

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Розробка методів формування рекомендацій в інформаційній системі з
використанням інструментів нечіткої логіки
(тема)

Виконав: здобувач групи ІТІм-21-1
спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

освітньої програми Інформаційні технології
проектування
(повна назва освітньої програми)

Носик К.А.

(прізвище, ініціали)

Керівник доц. каф. СТ, к.т.н. Ситникова П.Е.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри системотехніки

(підпис)

Гребеннік І.В.

(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Системотехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Освітня програма Інформаційні технології проектування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри

Гребеннік І.В.

(підпис)

«___» _____ 2022 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Носик Катерині Андріївні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка методів формування рекомендацій в інформаційній системі з використанням інструментів нечіткої логіки

затверджена наказом університету від «21» листопада 2022 р. № 1504 Ст

2. Термін подання здобувачем роботи 18 грудня 2022 р.

3. Вихідні дані до роботи Розробити методи формування рекомендацій в інформаційній системі електронної комерції «Магазин косметичних засобів». та клієнтську частину системи автоматизації обліку продажів канцтоварів. Серверна частина – база даних, розроблена для платформи СУБД Microsoft SQL Server. Клієнтська частина – веб-API з використанням технології .NET. Інтегроване середовище розробки – Microsoft Visual Studio C# 2022, мова програмування C#. Перелік використовуваних програмних засобів: ОС Microsoft Windows 10. Технічне забезпечення: Intel Core i5-6300U, 8Гб RAM.

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Вступ. 4.2 Аналіз предметної області. 4.2.1 Аналіз предметної області, яка визначає діяльність фірми. 4.2.2 Аналіз реалізованих систем. 4.2.3 Методи, технології та джерела інформації з дослідженням предметної області. 4.2.4 Постановка задачі дослідження. 4.3 Огляд методів та технологій, які застосовуються в предметній області. 4.3.1 Методи, типи та особливості побудови рекомендаційних систем. 4.3.2 Методи нечіткої логіки. 4.3.3 Огляд можливих проектних рішень. 4.3.3.1 Огляд мови програмування C#. 4.3.3.2 Огляд СУБД MS SQL Server. 4.3.3.3 Огляд можливостей розробки API. 4.4 Математичний опис. 4.4.1 Опис методів нечіткої логіки. 4.4.1.1 Фаззифікація 4.4.1.2 Нечітке логічне виведення. Механізм Мамдані. 4.4.1.3 Дефаззифікація. 4.4.2 Застосування інструментів нечіткої логіки до обраної

предметної області. 4.4.2.1 Визначення лінгвістичних змінних. 4.4.2.2 Оголошення нечітких правил. 4.4.2.3 Оператори нечіткої логіки. 4.4.2.4 Алгоритм функціонування нечіткої логіки. 4.4.3 Побудова математичної моделі. 4.5 Програмна реалізація рекомендаційного інтерфейсу. 4.5.1 Розробка бази даних на обраній платформі СУБД. 4.5.2 Розробка коду програми з використанням обраної мови програмування. 4.5.3 Опис розробленого рекомендаційного веб-API. 4.5.4 Проведення експериментів стосовно надання рекомендацій. 4.6 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 5.1. Схема алгоритму застосування нечіткої логіки у обраній ІС. 5.2 Графік значень лінгвістичної змінної «Частота миття волосся». 5.3 Графік значень лінгвістичної змінної «Стан шкіри». 5.4 Діаграма класів системи нечіткого виведення. 5.5 Діаграма класів у Microsoft Visual Studio 2022. 5.6 Екранна форма 1 розробленого Web API. 5.7 Екранна форма 2 розробленого Web API. 5.8 Екранна форма отримання рекомендацій товарів та послуг.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Отримання завдання атестаційної роботи	10.09.22	виконано
2.	Аналіз завдання, літератури та аналогів з теми атестаційної роботи	11.09 — 25.09.22	виконано
3.	Розробка алгоритму	26.09 — 06.10.22	
4.	Вибір засобів для розробки технічних вимог до програми	07.10.22	виконано
5.	Вибір середовища розробки програми	08.10.22	виконано
6.	Розробка програми	09.10 — 29.10.22	виконано
7.	Тестування програми	30.10 — 16.11.22	виконано
8.	Оформлення пояснювальної записки та програмної документації	17.11 — 09.12.22	виконано
9.	Оформлення графічної частини та презентаційних матеріалів комп'ютерного захисту	11.12.2022	виконано
10.	Представлення на рецензування	18.12.2022	виконано
11.	Представлення кваліфікаційної роботи до ЕК	21.12.2022	виконано

Дата видачі завдання 10 вересня 2022 р.

Здобувач _____
(підпис)

Керівник роботи _____
(підпис)

доц. каф. СТ, к.т.н. Ситникова П.Е.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: 98 с., 2 табл., 40 рис., 3 додатків, 29 джерел інформації.

API, БАЗА ДАНИХ, ЕЛЕКТРОННА КОМЕРЦІЯ, ІНТЕРНЕТ, ІНФОРМАЦІЙНА СИСТЕМА, НЕЧІТКА ЛОГІКА, РЕКОМЕНДАЦІЙНА СИСТЕМА.

Об'єктом досліджень є рекомендаційні системи електронної комерції.

Предметом досліджень є методи та моделі формування персоналізованих рекомендацій, методи нечіткої логіки.

Мета дослідження: дослідження та розробка методів формування рекомендацій у інформаційних системах з використанням інструментарію нечіткої логіки.

Методи дослідження – спостереження, розглядання загальноприйнятних алгоритмів формування рекомендацій та підходів із залученням нечіткої логіки, пояснення, прогнозування. Розробка методу надання твоярних та процедурних побажань із залученням методу Мамдані.

У роботі досліджені методи формування рекомендацій у інформаційних системах з використанням алгоритмів нечіткої логіки. Результатом роботи є запропонований власний алгоритм формування рекомендацій на основі математичної моделі нечіткої логіки. Практичним результатом є створення веб-API для взаємодії з цією моделлю.

Галузь застосування – рекомендаційні системи для компаній, які дозволяють знаходити найбільш підходящі товари з точки зору клієнта.

ABSTRACT

Certification work contains: 98 pages, 40 images, 2 tables, 3 appendices, 29 sources.

API, DATABASE, E-COMMERCE, INTERNET, INFORMATION SYSTEM, FUZZY LOGIC, RECOMMENDER SYSTEM.

The object of research is recommendation systems of e-commerce.

The subject of research are methods and models of personalized recommendations, fuzzy logic methods.

Purpose of the study: research and development of methods for generating recommendations in information systems using fuzzy logic tools.

Research methods – observation, consideration of generally accepted algorithms for the formation of recommendations and approaches involving fuzzy logic, explanation, forecasting. Development of a method of providing creative and procedural wishes with the involvement of the Mamdani method.

The paper investigates the methods of forming recommendations in information systems using fuzzy logic algorithms. The result of the work is the proposed own algorithm for generating recommendations based on the mathematical model of fuzzy logic. The practical result is the creation of a web API for interaction with this model.

Scope – recommendation systems for companies that allow you to find the most suitable products from the customer's point of view.

ЗМІСТ

Вступ.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Аналіз предметної області, яка визначає діяльність фірми	10
1.2 Аналіз реалізованих систем	17
1.3 Методи, технології та джерела інформації з дослідженням предметної області.....	17
1.4 Постановка задачі дослідження.....	19
2 ОГЛЯД МЕТОДІВ ТА ТЕХНОЛОГІЙ, ЯКІ ЗАСТОСОВУЮТЬСЯ В ПРЕДМЕТНІЙ ОБЛАСТІ.....	22
2.1 Методи, типи та особливості побудови рекомендаційних систем	22
2.2 Методи нечіткої логіки	26
2.3 Огляд можливих проектних рішень	27
2.3.1 Огляд мови програмування C#	27
2.3.2 Огляд СУБД MS SQL Server.....	29
2.3.3 Огляд можливостей розробки API	30
3 МАТЕМАТИЧНИЙ ОПИС ЗАДАЧІ.....	33
3.1 Опис методів нечіткої логіки.....	33
3.1.1 Фазифікація	35
3.1.2 Нечітке логічне виведення. Механізм Мамдані.....	38
3.1.3 Дефазифікація	41
3.2 Застосування інструментів нечіткої логіки до обраної предметної області.....	43
3.2.1 Визначення лінгвістичних змінних	43
3.2.2 Оголошення нечітких правил	47
3.2.3 Оператори нечіткої логіки.....	48
3.2.4 Алгоритм функціонування інструментарію нечіткої логіки	49
3.3 Побудова математичної моделі	27
4 ПРОГРАМНА РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНОГО ІНТЕРФЕЙСУ	56
4.1 Розробка бази даних на обраній платформі СУБД.....	56

4.2 Розробка коду програми з використанням обраної мови програмування.....	58
4.3 Опис розробленого рекомендаційного веб-API.....	65
4.4 Проведення експериментів стосовно надання рекомендацій	67
ВИСНОВКИ.....	71
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	73
ДОДАТОК А Графічні матеріали.....	76
ДОДАТОК Б Текст програми	85
ДОДАТОК В Відомість кваліфікаційної роботи	97

ВСТУП

Сьогодні сфера електронної комерції стрімко розвивається та набуває все більшої популярності.

Дедалі більше людей віддають перевагу покупкам через Інтернет, тому що це відносно зручно, надійно і швидко. Даний формат здійснення покупок дозволяє зберігати цінний час, запобігати ризику для здоров'я (як показав досвід пандемії COVID-19), порівнювати різні пропозиції, економити кошти, беручи участь у програмах лояльності тощо.

Електронна комерція має не тільки низку переваг, але й призводить до появи деяких труднощів: через інформаційне навантаження покупцю іноді складно знайти бажане. В той самий час фірми, що ведуть власну діяльність у режимі онлайн, перевантажені великими масивами даних. Тоді перед ними постає питання, як пропонувати клієнту саме ту продукцію, яка підходить саме йому.

Крім того, існує проблема неефективного використання великого обсягу інформації про товарний асортимент, що було отримано у ході онлайн-операцій, для підтримання рішень як покупцями, так і продавцями.

Аби вирішити ці проблеми, сучасні інформаційні системи електронної комерції добігають до застосування принципів масової персоналізації як до продукції, так і до її представлення в Інтернет-магазинах.

Одним із найефективніших способів її досягнення є використання рекомендаційних систем. Вони, як правило, аналізують дані про товари або взаємодію між користувачами та товарами, щоб знайти асоціації між ними, допомагають клієнтам врахувати їх потреби і вподобання.

Рекомендаційні сервіси дозволяють швидше знаходити найбільш підходящі товари з точки зору клієнта. Тут враховуються різні фактори: кращі продавці конкретного продукту, демографічна інформація покупців, а також аналіз поведінки

покупців в минулому для прогнозування їх купівельної поведінки в майбутньому. Крім того, полегшується доступ до відповідного контенту, що дозволяє краще задовольнити потребу клієнта. В результаті клієнт відчуває, що його інтереси краще зрозумілі, і готовий купувати більше продуктів або отримувати доступ до додаткового контенту.

Системи рекомендацій є цінними для компаній, оскільки вони отримують конкурентну перевагу завдяки релевантному контенту та мінімізують ризик відтоку клієнтів.

Оскільки інтереси користувачів змінюються з плином часу, існуючі системи рекомендацій іноді не можуть знайти актуальні товари для клієнтів. Для подолання цієї проблеми створюються системи рекомендацій на основі нечіткої логіки, яка динамічно прогнозує найбільш релевантні товари для покупців в інтернет-магазинах відповідно до поточних інтересів користувачів.

Таким чином, задоволеність і лояльність клієнтів можуть бути підвищені, а збільшення частоти відвідувань клієнтів може створити більше можливостей для здійснення угод і вигоду для інтернет-підприємств.

Отже, метою даної кваліфікаційної роботи є дослідження та розробка методів формування рекомендацій у інформаційних системах з використанням інструментарію нечіткої логіки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області, яка визначає діяльність фірми

Збільшення продажів – мета кожної компанії. Збільшення обсягів продажів може підвищити частку ринку, маржу та конкурентоспроможність.

Інтеграція інформаційних систем є основоположним елементом для отримання додаткового простору для розвитку компанії та об'єднує комплекс взаємопов'язаних процедур, а також генерує та розподіляє певну інформацію з використанням ІТ-інфраструктури. Основною метою ІС є підтримка прийняття рішень в компаніях, щоб бізнес-цілі досягалися швидше та ефективніше.

Задачею даної кваліфікаційної роботи магістра є розробка методів формування рекомендацій в інформаційній системі з використанням інструментів нечіткої логіки.

Система електронної комерції є одним з підвидів інформаційних систем, що функціонує у формі веб-ресурсів та має у своїй структурі рекомендаційні сервіси.

У даному випадку у якості ІС виступає «Інтернет-магазин косметичних засобів», що повинен забезпечувати роботу фірми, яка реалізує косметику та товари для краси у онлайн-режимі.

Основна функція даного підприємства – це продаж косметичних засобів, який можна реалізувати у роздрібний та оптовий способи.

Про актуальність обраної предметної області свідчить те, що індустрія краси розвивається дуже стрімко, наповнюючи ринок все більшим спектром різноманітних б'юті-продуктів. Велика кількість рекламних кампаній, залучення соціальних мереж зробили цей сектор продажів одним з найбільш затребуваних та прибуткових. Незважаючи на світову ситуацію, косметика завжди реалізується та має високий товарообіг, це свідчить про доцільність реалізації ІС такого типу.

У якості продукції розглядаються два типи косметики – декоративна та доглядова. Прикладами декоративних засобів є креми, маски, бальзами, шампуні,

гелі тощо, а доглядових – помади, тональні засоби, туші, олівці для очей чи брів. Косметику також можна поділяти за категоріями стосовно особливостей використання або застосування до окремих частин тіла або обличчя.

Цільова аудиторія (ЦА) представлена обома статями, незалежно від віку (дитяча косметика також є у наявності), хоча й лівовою часткою ЦА є саме жіноча стать.

Звичайно, через високу конкуренцію ведення бізнесу у цій ніші наражається на труднощі, але долучення надсучасних технологій, грамотна маркетингова кампанія та широка лінійка товарів дозволять їх подолати. Ключовим у цьому випадку також є використання у інформаційній системі новітніх методів та інструментів, до яких відносяться нечітка логіка та формування товарних або послугових рекомендацій на її основі.

У даному випадку Інтернет-магазин можна розглядати як прикладну систему, що побудована як елемент технології системи електронної комерції. Подібно до звичайного магазину, він реалізує такі основні функції: надання інформації про асортимент товарів, оформлення й обробку замовлень, реалізацію продукції, здійснення продажів. Таким чином користувачі можуть формувати список необхідних для придбання товарів і одразу робити замовлення [1].

Як і у звичайному фізичному магазині, під час здійснення діяльності у мережі Інтернет важливим залишається дотримання українського законодавства, відповідно до якого організоване документаційне забезпечення управління.

1.2 Аналіз реалізованих систем

Під час розробки ІС доцільним є розгляд вже існуючих систем-аналогів. Це надає можливість більш чітко сформулювати вимоги до майбутньої системи, виявити характерні особливості у даній предметної області та запобігти недоліків, що будуть виявлені у існуючих аналогах.

Як приклад розглянемо наступні системи електронної комерції продажів парфумерії та косметичних засобів: «MakeUp» (Україна) та «Douglas» (Німеччина).

Український представник «MakeUp» не має фізичних представництв та зосереджується на діяльності у мережі Інтернет за допомогою веб-сайту makeup.ua [2]. Вигляд головної сторінки сайту наведено на рисунку 1.1.

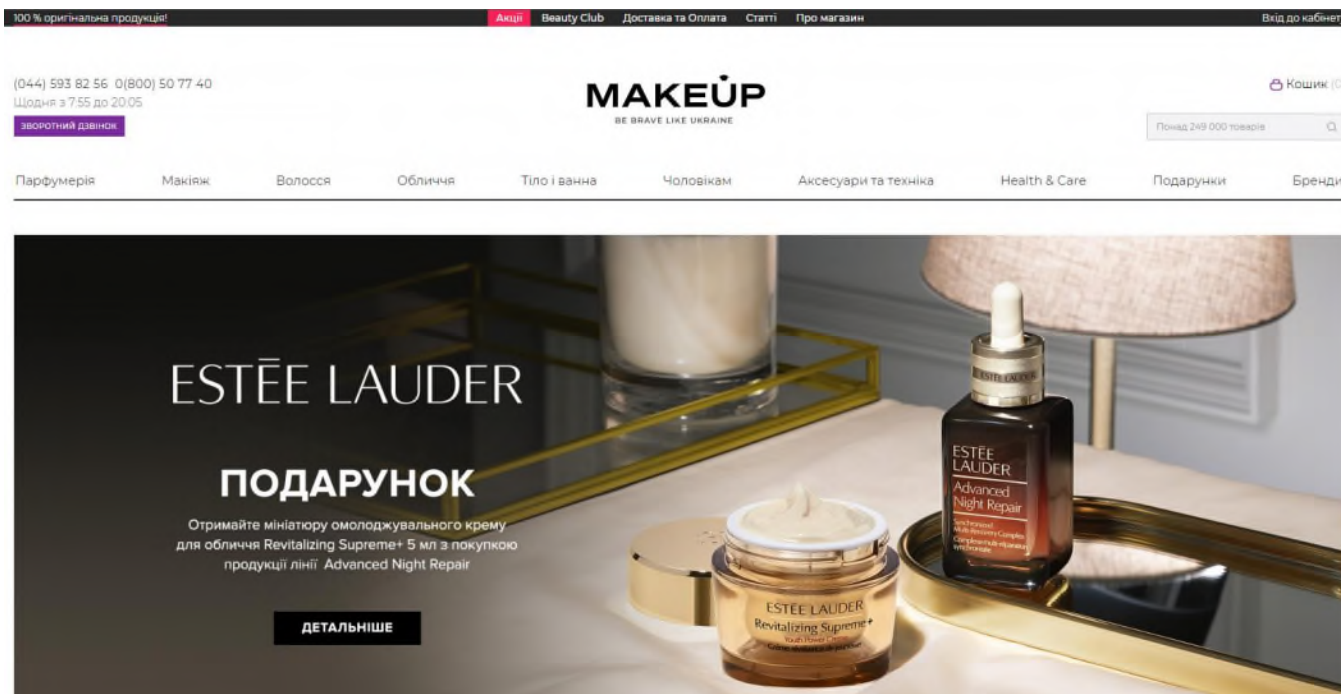


Рисунок 1.1 – Головна сторінка веб-сайту магазину «MakeUp»

Після детального огляду функціоналу даної системи електронної комерції можна виділити основні бізнес-функції для користувача ІС, а саме:

- реєстрація та авторизація у системі;
- надання загальної інформації про роботу магазину;
- перегляд каталогу товарів;
- перегляд детальної інформації про обраний товар;
- наповнення та редагування кошику товарів;
- формування замовлення;

- створення та скасування замовлення;
- відстеження змін замовлень;
- наявність доставки замовлення;
- вихід з системи.

Крім того було виявлено наступні особливості системи електронної комерції «MakeUp», що можна вважати її перевагами:

- можливість долучитися до низки програм лояльності;
- інформування про акційні пропозиції;
- зворотній зв'язок та відгуки інших покупців;
- надання переліку нових товарів;
- можливість фільтрації за категорією або виробником продукту;
- можливість пошуку товару за назвою або категорією;
- рекомендації схожих товарів із тим, що розглядається (див. рис. 1.2);
- рекомендації товарів, які користуються попитом у інших клієнтів (див. рис. 1.3);
- рекомендації товарів, які часто переглядаються (див. рис. 1.4);
- рекомендації товарів «Саме для Вас» (див. рис. 1.5).

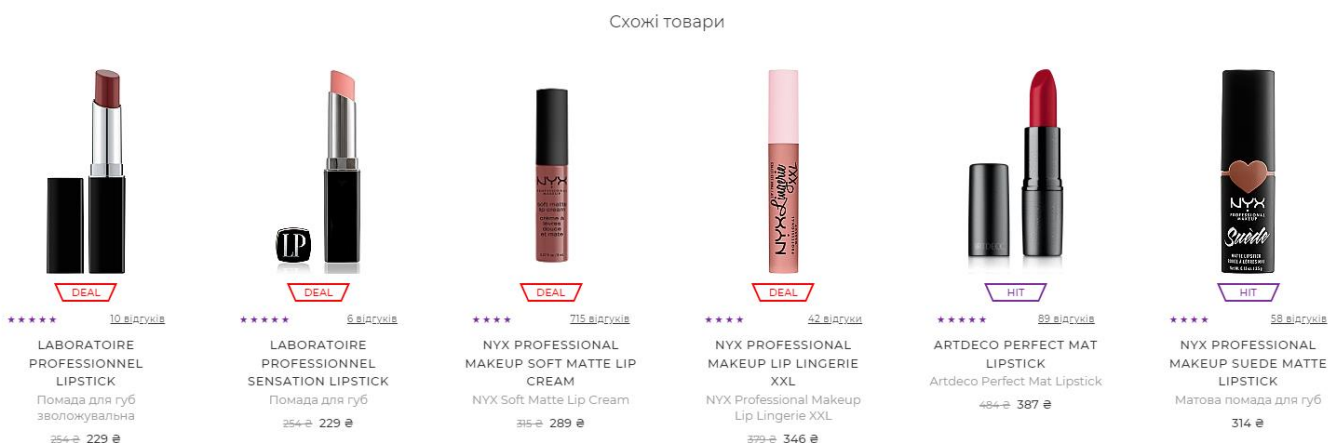


Рисунок 1.2 – Рекомендації схожих товарів на веб-сайті «MakeUp»

Інші клієнти також купили



Рисунок 1.3 – Рекомендації товарів «Інші клієнти також купили»

Інші клієнти також цікавились

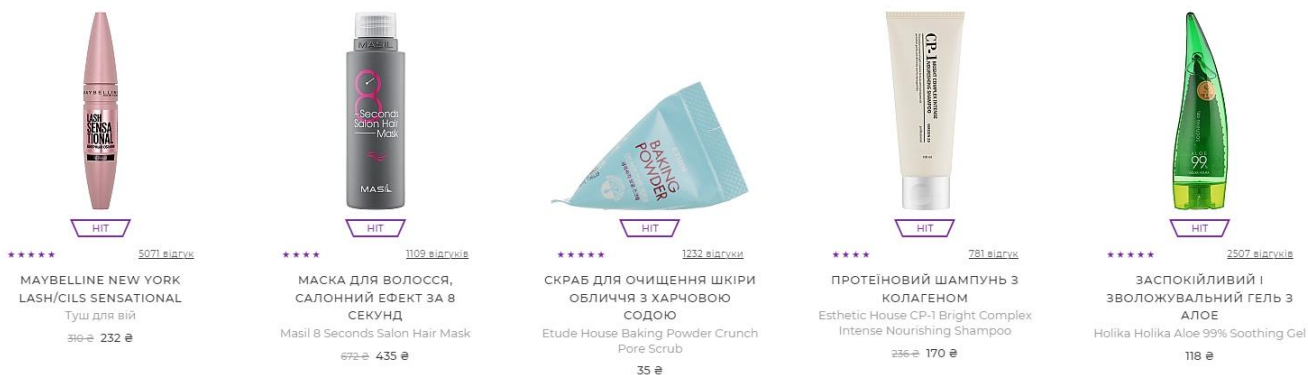


Рисунок 1.4 – Рекомендації товарів «Інші клієнти також цікавились»

Спеціально для вас

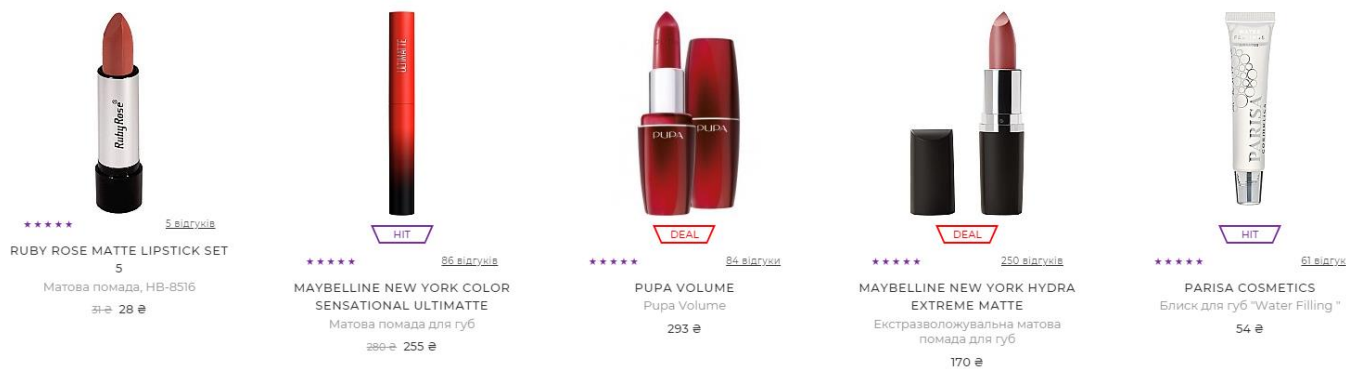


Рисунок 1.5 – Рекомендації товарів «Спеціально для Вас»

Що стосується недоліків системи, до них можна віднести відсутність персональних рекомендацій, засновуючись на вже придбаній продукції, а також на особистих особливостях покупця. Хоча система і пропонує списки товарів за популярністю придбання чи переглядів на сайті, а також списку «Для Вас», незрозумілим залишається логіка формування таких рекомендацій.

Крім того, наповнення списку персоналізованих пропозицій для клієнта визиває сумніви у тому, чи підійде ця продукція, бо ніде не було вказано власних особливостей чи вподобань покупця.

Даний функціонал був би доцільним, бо у даній предметній області мова йде про продажі косметичних засобів, у використанні та підборі яких дуже важливими аспектами є індивідуальність клієнта, безпечність та правильність застосування ним обраної косметики.

Електронна комерція у косметичній галузі стрімко розвивається і закордоном, через що має сенс розглядати іноземні системи-аналоги. Наприклад, одним з найбільших представників ринку Німеччини є магазин парфумерії та косметичних засобів «Douglas», що має філіали у багатьох німецьких містах та займається онлайн-дистрибуцією продукції через власний веб-сайт (див. рис. 1.6) [3].

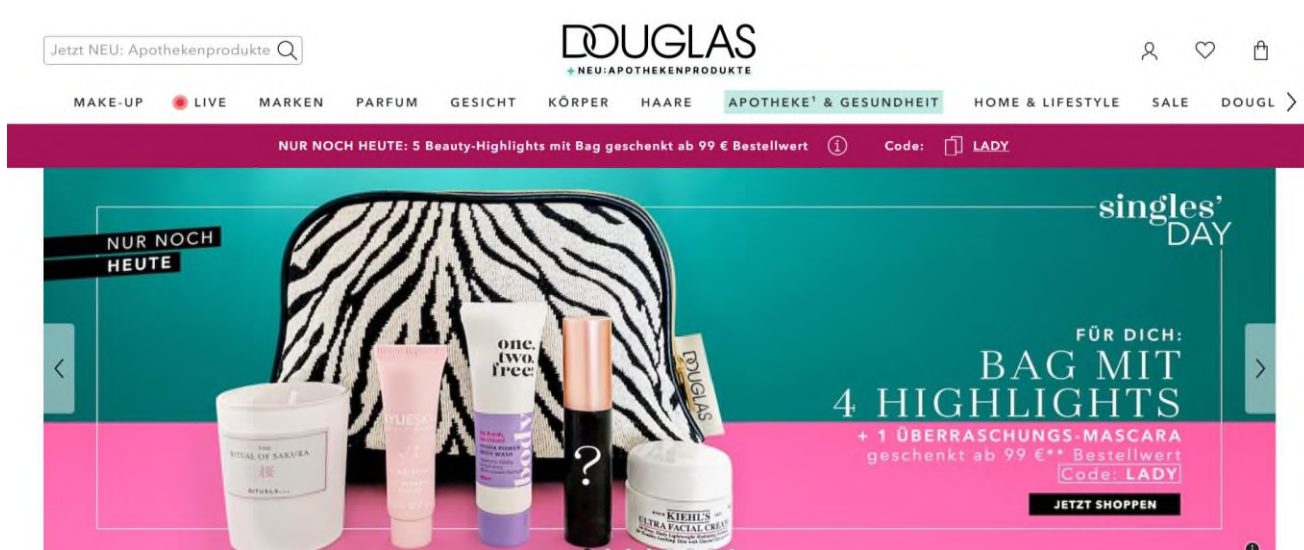


Рисунок 1.6 – Головна сторінка веб-сайту магазину «Douglas»

Як і у розглянутому вище українському прикладі інформаційної системи з продажів парфумерії та косметичних засобів, у «Douglas» наявні наступні бізнес-процеси:

- інформування про діяльність магазину;
- можливість створити акаунт, або за його наявності, увійти до системи;
- надання каталогу товарів у вигляді списку представлених брендів;
- пошук товару за ключовими словами або використовуючи фільтри;
- створення списку вподобаних товарів;
- створення та редагування кошику товарів;
- оформлення замовлення із відстеженням змін у особистому кабінеті;
- можливість доставки або самовивозу замовлення із обраного філіалу;
- наявність програми лояльності;
- інформування про акційні пропозиції;
- вихід із системи.

Розглядаючи аспект наявності рекомендацій, було виявлено два типи: перший – запропонування інших товарів тієї самої торгової марки, продукт якої розглядається (див. рис. 1.7), та другий – перелік схожої продукції (див. рис. 1.8).

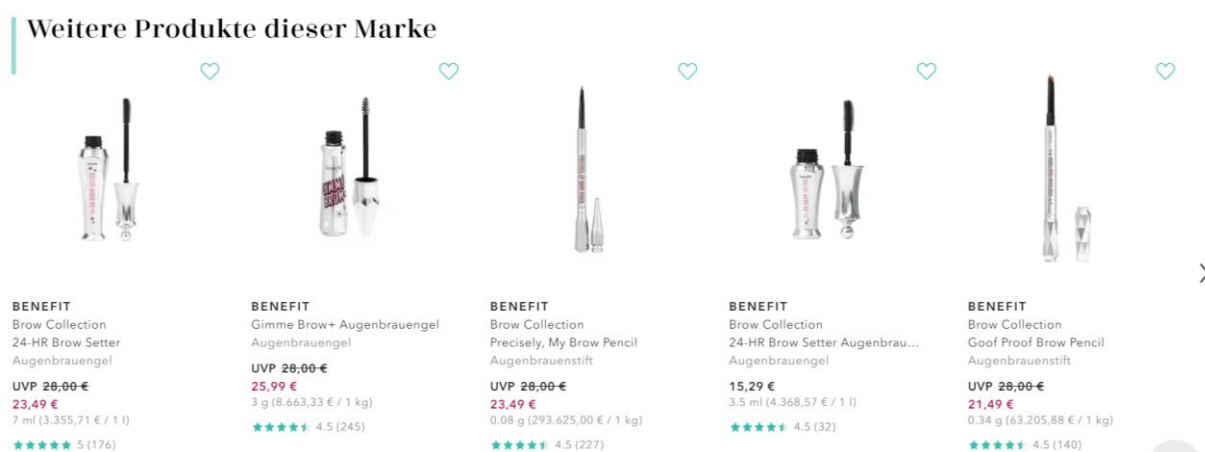


Рисунок 1.7 – Рекомендації товарів окремої марки на веб-сайті «Douglas»

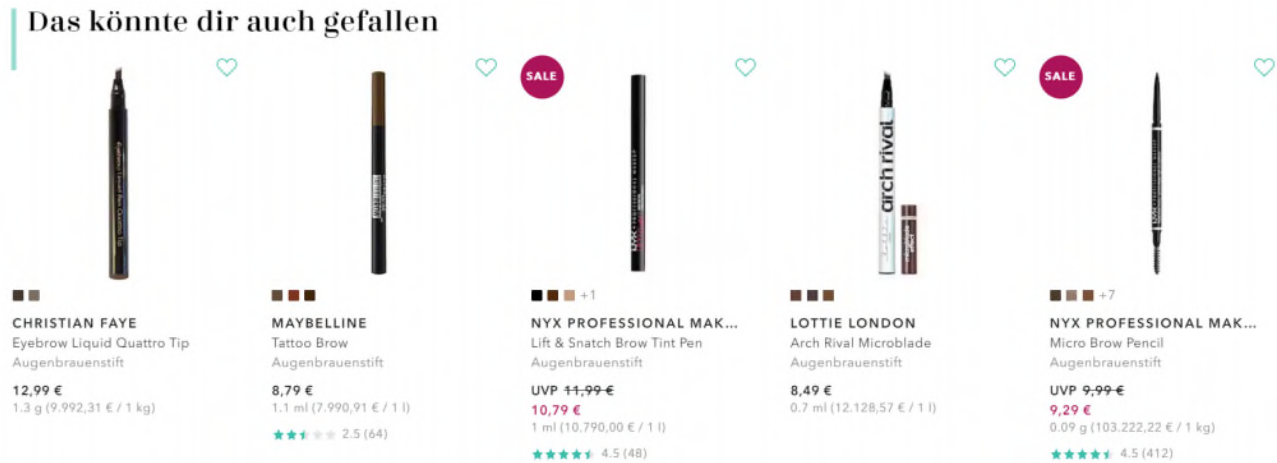


Рисунок 1.8 – Рекомендації схожих товарів на веб-сайті «Douglas»

Веб-сайту магазину парфумерії та косметичних засобів «Douglas» надає доволі широкий функціонал та має реалізованими усі основні бізнес-функції, який повинен мати сучасна ІС електронної комерції.

Недоліками можна чітко назвати ускладнену навігацію веб-сторінками через зовнішню кількість елементів інтерфейсу. Поміж цього відсутність персоналізованих рекомендацій чи пропозицій товарів, які обрали покупці зі схожими смаками, теж погіршує враження від магазину «Douglas».

Таким чином, аналіз обох розглянутих систем-аналогів надав змогу виявити їх особливості, які є як сильними, так і слабкими сторонами даних систем. Під час виконання кваліфікаційної роботи магістра потрібно звернути на це увагу, а також виправити знайдені недоліки.

1.3 Методи, технології та джерела інформації з дослідженням предметної області

Впровадження систем персоналізованих рекомендацій почалося відносно нещодавно, але дане питання вже дуже активно досліджується, створюються наукові

статті та роботи. Накопичений фахівцями досвід дозволяє детально оглянути цю задачу.

Вперше рекомендаційна система була згадана в технічному звіті Юссі Карлгрена «Алгебра рекомендацій» як «цифрова книжкова полиця» в 1990 році в Колумбійському університеті [5]. У 1994 він опрацював та реалізовував це питання у його наступному дослідницькому звіті у SICS «Кластеризація груп новин на основі поведінки користувачів – алгебра рекомендацій» [6].

Надалі все більше дослідницьких груп розглядали тему реалізації рекомендаційних структур у різноманітних системах. Наприклад, у 1995 була видана книга «Recommending and Evaluating Choices in a Virtual Community of Use» американських авторів Гілла, Стеада, Розенштайна та Фурнаса [7].

У 2003 році Монтанер надав перший огляд рекомендаційних систем з точки зору інтелектуальних агентів у книзі «Таксономія рекомендаційних агентів у Інтернеті» [8].

Потім Адомавічюс надав новий, альтернативний огляд рекомендаційних систем [9].

Херлокер надав додатковий огляд методів оцінки для рекомендаційних систем [10], а Біл та ін. обговорили проблеми офлайн-оцінок [11].

Біл та його колеги також надали огляд літератури про наявні системи рекомендацій наукових робіт та існуючі проблеми [12].

У наступні роки було створено наступні фундаментальні роботи: Recommender Systems Handbook [13], що є одним із найвідоміших довідників з рекомендаційних систем. У цій роботі було систематизовано все різноманіття методологій і концепцій, які стосуються систем персоналізованих рекомендацій у різних прикладних галузях, як аналіз даних, системи ухвалення рішень, маркетинг, статистика.

Наведений довідник також містить практичне застосування підходів, що використовуються, наприклад, відомими компаніями такими, як «Google» та «Amazon» у власних рекомендаційних системах.

У подальших дослідженнях можна користатися матеріалами з наукової праці «Recommender Systems: The Textbook», де висвітлено різні фундаментальні алгоритми та методи оцінювання точності їхньої роботи. Крім того, висвітленими є практичні аспекти створення рекомендаційних систем, як надійність та захист, ранжування результатів [14].

1.4 Постановка задачі дослідження

Після того, як проаналізовано предметну область, знайдено особливості майбутньої створюваної системи, необхідно чітко визначити питання, які будуть вирішені у подальшій роботі.

Задачею кваліфікаційної роботи магістра є розробка методів створення рекомендаційного інтерфейсу у інформаційній системі «Інтернет-магазин косметичних засобів» із використанням інструментів нечіткої логіки.

Під час здійснення покупок у даній ІС користувач отримує персоналізовані рекомендації окремих косметичних засобів та послуг залежно від обраної категорії.

Отже, до завдань, які необхідно реалізувати віднесено наступні:

1. Створення бази даних, до якої будуть збережені усі вхідні дані у системі, зібрана інформація про асортимент товарів і послуг магазину. З огляду можливостей реалізацій найкращим варіантом є створення БД на платформі СУБД MS SQL Server.

2. Збір вхідних даних.

- а) Отримання даних від користувача з метою інформування про його персональні особливості покупця. Тут необхідно ввести значення до запропонованих полів у формі, що стосуються статі, віку, стану шкіри, волосся тощо.

б) Формування списку послуг та товарів, що надає Інтернет-магазин косметичних засобів.

3. Формування бази нечітких правил.

а) Пошук та залучення експертної комісії, що знається на предметній області.

б) Формулювання нечітких правил даною комісією та утворення бази нечітких правил.

в) Проведення маркування асортименту товарів, на основі чого у подальшому надаються поради.

4. Створення математичного опису розроблюваної системи.

а) Розглядання та вибір нечітких методів та алгоритмів для реалізації інтерфейсу з надання рекомендацій.

б) Розробка моделі функціонування рекомендаційного інтерфейсу із використанням механізму нечіткої логіки. Залучення одного з обраних алгоритмів fuzzy logic для коректного надання повної вихідної інформації, а саме списку рекомендованих косметичних товарів та послуг.

5. Програмна реалізація рекомендаційного веб-API з використанням обраної мови програмування, що включає розробку алгоритму та написання програмного коду мовою C#.

6. Тестування розробленого рекомендаційного веб-інтерфейсу та отримання висновків про загальну ефективність створеної системи.

7. Оцінка подальшого розвитку розробленої системи.

Зауважимо, що під час реалізації рекомендаційного елемента інформаційної системи, треба звернути увагу на такі виклики, пов'язані із його подальшим використанням:

– недостатність даних: рекомендації можуть бути надані лише за наявності достатньо якісної інформації, тому з самого початку потрібно приділити увагу правильному вибору та збору відповідних даних. Тут треба також врахувати, що іноді клієнт не хоче надавати приватну інформацію або він не знає, як відповісти на окремі питання;

– суперечливість даних: людина іноді може помиляться або мати складнощі під час надання вірних даних про себе (наприклад, тип шкіри), такі дані можуть суперечити один одному;

– неоднозначність даних: якщо маркування товарів зроблено неточно, це може призвести до того, що однакові продукти з різним маркуванням будуть неправильно оцінюватися. Як наслідок, клієнту може бути запропонована продукція, яка не відповідає його потребам.

– масштабованість: різноманітність продуктів та клієнтів сьогодні переповнює багато компаній у сфері електронної комерції. Тому використання зростаючих масивів даних для рекомендаційних систем стає викликом.

2 ОГЛЯД МЕТОДІВ ТА ТЕХНОЛОГІЙ, ЯКІ ЗАСТОСОВУЮТЬСЯ В ПРЕДМЕТНІЙ ОБЛАСТІ

2.1 Методи, типи та особливості побудови рекомендаційних систем

Більшість сучасних інформаційних систем сьогодні мають спеціальне програмне забезпечення, що використовує зібрану інформацію, аби спрогнозувати інтереси користувача і персоналізувати електронну комерцію та спеціалізовані додатки – системи рекомендацій.

Метою рекомендаційного сервісу є підбір об'єктів, які є найбільш релевантними для користувача з набору всіх доступних, це може бути будь-що: кінофільм, пісня, елемент одягу або стайлінговий спрей для волосся. Бути найбільш релевантним означає максимально відповідати смаку, інтересам або потребам клієнта.

Історично рекомендаційні системи народилися як відгалуження завдань пошуку. У пошуку є запит та контекст у вигляді вподобання користувача та його характеристики, а також є елементи, які потрібно ранкувати. У рекомендаціях усе аналогічно, але запитом виступає сам користувач, його вподобання [15].

Сьогодні системи рекомендацій розроблені набагато складніше і використовують поточну та історичну поведінку користувачів. Для цього використовується як інтерес, так і поведінка використання вже «оціненого» об'єкта. При цьому для розрахунку рекомендації використовується інформація про неявну взаємодію, таку як перегляд або покупка (про що можна було запевнитись під час огляду існуючих систем-аналогів) або явну (конкретна оцінка, наприклад, 5 зірок).

Ці рейтингові дані в кінцевому підсумку надходять як вхідні дані в алгоритми машинного навчання, які дозволяють розраховувати рекомендації.

Існують різні типи рекомендаційних систем: на основі змісту, колаборативні, гібридні та контекстно-орієнтовані (сесійні) (див. рис. 2.1) [16].



Рисунок 2.1 – Класифікація рекомендаційних систем

Рекомендаційні системи, засновані на змісті (англ. content-based filtering) беруть описову інформацію про окремі об'єкти за основу для розрахунку рекомендацій. Це може бути, наприклад, порівняння описових текстів товару або схожість зображень цього товару.

Для цього визначається схожість між усіма об'єктами (за допомогою міри відстані, наприклад, відстані Жаккара), а також історичні взаємодії користувачів, які слугують основою для розрахунку індивідуальних переваг користувача.

Колаборативна, або спільна фільтрація (англ. collaborative filtering) рекомендує об'єкти на основі користувачів зі схожою поведінкою оцінювання. Найпростішим типом колаборативної фільтрації є фільтрація на основі товарів, де використовується вподобання користувача (історична взаємодія з товаром, наприклад, купівля, перегляд тощо), які множаться на матрицю схожості [20]. Така матриця описує,

наскільки схожі об'єкти, і обчислює, які продукти найчастіше купують разом користувачі на основі взаємодій. Приклад матриці схожості наведено на рисунку 2.2.

		Об'єкт рекомендації			
Користувач	1	1	3	0	4
	5	5	5	2	0
	2	2	1	1	0
	0	0	4	0	3

Рейтинги / вподобання

Рисунок 2.2 – Матриця схожості

У колаборативних системах часто використовується метод найменших квадратів з чергуванням (англ. Alternating Least Squares, ALS), зокрема, відомий своєю надзвичайно високою якістю рекомендацій. Даний алгоритм використовує як взаємодію користувача, так і взаємодію товару і, таким чином, може визначати більш точні рекомендації. Більш наочно розглянуті два методи представлені на рисунку 2.3.

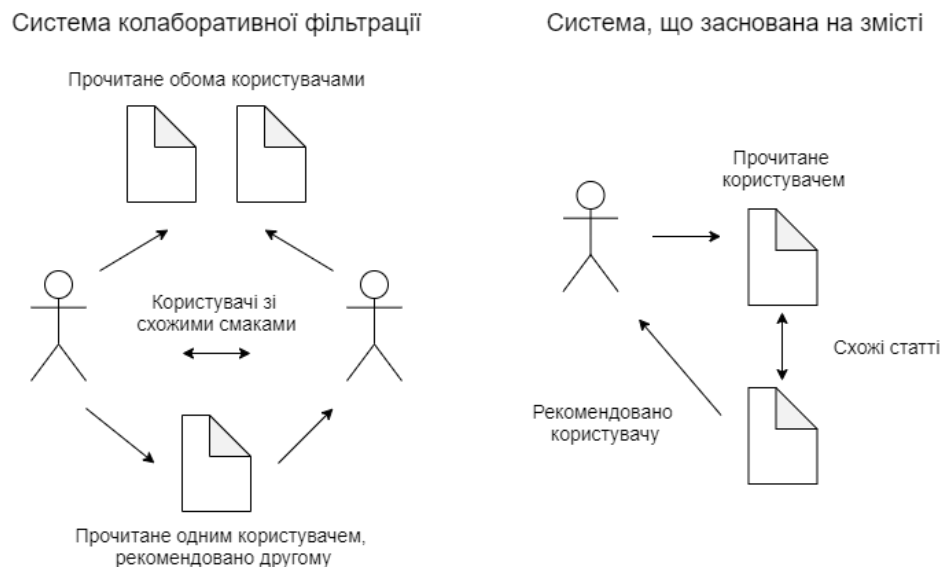


Рисунок 2.3 – Порівняння рекомендаційних систем двох типів

Наступний тип сервісів рекомендацій – сесійний, використовує взаємодію користувача в межах сеансу. Такі рекомендаційні системи використовуються, зокрема, на платформах електронної комерції або стрімінгових сервісів.

Даний підхід має, перш за все, таку перевагу, що не потрібно мати історичних даних про кліки або покупки.

Більшість рекомендаційних сервісів, що базуються на сеансових рекомендаціях, використовують лише дані з послідовності останніх взаємодій сеансу. В основному використовуються методи, засновані на послідовних моделях, таких як рекурентні нейронні мережі, трансформатори або інші підходи глибокого навчання.

Гібридні сервіси рекомендацій поєднують фільтрацію на основі контенту та спільну фільтрацію для покращення якості рекомендацій. Наприклад, проблема так званого «холодного старту» системи спільних рекомендацій (коли користувач або об'єкт ще дуже невідомий або новий) зведена до мінімуму, що забезпечує більш швидке надання відповідних рекомендацій користувачам.

Гібридні сервіси рекомендацій поєднують фільтрацію на основі контенту та спільну фільтрацію для підвищення якості рекомендацій. Наприклад, проблема «холодного старту» системи спільних рекомендацій (коли об'єкт чи товар ще дуже невідомий або новий) зведена до мінімуму, що забезпечує більш швидке надання відповідних рекомендацій користувачам.

Гібридні рекомендаційні системи поєднують в собі методи на основі контенту та спільної роботи. Гібридні підходи в основному використовують крупні корпорації, що володіють великою кількістю інформації як про своїх користувачів, так і про їхній контент.

2.2 Методи нечіткої логіки

Нечітка логіка, або англійською *fuzzy logic*, є добре відомою інженерам-програмістам систем управління як зручний засіб програмування та моніторингу додатків управління технологічними процесами [22].

За аналогією з традиційними засобами управління технологічними процесами, системи на основі нечіткої логіки можуть використовуватися для опису процесів регулювання та брати участь у обчисленні керуючого впливу відповідно кількості точок завдання для одного чи декількох вимірювань.

До переваг систем з нечіткою логікою у порівнянні до звичайних систем можна віднести можливість роботи зі значеннями, що безперервно змінюються в часі, та значеннями, які неможливо задати однозначно (наприклад, результати статистичних опитувань, рекламні компанії тощо).

Нечітка логіка також дозволяє оцінювати та порівнювати нечітко формалізовані критерії, проводити якісні оцінки вхідних і вихідних результатів.

Крім того, існує можливість проведення швидкого моделювання складних динамічних систем та їхній порівняльний аналіз із заданим ступенем точності, оперуючи принципами поведінки системи, описаними нечіткими методами. У цьому разі, по-перше, не витрачається багато часу на з'ясування точних значень змінних і складання описувальних рівнянь опису, по-друге, стає можливим оцінити різні варіанти вихідних значень [23].

Нечіткі системи стають дуже популярними в наш час, використовуючись, наприклад, в експертних системах, прогнозуванні та прийнятті рішень. Основні застосування можна класифікувати як системи нечіткого виводу (СНВ). СНВ складається з двох основних компонентів - бази даних, що є сховищем лінгвістичних змінних, та бази правил - сховища знань, які представляє система, у вигляді нечітких правил.

Правила нечіткої логіки дозволяють забезпечити наступне:

- застосування наявного досвіду управління;
- використовувати гнучкі правила у разі неможливості точно моделювати систему за допомогою традиційних засобів;
- покращення якості управління за допомогою саморегулювання системи керування;
- покращення якості управління за допомогою так званої функції попередження, базуючись на подіях, які не можуть бути враховані у разі застосування традиційних способів керування.

Нечітка логіка, крім цього, дозволяє отримати максимальну користь від практичного досвіду та забезпечити відсутність втрат.

Етапу накопичення досвіду та виявлення корисних нововведень необхідно приділяти максимальну увагу, оскільки упущення важливих особливостей функціонування систем може призвести до некоректної обробки. У зв'язку з цим даний етап розробки системи має виконуватися вручну з особливою ретельністю.

За наявності практичного досвіду можна описати цей досвід у вигляді правил нечіткої логіки. Коли система є складною та/або немає готового досвіду її створення чи моделювання, необхідно використовувати глобальний підхід до описів частин цієї системи. Методи нечіткої логіки не замінюють традиційних підходів до створення систем управління, а навпаки, доповнюють їх.

2.3 Огляд можливих проектних рішень

2.3.1 Огляд мови програмування C#

C# - це одна з сучасних кросплатформних та об'єктно-орієнтованих мов програмування, що дозволяє розробникам створювати численні безпечні та надійні додатки, які працюють у середовищі .NET.

C# надає мовні конструкції для безпосередньої підтримки цих концепцій, що робить її природною мовою для створення та використання програмних компонентів.

Деякі функції C# допомагають створювати стабільні та довговічні додатки, а саме наступні:

- збірка сміття, що автоматично повертає пам'ять, зайняту недосяжними, невикористовуваними об'єктами;
- нульові типи, що забезпечують захист від змінних, які не посилаються на виділені об'єкти;
- обробка виключень, яка забезпечує структурований та розширюваний підхід до виявлення та відновлення помилок;
- лямбда-вирази, які підтримують методи функціонального програмування;
- мова інтегрованого синтаксису запитів (LINQ), що створює загальний шаблон для роботи з даними з будь-якого джерела;
- мовна підтримка асинхронних операцій, яка надає синтаксис для побудови розподілених систем;
- наявність єдиної системи типів. Усі типи C#, включаючи `int` та `double`, успадковуються від єдиного кореневого типу об'єкту та мають набір спільні операції. Значення будь-якого типу можуть зберігатися, транспортуватися та редагуватися єдиним чином [18].

Окрім того, C# робить акцент на випусках версій, щоб гарантувати, що програми та бібліотеки можуть розвиватися з часом сумісним чином. Аспекти розробки, на які безпосередньо впливають міркування контролю версій, включають окремі віртуальні та перевизначальні модифікатори, правила вирішення переважань та підтримку явних оголошень членів інтерфейсу.

Програми мовою C# виконуються на основі .NET, віртуальної системи виконання Common Language Runtime (CLR) та бібліотек класів.

CLR є реалізацією міжнародного стандарту Microsoft Common Language Infrastructure (CLI), що, у свою чергу, є основою для створення середовищ виконання та розробки, в яких мови та бібліотеки безперешкодно працюють разом.

Однією з найважливіших особливостей .NET є сумісність мов. Код, що створений на C#, може взаємодіяти з кодом F#, Visual Basic або C++.

Крім служб часу виконання, .NET також містить великі бібліотеки. Ці бібліотеки підтримують багато різних робочих навантажень. Вони організовані в простори імен, які забезпечують широкий спектр корисних функцій. Бібліотеки включають в себе все - від введення і виведення файлів до редагування рядків, аналізу XML, фреймворків веб-додатків і елементів управління Windows Forms. Типовий додаток на C# широко використовує бібліотеку класів .NET для виконання рутинних операцій.

2.3.2 Огляд СУБД MS SQL Server

Реляційні бази даних вже давно є стандартом, коли інформація зберігається у вигляді таблиць. Така БД використовує фільтри для пошуку, збору та сортування інформації на основі певних полів, що дозволяє створювати користувацькі звіти.

Наявні сьогодні програмні рішення використовують мову структурованих запитів SQL як стандартну мову для взаємодії баз даних.

Microsoft SQL Server є багатофункціональною реляційною системою управління базами даних (СУБД), що розроблена та впроваджена компанією Microsoft і дозволяє зручно обробляти великі бази даних.

SQL Server Database Engine є основною компонентою SQL Server, що відповідає за контроль, обробку та безпеку зберігання даних. Движок бази даних розділений на два сегменти. Один з них - це реляційний механізм, який використовується для обробки команд і запитів, а інший - механізм зберігання, який

призначений для управління різними функціями бази даних, такими як таблиці, сторінки, файли, індекси і транзакції.

Для адміністрування даних клієнтів, обробки замовлень або фінансового обліку, Microsoft SQL Server використовується різними компаніями в найрізноманітніших сферах небезпідставно.

Цю СУБД можна легко встановити за допомогою майстра установки. На відміну від інших серверів баз даних, які вимагають великої конфігурації командного рядка, MS SQL Server пропонує зручний інтерфейс установки. Необхідні оновлення завантажуються автоматично за допомогою майстра інсталяції, що зменшує ручне навантаження.

Завдяки інтегрованим прозорим функціям стиснення і шифрування даних, MS SQL Server пропонує переконливу продуктивність, яка залишає інші рішення в тіні. Користувачам не потрібно налаштовувати зовнішні програми для резервного копіювання та шифрування даних. СУБД надає ефективні інструменти управління дозволами з контролем доступу, які допомагають користувачам захистити конфіденційну ділову інформацію.

З-поміж усього, база даних має високий рівень захисту і використовує складні алгоритми шифрування, які практично унеможливають злам рівнів безпеки.

У разі чого за допомогою сучасних засобів відновлення можна відновити всю базу даних. Database Engine контролює зберігання даних і допомагає виконувати запити користувачів, включаючи транзакції, файли та індекси. Великі організації часто використовують ці можливості SQL Server [19].

2.3.3 Огляд можливостей розробки API

Без добре задокументованих інтерфейсів програмування будь-які інформаційні технології сьогодні вже майже не існують. Інтерфейс програмування (Application

Programming Interface, API) надає можливість різним програмам і сервісам взаємодіяти один з одним.

Головною метою використання API є впровадження функцій стороннього застосування до власного продукту.

Практично кожна програма отримує доступ до джерел даних та інших систем через API. У зв'язку зі зростаючим поширенням сервіс-орієнтованих та мікросервісних архітектур і пов'язаним з цим поділом, збільшилася кількість API, які задіяні у кожному варіанті використання.

У минулому опис API був дуже складним через різні технології та мови програмування. Зараз парадигмою програмування для розробки API є так звана «передача репрезентативного стану» REST (Representational State Transfer), що використовують великі відомі корпорації та компанії [20].

Раніше інтерфейси описувалися мовою опису веб-сервісів WSDL, але від цього відійшли через громіздкість. Мова опису веб-додатків (WADL) повинна була вирішити цю проблему, але так і не змогла бути стандартизована через свою XML-структуру.

Через це чи не найпопулярнішою технологією серед API документації став Swagger, що також відомий як специфікація OpenAPI.

Swagger - це потужний, але простий у використанні набір інструментів для розробників API для команд і окремих користувачів, що дозволяє здійснювати розробку протягом усього життєвого циклу API, від проектування і документації до тестування і розгортання [21].

Swagger складається з поєднання відкритих, безкоштовних і комерційно доступних інструментів, які дозволяють будь-кому, від технічних інженерів до вуличних розумних менеджерів продуктів, створювати якісні API.

Протоколи, обробка помилок та модульність коду - це лише деякі з питань, які необхідно вирішити перед створенням API. Swagger надає інструменти для швидкого створення прототипів та побудови функціональності API.

Також при створенні API, Swagger Tooling можна використовувати для автоматичної генерації документа Open API на основі самого коду. Це неофіційно називається розробкою API "знизу вгору". Хоча сам програмний код може точно представляти документ Open API, багато розробників API вважають це застарілим методом, оскільки він вбудовує опис API у вихідний код проекту і, як правило, не розробникам складніше зробити свій внесок у нього.

Swagger забезпечує також взаємодію з API. Завдяки проекту Swagger Codegen кінцеві користувачі генерують клієнтські SDK безпосередньо з документа Open API, що зменшує потребу в клієнтському коді, створеному людиною.

Крім того, розробники можуть використовувати Swagger Codegen, щоб відокремити вихідний код від документа Open API і генерувати клієнтський і серверний код безпосередньо з чернетки. Хоча це вважається складним, багато галузевих експертів вважають його більш сучасним робочим процесом API і надає більше свободи в розробці API за рахунок перенесення аспекту кодування.

3 МАТЕМАТИЧНИЙ ОПИС ЗАДАЧІ

3.1 Опис методів нечіткої логіки

Теоретично кожний елемент множин або належить множині, або ні. Поняття множин використовується у багатьох математичних теоріях. Воно важливе, проте не розглядає прості ситуації, де усе ясно і зрозуміло.

Теорія нечітких множин базується на понятті часткової приналежності до множини: кожен елемент належить до нечіткої множини трохи або частково. Обрис нечіткої множини (див. рис. 3.1) не має «явної» межі, а видається «нечітким» або «розмитим» [24].

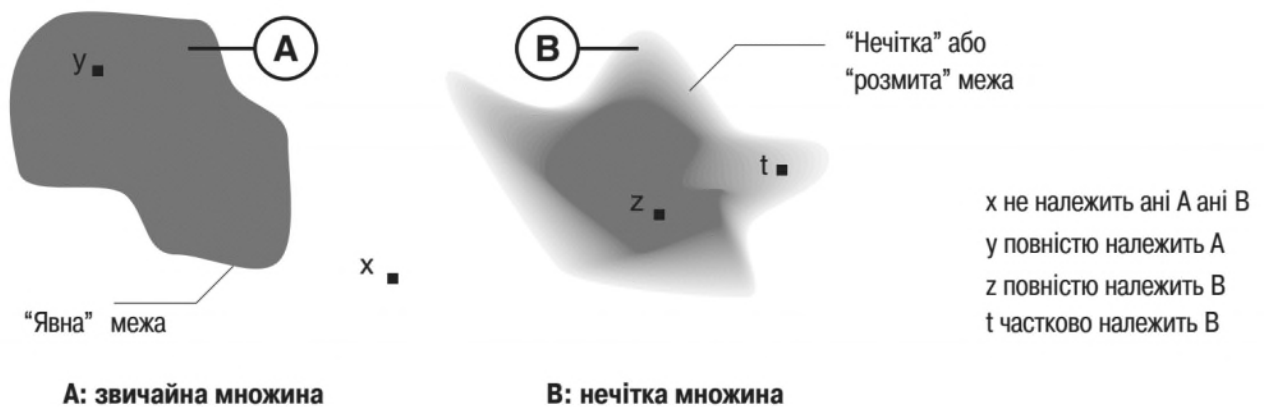


Рисунок 3.1 – Порівняння звичайної та нечіткої множини

З метою формалізації та застосування людського висновку формуються базові правила нечіткої логіки, які утворюють бази правил.

Бази правил нечіткої логіки є набором правил, які зазвичай використовуються паралельно, але в деяких системах можуть бути об’єднані.

У таких базах нечітких правил використовуються функції приналежності для системних змінних і правила, які можна записані у текстовому вигляді, як «IF «твердження» THEN «результат».

Бази правил нечіткої логіки, подібно до традиційних експертних систем, ґрунтуються на базі знань, побудованої на основі людського досвіду.

У той самий час існують суттєві відмінності у обробці та характеристиках цих знань. Коли експертна система оперує великою кількістю правил та займається їх послідовною обробкою, база правил нечіткої логіки має небагато правил та обробляє їх одночасно [17].

Особливістю такої бази знань у порівнянні до звичайної системи є також наявність інтерполяції між правилами, коли вони суперечать один одному. У експертній системі це відсутнє.

Процес обробки нечітких множин складається із трьох частин (див. рис. 3.2): постановки задачі, або фаззифікації, процесу нечіткого логічне виведення та дефаззифікації.

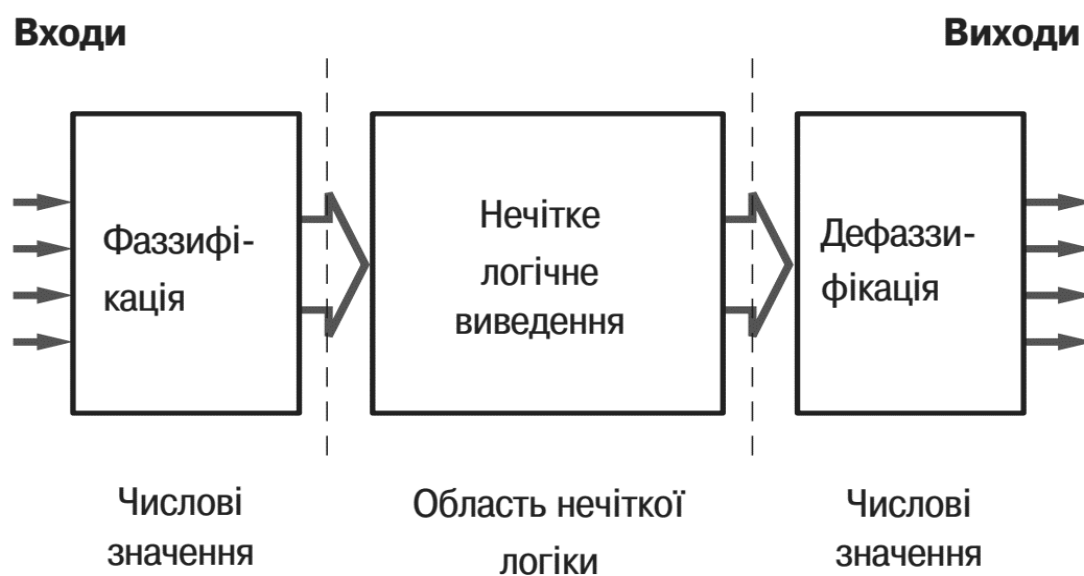


Рисунок 3.2 – Процес обробки нечіткої логіки

3.1.1 Фаззифікація

Фаззифікація – це процес перетворення чіткого вхідного значення в нечітке значення, який виконується з використанням інформації, що міститься в базі знань. Під час підготовки задачі для вирішення методами нечіткої логіки необхідно визначити ступінь приналежності змінної до нечіткої множини, що використовуються в затвердженнях правил [25].

Нечітка множина визначається за допомогою «функції приналежності», що відповідає поняттю «характеристична функція» у класичній логіці.

Припустимо, завданням є визначити множину людей «середнього зросту». У класичній логіці треба визначити, що середнім зростом вважаємо зріст між 1.60 м та 1.80 м. Характеристична функція у разі, що наведено на рисунку 3.3 надає значення «0» усім, хто відмінний від заданого діапазону зросту і «1» за умови відповідності заданому. Нечітка кількість людей «середнього зросту» задається за допомогою функції приналежності, яка в цьому випадку може набувати значення в діапазоні $[0;1]$ [25].



Рисунок 3.3 – Характеристична функція

Кожен можливий зріст у разі буде представлений у якості функції приналежності до нечіткої множини людей «середнього зросту» (див. рис. 3.4), що має значення між 0 і 1.

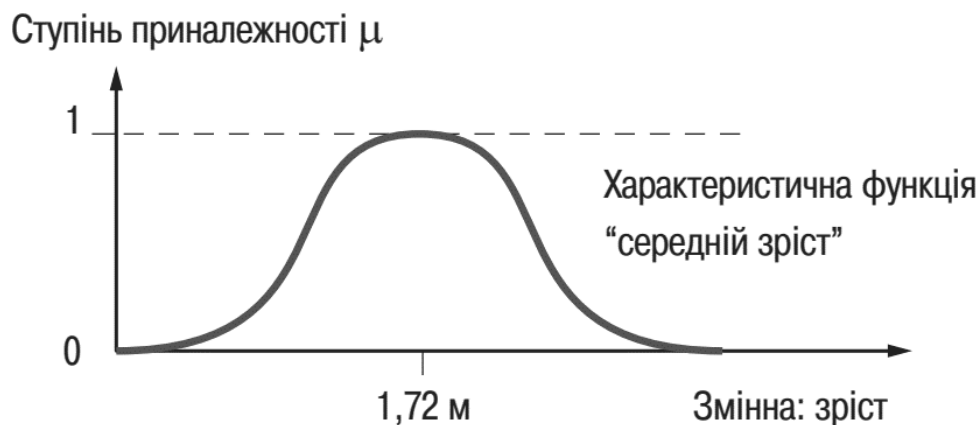


Рисунок 3.4 – Функція приналежності

Декілька нечітких множин можна визначити через одну змінну, яка набуває кількох значень, наприклад, «маленький зріст», «середній зріст» і «великий зріст». Кожне поняття визначає функція приналежності (див. рис. 3.5).

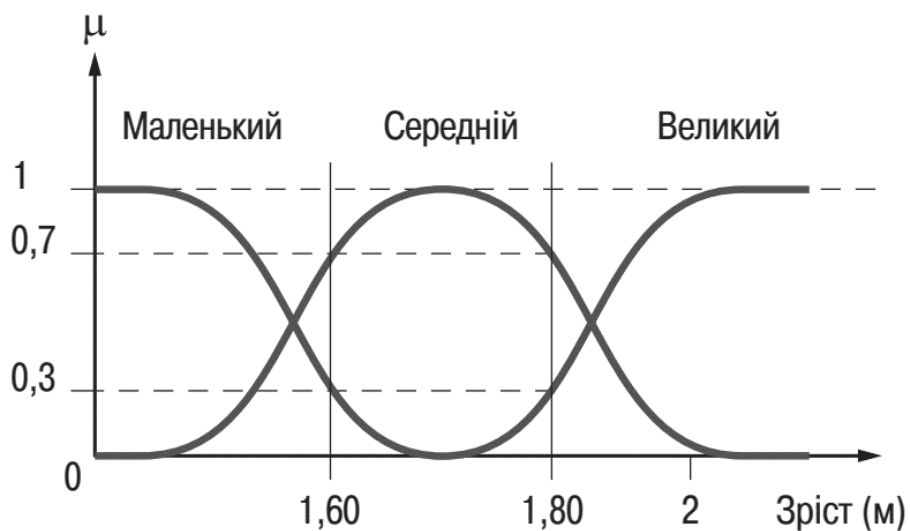


Рисунок 3.5 – Функція приналежності, змінна та лінгвістичний опис

Цей приклад показує можливість застосування нечіткої логіки. Людину зі зростом 1.80 м можна віднести до групи «великий зріст» зі ступенем належності 0.3 та до групи «середній зріст» зі ступенем належності 0.7.

У класичній логіці перехід від середнього росту до великого є миттєвим, тобто зріст 1.80 м швидше за все належить до групи «середній», а 1.81 м – швидше за все до «високий». Змінна (зріст) та її описи (середній, великий), що визначені функцією приналежності, є лінгвістичною змінною та лінгвістичними описами відповідно. Обидва поняття можуть бути використані безпосередньо в правилах.

Функція приналежності може набувати будь-якої форми. Найчастіше для їх подання використовуються шматково-лінійні функції (див. рис. 3.6).

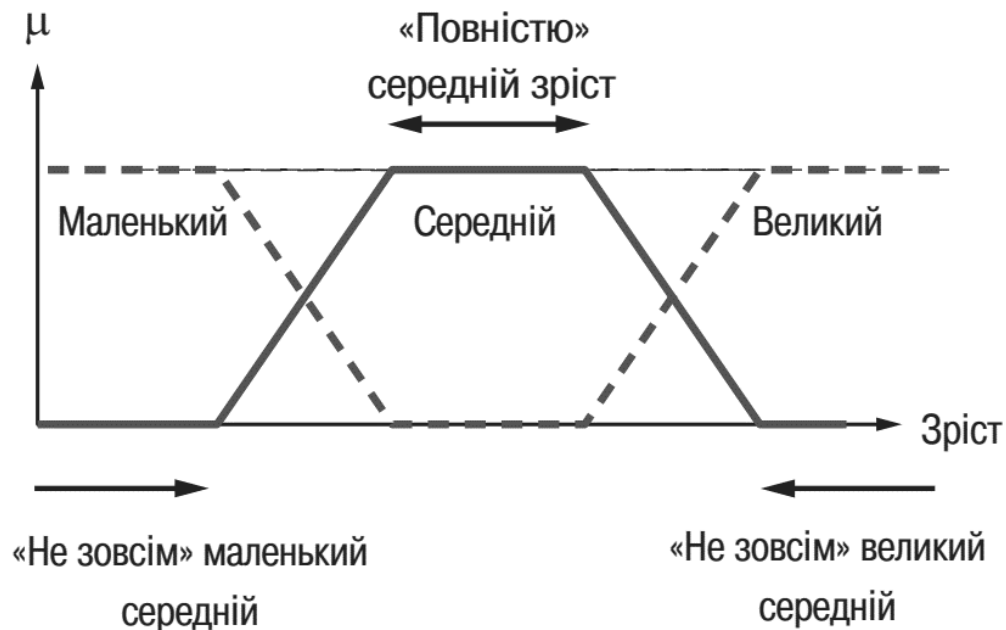


Рисунок 3.6 – Шматково-лінійна функція приналежності

Шматково-лінійні функції приналежності зазвичай використовуються, оскільки, по-перше, характеризуються простотою, та, по-друге, містять точки, що дозволяють встановити область, де поняття є істинним, а де хибним, що спрощує опис системи.

Крім того, існують так звані функції приналежності одноелементної множини. У цьому випадку така функція дорівнює «1» для лише одного значення змінної та дорівнює «0» для усіх інших значень. Як приклад на рисунку 3.7 функція приналежності виражена специфічним значенням змінної «зріст Катерини».

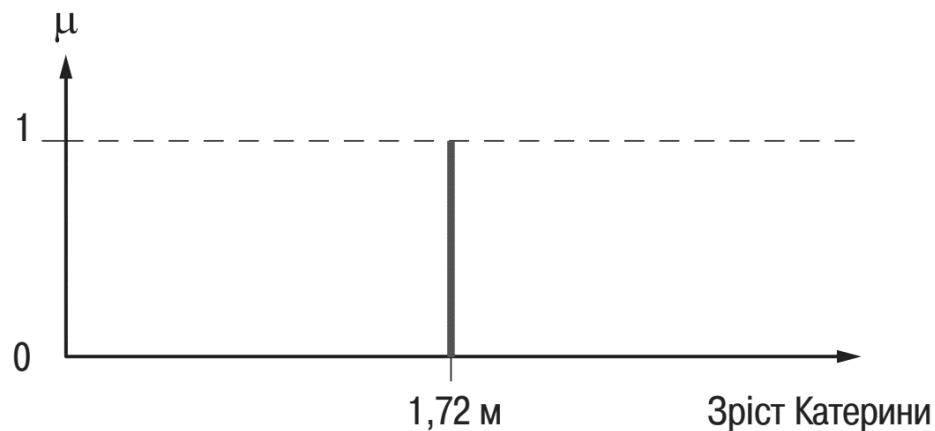


Рисунок 3.7 – Функція приналежності одноелементної множини

3.1.2 Нечітке логічне виведення. Механізм Мамдані

На етапі пошуку відповідної умовної пропозиції та її обробки правила або формуються експертами, або беруться як лінгвістичні значення з бази знань як результат процесу фазифікації.

Тут використовується поняття нечіткого висловлювання. Це пропозиція, щодо якої можна судити про ступінь його істинності чи складності на теперішній момент. Ступінь істинності або ступінь складності кожного нечіткого висловлювання набуває значення із замкнутого інтервалу $[0,1]$, причому 0 і 1 є граничними значеннями ступеня істинності і збігаються з поняттями «хибність» та «істина» для чітких висловлювань.

Істинність нечіткого висловлювання, взагалі є суб'єктивною характеристикою

і залежить від багатьох факторів, зокрема від використання нечіткого висловлювання. У теорії нечітких множин широке поширення отримали нечіткі умовні висловлювання і правила нечіткого логічного висновку. Ця обставина пов'язана з тим, що в семантиці звичайної мови присутня певна кількість нечітких понять і умовних правил, в яких передумови і наслідки включають такі поняття.

Дослідження правил умовного логічного висновку охоплюють в основному наступні три види умовних пропозицій P :

- P_1 : якщо $x \in A$, то $y \in B$;
- P_2 : якщо $x \in A$, то $y \in B$, інакше $y \in C$;
- P_3 : якщо $x_1 \in A_1$ та $x_2 \in A_2$ та ... $x_n \in A_n$, то $y \in B$.

Основою формалізації правил умовного логічного виведення є правило відокремлення, що стверджує: якщо $(\alpha \rightarrow \beta)$ істинно та α істинно, то β істинно.

На практиці формалізація правил для нечітких умовних висновків може бути надзвичайно різноманітною і повинна ґрунтуватися на використанні багатозначних логічних схем.

Підбір відповідної пропозиції виконується за змістом умовної частини нечіткої пропозиції. Тут є два варіанти. Перший полягає в тому, що другим лінгвістичним значенням нехтують і обирають одне умовне правило з бази знань, а потім переходять до його обробки.

Якщо приймається рішення про необхідність урахування обох лінгвістичних значень, то як мінімум у базі знань може виявитися дві відповідні умовні пропозиції. У складніших ситуаціях кількість таких пропозицій може виявитися більшою.

При обліку декількох відповідних правил можуть використовуватися алгоритми Мамдані, Сукамото, Ларсена та інші.

Під час процесу нечіткого логічного виведення найчастіше використовується метод Мамдані. База нечітких правил Мамдані містить лінгвістичні правила, що використовують функції приналежності для опису концепцій системи.

Метод Мамдані складається з декількох етапів:

1. Визначення ступеню істинності для передумов кожного правила для значень x_0 і y_0 : $\mu_{A_1}(x_0)$, $\mu_{B_1}(y_0)$, $\mu_{A_2}(x_0)$, $\mu_{B_2}(y_0)$.

2. Знаходження рівнів відсікання для передумов кожного з правил із використанням операції знаходження мінімуму:

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0); \alpha_2 = A_2(x_0) \wedge B_2(y_0).$$

Та подальше знаходження усічених функцій приналежності:

$$c'_1 = \alpha_1 \wedge c_1(z); c'_2 = \alpha_2 \wedge c_2(z).$$

3. Об'єднання усічених функції з використанням операції знаходження максимуму (див. рис. 3.8), що дає підсумкову нечітку підмножину для змінної виходу з функцією приналежності:

$$\mu_{\Sigma} = \max\{\min[\mu_{A_1}(x_0), \mu_{B_1}(y_0)], \min[\mu_{A_2}(x_0), \mu_{B_2}(y_0)]\}.$$

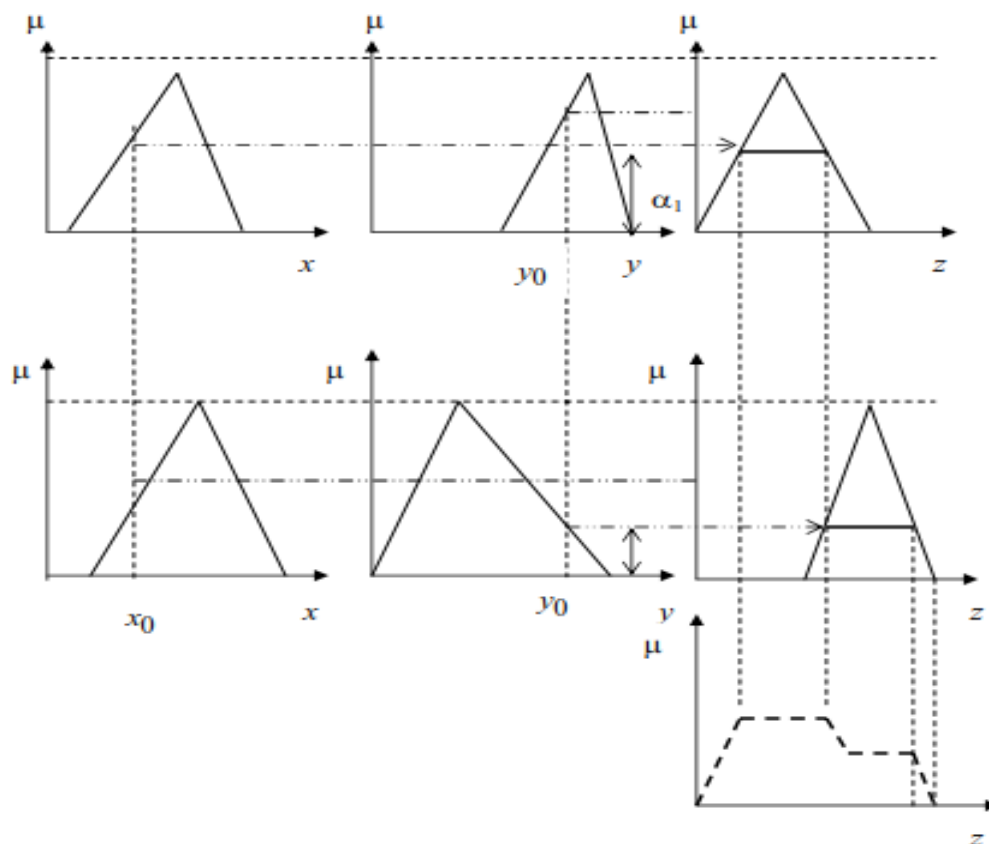


Рисунок 3.7 – Об'єднання функцій

3.1.3 Дефазифікація

На останньому етапі нечіткого логічного виведення вихідна нечітка множина є вже визначеною, але ще не може бути безпосередньо використана для надання точної інформації користувачу. Через це необхідно добігти до дефазифікації – переведення нечітких значень до єдиного чіткого, за ступенем приналежності значень, як і в процесі фазифікації [26].

При цьому можна використовувати різноманітні методи, проте найчастіше використовується метод обчислення центру області, який бере за центр ваги нечіткої множини.

На цьому етапі на основі підсумкової нечіткої множини знаходиться можливе рішення, що відповідає нечітким пропозиціям, що використовуються. Це рішення визначається за підсумковою функцією, отриманою після обробки правил умовного логічного висновку.

Одним із способів є вибір такого значення, коли функція приналежності має найбільше значення (див. рис. 3.8, а). За наявності кількох екстремумів на функції приналежності (див. рис. 3.8, б) обирається середнє для цих точок. В останньому випадку має місце так званий метод середнього по максимуму.

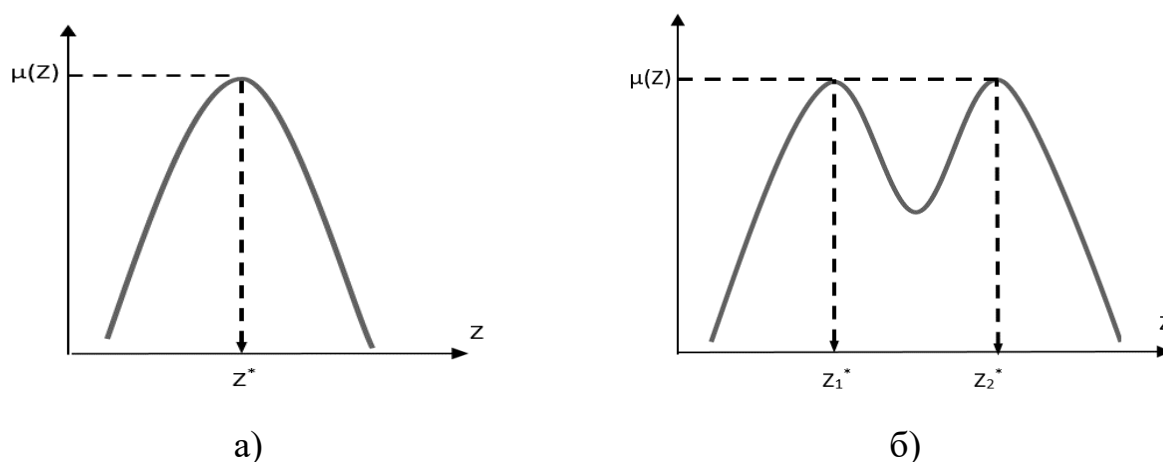


Рисунок 3.8 – Функції приналежності: а) з найбільшим значенням; б) з кількома значеннями

Широке застосування під час вибору рішення отримав метод центру тяжкості, коли множина значень результуючої функції власності розглядається, як множина матеріальних точок, маси яких рівні значенням функції власності з відповідними координатами:

$$z^* = \frac{\sum \mu(z_i) z_i}{\sum \mu(z_i)}$$

Застосування цього найбільш доцільно, коли немає достатніх підстав застосовувати метод максимуму або метод центру площ.

При виборі рішення слід контролювати рівень функцій приналежності, тобто необхідно встановити деякий поріг рішення, нижче якого приймати рішення не слід: на рис. 3.9, а – вибір рішення можна виконати, на рис. 3.9, б – вибір рішення проводити не треба.

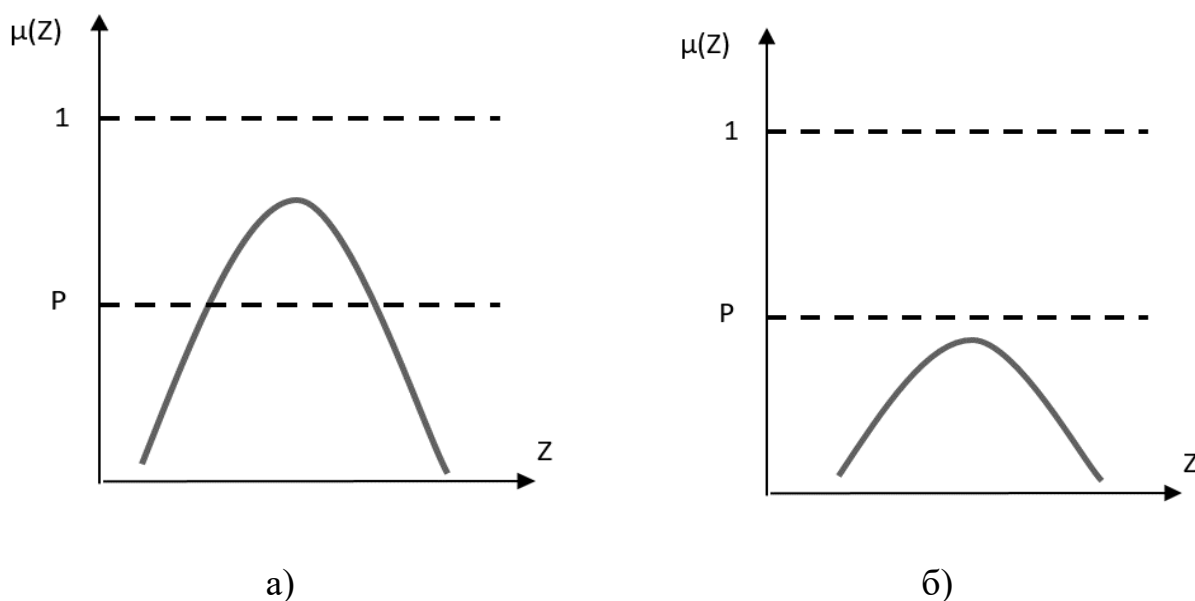


Рисунок 3.9 – Рівень функцій приналежності: а) з необхідністю вибору рішення; б) без необхідності вибору

3.2 Застосування інструментів нечіткої логіки до обраної предметної області

Як вже було зазначено у попередніх пунктах, класична логіка визначає класичні множини, де заданий об'єкт належить або не належить множині.

Нечіткий підхід розширює класичні множини, дозволяючи функції $f(x)$ повертати значення x в діапазоні $[0,1]$. Дане значення може належати множині в деякій певній мірі. Класична поведінка все ще може бути досягнута шляхом повернення 0 або 1, але функція належності може повертати, наприклад, 0.8, що означає, що існує ймовірність 80%, що значення належить до множини.

3.2.1 Визначення лінгвістичних змінних

Під час роботи із нечіткою логікою важливо приділити достатню увагу використанню лінгвістичних змінних – даних нечітких систем, що зазвичай зберігають замість числових лінгвістичні.

Наприклад, лінгвістична змінна «Вік (Age)» може зберігати значення «Похилий», яке має сенс, а не числове значення, як зазвичай, або ж «Тип шкіри (SkinType)» із значенням «Суха (Dry)» або «Жирна (Oily)». При цьому використання нечіткої логіки виходить дуже зручним, бо виміряти жирність або сухість шкіри та отримати якесь числове значення неймовірно важко.

Лінгвістична змінна розглядається також і як набір нечітких множин. Кожна нечітка множина є значенням, яке може бути присвоєне змінній, а всередині змінної нечіткі множини зазвичай називаються нечіткими мітками [27].

На рисунку нижче показано лінгвістичну змінну «Вік (Age)» та її лінгвістичні мітки (див. рис. 3.10).

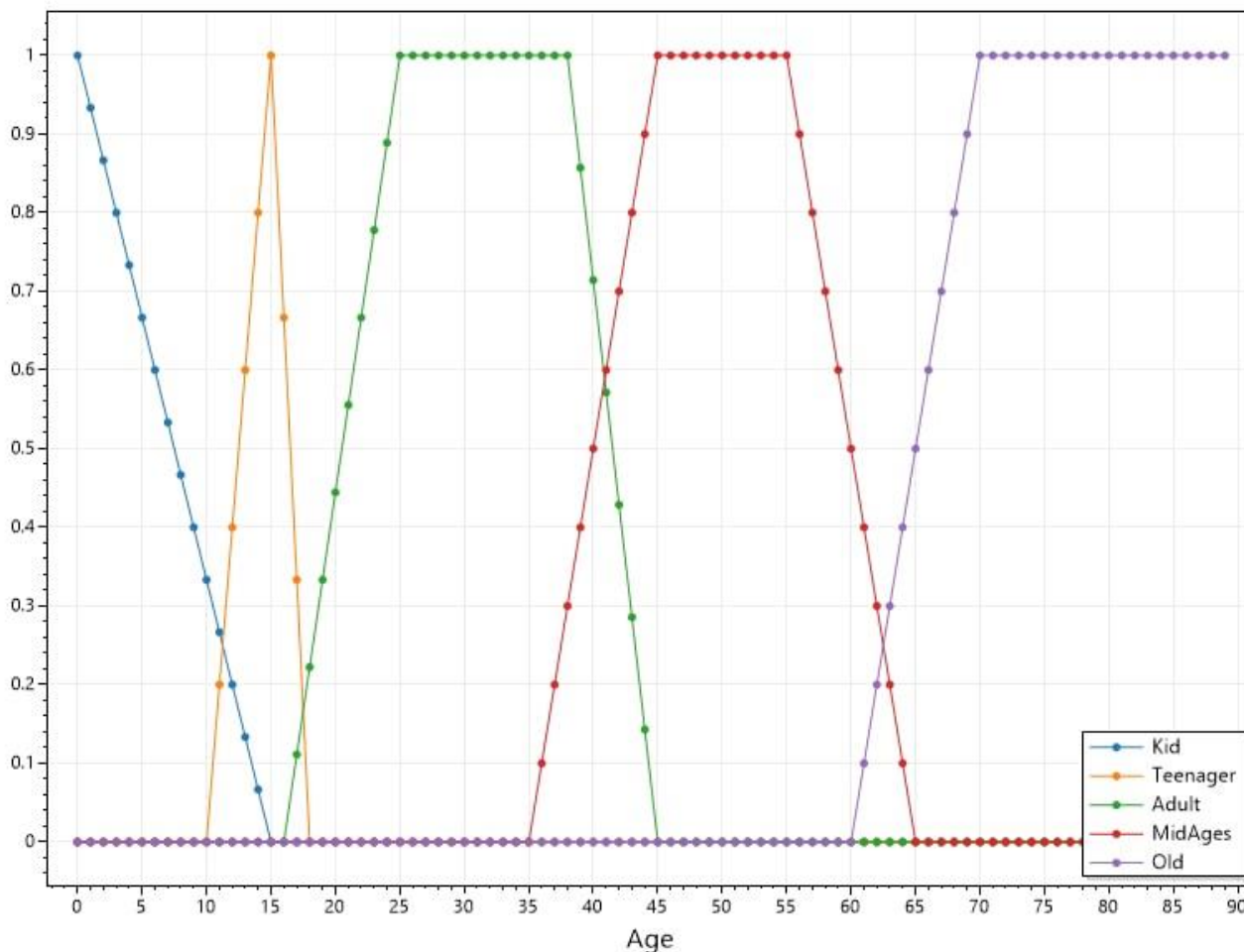


Рисунок 3.10 – Лінгвістична змінна «Age» та її лінгвістичні мітки

Тобто лінгвістична змінна «Age» має у цьому випадку 5 лінгвістичних міток: «дитина» (Kid), «підліток» (Teenager), «дорослий» (Adult), «людина середніх років» (MidAges) та «людина похилого віку» (Old).

У цьому випадку, якщо людині менше 15 років, її ще можна назвати дитиною. У якому віці дитина стає підлітком виявити однозначно складно, тому поняття дитинства та підліткового періоду перетинаються. Також на рисунку можна побачити, що, наприклад, користувач від 35 до 45 років може зватися як дорослим, так і людиною середніх років. За віднесенні користувачів до тих чи інших вікових категорій відповідальні саме функції приналежності.

Тобто на даному прикладі можна показати, що якщо x має числове значення

62, ми можемо сказати, що користувач вже похилого віку із приналежністю 0,3 і середніх років із приналежністю 0,25.

До предметної області «Інформаційна система «Інтернет магазин косметичних засобів» можна віднести безліч лінгвістичних змінних, тож наведемо деякі з них.

Для того, аби користувачу рекомендувати правильний догляд за волоссям, потрібно врахувати різні аспекти, що впливають на його стан, як, наприклад частоту миття волосся. Лінгвістична змінна у цьому випадку названо FrequencyWashingHair (див. рис. 3.11).

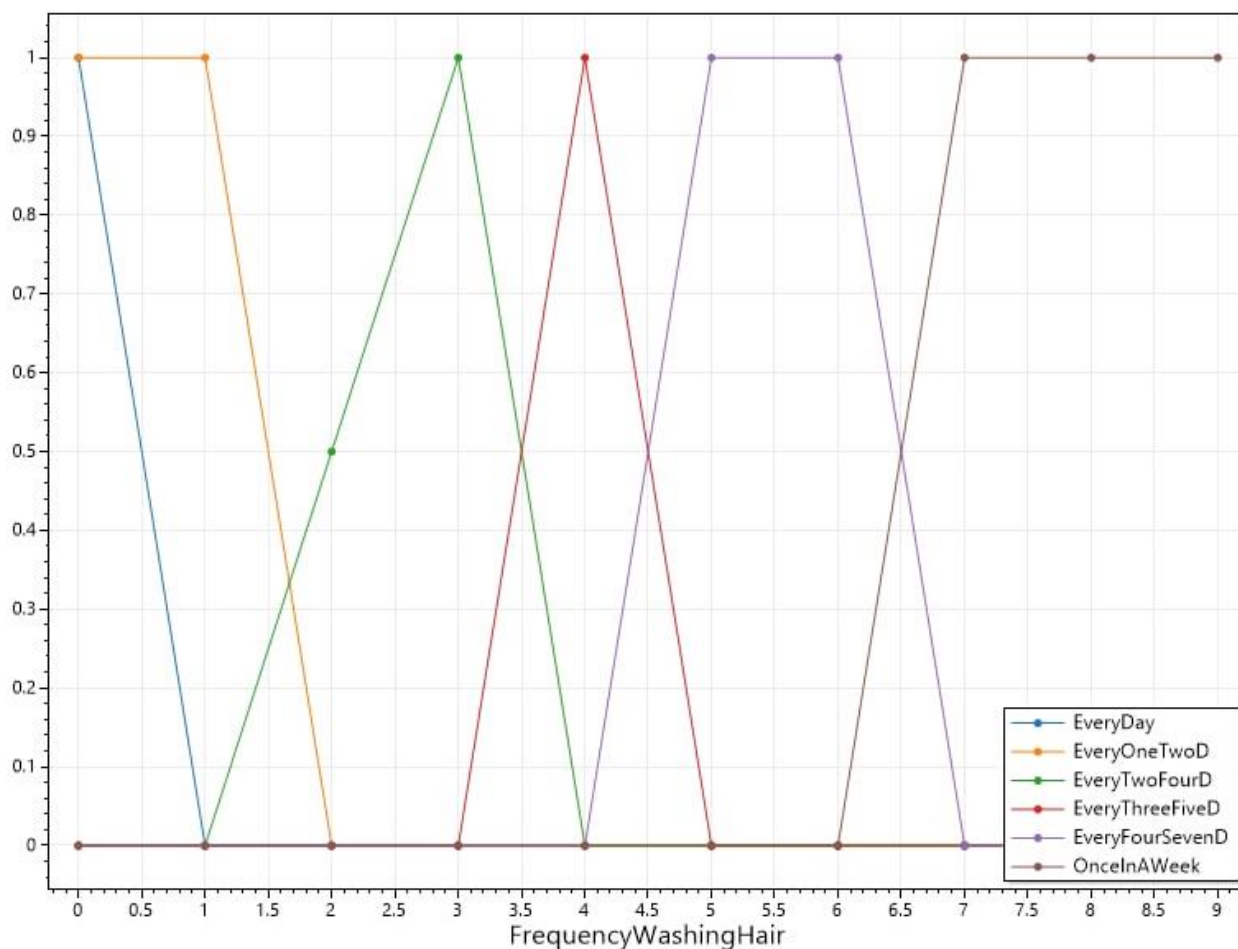


Рисунок 3.11 – Лінгвістична змінна «FrequencyWashingHair»

Ця змінна має 6 міток, що позначають кожний окремий графік миття волосся:

дехто робить це раз на тиждень, а хтось щодня, і це забирає купу часу.

Враховуючи таку інформацію про клієнта, можна рекомендувати сухий шампунь, використання якого зекономить кучу часу, або лінійку спеціального догляду, що знижуватиме жирність шкіри голови.

Обидва приклади, що розглядаються, представлені шматково-лінійними функціями приналежності.

Крім того, у інформаційних системах, в тому числі і у цій, використовуються так звані функції приналежності одноелементної або декілька елементної множини [28].

Якщо клієнта Інтернет-магазину цікавить догляд за шкірою обличчя, доцільно дізнатися, чи є в нього висипання. Відповіддю на це питання є або ні, або так. Тобто функція дорівнює «1» для лише одного значення змінної «Так» або для лише одної «Ні» (див. рис. 3.12).

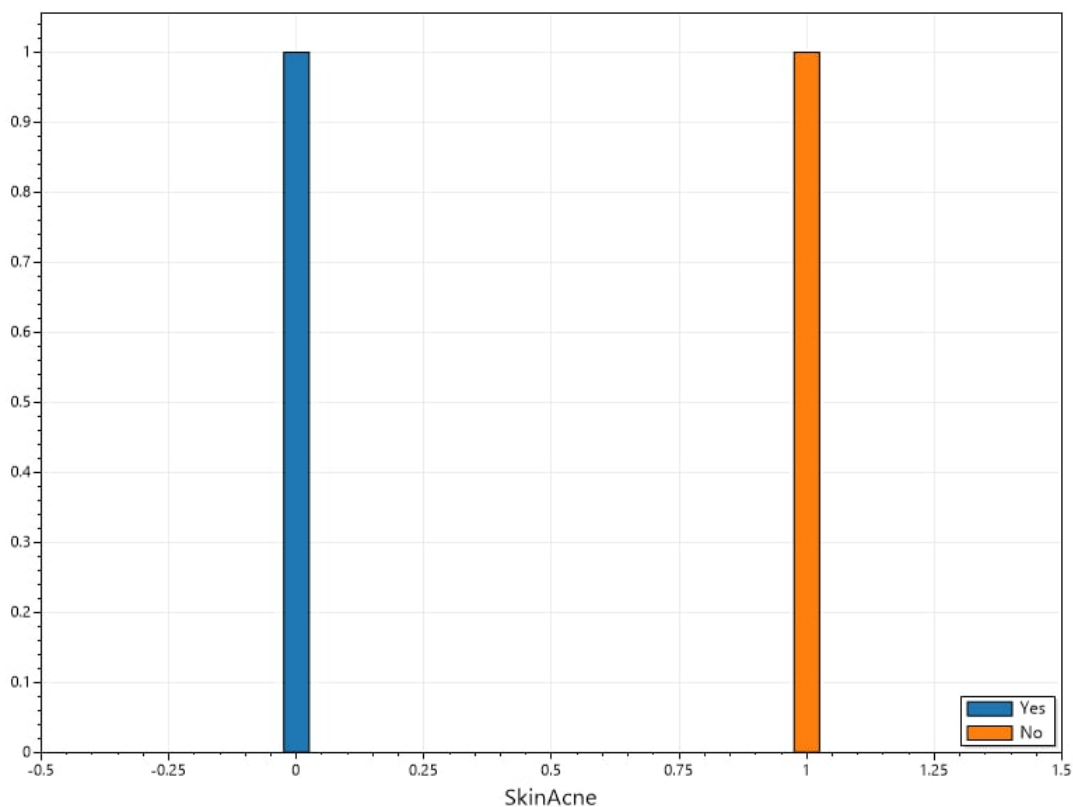


Рисунок 3.12 – Лінгвістична змінна «SkinAcne» із мітками «Yes» і «No»

Аналогічним чином можна врахувати і статі людини, лінгвістичні змінні яких створюються для запобігання надання рекомендацій жінці чоловічих товарів та навпаки (див. рис. 3.13).

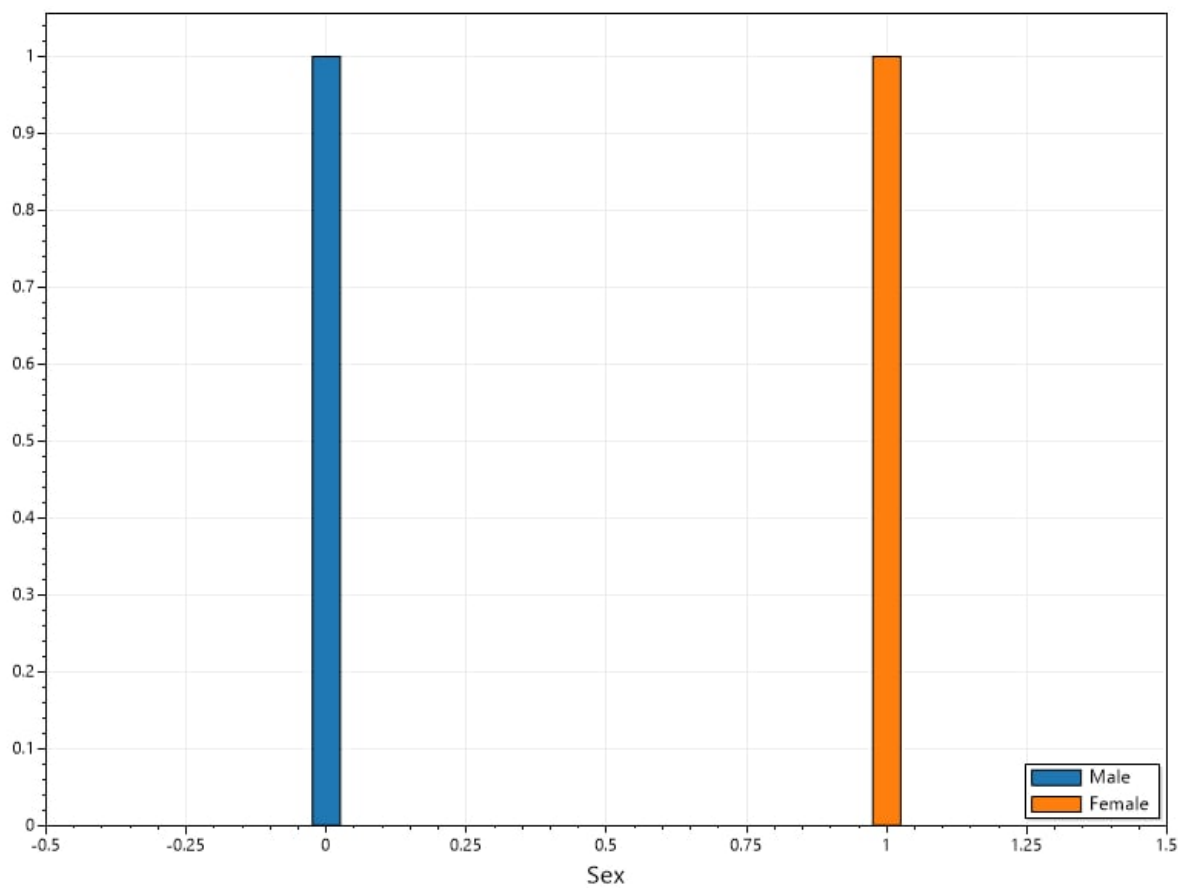


Рисунок 3.13 – Лінгвістична змінна «Sex» із мітками «Male» і «Female»

3.2.2 Оголошення нечітких правил

Набір лінгвістичних змінних для представлення значення даних, якими маніпулюють у нечітких системах, утворює нечітку базу даних.

Для виконання лінгвістичних обчислень з нечіткою базою даних потрібно оголосити нечіткі правила. Вони дуже схожі на спілкування природною мовою, як деякі інструкції від людини до людини: «IF (ЯКЩО) тип шкіри IS (Є) сухий, THEN

(ТОДІ) товар IS (Є) зволожувальний крем";

У загальному вигляді нечіткі правила мають так звані антецедент і консеквент, відокремлені твердженням «THEN» (ТОДІ). Антецедент являє собою з'єднання декількох нечітких речень (мітка змінної IS) з відповідними операторами між ними. Консеквент являє собою дію, яку система повинна виконати, якщо антецедент істинний (при будь-якому ступені приналежності).

Обчислення належності антецедента виконується за допомогою нечітких операторів I, АБО і НЕ (T-Norm і T-Conorm). Належність антецедента зазвичай називають вогнестійкістю правила до заданого вхідного значення. Нечіткі системи часто мають набір нечітких правил, які представляють поведінку системи, відомий як база нечітких правил.

3.2.3 Оператори нечіткої логіки

Під час обчислення належності антецедента використовуються так звані оператори нечіткої логіки.

Належність антецедента зазвичай називають стійкістю правила до заданого вхідного значення. Нечіткі системи часто мають набір нечітких правил, які представляють поведінку системи, відому як база нечітких правил.

Оператори нечіткої логіки дозволяють записувати комбінації логічних понять з метою знаходження ступеню істинності. Нечіткі оператори визначені для виконання операцій перетину, об'єднання та доповнення. Тобто існує три стандартні оператори нечіткої логіки, а саме: AND, OR, NOT. Вони відіграють важливу роль у на не менш важливому етапі залучення апарату нечіткої логіки – формуванні нечітких правил експертами.

За перетин множин відповідає логічний оператор AND (I). Наприклад, якщо розглядаються дві множини X та Y, ступенем істинності припущення виступає мінімум з ступенів істинності цих X та Y.

Розглянемо вплив оператора AND у наступному нечіткому правилі: «IF SkinType IS Дуже Суха AND Age IS Похилий THEN Product IS Антивіковий крем від Vichy для шкіри із сухістю».

Якщо користувач із сухою шкірою шукає підходящий крем, але його вік умовно менший за 65 років, товар «Антивіковий крем від Vichy для шкіри із сухістю» даному покупцю не буде запропоновано, хоча й наполовину задовольняє запит. Тобто при використанні AND результат аналогічний традиційній логіці: з 0 та 1 виходить 0.

OR (АБО) є оператором об'єднання. У цьому випадку ступенем істинності буде максимум зі ступенів істинності припущень X та Y. Такий оператор доцільно використати, коли користувачеві можна рекомендувати більше одного продукту: «IF SkinType IS Дуже Суха THEN Product IS Крем від Garnier для сухої шкіри OR Зволожуючий крем Bioderma» чи створювати умови, коли товар задовільнить декілька категорій покупців одразу: «IF Age IS Середніх років AND Age IS Похилий THEN Product IS Антивіковий крем з ефектом ліфтингу від Lancome».

Логічний оператор заперечення NOT (НІ) дозволяє виключити ситуації, коли рекомендації будуть містити позиції, що не підходять за персональними особливостями: «IF Age IS NOT Підліток THEN Product IS Крем для корекції зморшок та відновлення пружності від Cerave».

Таким чином формуються правила нечіткої логіки і з них утворюється нечітка база правил.

3.2.4 Алгоритм функціонування інструментарію нечіткої логіки у обраній предметній області

Принцип роботи системи нечіткого виведення можна просто описати у вигляді наступного алгоритму (див. рис. 3.14).

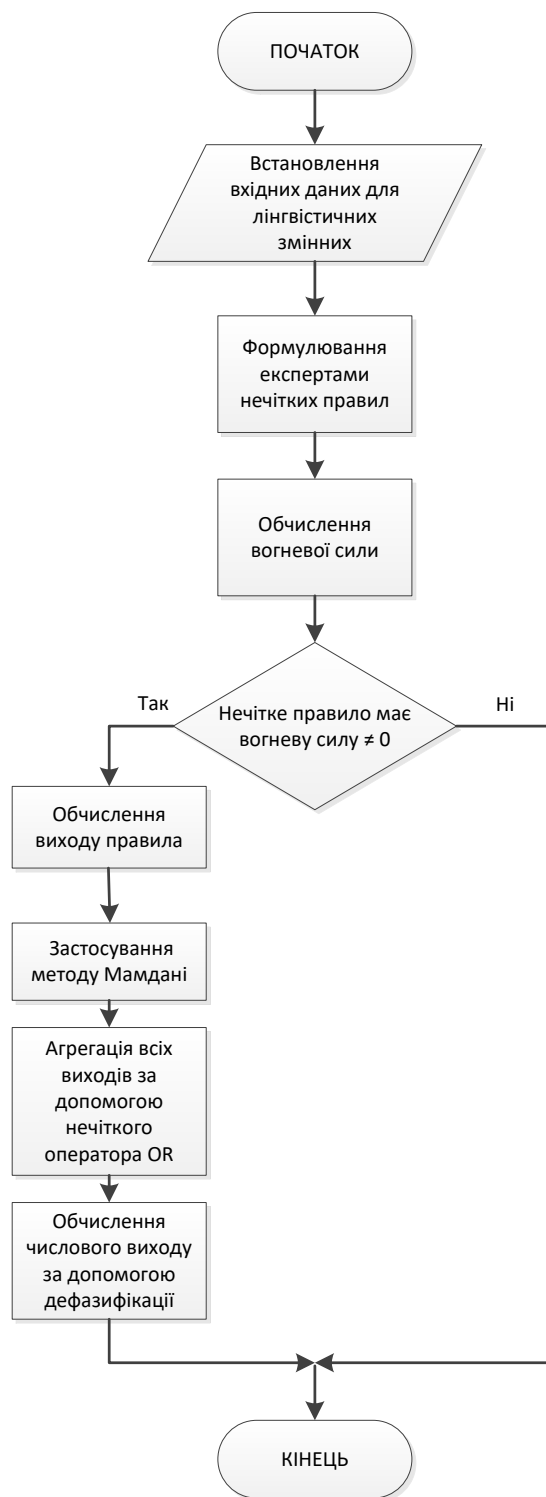


Рисунок 3.14 – Алгоритм застосування нечіткої логіки у обраній ІС

Спочатку необхідно встановити вхідні дані у вигляді числових значень для усіх лінгвістичних змінних. Такі значення будуть представлені всередині системи як

значення приналежності до кожної нечіткої множини кожної лінгвістичної змінної, після чого експерти мають сформулювати нечіткі правила, що містять ці лінгвістичні змінні.

Як вже було зазначено раніше, функції приналежності або вогнева сила грають важливу роль для отримання нечіткого результату, тому обчислення вогневої сили кожного правила є обов'язковим наступним завданням. Це досягається шляхом обчислення приналежності кожного нечіткого твердження та об'єднання цих приналежностей за допомогою нечітких операторів.

Далі для кожного нечіткого правила з ненульовою вогневою силою обчислюється так званий вихід правила. Існує декілька методів, але метод Мамдані є найбільш підходящим у цьому випадку, бо він може допомогти відобразити ступінь приналежності антецедента правила до консеквенту правила.

Всі виходи (нечіткі множини) агрегуються за допомогою нечіткого оператора АБО. Нарешті, числовий вихід обчислюється за допомогою методу дефазифікації, коли числове значення обчислюється з нечіткого виводу.

3.3 Побудова математичної моделі

У результаті експертної оцінки було визначено наступне:

- лінгвістичні змінні;
- відповідні функції приналежності;
- терми-множини лінгвістичних змінних;
- відповідні лінгвістичні мітки;
- множини значень таких міток.

Вищезазначене наочно представлено у таблиці 3.1, де ШЛФП – шматково-лінійна функція приналежності, а ФПОМ – функція приналежності одноелементної множини.

Таблиця 3.1 – Лінгвістичні змінні та їх можливі значення

Лінгвістична змінна	Опис	Універсальна множина	Терм-множина		Тип функції приналежності
			Лінгвістична мітка	Область визначення	
Age	Вік клієнта	[0;100]	Kid	[0;14]	ШЛФП (трапецієвидна)
			Teenager	[10;19]	
			Adult	[17;45]	
			MidAges	[35;65]	
			Old	[60;100]	
EyeBrow-Color	Колір брів	[0,1]	Light	[0]	ФПОМ
			Dark	[1]	
Frequency-WashingHair	Частота миття волосся	[0;10]	EveryDay	[0;1]	ШЛФП (трапецієвидна)
			EveryOneTwoD	[0;2]	
			EveryTwoFourD	[1;4]	
			EveryThreeFiveD	[3;5]	
			EveryFourSevenD	[4;7]	
			OnceInAWeek	[6;10]	
HairType	Тип волосся	[0,1]	Curly	[0]	ФПОМ
			Straight	[1]	
Product-Category	Категорія товарів	[1,2,3]	Skin	[1]	ФПОМ
			Hair	[2]	
			MakeUp	[3]	
Sex	Стать клієнта	[0,1]	Male	[0]	ФПОМ
			Female	[1]	
SkinAcne	Схильність до висипань	[0,1]	No	[0]	ФПОМ
			Yes	[1]	

Продовження табл. 3.1

Лінгвістична змінна	Опис	Універсальна множина	Терм-множина		Тип функції приналежності
			Лінгвістична мітка	Область визначення	
SkinType	Тип шкіри	[0;10]	VeryDry	[0;3]	ШЛФП (трапецієвидна)
			Dry	[1;5]	
			Combined	[4;8]	
			Oily	[7;10]	

Під час розробки моделі застосовується апарат нечіткого логічного виведення Мамдані, згідно до якого максимум буде змодельовано, застосовуючи агрегацію, а мінімум – імплікацію.

Дослідження предметної області, що розглядається, також дозволило побудувати базу нечітких правил, частину вмісту якої наведено у таблиці 3.2.

Таблиця 3.2 – Правила нечітких продукцій системи

ID	Нечітке правило
Rule1	IF ProductCategory IS Skin AND Age IS Old AND Age IS NOT Kid AND Age IS NOT Teenager AND Sex IS Female THEN Product IS VichyLiftactiv
Rule2	IF ProductCategory IS Hair AND FrequencyWashingHair IS EveryoneTwoD AND Age IS NOT Kid THEN Product IS KerastaseFreshAffairDryShampoo
Rule4	IF ProductCategory IS Skin AND SkinType IS VeryDry AND SkinType IS NOT Oily AND SkinType IS NOT Combined AND Age IS NOT Kid AND Age IS NOT Teenager THEN Product IS NutriticIntenseRiche
Rule5	IF ProductCategory IS Skin AND SkinType IS Oily AND SkinAcne IS Yes THEN Product IS BiotradeAcneOutMattifyingTonic
Rule6	IF ProductCategory IS Hair AND Age IS Kid THEN Product IS ShampooJohnsonsBaby

Продовження табл. 3.1

ID	Нечітке правило
Rule7	IF ProductCategory IS Hair AND HairType IS Curly AND Age IS NOT Kid THEN Product IS HairBotox
Rule8	IF ProductCategory IS MakeUp AND EyeBrowColor IS Light THEN Product IS LightCoutureBrowYSL
Rule9	IF ProductCategory IS Skin AND SkinAcne IS Yes AND Age IS NOT Kid THEN Product IS CtrlATeatreementToner
Rule10	IF ProductCategory IS MakeUp AND Age IS Teenager THEN Product IS MaybellineNewYorkLashesSensational

Розглянемо роботу визначеної моделі на прикладі ситуації, коли жінка 68 років із сухою шкірою обличчям та без проблем висипань проглядає товари категорії «Шкіра».

У даному випадку є наступні вхідні дані:

- лінгвістична змінна «ProductCategory» із термом «Skin» та значенням 1;
- лінгвістична змінна «Age» із термом «Old» та значенням 68;
- лінгвістична змінна «Sex» із термом «Female» та значенням 1;
- лінгвістична змінна «SkinType» із термом «Dry» та значенням 1;
- лінгвістична змінна «SkinAcne» із термом «No» та значенням 0.

Під час перебору нечітких правил у базі правил, система, наприклад, знаходить наступні три:

1) «IF ProductCategory IS Skin AND SkinType IS VeryDry AND SkinType IS NOT Oily AND SkinType IS NOT Combined AND Age IS NOT Kid AND Age IS NOT Teenager THEN Product IS NutriticIntenseRiche»;

2) «IF ProductCategory IS Skin AND SkinType IS VeryDry OR SkinType IS Dry AND Age IS NOT Kid THEN Product IS LOrealParisFaceCream»;

3) «IF ProductCategory IS Skin AND SkinType IS Dry THEN Product IS HolikaAloeSoothingGel».

Результатом виведення (рекомендаціями) стануть перші два товари: «NutriticIntenseRiche» та «LOrealParisFaceCream». Тут функції приналежності дорівнюватимуть 1, тому що усі складові правил відповідають усім персональним особливостям клієнта.

У випадку третього правила, такий товар також може бути рекомендовано, але у випадку, коли потрібно відобразити більше двох товарів. Відповідна функція приналежності має значення 0.5, що менше за одиницю, тобто гірше задовольняє умови.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ РЕКОМЕНДАЦІЙНОГО ІНТЕРФЕЙСУ

4.1 Розробка бази даних на обраній платформі СУБД

Одним із важливих кроків реалізації системи видачі рекомендацій у інформаційній системі є створення бази даних, що спочатку необхідно спроектувати, тобто треба визначити сутності та атрибути, що використовуватимуться у предметній області «Інтернет-магазин косметичних засобів».

Задля цього можна використати CASE-засіб All Fusion Data Modeler, що дозволить розробити логічну, і я наслідок, фізичну модель майбутньої бази даних.

Логічна модель допомагає описати дані так, що згодом СУБД зможе їх автоматизовано оброблювати, а фізична модель описує як будуть представлені ці дані та має усі деталі, що необхідні для створення самої бази даних у обраній СУБД.

На рисунку 4.1 представлена логічна модель даних, що містить три сутності: «Product», «ProductCategory», «FuzzyRule».

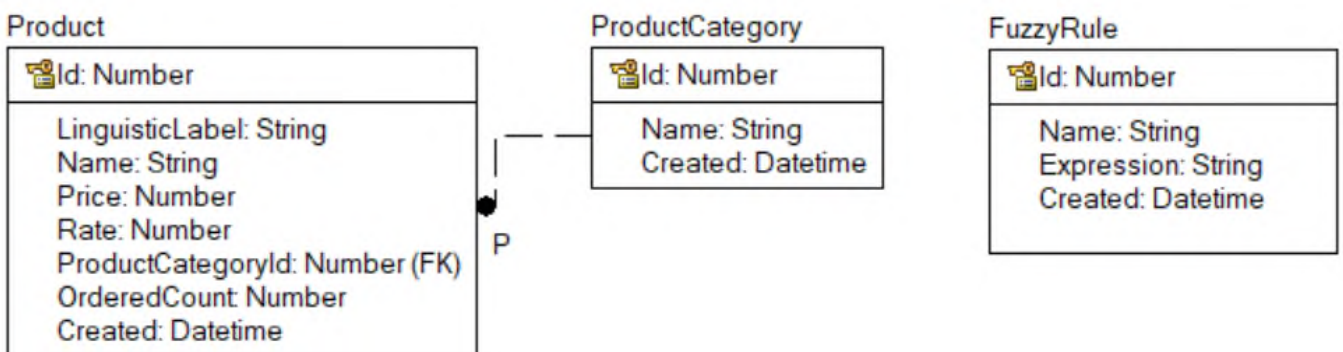


Рисунок 4.1 – Логічна модель даних

Сутність «Product» має атрибути, що описують характеристики усієї продукції, якою торгують у магазині косметичних засобів, а саме найменування, ціну, рейтинг, продану кількість, дату занесення до бази даних і відповідну лінгвістичну змінну, що

є унікальною. Завдяки зовнішньому ключу з ідентифікатором категорії ця сутність пов'язана із сутністю «ProductCategory».

Сутність «ProductCategory», у свою чергу, слугує довідником про усі існуючі у магазині категорії товарів та послуг.

Задля зберігання усіх нечітких правил створено сутність «FuzzyRule» із атрибутами унікальної назви самого правила, нечіткого висловлення та дати додавання правила до БД.

Завдяки обраному програмному засобу можна згенерувати фізичну модель даних (див. рис. 4.2), засновуючись на створеній логічній моделі.

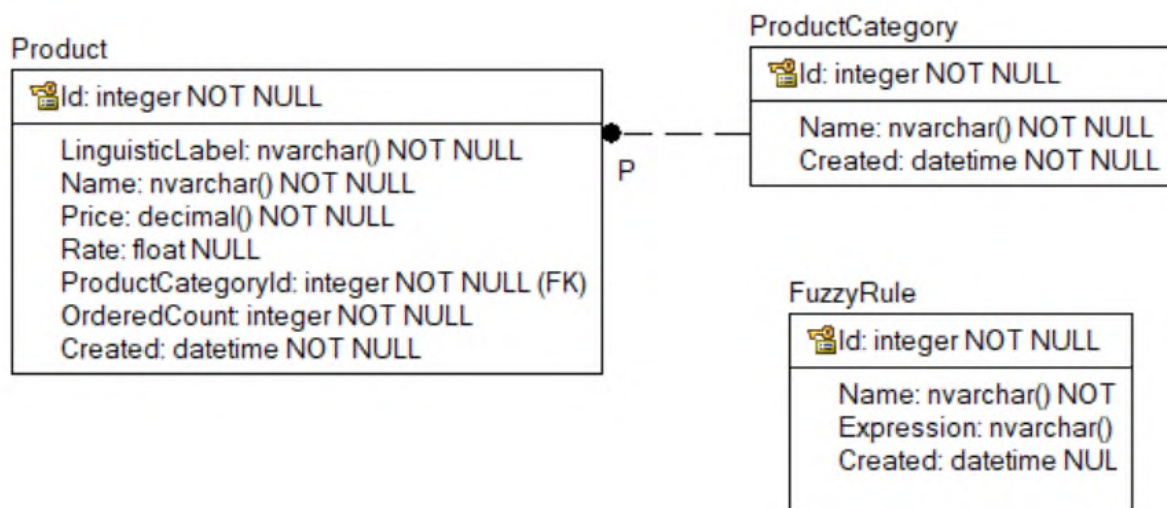


Рисунок 4.2 – Фізична модель даних

CASE-засіб All Fusion Data Modeler також надає можливість автоматичного генерування SQL-коду створення таблиць майбутньої бази даних.

Отже, ґрунтуючись вже на розробленій фізичній моделі даних, у СУБД MS SQL Server була створена та згодом заповнена база даних «FuzzyCosmetic» (див. рис. 4.3).

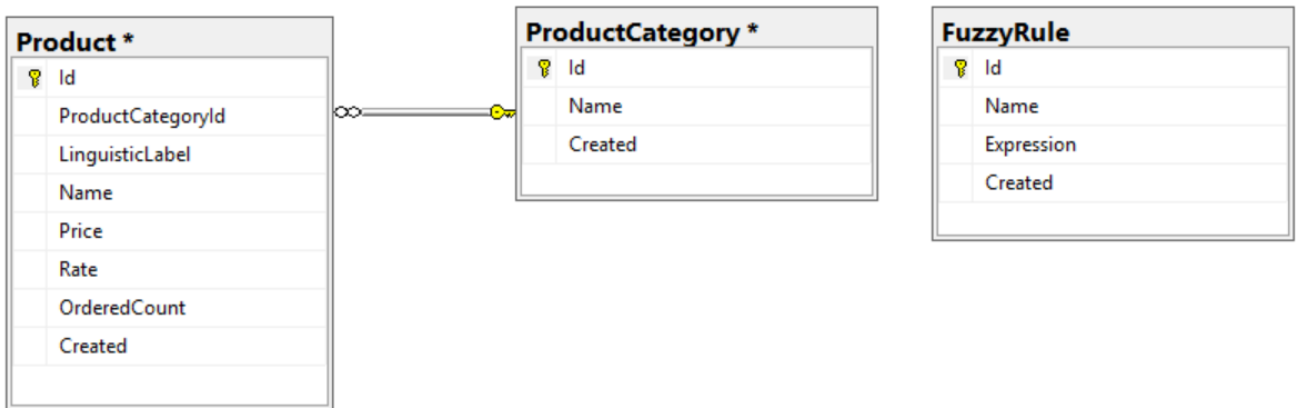


Рисунок 4.3 – Схема реалізованої бази даних «FuzzyCosmetic»

Частина написаних експертами нечітких правил, що внесені до таблиці «FuzzyRule», наведена на рисунку 4.4.

	Id	Name	Expression	Created
1	1	Rule1	IF ProductCategory IS Skin AND Age IS Old AND Age IS NOT Kid AND Age IS NOT Teenager AND Sex ...	2022-12-05 22:06:10.520
2	2	Rule2	IF ProductCategory IS Skin AND Age IS MidAges AND Age IS NOT Kid AND Age IS NOT Teenager AN...	2022-12-05 22:06:10.520
3	3	Rule3	IF ProductCategory IS Skin AND Age IS MidAges AND Age IS NOT Kid AND Age IS NOT Teenager AN...	2022-12-05 22:06:10.520
4	4	Rule4	IF ProductCategory IS Skin AND Age IS Old OR Age IS MidAges AND Age IS NOT Kid AND Age IS NOT...	2022-12-05 22:06:10.520
5	5	Rule5	IF ProductCategory IS Skin AND SkinType IS VeryDry AND SkinType IS NOT Oily AND SkinType IS NO...	2022-12-05 22:06:10.520
6	6	Rule6	IF ProductCategory IS Skin AND Age IS Old AND Sex IS Female THEN Product IS LorealParisSkinExpert	2022-12-05 22:06:10.520
7	7	Rule7	IF ProductCategory IS Skin AND SkinType IS VeryDry OR SkinType IS Dry AND Age IS NOT Kid THEN ...	2022-12-05 22:06:10.520
8	8	Rule8	IF ProductCategory IS Skin AND SkinAcne IS Yes AND Age IS NOT Kid THEN Product IS CtrlATeatreem...	2022-12-05 22:06:10.520
9	9	Rule9	IF ProductCategory IS Skin AND SkinAcne IS Yes AND Age IS NOT Kid THEN Product IS AscorbylGlucos...	2022-12-05 22:06:10.520
10	10	Rule10	IF ProductCategory IS Skin AND SkinAcne IS Yes AND Age IS NOT Kid THEN Product IS GlowTonicExf...	2022-12-05 22:06:10.520
11	11	Rule11	IF ProductCategory IS Skin AND Age IS Old OR Age IS MidAges AND Age IS NOT Kid AND Sex IS Fem...	2022-12-05 22:06:10.520
12	12	Rule12	IF ProductCategory IS MakeUp AND EyeBrowColor IS Light AND Age IS NOT Kid THEN Product IS Rim...	2022-12-05 22:06:10.520
13	13	Rule13	IF ProductCategory IS MakeUp AND EyeBrowColor IS Dark AND Age IS NOT Kid THEN Product IS Rim...	2022-12-05 22:06:10.520
14	14	Rule14	IF ProductCategory IS MakeUp AND EyeBrowColor IS Dark AND Age IS NOT Kid THEN Product IS Rim...	2022-12-05 22:06:10.520
15	15	Rule15	IF ProductCategory IS MakeUp AND EyeBrowColor IS Light AND Age IS NOT Kid THEN Product IS Eye...	2022-12-05 22:06:10.520

Рисунок 4.4 – Нечіткі правила з таблиці «FuzzyRule»

4.2 Розробка коду програми з використанням обраної мови програмування

Коли база даних вже створена, можна переходити до програмної реалізації надання рекомендацій у інформаційній системі «Інтернет-магазин косметичних засобів».

Застосування інструментів нечіткої логіки має свої особливості, як необхідність розробки набору класів, що містять відповідні функціонал та методи, які врешті зможуть виконувати всі задачі типової системи нечіткого виведення.

Враховуючи раніше встановлений алгоритм дій щодо реалізації API надання рекомендацій товарів та послуг доцільно розробити наступні класи:

- клас системи виведення («InferenceSys»), що представляє повну систему нечіткого виводу та містить базу даних, базу правил, власне, нечітке виведення і метод дефазифікації;
- клас бази даних («Database») зі списком усіх лінгвістичних змінних системи;
- клас лінгвістичних змінних («LingVar»), що використовуються в нечітких системах та можуть зберігати деяку нечітку множину;
- клас нечіткої множини («FuzzySet»), що є основою усієї нечіткої теорії та містить множину, члени якої мають ступінь приналежності від 0 до 1;
- інтерфейс функцій приналежності («IMemberFunc») для відповідних можливих функцій нечітких множин;
- клас шматково-лінійної функції («PiecewisLinFunc») із функціями приналежності, що утворені з декількох лінійних функцій;
- клас трапецієподібної функції («TrapezoFunction») із функціями приналежності з формою трапеції;
- клас бази правил («RuleDatabase») зі списком усіх лінгвістичних правил системи;
- клас правила («Rule»), що являє собою нечітке правило та містить нечіткі висловлення та оператори для їх об'єднання;
- клас висловлення («Clause») із нечіткими речення типу «A IS B», де «A» – це лінгвістична змінна, а «B» – лінгвістичне значення, до якого віднесена змінна;
- інтерфейс для нечітких норм («INormal») – методів, що використовуються для виконання операцій «I» між нечіткими множинами;

- інтерфейс для нечітких конорм («ICoNormal») – методів, що використовуються для виконання операцій «АБО» між нечіткими множинами;
- клас обчислення конорм за максимальним з двох значень («MaxCoNormal»);
- клас обчислення норм за мінімумом з двох значень («MinNormal»);
- клас обчислення норм множенням двох значень («ProdNormal»);
- клас нечіткого виведення системи («FuzzyOut»), який може бути використаний у лінгвістичному вигляді або дефазифікованим;
- інтерфейс для методів дефазифікації («IDefuzzy»), що використовуються для отримання числового значення з нечіткого виведення;
- клас дефазифікатора, який обчислює центроїд нечіткого виводу («CentroDefuzzy»).

Схематично взаємозв'язок та взаємодію визначених класів можна представити таким чином, як подано на рисунку 4.5.

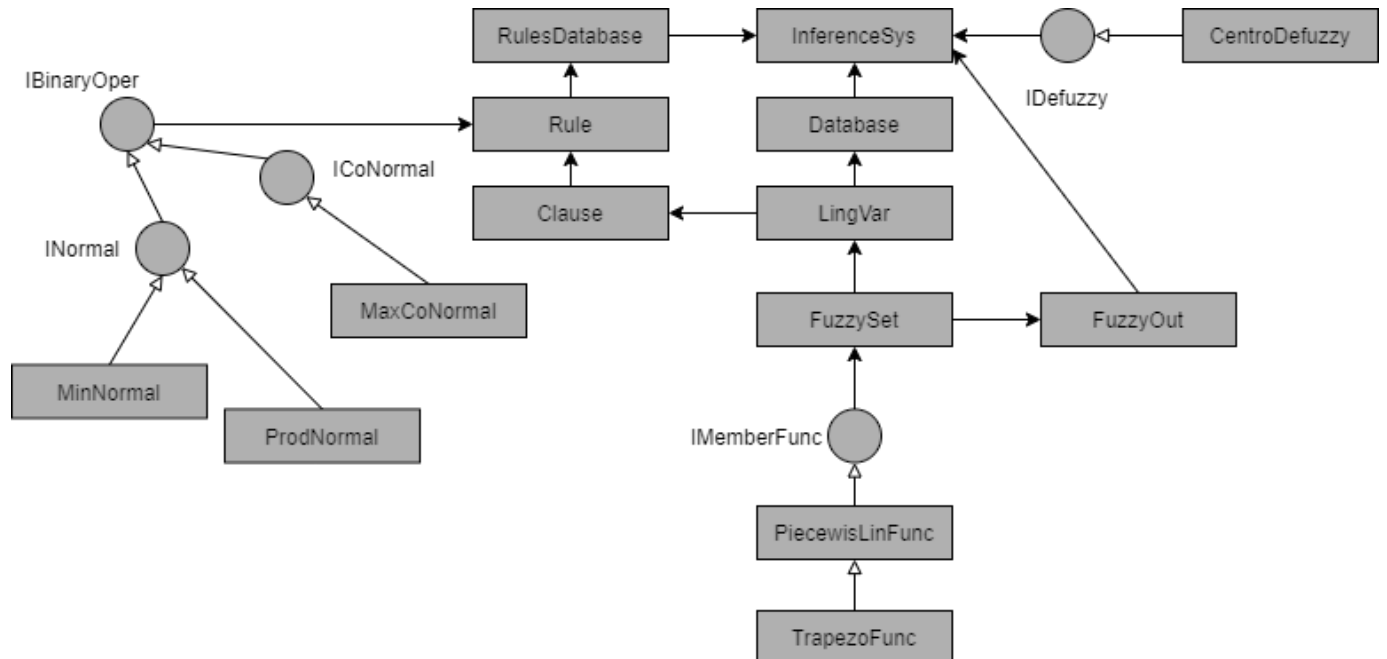


Рисунок 4.5 – Класи розроблюваної системи нечіткого виведення

У якості прикладу подамо програмний код класу процесу дефазифікації «CentroDefuzzy». Цей клас реалізує метод центроїдної дефазифікації.

Результатом роботи системи нечіткого виводу є набір правил з вогневою силою, що більша за нуль. Ця вогнева сила накладає обмеження на наступні нечіткі множини правил. Об'єднання всіх цих нечітких множин разом призводить до форми, яка є лінгвістичним вихідним значенням.

Метод центроїда обчислює центр області цієї фігури, щоб отримати числове представлення результату. Він використовує числове наближення, тому необхідно обрати певну кількість інтервалів. Зі збільшенням кількості інтервалів зростає точність числового результату зростає.

Лістинг 4.1 – Фрагмент коду класу «CentroDefuzzy»

```
public class CentroDefuzzy : IDefuzzifier{
    private int intervals;
    public CentroidDefuzzifier(int intervals){
        this.intervals = intervals;
    }
    public float Defuzzy(FuzzyOut fuzzyOut, INorm normOperator){
        float weightSum = 0, membershipSum = 0;
        float start = fuzzyOut.OutputVariable.Start;
        float end = fuzzyOut.OutputVariable.End;
        float increment = ( end - start ) / this.intervals;
        for (float x = start; x < end; x += increment){
            foreach(FuzzyOut.OutputConstraint oc in fuzzyOut.OutputList){
                float membership = fuzzyOut.OutputVariable.GetLabelMembership(
                    oc.Label, x);
                float constrainedMembership = normOperator.Evaluate(membership,
                    oc.FiringStrength);
                weightSum += x * constrainedMembership;
                membershipSum += constrainedMembership;}
        }
        if (membershipSum == 0)
            throw new Exception("The numerical output is not available");
        return weightSum / membershipSum;}
}
```

Одним з найважливіших етапів у нечітких системах є отримання результату – виведення нечіткої системи. У лістингу 4.2 поданий програмний код класу

«FuzzyOut», що надає такий результат, зберігаючи множину виходів правила у вигляді пари з вихідною нечіткою міткою та вогнестійкістю самого правила.

Лістинг 4.2 – Фрагмент коду класу «FuzzyOut»

```
public class FuzzyOut {
    public class OutConstraint {
        private float firingStrength;
        private string label;
        internal OutConstraint(string label, float fireStrength){
            this.label = label;
            this.fireStrength = fireStrength;
        }
        public string Label
        { get { return label; } }
        public float FireStrength
        { get { return fireStrength; } }
    }
    private List<OutConstraint> outList;
    private LingVar outVar;
    public List<OutConstraint> OutList
    { get { return outList; } }
    public LingVar OutVar
    { get { return outVar; } }
    internal FuzzyOut(LingVar outVar){
        this.outList = new List<OutConstraint>(20);
        this.outVar = outVar;
    }
    internal void AddOut( string labelName, float fireStrength ){
        this.outVar.GetLabel( labelName );
        this.outList.Add(new OutConstraint(labelName, fireStrength ) );
    }
    internal void ClearOut()
    {this.outList.Clear( );}
}
}
```

Серед усього іншого, важливо створити шар доступу до бази даних DAL (Data Access Layer), бо слугує для отримання даних та має усі необхідні методи і класи звернення до реляційної БД.

Як приклад розглянемо код класу «FuzzyRuleDAO», що містить CRUD-операції (додавання, читання, оновлення, видалення) для нечітких правил у базі

даних. У лістингу 4.3 наведено програмний код методу отримання правил нечіткої системи.

Лістинг 4.3 – Фрагмент коду класу «FuzzyRuleDAO»

```
public async Task<IList<FuzzyRuleModel>> Get() {
    try{
        var sqlQuery = @"SELECT * FROM [FuzzyRule] WITH(NOLOCK)";
        using var connection = CreateConnection();
        var result = await connection.QueryAsync<FuzzyRuleModel> (sqlQuery);
        return result.ToList();}
    catch (Exception ex){
        _logger.LogError(ex, "Cannot read FuzzyRule entities");
        throw;}
}
```

Функціонування рекомендаційного елемента у системі, що займається продажем косметичних засобів, неможливе без даних самого клієнта. Коли людина шукає підходящі товари, вона вносить низку відповідей до наданої їй форми.

Саме для цього був розроблений клас об'єкту передачі даних «CustomerDataDto» (DTO, Data Transfer Object), що здійснюватиме передачу даних між процесами та відповідатиме за зчитування інформації з форми та віднесення її до відповідних змінних.

Лістинг 4.4 – Код класу «CustomerDataDto»

```
namespace FuzzyCosmetic.Dto{
    public class CustomerDataDto{
        public int ProductCategoryId { get; set; }
        public int Age { get; set; }
        public SexType Sex { get; set; }
        public float FrequencyWashingHairDays { get; set; }
        public EyeBrowColorType EyeBrowColor { get; set; }
        public HairType HairType { get; set; }
        public float SkinType { get; set; }
        public SkinAcneType SkinAcne { get; set; }}}
```

У результаті написання програмного коду у Microsoft Visual Studio 2022 створена така діаграма класів, що подана на рисунку 4.6.

4.3 Опис розробленого рекомендаційного веб-API

У результаті реалізації програмної логіки був розроблений WEB-API «FuzzyCosmetic». Підключення бібліотеки Swagger дозволяє побудувати інтерфейс для взаємодії з даним API (див. рис. 4.7).

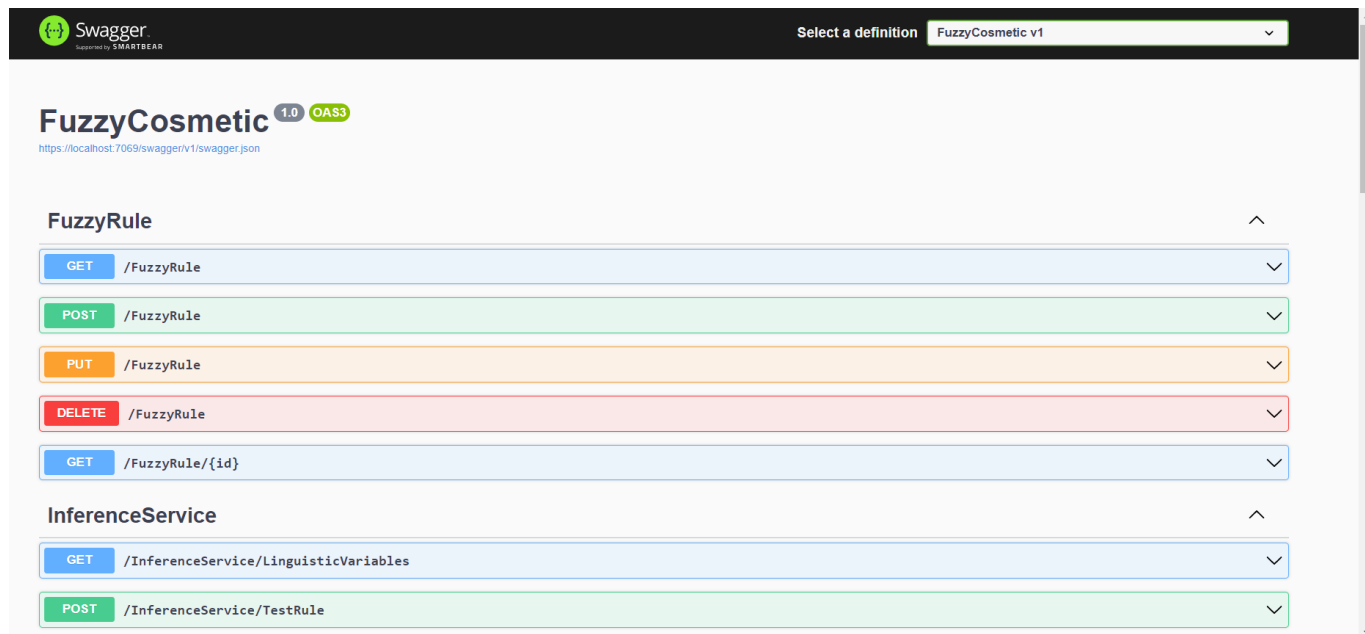


Рисунок 4.7 – Інтерфейс для взаємодії з WEB-API «FuzzyCosmetic»

Окрім контролера «FuzzyRule», що дозволяє експертам здійснювати CRUD-операції над усіма нечіткими правилами, розроблений API має такі наступні три:

- «InferenceService», де реалізована основна логіка нечіткого виведення;
- «ProductCategory» із функціоналом, що дозволяє експертам або бізнес-аналітикам взаємодіяти зі списком категорій товарів у системі за допомогою операцій занесення, зчитування, оновлення та видалення даних;

- «Product» із аналогічним CRUD-функціоналом стосовно усієї продукції Інтернет-магазину.

Інтерфейс вищезазначених контролерів подано на рисунку 4.8.

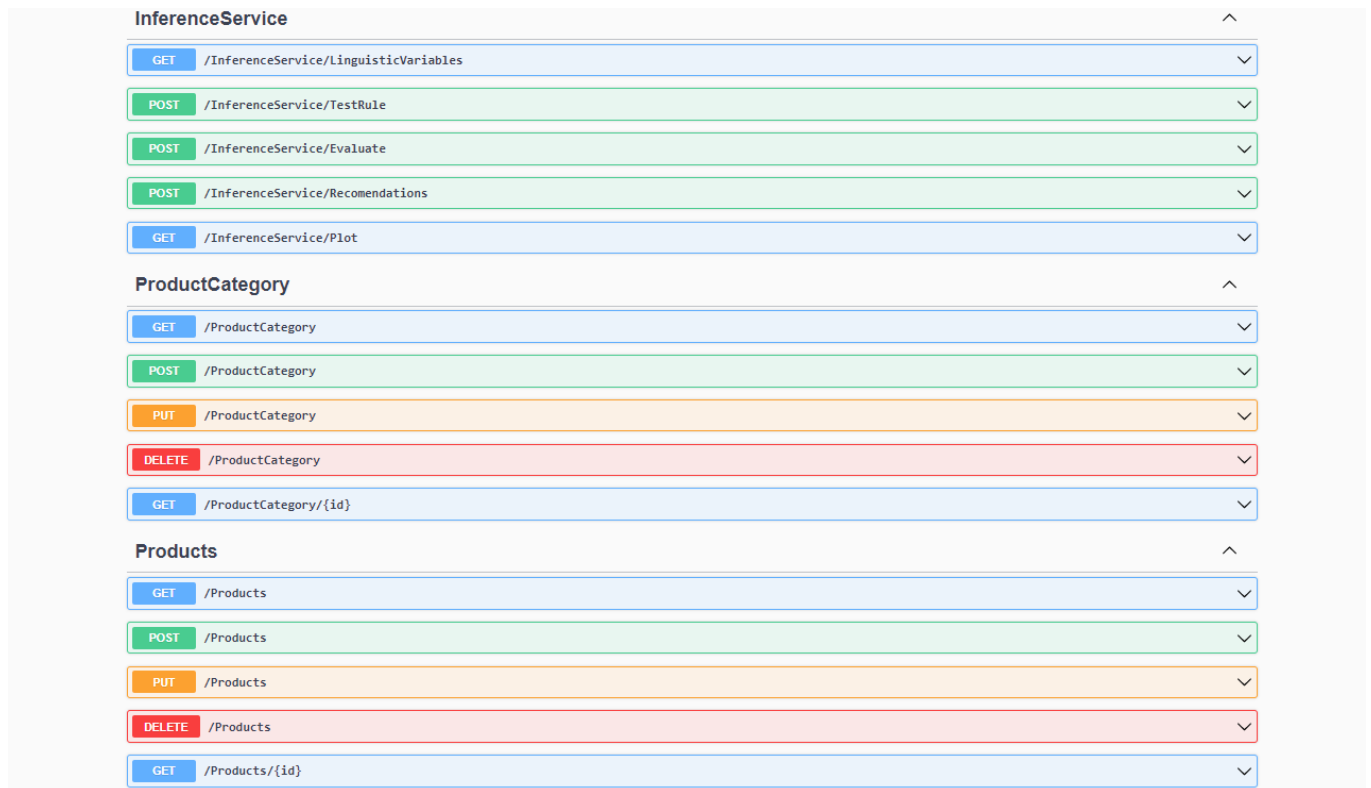


Рисунок 4.8 – Інтерфейс для взаємодії з WEB-API «FuzzyCosmetic»

Отже, реалізований сервіс нечіткого виведення «InferenceService» містить у собі увесь інструментарій нечіткої логіки, який застосовано у обраній предметній області.

У «InferenceService» можна отримувати список лінгвістичних змінних системи («Linguistic Variables»), тестувати наявні нечіткі правила («TestRule») та оцінювати, розраховуючи значення функції приналежності («Evaluate»), будувати графіки для кожної лінгвістичної змінної («Plot») і, що найголовніше у даній кваліфікаційній роботі, надавати рекомендації товарів та послуг, враховуючи персональні показники клієнта («Recommendations»).

Даний сервіс складається з тіла запиту, де вказуються особисті дані користувача, кнопки «Спробувати» та відповідного відклику системи у вигляді списку, що може бути поданий у декількох форматах даних (див. рис. 4.9).

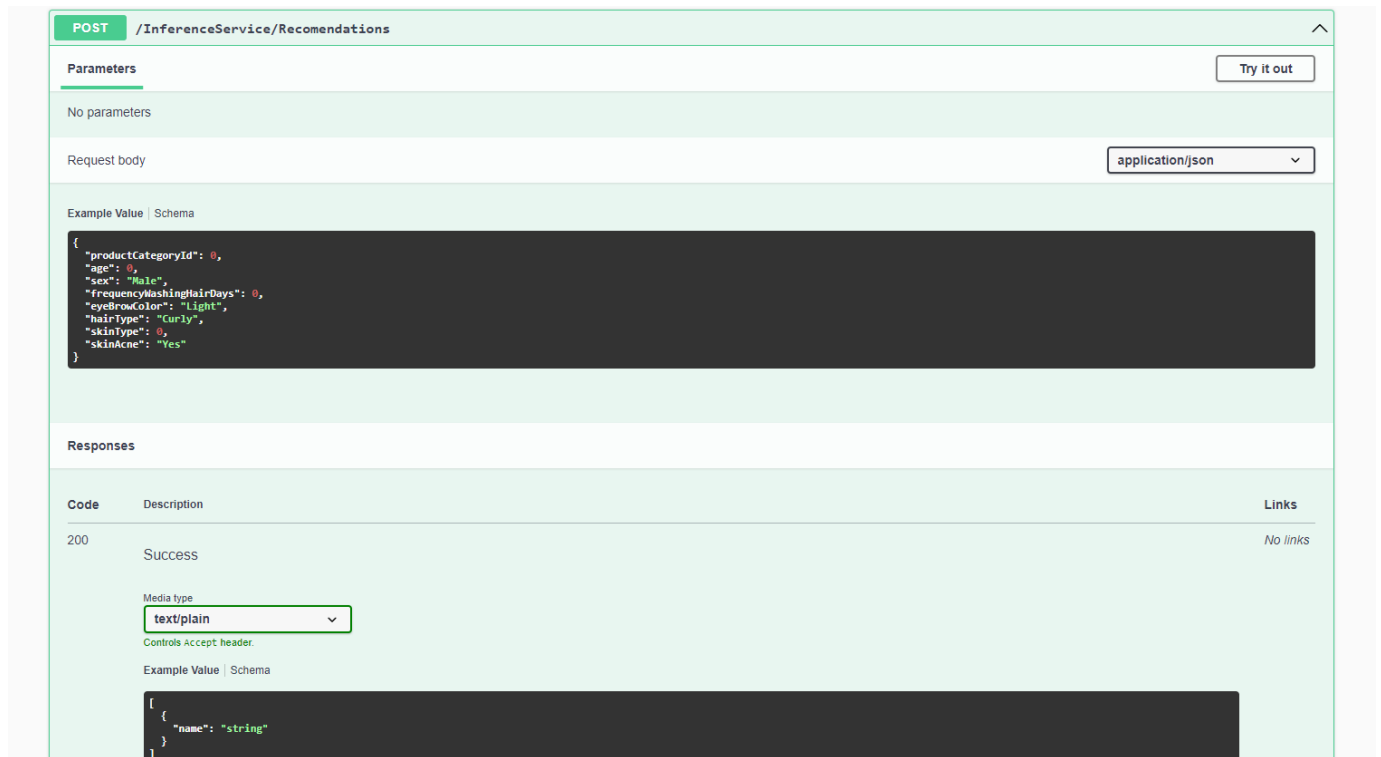


Рисунок 4.9 – Інтерфейс для отримання рекомендацій товарів та послуг «Inference Service Recommendations»

4.4 Проведення експериментів стосовно надання рекомендацій

Невід’ємною частиною кожного процесу розробки є тестування програмної частини проведеного дослідження.

Розроблений веб-АРІ може використовуватись інформаційними системами, що займаються продажем косметичних засобів та надання послуг у сфері краси.

Через те, що цільова аудиторія налічує людей з різними потребами, доцільно протестувати низку типових запитів під час пошуку у відповідних онлайн-крамницях.

Наприклад, дівчина 22 років шукає собі догляд за проблемною шкірою, що схильна до сухості. Приклад тіла запиту наведено на рисунку 4.10.

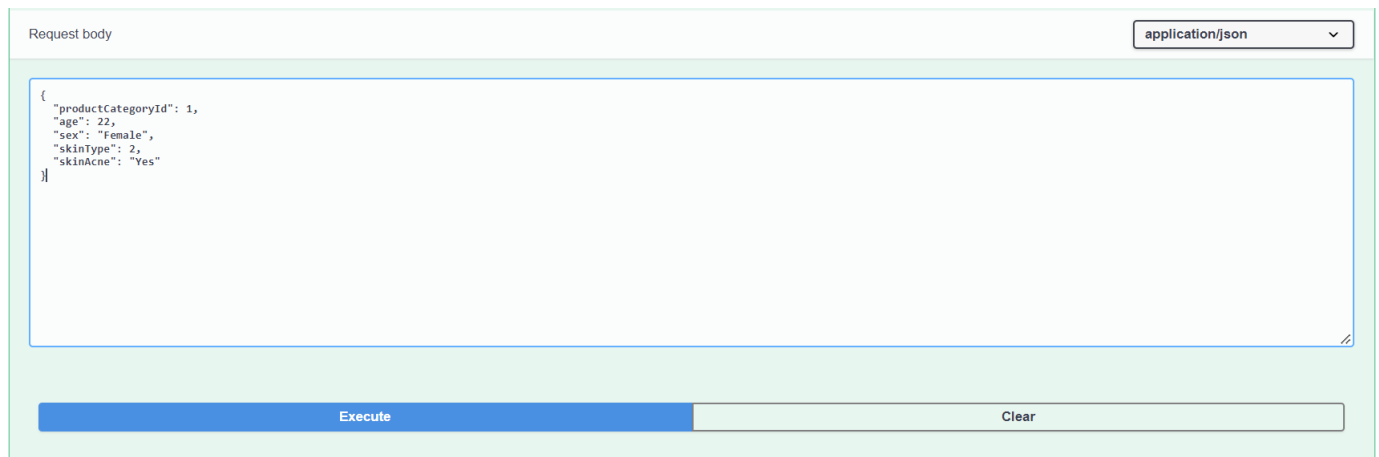


Рисунок 4.10 – Приклад запиту 1

У якості результату видано 3 найкращі та підходящі товари, опис яких свідчить про правильність підбору догляду, а саме такі позиції (див. рис. 4.11):

- поживний крем для глибокого відновлення для дуже сухої шкіри «La Roche-Posay Nutritic Intense Riche»;
- зволожуючий крем для обличчя «Зволоження Експерт» для сухої і чутливої шкіри обличчя з маслом троянди «Loreal Paris Face Cream»;
- тонік для очищення шкіри «Dr. Jart+ Ctrl-A Teatment Toner».

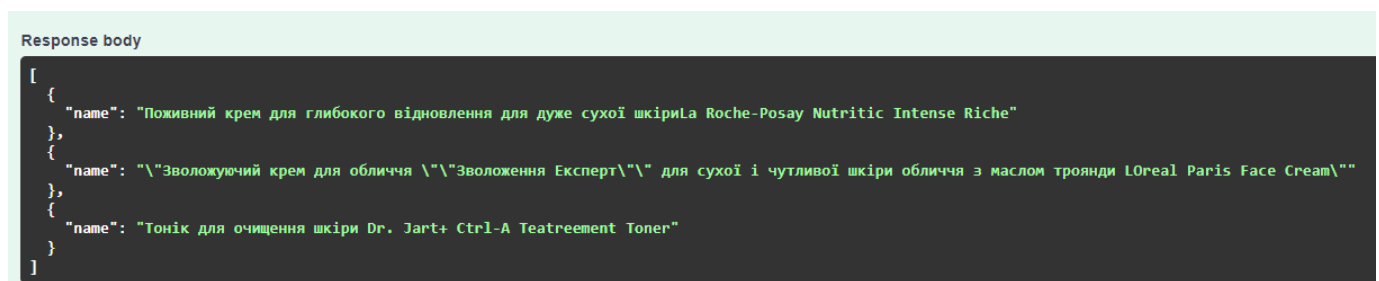


Рисунок 4.11 – Результат видачі рекомендацій на запит 1

Наступним прикладом може бути ситуація, коли дівчинка-білявка 16 років шукає підходящу косметику для її типуажу.

Тобто такому клієнту можна було б рекомендувати засоби для брів світлого кольору та навіть процедуру їх фарбування.

Приклад значень змінних та відповідна відповідь системи наведена на рисунках 4.12 та 4.13, відповідно.

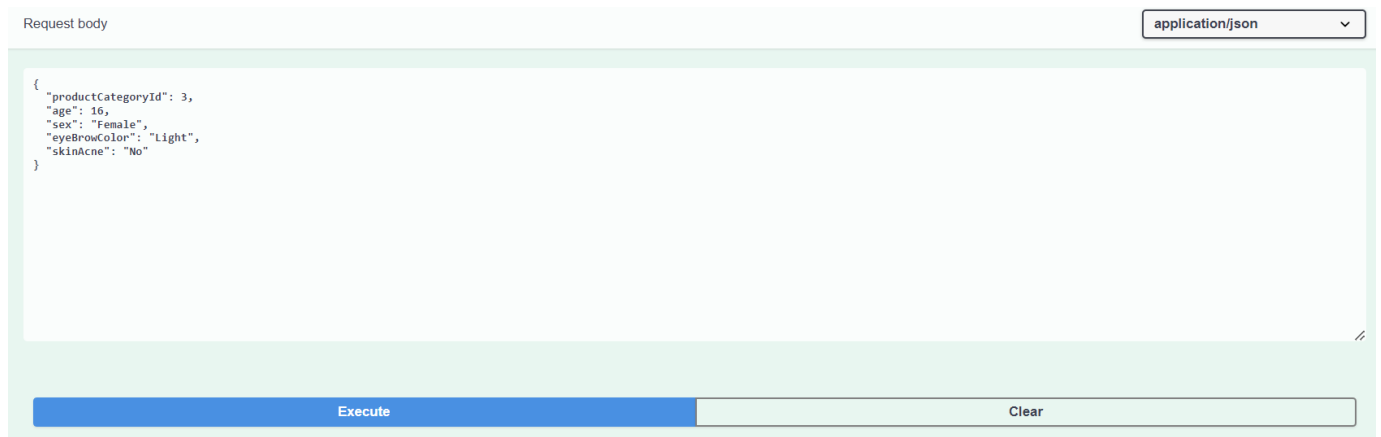


Рисунок 4.12 – Приклад запиту 2

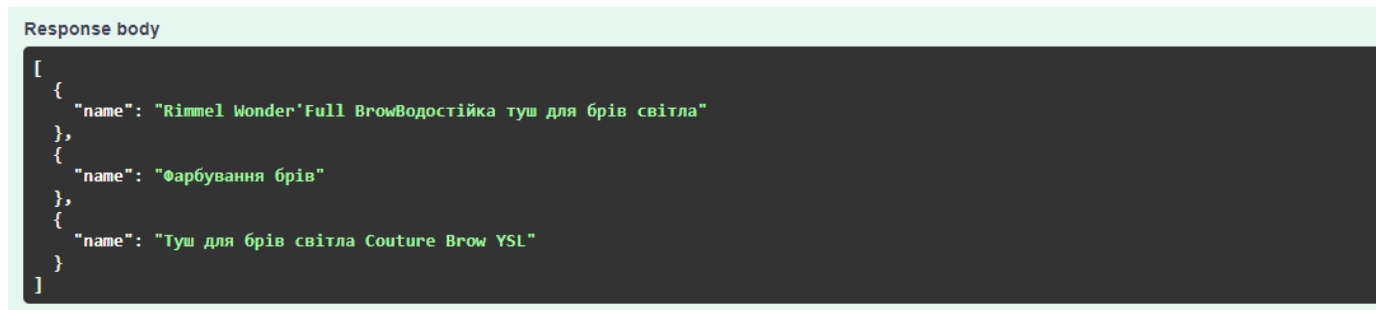


Рисунок 4.13 – Результат видачі рекомендацій на запит 2

Крім того, рекомендації можуть видаватись не лише жінкам, а й усім статям, наприклад, під час пошуку засобів для кучерявого волосся, але швидко брудниться та яке доводиться мити майже щодня.

У такому разі видаються рекомендації стосовно універсальних продуктів, як сухі шампуні або кондиціонери, що знижують рівень вироблення жиру шкірою голови, зволожуючи її.

Запит та відповідні рекомендації приведені на рисунках 4.14 та 4.15 відповідно.

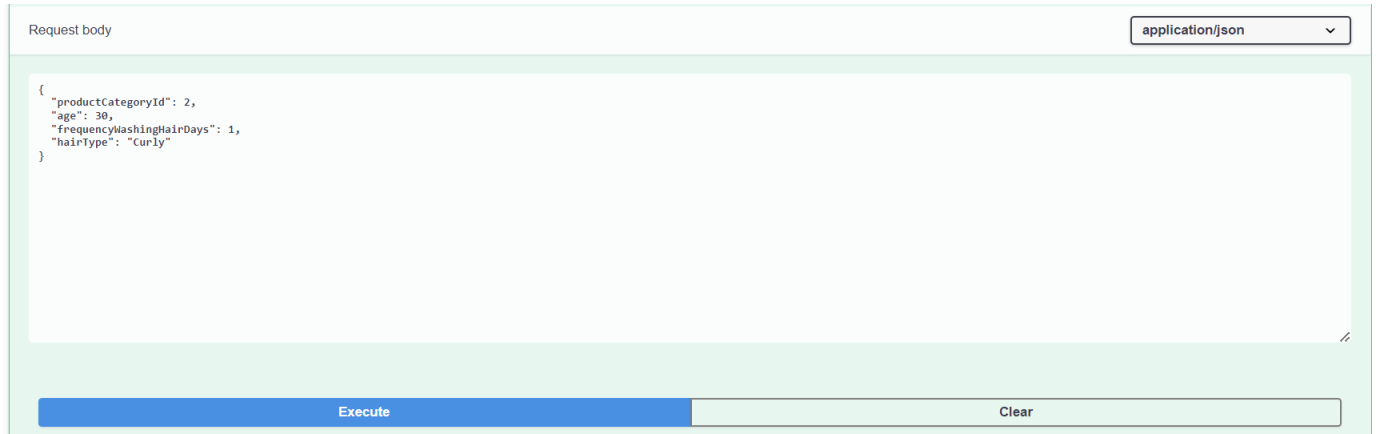


Рисунок 4.14 – Приклад запиту 3

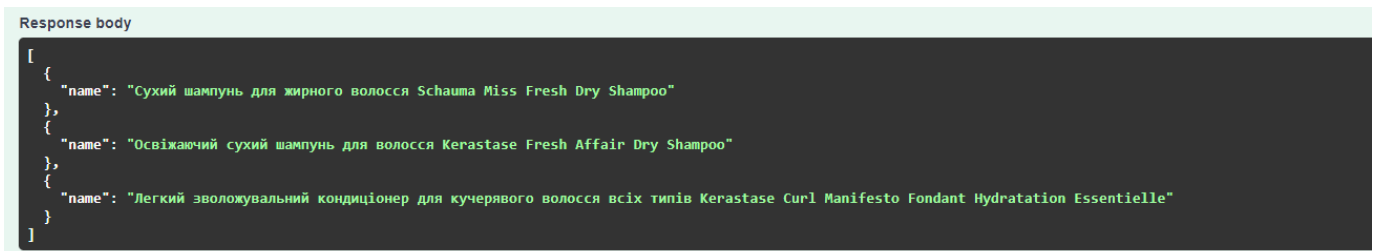


Рисунок 4.15 – Результат видачі рекомендацій на запит 3

Отже, завдяки проведеній низці експериментів можна запевнитись у правильності надання товарних рекомендацій через розроблений API.

Крім того, розроблені нечіткі правила та результат їх застосування дозволяють набрати базову інформацію, яку у подальшому можуть використовувати нейронні мережі задля власного навчання та тренування. А беручи за основу, в свою чергу, розроблену систему, може бути створена інтелектуальна система.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи магістра було проведено дослідження предметної області та вже існуючих інформаційних систем електронної комерції, що займаються продажем косметичних засобів. Дане дослідження дозволило виділити основні проблеми та особливості, що пов'язані з обраною предметною областю.

Крім того, були з'ясовані можливості застосування сучасних технологій для управління інформаційною системою. Для більш швидкого розширення інформаційних об'єктів системи та реалізації усіх її можливостей була розроблена модель функціонування інтерфейсу з надання рекомендації, що базується на застосуванні механізму нечіткої логіки.

З технічної точки зору впровадження розробленого елемента системи не вимагає великих економічних витрат. Об'єктно-орієнтований підхід, використаний в процесі розробки, дозволяє здійснювати подальшу модернізацію системи. Створення рекомендаційної системи у формі API дозволяє інтегрувати її до інших подібних ІС.

Функціональні можливості спроектованого та розробленого рекомендаційного інтерфейсу можуть бути доопрацьовані та розширені у відповідності до вимог тієї сфери, де буде застосовано розроблену систему.

Практичне значення одержаних результатів полягає в запропонованих технологічних рішеннях щодо удосконалення процесів побудови інформаційних систем, які можуть бути використані розробниками програмного забезпечення під час створення системи персоналізованих рекомендацій та при моделюванні подібних їй систем.

Отже, розробленими є інтерфейс формування рекомендацій у інформаційній системі «Інтернет-магазин косметичних засобів» на основі використання інструментів нечіткої логіки, що дозволяє у повній мірі використовувати основні можливості новітніх технологій.

Як результат, можна запевнитись, що розвиток електронної комерції показує, як гнучкі структури стають все більш важливими і що клієнти очікують відповідних рекомендацій щодо продуктів. Вибір хорошої системи рекомендацій може вплинути, перш за все, на величину вартості кошика для покупок. Але не тільки компанія виграє від цікавих рекомендацій. Також підвищується задоволеність клієнтів, що призводить до довготривалих відносин з ними.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Носик К.А. Розробка компонентів системи електронної комерції магазину канцелярських товарів. //25-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у ХХІ столітті». Зб. матеріалів форуму. Т.6. Конференція «Інформаційні інтелектуальні системи» – Харків: ХНУРЕ. 2021. – С.254-255.

2. Магазин парфумерії та косметичних засобів «MakeUp» [Електронне джерело] – Режим доступу до ресурсу: <https://makeup.com.ua/ua/>.

3. Магазин парфумерії та косметичних засобів «Douglas» [Електронне джерело] – Режим доступу до ресурсу: <https://www.douglas.de/de/>.

4. Методичні вказівки до організації виконання та захисту кваліфікаційної роботи на здобуття вищої освіти ступеня магістра спеціальності 122 Комп'ютерні науки, освітньо-професійна програма «Інформаційні технології проектування» / Упорядники: І.В. Гребеннік, В.Г. Іванов, Н.І. Калита, А.І. Коваленко, Д.Е.Ситніков, І.А. Урняєва. Харків: ХНУРЕ, 2021. 52 с.

5. Karlgren, Jussi. 1990. "An Algebra for Recommendations." Syslab Working Paper 179 (1990).

6. Karlgren, Jussi. "Newsgroup Clustering Based On User Behavior-A Recommendation Algebra." SICS Research Report (1994).

7. Hill W., Stead L., Rosenstein M., Furnas G. Recommending and Evaluating Choices in a Virtual Community of Use // Proceeding Conference Human Factors in Computing Systems, 1995. P. 194-201.

8. Montaner, M.; Lopez, B.; de la Rosa, J. L. (June 2003). "A Taxonomy of Recommender Agents on the Internet". Artificial Intelligence Review. 19 (4), p. 285–330.

9. Adomavicius, G.; Tuzhilin, A. (June 2005). "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". IEEE Transactions on Knowledge and Data Engineering. 17 (6), pp. 734–749.

10. Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; Riedl, J. T. (January 2004). "Evaluating collaborative filtering recommender systems". ACM Trans. Inf. Syst. 22 (1), pp. 5-53.

11. Beel, J., Langer, S., Genzmehr, M., Gipp, B., Breitinger, C., Nürnberger, A.: Research paper recommender system evaluation: a quantitative literature survey. In:

Proceedings of the Workshop on Reproducibility and Replication in Recommender Systems Evaluation (RepSys) at the ACM RecSys Conference (RecSys), pp. 15–22 (2013).

12. Beel, J.; Langer, S.; Genzmehr, M.; Gipp, B.; Breiting, C. (October 2013). "Research Paper Recommender System Evaluation: A Quantitative Literature Survey" (PDF). Proceedings of the Workshop on Reproducibility and Replication in Recommender Systems Evaluation (RepSys) at the ACM Recommender System Conference (RecSys).] [Beel, J.; Gipp, B.; Langer, S.; Breiting, C. (26 July 2015). "Research Paper Recommender Systems: A Literature Survey". International Journal on Digital Libraries. 17 (4), pp. 305–338.

13. Ricci F., Rokach L., Shapira B., Kantor P.B. Recommender Systems Handbook. Springer US, 2011, 842 p.

14. Aggarwal C. C. Data mining. The Textbook. Springer International Publishing, 2015, 734 p.

15. Goldberg D., Nichols D., Oki B. M., Terry D. Using collaborative filtering to weave an information Tapestry // Special issue on information filtering, 1992. Vol. 35, pp. 61-70.

16. Francesco Ricci; Lior Rokach; Bracha Shapira; Paul B. Kantor, ed. Recommender Systems Handbook. Springer, ISBN 978-0-387-85819-7.

17. Erdal Kayacan, Mojtaba Ahmadi Khanesar, in Fuzzy Neural Networks for Real Time Control Applications, 2016) 2.3.1.1 Fuzzification <https://www.sciencedirect.com/topics/computer-science/fuzzification>.

18. C# Language Specification (PDF) (4th ed.). Ecma International. June 2006. Retrieved January 26, 2012.

19. "Editions and supported features of SQL Server 2019 (15.x)". 4 November 2019. Retrieved 23 December 2020.

20. Clarke, Steven (2004). "Measuring API Usability". Dr. Dobb's. Retrieved 29 July 2016.

21. <https://swagger.io/>

22. Zadeh, L.A. (1965). "Fuzzy sets". Information and Control 8(3), pp. 338–353.

23. Filho, M.M., Wanke, P.F. and Correa, H.L. (2016) 'Fuzzy logic in production sequencing: the case of a cosmetics manufacturer in Brazil', Int. J. Business Intelligence and Systems Engineering, Vol. 1, No. 1, pp. 2–31.

24. Ross T.J. Fuzzy logic with engineering applications. – McDraw-Hill? 1995. – 600 p.
25. <https://ieeexplore.ieee.org/abstract/document/53>
26. Носик К.А. Розробка методів формування рекомендацій в інформаційній системі з використанням інструментів нечіткої логіки. // Формування сучасної науки: методика та практика: матеріали I Міжнародної студентської наукової конференції. ГО «Молодіжна наукова ліга». — Вінниця: ГО «Європейська наукова платформа», 2022. — С. 213-215.
27. Носик К.А., Лебедев О.Г. Використання інструментарію нечіткої логіки для покращення функціонування електронної комерції. // Проблеми інформатизації Тези доповідей десятої міжнародної науково-технічної конференції — Харків, 2022. — С. 58.
28. Dmitry Sitnikov, Polina Sytnikova, Andrii Kovalenko. Methods of Eliminating Features from Linguistic Equations //Computational Linguistics and Intelligent Systems conference CoLInS 2021/ : CEUR Workshop Proceedings, 2021, Vol-2870, pp. 877-889.
29. ДСТУ 3008-2015. Документація. Звіти в сфері науки і техніки. Структура і правила оформлювання. Чинний від 2017-07-01. – Київ: ДП «УкрНДНЦ», 2016. – 26 с.