,82
Консультант-аналітик технології нечітких множин	17300,00	786,36	5	3931,82
Всього				48540,91

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників ($З_р$) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків. Частина 2. Нечітка логіка» розраховуємо за формулою:

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i , \qquad (4.7)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

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

$$C_i = \frac{М_М \cdot K_i \cdot K_c}{T_р \cdot t_{зм}} , \qquad (4.8)$$

де $М_М$ – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо *Мм*=6700,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [19];

$К_с$ – 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.

$Т_р$ – середнє число робочих днів в місяці, приблизно $Т_р$ = 22 дн;

$t_{зм}$ – тривалість зміни, год.

$С_1$ = 6700,00 · 1,10 · 1,35 / (22 · 8) = 56,53 *грн.*

$З_{р1}$ = 56,53 · 5,00 = 282,66 *грн.*

Таблиця 4.8 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Встановлення обладнання робочих місць інженерно-технічних працівників (розробників і дослідників)	5,00	2	1,10	56,53	282,66
Установка обчислювального обладнання для проведення моделювання та досліджень галузі нечіткої логіки	6,50	3	1,35	69,38	450,97
Встановлення програмного забезпечення інструментарію розробки технології нечіткої логіки	7,50	5	1,70	87,37	655,25
Навчання технології нечіткої логіки	11,00	4	1,50	77,09	847,97
Всього					2236,84

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 … 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{Н_{дод}}{100\%}, \quad (4.9)$$

де $Н_{дод}$ – норма нарахування додаткової заробітної плати. Приймемо 10%.

$З_{дод}$ = (48540,91 + 2236,84) · 10 / 100% = 5077,77 грн.

4.4.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_н = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (4.10)$$

де $Н_{зп}$ – норма нарахування на заробітну плату. Приймаємо 22%.

Зн = (48540,91 + 2236,84 + 5077,77) · 22 / 100% = 12288,21 грн.

4.4.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків. Частина 2. Нечітка логіка».

Витрати на матеріали (*М*), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$М = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j - \sum_{j=1}^{n} В_j \cdot Ц_{в\,j}, \quad (4.11)$$

де $Н_j$ – норма витрат матеріалу *j*-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу *j*-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = *1,1 … 1,15*);

B_j – маса відходів *j*-го найменування, кг;

$Ц_{вj}$ – вартість відходів *j*-го найменування, грн/кг.

$M_1 = 1,0 \cdot 195,00 \cdot 1,1 - 0 \cdot 0 = 214,50$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.9 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за од, грн	Норма витрат,	Величина відходів, од	Ціна відходів, грн/од	Вартість витраченого матеріалу, грн
ПАПІР CANYO A4 STAND (PAPER_MS80/MS .A4.80.ST)	195,00	1,0	0	0	214,50
Папір CANYO A5, 80 г, 500 арк. Premium, клас А (149366)	120,00	2,0	0	0	264,00
Настільний набір Buromax 16 items, black (BM.6302-01)	210,00	2,0	0	0	462,00
Органайзер настільний металевий, 22x14x13 см, чорний H-Tone (JJ41220)	240,00	2,0	0	0	528,00
Картридж Canon 725 Black (3484B002)	1230,00	1,0	0	0	1353,00
Диск DVD Verba 4.7Gb 16X SlimBox 1шт Matte AZO (43547-1disk)	30,00	2,0	0	0	66,00
USB накопичувач DRIVE 128GB Silver (L)	240,00	2,0	0	0	528,00
Всього					3415,50

4.4.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі *($К_в$)*, які використовують при проведенні НДР на тему «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків. Частина 2. Нечітка логіка», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$К_в = \sum_{j=1}^{n} Н_j \cdot Ц_j \cdot К_j \quad (4.12)$$

де $Н_j$ – кількість комплектуючих j-го виду, шт.;

$Ц_j$ – покупна ціна комплектуючих j-го виду, грн;

$К_j$ – коефіцієнт транспортних витрат, ($К_j$ = *1,1 … 1,15*).

$К_в = 1 \cdot 2369{,}00 \cdot 1{,}1 = 2605{,}90$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.10 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Зовнішній жорсткий диск 2.5" 2TB TOSHIBA (HDTB520EK3AA)	1	2369,00	2605,90
NVidia GeForce GTX 1050, 4 ГБ	1	6520,00	7172,00
Всього			9777,90

4.4.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$В_{спец} = \sum_{i=1}^{k} Ц_i \cdot С_{пр.i} \cdot К_i \ , \quad (4.13)$$

де $Ц_i$ – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$С_{пр.i}$ –кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

$К_i$ – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($К_i$ = *1,10…1,12);*

k – кількість найменувань устаткування.

$В_{спец} = 13800{,}00 \cdot 1 \cdot 1{,}1 = 15180{,}00$ *грн.*

Отримані результати зведемо до таблиці:

Таблиця 4.11 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Джерело безперебійного живлення LPE-B-PSW-2300BA/1600Вт	1	13800,00	15180,00
Всього			15180,00

4.4.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прг} = \sum_{i=1}^{k} Ц_{iпрг} \cdot C_{прг.i} \cdot K_i \ , \qquad (4.14)$$

де $Ц_{iпрг}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{прг.i}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1,10…1,12$);

k – кількість найменувань програмних засобів.

$B_{прг} = 14300,00 \cdot 1 \cdot 1,1 = 15730,00$ грн.

Отримані результати зведемо до таблиці:

Таблиця 4.12 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Прикладне середовище розробки ПЗ на основі застосування технології нечітких множин	1	14300,00	15730,00
Програмне забезпечення з відкритим кодом	1	2100,00	2310,00
Всього			18040,00

4.4.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_б}{T_в} \cdot \frac{t_{вик}}{12}, \quad (4.15)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$A_{обл}$ = (92569,00 · 2) / (3 · 12) = 5142,72 грн.

Проведені розрахунки зведемо до таблиці.

Таблиця 4.13 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Електронно-обчислювальний центр моделювання та тестування технології КОМП'ЮТЕР HP ELITEONE 870 G9 AIO / I7-13700 (7B0P5EA)	92569,00	3	2	5142,72
Робоче місце Project Manager 2/0	9899,00	5	2	329,97
Робоче місце інженера-дослідника Ноутбук Lenovo IdeaPad 330-15ICH (81FK00G1RA)	16599,00	3	2	922,17

Продовження таблиці 4.13

Пристрій виводу інформації CANON Laser 6900	8699,00	4	2	362,46
Система мережевого обладнання передачі даних	19200,00	4	2	800,00
Офісна оргтехніка	10100,00	5	2	336,67
Приміщення лабораторії	255500,00	25	2	1703,33
ОС Windows 11	5645,00	3	2	313,61
Прикладний пакет Microsoft Office 2019	5155,00	3	2	286,39
Прикладний пакет моделювання	7646,00	3	2	424,78
Всього				10622,09

4.4.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^{n} \frac{W_{yi} \cdot t_i \cdot Ц_e \cdot K_{вni}}{\eta_i}, \quad (4.16)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

$Ц_e$ – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), приймемо $Ц_e = 7,50$ грн;

$K_{вni}$ – коефіцієнт, що враховує використання потужності, $K_{вni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$B_e = 0,32 \cdot 200,0 \cdot 7,50 \cdot 0,95 / 0,97 = 480,00$ *грн.*

Проведені розрахунки зведемо до таблиці.

Таблиця 4.14 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Електронно-обчислювальний центр моделювання та тестування технології КОМП'ЮТЕР HP ELITEONE 870 G9 AIO / I7-13700 (7B0P5EA)	0,32	200,0	480,00
Робоче місце Project Manager 2/0	0,06	280,0	126,00
Робоче місце інженера-дослідника Ноутбук Lenovo IdeaPad 330-15ICH (81FK00G1RA)	0,05	260,0	97,50
Пристрій виводу інформації CANON Laser 6900	0,25	5,0	9,38
Система мережевого обладнання передачі даних	0,10	200,0	150,00
Офісна оргтехніка	0,55	2,0	8,25
Фізичне обладнання серверу бази даних	0,20	200,0	300,00
Всього			1171,13

4.4.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків. Частина 2. Нечітка логіка» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов’язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з’їзди, конференції, наради, пов’язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20…25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{св} = (З_о + З_р) \cdot \frac{H_{св}}{100\%}, \qquad (4.17)$$

де $H_{св}$ – норма нарахування за статтею «Службові відрядження», приймемо $Н_{св}$ = 20%.

$B_{св} = (48540,91 + 2236,84) \cdot 20 / 100\% = 10155,55$ *грн.*

4.4.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» відсутні.

4.4.11 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» розраховуємо як 50…100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.18)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати», приймемо $Н_{ів} = 55\%$.

$I_в = (48540,91 + 2236,84) \cdot 55 / 100\% = 27927,76$ *грн.*

4.4.12 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» розраховуємо як 100…150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.19)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», приймемо $Н_{нзв} = 105\%$.

$B_{нзв} = (48540,91 + 2236,84) \cdot 105 / 100\% = 53316,64$ *грн.*

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків. Частина 2. Нечітка логіка» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = З_о + З_р + З_{дод} + З_н + М + К_в + В_{спец} + В_{прг} + А_{обл} + В_е + В_{св} + В_{сп} + І_в + В_{нзв}. \quad (4.20)$$

$B_{заг}$= 48540,91 +2236,84 +5077,77 +12288,21 +3415,50 +9777,90 +15180,00 + 18040,00 + 10622,09 +1171,13 +10155,55 +0,00 +27927,76 +53316,64 = 217750,30 грн.

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

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (4.21)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, приймемо η=0,95.

ЗВ = 217750,30 / 0,95 = 229210,84 грн.

4.5 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів за темами «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків. Частина 1. Надання рекомендацій» та «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків. Частина 2. Нечітка логіка» становлять:

$ЗВ_1$ + $ЗВ_2$ = 307712,19 + 229210,84 = 536 923,03 грн.

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.

Результати дослідження проведені за темою «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку основу майбутнього економічного ефекту будуть формувати:

ΔN – збільшення кількості споживачів яким надається відповідна інформаційна послуга у періоди часу, що аналізуються;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	25000	30000	20000	10000

N – кількість споживачів яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки, приймемо 100000 осіб;

$Ц_б$ – вартість послуги у році до впровадження інформаційної технології, приймемо 60,00 грн;

$\pm\Delta Ц_o$ – зміна вартості послуги від впровадження результатів, приймемо 32,88 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta П_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою [19]:

$$\Delta П_i = (\pm\Delta Ц_o \cdot N + Ц_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \qquad (4.22)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту). Приймемо $\rho = 40\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$\Delta П_1 = (32,88 \cdot 100000,00 + 92,88 \cdot 25000) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 1527266,40$ грн.

Збільшення чистого прибутку 2-го року:

$\Delta П_2 = (32,88 \cdot 100000,00 + 92,88 \cdot 55000) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 2285835,94$ грн.

Збільшення чистого прибутку 3-го року:

$\Delta П_3 = (32,88 \cdot 100000,00 + 92,88 \cdot 75000) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 2791548,96$ грн.

Збільшення чистого прибутку 4-го року:

$\Delta П_4 = (32,88 \cdot 100000,00 + 92,88 \cdot 85000) \cdot 0,83 \cdot 0,4 \cdot (1 - 0,18/100\%) = 3044405,47$ грн.

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

$$ПП = \sum_{i=1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (4.23)$$

де $\Delta П_i$ – 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;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,15$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = 1527266,40/(1+0,15)^1+2285835,94/(1+0,15)^2+2791548,96/(1+0,15)^3+$$
$$+3044405,47/(1+0,15)^4=1328057,74+1728420,37+1835488,75+1740648,71=$$
$$=6632615,57 \text{ грн.}$$

Величина початкових інвестицій PV, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{інв} \cdot ЗВ, \quad (4.24)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{інв}$=1,7;

$ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 536 923,03 грн.

$PV = k_{інв} \cdot ЗВ$ = 1,7 · 536 923,03 = 916843,37 грн.

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV \quad (4.25)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 6632615,57 грн;

PV – теперішня вартість початкових інвестицій, 916843,37 грн.

$E_{абс} = ПП - PV$ = 6632615,57 - 916843,37 = 5715772,20 грн.

Внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.26)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 5715772,20 грн;

PV – теперішня вартість початкових інвестицій, 916843,37 грн;

$T_ж$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$E_в = \sqrt[Tж]{1 + \frac{E_{абс}}{PV}} - 1$ = (1+5715772,20/916843,37)$^{1/4}$= 0,64.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$:

$$\tau_{мін} = d + f, \quad (4.27)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні d =0,1;

f – показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,3.

$\tau_{мін}$ = 0,1+0,3 = 0,4 < 0,64 свідчить про те, що внутрішня економічна дохідність інвестицій $E_в$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою

«Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_в}, \quad (4.28)$$

де $E_в$ – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок}$ = 1 / 0,64 = 1,56 р.

$T_{ок}$ < 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.6 Висновки до розділу 4

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків» становить 44,3 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 3,55 рази.

Також термін окупності становить 1,56 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків».

ВИСНОВКИ

При виконанні магістерської кваліфікаційної роботи розроблено інформаційну технологію підтримки прийняття рішень щодо вибору ноутбуків на основі нечіткої логіки.

У першому розділі магістерської роботи було проведено аналіз доцільності та обґрунтування вибору нечіткої логіки як основи для реалізації технології, проведено вибір та визначення нечітких даних, визначення об'єкту проектування показав, що завдання вибору ноутбуків є складним завданням, яке вимагає врахування різних параметрів та відомостей про різні моделі та бренди.

У другому розділі магістерської роботи детально розглянуто методи та моделі нечіткої логіки. Виконано класифікацію цих методів, обрано оптимальні для даної задачі та проведено комп'ютерне моделювання процесу надання рекомендацій за допомогою середовища Google Colab. Також, детально розглянуто методи та моделі нечіткої логіки, їх застосування та ефективність у контексті підтримки прийняття рішень щодо вибору ноутбуків. Значна увага була приділена розробці комп'ютерних моделей на основі нечіткої логіки, які забезпечують точне та адаптивне рішення для вибору ноутбуків. Особливо було відмічено важливість такого підходу в умовах постійного розширення асортименту та технічних характеристик ноутбуків, а також зростаючих вимог користувачів. Акцентовано увагу, що завдяки ефективному використанню нечіткої логіки, система допомагає користувачам уникати непотрібних витрат часу та грошей на моделі, які не відповідають їх потребам, забезпечуючи більш економічно вигідний та зручний вибір. Враховуючи швидкість технологічного прогресу та розмаїтість вибору на ринку, нечітка логіка стає незамінним інструментом для сучасних користувачів, які прагнуть оптимізувати свої рішення з урахуванням широкого спектру факторів.

Третій розділ магістерської роботи присвячено вибору мови програмування та середовища розробки враховувалася їхня здатність до

ефективного впровадження нечітких логічних модулів. Обрана мова забезпечує необхідну гнучкість та широкий спектр інструментів для легкої інтеграції та розробки нечітких логічних компонентів. Вибране середовище програмування відзначається своєю зручністю, продуктивністю та активною підтримкою спільноти, що сприяє ефективній розробці, тестуванню та впровадженню нечіткої логіки у технологію. Алгоритм розроблено на засадах нечіткої логіки, що дозволяє системі ефективно обробляти неоднозначні та суб'єктивні дані, забезпечуючи адекватність висновків.

Визначення нечітких множин, правил та методів дефазифікації забезпечує точність рішень, виходячи з введених користувачем даних. Система демонструє гнучкість та адаптивність до різноманітних сценаріїв використання, відповідаючи на специфічні вимоги користувачів.

Результати тестування підтвердили ефективність та надійність розробленої технології. Аналіз результатів виявив покращення у якості надання рекомендацій, досягнувши 7,18% вищої якості у порівнянні з аналогом.

У четвертому розділі магістерської роботи здійснено детальний розрахунок витрат на розробку та виготовлення нового технічного рішення, загальна сума яких становить 536 926 гривень. Витрати були розподілені за різними статтями, що дало змогу зробити точнішу оцінку загальних витрат. Враховуючи потенційний дохід від комерціалізації розробки, було також розраховано чистий прибуток для виробника. Знайдено, що термін окупності витрат виробника складе приблизно 1.56 року. Окрім того, період окупності проекту складає 1,56 року, що є значно меншим, ніж 3 роки. Це підкреслює комерційну ефективність науково-технічної розробки

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інформаційна технологія надання рекомендацій щодо вибору ноутбуків / В.О. Білошкурський, І.С. Мельник, А.А. Яровий, Ю.М. Паночишин – Тези LII науково-технічної конференції підрозділів ВНТУ (НТКП ВНТУ-2023). – [Електронний ресурс]. – Тип доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/18718/15504

2. Проектування інформаційної технології підтримки прийняття рішень щодо вибору ноутбука / Мельник І. С., Білошкурський В. О., Яровий А. А., Сімончук С.В. // Global science: prospects and innovations. Proceedings of the 4th International scientific and practical conference. Cognum Publishing House. Liverpool, United Kingdom. 2023. Pp. 306-314.3. Практичний підхід до вибору ноутбуків [Електронний ресурс]. Режим доступу - https://www.laptopmag.com/articles/laptop-buying-guide

4. Використання нечіткої логіки та штучного інтелекту в системах вибору ноутбуків [Електронний ресурс]. Режим доступу - https://www.researchgate.net/publication/331933912_Fuzzy_logic_and_intelligent_systems

5 technot.com.ua [Електронний ресурс]. Режим доступу - https://technot.com.ua/computers/laptops/

6. Туторіал Google Colab [Електронний ресурс]. Режим доступу - https://Colab.research.google.com/?hl=uk#

7. Актуальність та потреба в системах підтримки прийняття рішень [Електронний ресурс]. Режим доступу - https://www.sciencedirect.com/science/article/pii/S0957417415006004

8. Використання нечітких систем в процесі вибору ноутбуків [Електронний ресурс]. Режим доступу - https://www.researchgate.net/publication/220359731_Fuzzy_Decision_Making_in_Selecting_Laptop_Computers

9. Вибір мови програмування для розробки програмного забезпечення [Електронний ресурс]. Режим доступу - https://www.codecademy.com/learn/paths/choose-the-right-coding-language

10. Про бібліотеку scikit-fuzzy [Електронний ресурс]. Режим доступу - https://scikit-fuzzy.readthedocs.io/en/latest/

11. Огляд бібліотеки NumPy [Електронний ресурс]. Режим доступу - https://numpy.org/doc/stable

12. Розробка програмного забезпечення з використанням бібліотеки scikit-fuzzy [Електронний ресурс]. Режим доступу - https://ieeexplore.ieee.org/abstract/document/7440766

13. Можливості інших мов програмування для розробки нечітких систем [Електронний ресурс]. Режим доступу - https://link.springer.com/article/10.1007/s00500-019-03785-y

14. Реалізація алгоритму нечіткої інтелектуальної технології вибору ноутбуків [Електронний ресурс]. Режим доступу - https://www.researchgate.net/publication/323779496_Fuzzy_logic_based_Intelligent_Algorithm_for_choice_of_Laptop

15. Приклад використання бібліотеки scikit-fuzzy [Електронний ресурс]. Режим доступу - https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem_newapi.html

16. Експертні технології. Частина 1. Навчальний посібник. – Вінниця : ВНТУ, 2006. – 114 с.

17. Експертні технології. Частина 2 : навчальний посібник / Яровий А. А., Арсенюк І. Р., Месюра В. І. – Вінниця : ВНТУ, 2017. – 106 с.

18.Google Colab [Електронний ресурс]. Режим доступу: https://www.hwlibre.com/uk/%D1%81%D0%BF%D1%96%D0%B2%D1%80%D0%BE%D0%B1%D1%96%D1%82%D0%BD%D0%B8%D1%86%D1%82%D0%B2%D0%BE-Google/

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

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

ДОДАТКИ

Додаток А (обов'язковий)

Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків. Частина 2. Нечітка логіка.

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 94,69% Схожість 5,31%

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 Unicheck щодо роботи.

Автор роботи Мельник І.С.

Керівник роботи Яровий А.А.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку Озеранський В.С.

Додаток Б(обов'язковий)

Лістинг програми

```
!pip install networkx
!pip install scikit-fuzzy
!pip install  matplotlib

# @title Модуль параметрів
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt

# Define the Antecedents and Consequents
RAM = ctrl.Antecedent(np.arange(4, 65, 4), 'RAM')
SSD = ctrl.Antecedent(np.arange(128, 3072, 128), 'SSD')
HDD = ctrl.Antecedent(np.arange(512, 4000, 512), 'HDD')
keyboard = ctrl.Antecedent(np.arange(1, 11, 1), 'Keyboard')
matrix_type = ctrl.Antecedent(np.array([0, 1, 2, 3]), 'Matrix Type')
usb_ports = ctrl.Antecedent(np.arange(0, 6, 1), 'usb_ports')
screen = ctrl.Antecedent(np.arange(1, 5, 1), 'Screen')

prise = ctrl.Consequent(np.arange(200, 4000, 1), 'Price')
laptop_name = ctrl.Consequent(np.arange(0, 15, 1), 'Laptop Name')
processor_name = ctrl.Consequent(np.arange(0, 31, 1), 'Processor Name')
graphics_card_name = ctrl.Consequent(np.arange(0, 31, 1), 'Graphics Card
Name')

# Мовні терміни
RAM.automf(5, names=['very low', 'low', 'medium', 'good', 'very good'])
SSD.automf(5, names=['very low', 'low', 'medium', 'good', 'very good'])
HDD.automf(5, names=['very low', 'low', 'medium', 'good', 'very good'])
prise.automf(5, names=['very low', 'low', 'medium', 'good', 'very good'])
```

```
laptop_name['Asus'] = fuzz.trimf(laptop_name.universe, [0, 0, 1])
laptop_name['Dell'] = fuzz.trimf(laptop_name.universe, [0, 1, 2])
laptop_name['Acer'] = fuzz.trimf(laptop_name.universe, [1, 2, 3])
laptop_name['MSI'] = fuzz.trimf(laptop_name.universe, [2, 3, 4])
laptop_name['Hp'] = fuzz.trimf(laptop_name.universe, [3, 4, 5])
laptop_name['Lenovo'] = fuzz.trimf(laptop_name.universe, [4, 5, 6])
laptop_name['Aplle'] = fuzz.trimf(laptop_name.universe, [5, 6, 7])
laptop_name['Xiaomi'] = fuzz.trimf(laptop_name.universe, [6, 7, 8])
laptop_name['Gygabite'] = fuzz.trimf(laptop_name.universe, [7, 8, 9])
laptop_name['Microsoft'] = fuzz.trimf(laptop_name.universe, [8, 9, 10])

processor_name['Intel i3'] = fuzz.trimf(processor_name.universe, [0, 0,
2])
processor_name['Intel i5'] = fuzz.trimf(processor_name.universe, [0, 2,
4])
processor_name['Intel i7'] = fuzz.trimf(processor_name.universe, [2, 4,
6])
processor_name['Intel i9'] = fuzz.trimf(processor_name.universe, [4, 6,
8])
processor_name['AMD Ryzen 3'] = fuzz.trimf(processor_name.universe, [6,
8, 10])
processor_name['AMD Ryzen 5'] = fuzz.trimf(processor_name.universe, [8,
10, 12])
processor_name['AMD Ryzen 7'] = fuzz.trimf(processor_name.universe, [10,
12, 14])
processor_name['AMD Ryzen 9'] = fuzz.trimf(processor_name.universe, [12,
14, 16])

graphics_card_name['NVIDIA GeForce MX150'] =
fuzz.trimf(graphics_card_name.universe, [0, 0, 2])
graphics_card_name['NVIDIA GeForce GTX 1050'] =
fuzz.trimf(graphics_card_name.universe, [0, 2, 4])
graphics_card_name['NVIDIA GeForce GTX 1650'] =
fuzz.trimf(graphics_card_name.universe, [2, 4, 6])
```

```
graphics_card_name['NVIDIA GeForce RTX 2060'] =
fuzz.trimf(graphics_card_name.universe, [4, 6, 8])
graphics_card_name['NVIDIA GeForce RTX 3070'] =
fuzz.trimf(graphics_card_name.universe, [6, 8, 10])
graphics_card_name['AMD Radeon RX 5500M'] =
fuzz.trimf(graphics_card_name.universe, [8, 10, 12])
graphics_card_name['AMD Radeon RX 5700 XT'] =
fuzz.trimf(graphics_card_name.universe, [10, 12, 14])
graphics_card_name['AMD Radeon RX 6800 XT'] =
fuzz.trimf(graphics_card_name.universe, [12, 14, 16])
graphics_card_name['Radeon Graphics'] =
fuzz.trimf(graphics_card_name.universe, [14, 16, 18])
graphics_card_name['Intel UHD Graphics'] =
fuzz.trimf(graphics_card_name.universe, [16, 18, 20])
graphics_card_name['Intel Iris Xe Graphics'] =
fuzz.trimf(graphics_card_name.universe, [18, 20, 22])
graphics_card_name['NVIDIA GeForce RTX 3050'] =
fuzz.trimf(graphics_card_name.universe, [20, 22, 24])
graphics_card_name['NVIDIA GeForce RTX 4060'] =
fuzz.trimf(graphics_card_name.universe, [22, 24, 26])

keyboard.automf(5, names=['very bad', 'bad', 'average', 'good', 'very
good'])
matrix_type.automf(4, names=['TN', 'IPS', 'OLED', 'Retina'])
usb_ports.automf(5, names=['none', 'few', 'average', 'many', 'very
many'])
screen.automf(4, names=['small', 'medium', 'large', 'extra large'])

SSD.view()
HDD.view()
prise.view()
laptop_name.view()
processor_name.view()
graphics_card_name.view()
```

```
plt.show()

# @title База правил

rules = [

ctrl.Rule(RAM['very low'] & SSD['very low'] & HDD['very low'] &
keyboard['very bad'] & matrix_type['TN'] & usb_ports['none'] &
screen['small'],
                  (prise['very low'], laptop_name['Asus'],
processor_name['Intel i3'], graphics_card_name['Intel UHD Graphics'])),

ctrl.Rule(RAM['medium'] & SSD['medium'] & HDD['medium'] &
keyboard['average'] & matrix_type['IPS'] & usb_ports['average'] &
screen['medium'],
                  (prise['good'], laptop_name['MSI'],
processor_name['Intel i5'], graphics_card_name['NVIDIA GeForce GTX
1650'])),

ctrl.Rule(RAM['very good'] & SSD['good'] & HDD['good'] & keyboard['good']
& matrix_type['OLED'] & usb_ports['many'] & screen['large'],
                  (prise['good'], laptop_name['MSI'], processor_name['AMD
Ryzen 7'], graphics_card_name['NVIDIA GeForce RTX 2060'])),

ctrl.Rule(RAM['very good'] & SSD['very good'] & HDD['very good'] &
keyboard['very good'] & matrix_type['Retina'] & usb_ports['very many'] &
screen['extra large'],
                  (prise['very good'], laptop_name['Asus'],
processor_name['Intel i9'], graphics_card_name['NVIDIA GeForce RTX
3070'])),

ctrl.Rule(RAM['low'] & SSD['low'] & HDD['low'] & keyboard['bad'] &
matrix_type['TN'] & usb_ports['few'] & screen['small'],
```

```
                (prise['low'], laptop_name['Acer'],
processor_name['Intel i3'], graphics_card_name['Intel UHD Graphics'])),

ctrl.Rule(RAM['medium'] & SSD['good'] & HDD['medium'] &
keyboard['average'] & matrix_type['IPS'] & usb_ports['average'] &
screen['medium'],
                (prise['medium'], laptop_name['Hp'],
processor_name['AMD Ryzen 3'], graphics_card_name['Radeon Graphics'])),

ctrl.Rule(RAM['good'] & SSD['very good'] & HDD['good'] & keyboard['good']
& matrix_type['OLED'] & usb_ports['many'] & screen['large'],
                (prise['good'], laptop_name['Lenovo'],
processor_name['AMD Ryzen 5'], graphics_card_name['NVIDIA GeForce GTX
1650'])),

ctrl.Rule(RAM['very low'] & SSD['very low'] & HDD['very low'] &
keyboard['very bad'] & matrix_type['TN'] & usb_ports['few'] &
screen['small'],
                    (prise['very low'], laptop_name['Asus'],
processor_name['Intel i3'], graphics_card_name['Intel UHD Graphics'])),

ctrl.Rule(RAM['good'] & SSD['good'] & HDD['good'] & keyboard['good'] &
matrix_type['IPS'] & usb_ports['many'] & screen['large'],
          (prise['good'], laptop_name['Dell'], processor_name['Intel
i7'], graphics_card_name['NVIDIA GeForce GTX 1650'])),

ctrl.Rule(RAM['very low'] & SSD['very low'] & HDD['very low'] &
keyboard['very bad'] & matrix_type['TN'] & usb_ports['none'] &
screen['small'],
          (prise['very low'], laptop_name['Acer'], processor_name['Intel
i3'], graphics_card_name['Intel UHD Graphics'])),
```

```
ctrl.Rule(RAM['medium'] & SSD['very good'] & HDD['none'] &
keyboard['good'] & matrix_type['IPS'] & usb_ports['many'] &
screen['large'],
          (prise['good'], laptop_name['Lenovo'], processor_name['AMD
Ryzen 5'], graphics_card_name['NVIDIA GeForce GTX 1650'])),

ctrl.Rule(RAM['very good'] & SSD['very good'] & HDD['very good'] &
keyboard['very good'] & matrix_type['Retina'] & usb_ports['very many'] &
screen['extra large'],
          (prise['very good'], laptop_name['Asus'], processor_name['Intel
i9'], graphics_card_name['NVIDIA GeForce RTX 3070'])),

ctrl.Rule(RAM['low'] & SSD['low'] & HDD['low'] & keyboard['bad'] &
matrix_type['TN'] & usb_ports['few'] & screen['small'],
          (prise['low'], laptop_name['Acer'], processor_name['Intel i3'],
graphics_card_name['Intel UHD Graphics'])),

ctrl.Rule(RAM['medium'] & SSD['good'] & HDD['medium'] &
keyboard['average'] & matrix_type['IPS'] & usb_ports['average'] &
screen['medium'],
          (prise['medium'], laptop_name['Hp'], processor_name['AMD Ryzen
3'], graphics_card_name['Radeon Graphics'])),

ctrl.Rule(RAM['good'] & SSD['very good'] & HDD['good'] & keyboard['good']
& matrix_type['OLED'] & usb_ports['many'] & screen['large'],
          (prise['good'], laptop_name['Lenovo'], processor_name['AMD
Ryzen 5'], graphics_card_name['NVIDIA GeForce GTX 1650'])),

ctrl.Rule(RAM['medium'] & SSD['low'] & HDD['none'] & keyboard['bad'] &
matrix_type['TN'] & usb_ports['few'] & screen['small'],
          (prise['low'], laptop_name['Acer'], processor_name['Intel i3'],
graphics_card_name['Intel UHD Graphics'])),
```

```
ctrl.Rule(RAM['good'] & SSD['very good'] & HDD['none'] &
keyboard['excellent'] & matrix_type['IPS'] & usb_ports['few'] &
screen['medium'],
          (prise['high'], laptop_name['Dell XPS'], processor_name['Intel
i7'], graphics_card_name['Intel Iris Xe'])),

ctrl.Rule(RAM['medium'] & SSD['medium'] & HDD['none'] &
keyboard['average'] & matrix_type['IPS'] & usb_ports['average'] &
screen['medium'],
          (prise['medium'], laptop_name['HP Spectre x360'],
processor_name['Intel i5'], graphics_card_name['Intel Iris Xe'])),

ctrl.Rule(RAM['very good'] & SSD['very good'] & HDD['none'] &
keyboard['excellent'] & matrix_type['OLED'] & usb_ports['many'] &
screen['large'],
          (prise['high'], laptop_name['Dell Precision'],
processor_name['Intel i5'], graphics_card_name['NVIDIA Quadro RTX
4000'])),

ctrl.Rule(RAM['medium'] & SSD['low'] & HDD['none'] & keyboard['average']
& matrix_type['TN'] & usb_ports['few'] & screen['medium'],
          (prise['low'], laptop_name['Acer Swift'], processor_name['Intel
Pentium'], graphics_card_name['Intel UHD Graphics'])),

ctrl.Rule(RAM['low'] & SSD['medium'] & HDD['none'] & keyboard['average']
& matrix_type['IPS'] & usb_ports['average'] & screen['small'],
          (prise['low'], laptop_name['Lenovo Ideapad'],
processor_name['Intel i5'], graphics_card_name['Intel UHD Graphics'])),

ctrl.Rule(RAM['very good'] & SSD['very good'] & HDD['large'] &
keyboard['excellent'] & matrix_type['IPS'] & usb_ports['many'] &
screen['large'],
```

```
          (prise['very high'], laptop_name['HP ZBook'],
processor_name['Intel i5'], graphics_card_name['NVIDIA Quadro RTX
5000'])),

ctrl.Rule(RAM['excellent'] & SSD['excellent'] & HDD['none'] &
keyboard['excellent'] & matrix_type['OLED'] & usb_ports['many'] &
screen['extra large'],
          (prise['very high'], laptop_name['Alienware'],
processor_name['AMD Ryzen 9'], graphics_card_name['NVIDIA GeForce RTX
3080'])),

ctrl.Rule(RAM['medium'] & SSD['medium'] & HDD['none'] &
keyboard['average'] & matrix_type['IPS'] & usb_ports['average'] &
screen['medium'],
          (prise['medium'], laptop_name['System76'], processor_name['AMD
Ryzen 5'], graphics_card_name['AMD Radeon Graphics'])),
]

# @title Вибір ноутбуків
# @markdown RAM:
RAM = 4 # @param {type:"slider", min:4, max:65, step:4}
# @markdown SSD:
SSD = 128 # @param {type:"slider", min:128, max:3072, step:128}
# @markdown HDD:
HDD = 512 # @param {type:"slider", min:512, max:4096, step:512}
# @markdown Клавіатура:
Keyboard = 1 # @param {type:"slider", min:1, max:11, step:1}
# @markdown Тип матриці:
Matrix_Type = 0 # @param {type:"slider", min:0, max:3, step:1}
# @markdown Кількість портів:
usb_ports= 2 # @param {type:"slider", min:0, max:6, step:1}
# @markdown Екран:
Screen = 1 # @param {type:"slider", min:1, max:5, step:1}
```

```
# Створення технології контролю
laptop_selection_ctrl = ctrl.ControlSystem(rules)

# Створення симуляції
laptop_selection_sim =
ctrl.ControlSystemSimulation(laptop_selection_ctrl)

# Введення значень
laptop_selection_sim.input['RAM'] = RAM
laptop_selection_sim.input['SSD'] = SSD
laptop_selection_sim.input['HDD'] = HDD
laptop_selection_sim.input['Keyboard'] = Keyboard
laptop_selection_sim.input['Matrix Type'] =Matrix_Type
laptop_selection_sim.input['usb_ports'] = usb_ports
laptop_selection_sim.input['Screen'] = Screen
# Обчислення виводу
laptop_selection_sim.compute()

# Виведення результатів
print(f"Рекомендований бренд ноутбуків:
{laptop_selection_sim.output['Laptop Name']}")
print(f"Рекомендований процесор: {laptop_selection_sim.output['Processor
Name']}")
print(f"Рекомендована відеокарта: {laptop_selection_sim.output['Graphics
Card Name']}")
print(f"Діапазон ціна: {laptop_selection_sim.output['Price']}")

# @title Модуль дефазифікації

def get_label(crisp_value, antecedent):
    # Find the closest membership function to the crisp value
    closest_label = None
    closest_distance = float('inf')
    for term in antecedent.terms:
```

```
        term_value = antecedent[term].mf[round(crisp_value)]
        distance = abs(term_value - crisp_value)
        if distance < closest_distance:
            closest_distance = distance
            closest_label = term
    return closest_label

laptop_name_str = get_label(laptop_selection_sim.output['Laptop Name'],
laptop_name)
processor_name_str = get_label(laptop_selection_sim.output['Processor
Name'], processor_name)
graphics_card_name_str = get_label(laptop_selection_sim.output['Graphics
Card Name'], graphics_card_name)
price_range_str = get_label(laptop_selection_sim.output['Price'], prise)

print(f"Рекомендований бренд ноутбуків: {laptop_name_str}")
print(f"Рекомендований процесор: {processor_name_str}")
print(f"Рекомендована відеокарта: {graphics_card_name_str}")
print(f"Діапазон ціна: {price_range_str}")

# @title Модуль візуалізації
RAM.view(laptop_selection_sim)
SSD.view(laptop_selection_sim)
HDD.view(laptop_selection_sim)
graphics_card_name.view(laptop_selection_sim)
processor_name.view(laptop_selection_sim)
laptop_name.view(laptop_selection_sim)
prise .view(laptop_selection_sim)
laptop_selection_ctrl.view()
```

Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВИБОРУ НОУТБУКІВ. ЧАСТИНА 2. НЕЧІТКА ЛОГІКА.

Виконав: студент 2-го курсу,
групи 3КН-22м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напряму підготовки, спеціальності)
_____ Мельник І,С
(прізвище та ініціали)

Керівник: д.т.н., проф. кафедри КН
_____ Яровий А.А.
(прізвище та ініціали)
«_07_»_____12_____2023 р.

Вінниця ВНТУ - 2023 рік

Візуалізація нечітких термів

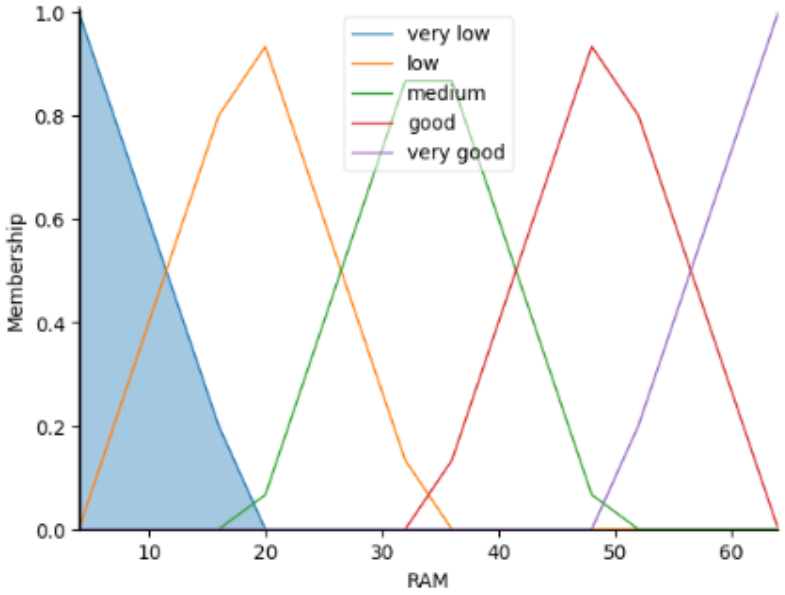


Рисунок В.1 — Нечіткі терми параметру RAM

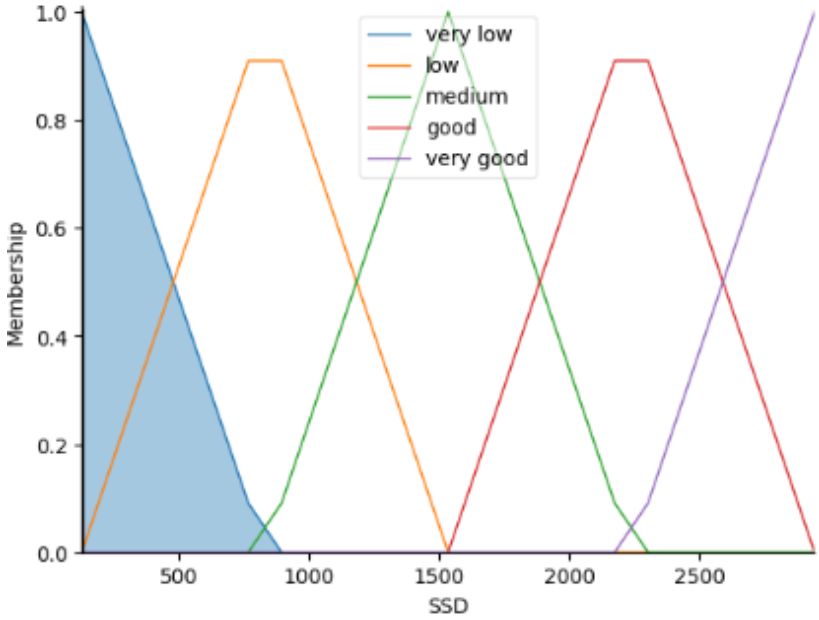


Рисунок В.2 — Нечіткі терми параметру SSD накопичувача

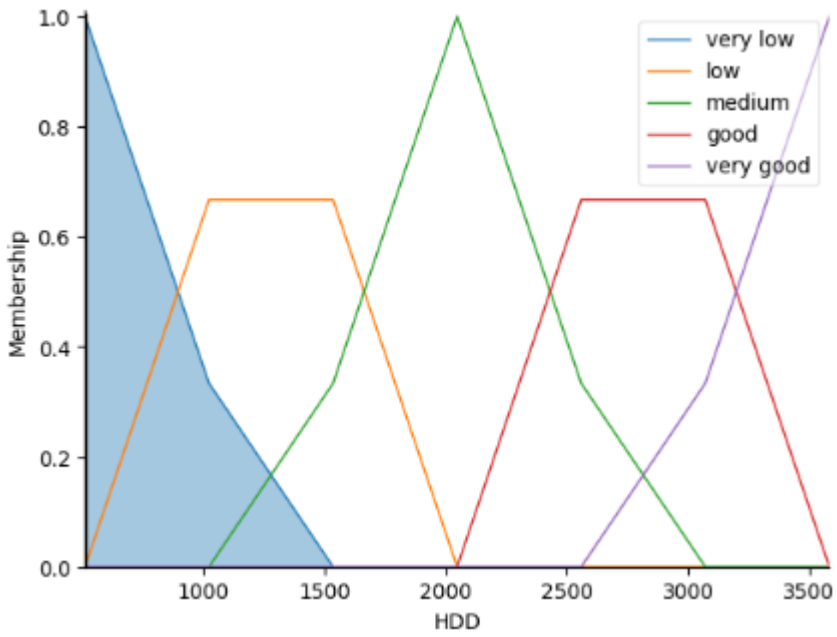


Рисунок В.3 — Нечіткі терми параметру HDD накопичувач

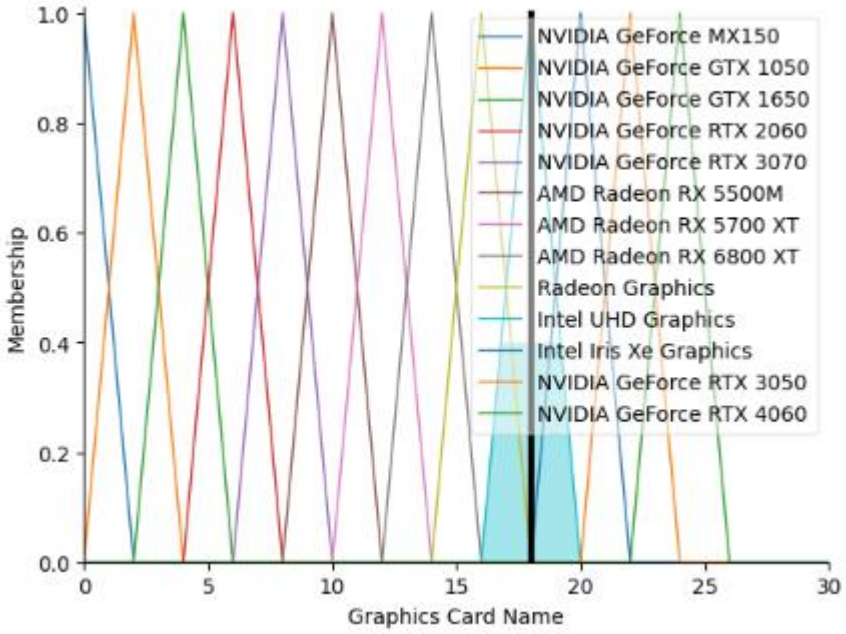


Рисунок В.4 — Нечіткі терми параметру Graphics Card Name

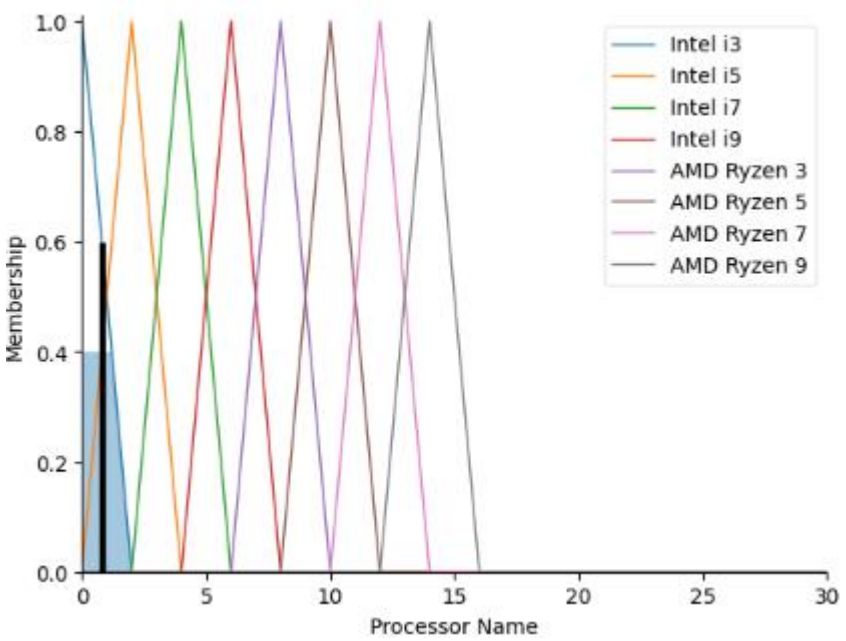


Рисунок В.5 — Нечіткі терми параметру Processor Name

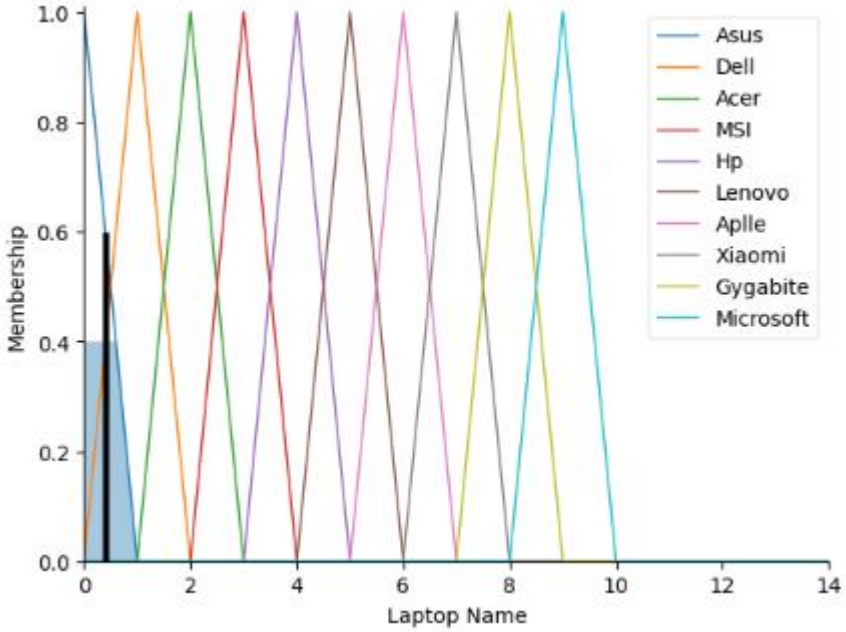


Рисунок В.6 — Нечіткі терми параметру Laptop Name

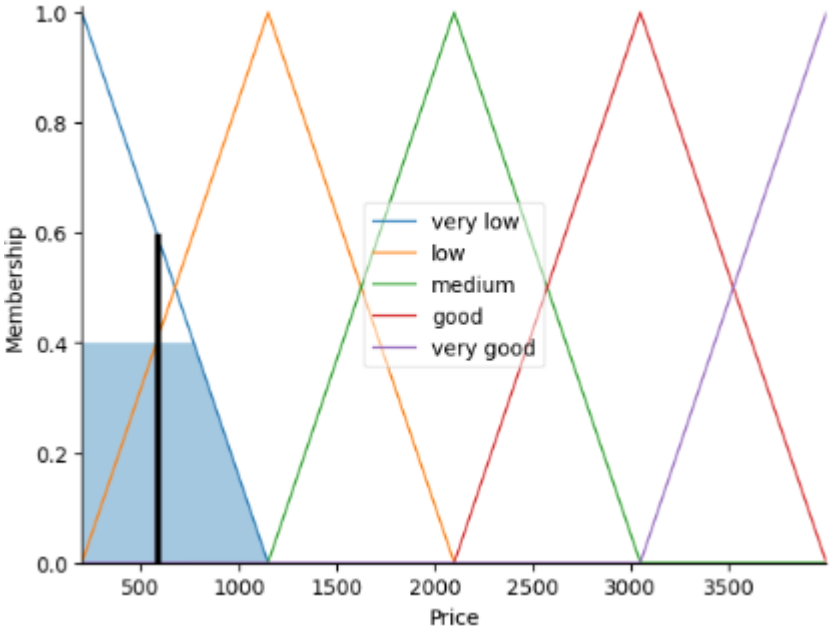


Рисунок В.7 — Нечіткі терми параметру Price

Логічні ланцюги правил та контролер

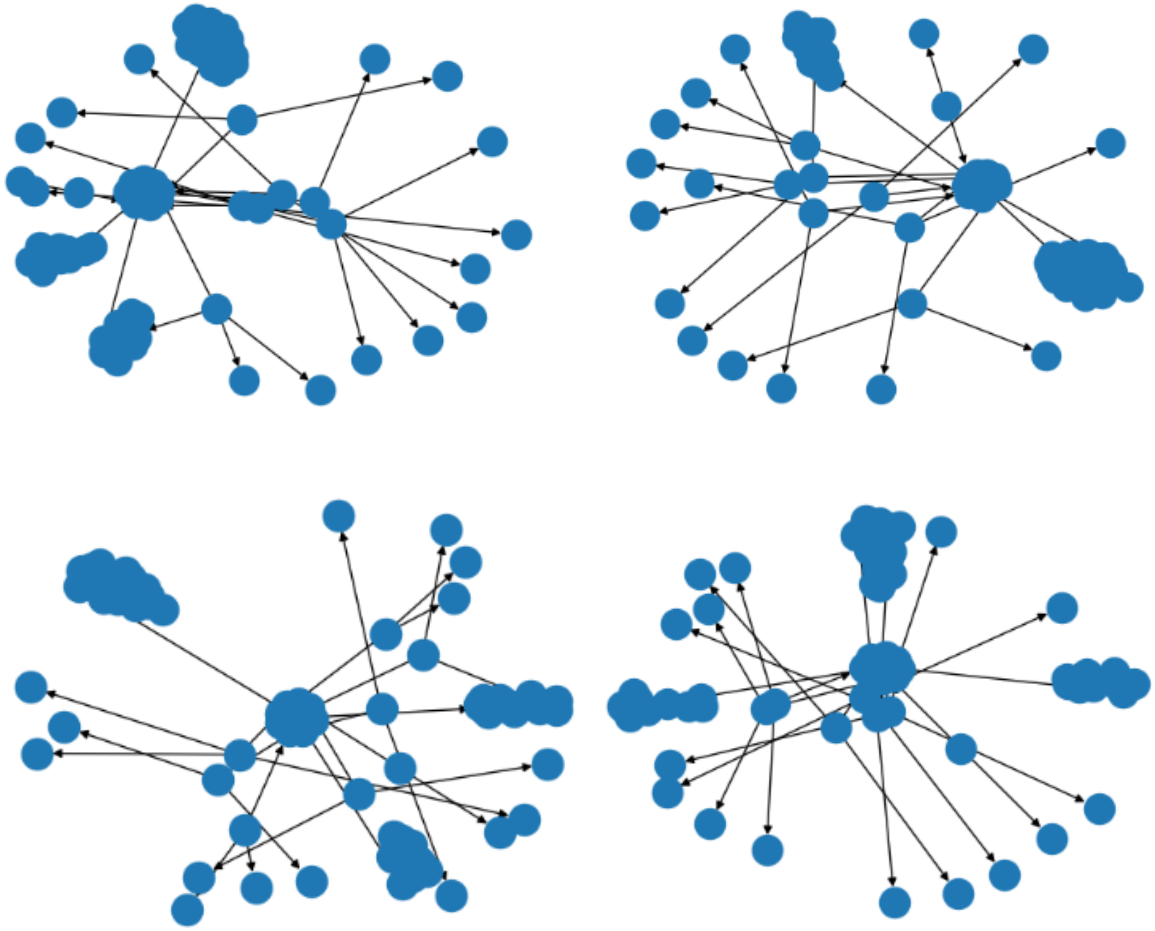


Рисунок В.8 — Фрагменти логічних ланцюгів розроблених правил

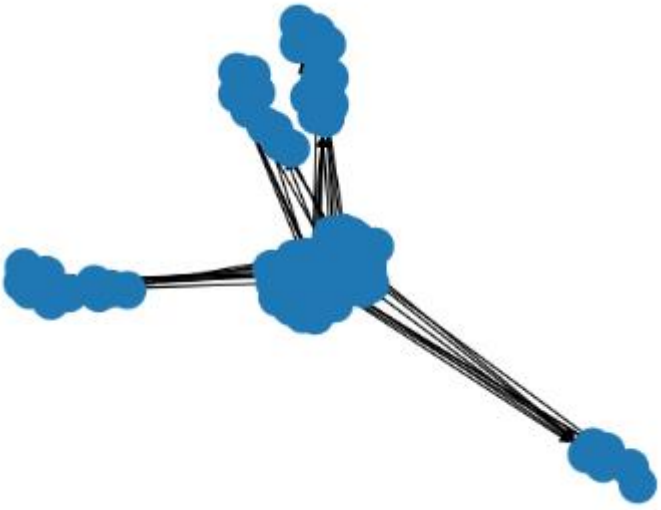


Рисунок В.9 —Візуалізація контролера

UML діаграма визначення методу

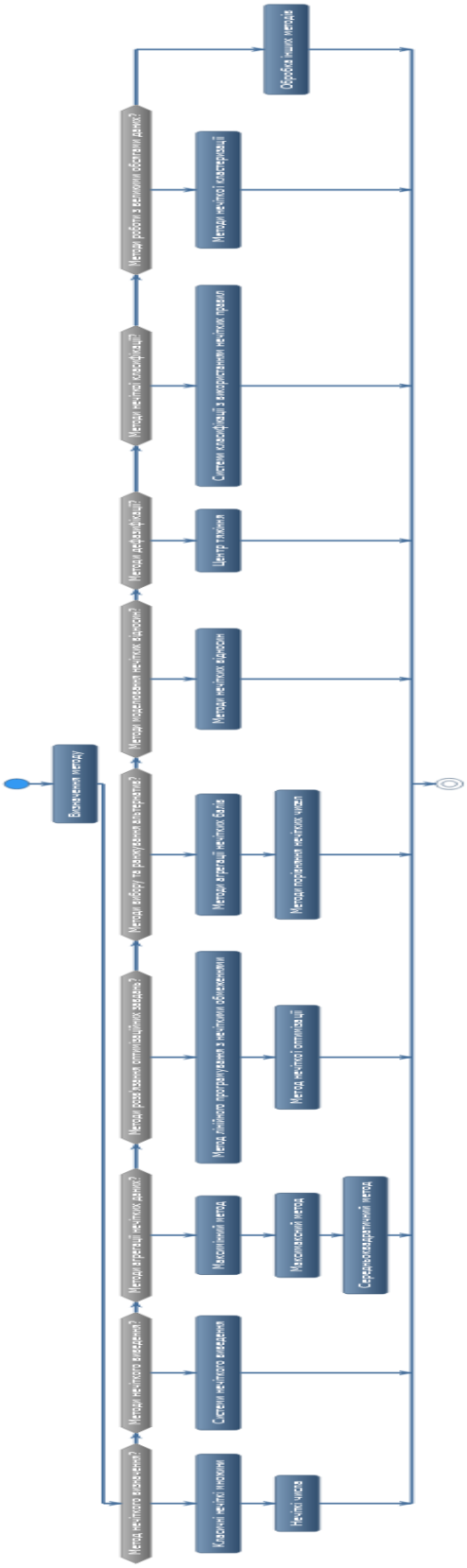


Рисунок В.10 — UML діаграма визначення методів вирішення

UML діаграма взаємодії компонентів технології

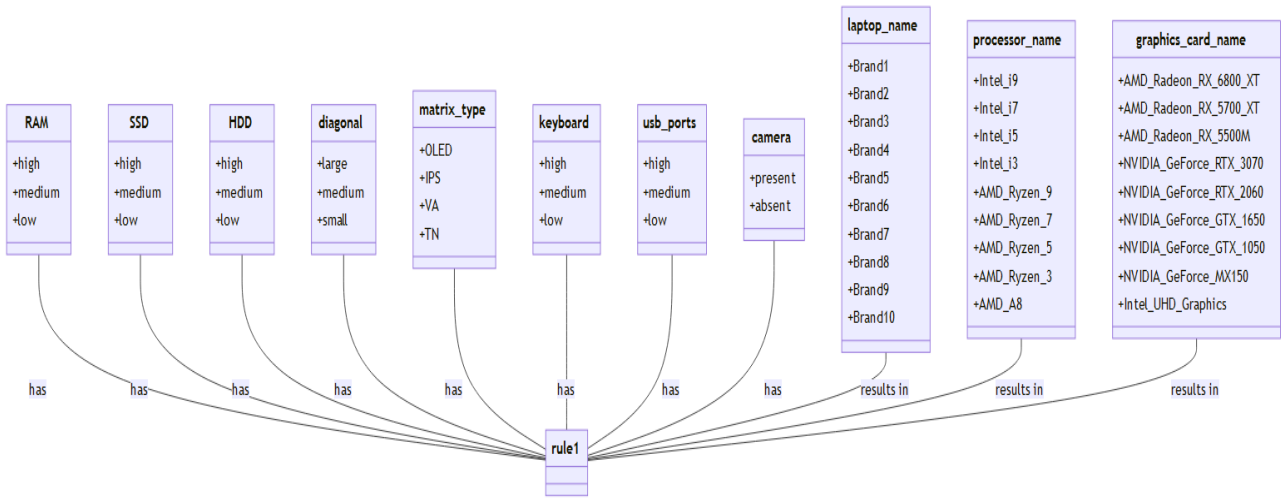


Рисунок В.11 — UML діаграма взаємодії параметрів з правилами

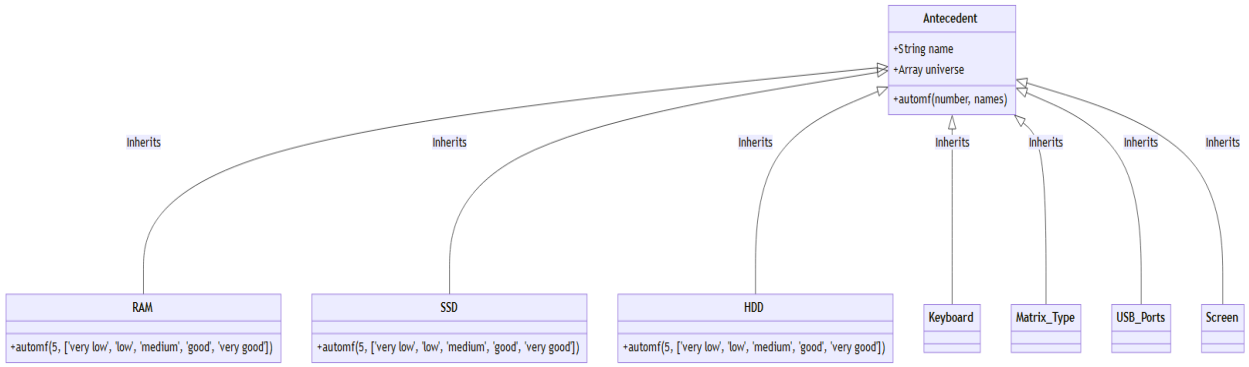
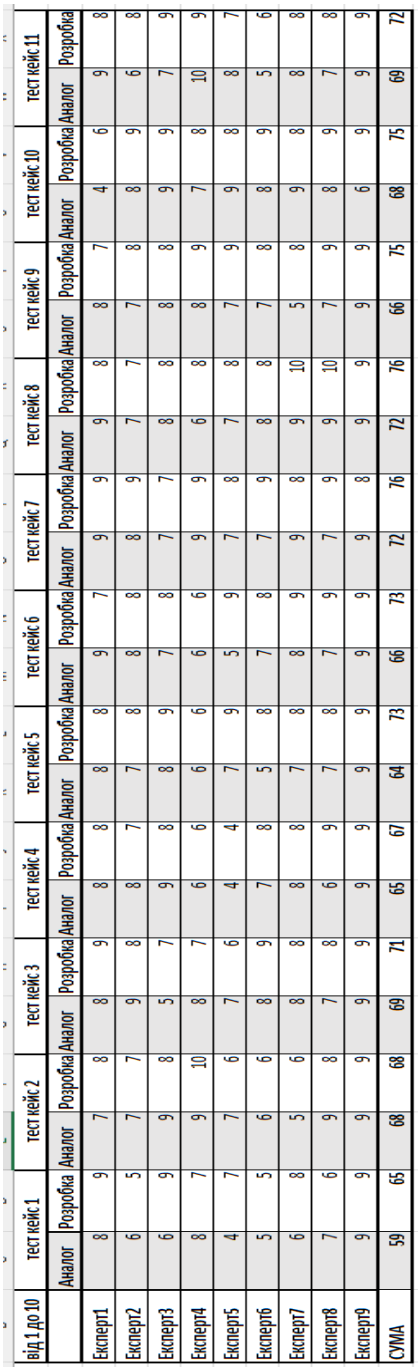


Рисунок В.12 — UML діаграма антицидентів

Таблиця тестування



від 1 до 10	тест кейс 1		тест кейс 2		тест кейс 3		тест кейс 4		тест кейс 5		тест кейс 6		тест кейс 7		тест кейс 8		тест кейс 9		тест кейс 10		тест кейс 11	
	Аналог	Розробка	Аналог	Розробка	Аналог	Розробка	Аналог	Розробка	Аналог	Розробка	Аналог	Розробка	Аналог	Розробка	Аналог	Розробка	Аналог	Розробка	Аналог	Розробка	Аналог	Розробка
Експерт1	8	9	7	8	8	9	8	8	8	8	9	7	9	9	9	8	8	8	7	4	6	9	8
Експерт2	6	5	7	7	9	8	8	7	7	8	8	8	8	9	7	7	7	7	8	8	9	6	8
Експерт3	6	9	9	8	5	7	9	8	8	9	7	8	7	7	8	8	8	8	8	9	9	7	9
Експерт4	8	7	9	10	8	7	6	6	6	6	6	6	9	9	6	8	8	8	9	7	8	10	9
Експерт5	4	7	7	6	7	6	4	4	7	9	5	9	7	8	7	8	7	7	9	9	8	8	7
Експерт6	5	5	6	6	8	9	7	8	5	8	7	8	7	9	8	8	7	7	8	8	9	5	6
Експерт7	6	8	5	6	8	8	8	8	7	8	8	9	9	8	9	10	5	8	9	9	8	8	8
Експерт8	7	6	9	8	7	8	6	9	7	8	7	9	7	9	9	10	7	9	8	9	9	7	8
Експерт9	9	9	9	9	9	9	9	9	9	9	9	9	9	8	9	9	9	9	9	6	9	9	9
СУМА	59	65	68	68	69	71	65	67	64	73	66	73	72	76	72	76	66	75	75	68	75	69	72

Рисунок В.13 — Результати тестування в порівнянні з аналогом

Додаток Г (довідниковий)

Акт впровадження

ФОП Гуменюк Сергій Валерійович «СЦ СЕРВІС+»

Україна, 23210 смт. Стрижавка, вул. Присадибна, буд 12 тел.+380636491460

11 жовтня 2023 року

Довідка дана студентові Вінницького національного технічного університету групи 3КН-22м Мельнику Івану Сергійовичу в тому, що результати магістерської кваліфікаційної роботи на тему «Інформаційна технологія підтримки прийняття рішень щодо вибору ноутбуків. Частина 2. Нечітка логіка.» використані в інтелектуальній системі з використанням нечіткої логіки у виборі ноутбуків враховуючи технічні характеристики, потреби та ціни. За результатами роботи інтелектуальна система у вигляді готового продукту планується до впровадження.

Директор ФОП С. В. Гуменюк

М.П.