

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)

Кафедра _____ Системотехніки _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

_____ Розробка рекомендаційної системи для онлайн-платформи _____
інтерактивних вікторин _____
(тема)

Виконав:
здобувач _____ другого _____ року навчання,
групи _____ ІТІМ-24-1 _____

_____ Ілля КОТЕНКО _____
(власне ім'я, прізвище)

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

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Інформаційні технології _____
проектування _____
(повна назва освітньої програми)

Керівник _____ доц. Тетяна БІЛОВА _____
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри _____
(підпис)

_____ Ігор ГРЕБЕННИК _____
(власне ім'я, прізвище)

2025 р.

Я як студент ХНУРЕ розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів від інших авторів мають посилання на відповідне джерело.

18.12.2025

J. Kotsenko

Котенко

(дата, підпис, прізвище студента)

Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Керівник кваліфікаційної роботи

доц. Білова Т.Г.

Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Керівник кваліфікаційної роботи

доц. Білова Т.Г.

Попередній захист проведено 18.12.2025

Керівник кваліфікаційної роботи

доц. Білова Т.Г.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
Кафедра _____ Системотехніки _____
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)
Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Інформаційні технології проектування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
«___» _____ 20 25 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві КОТЕНКУ Іллі Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи _____ *Розробка рекомендаційної системи для онлайн-платформи інтерактивних вікторин* _____

затверджена наказом університету від 24 листопада 20 25р. № 1058 Ст _____

2. Термін подання здобувачем роботи до екзаменаційної комісії 18 грудня 20 25р.

3. Вихідні дані до роботи Розробити метод та програмні компоненти гібридної рекомендаційної системи для онлайн-платформи інтерактивних вікторин. Провести аналіз предметної області та методів рекомендації контенту, виділити проблеми «холодного старту» та невизначеності даних. Розробити метод гібридної рекомендації, що базується на моделі нечіткого логічного виведення типу Мамдані. Провести функціональне моделювання системи з використанням стандарту IDEF0, розробити діаграми активності та послідовності. Спроекувати архітектуру сервісу рекомендацій. Розробити структуру бази даних для зберігання агрегованої статистики на платформі СУБД MongoDB. Виконати програмну реалізацію серверної частини з використанням Java та Spring Boot. Провести експериментальне тестування розробленого методу та програмного забезпечення.

4. Перелік питань, що потрібно опрацювати в роботі 1 Аналіз предметної області дослідження, 1.1 Аналіз предметної області, 1.2 Аналіз методів вирішення задачі, 1.3 Дослідження існуючих підходів, методів, засобів, технологій, інструментарію для вирішення подібних задач 1.4 Постановка задачі дослідження, 2 Розробка методу гібридної рекомендації на основі нечіткого логічного виведення, 2.1 Структурні обмеження: «Холодний старт» та переспеціалізація, 2.2 Проблематика математичного моделювання метрик: часова динаміка та статична достовірність, 2.3 Узагальнена схема методу вирішення задачі, 2.4 Удосконалена модель функціонування методу гібридної рекомендації, 3 Проєктування методу гібридної рекомендації, 3.1 Опис методу гібридної рекомендації, 3.2 Опис архітектури розробленої системи, 4 Розробка та тестування методу гібридної рекомендації, 4.1 Реалізація гібридної рекомендації, 4.2 Експериментальна перевірка, 4.3 Проблеми використання та перспективи розвитку гібридної

рекомендаційної системи.

5. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основний розділ	Доц. Білова Тетяна Георгіївна		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів	Примітка
1	Отримання завдання кваліфікаційної роботи	24.11	Виконано
2	Аналіз предметної області та методів рекомендації	24.11 – 25.11	Виконано
3	Аналіз існуючих підходів та інструментарію	26.11	Виконано
4	Постановка задачі дослідження	27.11	Виконано
5	Розробка методу гібридної рекомендації	28.11	Виконано
6	Розробка математичних моделі обробки вхідних даних	29.11	Виконано
7	Побудова моделі нечіткого логічного виведення	30.11 – 01.12	Виконано
8	Функціональне моделювання системи	02.12	Виконано
9	Проектування архітектури та UML-діаграм взаємодії	03.12	Виконано
10	Проектування бази даних на платформі СУБД MongoDB	04.12	Виконано
11	Програмна реалізація компонентів розрахунку метрик	05.12 – 06.12	Виконано
12	Програмна реалізація ядра нечіткого виведення	07.12 – 08.12	Виконано
13	Реалізація API та інтеграція сервісу в систему	09.12 – 10.12	Виконано
14	Експериментальне дослідження роботи системи	11.12	Виконано
15	Аналіз результатів та верифікація коду	12.12	Виконано
16	Оформлення пояснювальної записки та графічних матеріалів	13.12 – 14.12	Виконано
17	Представлення на рецензування	15.12	Виконано
18	Представлення кваліфікаційної роботи	16.12	Виконано

Дата видачі завдання 24 листопада 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____
(підпис) доц. Тетяна БІЛОВА
(посада, власне ім'я, прізвище)

РЕФЕРАТ

Звіт з кваліфікаційної роботи магістра містить: 72 сторінки, 20 ілюстрацій, 2 таблиці, 26 джерел.

БАЙЄСІВСЬКЕ УСЕРЕДНЕННЯ, ГІБРИДНА СИСТЕМА, КОРИСТУВАЦЬКИЙ КОНТЕНТ, МОДЕЛЬ МАМДАНІ, НЕЧІТКА ЛОГІКА, ПРОБЛЕМА «ХОЛОДНОГО СТАРТУ», РЕКОМЕНДАЦІЙНА СИСТЕМА.

Об'єкт дослідження - процеси персоналізованого відбору, фільтрації та ранжування контенту в веб-орієнтованих інформаційних системах управління інтерактивними вікторинами. Ці процеси охоплюють повний цикл обробки даних: від збору первинних сигналів про взаємодію користувача (перегляди, оцінки, додавання до бажаного) до формування фінального списку рекомендацій.

Предмет дослідження – методи та моделі побудови гібридних рекомендаційних систем, що використовують апарат нечіткої логіки для прийняття рішень в умовах невизначеності вхідних даних, розрідженості простору оцінок та проблеми «холодного старту».

Мета дослідження – підвищення ефективності функціонування онлайн-платформи інтерактивних вікторин шляхом розробки та програмної реалізації гібридної рекомендаційної системи, що базується на моделі нечіткого логічного виведення типу Мамдані.

Методи досліджень – апарат теорії нечітких множин, Байєсівський підхід, функціональний аналіз, системний аналіз, методи програмної інженерії.

Результатом дослідження є розроблена та програмно реалізована гібридна рекомендаційна система, яка функціонує як окремий сервіс. Наукова новизна дослідження полягає у розширенні методу рекомендацій моделлю нечіткого логічного виведення типу Мамдані, що дозволяє інтегрувати Байєсівські та експоненціальні метрики для формування стійких і прозорих рекомендацій.

Сферою застосування є веб-орієнтовані інформаційні системи управління користувачьким контентом, зокрема, онлайн-платформи інтерактивних вікторин, що функціонують у розподіленому середовищі сервісної архітектури.

ABSTRACT

Master thesis: 72 pages, 20 figures, 2 tables, 26 references.

BAYESIAN AVERAGING, COLD START PROBLEM, FUZZY LOGIC, HYBRID SYSTEM, MAMDANI MODEL, RECOMMENDER SYSTEM, USER-GENERATED CONTENT.

Research object is the processes of personalized selection, filtration, and ranking of content in web-oriented information systems for managing interactive quizzes. These processes cover the full data processing cycle: from collecting primary signals about user interaction (views, evaluations, adding to wishlist) to forming the final list of recommendations.

Research subject is methods and models for constructing hybrid recommender systems that use the fuzzy logic apparatus to make decisions under conditions of input data uncertainty, sparsity of the rating space, and the "cold start" problem.

Aim of the research is to increase the efficiency of the online interactive quiz platform functioning by developing and implementing a hybrid recommender system based on the Mamdani fuzzy inference model.

The research methods include the fuzzy set theory apparatus, Bayesian approach, functional analysis, system analysis, and software engineering methods.

The result of the research is a developed and software-implemented hybrid recommender system that functions as a separate service. The scientific novelty of the research lies in expanding the recommendation method with a Mamdani-type fuzzy logic inference model, which allows integrating Bayesian and exponential metrics to form stable and transparent recommendations.

The sphere of application is web-oriented information systems for managing user-generated content, in particular, online interactive quiz platforms functioning in a distributed service architecture environment.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної області дослідження	10
1.1 Аналіз предметної області	10
1.2 Аналіз методів вирішення задачі	11
1.3 Дослідження існуючих підходів, методів, засобів, технологій, інструментарію для вирішення подібних задач	13
1.4 Постановка задачі дослідження	15
2 Розробка методу гібридної рекомендації на основі нечіткого логічного виведення	17
2.1 Структурні обмеження: «Холодний старт» та переспеціалізація	17
2.2 Проблематика математичного моделювання метрик: часова динаміка та статистична достовірність	18
2.3 Узагальнена схема методу вирішення задачі	19
2.4 Удосконалена модель функціонування методу гібридної рекомендації.....	21
3 Проектування рекомендаційної системи на основі методу гібридної рекомендації.....	34
3.1 Опис методу гібридної рекомендації	34
3.2 Опис архітектури розробленої системи	42
4 Розробка та тестування гібридної рекомендаційної системи	45
4.1 Реалізація гібридної рекомендації	45
4.2 Експериментальна перевірка.....	52
4.3 Проблеми використання та перспективи розвитку гібридної рекомендаційної системи	57
Висновки	58
Перелік джерел посилання	59
Додаток А Графічні матеріали кваліфікаційної роботи	5962

ВСТУП

Актуальність роботи зумовлена необхідністю підвищення ефективності функціонування веб-орієнтованих платформ, в основі яких лежить користувацький контент (UGC), зокрема платформ інтерактивних вікторин. Нестримне зростання обсягів контенту призвело до гострої проблеми інформаційного перевантаження [1]. Це створює двосторонній бар'єр: користувачі не можуть знайти цікавих для них вікторин серед великої кількості пропозицій, що знижує їхню залученість, а автори, в свою чергу, страждають від низької активності та видимості своїх робіт. Ефективність традиційних методів фільтрування, таких як колаборативна фільтрація, для такого суб'єктивного та динамічного контенту є недостатньо високою, оскільки вони не здатні вирішувати проблему «холодного старту» для нових об'єктів і коректно обробляти нечітку природу людських вподобань.

Це формує науково-технічне протиріччя між можливостями існуючих алгоритмів та задачами практики, зокрема необхідністю створення системи, яка б поєднувала адаптивність, інтерпретованість та стійкість до відсутності даних. Саме тому розробка рекомендаційної системи, здатної моделювати нечіткі людські вподобання та вирішувати проблему видимості контенту, є актуальною задачею.

Об'єктом дослідження є процеси персоналізованого відбору, фільтрації та ранжування контенту в веб-орієнтованих інформаційних системах управління інтерактивними вікторинами.

Метою роботи є підвищення ефективності функціонування онлайн-платформи інтерактивних вікторин шляхом розробки та програмної реалізації гібридної рекомендаційної системи, що базується на моделі нечіткого логічного виведення типу Мамдані.

Для досягнення поставленої мети використовується комплекс методів дослідження, включаючи апарат теорії нечітких множин (модель Мамдані) для інтеграції різнорідних факторів релевантності та Байєсівський підхід для забезпечення статистичної достовірності оцінок. Додатково використовувався функціональний аналіз для моделювання нелінійної часової актуальності контенту та системний аналіз для декомпозиції задачі і реалізації гібридної моделі в розподіленій сервісній архітектурі.

Наукова новизна дослідження полягає у розробці удосконаленого гібридного методу рекомендації, який поєднує модель нечіткого логічного виведення типу

Мамдані як інтелектуальний агрегатор для обробки суб'єктивних факторів вподобань, а також адаптовані математичні моделі попередньої обробки даних: Байєсівське усереднення рейтингу для вирішення проблеми малих вибірок та експоненціальне згасання для коректного моделювання часової актуальності контенту.

Практичне значення роботи полягає у програмній реалізації розробленого методу у вигляді окремого сервісу, що забезпечує гнучкість, масштабованість та високу інтерпретованість рекомендацій в умовах динамічного UGC-контенту.

Результати кваліфікаційної роботи апробовано на IX Всеукраїнській студентській науковій конференції «Розвиток сучасної науки: актуальні питання теорії та практики» [2].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДОСЛІДЖЕННЯ

1.1 Аналіз предметної області

Інформаційна система є фундаментальним поняттям у сучасній ІТ-сфері, яке можна визначити як організовану сукупність взаємопов'язаних компонентів (людей, обладнання, програмного забезпечення, баз даних та процедур), що збирають, обробляють, зберігають та розповсюджують інформацію для підтримки прийняття рішень, координації та контролю в організації [3]. Згідно з класичним визначенням Кеннета та Джейн Лаудон (2020), інформаційна система технічно являє собою набір компонентів, які перетворюють "сирі" дані на значущу інформацію через три базові активності: введення, обробку та виведення, доповнені механізмом зворотного зв'язку. У контексті веб-розробки, ІС часто розглядаються як соціотехнічні системи, де технологічна складова нерозривно пов'язана з соціальною (користувачами та їхніми задачами).

Класифікація інформаційних систем є розгалуженою і залежить від обраного критерію. За рівнем управління в організації виділяють системи обробки транзакцій для операційного рівня, управлінські інформаційні системи та системи підтримки прийняття рішень для тактичного рівня, а також системи підтримки керівників для стратегічного рівня. За функціональним призначенням системи поділяються на маркетингові, фінансові, кадрові, виробничі тощо. Окремий важливий пласт складають веб-орієнтовані системи управління контентом (CMS), які забезпечують життєвий цикл цифрової інформації. Саме до цього класу, з певною специфікацією, можна віднести платформи для інтерактивних вікторин.

Система управління користувачьким контентом (UGS) є специфічним класом інформаційних систем, що функціонують як багатосторонні платформи, де основна цінність створюється самими користувачами, а не адміністрацією ресурсу. Такі системи поєднують у собі риси системи керування контентом для створення та зберігання медіа-об'єктів (вікторин, зображень, текстів) та соціальних мереж. Їх головна відмінність полягає у тому, що контент є динамічним, неструктурованим та масовим, що вимагає від архітектури системи особливих підходів до модерації, категоризації та дистрибуції інформації.

Структура сучасної UGS-платформи є складною і зазвичай включає кілька ключових підсистем, серед яких критичну роль відіграють модуль управління

взаємовідносинами з клієнтами (CRM) та модуль рекомендацій. CRM-компонента в таких системах трансформується у так званий соціальний CRM. На відміну від традиційних CRM, які фокусуються на транзакціях, соціальний CRM у UGC-платформах відповідає за управління профілем користувача, відстеження його соціальної активності (лайки, коментарі, підписки) та формування спільноти навколо авторів контенту. Це дозволяє перетворити пасивного споживача на лояльного учасника спільноти або навіть автора.

Рекомендаційна система у структурі UGC-платформи виступає не просто як допоміжний сервіс, а як основний інструмент навігації та утримання користувачів [4]. В умовах інформаційного перевантаження, коли користувачі генерують контент швидше, ніж його можна переглянути, рекомендаційна система виконує роль інтелектуального фільтра. Вона аналізує історичну поведінку користувача (проходження вікторин, оцінки) та атрибути контенту (теги, категорії), щоб персоналізувати рекомендації. Дослідження показують, що ефективна рекомендаційна система безпосередньо впливає не лише на споживання контенту, але й стимулює його створення, оскільки автори отримують більше переглядів та зворотного зв'язку, що мотивує їх продовжувати діяльність на платформі. Таким чином, архітектура UGC-системи базується на тісній інтеграції бази даних контенту, CRM-модуля для обліку користувацьких переваг та рекомендаційного рушія для персоналізації досвіду.

1.2 Аналіз методів вирішення задачі

У сфері розробки систем для платформ користувацького контенту задача персоналізації вирішується за допомогою різноманітних алгоритмічних підходів. Аналіз предметної області дозволив виділити три класичні групи методів рекомендацій: фільтрація на основі змісту, колаборативна фільтрація та гібридні методи. Крім того, для задач із високим рівнем невизначеності все частіше застосовуються методи нечіткої логіки.

Фільтрація на основі змісту базується на аналізі атрибутів об'єктів та їх порівнянні з профілем інтересів користувача. У контексті вікторин цей метод передбачає створення опису кожного об'єкта за допомогою метаданих: тегів, категорій, ключових слів. Головною перевагою підходу є стійкість до проблеми «холодного старту» для нових об'єктів: як тільки нова вікторина публікується, вона

одразу доступна для рекомендацій. Основним недоліком є схильність до переспеціалізації, коли користувачу пропонується лише одноманітний контент, що обмежує новизну пропозицій.

Колаборативна фільтрація спирається не на властивості об'єктів, а на історію колективної взаємодії користувачів із системою. Метод базується на припущенні, що користувачі зі схожими оцінками в минулому матимуть спільні інтереси й у майбутньому. Цей підхід забезпечує високу точність та здатність пропонувати несподіваний, але цікавий контент. Проте він має суттєві обмеження: залежність від кількості накопичених даних (проблема «холодного старту» для нових користувачів і вікторин) та зниження ефективності при розрідженості матриці оцінок.

Гібридні рекомендаційні системи поєднують переваги попередніх методів для нівелювання їхніх недоліків. Зазвичай вони використовують змістовні атрибути для нових об'єктів та колаборативні дані для популярних, що підвищує загальну точність та стійкість системи.

Особливе місце в сучасних інтелектуальних системах займають методи на базі нечіткої логіки. Вони дозволяють оперувати неточними та суб'єктивними поняттями (наприклад, «подобається», «трохи цікаво»), що є характерним для людського сприйняття. Серед моделей нечіткого логічного виведення виділяють три основні підходи: Мамдані, Сугено (Такагі-Сугено) та Цукамото.

Метод Мамдані є найбільш інтуїтивним та поширеним у системах підтримки прийняття рішень. Його ключова особливість полягає в тому, що як вхідні, так і вихідні змінні описуються нечіткими множинами. Це забезпечує найвищий рівень інтерпретованості: база правил формулюється природною мовою (наприклад, «Якщо оцінка висока, то рекомендувати»). До недоліків методу відносять високу обчислювальну складність на етапі дефазифікації (перетворення нечіткого висновку в чітке число), що може впливати на швидкодію. Однак для задач, де важлива прозорість логіки рекомендацій, цей метод є пріоритетним.

Метод Сугено відрізняється способом формування висновку: замість нечіткої множини використовується лінійна математична функція або константа. Це робить алгоритм обчислювально ефективним та зручним для задач автоматичного керування та оптимізації. Проте заміна лінгвістичного висновку на формулу суттєво знижує інтерпретованість правил для людини, що ускладнює налаштування системи експертами у предметній області рекомендацій.

Метод Цукамото є компромісним варіантом, де висновки правил задаються нечіткими множинами з монотонними функціями належності. Це спрощує процес отримання чіткого результату порівняно з Мамдані, але накладає жорсткі обмеження на вигляд функцій, що зменшує гнучкість моделювання складних залежностей.

Враховуючи специфіку платформи інтерактивних вікторин, де критично важливою є здатність системи емулювати людську логіку вибору та пояснювати причини рекомендацій, найбільш доцільним є використання методу Мамдані. Він дозволяє побудувати гнучку базу знань, яка ефективно обробляє суб'єктивні вподобання користувачів навіть в умовах неповної інформації.

1.3 Дослідження існуючих підходів, методів, засобів, технологій, інструментарію для вирішення подібних задач

У процесі проектування рекомендаційних систем для платформ з динамічним контентом критично важливим етапом є вибір методологічного апарату, який би дозволив ефективно нівелювати недоліки класичних алгоритмів. Аналіз наукових джерел свідчить про те, що для складних предметних областей, де присутня висока невизначеність вподобань користувачів, найбільш перспективним є використання гібридних моделей та апарату нечіткої логіки.

Дослідження сучасних підходів до побудови рекомендаційних систем у різних доменах, від електронного навчання до туризму та агропромисловості, підтверджує ефективність інтеграції різномірних методів фільтрації. Так, у роботі, присвяченій рекомендаційній системі для сфери туризму, авторами було запропоновано новий гібридний алгоритм, що поєднує байєсівський компонент виявлення вподобань з контент-орієнтованою фільтрацією. Результати експериментів показали, що такий гібридний підхід дозволяє ефективно вирішувати проблему «холодного старту» — ситуації, коли система не може надати релевантні рекомендації новим користувачам через відсутність історії оцінок. Для платформи інтерактивних вікторин, де контент постійно оновлюється, а життєвий цикл окремої вікторини може бути коротким, стійкість до проблеми холодного старту є визначальним фактором якості системи. Крім того, гібридизація дозволяє використовувати як семантичну схожість об'єктів, так і поведінкові фактори

користувачів, забезпечуючи вищу точність рекомендацій порівняно з монолітними підходами [5].

Окрему увагу в науковій літературі приділено використанню нечіткої логіки як інструменту для моделювання суб'єктивних та неточних даних. У системах електронного навчання було успішно застосовано нечітку логіку для персоналізації навчального процесу. Автори зазначають, що використання нечітких множин дозволяє системі оперувати лінгвістичними змінними (наприклад, оцінка «добре» або «незадовільно») замість чітких числових значень, що наближає процес прийняття рішень до людського мислення. Це особливо актуально для систем користувацького контенту, де оцінки користувачів (лайки, дизлайки) є суб'єктивним відображенням їхнього задоволення, яке важко формалізувати чіткими правилами [6].

Серед різних моделей нечіткого виведення (Мамдані, Сугено, Цукамото) для задач рекомендації найчастіше обирається модель Мамдані. У дослідженні системи підтримки прийняття рішень для гуманітарного реагування обґрунтовано, що система нечіткого виведення Мамдані має суттєві переваги: легку інтерпретованість правил виведення, гнучкість при роботі з суперечливою інформацією та простоту імплементації. Оскільки рекомендаційна система повинна бути прозорою («чому мені це порекомендували?»), можливість сформулювати базу знань у вигляді зрозумілих правил «ЯКЩО-ТОДІ» (IF-THEN) є критичною перевагою методу Мамдані над іншими підходами, такими як Сугено, де висновок є математичною функцією [7].

Ефективність моделі Мамдані також підтверджується у прикладних задачах рекомендації, де вхідні дані мають різну природу. Наприклад, у розробленій системі рекомендацій для фермерів використовується модель Мамдані для прогнозування врожайності на основі нечітких вхідних змінних (температура, вологість). Система використовує трикутні функції належності для фазифікації вхідних даних та набір з 27 правил для логічного виведення, що дозволяє надавати точні рекомендації навіть при неповних вхідних даних. Такий підхід корелює із задачею рекомендації вікторин, де вхідні метрики (популярність, схожість тегів) також доцільно представляти у вигляді лінгвістичних термів («низька», «середня», «висока») для подальшої агрегації [8].

Отже, аналіз існуючих розробок дозволяє зробити висновок, що для створення рекомендаційної системи платформи інтерактивних вікторин найбільш доцільним є використання гібридного підходу, посиленого механізмом нечіткого

логічного виведення типу Мамдані. Це забезпечить гнучкість обробки різномірних метрик, вирішення проблеми холодного старту та високу інтерпретованість результатів роботи системи.

1.4 Постановка задачі дослідження

Проведений у попередніх розділах аналіз предметної області та існуючих інструментальних засобів дозволив виявити ключові виклики, що постають перед розробниками систем управління користувачьким контентом (UGC). Специфіка платформ інтерактивних вікторин полягає у високій динаміці появи нового контенту та значній суб'єктивності критеріїв його оцінювання користувачами. В таких умовах класичні методи рекомендацій часто виявляються недостатньо ефективними, що зумовлює необхідність формулювання науково-обґрунтованої постановки задачі для розробки вдосконаленої системи.

Об'єктом дослідження є процеси персоналізованого відбору, фільтрації та ранжування контенту в веб-орієнтованих інформаційних системах управління інтерактивними вікторинами. Ці процеси охоплюють повний цикл обробки даних: від збору первинних сигналів про взаємодію користувача (перегляди, оцінки, додавання до бажаного) до формування фінального списку рекомендацій.

Предметом дослідження є методи та моделі побудови гібридних рекомендаційних систем, що використовують апарат нечіткої логіки для прийняття рішень в умовах невизначеності вхідних даних, розрідженості простору оцінок та проблеми «холодного старту».

Ключова проблема дослідження полягає у наявності науково-технічного протиріччя між можливостями існуючих класичних алгоритмів та актуальними задачами практики. З одного боку, методи колаборативної фільтрації забезпечують високу точність лише за наявності великої історії взаємодій, що робить їх безпорадними перед новими вікторинами («холодний старт»). З іншого боку, жорсткі правила контент-орієнтованої фільтрації часто призводять до переспеціалізації системи («ефект бульбашки»), обмежуючи користувача однотипним контентом. Крім того, жоден з класичних методів не враховує повною мірою нечітку природу людських вподобань, де поняття «подобається» не завжди є бінарним (0 або 1), а має градації. Необхідність вирішення цього протиріччя

шляхом створення системи, яка б поєднувала адаптивність, інтерпретованість та стійкість до відсутності даних, і складає сутність науково-прикладної задачі.

Метою роботи є підвищення ефективності функціонування онлайн-платформи інтерактивних вікторин шляхом розробки та програмної реалізації гібридної рекомендаційної системи на основі нечіткого логічного виведення. Для досягнення цієї мети необхідно вирішити такий комплекс взаємопов'язаних задач:

- розробити підхід до попередньої обробки та нормалізації різнорідних вхідних даних. Це передбачає формування набору контентних та соціальних метрик, розробку математичних моделей їх обробки (зокрема Байєсівського усереднення та експоненціального згасання) і приведення всіх показників до уніфікованого діапазону для подальшої фазифікації;

- побудувати модель нечіткого логічного виведення типу Мамдані для оцінки релевантності контенту. Задача включає проєктування лінгвістичних змінних, функцій належності та бази експертних правил, що забезпечують коректну обробку суперечливих даних та прийняття рішень в умовах невизначеності;

- розробити метод гібридної рекомендації. Необхідно спроектувати алгоритм, який інтегрує розроблені математичні моделі та базу знань у єдиний конвеєр обробки даних. Ця задача включає опис послідовності етапів;

- виконати програмну імплементацію системи та її інтеграцію в сервісну архітектуру. Фінальною задачею є реалізація розроблених моделей у вигляді програмного коду з використанням сучасних технологій, таких як Java, Spring Boot, MongoDB. Це передбачає створення механізмів взаємодії з іншими сервісами платформи та забезпечення продуктивності системи, зокрема через використання кешування та асинхронної обробки статистики;

- провести експериментальне дослідження розробленого методу та програмного забезпечення. Для цього необхідно розробити сценарії тестування, які моделюють типові виклики, сформувати тестовий набір даних та виконати верифікацію коректності ранжування шляхом порівняння отриманих результатів з очікуваною експертною оцінкою.

Вирішення перелічених задач дозволить створити систему, яка здатна надавати релевантні рекомендації як для нових, так і для досвідчених користувачів, забезпечуючи при цьому високу прозорість прийняття рішень завдяки використанню лінгвістичних правил.

2 РОЗРОБКА МЕТОДУ ГІБРИДНОЇ РЕКОМЕНДАЦІЇ НА ОСНОВІ НЕЧІТКОГО ЛОГІЧНОГО ВИВЕДЕННЯ

2.1 Структурні обмеження: «Холодний старт» та переспеціалізація

Найбільш критичною для систем колаборативної фільтрації є проблема «холодного старту». У контексті вікторин вона проявляється у двох аспектах:

– «холодний старт об'єкта»: нова вікторина не має історії взаємодій (проходжень, лайків), а отже, має нульовий вектор ознак. Це створює ефект «замкненого кола невидимості», коли вікторину не рекомендують через відсутність оцінок, а оцінки не з'являються через відсутність рекомендацій;

– «холодний старт користувача»: новий учасник ще не сформував історію дій, тому система не може побудувати його профіль інтересів, що призводить до видачі нерелевантних, узагальнених рекомендацій [9].

Математично цю проблему можна продемонструвати через неможливість обчислення метрик подібності. Нехай r_u — вектор оцінок користувача u . Для нового користувача u_{new} , який ще не здійснив жодної дії, вектор є нульовим: $r_{u_{new}}$. При спробі знайти «сусідів» (схожих користувачів) за допомогою косинусної міри подібності виникає невизначеність через ділення на нуль:

$$sim(u_{new}, v) = \cos(0) = \frac{r_{u_{new}} \cdot r_v}{\|r_{u_{new}}\| \|r_v\|} = \frac{0}{0}.$$

Математична неможливість обчислити знаменник унеможливорює формування рекомендаційного списку класичними колаборативними методами, що створює бар'єр для входження нових користувачів у систему.

Іншою проблемою, притаманною методам фільтрації на основі змісту, є переспеціалізація («ефект бульбашки»). Хоча ці методи вирішують проблему холодного старту об'єкта, вони схильні замикати користувача у вузькому колі інтересів. Якщо користувач взаємодіяв лише з вікторинами категорії «Кіно», система пропонуватиме виключно схожий контент, ігноруючи можливість того, що користувачеві можуть бути цікаві популярні тренди в інших категоріях.

Формально це описується через ортогональність векторів інтересів. Нехай профіль користувача P_u та профіль об'єкта Q_i представлені у n -вимірному просторі

ознак. Релевантність розраховується як скалярний добуток: $Score(u, i) = P_u \cdot Q_i$. Якщо профіль користувача P_u зосереджений виключно на темі A , а новий популярний об'єкт Q_i належить до теми B , їх скалярний добуток дорівнюватиме нулю:

$$P_u^A \perp Q_i^B \rightarrow Score(u, i) = P_u \cdot Q_i = 0.$$

Це математично гарантує, що система ніколи не порекомендує користувачу щось нове з іншої категорії, навіть якщо цей контент є об'єктивно якісним, що обмежує дослідження платформи.

2.2 Проблематика математичного моделювання метрик: часова динаміка та статистична достовірність

Друга група проблем стосується недосконалості формул, що використовуються для розрахунку ваги (рейтингу) об'єктів.

Суттєвою проблемою є моделювання часової динаміки. Контент на UGC-платформах швидко «старіє». Класичні методи часто розглядають релевантність як статичну величину або використовують примітивні моделі врахування часу (наприклад, лінійне зниження ваги або жорстке відсікання контенту старше певного терміну). Такий підхід не відповідає психології сприйняття новизни користувачем:

- лінійне згасання не відображає реальної втрати актуальності, оскільки різниця між контентом віком 1 день і 7 днів є значно суттєвішою для користувача, ніж різниця між контентом віком 100 і 107 днів;

- жорсткі часові вікна можуть призвести до виключення з рекомендацій якісних «вічнозелених» вікторин, які залишаються цікавими незалежно від дати створення.

У багатьох системах використовується модель лінійного згасання:

$$Rel(t) = Rel_0 - \alpha \cdot t.$$

Такий підхід має критичні недоліки: при досягненні порогового часу $t > Rel_0/\alpha$ релевантність стає від'ємною, що вимагає штучних обмежень. Крім того,

це суперечить закону Вебера-Фехнера та кривій забування Еббінгауза, згідно з якими втрата актуальності відбувається експоненціально. Лінійна модель надмірно штрафувє старий якісний контент і недостатньо швидко знижує пріоритет короткострокових трендів.

Нехтування нелінійною природою старіння інформації призводить до захаращення стрічки рекомендацій застарілим контентом, який формально має високі показники сумісності, але втратив свою актуальність.

Також критичним є статистична недостовірність рейтингів. Стандартний підхід до розрахунку рейтингу як середнього арифметичного є вразливим до проблеми малих вибірок. Вікторина з однією позитивною оцінкою (1 лайк, 0 дизлайків) отримує рейтинг 100%, що формально ставить її вище за перевірений хіт із тисячею оцінок і рейтингом 95%. Це явище призводить до нестабільності ранжування, коли випадковий новий контент витісняє якісні популярні роботи.

Традиційно рейтинг R_{score} обчислюється як:

$$R_{score} = \frac{N_{pos}}{N_{pos} + N_{neg}}$$

Недолік цієї моделі полягає у ігноруванні довірчого інтервалу, який залежить від розміру вибірки. Наприклад, об'єкт із 1 позитивною оцінкою ($R=1.0$) формально ранжується вище, ніж об'єкт із 95 позитивними та 5 негативними оцінками ($R=0.9$), хоча статистична значущість другого результату є незрівнянно вищою. Це вимагає застосування ймовірнісних підходів, таких як байєсівське згладжування.

Вирішення зазначених проблем вимагає відмови від простих лінійних алгоритмів на користь гібридних моделей, здатних інтегрувати різнорідні фактори та адаптивно обробляти невизначеність.

2.3 Узагальнена схема методу вирішення задачі

Для реалізації вдосконаленої рекомендаційної системи пропонується метод, що являє собою ітеративний процес перетворення "сирих" даних про взаємодію користувачів та метаданих контенту в ранжований список персональних пропозицій. Загальна схема алгоритму складається з п'яти послідовних етапів:

а) агрегація та побудова динамічного профілю користувача. На цьому етапі система збирає різноманітні сигнали про активність користувача: історію оцінок («лайки», «дизлайки»), завершені вікторини та об'єкти, додані до списку бажаного. На основі цих даних формується комплексний цифровий профіль, що містить зважену карту інтересів, історію взаємодії з авторами та вподобання щодо формату контенту (наприклад, середня довжина вікторини). Важливим аспектом є те, що профіль оновлюється в реальному часі або з мінімальною затримкою, забезпечуючи актуальність даних;

б) розрахунок вхідних метрик з використанням удосконалених математичних моделей. Для кожного кандидата-вікторини система обчислює вектор ознак. На відміну від класичних підходів, тут застосовуються спеціалізовані моделі для корекції викривлень:

1) для розрахунку рейтингу використовується Байєсівське усереднення, що дозволяє згладити оцінки нових вікторин і усунути статистичну недостовірність малих вибірок;

2) для визначення актуальності застосовується функція експоненціального згасання, яка забезпечує нелінійне зниження ваги контенту з часом, що відповідає психології сприйняття новизни;

3) розраховуються метрики подібності (схожість тегів, спорідненість з автором) для подолання бар'єру «холодного старту»;

в) нормалізація даних. Оскільки вхідні метрики мають різну розмірність (наприклад, кількість проходжень може вимірюватися десятками, а схожість тегів — відсотками), виконується їх приведення до уніфікованого діапазону. Це є необхідною умовою для коректної роботи наступного етапу — фазифікації;

г) нечітке логічне виведення. Це ядро методу гібридної рекомендації, де відбувається прийняття рішення про релевантність контенту. Процес включає:

1) фазифікацію – трансформацію чітких нормалізованих значень у лінгвістичні змінні (наприклад, «Популярність» «СЕРЕДНЯ», «ВИСОКА») за допомогою функцій належності;

2) агрегацію правил – застосування бази експертних правил типу «ЯКЩО-ТОДІ», які описують логіку рекомендації в умовах невизначеності та дозволяють вирішувати конфлікти між різними метриками (наприклад, пріоритет улюбленого автора над слабкою схожістю тегів);

д) дефазифікація та ранжування результатів. Фінальний етап, на якому отримана нечітка множина (результат виконання правил) перетворюється у чітке

число — рекомендаційний бал. Для цього використовується метод центру тяжіння. На основі отриманих балів формується впорядкований список вікторин, який передається на клієнтську частину системи для відображення користувачеві.

Така послідовність дій дозволяє створити адаптивну систему, яка ефективно працює як з новим, так і з популярним контентом, забезпечуючи високу якість та прозорість рекомендацій.

2.4 Удосконалена модель функціонування методу гібридної рекомендації

Першим і фундаментальним етапом роботи рекомендаційної системи є формування цифрового профілю користувача, який виступає контекстом для подальшого логічного виведення. Профіль не є статичним набором даних, а динамічно перераховується при кожному запиті рекомендацій, що забезпечує миттєву реакцію системи на нові дії користувача.

Процес побудови профілю декомпозується на чотири паралельні потоки обробки даних:

- моделювання семантичних інтересів;
- аналіз соціальної взаємодії;
- формування множини виключень;
- визначення структурних вподобань.

При моделюванні семантичних інтересів система визначає ступінь зацікавленості користувача певними тегами шляхом аналізу його явних дій. Для цього використовується зважена модель накопичення балів. Кожному типу взаємодії присвоюється певний ваговий коефіцієнт w_{action} , що відображає силу інтересу:

- позитивна оцінка («лайк»): $w = 3$ (найсильніший сигнал підтвердження інтересу);
- додавання до списку бажаного: $w = 2$ (сигнал відкладеного інтересу);
- негативна оцінка («дизлайк»): $w = -1$ (сигнал відторгнення).

Для кожного унікального тегу, що зустрічається у вікторинах, з якими взаємодівав користувач, обчислюється сумарна вага:

$$Score_t = \sum_{i=1}^N w_{action,i}$$

де N — кількість взаємодій з контентом, що містить тег t .

Після агрегації виконується процедура нормалізації. Значення ваг усіх тегів масштабуються до діапазону $[0, 100]$ відносно тегу з максимальним накопиченим балом. Це дозволяє нівелювати різницю між активними користувачами (з тисячами дій) та новачками, зберігаючи при цьому пропорції їхніх інтересів [10]. Теги з від'ємним сумарним балом виключаються з профілю як небажані.

Для аналізу взаємодії система відстежує лояльність користувача до конкретних авторів контенту. Формується мапа взаємодій, де ключем є ідентифікатор автора, а значенням — кількість позитивних контактів (лайків, збережень, проходжень) з його роботами. Цей показник є накопичувальним і використовується для виявлення фанатів певних крейторів, дозволяючи системі в подальшому рекомендувати їхні нові роботи незалежно від тематики.

Для забезпечення новизни рекомендацій система агрегує ідентифікатори вікторин, з якими користувач вже мав активну взаємодію. Створюються дві множини: завершені та оцінені вікторини. Ці дані використовуються на етапі попередньої фільтрації для виключення вже спожитого контенту зі списку кандидатів, що гарантує користувачеві постійний притік нової інформації.

Окрім семантики, система аналізує формат контенту, якому користувач надає перевагу. На основі історії завершених вікторин обчислюється середня довжина вікторини — середнє арифметичне кількості питань у вікторинах, які користувач пройшов до кінця:

$$L_{avg} = \frac{1}{K} \sum_{k=1}^K Q_{count,k}$$

де K — кількість завершених вікторин;

Q_{count} — кількість питань у k -й вікторині. Цей параметр дозволяє системі адаптуватися до терпіння та звичок користувача: тим, хто полюбляє швидкі сесії, будуть менше рекомендуватися довгі турніри на 64+ питання, і навпаки [11].

Результатом цього етапу є комплексний об'єкт профілю, що містить нормалізовану карту тегів, історію взаємодії з авторами, списки виключень та метрику бажаної довжини. Цей профіль передається на вхід наступного етапу для розрахунку метрик схожості з кандидатами.

Після формування комплексного профілю користувача метод гібридної рекомендації переходить до етапу оцінювання множини вікторин-кандидатів. На цьому кроці для кожної пари «користувач — вікторина» обчислюється вектор ознак, компоненти якого характеризують різні аспекти релевантності. До складу цього вектору входять шість ключових показників:

- оцінка семантичної подібності;
- рівень соціальної лояльності до автора;
- структурна відповідність довжини;
- загальна популярність;
- адаптивний рейтинг схвалення;
- часова актуальність.

Для забезпечення стійкості системи до проблем, описаних у постановці задачі, розрахунок цих метрик базується на спеціалізованих імовірнісних та функціональних моделях.

Першим кроком є визначення тематичної відповідності контенту інтересам користувача, що реалізується через оцінку семантичної подібності. Оскільки профіль користувача містить не просто перелік тегів, а їхні ваги, класичний коефіцієнт Жаккара було модифіковано. Ступінь подібності S_{tags} розраховується як відношення суми ваг тегів, що є спільними для користувача та вікторини, до загального розміру об'єднання множин тегів. Формально це записується як:

$$S_{tags}(u, q) = \frac{\sum_{t \in (T_u \cap T_q)} w_u(t)}{|T_u \cup T_q|},$$

де T_u та T_q — множини тегів користувача та вікторини відповідно;

$w_u(t)$ — нормалізована вага інтересу.

Такий підхід дозволяє виявити контент, який відповідає найбільш пріоритетним інтересам особи [12].

Поряд із тегами, важливим фактором є особистість творця контенту, тому система моделює соціальну лояльність через метрику спорідненості з автором. Цей

показник базується на історії попередніх позитивних взаємодій користувача з роботами конкретного автора. Величина спорідненості $Aff(u, c)$ визначається як функція насичення від кількості інтеракцій N_{int} за формулою:

$$Aff(u, c) = \min\left(1.0, \frac{N_{int} \cdot k}{100}\right),$$

де k — коефіцієнт цінності однієї взаємодії.

Використання функції мінімуму гарантує, що вплив цього фактору не перевищить визначеної верхньої межі, запобігаючи монополізації рекомендацій одним автором.

Для адаптації рекомендацій під часові можливості та звички користувача розраховується структурна відповідність. Вона оцінює відхилення довжини вікторини від «комфортного» для користувача значення за допомогою відносної метрики:

$$M_{len} = \frac{L_{quiz} - L_{avg}}{L_{avg}},$$

де L_{quiz} — кількість питань у кандидаті;

L_{avg} — середньозважена довжина вікторин, успішно завершених користувачем у минулому.

Отримане значення характеризує, наскільки вікторина довша або коротша за звичний формат.

Наступним компонентом вектору ознак є загальна популярність, яка слугує індикатором «соціального доказу». Вона розраховується на основі загальної кількості унікальних проходжень вікторини та дозволяє виявляти трендовий контент, що може бути цікавим широкому загалу незалежно від специфічних тематичних вподобань. Значення нормалізується відносно порогового показника «віральності».

Для вирішення проблеми статистичної недостовірності оцінок нового контенту застосовується адаптивний рейтинг схвалення, що обчислюється методом Байєсівського усереднення [13]. Замість звичайного середнього арифметичного використовується формула зважування:

$$R_{final} = \alpha \cdot R_{actual} + (1 - \alpha) \cdot R_{prior},$$

де R_{actual} — фактичний рейтинг;

R_{prior} — апріорна середня оцінка по системі;

α — коефіцієнт довіри, який залежить від кількості голосів v як $\alpha = \min(1.0, \frac{v}{N_{threshold}})$.

Такий підхід забезпечує плавний перехід від апріорної оцінки до фактичної міру накопичення даних, усуваючи різкі стрибки рейтингу для нових вікторин.

Завершує формування вектору ознак метрика часової актуальності, яка необхідна для коректного моделювання життєвого циклу контенту. Для вирішення проблеми лінійного старіння використано модель експоненціального згасання, що базується на психофізичних закономірностях сприйняття новизни [14]. Коефіцієнт актуальності T_{score} визначається як:

$$T_{score} = 1.0 - e^{-\frac{\Delta t}{\tau}},$$

де Δt — час, що пройшов з моменту публікації;

τ — константа часу.

Використання експоненти дозволяє системі чітко диференціювати «свіжий» контент, водночас нівелюючи різницю в актуальності для архівного матеріалу [15].

Критичною передумовою для коректної роботи системи нечіткого виведення є узгодженість вхідних даних. Оскільки компоненти сформованого на попередньому етапі вектору ознак мають різну фізичну природу та діапазони значень, безпосереднє їх використання в єдиній базі правил є неможливим. Тому на третьому етапі виконується процедура нормалізації, метою якої є відображення всіх вхідних змінних у єдиний уніфікований інтервал $[0, 1]$ [16].

Для метрик, що характеризують відхилення від оптимуму та можуть набувати як додатних, так і від'ємних значень, зокрема для показника структурної відповідності довжини, використовується функція лінійного перетворення. В загальному вигляді вона описується формулою:

$$x'_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}.$$

Оскільки дана метрика визначена на інтервалі $[-1, 1]$, де значення 0 відповідає ідеальному збігу, а краї інтервалу — максимальному відхиленню, формула трансформується у вигляд:

$$x'_{norm} = \frac{x + 1}{2}.$$

Таким чином, значення -1 (вікторина занадто коротка) відображається в 0, +1 (занадто довга) — в 1, а ідеал 0 стає 0.5, що дозволяє нечіткому контролеру коректно інтерпретувати ступінь відхилення.

Для метрик, що не мають чітко визначеної верхньої межі, але характеризують інтенсивність явища, насамперед для загальної популярності вікторини, застосовується нормалізація відносно емпіричного порогу «насичення». Визначається контрольне значення $N_{threshold}$ (кількість проходжень, достатня для визнання контенту «вірусним»), і значення метрики нормалізується за формулою:

$$x'_{norm} = \min\left(1.0, \frac{x_{raw}}{N_{threshold}}\right).$$

Це гарантує, що будь-яке значення, яке перевищує порогове, буде інтерпретоване системою як максимальне (1.0), запобігаючи спотворенню результатів через аномально високі показники окремих об'єктів.

Для метрик, що за своєю природою є відсотковими частками, таких як схожість тегів, адаптивний рейтинг схвалення та спорідненість з автором, виконується просте ділення на масштабний коефіцієнт згідно з формулою:

$$x'_{norm} = \frac{x_{\%}}{100}.$$

Метрика часової актуальності, яка вже розраховується в діапазоні $[0, 1]$ функцією експоненціального згасання, не потребує додаткових перетворень. Результатом цього етапу є нормалізований вектор ознак $X' = \{x'_1, \dots, x'_6\}$, де кожен елемент $x'_i \in [0, 1]$, який подається на вхід фазифікатора.

Отриманий на попередньому етапі вектор нормалізованих даних стає вхідним сигналом для ядра методу — підсистеми нечіткого виведення. На відміну від жорсткої логіки, де рішення приймаються бінарно, цей етап імітує людський процес міркування, оперуючи ступенями істинності. Процес реалізується за алгоритмом Мамдані та розпочинається з фазифікації, під час якої контролер трансформує кожне чітке значення з діапазону $[0, 1]$ у набір лінгвістичних змінних. Для цього простір значень кожної метрики розбивається на терми, що описуються функціями належності. Для опису граничних станів, що охоплюють краї діапазону, використовуються трапецієподібні функції, а для проміжних станів — трикутні функції з перекриттям, що забезпечує плавність семантичних переходів.

Конфігурація лінгвістичних змінних системи є наступною. Кожна вхідна метрика приймає наступні терми та функції приналежності:

а) схожість тегів:

1) NO_MATCH — трапецієподібна функція належності з точками (0.0, 0.0, 0.10, 0.20);

2) WEAK_MATCH — трикутна функція належності з точками (0.10, 0.25, 0.40);

3) MODERATE_MATCH — трикутна функція належності з точками (0.35, 0.55, 0.70);

4) STRONG_MATCH — трапецієподібна функція належності з точками (0.65, 0.85, 1.0, 1.0);

б) популярність вікторини:

1) UNPOPULAR — трапецієподібна функція належності з точками (0.0, 0.0, 0.025, 0.05);

2) MODERATE — трикутна функція належності з точками (0.04, 0.15, 0.25);

3) POPULAR — трикутна функція належності з точками (0.20, 0.475, 0.75);

4) VIRAL — трапецієподібна функція належності з точками (0.50, 0.75, 1.0, 1.0);

в) рейтинг схвалення:

1) DISLIKED — трапецієподібна функція належності з точками (0.0, 0.0, 0.20, 0.35);

2) NEUTRAL — трикутна функція належності з точками (0.25, 0.50, 0.70);

- 3) LIKED — трикутна функція належності з точками (0.60, 0.75, 0.90);
- 4) LOVED — трапецієподібна функція належності з точками (0.85, 0.95, 1.0, 1.0);

г) часова актуальність:

- 1) NEW — трапецієподібна функція належності з точками (0.0, 0.0, 0.15, 0.30);
- 2) RECENT — трикутна функція належності з точками (0.20, 0.50, 0.75);
- 3) OLD — трикутна функція належності з точками (0.65, 0.85, 0.95);
- 4) OUTDATED — трапецієподібна функція належності з точками (0.90, 0.97, 1.0, 1.0);

д) спорідненість з автором:

- 1) NO_HISTORY — трапецієподібна функція належності з точками (0.0, 0.0, 0.05, 0.15);
- 2) SOME_INTERACTION — трикутна функція належності з точками (0.10, 0.35, 0.55);
- 3) REGULAR_USER — трикутна функція належності з точками (0.45, 0.70, 0.90);
- 4) FAN — трапецієподібна функція належності з точками (0.80, 0.92, 1.0, 1.0);

е) відповідність довжини:

- 1) TOO_SHORT — трапецієподібна функція належності з точками (0.0, 0.0, 0.20, 0.35);
- 2) PERFECT_MATCH — трикутна функція належності з точками (0.30, 0.50, 0.70);
- 3) TOO_LONG — трапецієподібна функція належності з точками (0.65, 0.80, 1.0, 1.0).

На рисунках 2.1-2.6 відповідно наведено функції приналежності для схожості тегів, популярності вікторин, рейтингу схвалення, часової актуальності, спорідненості з автором та відповідності довжини.

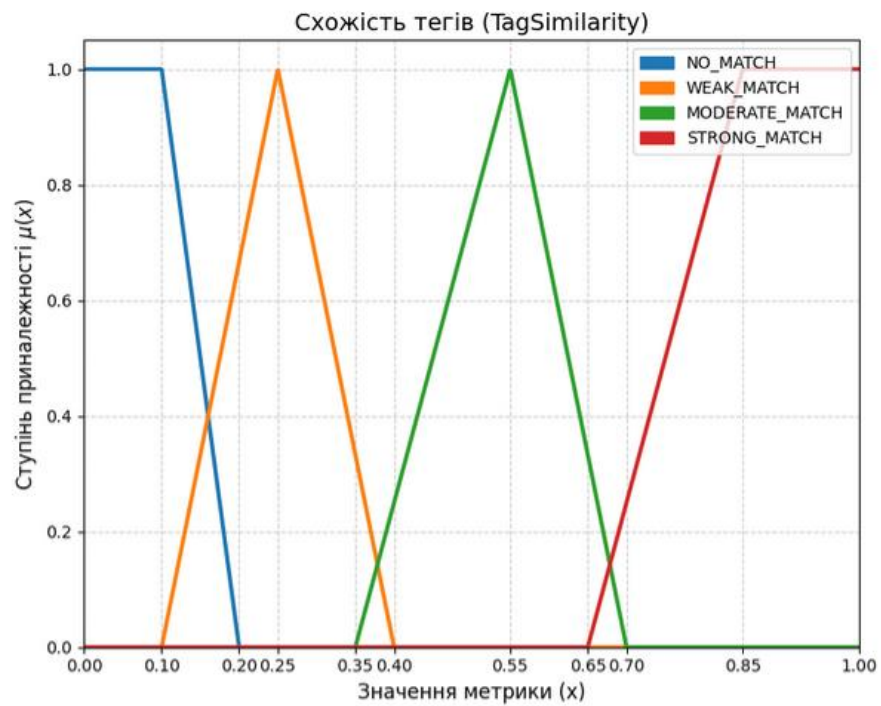


Рисунок 2.1 – Функції приналежності схожості тегів

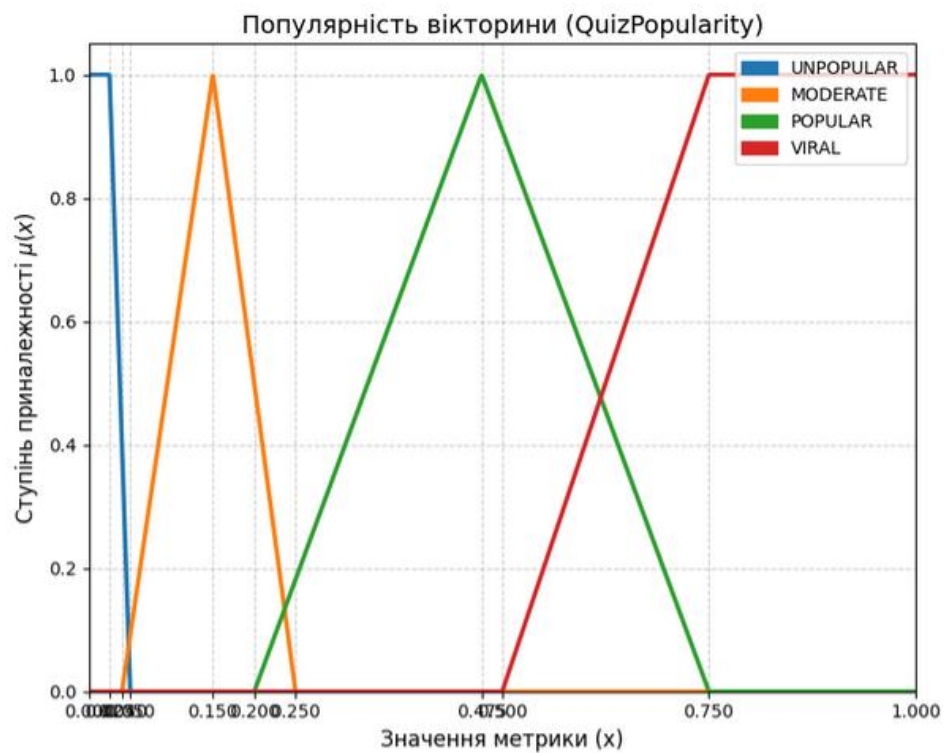


Рисунок 2.2 – Функції приналежності популярності вікторин

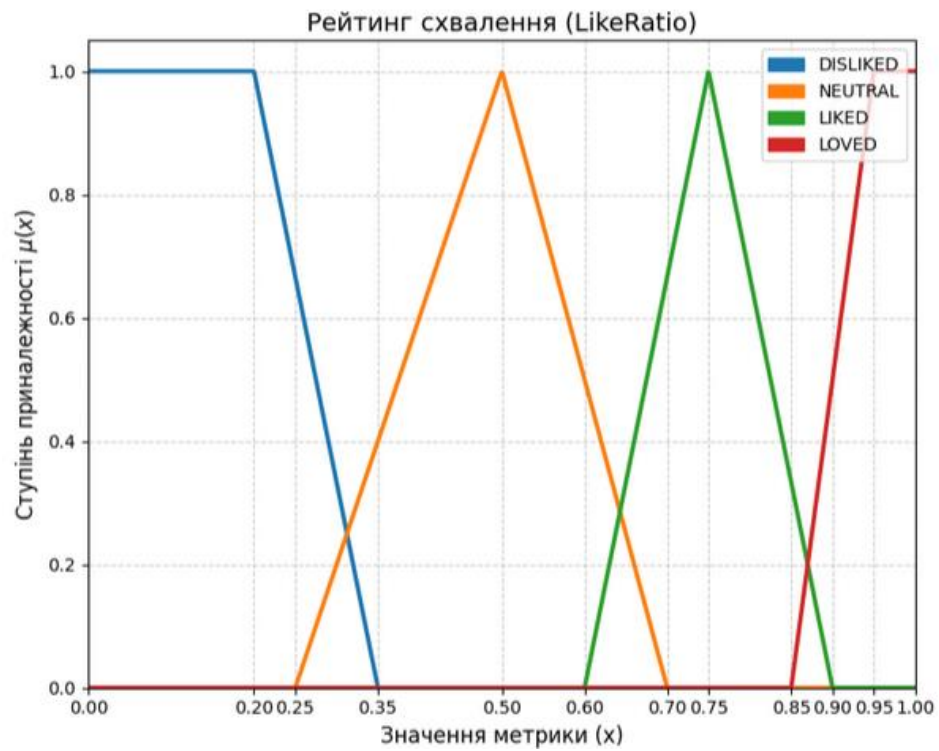


Рисунок 2.3 – Функції приналежності рейтингу схвалення

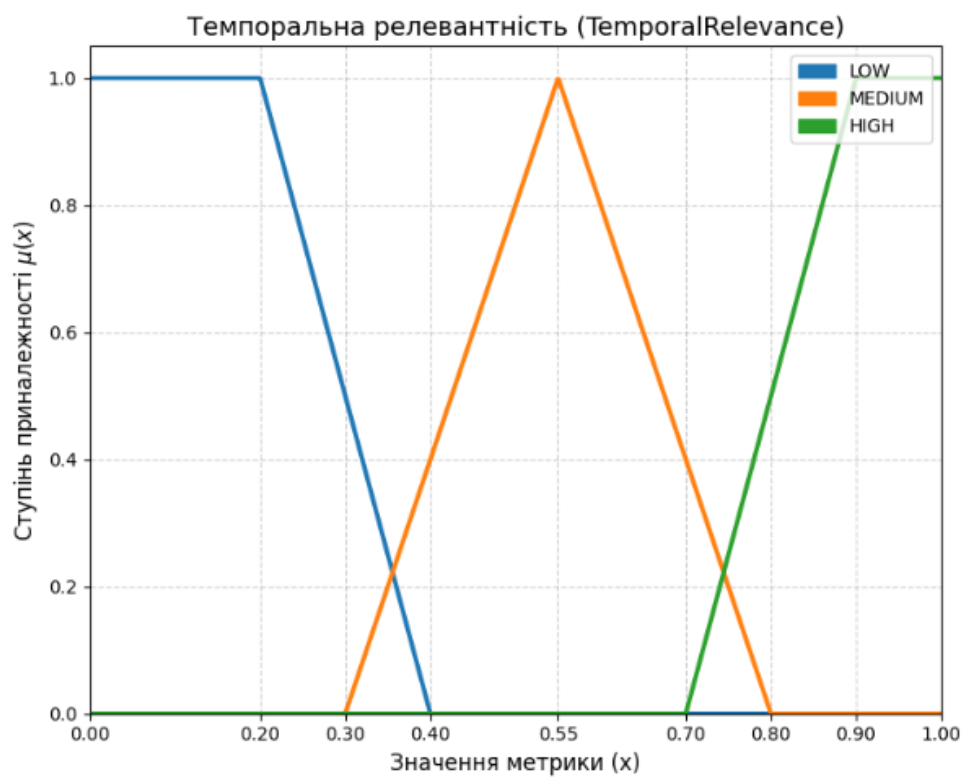


Рисунок 2.4 – Функції приналежності часової актуальності

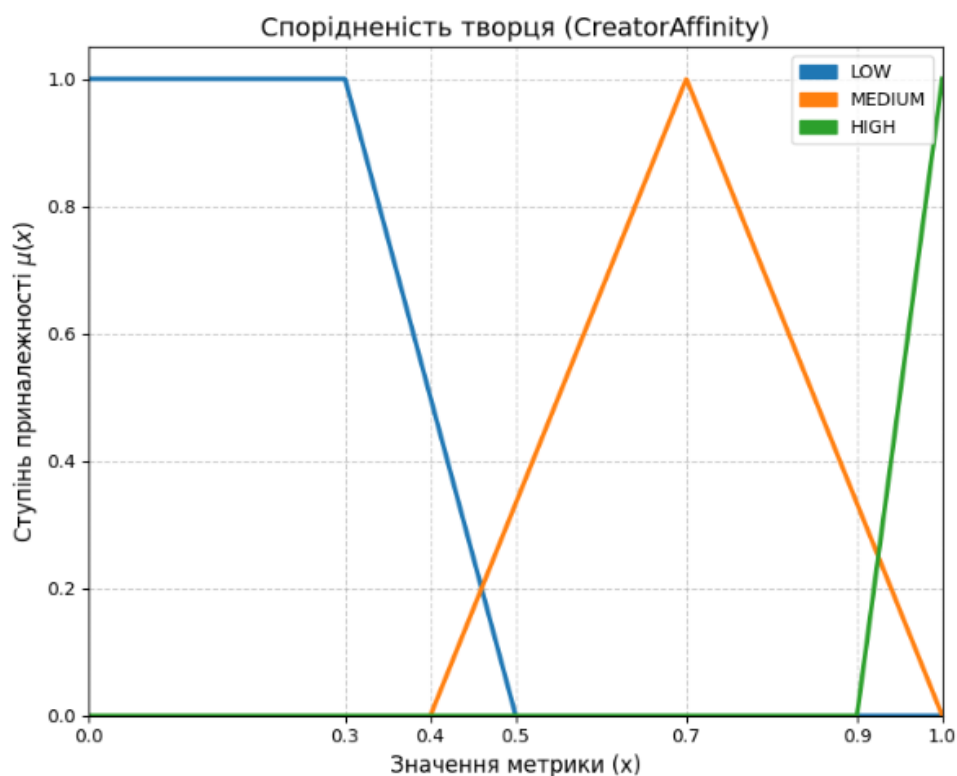


Рисунок 2.5 – Функції приналежності спорідненості з автором

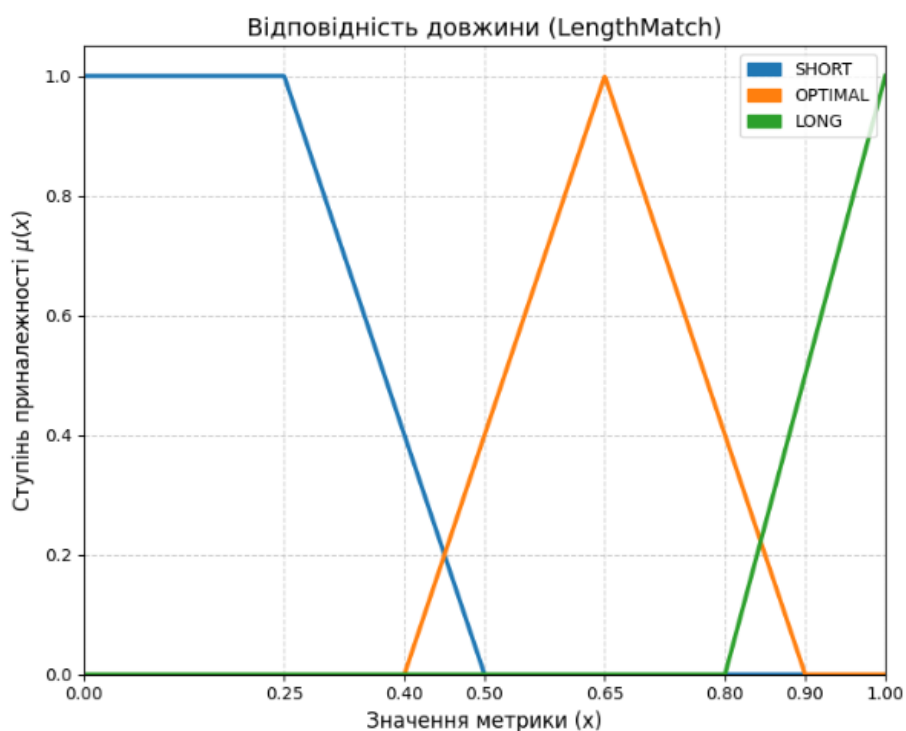


Рисунок 2.6 – Функції приналежності відповідності довжини

Після фазифікації відбувається оцінка правил, де активується база знань із 30 експертних правил. Система підтримує складні логічні конструкції, використовуючи оператори кон'юнкції (AND) та диз'юнкції (OR). Для правил, що вимагають одночасного виконання умов (наприклад, «сильна схожість тегів ТА

високий рейтинг»), застосовується оператор мінімуму, який визначає ступінь активації правила як найменше зі значень істинності передумов. Для правил з альтернативними умовами (наприклад, «занадто коротка АБО занадто довга») використовується оператор максимуму. Отриманий рівень активації застосовується до функції належності вихідного терму шляхом її «зрізання» (метод імплікації).

Завершальним етапом методу гібридної рекомендації є процедура дефазифікації та формування фінального переліку рекомендацій. Метою цього етапу є трансформація результуючої нечіткої множини, отриманої після агрегації правил, у єдине чітке числове значення, яке можна використати для порівняння та впорядкування об'єктів. Для отримання скалярної оцінки релевантності застосовується метод центру тяжіння, який є найбільш поширеним та об'єктивним методом у системах нечіткого виведення. Геометрична інтерпретація методу полягає у знаходженні абсциси центру мас фігури, обмеженої графіком функції належності результуючої нечіткої множини $\mu_{res}(y)$. У теоретичному вигляді для неперервної функції це описується відношенням визначених інтегралів:

$$y^* = \frac{\int_{Y_{min}}^{Y_{max}} y \cdot \mu_{res}(y) dy}{\int_{Y_{min}}^{Y_{max}} \mu_{res}(y) dy},$$

де y^* — шукане чітке значення;

Y_{min}, Y_{max} — відповідно нижче та верхнє значення інтервалу $[0, 100]$;

$\mu_{res}(y)$ — результуюча функція належності.

Чисельник формули являє собою момент першого порядку площі під кривою відносно початку координат, а знаменник — загальну площу під кривою.

Оскільки функція $\mu_{res}(y)$ має складну кусково-лінійну форму, що є результатом об'єднання багатьох усічених трикутників і трапецій, аналітичне обчислення інтегралу є ресурсомістким. Тому в рамках розробленого методу застосовується метод чисельного інтегрування шляхом дискретизації області визначення. Інтервал вихідних значень розбивається на N дискретних точок, і формула трансформується у дискретну суму:

$$y^* \approx \frac{\sum_{i=0}^N y_i \cdot \mu_{res}(y_i)}{\sum_{i=0}^N \mu_{res}(y_i)}.$$

Для забезпечення балансу між точністю та швидкістю системи використовується адаптивна кількість кроків дискретизації, що дозволяє мінімізувати похибку апроксимації. Система також передбачає обробку граничних випадків: якщо сума ступенів належності в знаменнику наближається до нуля (ситуація, коли жодне правило не спрацювало), системі присвоюється нейтральна оцінка для уникнення виключення потенційно придатного контенту.

На фінальному кроці отримане значення y^* інтерпретується як кількісна міра релевантності вікторини для конкретного користувача. Система формує впорядкований список пар «об'єкт — оцінка»:

$$L = \{(q_1, s_1), (q_2, s_2), \dots, (q_M, s_M)\},$$

де s_i — дефазифікована оцінка i -ї вікторини.

Цей список сортується за спаданням оцінки ($s_i \geq s_{i+1}$), після чого виконується відсікання для надання користувачу лише найбільш релевантних результатів, які передаються на рівень представлення для візуалізації.

3 ПРОЄКТУВАННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ НА ОСНОВІ МЕТОДУ ГІБРИДНОЇ РЕКОМЕНДАЦІЇ

3.1 Опис методу гібридної рекомендації

Розроблений метод гібридної рекомендації реалізується як автоматизований конвеєр обробки даних, що трансформує історію взаємодій користувача у впорядкований список релевантного контенту. На відміну від теоретичної моделі, технологічний процес враховує архітектурні особливості системи та виконується у такій послідовності:

- ідентифікація користувача та завантаження його історії (оцінки, перегляди, список бажаного) паралельно з метаданими активних вікторин;
- виконання зважування дій користувача для формування динамічного вектора інтересів та розрахунок статистики взаємодії з авторами;
- розрахунок набору із шести показників для кожного кандидата-вікторини. На цьому етапі застосовуються коригувальні математичні моделі (Байєсівське згладжування, експоненціальне згасання) для нівелювання похибок вхідних даних;
- приведення усіх розрахованих значень до єдиного діапазону для забезпечення сумісності з нечітким контролером;
- передача нормалізованих векторів в ядро нечіткого виведення, де на основі бази правил формується єдиний інтегральний показник релевантності;
- сортування отриманого списку з виключенням вже спожитих одиниць контенту, і формування фінального результату.

Для визначення функціональних вимог до розроблюваної гібридної рекомендаційної системи була розроблена модель процесів за стандартом IDEF0 [17]. Побудова моделі здійснювалася за допомогою стратегії покрокового уточнення методу спадного проєктування, що дозволило деталізувати логіку роботи системи від загальної взаємодії до конкретних алгоритмічних кроків.

На рисунку 3.1 представлена контекстна діаграма моделі інформаційної системи.

Контекстна діаграма має рівень A-0, яка в подальшому декомпозується до діаграми рівня A0. Головною метою системи є формування впорядкованого списку вікторин для користувача на основі вхідних даних: пройдених вікторин, оцінок, списку бажаного, а також статистики та метаданих вікторин. Управління процесом

здійснюється на основі математичних моделей, порогових констант, бази знань та параметрів запити, а механізмами виступають програмні компоненти сервісної архітектури.

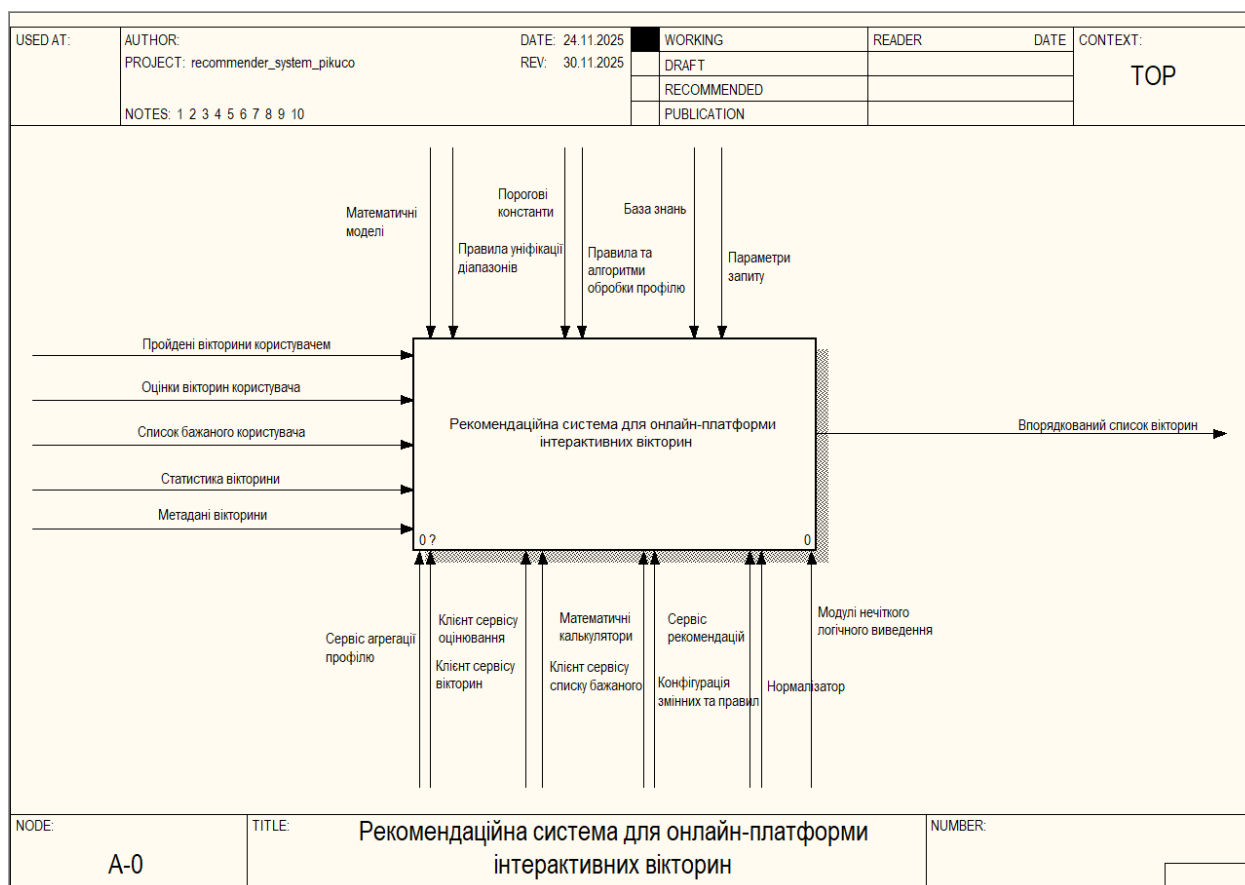


Рисунок 3.1 – Контекстна діаграма

На рисунку 3.2 представлена діаграма декомпозиції рівня А0.

Декомпозиція контекстної діаграми складається з таких процесів:

- формування динамічного профілю користувача;
- розрахунок та нормалізація вхідних метрик;
- виконання нечіткого логічного виведення;
- ранжування та відбір рекомендацій.

Функціональний блок «Формування динамічного профілю користувача» є першим етапом обробки даних, в якому відбувається агрегація історії взаємодій користувача та перетворення її в структурований профіль інтересів.

Процес «Розрахунок та нормалізація вхідних метрик» відповідає за математичну обробку даних кандидатів-вікторин. На цьому етапі для кожного

кандидата обчислюється набір показників релевантності, які потім нормалізуються до уніфікованого діапазону.

Процес «Виконання нечіткого логічного виведення» є ядром інтелектуальної системи. Він приймає нормалізовані метрики та, використовуючи базу нечітких правил, формує єдину дефазифіковану оцінку релевантності.

Процес «Ранжування та відбір рекомендацій» завершує роботу системи, сортуючи вікторини за отриманою оцінкою та формуючи фінальний список рекомендацій згідно з параметрами запити.

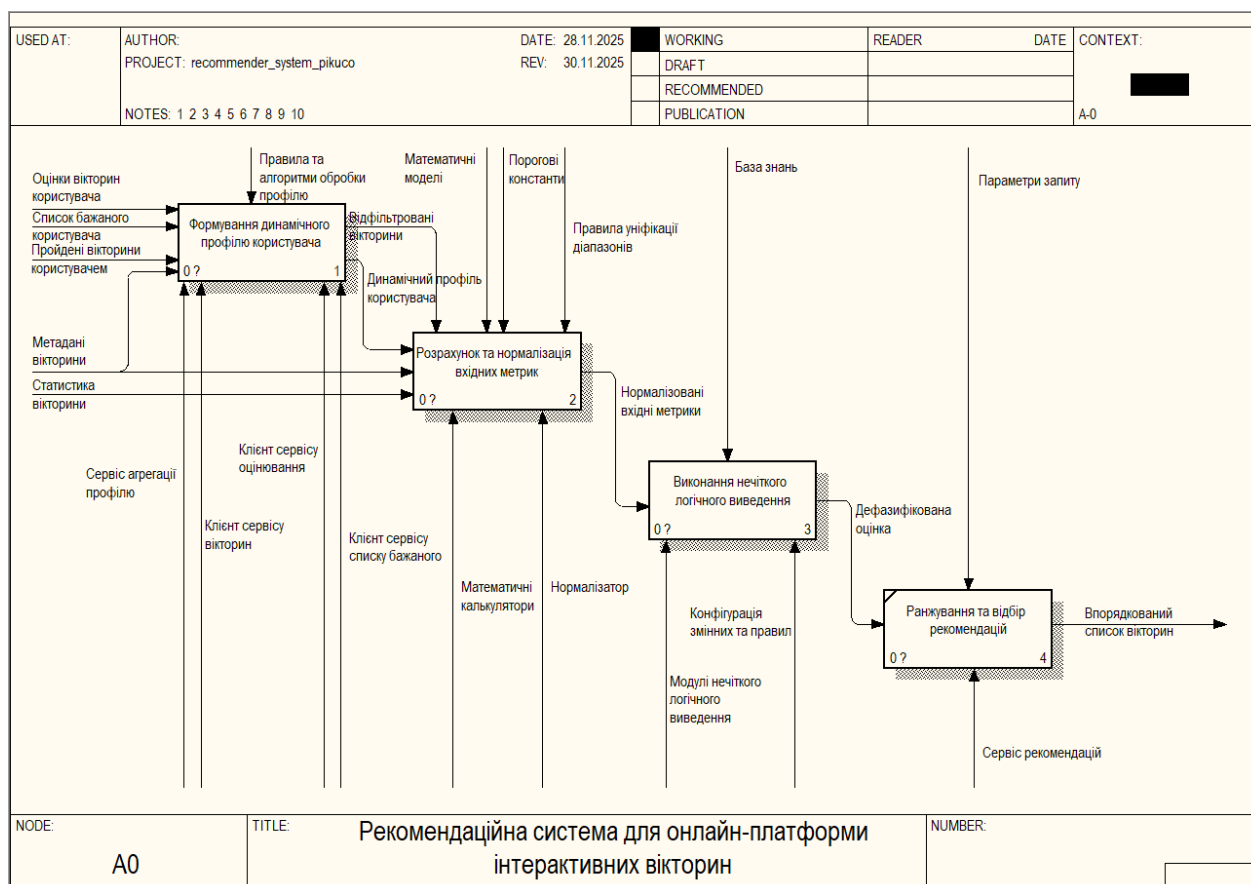


Рисунок 3.2 – Рекомендаційна система для онлайн-платформи інтерактивних вікторин

Декомпозуємо процеси, які необхідно уточнити.

На рисунку 3.3 представлена діаграма декомпозиції процесу «Формування динамічного профілю користувача».

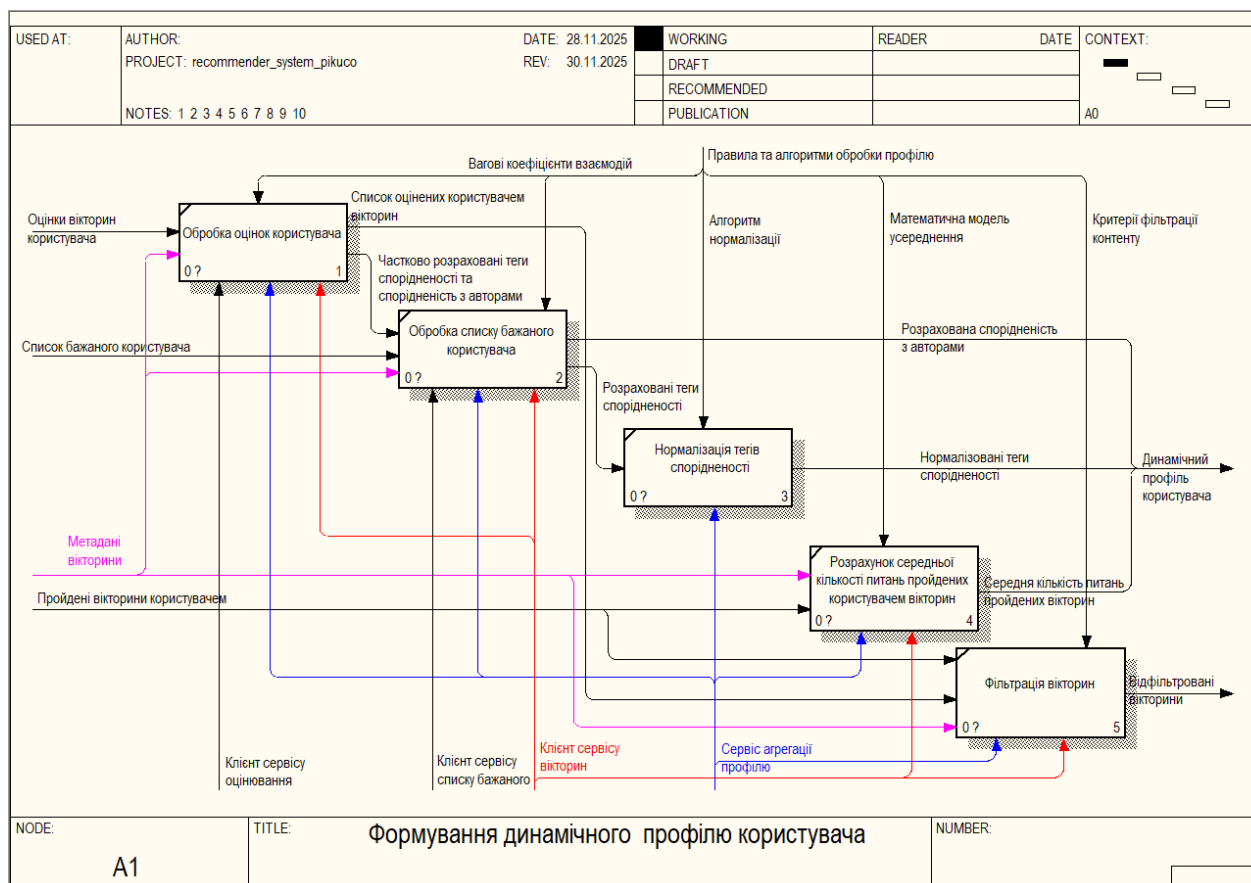


Рисунок 3.3 – Формування динамічного профілю користувача

Даний процес складається з наступних підпроцесів:

- обробка оцінок користувача – відповідає за аналіз явних оцінок (лайк, дизлайк) та отримання списку оцінених вікторин;
- обробка списку бажаного користувача – аналізує відкладений список вікторин користувача для уточнення його інтересів;
- нормалізація тегів спорідненості – здійснює розрахунок та зважування інтересу користувача до певних категорій (тегів) на основі його дій;
- розрахунок середньої кількості питань пройдених користувачем вікторин
- формує метрику структурних вподобань користувача щодо довжини контенту;
- фільтрація вікторин – виключає з переліку кандидатів вже пройдені, оцінені вікторини або ще не готові до публікації (чернетки) вікторини.

На рисунку 3.4 представлена діаграма декомпозиції процесу «Розрахунок та нормалізація вхідних метрик».

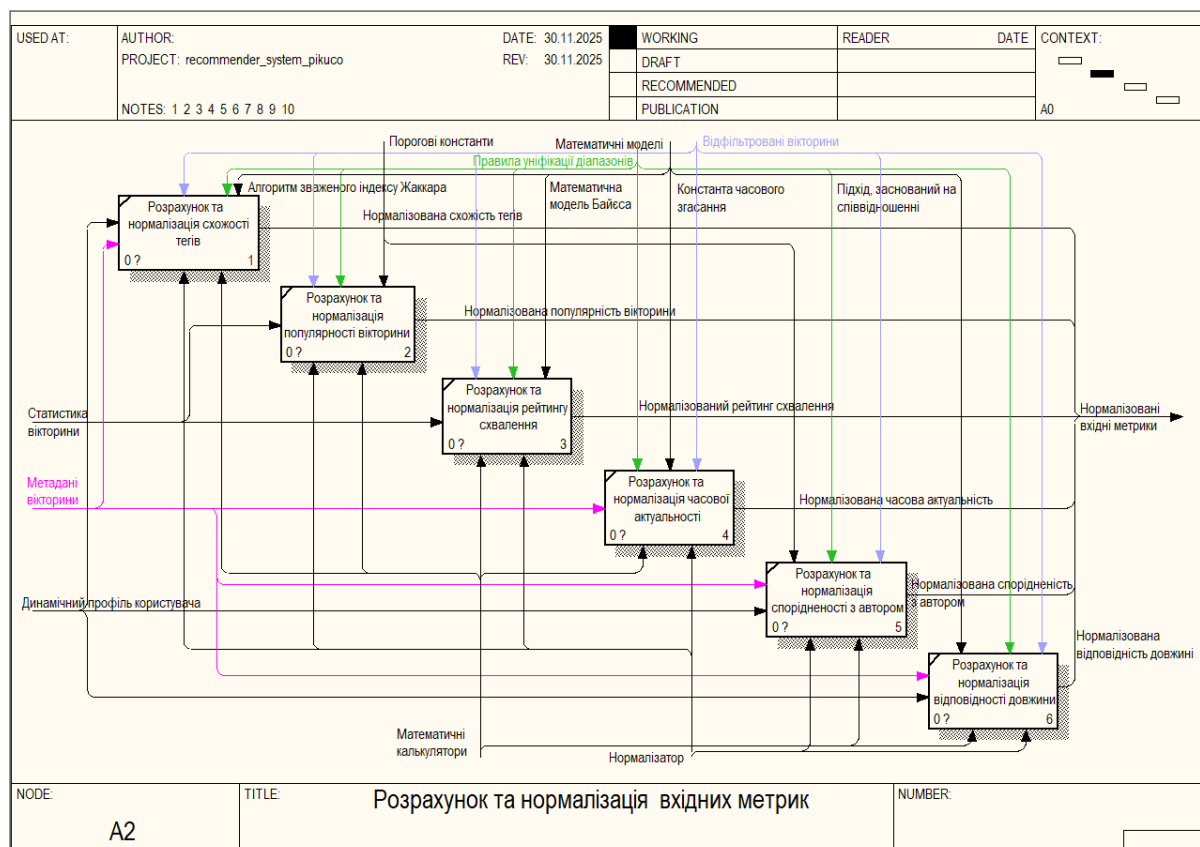


Рисунок 3.4 – Розрахунок та нормалізація вхідних метрик

Цей процес деталізується через шість паралельних підпроцесів розрахунку окремих факторів:

- розрахунок та нормалізація схожості тегів – визначає семантичну близькість контенту до інтересів користувача;
- розрахунок та нормалізація популярності вікторини – оцінює загальний попит на вікторину серед аудиторії;
- розрахунок та нормалізація рейтингу схвалення – використовує Байєсівське усереднення для оцінки якості контенту;
- розрахунок та нормалізація часової актуальності – визначає «свіжість» контенту за допомогою моделі експоненціального згасання;
- розрахунок та нормалізація спорідненості з автором – оцінює лояльність користувача до творця вікторини;
- розрахунок та нормалізація відповідності довжини – визначає, наскільки формат вікторини відповідає звичкам користувача.

На рисунку 3.5 представлена діаграма декомпозиції процесу «Виконання нечіткого логічного виведення».

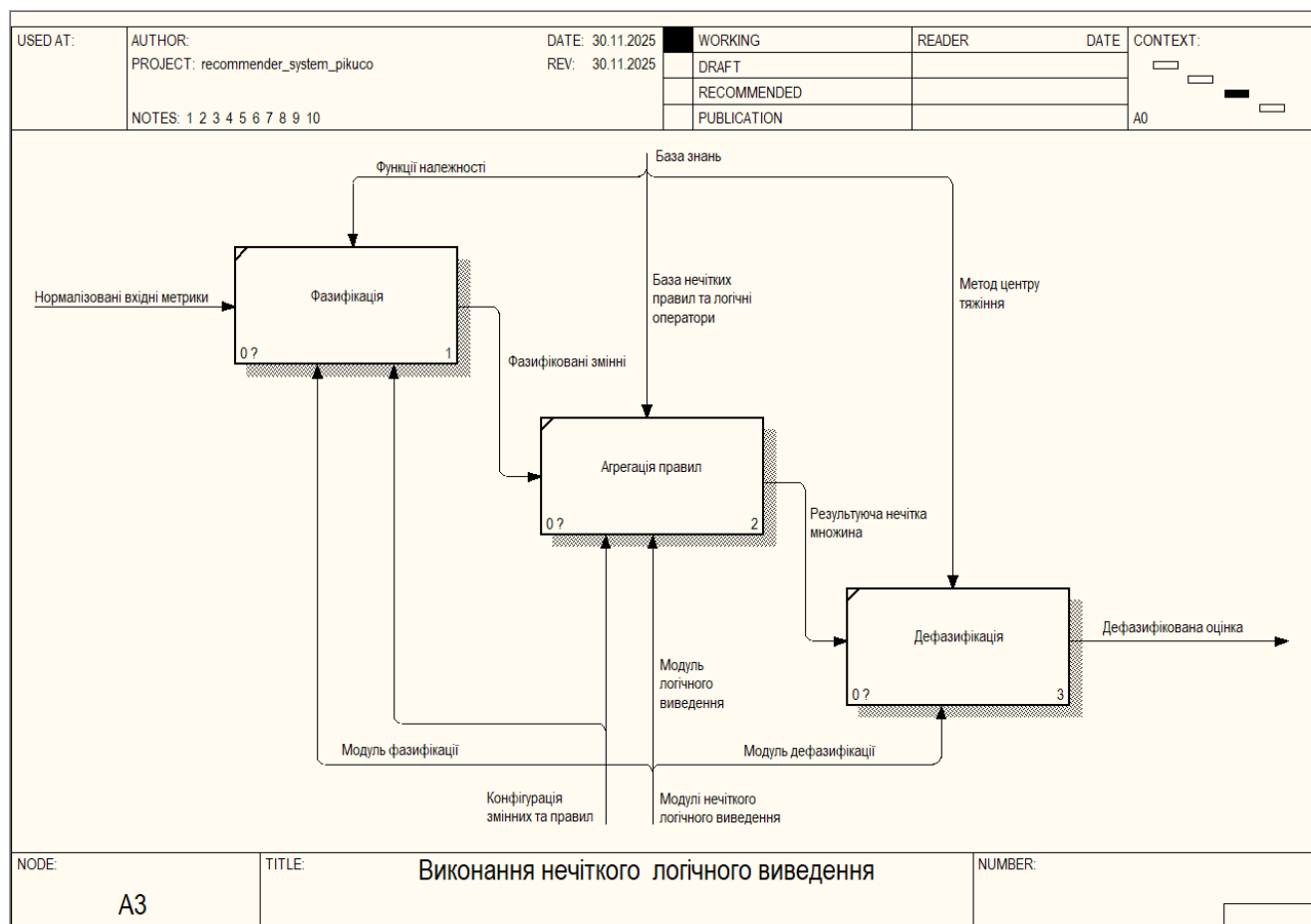


Рисунок 3.5 – Виконання нечіткого логічного виведення

Процес «Виконання нечіткого логічного виведення» складається з наступних пунктів:

- фазифікація – перетворює чіткі числові значення метрик у лінгвістичні змінні за допомогою функцій належності;
- агрегація правил – застосовує базу продукційних правил та логічні оператори для визначення ступеня істинності висновків;
- дефазифікація – трансформує результуючу нечітку множину в чітке числове значення (рекомендаційний бал) методом центру тяжіння.

Використовуючи функціонал ERwin Process Modeler, було згенеровано дерево вузлів на основі побудованої моделі процесів. На рисунку 3.6 наведена діаграма дерева вузлів.

Згідно наведеного дерева вузлів, основний процес «Рекомендаційна система для онлайн-платформи інтерактивних вікторин» складається з чотирьох підпроцесів, які були декомпозовані для детального відображення логіки роботи системи. Процес «Формування динамічного профілю користувача»

декомпозиується на 5 функціональних блоків, що відповідають за підготовку даних про користувача. Процес «Розрахунок та нормалізація вхідних метрик» містить 6 блоків розрахунку факторів релевантності. Процес «Виконання нечіткого логічного виведення» деталізується через 3 етапи алгоритму нечіткої логіки.

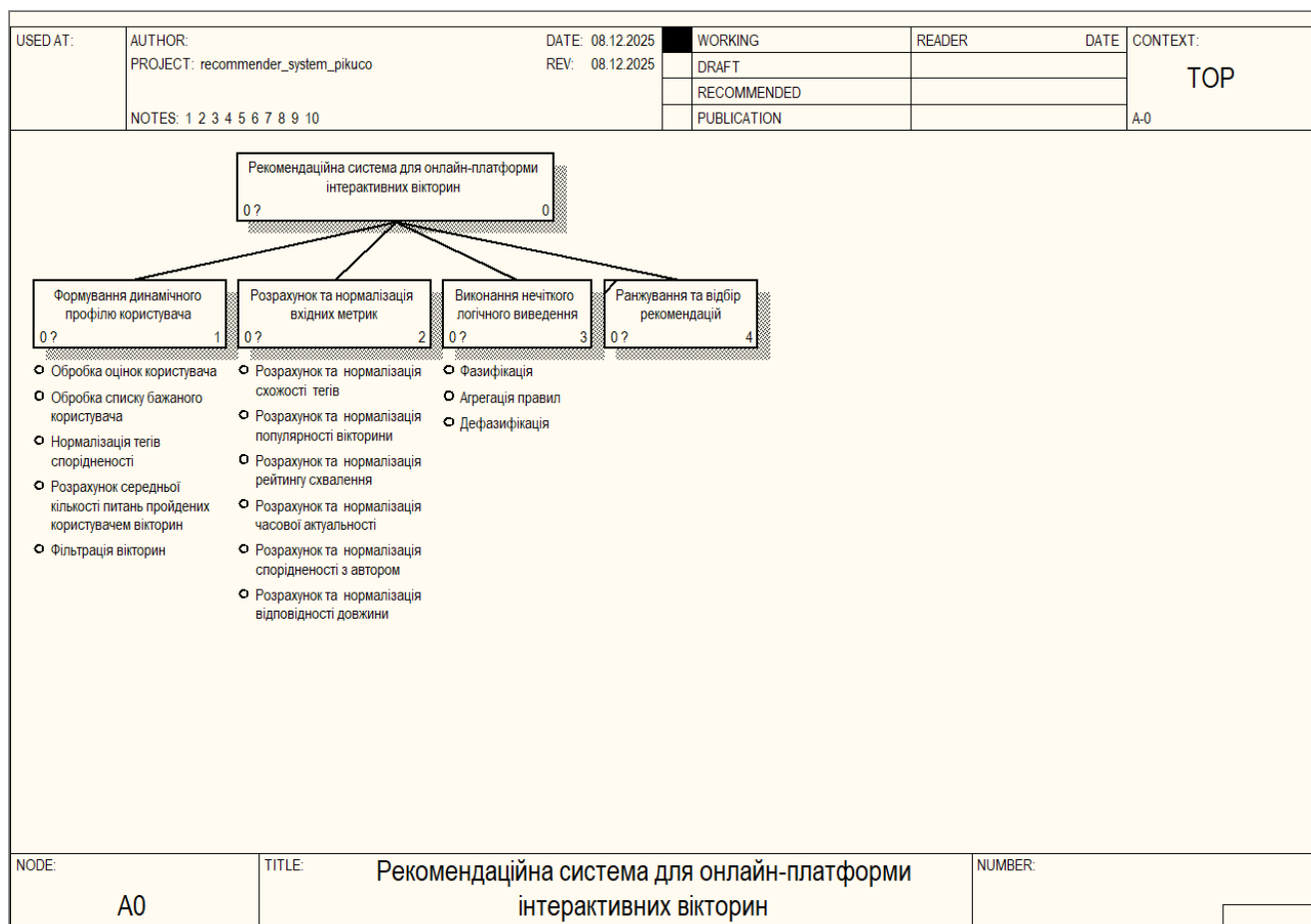


Рисунок 3.6 – Діаграма дерева вузлів

Для розуміння динаміки роботи розробленого методу та візуалізації потоків даних, що циркулюють між користувачем та системою, було виконано побудову діаграми активності [18]. Вона повною мірою описує алгоритмічну логіку рекомендаційного конвеєра, демонструючи послідовність етапів обробки запиту, розгалуження процесів на паралельні потоки обчислень та точки синхронізації результатів.

На рисунку 3.7 наведена діаграма активності алгоритму роботи гібридної рекомендаційної системи.

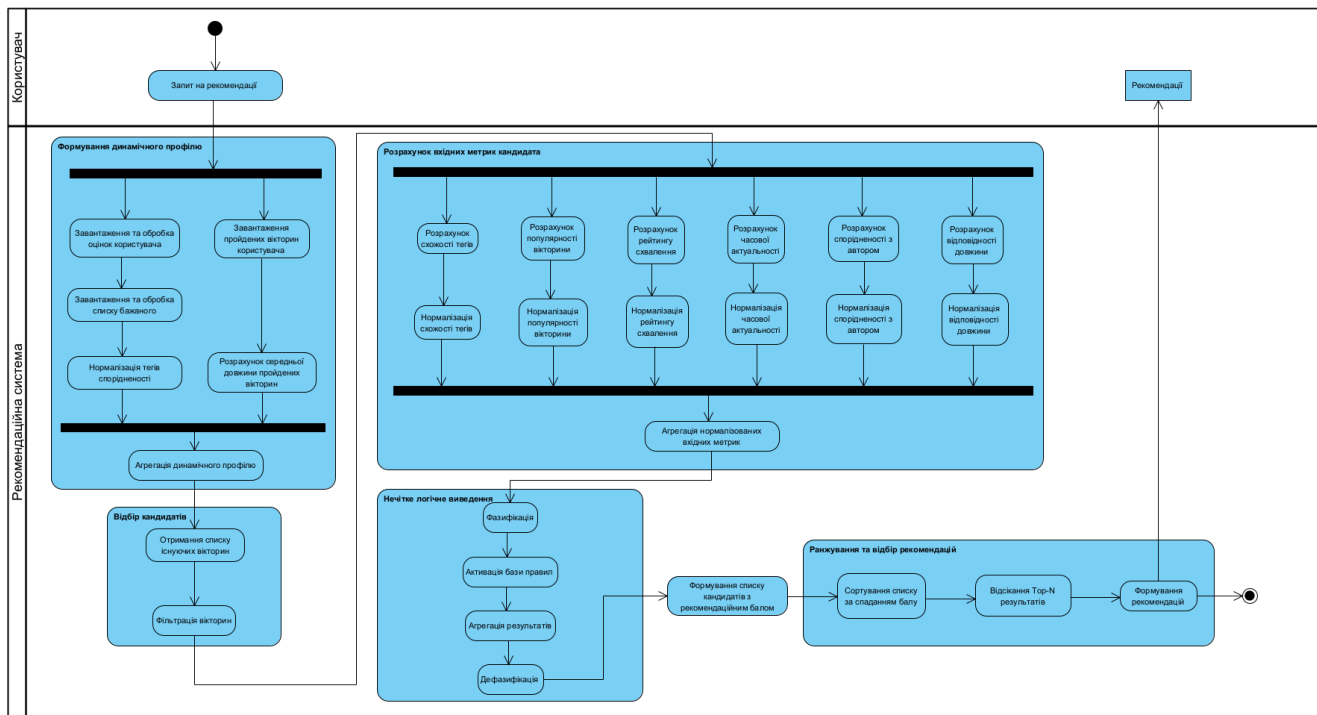


Рисунок 3.7 – Діаграма активності

Процес ініціюється запитом користувача на отримання рекомендацій. Після цього система переходить до етапу «Формування динамічного профілю», де відбувається паралельне завантаження та обробка історичних даних: оцінок, списку бажаного та пройдених вікторин. Це дозволяє одночасно розрахувати нормалізовані теги спорідненості та середню довжину вікторин, які згодом агрегуються в єдиний профіль користувача.

Після відбору кандидатів з бази даних та їх попередньої фільтрації, алгоритм переходить до найбільш ресурсомісткого етапу — «Розрахунок вхідних метрик кандидата». Як видно з діаграми, обробка шести ключових факторів (схожість тегів, популярність, рейтинг схвалення, часова актуальність, спорідненість з автором та відповідність довжини) виконується незалежно один від одного. Кожна метрика проходить власний шлях розрахунку та нормалізації, після чого всі результати синхронізуються та агрегуються для передачі на наступний рівень.

Завершальна фаза обробки відбувається в блоці «Нечітке логічне виведення», що реалізує послідовність операцій алгоритму Мамдані: фазифікацію вхідних даних, активацію бази правил, агрегацію результатів та дефазифікацію для отримання чіткого рекомендаційного балу. На фінальному етапі «Ранжування та відбір рекомендацій» сформований список кандидатів сортується за спаданням

балу, відбувається відсікання Top-N найкращих результатів, які у форматі JSON-відповіді повертаються користувачу.

3.2 Опис архітектури розробленої системи

Метод гібридної рекомендації реалізується у вигляді окремого незалежного сервісу, оскільки інформаційна система «Онлайн-платформа інтерактивних вікторин» побудована на принципах сервісної архітектури [19]. Такий підхід забезпечує слабку зв'язність компонентів та можливість незалежного масштабування.

З огляду на необхідність інтеграції в існуючу екосистему, технологічний стек реалізації співпадає з інструментарієм основної платформи:

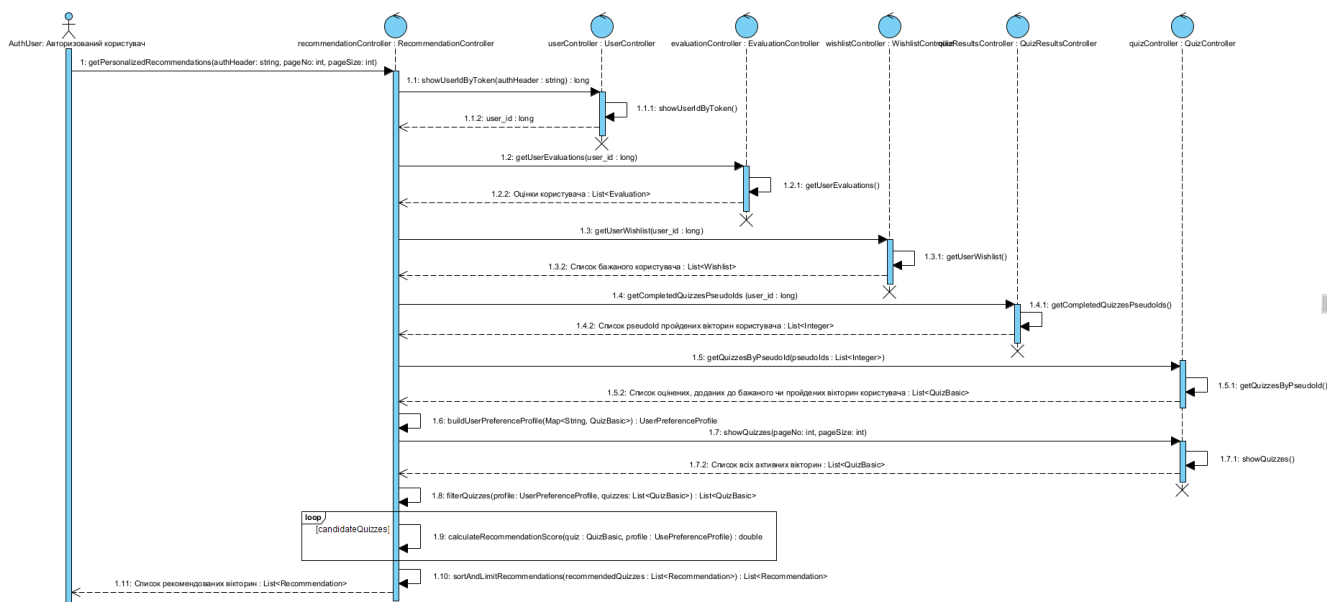
- серверна частина розроблена мовою програмування Java 17 [20] з використанням фреймворку Spring Boot 3 [21], що забезпечує високу продуктивність обробки даних та зручність створення REST API; збірка проєкту здійснюється за допомогою Maven;

- клієнтська частина для відображення списку рекомендацій реалізована на мові програмування JavaScript [22] з використанням бібліотеки React та менеджера стану Redux; взаємодія з сервером та кешування запитів забезпечується інструментом RTK Query [23]; візуальна складова базується на мові розмітки HTML5 та каскадних таблицях CSS [24].

Для зберігання попередньо агрегованої статистики обрано документо-орієнтовану СУБД MongoDB [25]. Використання окремої бази даних для рекомендаційного сервісу обумовлено необхідністю оптимізації продуктивності. Вибір NoSQL рішення аргументований необхідністю роботи з напівструктурованими даними та вимогою до високої швидкості операцій читання, що є пріоритетним для рекомендаційних систем порівняно з особливостями реляційних баз даних.

Для розуміння логіки взаємодії компонентів системи при формуванні персоналізованих рекомендацій розробили діаграму послідовності [26] для процесу отримання рекомендацій.

На рисунку 3.8 наведено діаграму послідовності для процесу отримання списку рекомендацій.



Рисунк 3.8 – Діаграма послідовності прецеденту отримання рекомендацій

Діаграма послідовності, що зображена на рисунку 3.8, має 1 актора — авторизованого користувача і використовує 6 контролерів, які взаємодіють між собою в межах сервісної архітектури для агрегації даних та розрахунку релевантності, а саме: recommendationController, userController, evaluationController, wishlistController, quizResultsController та quizController.

Для більш детального розбору діаграми виконали її опис:

- для ініціювання процесу, авторизований користувач виконує виклик функції getPersonalizedRecommendations контролера recommendationController, передаючи заголовок запиту authHeader з JWT токеном та параметри пагінації pageNo;

- контролер recommendationController виконує запит до контролеру userController, застосувавши його функцію showUserIdByToken, для отримання унікального ідентифікатора користувача user_id з токена;

- отримавши ідентифікатор, recommendationController виконує послідовні запити до суміжних сервісів для збору історичних даних: викликає getUserEvaluations у evaluationController для отримання оцінок, getUserWishlist у wishlistController для отримання списку бажаного та getCompletedQuizzesPseudoIds у quizResultsController для отримання списку пройдених вікторин;

- для збагачення отриманих історичних даних метаданими (тегами, категоріями), recommendationController виконує запит до quizController через функцію getQuizzesByPseudoId, передаючи список псевдоідентифікаторів;

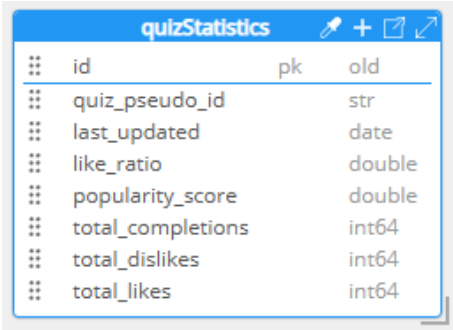
- зібравши всі дані, `recommendationController` виконує власну внутрішню функцію `buildUserPreferenceProfile` для створення об'єкта профілю користувача;
- далі контролер звертається до `quizController` через функцію `showQuizzes` для отримання списку всіх активних вікторин-кандидатів;
- отриманий список фільтрується внутрішньою функцією `filterQuizzes`, яка виключає оцінені, додані до списку бажаного та вже пройдені вікторини;
- у циклі для кожної вікторини-кандидата виконується внутрішня функція `calculateRecommendationScore`, яка розраховує бал релевантності на основі профілю використовуючи нечітке логічне виведення;
- після розрахунку всіх балів викликається функція `sortAndLimitRecommendations` для сортування списку за спаданням та відбору необхідної кількості результатів;
- після цього `recommendationController` виконує повернення сформованого списку рекомендованих вікторин користувачу.

4 РОЗРОБКА ТА ТЕСТУВАННЯ ГІБРИДНОЇ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

4.1 Реалізація гібридної рекомендації

Програмна реалізація гібридного методу рекомендації вимагає швидкого та ефективного доступу до попередньо оброблених статистичних даних, що є критичним для мінімізації часу відгуку системи. З цією метою, в рамках сервісу рекомендацій, була виділена окрема колекція `quizStatistics`, призначена для зберігання агрегованих метрик вікторин.

На рисунку 4.1 наведена структура бази даних статистики вікторин.



quizStatistics			
id	pk	old	
quiz_pseudo_id		str	
last_updated		date	
like_ratio		double	
popularity_score		double	
total_completions		int64	
total_dislikes		int64	
total_likes		int64	

Рисунок 4.1 – Структура бази даних статистики вікторин

Згідно наведеної структури, база даних статистики вікторин складається з наступних полів:

- `id` – унікальний ідентифікатор статистики вікторини, має тип `ObjectId`;
- `quiz_pseudo_id` – псевдоідентифікатор вікторини, до якої відноситься дана статистика, має тип `String`;
- `last_updated` – дата останнього оновлення документу зі статистикою вікторини, має тип `Date`;
- `like_ratio` – відношення кількості позитивних оцінок до загальної її кількості, має тип `Double`;
- `popularity_score` – числовий показник популярності, розрахований проходжень, має тип `Double`;
- `total_completions` – загальна кількість проходжень вікторини користувачами, має тип `Int64`;

– `total_dislikes` – загальна кількість негативних оцінок (дизлайків) вікторини, має тип `Int64`;

– `total_likes` – загальна кількість позитивних оцінок (лайків) вікторини, має тип `Int64`.

Для забезпечення логіки роботи формування статистики вікторини реалізували унікальний індекс для колекції `quizStatistics`, який забороняє мати два документи з однаковим `quiz_pseudo_id`. На рисунку 4.2 наведений даний індекс в програмному застосунку MongoDB Compass.

Name & Definition	Type	Size	Usage	Properties	Status
> <code>_id</code>	REGULAR	36.9 kB	Usage data unavailable	UNIQUE	READY
▼ <code>quiz_pseudo_id_1</code>	REGULAR	20.5 kB	Usage data unavailable	UNIQUE	READY

quiz_pseudo_id ↑

Рисунок 4.2 – Реалізовані індекси для колекції `quizStatistics`

Заповнення бази даних відбувається заплановано раз в годину на основі необхідної інформації про взаємодію користувачів з вікториною: її оцінки та проходження. На рисунку 4.3 наведено документ зі статистикою вікторини.

```

_id: ObjectId('69178a426c9a2b7b3d2ed855')
quiz_pseudo_id : 23
last_updated : 2025-11-16T21:00:05.412+00:00
like_ratio : 73.3
popularity_score : 13
total_completions : 13
total_dislikes : 4
total_likes : 11

```

Рисунок 4.3 – Приклад документа в колекції статистики вікторин

Окрім стандартних для Spring-додатків шарів архітектури MVC (контролери REST API, сервіси бізнес-логіки, репозиторії доступу до даних та DTO-об'єкти), для реалізації наукової складової методу було розроблено спеціалізовані категорії класів:

– `calculator` — містить імплементацію математичних моделей для розрахунку та нормалізації шести вхідних метрик;

- `fuzzy.membership` — реалізує математичний апарат функцій належності для фазифікації чітких значень;
- `fuzzy.variable` — описує абстракцію лінгвістичних змінних, що складаються з набору термів та відповідних їм функцій;
- `fuzzy.rule` — визначає структуру продукційних правил, включаючи умови, логічні оператори та висновки;
- `fuzzy.inference` — містить двигун нечіткого виведення, що реалізує алгоритм Мамдані (фазифікація, агрегація та дефазифікація);
- `fuzzy.config` — відповідає за ініціалізацію бази знань, налаштування лінгвістичних змінних та правил як Spring-бінів.

Основою інтелектуальної складової системи є база знань, яка програмно визначається через конфігурацію лінгвістичних змінних та правил виведення.

Конфігурація лінгвістичних змінних здійснюється у класі `FuzzyVariablesConfig`. Для кожного вхідного фактора створюється об'єкт `FuzzyInputVariable`, де визначається діапазон значень та набір термів. Геометрія термів задається класами `TriangularMF` (трикутна функція) та `TrapezoidalMF` (трапецієподібна функція).

У лістингу 4.1 наведено приклад конфігурації змінної `TagSimilarity` (Схожість тегів), де використовуються трапецієподібні функції для крайніх значень («Немає збігу», «Сильний збіг») та трикутні для проміжних, що забезпечує плавне перекриття зон визначення.

Лістинг 4.1 – Конфігурація лінгвістичної змінної `TagSimilarity`

```
private FuzzyInputVariable createTagSimilarity() {
    FuzzyInputVariable var = new FuzzyInputVariable("TagSimilarity", 0.0,
1.0);

    // Increased overlaps for smooth transitions
    var.addTerm("NO_MATCH", new TrapezoidalMF(0.0, 0.0, 0.10, 0.20));
    var.addTerm("WEAK_MATCH", new TriangularMF(0.10, 0.25, 0.40));
    var.addTerm("MODERATE_MATCH", new TriangularMF(0.35, 0.55, 0.70));
    var.addTerm("STRONG_MATCH", new TrapezoidalMF(0.65, 0.85, 1.0, 1.0));

    return var;
}
```

База правил визначається у класі `FuzzyRulesConfig` і являє собою набір продукцій типу «ЯКЩО (Умова А) І/АБО (Умова Б) ТО (Висновок)». Система

підтримує гнучке комбінування умов за допомогою операторів AND (мінімум) та OR (максимум), а також дозволяє задавати вагу правила для пріоритезації.

У лістингу 4.2 наведено приклади двох типів правил: перше визначає ідеальну рекомендацію на основі сильного збігу тегів та високого рейтингу, друге — знижує оцінку для контенту, що не відповідає вподобанням користувача за довжиною.

Лістинг 4.2 – Код правил

```
ruleBase.addRule(new FuzzyRule(Arrays.asList(
    new RuleCondition("TagSimilarity", "STRONG_MATCH"),
    new RuleCondition("LikeRatio", "LOVED")),
    LogicalOperator.AND, "VERY_HIGH", 1.0));
ruleBase.addRule(new FuzzyRule(List.of(
    new RuleCondition("LengthMatch", "TOO_SHORT")),
    LogicalOperator.OR, "LOW", 0.6));
```

Для розрахунку вхідних метрик системи реалізовано ряд спеціалізованих класів-калькуляторів, що трансформують теоретичні формули у програмний код.

Для визначення семантичної близькості контенту до інтересів користувача розроблено клас TagSimilarityCalculator. На відміну від класичного коефіцієнта Жаккара, який враховує лише факт перетину множин, реалізований алгоритм використовує зважений перетин. Це дозволяє врахувати силу інтересу користувача до конкретних тем. Якщо користувач має високу спорідненість до тегу, його наявність у вікторині вносить більший вклад у фінальну оцінку схожості. В лістингу 4.3 наведено код реалізації коефіцієнту Жаккара.

Лістинг 4.3 – Реалізація удосконаленого коефіцієнту Жаккара

```
public double calculate(List<String> quizTags, Map<String, Double>
userTagAffinities) {
    if (quizTags == null || quizTags.isEmpty() || userTagAffinities == null
|| userTagAffinities.isEmpty()) {
        return 0.0;
    }

    double weightedIntersection = 0.0;
    for (String tag : quizTags) {
        Double affinity = userTagAffinities.get(tag);
        if (affinity != null) {
            weightedIntersection += affinity;
        }
    }
}
```

```

int unionSize = quizTags.size();
for (String userTag : userTagAffinities.keySet()) {
    if (!quizTags.contains(userTag)) {
        unionSize++;
    }
}

if (unionSize == 0) {
    return 0.0;
}

double similarity = (weightedIntersection / unionSize) * 100.0;
return Math.max(0.0, Math.min(100.0, similarity));
}

```

Для оцінки якості контенту реалізовано клас `LikeRatioCalculator`, який використовує Байєсівське усереднення з динамічним ваговим коефіцієнтом довіри. Проблема класичного середнього полягає у ненадійності оцінок при малій кількості голосів. Реалізований метод, наведений в лістингу 4.4 вирішує це шляхом змішування «чистого» рейтингу та Байєсівського значення. Вага чистого рейтингу зростає лінійно зі збільшенням кількості голосів, досягаючи максимуму при 50 оцінках.

Лістинг 4.4 – Реалізація Байєсівського усереднення

```

public double calculate(QuizStatistics stats) {
    if (stats == null) return 50.0;

    double likes = stats.getTotalLikes();
    double dislikes = stats.getTotalDislikes();
    double totalVotes = likes + dislikes;

    if (totalVotes == 0) return 50.0;

    double priorTotal = priorLikes + priorDislikes;
    double posteriorLikes = likes + priorLikes;
    double posteriorTotal = totalVotes + priorTotal;
    double bayesianRatio = (posteriorLikes / posteriorTotal) * 100;

    double confidence = Math.min(1.0, totalVotes / 50.0);

    double actualRatio = (likes / totalVotes) * 100;

    return (confidence * actualRatio) + ((1 - confidence) * bayesianRatio);
}

```

Ще одним критичним фактором релевантності є «свіжість» контенту. Для її моделювання в класі `TemporalRelevanceCalculator` реалізовано функцію експоненціального згасання. На відміну від лінійного підходу, де релевантність спадає рівномірно, експоненціальна модель краще відображає людське сприйняття часу: різниця між "вчора" і "сьогодні" є значущою, тоді як різниця між контентом річної та дворічної давнини майже невідчутна.

У реалізації, наведеної в лістингу 4.5 використовується стала часу τ , яка визначає швидкість старіння. Метод повертає нормалізоване значення «сприйманого віку» у діапазоні $[0, 1]$, де 0 відповідає абсолютно новому контенту (щойно створеному), а значення, наближені до 1, — застарілому матеріалу.

Лістинг 4.5 – Реалізація експоненціального згасання

```
private static final double TAU = 30.0;

public double calculate(LocalDateTime createdAt) {
    if (createdAt == null) {
        return 1.0;
    }

    long daysOld = ChronoUnit.DAYS.between(createdAt, LocalDateTime.now());

    double perceivedAge = 1.0 - Math.exp(-daysOld / TAU);

    return Math.max(0.0, Math.min(1.0, perceivedAge));
}
```

Завершальним етапом роботи нечіткого контролера є процедура дефазифікації, яка перетворює результуючу нечітку множину у чітке числове значення — фінальний бал рекомендації. У класі `FuzzyInferenceEngine` реалізовано найбільш поширений та точний метод — метод центру тяжіння.

Програмно цей алгоритм реалізовано через чисельне інтегрування. Діапазон вихідної змінної розбивається на дискретні кроки, і для кожної точки обчислюється її внесок у загальний момент першого порядку (чисельник) та площу фігури (знаменник). Метод також містить захисний механізм для обробки випадків, коли жодне правило не спрацювало (площа дорівнює нулю), повертаючи у такому разі нейтральну оцінку. Код реалізації наведено в лістингу 4.6.

Лістинг 4.6 – Код реалізації методу центру тяжіння

```
private double defuzzify(Map<String, Double> outputMemberships) {
    double min = outputVariable.getMinValue();

    double max = outputVariable.getMaxValue();

    double step = (max - min) / 200.0;

    double numerator = 0.0;

    double denominator = 0.0;

    for (double x = min; x <= max; x += step) {
        double membership = calculateAggregatedMembership(x,
outputMemberships);

        numerator += x * membership;
        denominator += membership;
    }

    if (denominator < 0.0001) {
        return 50.0;
    }

    return numerator / denominator;
}
```

Реалізація методу гібридної рекомендації завершується наданням кінцевому користувачеві зручного інтерфейсу для доступу до сформованих рекомендацій. Оскільки рекомендаційний сервіс є окремим компонентом, він взаємодіє з клієнтською частиною платформи через програмний інтерфейс (API).

На клієнтській стороні реалізована можливість показу персоналізованих рекомендацій безпосередньо на сторінці вікторин. Користувач натискає на кнопку «Recommended for you», що ініціює запит до сервісу рекомендацій.

Після обробки запиту та повернення списку вікторин, клієнтська частина відображає їх у спеціальному розділі, сортованими за розрахованим балом релевантності. Це забезпечує наочний та швидкий доступ до контенту, який відповідає інтересам та уподобанням користувача.

На рисунку 4.4 наведено приклад списку рекомендацій, які отримав користувач.

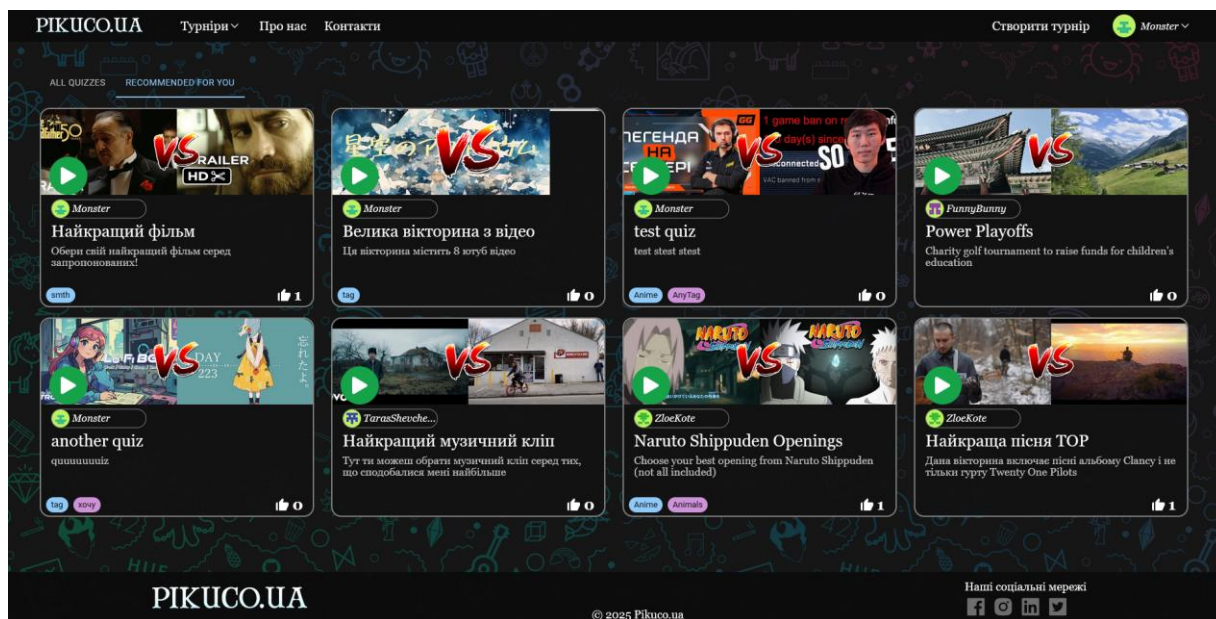


Рисунок 4.4 – Отримання користувачем списку рекомендацій

4.2 Експериментальна перевірка

Для перевірки коректності функціонування розробленого методу та програмного забезпечення було проведено експериментальне дослідження. Метою тестування є верифікація здатності системи розподіляти контент відповідно до сформованого профілю користувача, враховуючи фактори популярності, новизни та контекстні обмеження.

Експеримент проводився на заздалегідь підготовленому наборі даних, що моделює типову поведінку користувача на платформі.

У системі створено тестового користувача, який має пройдені, оцінені та додані до списку бажаного вікторини. Для формування чітко вираженого вектора інтересів було змодельовано історію з 5 пройдених вікторин, які включають наступні сценарії:

- позитивний досвід – користувач пройшов та позитивно оцінив 4 вікторини, пов'язані з темами «Programming», «Technology» та «Gaming». Це дозволило системі ідентифікувати ці теги як пріоритетні;
- негативний досвід – одна вікторина з категорії «Food» була оцінена негативно, що знизило вагу відповідних тегів у профілі;
- вподобання щодо авторів – серед оцінених вікторин присутні роботи автора «TechGuru», що дозволило сформувати метрику спорідненості з цим творцем контенту;

– вподобання по довжині вікторини – більшість пройдених вікторин мали довжину 8 питань. На основі цього система розрахувала середню бажану довжину контенту, що приблизно дорівнює 8.

З загального пулу вікторин було відібрано 5 тестових кандидатів, які не перетиналися з історією користувача. Кожна з цих вікторин була спроектована для перевірки конкретного правила або механізму роботи алгоритму. Характеристика кандидатів наведена у таблиці 4.1.

Таблиця 4.1 – Характеристика вхідного набору вікторин-кандидатів

Назва вікторини	Основні характеристики					Сценарій тестування
	Теги	Автор	Рейтинг	Довжина	Дата створення	
Java vs Python: The Ultimate Battle	Programming, Technology	TechGuru	99%	8 питань	10.12.2025	Перевірка здатності системи ідентифікувати контент, що максимально відповідає всім критеріям профілю
All 16 Mario Games Ranked	Gaming, Nintendo	NintendoFan	92%	16 питань	01.12.2025	Перевірка впливу відхилення довжини (16 проти бажаних 8) на зниження оцінки при високій тематичній релевантності
Best Makeup brands 2025	Beauty, Fashion	GlamGirl	88%	8 питань	05.12.2025	Перевірка правила, що дозволяє рекомендувати популярний контент навіть за відсутності перетину інтересів
Best PC Build 2010	Technology, Hardware	OldSchool	95%	4 питання	20.05.2010	Перевірка роботи експоненціальної функції згасання для контенту, який тематично підходить, але втратив актуальність
Worst Movies of All Time	Cinema	RandomUser	20%	4 питання	01.12.2025	Перевірка здатності системи відсіювати нерелевантний та неякісний контент

Цей набір даних дозволяє комплексно оцінити роботу всіх компонентів гібридної системи: від розрахунку подібності Жаккара до застосування нечітких правил для вирішення конфліктів між різними факторами.

В результаті обробки сформованого набору вхідних даних алгоритмом гібридної рекомендації було отримано впорядкований список вікторин. Система розрахувала інтегральний показник релевантності для кожного кандидата, враховуючи налаштовані правила нечіткої логіки.

На рисунку 4.5 наведено візуалізацію клієнтського інтерфейсу з відображенням сформованого списку рекомендацій.

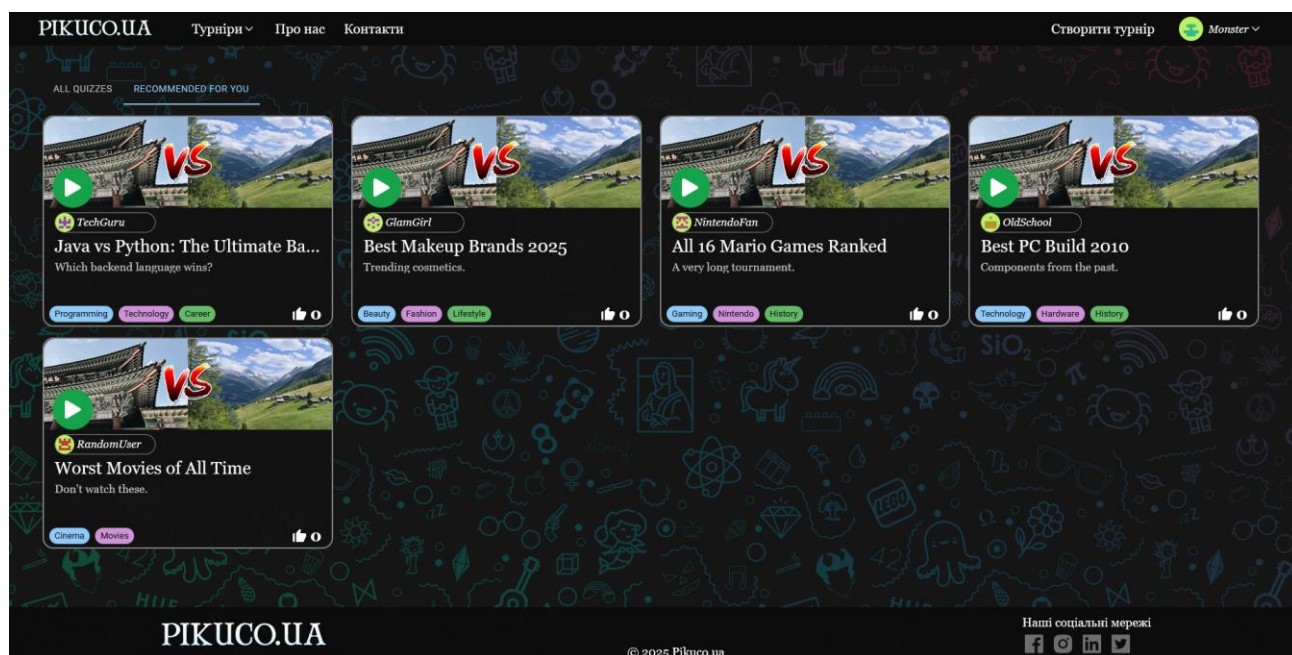


Рисунок 4.5 – Отриманий користувачем список вікторин

Результати роботи алгоритму представлено у таблиці 4.2. Вікторини відсортовано за спаданням розрахованого балу, що відповідає порядку їх відображення користувачу.

Таблиця 4.2 – Отриманий список рекомендацій користувачем

№	Назва вікторини	Бал релевантності
1	Java vs Python: The Ultimate Battle	94.30
2	Best Makeup Brands 2025	92.74
3	All 16 Mario Games Ranked	50.00
4	Best PC Build 2010	39.81
5	Worst Movies of All Time	23.47

Отримані числові значення демонструють високу чутливість розробленого методу. На основі розрахованих балів зробили інтерпретацію отриманого списку рекомендацій.

Вікторина «Java vs Python: The Ultimate Battle» отримала найвищий бал, який дорівнює 94.30, що свідчить про повну синергію всіх позитивних метрик: максимальний збіг тематики, приналежність до автора, якому користувач довіряє, оптимальна довжина контенту та високий рейтинг схвалення. Система ідентифікувала цей контент як «ідеальний збіг». Водночас, вікторина «Best Makeup Brands 2025» посіла друге місце та отримала 92.74 бал релевантності, незважаючи на відсутність тематичного перетину з профілем користувача. Такий високий бал пояснюється надзвичайно високою популярністю. Це підтверджує коректну роботу правил, які дозволяють контенту з сильним соціальним доказом «пробивати» тематичний фільтр, забезпечуючи різноманітність та елемент щасливого збігу обставин.

Результат для вікторини «All 16 Mario Games Ranked», який отримав 50 балів, є прямим наслідком конфлікту факторів: високий позитивний вплив збігу тегів був повністю нівельований значним негативним впливом відхилення довжини (16 питань проти бажаних 8). Метод дефазифікації в цьому випадку виявив стан невизначеності та повернув нейтральний центр тяжіння. Система також ефективно обробила контент, що втратив актуальність: вікторина «Best PC Build 2010» отримала низький бал – 39.84. Хоча тег Hardware відповідає інтересам користувача, її дата створення активувала механізм експоненціального згасання, що суттєво знизило фінальну оцінку. Це демонструє ефективну фільтрацію застарілого контенту. Найнижчий бал, а саме 23.47, отримала вікторина «Worst Movies of All Time», оскільки контент не має ані перетину тегів, ані високої якості, ані популярності, що призвело до спрацювання правил «шумового фільтру».

Отримані результати верифікують, що всі шість вхідних метрик та налаштована база правил коректно взаємодіють у рамках нечіткої логіки, формуючи адекватну розподільну оцінку для кожного сценарію.

Для оцінки адекватності роботи розробленого методу проведено порівняння отриманих кількісних результатів у вигляді балу релевантності з якісною експертною оцінкою очікуваної поведінки системи для кожного сценарію тестування.

На рисунку 4.6 наведено гістограму розподілу балів релевантності для тестового набору вікторин.

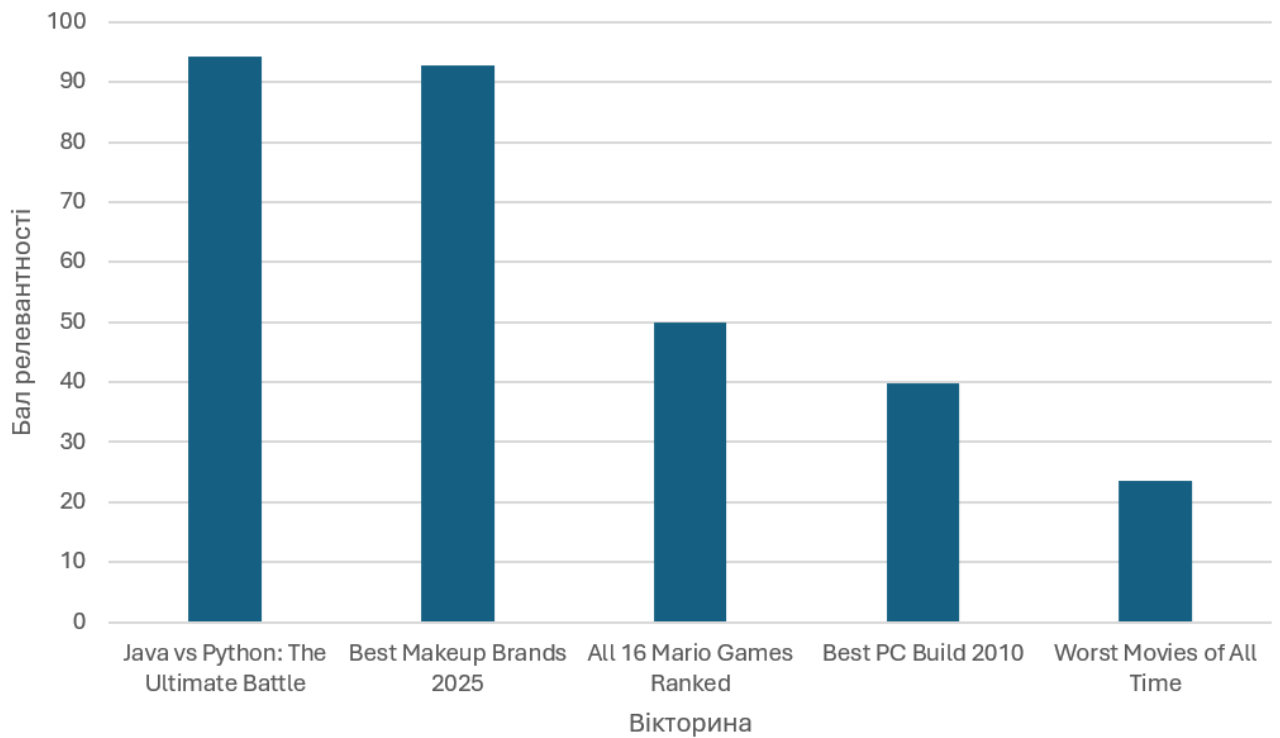


Рисунок 4.6 – Гістограма розподілу балів релевантності

Графік демонструє чітку диференціацію кандидатів: спостерігається значний розрив між групою рекомендованого контенту, який має більше 90 балів, та групою сумнівного або нерелевантного контенту, яка має менше 50 балів. Це свідчить про високу роздільну здатність алгоритму та його здатність формувати однозначні рекомендації.

Результати експериментального дослідження підтверджують, що реалізований гібридний метод рекомендації забезпечує коректне ранжування контенту в умовах багатокритеріального вибору. Система продемонструвала здатність ефективно вирішувати конфлікти між різнорідними факторами, враховувати часову актуальність інформації та адаптуватися до популярних трендів. Порівняльний аналіз показав повну відповідність отриманих результатів закладеній логіці нечіткого виведення, що свідчить про коректність програмної реалізації методу.

4.3 Проблеми використання та перспективи розвитку гібридної рекомендаційної системи

Незважаючи на доведену ефективність розробленої системи у вирішенні проблеми «холодного старту», її впровадження у високонавантажене середовище виявило ряд технологічних обмежень.

Суттєвим викликом є обчислювальна складність алгоритму нечіткого виведення, що при масштабуванні бази до десятків тисяч вікторин може призводити до затримок у реальному часі, вимагаючи ефективніших механізмів кешування. Також обмежуючим фактором є статичність бази знань, де правила налаштовані експертним шляхом і не мають механізму автоматичної адаптації до динамічних змін вподобань аудиторії без участі розробника. Крім того, точність рекомендацій критично залежить від якості тегування контенту авторами, що знижує релевантність рекомендацій у випадку некоректних або занадто загальних метаданих.

З урахуванням виявлених проблем, подальший розвиток системи доцільно спрямувати на наступні напрямки:

- трансформація класичної моделі Мамдані в адаптивну нейро-нечітку систему виведення, що дозволить автоматизувати налаштування функцій належності та ваги правил шляхом навчання на історичних даних про взаємодію користувачів;

- впровадження модулів обробки природної мови для автоматичного аналізу текстового вмісту вікторин, що зменшить залежність від ручного проставлення тегів та підвищить точність визначення семантичної схожості;

- інтеграція векторного пошуку для етапу попереднього відбору кандидатів, що дозволить застосовувати ресурсоємне нечітке ранжування лише для найбільш релевантної вибірки об'єктів, знижуючи навантаження на систему.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи магістра була розроблена гібридна рекомендаційна система для інформаційної системи «Онлайн-платформа інтерактивних вікторин». Система реалізована як окремий сервіс, що використовує апарат нечіткої логіки для прийняття рішень. Розробка системи пройшла наступні етапи: аналіз предметної області та методів; розробка теоретичного методу рекомендації; проєктування системи; програмна реалізація та експериментальне тестування.

Під час аналізу предметної області дослідили специфіку платформ користувацького контенту та виявили ключову проблему інформаційного перевантаження. Виконали аналіз існуючих методів рекомендації, визначивши їхні недоліки, зокрема вразливість до проблеми «холодного старту» та переспеціалізації. Обґрунтували необхідність використання гібридного підходу, посиленого нечіткою логікою, для обробки суб'єктивних та неточних даних про вподобання користувачів.

Для розробки методу гібридної рекомендації сформували математичний апарат попередньої обробки даних, впровадивши Байєсівське усереднення для підвищення достовірності рейтингів та функцію експоненціального згасання для моделювання часової актуальності. Побудували модель нечіткого логічного виведення типу Мамдані: визначили лінгвістичні змінні, спроектували функції належності та розробили базу експертних правил.

Під час проєктування методу побудували функціональну модель системи за стандартом IDEF0, декомпозивавши процес рекомендації на етапи формування профілю, розрахунку метрик та нечіткого виведення. Розробили UML-діаграму активності для візуалізації алгоритму роботи та діаграму послідовності для опису взаємодії сервісів при формуванні рекомендацій. Визначили структуру бази даних MongoDB для зберігання агрегованої статистики вікторин.

На етапі реалізації та тестування виконали програмну реалізацію методу у вигляді сервісу. Провели експериментальну перевірку на тестовому наборі даних, яка підтвердила здатність системи ефективно вирішувати проблему «холодного старту», розподіляти контент із високою роздільною здатністю та адаптуватися до популярних трендів, забезпечуючи високу релевантність рекомендацій для кінцевого користувача.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Yera R. Fuzzy tools in recommender systems: a survey / R. Yera, L. Martinez // Atlantis press. – 2017. – Т. 10, № 1. – С. 776–803.
2. Котенко І.І. Розробка рекомендаційної системи для онлайн-платформи інтерактивних вікторин / Розвиток сучасної науки: актуальні питання теорії та практики: матеріали ІХ Всеукраїнської студентської наукової конференції, м. Київ, 21 листопада, 2025 рік / ГО «Молодіжна наукова ліга». Вінниця: ТОВ «УКРЛОГОС Груп», 2025. С.516-517.
3. Laudon K. C. Management information systems: managing the digital firm / Kenneth C. Laudon, Jane P. Laudon. – New Jersey : Pearson Education, 2020. – 688 с.
4. Ricci F. Recommender systems handbook / Francesco Ricci, Lior Rokach, Barak Shapira; ред.: F. Ricci, L. Rokach, B. Shapira. – Boston : Springer, 2015. – 1003 с.
5. A novel hybrid recommender system for the tourism domain [Електронний ресурс] / G. Chalkiadakis [та ін.] // Algorithms, Basel, 21 квіт. 2023 р. – [Б. м.]. – Режим доступу: <https://doi.org/10.3390/a16040215> (дата звернення: 20.11.2025).
6. Pandey N. A fuzzy logic-based recommender system for e-learning system with multi-agent framework [Електронний ресурс] / Himanshu Pandey, Vir Singh // International journal of computer applications, 15 лип. 2015 р. – [Б. м.], 2015. – С. 18–21. – Режим доступу: <https://www.semanticscholar.org/paper/A-Fuzzy-Logic-based-Recommender-System-for-System-Pandey-Singh/e86b07582c1d040aaf6b0afbfb6a3922a4476ca5> (дата звернення: 20.11.2025).
7. Fuzzy ontological model of knowledge representation for the humanitarian response [Електронний ресурс] / Olha Chala [та ін.] // Proceedings of the phd workshop on artificial intelligence in computer science at 9th international conference on computational linguistics and intelligent systems, Харків, 15–16 трав. 2025 р. – [Б. м.], 2025. – С. 47–59. – Режим доступу: <https://ceur-ws.org/Vol-4015/> (дата звернення: 20.11.2025).
8. Juanr M. Crop recommender system for the farmers using mamdani fuzzy inference model [Електронний ресурс] / Mudhusree Juanr, Vikram Rath, S. Mohanty // International journal of engineering and technology : наук. робота, 7 жовт. 2018 р. – [Б. м.]. – Режим доступу: <https://www.sciencepubco.com/index.php/ijet/article/view/23006> (дата звернення: 20.11.2025).

9. Lika B. Facing the cold start problem in recommender systems / B. Lika, K. Kolomvatsos, S. Hadjiefthymiades // Expert systems with applications. – 2014. – Т. 41, № 4. – С. 2065–2073.
10. Ding Y. Time weight collaborative filtering / Y. Ding, X. Li // Proceedings of the 14th ACM international conference on information and knowledge management. – [Б. м.], 2005. – С. 485–492.
11. Son J. Content-based filtering for recommendation systems using multi-attribute networks / J. Son, S. Kim // Expert systems with applications. – 2017. – Т. 89. – С. 404–412.
12. Personalized tour recommendation based on user interests and points of interest visit durations / К. Н. Lim [та ін.] // Proceedings of the 24th international joint conference on artificial intelligence (IJCAI). – [Б. м.], 2015. – С. 1785–1791.
13. Miller D. A. Bayesian average ratings [Электронний ресурс] / D. A. Miller. – Режим доступу: <https://www.evanmiller.org/bayesian-average-ratings.html> (дата звернення: 20.11.2025).
14. Gao H. Weighted similarity and core-user-core-item based recommendations / H. Gao, Others // Entropy. – 2022. – Т. 24, № 5. – С. 609.
15. Al-Ghamdi M. Exponential decay function-based time-aware recommender system for e-commerce applications / M. Al-Ghamdi, A. El-Halees // International journal of advanced computer science and applications. – 2022. – Т. 13, № 10. – С. 554–561.
16. Patel M. M. A review on data normalization techniques for data mining / M. M. Patel, S. Patel // International journal of engineering research & technology. – 2015. – Т. 4, № 11. – С. 120–122.
17. Integration Definition for Function Modeling (IDEF0). Software Standard, Modeling Techniques. – Nation Institute of Standards and Technology, 1993.
18. Jain A. Activity Diagrams - Unified Modeling Language (UML). GeeksforGeeks. URL: <https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-activity-diagrams/> (дата звернення: 15.12.2025).
19. Newman S. Building Microservices: Designing Fine-Grained Systems. 2-ге вид. O'Reily Media, 2021. 612 с.
20. Sierra K., Bates B., Gee T. Head first java: a brain-friendly guide. 3-тє вид. O'Reilly Media, Incorporated, 2022. 752 с.
21. Spring framework documentation. Spring. URL: <https://docs.spring.io/spring-framework/reference/index.html> (дата звернення: 15.12.2025).

22. Flanagan D. JavaScript: the definitive guide. 7-ме вид. O'Reilly Media, 2020. 706 с.
23. Wieruch R. The road to react: your journey to master plain yet pragmatic react.js. Independently published, 2025. 286 с.
24. Ruvalcaba Z., Boehm A. Murach's HTML and CSS (6th Edition). Murach & Associates, Incorporated, Mike, 2024.
25. What is mongodb? - mongodb manual v8.2. MongoDB: The Developer Data Platform | MongoDB. URL: <https://www.mongodb.com/docs/manual/> (дата звернення: 15.12.2025).
26. Jain A. Sequence Diagrams - Unified Modeling Language (UML). GeeksforGeeks. URL: <https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-sequence-diagrams/> (дата звернення: 15.12.2025).