НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Факультет прикладної математики

Кафедра прикладної математики

«До захисту допущено»

Завідувач кафедри

_____ О. Р. Чертов

«___» _____ 2016 р.

Дипломна робота

на здобуття ступеня бакалавра

з напряму підготовки 6.040301 «Прикладна математика»

на тему: «Автоматизована підсистема класифікації публікацій із використанням універсального десяткового коду»

Виконала: студентка IV курсу, групи КМ-23

Нагорна Віка Андріївна _____

Керівник асистент

Ковальчук-Химюк Л. О. _____

Консультант із старший викладач

нормоконтролю Мальчиков В. В. _____

Рецензент доцент, канд. техн. наук

Селіванов В. Л. _____

Засвідчую, що в цій дипломній роботі немає запозичень із праць інших авторів без відповідних посилань.
Студентка _____

Київ — 2016

Національний технічний університет України

«Київський політехнічний інститут»

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти — перший (бакалаврський)

Напрям підготовки 6.040301 «Прикладна математика»

ЗАТВЕРДЖУЮ

завідувач кафедри

_____ О. Р. Чертов

«___» _____ 2016 р.

ЗАВДАННЯ

на дипломну роботу студентці

Нагорній Віці Андріївні

1. Тема роботи: «Автоматизована підсистема класифікації публікацій із використанням універсального десяткового коду»,

керівник роботи Ковальчук-Химюк Людмила Олександрівна, асистент,

затверджені наказом по університету від «06» травня 2016 р. № 1499-С.

2. Термін подання студенткою роботи: «16» червня 2016 р.

3. Вихідні дані до роботи: розроблювана підсистема повинна працювати з даними публікацій, мінімальна точність класифікації — 70%.

4. Зміст роботи: виконати аналіз існуючих методів розв'язання задачі, вибрати метод побудови класифікатора для класифікації публікацій, спроектувати автоматизовану підсистему класифікації публікацій, здійснити програмну реалізацію розробленої підсистеми, провести тестування розробленої підсистеми.

5. Перелік ілюстративного матеріалу: графік зміщення оцінки, ілюстрації результатів роботи програми.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка завдання (розділ 1). Аналіз предметної області (розділ 2).	Ковальчук-Химюк Л.О, асистент кафедри ПМА, ФПМ		

7. Дата видачі завдання: «22» лютого 2016 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за тематикою та збір даних	23.12.2015	
2	Проведення порівняльного аналізу математичних методів класифікації публікацій	12.01.2016	
3	Підготовка матеріалів першого розділу роботи	15.02.2016	
4	Підготовка матеріалів другого розділу роботи	24.02.2016	
5	Підготовка матеріалів третього розділу роботи	05.03.2016	
6	Розроблення математичного забезпечення для побудови класифікатора за яким відбувається класифікація публікацій	30.03.2016	
7	Підготовка публікації на Міжнародну конференцію молодих учених, студентів та аспірантів в інституті харчових технологій	01.04.2016	
8	Підготовка матеріалів четвертого розділу роботи	11.05.2016	
9	Розроблення програмного забезпечення для класифікації та пошуку публікацій	15.05.2016	

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
10	Оформлення пояснювальної записки	7.06.2016	

Студентка _____ Нагорна В. А

Керівник роботи _____ Ковальчук-Химюк Л. О

АНОТАЦІЯ

Дипломну роботу виконано на 66 аркушах, вона містить 2 додатки та перелік посилань на використані джерела з 10 найменувань. У роботі наведено 18 рисунків, 1 графік та 2 таблиці.

Дана робота присвячена розробці автоматизованої підсистеми, яка виконує автоматичну класифікацію, збереження, пошук та відображення публікацій.

В дипломній роботі детально описано всі етапи проектування підсистеми «Автоматизована підсистема класифікації публікацій із використанням універсального десяткового коду» та особливості створення цієї підсистеми. Наведено порівняльний аналіз основних математичних методів та підходів класифікації публікацій, та обирається найкраща модель за певними критеріями з точки зору практичності та адекватності серед інших для побудови класифікатора. Отримується формалізований алгоритм даної моделі і виконується його програмна реалізація. Проводиться тестування побудованого класифікатора на основі вибраної моделі на матеріалі існуючих публікацій. Результати роботи порівнюються із даними результатів іншого існуючого класифікатора і встановлюється рівень універсальності впровадженого в даній роботі методу.

Результатом розробки даної підсистеми є модель даних, що описує дану підсистему та створена окрема схема даних, в якій зберігається розроблена база даних для цієї підсистеми.

Основні положення дипломної роботи опубліковано у вигляді тез доповіді на Міжнародній науково-технічній конференції молодих учених, студентів та аспірантів, що проходила в університеті харчових технологій.

Ключові слова: автоматизована підсистема, байєсові мережі довіри, класифікація публікації, класифікатор Байєса, публікація, універсальна десяткова класифікація.

ABSTRACT

The thesis is presented in 66 pages. It contains 2 appendixes and bibliography of 10 references. Eighteen figures, one graph and two tables are given in the thesis.

This thesis is dedicated to the development of automated subsystem, which performs automatic classification, storage, search and display publications.

This paper describes in detail all the stages of designing the subsystem " Subsystem for automated classification of publications using Universal Decimal Classifier " and features the creation of this subsystem. In this paper the comparative analysis of the basic mathematical methods and approaches of classification of publications and elected best model in terms of practicality and adequacy among others for building the classifier.

Formal algorithm of the model has been composed and its implementation has been performed. Built classifier had tested based on the model chosen on the material of the finished publications. The results are compared with the data of the results of other similar qualifiers and then established the level of universality method embedded in this work.

The result of the development of this subsystem is the data model, who describes this subsystem, constructed a separate database schema where the database is stored designed for this subsystem.

Main ideas of the thesis were published in the Proceedings of the International Scientific and Technical Conference of young scientist and students.

Keywords: automated subsystem, bayesian networks of trust, the classification of publications, Bayesian classifier, publication, Universal Decimal Classification.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 10
ВСТУП 11
1 ПОСТАНОВКА ЗАДАЧІ 13
2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 15
2.1 Огляд основних підходів класифікації публікацій 15
2.1.1 Ручна класифікація 15
2.1.2 Автоматична класифікація 16
2.2 Аналіз існуючих рішень класифікації публікацій 17
2.2.1 Система класифікації документів у науково-технічній бібліотеці імені Г.І Денисенка 18
2.2.2 ABBY FineReader Engine для Windows 18
2.2.3 Морфологічний додаток Mystem 19
2.3 Вибір технології розробки 20
2.4 Огляд технологій 20
2.4.1 Мова програмування PHP 21
2.4.2 Мова програмування JavaScript 22
2.4.3 Огляд СКБД 23
2.4.4 MySQL Workbench 24
2.5 Огляд математичних методів 25
2.5.1 Формалізація задачі 27
2.5.2 Порівняльний аналіз існуючих методів 27
2.5.2.1 Метод Байєса (Naive Bayes) 28
2.5.2.2 Метод Rocchio 30
2.5.2.3 Метод k - найближчих сусідів 31
2.5.2.4 Метод опорних векторів 32
2.5.2.5 Метод дерев прийняття рішень 33
2.5.2.6 Нейронні мережі 34

2.5.3 Вибір і обґрунтування методів рішення 36
2.6 Висновки до розділу 37
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ КЛАСИФІКАЦІЇ ПУБЛІКАЦІЙ 39
3.1 Формулювання математичної частини класифікації публікацій обраним методом..... 39
3.2 Побудова алгоритму класифікації 39
3. 2.1 Проблема арифметичного переповнення 41
3.2.2 Оцінка параметрів моделі Байєса 41
3.2.3 Проблема невідомих слів 42
3.3 Реалізація класифікатора 44
3.4 Висновки до розділу 45
4 СТРУКТУРА І ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 46
4.1 Загальна структура 46
4.1.1 Границі проекту 46
4.1.2 Бізнес-потреби 46
4.1.3 Безпека 46
4.1.4 Продуктивність 48
4.1.5 Супровід 49
4.1.6 Доступність 49
4.1.7 Людський фактор 49
4.1.8 Методології 50
4.1.9 Масштабованість 50
4.2 Структура програмних засобів 50
4.2.1 Концептуальне моделювання 50
4.2.2 Структурне моделювання 52
4.3 Керівництво користувача 54
4.4 Випробування компонентів підсистеми 60
4.4.1 Контрольний приклад № 1 60
4.4.2 Контрольний приклад № 2 61

4.4.3 Контрольний приклад № 3 62
4.5 Висновки до розділу 62
ВИСНОВКИ 64
ПЕРЕЛІК ПОСИЛАНЬ 65
Додаток А Лістинги програм 67
Додаток Б Ілюстративний матеріал 87

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ERD – Entity-relationship Diagram – діаграма «сутність-зв'язок», описує концептуальну схему предметної області.

IDEF3 – Integrated DEFinition for Process Description Capture Method – методологія моделювання і стандарт документування процесів, що походять в системі.

АС – автоматизована система.

БД – база даних.

ДНФ – диз'юнктивна нормальна форма.

Оверфіттинг – явище коли при побудові алгоритму класифікації виходить такий алгоритм, який дуже добре працює на тестових прикладах, але досить погано працює взагалі.

СКБД – система керування базами даних.

Стемінг – процес скорочення слова до основи шляхом відкидання допоміжних частин, таких як закінчення чи суфікс.

УДК – універсальна десяткова класифікація.

ВСТУП

З кожним роком інформаційні технології відіграють все більшу роль в житті людей. Всі підприємства, освітні та державні установи мають свою власну систему, несучу в собі весь обсяг інформації для підтримки прямої діяльності даної організації.

В даний час існує безліч різних інформаційних систем, кожна з яких має свої власні особливості. Для того, щоб полегшити і прискорити роботу людей, дані системи та програми повинні бути побудовані на однакових принципах. При цьому тимчасові витрати на адаптацію до роботи в даних інформаційних системах для непідготовленої людини повинні бути мінімальні.

Процес розробки стратегії розміщення наукових публікацій у єдиній інформаційній системі містить в собі багатоетапний аналіз статистичних даних великих об'ємів, на основі якого приймаються рішення. За допомогою автоматизації даного процесу можна систематизувати та впорядкувати великі об'єми інформації, в результаті чого підвищиться ефективність використання часових, майнових та людських ресурсів, а також суттєво зменшиться ймовірність прийняття неоптимального рішення.

З огляду на зростання кількості текстової інформації повсюдно, а особливо в мережі Інтернет, все більшу роль відіграє можливість класифікувати її, відбирати лише актуальну її частину.

Для вирішення цього завдання часто застосовуються різноманітні тематичні класифікатори, рубрикатори, які дозволяють шукати (автоматично або вручну) документи в невеликій підмножині документної бази по відповідній тематиці, що цікавить користувача.

Класифікація документів – це сортування документів заздалегідь відомими категоріями. Задачі класифікації зустрічаються у будь-яких випадках, коли є потреба автоматичної організації документів, наприклад, категоризація сторінок в Інтернеті, виявлення небажаної поштової кореспонденції, автоматичної генерації метаданих.

Тому, тема даної роботи пов'язана з розробкою автоматизованої підсистеми, яка виконує автоматичну класифікацію, збереження, пошук та відображення публікацій. Вона є актуальною та корисною у житті людини, адже інформаційні системи стають все більш звичайною частиною щоденного життя.

1 ПОСТАНОВКА ЗАДАЧІ

Метою даної роботи є створити автоматизовану підсистему, яка виконує автоматичну класифікацію, збереження, пошук та відображення публікацій.

Основною задачею підсистеми є класифікація, збереження та відображення наукових публікацій, швидкий пошук цих публікацій, що зможе значно полегшати життя користувачів.

Дана підсистема повинна включати наступне:

- класифікація нової публікації, що надходить до підсистеми;
- швидкий та зручний пошук публікацій;
- веб-застосунок, що формує інтерфейс;
- створення єдиної бази, де будуть зберігатись усі завантажені до підсистеми нові публікації.

Етапи розв'язку поставленої задачі:

- аналіз предметної області, розгляд та вибір моделі класифікації публікацій;
- математичне забезпечення вибраного класифікатора;
- реалізація алгоритму класифікації публікацій;
- проектування бази даних;
- проектування графічного інтерфейсу користувача;
- програмна реалізація збудованого класифікатора;
- тестування і перевірка отриманого програмного продукту.

Перший етап передбачає ознайомлення із існуючими моделями класифікації публікацій і вибір серед них найбільш придатної з точки зору оптимальності і складності реалізації. Наступні етапи потребують виведення і обґрунтування послідовності дій, що мають бути застосовані до обраної моделі з метою забезпечення адекватного вибору категорії, до якої відноситься публікація. Кінцевим продуктом є формалізований опис алгоритму розрахункової частини розробленої програми. Отримана алгоритмічна документація має бути використана в частині, присвяченій

програмній реалізації. До цієї частини також входить проектування графічного інтерфейсу користувача, яким має задовольняти кінцевий програмний продукт. Останній етап слугує власне перевірці обраної моделі класифікації публікацій із існуючими аналогами.

Вимоги до програмного забезпечення:

- PHP 5.5 з використанням MySQL 5.7 або старші для сервера, оскільки MySQL є безкоштовною системою керування базами даних, зручною у використанні, і при її встановленні не виникає великих труднощів;
- браузер з підтримкою JavaScript;
- підключення до мережі Інтернет для клієнта.

2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

2.1 Огляд основних підходів класифікації публікацій

Розглянемо основні підходи, що використовуються на різних етапах розв'язку задачі класифікації публікацій.

2.1.1 Ручна класифікація

Класифікація не завжди здійснюється за допомогою комп'ютера. Наприклад, у звичайній бібліотеці тематичні рубрики присвоюються книгам вручну бібліотекарем. Подібна ручна класифікація дорога і непридатна у випадках, коли необхідно класифікувати велику кількість документів з високою швидкістю.

Використовується такий підхід в:

- парламентських службах;
- Looksmart – публічна інтернет-рекламна компанія;
- About.com – англомовний веб-сайт, онлайн-джерело інформації і порад для споживачів;
- PubMed – електронна база даних медичних і біологічних публікацій, в якій викладені абстракти публікацій англійською мовою;
- CSV – Comma-Separated Values – текстовий формат, призначений для представлення табличних даних. Кожен рядок файлу – це один рядок таблиці;
- бібліотеки (УДК).

2.1.2 Автоматична класифікація

Інший підхід полягає в написанні правил, за якими можна віднести текст до тієї чи іншої категорії.

Наприклад, одне з таких правил може виглядати наступним чином: «якщо текст містить слова такі як похідна і рівняння, то віднести його потрібно до категорії – математика». Спеціаліст, знайомий з предметною областю і володіє навиком написання регулярних виразів, може скласти ряд правил, які потім автоматично застосовуються до надходжених документів для їх класифікації.

Даний підхід ґрунтується на машинному навчанні. У ньому набір правил або, більш загально, критерій прийняття рішення текстового класифікатора, обчислюється автоматично з навчальних даних (іншими словами, проводиться навчання класифікатора).

Навчальні дані – це деяка кількість хороших зразків документів з кожного класу. У машинному навчанні зберігається необхідність ручної розмітки (термін розмітка означає процес приписування класу документу). Але розмітка є більш простим завданням, ніж написання правил. Крім того, розмітка може бути проведена в звичайному режимі використання системи. Наприклад, в програмі електронної пошти може існувати можливість позначати листи як спам, тим самим формуючи навчальну множину для класифікатора - фільтра небажаної пошти.

Таким чином, класифікація текстів, заснована на машинному навчанні, є прикладом навчання з учителем, де в ролі вчителя виступає людина, що задає набір класів і розмічає навчальну множину. Цей підхід кращий за попередній, оскільки процес класифікації автоматизується і отже, кількість оброблюваних документів практично не обмежена.

2.2 Аналіз існуючих рішень класифікації публікацій

Класифікація (рубрикація) текстових публікацій є задачею автоматичного визначення публікації в одну чи декілька категорій (тематик) на основі змісту публікації. В зарубіжній літературі отримав широке розповсюдження термін Text Categorization [1,2].

На даний час ми маємо діло із постійним збільшенням об'єму оброблювальної і накопичуваної інформації, що робить задачу класифікації все більш актуальною. Використання класифікаторів дає змогу обмежити пошук необхідної інформації відносно невеликої множини документів.

Поміж звуження області пошуку в пошукових системах задача класифікації має практичне призначення в наступних областях:

- фільтрація спаму;
- створення тематичних каталогів;
- контекстна реклама;
- системи документовідбору;
- автоматичний перевід тексту.

Проблемам класифікації текстів присвячено чимало досліджень. Зокрема, в [1] дається загальний підхід до процесу класифікації, виділяються основні етапи, вимоги до результатів кожного з них. Також розглядається лінійний онлайн-класифікатор, метод ДНФ -правил, метод регресії. Але в [1] не виконано оцінювання кожного з методів з точки зору доцільності та оптимальності використання на практиці. Загалом, у [1] залишилась невивченою практична сторона описаних методів, деякі з них не придатні для якісної класифікації документів, не враховуються питання оптимального алгоритму, за яким формуються терми та їх вагові коефіцієнти.

2.2.1 Система класифікації документів у науково-технічній бібліотеці імені Г.І Денисенка

Однією із не дуже вдалих розробок класифікації документів по каталогам є розробка класифікації різних друкованих видань у науково-технічній бібліотеці імені Г.І Денисенка Національного технічного університету України «Київський політехнічний інститут». Дана класифікація переважно відбувається у ручну. Автоматичний класифікатор, що вбудований, може класифікувати лише певні друковані видання по існуючим каталогам. Переважна більшість видань лишається на ручне каталогіювання. Людина, що відповідає за внесення книги до бази бібліотеки, повинна сама вирішити у який каталог її віднести, але на основі вже існуючих. Система є трохи застарілою і не передбачається внесення нових каталогів до бази. Пошук відбувається лише по ключовим словам.

2.2.2 ABBY FineReader Engine для Windows

Одним із сучасних середовищ розробки є ABBY FineReader Engine для Windows – призначеного для інтеграції в Windows - додатках передових технологій розпізнавання друкованих (OCR) і рукопечатних (ICR) символів, знаків (OMR) і штрих-кодів, а також технологій перетворення і класифікації публікацій.

У новій версії ABBYY FineReader Engine реалізована функція автоматичної класифікації. Вона дозволяє сортувати публікації у вхідних пакетах по заздалегідь заданим класам (наприклад, рахунки, рахунки-фактури, договори, довідки). Нова функція дозволяє класифікувати публікації зі швидкістю до 120 сторінок на хвилину на одне ядро процесора, при цьому може бути досягнута більш висока точність

класифікації, ніж при ручному сортуванні. Розсортовані публікації в подальшому можуть бути збережені в архів, розпізнані, відправлені до відповідних розділів.

Але головним недоліком даної функції є її налаштування. Даний процес виявився трохи складним для звичайного користувача. Щоб налаштувати класифікатор необхідно вручну зібрати приклади публікацій для кожного класу, запустити процедуру автоматичного навчання на зібраній вибірці, при цьому вказавши для кожної публікації свій клас. І тільки після цього класифікатор буде готовий розділяти публікації на класи на які він навчений. Також дана розробка написана тільки під систему Windows.

2.2.3 Морфологічний додаток Mystem

Ще одним прикладом класифікації інформації є Mystem (автори - Ілля Сегалович, Віталій Тітов, компанія Яndex) – це компактний, дуже швидкий і безкоштовний морфологічний додаток російськомовних текстів, реалізований на основі словника А. Залізняка. Доступні для завантаження версії для Windows і Linux. Працює як консольний додаток і має різні режими представлення результатів. Загалом, програма Mystem виробляє морфологічний аналіз літературного нормативного тексту російською мовою. Для слів, відсутніх в словнику, породжуються гіпотези на підставі частотності суфіксів. Але на жаль, в переважній кількості відгуків, які залишають користувачі цієї програми, відзначається складність установки програми і введення потрібних параметрів дослідження.

2.3 Вибір технології розробки

При виборі технологій розробки було розглянуто основні можливі варіанти для побудови підсистеми класифікації публікацій.

Для даної підсистеми можливо побудувати базу даних, автоматизовану інформаційну систему.

Автоматизована система — організаційно-технічна система, що складається із засобів автоматизації певного виду чи кількох видів діяльності людей та персоналу, що здійснює цю діяльність. Автоматизована система (у інформаційних технологіях) — система, що реалізує інформаційну технологію виконання встановлених функцій за допомогою персоналу чи комплексу засобів автоматизації.

У цьому випадку автоматизовані системи розглядаються як інформаційні системи. АС реалізують інформаційну технологію у вигляді певної послідовності інформаційно пов'язаних функцій, завдань або процедур, що виконуються в автоматизованому (інтерактивному) або автоматичному режимах.

Для даної підсистеми було обрано саме автоматизовану підсистему, яка включає в себе функціонал, за допомогою якого реалізується класифікація публікацій, пошук цих публікацій та їх відображення, базу даних, де зберігаються всі необхідні дані для аналізу інформації та збереження результатів, графічний інтерфейс користувача з яким користувач може взаємодіяти.

2.4 Огляд технологій

Для того, щоб реалізувати дану підсистему потрібно застосувати технології об’єктно-орієнтованого програмування, розробки та побудови бази даних,

середовища її реалізації, технології підключення та зв'язку бази даних з функціоналом програми та графічним інтерфейсом.

2.4.1 Мова програмування PHP

PHP – це скриптова мова програмування загального призначення, інтенсивно застосовується для розробки веб-додатків. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів. Ця мова і його інтерпретатор розробляються групою ентузіастів в рамках проекту з відкритим кодом. Проект поширюється під власною ліцензією. Популярність в області побудови веб-сайтів визначається наявністю великого набору вбудованих засобів для розробки веб-додатків. Основні з них:

- автоматичне вилучення POST і GET-параметрів, а також змінних оточення веб-сервера в зумовлені масиви;
- взаємодія з великою кількістю різних систем управління базами даних (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Paradox File Access, MaxDB, Інтерфейс PDO та багато інших);
- автоматизована відправка HTTP-заголовків;
- робота з HTTP-авторизацією;
- робота з локальними і віддаленими файлами, сокетами;
- обробка файлів, що завантажуються на сервер.

В даний час PHP використовується сотнями тисяч розробників. Згідно з рейтингом корпорації TIOBE, що базується на даних пошукових систем, у вересні 2015 року PHP знаходився на 6 місці серед мов програмування. До найбільших сайтів, які використовують PHP, відносяться Facebook, Wikipedia та ін.

2.4.2 Мова програмування JavaScript

JavaScript – прототипно-орієнтована сценарна мова програмування. Є реалізацією мови ECMAScript. JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів – додатків. Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності веб-сторінок. Основні архітектурні риси: динамічна типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу. На JavaScript вплинули багато мов, при розробці була мета зробити мову схожою на Java, але при цьому легким для використання не програмістів.

JavaScript є об'єктно-орієнтованою мовою, що має ряд властивостей, властивих функціональним мовам – функції як об'єкти першого класу, об'єкти як списки, анонімні функції, замикання, що додає мові додаткову гнучкість. Незважаючи на схожий з C синтаксис, JavaScript в порівнянні з мовою C має корінні відмінності:

- об'єкти, з можливістю інтроспекції;
- функції як об'єкти першого класу;
- автоматичне приведення типів;
- автоматичне прибирання сміття;
- анонімні функції.

У мові відсутні такі корисні речі як: стандартні інтерфейси до веб-серверів і баз даних; система управління пакетами, яка б відстежувала залежності і автоматично встановлювала їх. JavaScript використовується в клієнтській частини веб-додатків, клієнт-серверних програм, в якому клієнтом є браузер, а сервером – веб-сервер, що мають розподілену між сервером і клієнтом логіку. Обмін інформацією в веб-додатках відбувається по мережі. Одним з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб-додатки є крос-платформними сервісами.

2.4.3 Огляд СКБД

Oracle – об'єктно-реляційна система керування базами даних від Oracle Corporation. Ця СКБД має великі та потужні можливості не лише для зберігання даних, а і для їх захисту, аналітики та реорганізації.

Переваги застосування для поставленої задачі:

- надійність та захищеність системи;
- можливе застосування вбудованих аналітичних функцій для подальшого аналізу історії класифікацій публікацій.

До недоліків відноситься надлишковий функціонал СКБД, необхідність мати адміністратора баз даних, що не є обов‘язковим для невеликої серверної аплікації.

Інша популярна система управління базами даних (СКБД) є MySQL – дуже часто застосовується в поєднанні з PHP.

MySQL – це програмне забезпечення з відкритим кодом. Застосовувати його і модифікувати може будь-хто. Таке програмне забезпечення можна отримувати по Internet і використовувати безкоштовно. При цьому кожен користувач може вивчити вихідний код і змінити його відповідно до своїх потреб. MySQL є дуже швидким, надійним і легким у використанні.

Основні характеристики MySQL:

- багатопоточність. Підтримка декількох одночасних запитів;
- оптимізація зв’язків з приєднанням багатьох даних за один прохід;
- записи фіксованої і змінної довжини;
- ODBC драйвер в комплекті з вихідними файлами;
- гнучка система привілеїв і паролів;
- підтримка ключових полів і спеціальних полів в операторі CREATE;
- заснована на потоках, швидка система пам’яті;
- утиліта перевірки і ремонту таблиці;
- всі дані зберігаються у форматі ISO8859_1;

- всі операції роботи з рядками не звертають уваги на регістр символів в оброблюваних рядках;

- всі поля мають значення за замовчуванням. INSERT можна використовувати на будь-якому підмножині полів;

- легкість керування таблицею, включаючи додавання і видалення ключів і полів.

MySQL є системою клієнт-сервер, яка містить багатопоточний SQL-сервер, що забезпечує підтримку різних обчислювальних машин баз даних, а також кілька різних клієнтських програм і бібліотек, засоби адміністрування і широкий спектр програмних інтерфейсів (API). Можливо також представити сервер MySQL у вигляді багатопотокової бібліотеки, яку можна підключити до призначеного для користувача додатку і отримати компактний, більш швидкий і легкий в управлінні продукт.

2.4.4 MySQL Workbench

MySQL Workbench – інструмент для візуального проектування баз даних, що інтегрує проектування, моделювання, створення та експлуатацію БД в єдине безкоштовне оточення для системи баз даних MySQL.

Можливості програми:

- дозволяє наочно уявити модель бази даних в графічному вигляді;

- наочний і функціональний механізм установки зв'язків між таблицями, в тому числі «багато до багатьох» зі створенням таблиці зв'язків;

- Reverse Engineering - відновлення структури таблиць з уже існуючою на сервері БД;

- зручний редактор SQL запитів, що дозволяє відразу ж відправляти їх із сервером і отримати відповідь у вигляді таблиці;

- можливість редагування даних в таблиці у візуальному режимі.

MySQL Workbench пропонується в двох редакціях:

- Community Edition - поширюється під вільною ліцензією GNU GPL;
- Standard Edition - доступна по щорічній оплачуваній підписці. Ця версія включає в себе додаткові функції, які підвищують продуктивність розробників і адміністраторів БД.

2.5 Огляд математичних методів

2.5.1 Формалізація задачі

Задача класифікації тексту може бути формалізована як задача апроксимації невідомої функції $\Phi: D \times C \rightarrow \{0, 1\}$ (яким чином публікації повинні бути класифіковані) через функцію $\widehat{\Phi}: D \times C \rightarrow \{0, 1\}$, що названо класифікатором, де $C = \{c_1, \dots, c_{|C|}\}$ – множина можливих категорій, а $D = \{d_1, \dots, d_{|D|}\}$ – множина публікацій.

$$\Phi\,(d_j, c_i) = \begin{cases} 0\,, & \text{якщо } d_j \notin c_i; \\ 1, & \text{якщо } d_j \in c_i\,. \end{cases} \tag{2.1}$$

Публікація d_j називають додатним прикладом категорії c_i, якщо $\Phi\,(d_j, c_i) = 1$, і від'ємним в противному випадку.

Якщо в задачі кожній публікації $d_j \in D$ може відповідати лише одна категорія $c_i \in C$, то має місце однозначна класифікація, а якщо довільна кількість категорій $0 < n_j < |C|$ – множинна класифікація.

Виділяють окремий вид класифікаторів – бінарні (двійкові), множина категорій яких складається із двох елементів (c_i та його довонення $\overline{c_i}$). Бінарний класифікатор для $\{c_i, \overline{c_i}\}$ визначається функцією $\widehat{\Phi}: D \rightarrow \{0, 1\}$, яка являється апроксимацією невідомої функції $\Phi: D \rightarrow \{0, 1\}$.

Знаходження класифікатора для множини категорій $C = \{c_1, \dots, c_{|C|}\}$ зазвичай розглядають як пошук $|C|$ бінарних класифікаторів $\{c_i, \overline{c_i}\}$, де $i = 1, \dots, |C|$.Таким чином, класифікатор $\widehat{\Phi}$ представляє собою множину бінарних класифікаторів.

Зазвичай при класифікації, публікація представляє собою вектор в деякому просторі (простір ознак), в якому кожному терму (ознаці) ставиться у відповідність його вага (значимість):

$$\overrightarrow{d_j} = \langle \omega_{1j,\dots,}\omega_{|T|j} \rangle,$$

де T – словник, тобто множина термів, які зустрічаються в $|\mathcal{L}|$ навчаючих класифікатором публікацій, та $0 \leq \omega_{kj} \leq 1$ визначає значимість терму t_k в публікації d_j.

Термами являються слова, що зустрічаються в публікації. Ці слова, зазвичай, піддаються морфологічному розбіру чи стемінгу. Вага може просто визначити наявність терму в публікації, в такому випадку $\omega_{kj} \in \{0,1\}$. Частіше, у якості вагових значень використовуються дійсні числа із діапазону $0 \leq \omega_{kj} \leq 1$. Такі ваги мають статичну чи ймовірнісну природу і мають залежність від методу побудови класифікатора. Найбільш популярним класом статичних вагових функцій являється $tf * idf$, в якому визначено: чим частіше зустрічається терм t_k в публікації d_j,тим більш значим він в ньому. Один із варіантів $tf * idf$:

$$\omega_{ij} = \frac{tf_{ij} \cdot idf_i}{\sqrt{\sum_k (tf_{kj} \cdot idf_k)^2}}, \tag{2.2}$$

де ω_{ij} – вага i – того терма в публікації d_j;

tf_{ij} – частота зустрічі i – того терма в даній публікації;

$idf_i = \frac{logN}{n}$ – логарифм відношення кількості публікацій в колекції до кількості публікацій, в яких зустрічається i – тий терм.

Ваги, що обчислені по (2.2) нормалізовані таким чином, що сума квадратів ваг кожної публікації рівна одиниці.

2.5.2 Порівняльний аналіз існуючих методів

На даний час вже існує безліч математичних методів для побудови класифікаторів автоматичної класифікації публікацій. Одні з них будують бінарні функції $\widehat{\Phi}$: $D \times C \rightarrow \{0, 1\}$, а деякі – дійсні функції CSV: $D \times C \rightarrow [0, 1]$ (Categorization Status Value). Якщо використовуються перші, то має місце точна класифікація, якщо другі – порогова класифікація. Для останніх необхідно визначити множину порогових значень τ_i при $i = 1, \dots, |C|$ (розв'язуються експериментально на навчальному наборі), які дозволяють розглядати дійсні значення CSV як бінарні:

$$\widehat{\Phi}\,(d_j, c_i) = \begin{cases} 0\,, \text{ якщо } CSV_i\left(d_j\right) < \tau_i; \\ 1, \text{якщо } CSV_i\left(d_j\right) \geq \tau_i\,. \end{cases} \tag{2.3}$$

Слід відзначити, що в деяких випадках дійсні функції можуть з успіхом використовуватись без необхідності перетворення в бінарні. Наприклад, класифікатори з такими функціями можуть будувати «рейтинг» категорій для публікацій.

Для гарної класифікації публікацій, даний класифікатор має задовольняти певним основним критеріям, а саме:

- чи усі класи в тренувальній вибірці;
- чи передбачено в методі використання декількох класів, чи є вони невиключеними;

- оверфіттинг і локальна визначеність;
- відповідність навчальній вибірці;
- чутливість методу до початкової вибірки;
- Постійне навчання та складність методу.

Розглянемо декілька основних класичних методів побудови текстових класифікаторів, що можуть слугувати відправною точкою для розробки більш ефективних методик, та порівняємо їх.

2.5.2.1 Метод Байєса (Naive Bayes)

Через припущення незалежності ознак такий класифікатор називають «Наївним байєсовим» класифікатором (Naive Bayes Classifier).

Наївний байєсовий класифікатор – простий ймовірнісний класифікатор, заснований на застосуванні теореми Байєса зі строгими (наївними) припущеннями про незалежність.

Залежно від точної природи ймовірнісної моделі, наївні байєсові класифікатори можуть вивчатись дуже ефективно. У багатьох практичних додатках для оцінки параметрів для наївних байєсових моделей використовують метод максимальної правдоподібності; іншими словами, можна працювати з наївною байєсівською моделлю, не вірячи в байєсову ймовірність і не використовуючи байєсовські методи.

В ймовірнісному класифікаторі використовується векторне представлення публікацій, а функції $CSV_i\left(d_j\right)$ розглядаються в термах умовних ймовірностях $P(c_i \mid \vec{d}_j)$ при $i = 1, \dots, |C|$ і знаходженні найбільшої такої ймовірності:

$$H\left(\vec{d}_j\right) = argmax\, P(c_i \mid \vec{d}_j),\ c_i \in C. \tag{2.4}$$

Умовну вірогідність $P(c_i \mid \vec{d}_j)$ згідно теоремі Байєса можна переписати як

$$P(c_i \mid \vec{d}_j) = \frac{P(\vec{d}_j \mid c_i) \cdot P(c_i)}{P(\vec{d}_j)}, \tag{2.5}$$

де $P(c_i)$ – це апріорна вірогідність того, що публікація відноситься до категорії ;

$P(\vec{d}_j \mid c_i)$ – вірогідність знайти публікацію, що представлена вектором $\vec{d}_j$ в категорії c_i ;

$P(\vec{d}_j)$ – вірогідність того, що випадково обрана публікація буде мати вектор $\vec{d}_j$.

По суті $P(c_i)$ являється відношенням кількості публікацій із навчаючої вибірки $\mathcal{L}$, віднесених до категорії c_i , до кількості всіх публікацій із $\mathcal{L}$:

$$P(c_i) = \frac{|\{d_j \in \mathcal{L} | d_j \in c_i\}|}{|\mathcal{L}|}. \tag{2.6}$$

Щоб визначити $P(\vec{d}_j \mid c_i)$ та $P(\vec{d}_j)$ необхідно зробити припущення про те, що надходження термів в публікацію залежить від категорії, но не залежить від інших термів цієї публікації. Таким чином, $P(\vec{d}_j \mid c_i)$ можна записати як:

$$P(\vec{d}_j \mid c_i) = \prod_{k=1}^{|T|} P(\omega_{kj} \mid c_i). \tag{2.7}$$

В свою чергу $P(\omega_{kj} \mid c_i)$ можна визначити як відношення кількості публікацій із навчаючої вибірки $\mathcal{L}$, віднесених до категорії c_i і, маючи терм t_k до загальної кількості публікацій із $\mathcal{L}$, віднесених до категорії c_i:

$$P(\omega_{kj} \mid c_i) = \frac{|\{d_j \in \mathcal{L} | d_j \in c_i, t_k \in d_j\}|}{|\{d_j \in \mathcal{L} | d_j \in c_i\}|}. \tag{2.8}$$

Незважаючи на наївний вигляд і, безсумнівно, дуже спрощені умови, наївні байєсовські класифікатори часто працюють набагато краще в багатьох складних життєвих ситуаціях. Згідно усіх зазначених критеріїв, наївний байєсовий класифікатор відповідає усім цим вимогам.

2.5.2.2 Метод Rocchio

Використовувати метод Rocchio для побудови лінійного класифікатора публікацій вперше було запропоновано в роботі [3]. Для кожної категорії c_i вираховуємо вектор $\vec{c}_i = \langle \omega_{1i}, \dots, \omega_{|T|i} \rangle$ за формулою:

$$\omega_{ki} = \beta \cdot \sum_{d_j \epsilon D_i^+} \frac{\omega_{ki}}{|D_i^+|} - \gamma \cdot \sum_{d_j \epsilon D_i^+} \frac{\omega_{ki}}{|D_i^-|}, \tag{2.9}$$

де ω_{kj} – вага терму t_k в публікації d_j, $D_i^+ = \{d \in \mathcal{L} | \Phi(d_j, c_i) = 1\}$ та $D_i^- = \{d \in \mathcal{L} | \Phi(d_j, c_i) = 0\}$.

Параметри β і γ визначають значимість додатних і від'ємних прикладів. У разі, коли $\beta = 1$ та $\gamma = 0$, вектор $\vec{c}_i$ буде являтись центроїдою додатних прикладів категорії c_i.

В даному методі немає гарантії, що визначений клас відповідатиме навчальній вибірці, велика кількість хибних результатів для виключних класів.

2.5.2.3 Метод k - найближчих сусідів

Метод k - найближчих сусідів (k-Nearest Neighbors, k-NN) – метричний алгоритм для автоматичної класифікації об'єктів. Основним принципом методу найближчих сусідів є те, що об'єкт присвоюється тому класу, який є найбільш поширеним серед сусідів даного елемента. Сусіди беруться виходячи з безлічі об'єктів, класи яких вже відомі, і, виходячи з ключового для даного методу значення k вираховується, який клас найбільш численний серед них. Кожен об'єкт має кінцеву кількість атрибутів (розмірностей). Передбачається, що існує певний набір об'єктів з уже наявною класифікацією.

Метод k - найближчих сусідів (k-Nearest Neighbors, k-NN) на відміну від інших деяких класифікаторів, не потребує великого вивчення. Для того аби найти категорії, до яких відноситься публікація d_j, класифікатор порівнює d_j зі всіма публікаціями із навчаючої вибірки $\mathcal{L}$: для кожного $d_z \in \mathcal{L}$ вираховується «відстань» $p(d_j, d_z)$. Далі із навчаючої вибірки вибираються k публікацій найближчих до d_j.

Для категорій визначаються функції ранжування $CSV_i\left(d_j\right)$ за формулою:

$$\sum_{d_z \in \mathcal{L}_k(d_j)} p\left(d_j, d_z\right) \cdot \Phi(d_z, c_i), \tag{2.10}$$

де $\mathcal{L}_k\left(d_j\right)$ – це найближчі k публікації із $\mathcal{L}$ до d_j.

Параметр k зазвичай обирається в інтервалі від 20 до 50. Публікації d_j визначені в категорії, для яких $CSV_i\left(d_j\right) \geq \tau_i$.

Даний метод дає високу ефективність, але при цьому дуже вимогливий до обчислювальних ресурсів на етапі класифікації. А також, вибір параметру k сильно впливає на якість класифікації при цьому, що немає єдиного способу підібрати правильно параметр k. Локальна визначеність - межові випадки сильно впливають на визначення класу.

2.5.2.4 Метод опорних векторів

Метод опорних векторів (Support vector machine, SVM) — метод класифікації, належить до групи граничних методів; визначає класи за допомогою меж просторів. Опорними векторами вважаються об'єкти множини, що лежать на цих межах. Класифікація вважається вдалою, якщо простір між межами порожній. SVM-класифікатор реалізує бінарну класифікацію, тобто розділяє безліч вхідних векторів на дві частини: позитивний-негативний, свої-чужі, так-ні і т. п.

Даний метод був розроблений Володимиром Вапніком в 1995 році на основі принципу структурної мінімізації ризику – одночасного контролю кількості помилок класифікації на множині для навчання та «ступеню узагальнення» виявлених залежностей. Метод вперше був застосований до задачі класифікації текстів Йоахімсом в 1998 році. В своєму початковому вигляді метод опорних векторів вирішував задачу розрізнення об’єктів двох класів.

SVM - це набір схожих алгоритмів виду «навчання із вчителем». Ці алгоритми зазвичай використовуються для задач класифікації та регресійного аналізу. Метод належить до розряду лінійних класифікаторів. Особливою властивістю методу опорних векторів є безперервне зменшення емпіричної помилки класифікації та збільшення проміжку. Тому цей метод також відомий як метод класифікатора з максимальним проміжком. Основна ідея методу опорних векторів – перевід вихідних векторів у простір більш високої розмірності та пошук роздільної гіперплощини з максимальним проміжком у цьому просторі. Дві паралельні гіперплощини будуються по обидва боки гіперплощини, що розділяє наші класи. Роздільною гіперплощиною буде та, що максимізує відстань до двох паралельних гіперплощини. Алгоритм працює у припущенні, що чим більша різниця або відстань між цими паралельними гіперплощинами, тим меншою буде середня помилка класифікатора. Для класифікатора на базі методу опорних векторів, як правило, не потрібно зменшувати розмірність простору ознак, вони стійкі і добре масштабуються [4]. Серед основних

недоліків методу – багато обчислень при великій кількості категорій та необхідні розмічені приклади для всіх класів.

2.5.2.5 Метод дерев прийняття рішень

Метод дерев прийняття рішень (дерево класифікації, Decision Trees) – це засіб підтримки прийняття рішень, що використовується в статистиці і аналізі даних для прогнозування моделей. Структура дерева являє собою «листя» і «гілки». На ребрах («гілках») дерева рішення записані атрибути, від яких залежить цільова функція, в «листі» записані значення цільової функції, а в інших вузлах - атрибути, за якими розрізняються випадки. Щоб класифікувати новий випадок, треба спуститися по дереву до листа і видати відповідне значення. Подібні дерева рішень широко використовуються в інтелектуальному аналізі даних. Мета полягає в тому, щоб створити модель, яка передбачає значення цільової змінної на основі декількох змінних на вході.

Класифікатор на базі дерев рішень (Decision Trees) для категорії c_i представляє собою дерево, вузлами якого являються терми t_k, кожне ребро позначене умовою $\geq v_k$ або $< v_k$, а листя позначені як c_i або $\overline{c_i}$. Щоб класифікувати публікацію d_j до категорії c_i або $\overline{c_i}$, необхідно пройти по вузлам дерева, починаючи від кореня, порівнюючи ваги терму в публікації ω_{kj} зі значенням v_k на ребрах. На практиці зазвичай використовуються бінарні дерева рішень, в яких прийняття рішень переходу по ребрам здійснюється простою перевіркою наявності терму в публікації.

Один із способів автоматичної побудови дерев рішень полягає в послідовному розбитті множини навчаючих публікацій $\mathcal{L}$ на класи до тих пір, поки у класі не залишиться публікацій, визначених тільки в одну із категорій c_i або $\overline{c_i}$. На кожному етапі в якості вузла дерева обирається терм t_k і визначається v_k, потім множина публікацій розбивається на два класи: $\omega_k \geq v_k$ та $\omega_k < v_k$.

Зазвичай, побудова класифікатора методом дерев рішень являється сильно деталізованим (ефект перенавчання), тому застосовуються різні алгоритми відсікання дерева. Широке застосування отримали алгоритми ID3 [5] та C4.5 [6]. Також є такі недоліки, як проблема оверфітингу: деякі гілки вибору можуть бути надто складними відносно всього дерева, немає чіткого механізму підбору стратегії побудови рішень, для включення нових тренувальних даних необхідна перебудова усього дерева.

2.5.2.6 Нейронні мережі

Нейронні мережі (Neural networks) — математичні моделі, а також їхня програмна та апаратна реалізація, побудовані за принципом функціонування біологічних нейронних мереж — мереж нервових клітин живого організму. Системи, архітектура і принцип дії базується на аналогії з мозком живих істот. Ключовим елементом цих систем виступає штучний нейрон як імітаційна модель нервової клітини мозку — біологічного нейрона. Цей термін виник при вивченні процесів, які відбуваються в мозку, та при спробі змоделювати ці процеси. Першою такою спробою були нейронні мережі Маккалока і Піттса. Як наслідок, після розробки алгоритмів навчання, отримані моделі стали використовуватися в практичних цілях: в задачах прогнозування, для розпізнавання образів, в задачах керування та інші.

І тим не менш, будучи з'єднаними в досить велику мережу з керованою взаємодією, такі локально прості процесори разом здатні виконувати достатньо складні завдання. З точки зору машинного навчання, нейронна мережа є окремим випадком методів розпізнавання образів, дискримінантного аналізу, методів кластеризації тощо.

Базовою одиницею нейронних мереж є нейрон. Кожний нейрон при класифікації публікацій отримує набір входів, які будемо позначати d_j. В нашій задачі

будуть означати вхід термів в j-у публікацію. Кожний нейрон також асоційований з набором ваг ω, які використовуються для підрахунку функції входів f. Типовою такою функцією, що використовується в нейронній мережі, є лінійна:

$$p_j = \omega \cdot d_j. \tag{2.11}$$

Таким чином, для вектора d_j і лексикону з d слів, вектор ваг ω повинен також містити d елементів.

Тепер розглянемо задачу бінарної класифікації, в якій мітки будуть із множини {+1, -1}. Припустимо також, що класом d_j являється y_j. У такому випадку, знак функції p_j буде визначати як раз мітку класа. Головною ідеєю навчання є спочатку довільний вибір ваг, а затим покрокове оновлення при наявності помилок функції на тренувальних даних. «Силу» поновлення на кожному кроці буде визначати параметр μ, що є швидкістю навчання. Разом, ми отримали так званий персептронний алгоритм:

1. вхід: швидкість навчання μ, навчаюча вибірка (d_j, y_j), $j = 1 \dots J$;

2. вихід: ваги ω;

3. для $j = 1 \dots n$;

4. якщо знак $\omega \cdot d_j$ не однаковий з y_j, то

5. обновити ваги ω відповідно з швидкістю навчання μ;

6. кінець умови

7. кінець циклу,

8. поки усі ваги не стабілізуються.

Ваги ω, зазвичай, змінюються на величину, пропорційну $\mu \cdot d_j$. Також можна змінювати ваги кожного разу на μ, а не на $\mu \cdot d_j$. Це розумно робити в області класифікації текстів, де всі ознаки мають невеликі невід'ємні значення. Кілька

реалізацій методів на основі нейронних мереж були запропоновані в [7, 8, 9]. Природним питанням є те, як же використовувати нейронну мережу, коли класи можуть бути лінійно нероздільні. У цьому випадку, якщо використовувати багатошарові нейронні мережі, можна отримати більш повні класи розділяючих поверхонь. У таких мережах виходи одного шару подаються на входи нейронів наступного шару. Двох шарів досить для наближення скільки завгодно складних структур з багатокутників. Найпоширенішим алгоритмом навчання являється алгоритм зворотного поширення помилки [10]. Проте, експерименти показали [9], що для класифікації текстів розгляд багатошарових нейронний мереж не дає значного виграшу перед одношаровим персептроном. Нейронні мережі є ефективними для побудови простих лінійних класифікаторів, однак мають складні обчислення та сильно залежать від структури навчальної вибірки.

2.5.3 Вибір і обґрунтування методів рішення

При виборі математичного методу для побудови класифікатора враховувались усі переваги та недоліки кожного з запропонованих методів. Для більшої наглядності зведем результати переваг та недоліків до таблиці 2.1 порівнянь за основними критеріями відбору. Дані критерії були представлені у підрозділі 2.5.2.

Таблиця 2.1 – Порівняння математичних методів за критеріями

Метод\ Критерій	Bayes	Rocchio	K-NN	SVM	Des. trees	Neural networks
Усі класи в трен. вибірці	+	+	+	+	+	+
Неодноз. клас-ція	+	+	-	-	-	+
Невиключні класи	+	+	-	-	-	+

Продовження таблиці 2.1

Метод\ Критерій	Bayes	Rocchio	K-NN	SVM	Des. trees	Neural networks
Відповідність навчальній вибірці	+	-	+/-	+	+	+/-
Оверфіттинг	+	+	-	+/-	-	+/-
Чутливість до почат. вибірки	+	+	+	+	-	-
Постійне навчання	+	+	+/-	+	-	+
Обчислювальна складність	+	+	+	-	+	+
Всього	8	7	4	4,5	3	6

Виходячи із результатів, що подані у таблиці 2.1, для розв'язання поставленої задачі було вирішено обрати метод Байєса, на основі якого побудовано байєсовий класифікатор, оскільки у сумі за отриманням додатного результату виявились найбільшими, що наштовхують на його детальне вивчення та побудову.

2.6 Висновки до розділу

Кожний із розглянутих підходів класифікації публікацій має свої переваги та недоліки. У якості дослідження візьмемо автоматичну класифікацію, оскільки проблема автоматичної класифікації текстів є в наш час дуже актуальною. Є велика кількість галузей, у яких може бути використана саме автоматична класифікація. Було розглянуто вже існуючі рішення автоматичної класифікації публікацій. Але всі вони мають свої недоліки, серед них - застаріла система класифікації, що має замалу кількість класів публікацій, вузький пошук цих публікацій, складність установки програм для звичайного користувача. Всі ці недоліки можна уникнути шляхом обрання вдалих середовищ розробки для усунення складності встановлення

додаткових програм для класифікації. Замалу кількість класів публікацій можна виправити, використавши для навчання набір даних більшого об‘єму. А також розробити більш розширений пошук класифікованих публікацій.

Переваги та недоліки зазначених методів дозволяють зробити висновок щодо необхідності подальшого вдосконалення алгоритмів класифікації на основі зазначених методів, що були б простими в реалізації, ефективними, мали низькі обчислювальні витрати при навчанні та високу якість класифікації в реальних завданнях.

На основі порівняльного аналізу (таблиці 2.1) можна сказати, що підхід із застосуванням байєсових мереж довіри в середньому краще задовольняє основні критерії при побудові класифікатора, що може бути зумовлено ліпшим математичним описанням поставленої задачі, оскільки даний підхід найкраще враховує нечіткість даних, із якими працює. Тому, враховуючи усі переваги та недоліки, як класифікатор було обрано модифікований наївний байєсовий алгоритм, з модифікаціями для *n* категорій.

У подальшому вибраний метод може бути вдосконалений завдяки поліпшеному вибору термів на етапі індексації публікації. Проблематичним є використання описаної методики для нетекстових даних, наприклад, для зображень. Це пояснюється тим, що для нетекстових даних є дуже велика кількість різноманітних поєднань термів, в той час як для текстових даних така кількість є порівняно невеликою. Але якщо буде вирішена проблема індексації мультимедійних даних, описану методику можна буде застосувати і для не текстових даних.

Для реалізації даної задачі, в якості головної мови програмування було використано PHP версії 5.5 для генерації сторінок на сервері, з підтримкою СКБД MySQL версії 5.7. В якості мови програмування, що виконується на стороні користувача, використовувався JavaScript. MySQL Workbench було використано як середовище розробки, що містить кейс-засоби і засоби адміністрування БД.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ КЛАСИФІКАЦІЇ ПУБЛІКАЦІЙ

3.1 Формулювання математичної частини класифікації публікацій обраним методом

У розділі 2 в загальних рисах описана задача класифікації публікацій, а також традиційні підходи та методи, що використовуються в задачах класифікації. В даному розділі більш детально описано вибраний метод класифікації публікацій, як працює один із оптимальних, і в той же час, один із часто використаних при обробці натуральних мов алгоритм класифікації – наївний байеєсовий класифікатор.

3.2 Побудова алгоритму класифікації

В основі байєсового алгоритму класифікації лежить теорема Байєса:

$$P(c|d) = \frac{P(d|c) \cdot P(c)}{P(d)}, \tag{3.1}$$

де $P(c|d)$ – ймовірність, що публікація d належить класу c, яку і треба знайти;

$P(d|c)$ – ймовірність зустріти публікацію d серед усіх публікацій класу c;

$P(c)$ – безумовна ймовірність зустріти публікацію класу c серед публікацій;

$P(d)$ – безумовна ймовірність публікації d серед публікацій.

Теорема Байєса дозволяє переставити місцями причину та наслідок. Знаючи з якою ймовірністю причина приводить до деякої події, ця теорема дозволяє

обрахувати ймовірність того, що саме ця причина привела до події, яка спостерігалась.

Мета класифікації публікацій полягає в тому, щоб зрозуміти до якого класу належить публікація, тому, в даному випадку, потрібна не сама ймовірність, а найбільш ймовірний клас. Байєсовий класифікатор виконує оцінку апостеріорного максимуму (Maximum a posteriori estimation) для визначення найбільш вірогідного класу. Грубо кажучи, це клас із найбільшою ймовірністю:

$$c_{max} = argmax_{c \in C} \frac{P(d|c) \cdot P(c)}{P(d)}, \tag{3.2}$$

Тобто потрібно обчислити ймовірність для усіх класів і вибрати той, який є максимальним. Знаменник (ймовірність публікації) є константою і ніяк не впливає на ранжування класів, тому ми можемо ним знехтувати.

Формула (3.2) набуде іншого вигляду:

$$c_{max} = argmax_{c \in C}[P(d|c) \cdot P(c)]. \tag{3.3}$$

Байєсовий класифікатор представляє публікацію як набір слів ймовірності які умовно не залежать один від одного. Цей підхід ще має назву «bag of words model». Виходячи з цього, умовна ймовірність публікації апроксимується добутком умовних ймовірностей усіх слів, що входять до публікації.

$$P(d|c) \approx P(\omega_1|c) \cdot P(\omega_2|c) \dots P(\omega_n|c) = \prod_{i=1}^{n} P(\omega_i|c). \tag{3.4}$$

Підставив отримані дані у формулу (3.3), ми маємо:

$$c_{max} = argmax_{c \in C}[P(c) \cdot \prod_{i=1}^{n} P(\omega_i|c)]. \tag{3.5}$$

Таким чином, отримана дана формула описує математичний алгоритм побудови класифікатора.

3.2.1 Проблема арифметичного переповнення

При достатньо великій довжині публікацій потрібно перемножити велику кількість дуже малих чисел. Щоб цього уникнути арифметичного переповнення знизу, зазвичай користуються властивістю логарифма добутку $logab = loga + logb$. Так як логарифм це монотонна функція, його застосування до обох частин виразу змінить тільки його чисельне значення, але не параметри, при яких досягається максимум. При цьому, логарифм від числа близького до нуля буде числом від'ємним, але в абсолютному значенні істотно більшим ніж вихідне число, що робить логарифмічні значення ймовірностей більш вдалими для аналізу. Тому, переписуємо нашу формулу з використанням логарифму:

$$c_{max} = argmax_{c \in C}[logP(c) \cdot \sum_{i=1}^{n} logP(\omega_i|c)]. \quad (3.6)$$

Основа логарифму, в даному випадку, не має значення. Можна використовувати як натуральний, так і будь-який інший логарифм.

3.2.2 Оцінка параметрів моделі Байєса

Оцінка ймовірностей $P(c)$ та $P(\omega_i|c)$ визначається на навчаючій вибірці. Ймовірність класу можна оцінити як:

$$P(c) = \frac{D_c}{D}, \tag{3.7}$$

де D_c – кількість публікацій, що належать класу c, а D – загальна кількість публікацій в навчаючій вибірці.

Оцінка ймовірності слова у класі може вираховуватись наступним чином:

$$P(\omega_i|c) = \frac{W_{ic}}{\sum_{i' \epsilon V} W_{i'c}}, \tag{3.8}$$

де, W_{ic} – кількість разів зустрічі i – того слова в публікації класу с;

V – словник публікацій (список усіх унікальних слів).

Іншими словами, чисельник описує скільки разів слово зустрічається в публікації певного класу, а знаменник – сумарну кількість слів в усіх публікаціях даного класу.

3.2.3 Проблема невідомих слів

У формулі (3.8) є одна невелика проблема. Якщо на етапі класифікації зустрінеться слово, якого не було на етапі навчання, то значення W_{ic}, а слідчо і $P(\omega_i|c)$ дорівнюватимуть нулю. Це призведе до того що публікацію з цим словом не можна буде класифікувати, так як він буде мати нульову ймовірність по всіх класах. Позбутися від цієї проблеми шляхом аналізу більшої кількості публікацій не вийде. Ніколи не можна скласти навчальну вибірку, що містить всі можливі слова включаючи неологізми, друкарські помилки, синоніми і т.п. Типовим рішенням проблеми невідомих слів є адитивне згладжування (згладжування Лапласа). Ідея полягає в тому, що ми припускаємо, начебто бачили кожне слово на один раз більше, тобто додаємо одиницю до частоти кожного слова.

$$P(\omega_i|c) = \frac{W_{ic}+1}{\sum_{i'\epsilon V}(W_{i'c}+1)} = \frac{W_{ic}+1}{|V|\sum_{i'\epsilon V} W_{i'c}}. \quad (3.9)$$

Даний підхід змішує оцінку ймовірностей в сторону менш вірогідних результатів. Таким чином, слова які ми не бачили на етапі навчання моделі отримують нехай маленьку, але все-таки не нульову ймовірність.

Припустимо, в нашому випадку, на етапі навчання класифікатор бачив три імені вказану кількість разів (таблиця 3.1).

Таблиця 3.1 – Частота зустрічі слів у публікації

Ім'я	Частота зустрічі слова
Василь	3
Петро	2
Євген	1

І нехай на етапі класифікації зустрічається ім'я Інокентій, якого не було ні разу на етапі навчання. Тоді оригінальна і зміщена по Лапласу оцінка ймовірностей буде виглядати наступним чином:

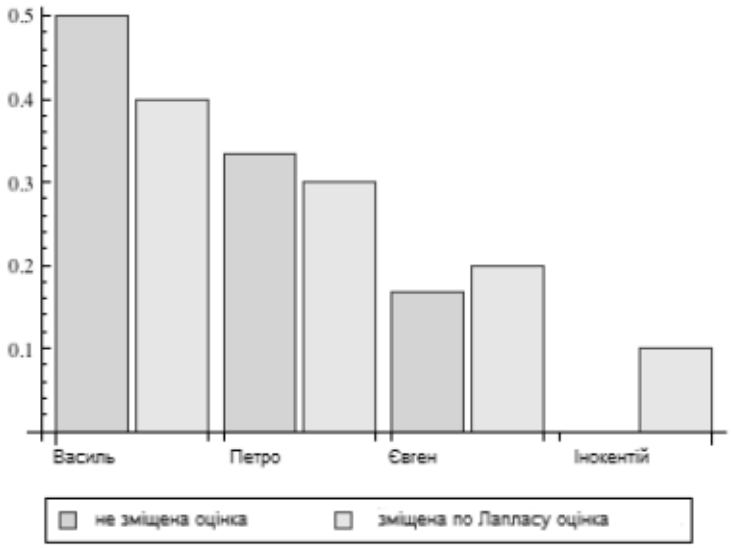

Рисунок 3.1 – Графік зміщення оцінки

З графіка видно, що зміщена оцінка ніколи не буває нульовою, що захищає нас від проблеми невідомих слів. Враховуючи цей факт, отримаєм:

$$c_{max} = argmax_{c \in C} \left[log \frac{D_c}{D} + \sum_{i=1}^{n} log \frac{W_{ic}+1}{|V| \sum_{i' \epsilon V} W_{i'c}} \right] \quad (3.10)$$

В даній формулі підставили обрані оцінки в формулу (3.6), звідки і отримали остаточну формулу за якою відбувається байєсова класифікація.

3.3 Реалізація класифікатора

Для реалізації Байєсового класифікатора необхідна була навчальна вибірка, в якій проставлені відповідності між текстовими публікаціями і їх класами. Потім зібрано наступну статистику з вибірки, яка використовувалась на етапі класифікації:

- відносні частоти класів в публікаціях. Тобто, як часто зустрічаються публікації того чи іншого класу;
- сумарна кількість слів у публікаціях кожного класу;
- відносні частоти слів у межах кожного класу;
- розмір словника вибірки. Кількість унікальних слів у вибірці.

Сукупність цієї інформації називають моделлю класифікатора. Потім на етапі класифікації необхідно для кожного класу розрахувати значення наступного виразу і вибрати клас з максимальним значенням. Для зручності спростили формулу (3.10):

$$log \frac{D_c}{D} + \sum_{i \in Q} log \frac{W_{ic}+1}{|V|+L_c}, \quad (3.11)$$

де в цій формулі: $D_c -$ кількість публікацій в навчальній вибірці, що належать класу c;

D – загальна кількість публікацій в навчальній вибірці;

$|V|$ – кількість унікальних слів у всіх публікаціях навчальної вибірки;

L_c – сумарна кількість слів у публікаціях класу c в навчальній вибірці;

W_{ic} – скільки разів i – те слово зустрічалося в публікаціях класу c в навчальній вибірці;

Q – безліч слів класифікованої публікації.

3.4 Висновки до розділу

В даному розділі було побудовано основну математичну модель, що потрібна для реалізації підсистеми класифікації публікацій. В якості формування математичного алгоритму був обраний байеєсовий класифікатор. Даний алгоритм є одним із оптимальних алгоритмів класифікації, що відповідає усім висунутим критеріям. При вирішені задач, що пов'язані із класифікацією публікацій, байєсовий алгоритм перевершує багатьох інших. Завдяки цьому, даний метод знаходить широке застосування в області фільтрування спаму і аналізу тональності тексту. Він забезпечує можливість багатокласової публікації. Це дозволяє прогнозувати ймовірності для множини значень цільової функції. Байєсовий алгоритм швидко навчається і його можна застосовувати для обробки в режимі реального часу.

Недоліком даного алгоритму є те, що якщо у тестовому наборі даних присутнє деяке значення ознаки, яке не зустрічалось раніше в навчальному наборі даних, така модель присвоїть нульову ймовірність цього значення і як наслідок, не може зробити прогноз. Це явище відоме під назвою «нульова частота». Але дану проблему можна вирішити за допомогою згладжування. Одним з найпростіших методів є згладжування по Лапласа (Laplace smoothing).

4 СТРУКТУРА І ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Загальна структура

В даному розділі описано загальну структуру та опис програмного забезпечення автоматизованої підсистеми класифікації публікацій, проведено аналіз її функціональних можливостей і призначення.

4.1.1 Границі проекту

Мета даної роботи полягає у створені автоматизованої підсистеми для автоматичної класифікації публікацій за єдиним універсальним десятковим кодом, збереженням цих публікацій у єдиній базі даних, та зручним пошуком і відображенням, що є важливим для автора тієї чи іншої публікації з метою економії часу, що йде на їх пошуки та класифікацію. Адже буде створена єдина база, в яку співробітники можуть завантажувати свої наукові публікації для перегляду та розповсюдження серед інших користувачів даної бази. Тим самим людина буде бачити, що розробив в якій сфері його співробітник, збережеться авторство розробок користувача, що є важливим у нас час як для автора, так і науки в цілому.

Користувачами даного бізнес-процесу є працівники різних галузей, що завантажують свої розробки у сфері науки та усі зареєстровані на сайті гості (студенти, аспіранти, викладачі…).

Ресурс відноситься до веб-рішень, спрямованих на розповсюдження публікації через мережу. Якість визначається замовником у відповідності до реалізації поставлених вимог. Особливих затрат на використання підсистеми немає. Особливих технічних знань задля користування підсистемою не вимагається.

Розробка даного рішення є найбільш економічно вигідним, так як не використовує великих фінансових витрат на технічні засоби і їх супровід, дає прийнятний час виконання роботи, а отже і вигідне співвідношення витрат та прибутку.

Термін дії даного проекту залежить від самих користувачів, чи є новинки у працях співробітників, чи завантажують вони їх, чи допомагає дана підсистема іншим користувачам. До поки є що розроблювати та писати про це, що буде актуальним для користувача – підсистема буде працювати.

4.1.2 Бізнес-потреби

Дана розробка відноситься до сфери обслуговування споживачів – надання послуг, що полегшить їх життя у сфері науки. Для успішного функціювання необхідний будь-який обчислювальний пристрій з можливістю доступу до мережі Інтернет та будь-якою операційною системою. В даній програмі передбачаються такі функції:

- реєстрація нового користувача;
- авторизація існуючого користувача;
- додавання та видалення адміністратором необхідної інформації;
- класифікація публікацій;
- збереження та занесення в базу наукових публікацій.

4.1.3 Безпека

Незареєстрований користувач немає прав доступу до підсистеми. Підсистемою можуть користуватися лише зареєстровані користувачі, які пройшли авторизацію та мають різні права доступу відповідно до своєї посади. Вони можуть добавляти, видаляти чи корегувати дані підсистеми в залежності від своїх прав доступу.

Користувач після авторизації має право лише подавати та корегувати відомості про свої публікації. Інші – лише мають доступ у вигляді перегляду наукових публікацій та їх завантаження.

Особисті дані користувачів повинні надійно зберігатись з обмеженим доступом та використовуватись лише з особистої згоди користувачів. При виявлені багів у заявленому функціоналі, повідомляється розробнику і вони усуваються.

4.1.4 Продуктивність

Продуктивність підсистеми залежить від апаратного забезпечення на якому вона працює. Час відклику в найгіршому випадку може становити до 1 хвилини, в середньому близько 2 секунд. В якості перешкод може виступати перенавантаження серверу або слабке інтернет з'єднання.

Отже, продуктивність даної системи залежить від:

- потужності системи;
- кількості користувачів, які одночасно можуть знаходитись на сайті та виконувати операції;
- кількості транзакцій за одиницю часу;
- бажаного часу відгуку від сервера;
- взаємодії з існуючими стандартами;

- пропускної здатності.

4.1.5 Супровід

Супроводом підсистеми, тобто підтримкою в працюючому стані займається адміністратор БД, який відповідає за доступ до бази даних, її цілісність, внесення до неї змін та реєстрацію користувачів підсистеми. Та в якості супроводу передбачається усунення знайдених дір в заявленому функціоналі.

4.1.6 Доступність

Підсистема повинна бути доступною користувачам в будь-який момент часу, резервне збереження даних має здійснюватися замовником не відключаючи сервер СКБД. Незаплановані простої можуть здійснюватися під час порушення роботи апаратного забезпечення чи мережевих атак. Також робота програми може бути призупинена під час введення нових змін в програмному коді.

4.1.7 Людський фактор

З точки зору локалізації, дана підсистема при необхідності може бути розширеною на різні мови.

Інтерфейси повинні бути інтуїтивно зрозумілими для усіх категорій користувачів, тому потреби у спеціальному навчанні немає.

Спеціальної мобільної версії застосунку немає, але вона може бути доступною з мобільних пристроїв, як звичайний веб-ресурс.

4.1.8 Методології

Розробка підсистеми здійснювалася по стандартній методології функціонального проектування: IDEF3, ERD згідно відповідних стандартів. Проектування виконувалось в програмних засобах ERWin та MySQL Workbench.

Для роботи зі спроектованою базою було використано MySQL. Користувач буде взаємодіяти з програмою, використовуючи графічний інтерфейс.

4.1.9 Масштабованість

Потрібно передбачити можливість росту підсистеми в зв'язку з можливим збільшенням в майбутньому кількості користувачів підсистеми та операцій, а також об'єму інформації в базі даних.

4.2 Структура програмних засобів

4.2.1 Концептуальне моделювання

В даному розділі представлена концептуальна модель даних – ERD (Рисунок 4.1). Дана ERD поділяється на декілька блоків, що можемо спостерігати. Кожен блок

має декілька сутностей, що взаємодіють між собою та в сукупності відповідають за окрему функціональну частину програми. Основні з них:

- синій блок сутностей – відповідає за публікації користувача, яка публікація кому належить, її класифікація та відображення в підсистемі;
- салатовий блок сутностей – відповідає за розподілення користувачів на ролі, яка публікація якому користувачу належить;
- червоний блок - відповідає за зв'язок та підтримку користувача із підсистемою.

Зв’язки між сутностями та взаємодія блоків між собою також представлена на рисунку 4.1.

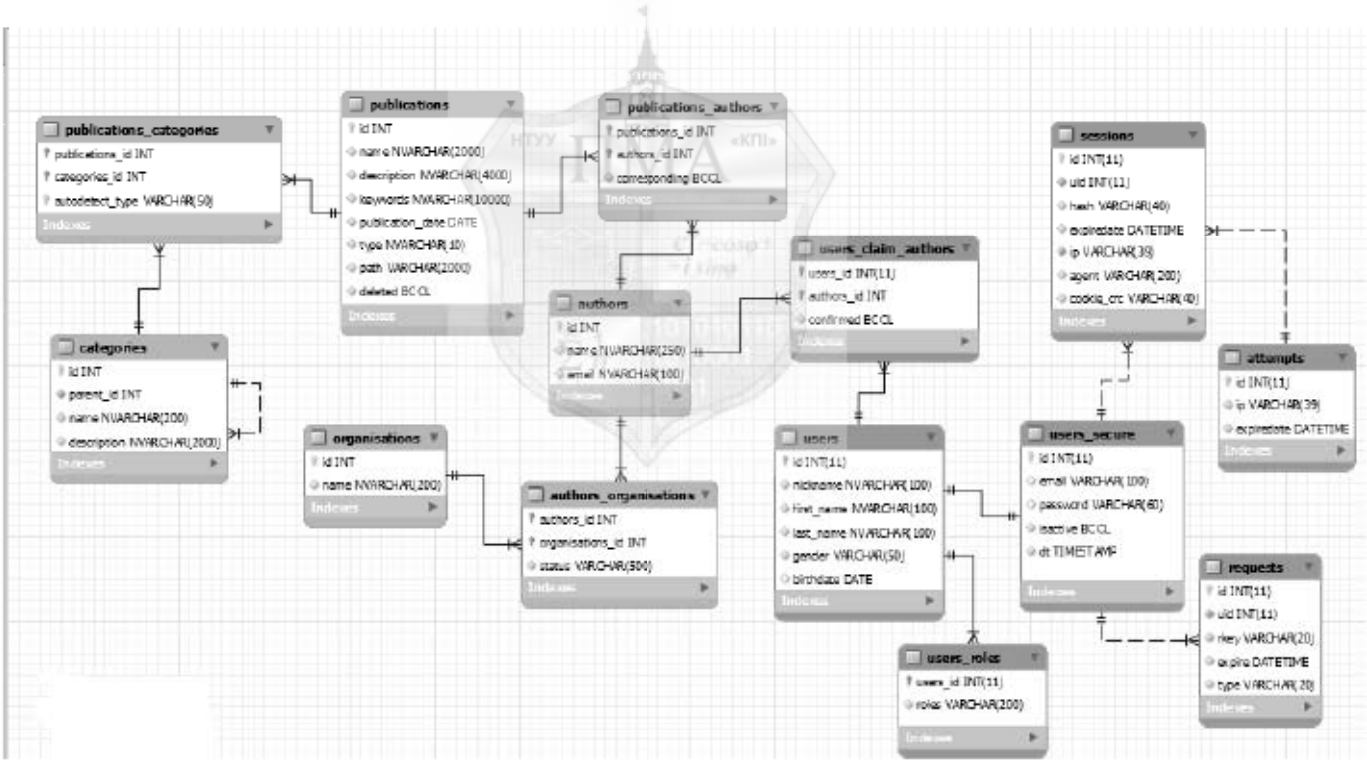

Рисунок 4.1 – Концептуальна модель даних (ERD)

Оскільки метою даної роботи є розробка математичного програмного забезпечення, що класифікує публікації, то основними користувачами можуть бути співробітники різних установ у сферах освіти, які хочуть використовувати класифікацію публікацій у своїх потребах. Тому дане програмне забезпечення має

обов’язково виконувати функцію: приймаючи на вхід адресу публікації, її дані, видати клас публікації та ймовірність саме цього результату за байєсовими мережами довіри.

4.2.2 Структурне моделювання

У даному підрозділі наведено відображення роботи процесів та потоків даних підсистеми.

Розглянемо життєві цикли об’єктів даних про користувача та публікації.

На рисунку 4.2 представлено життєвий цикл об'єкта "Користувач". В системі може бути доданий новий користувач. Також дані про нього можуть бути змінені, або видалені.

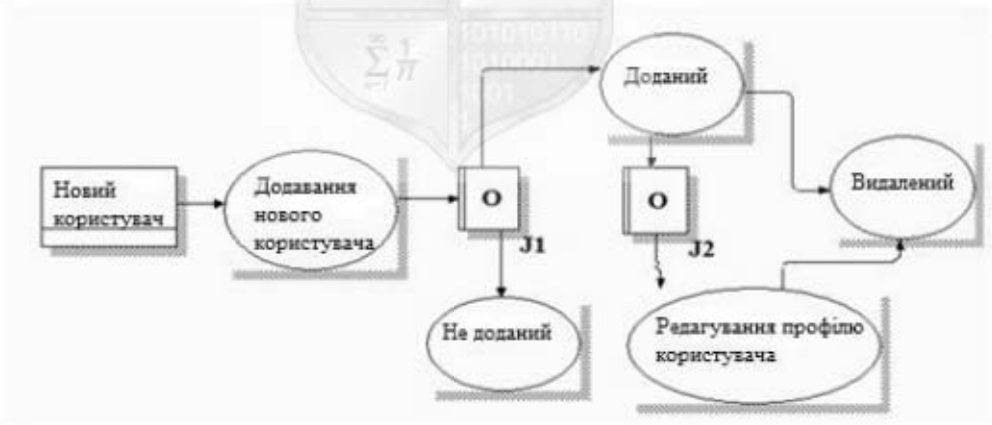

Рисунок 4.2 – Життєвий цикл користувача

На рисунку 4.3 представлено життєвий цикл об'єкта "Наукова публікація". В систему можна додати нову публікацію, яка буде прокласифікована. Також її можна буде видалити, якщо цього забажає автор.

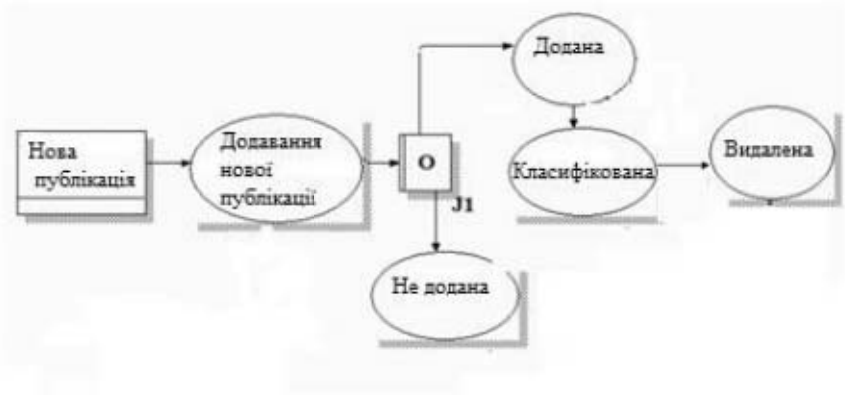

Рисунок 4.3 – Життєвий цикл публікації

За допомогою IDEF3 описано логіку виконання операцій, послідовність їх запуску та завершення. Дана IDEF3 представлена на рисунку 4.4.

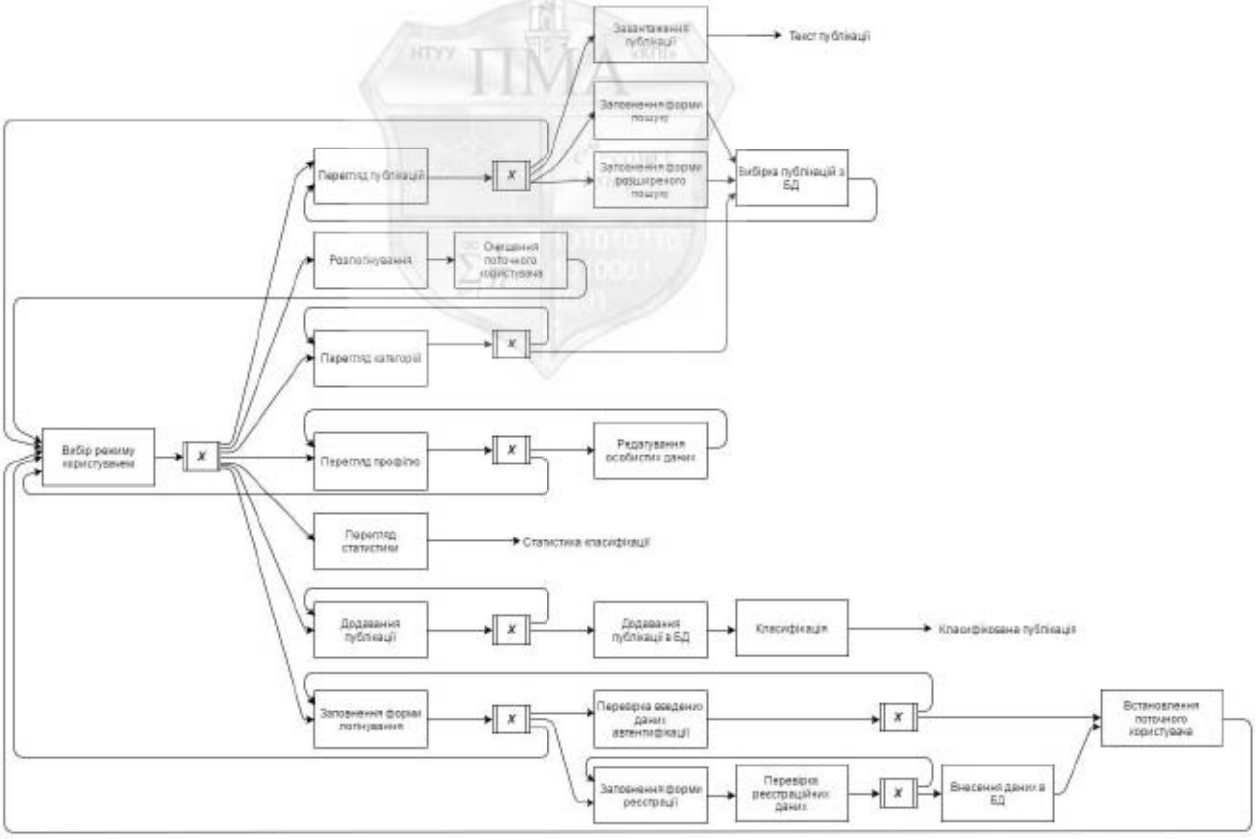

Рисунок 4.4 – Діаграма послідовності виконання процесів (IDEF3)

4.3 Керівництво користувача

Так як програма відноситься до веб-додатків, користувачу необхідно мати підключення до мережі Інтернет. Сама структура сайту є інтуїтивно-зрозумілою, тому особливих навичок не потребує. Не авторизованому користувачу надається можливість пошуку та перегляду публікацій, що вже є в базі. Щоб завантажити нову публікацію для класифікації, користувачу необхідно зареєструватись або авторизуватись на даному сайті. На рисунку 4.5 представлено головне вікно сайту.

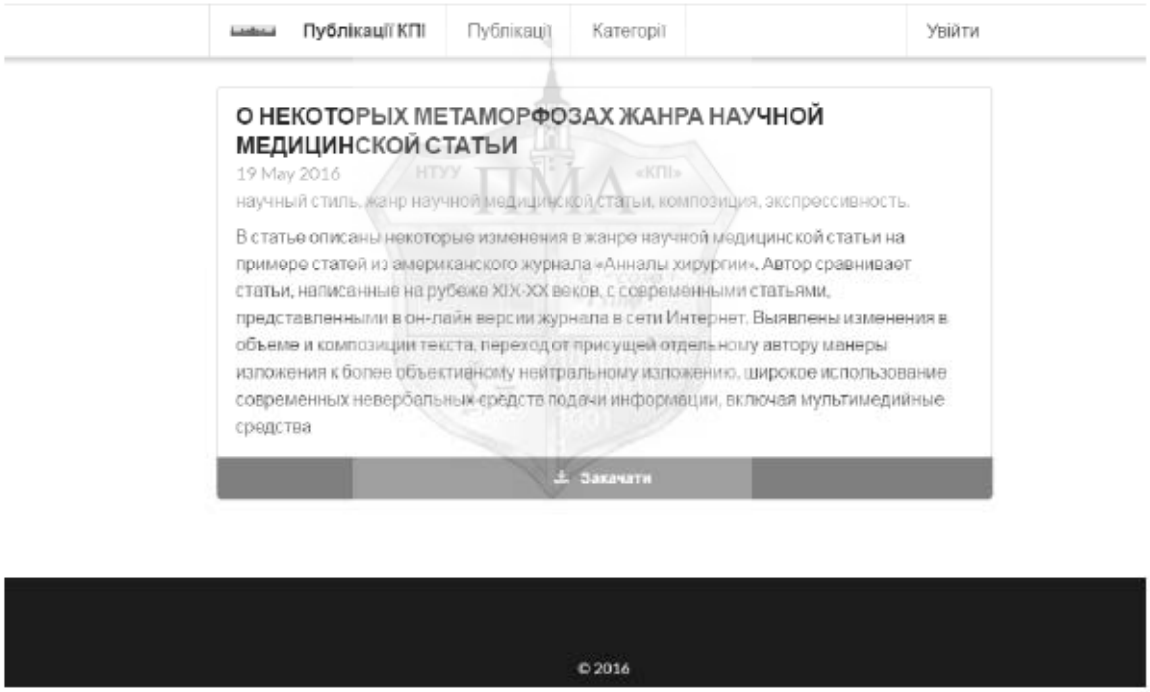

Рисунок 4.5 – Головне вікно сайту

Перейшовши у вікно авторизації, користувач повинен ввести свій логін та пароль, потім натиснути «Увійти». Якщо користувач ввів не вірні дані, або його немає у базі даних, буде видано відповідне повідомлення, що продемонстровано на рисунку 4.7. Саме вікно авторизації представлено на рисунку 4.6.

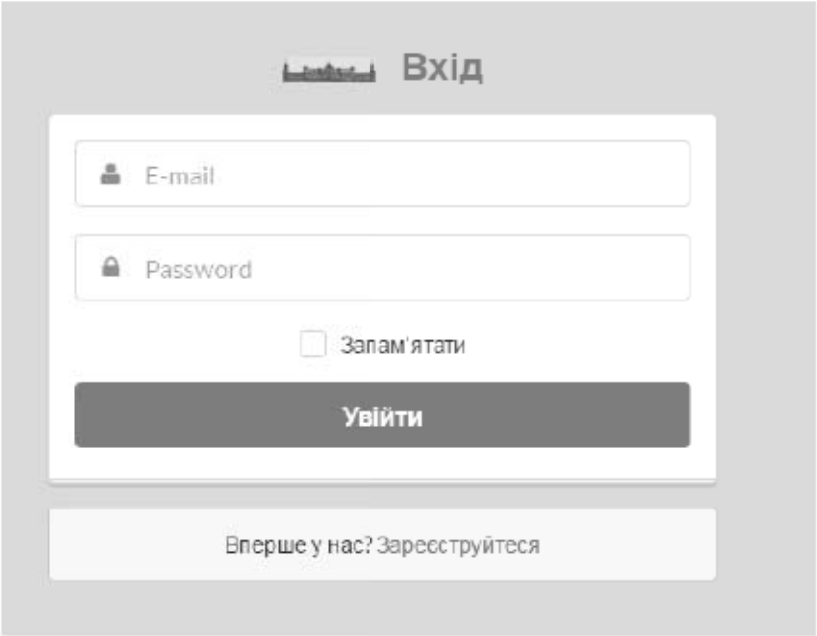

Рисунок 4.6 – Авторизація користувача

Рисунок 4.7 – Не правильно введені дані при авторизації

Перейшовши у вікно реєстрації (рисунок 4.8), користувач повинен ввести логін, пароль, свою електронну пошту, після чого натиснути кнопку «Зареєструватися».

При неправильному вводі якихось даних, буде виведено повідомлення про помилку (рисунок 4.9).

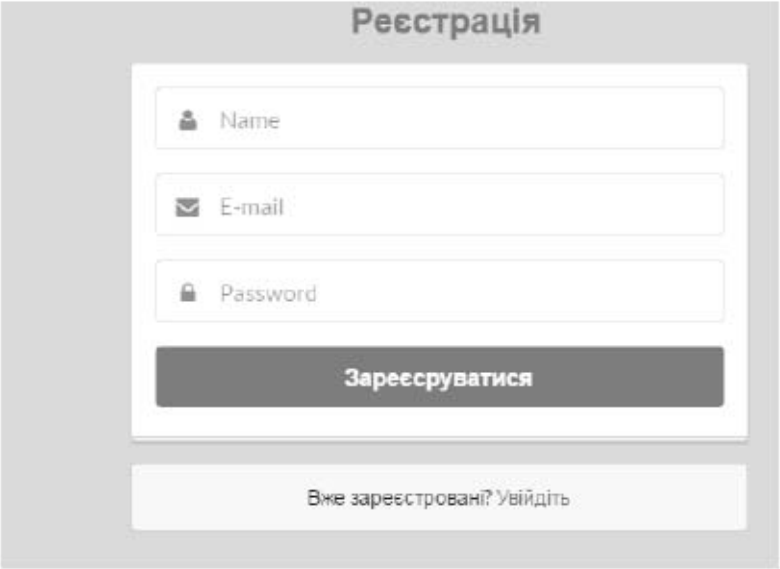

Рисунок 4.8 – Форма реєстрації користувача

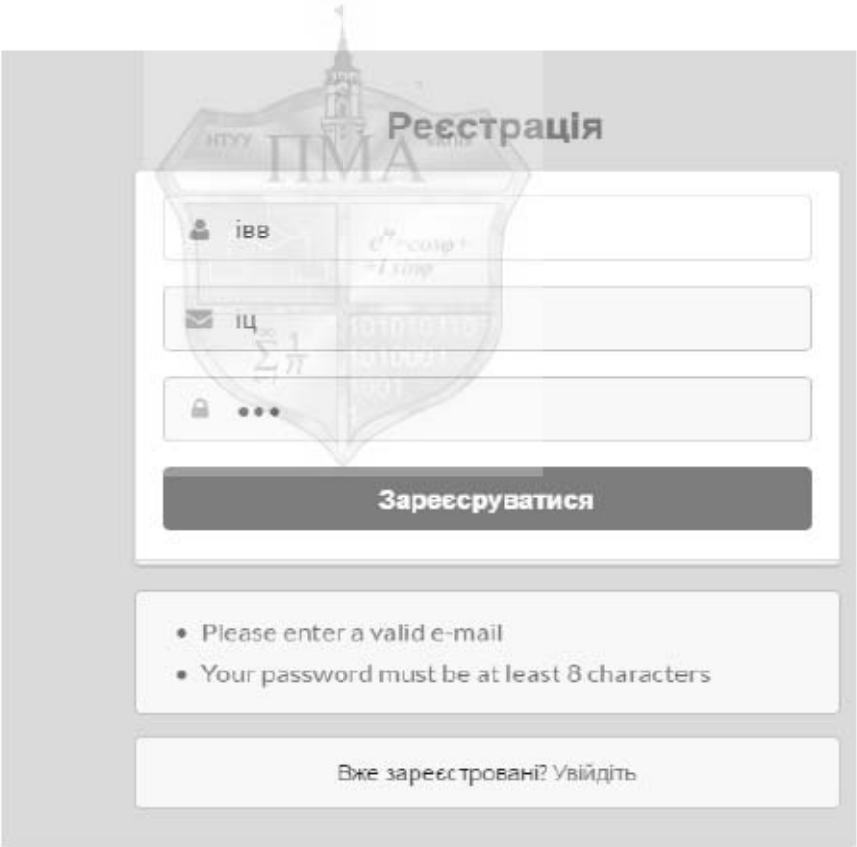

Рисунок 4.9 – Не правильно введені дані при реєстрації

Авторизованому користувачу надається можливість завантажити свою публікацію. Для цього надається відповідна форма для заповнення (рисунок 4.10). Користувач повинен ввести у відповідні поля форми: автора, назву публікації,

короткий зміст, ключові слова, вибрати дату написання та завантажити публікацію в форматі .pdf. Якщо даний автор є в вже базі, то його можна вибрати із відповідного списку (рисунок 4.11).

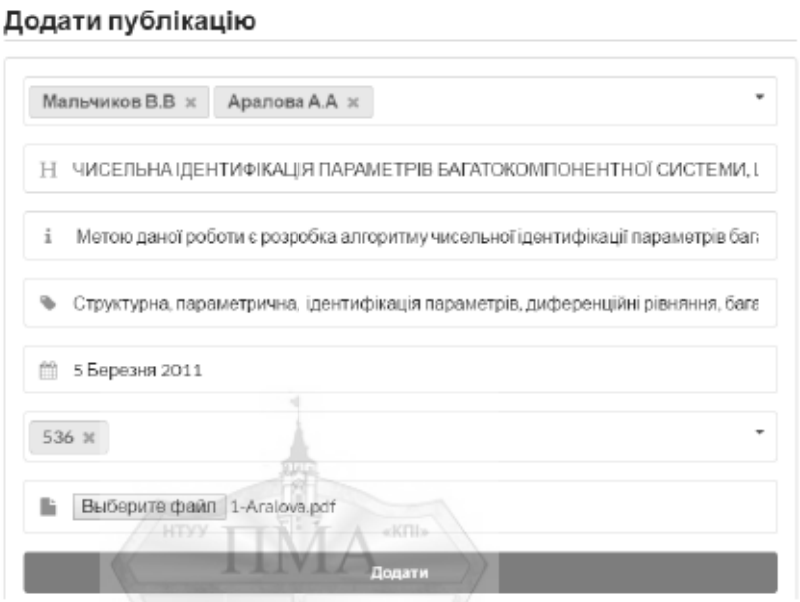

Рисунок 4.10 – Додавання нової публікації

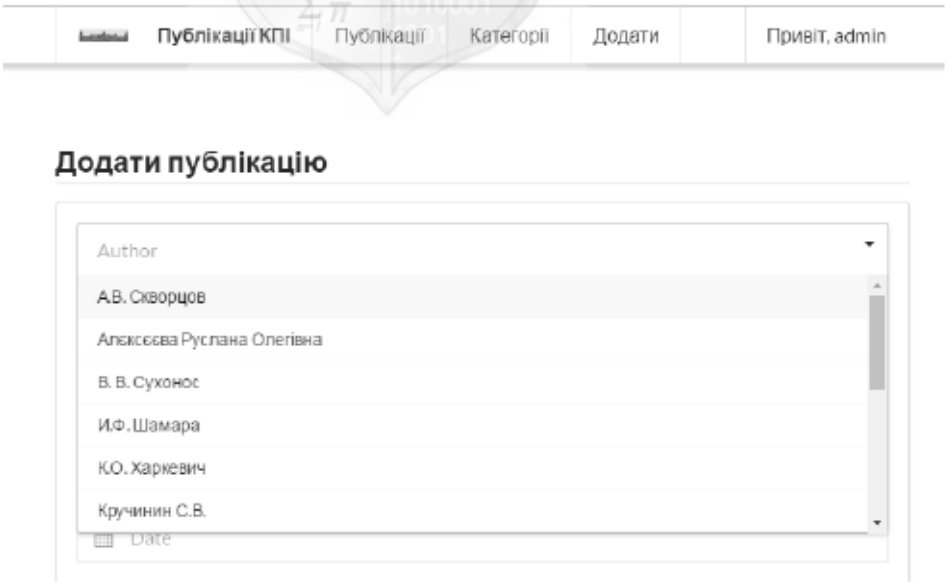

Рисунок 4.11 – Вибір автора із списку існуючих

На даному сайті користувачеві надається можливість переглянути публікації, що були занесені в базу даних, переглянути публікацію та завантажити її собі на

пристрій при необхідності (рисунок 4.12). Також розроблені як звичайний, так і розширений за певними критеріями пошуки публікацій у базі (рисунок 4.13).

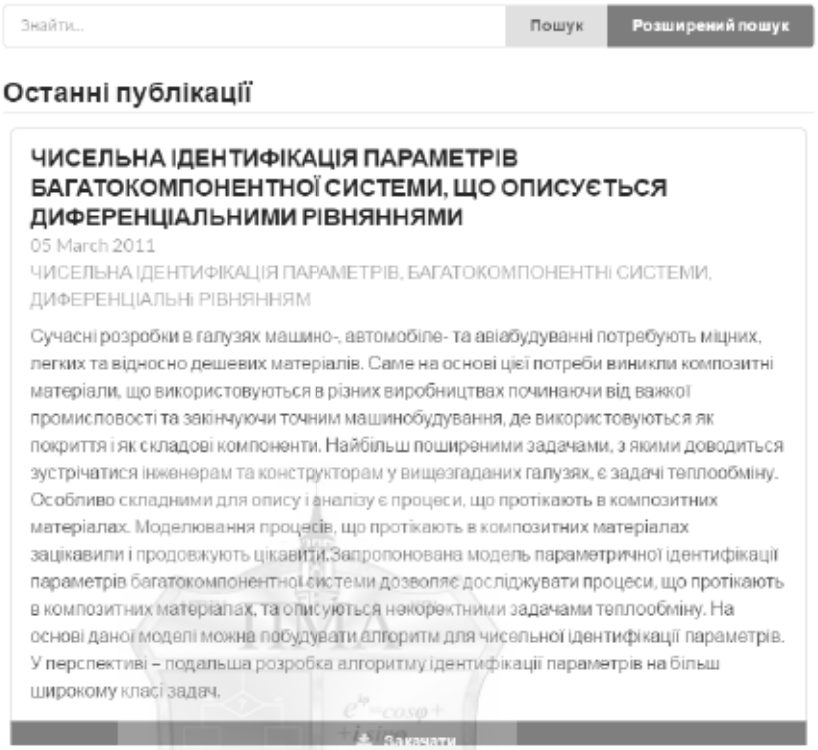

Рисунок 4.12 – Перегляд останніх публікацій

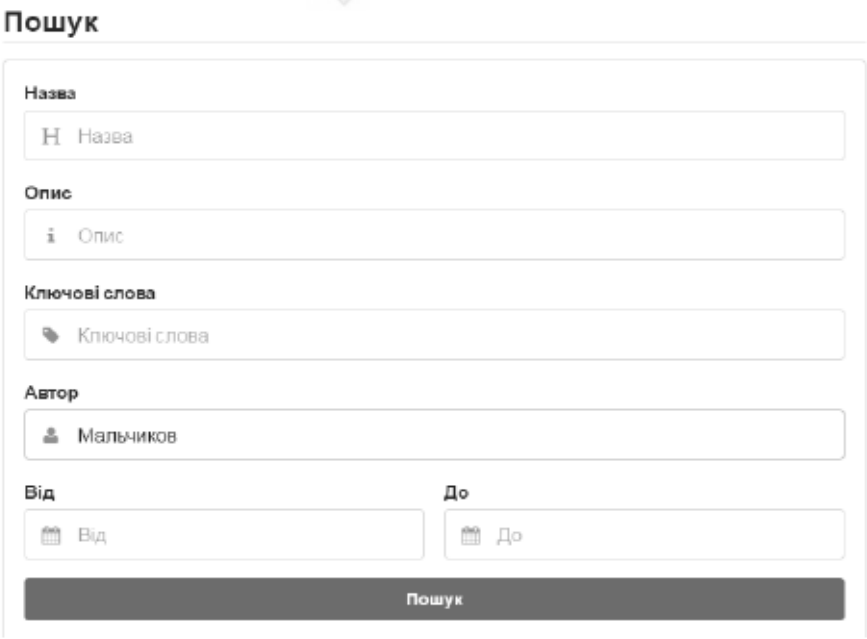

Рисунок 4.13 – Розширений пошук публікації у базі

Оскільки основною метою даної роботи є класифікувати публікацію, то користувачеві надається можливість переглянути в яку категорію, з якого ймовірністю увійшла зацікавлена публікація, де відображаються фактичні дані – дані за уніфікованим десятковим кодом; дані за класифікатором Байєса, та оцінка співпадіння отриманих даних із фактичними (рисунок 4.14).

Публікація	Категорії		Оцінка, %
	Фактичні	За класифікатором	
АЛГОРИТМ ПОШУКУ ЗАКОНОМІРНОСТЕЙ В ДАНИХ ПЕРЕПИСУ НАСЕЛЕННЯ	519.6	519	80

Рисунок 4.14 – Оцінка співпадіння класу за класифікатором

На даному сайті користувач також має можливість переглнути свої особисті дані (рисунок 4.15) та за бажанням їх відредагувати. Для цього буде надана форма для редагування особистих даних, що представлена на рисунку 4.16.

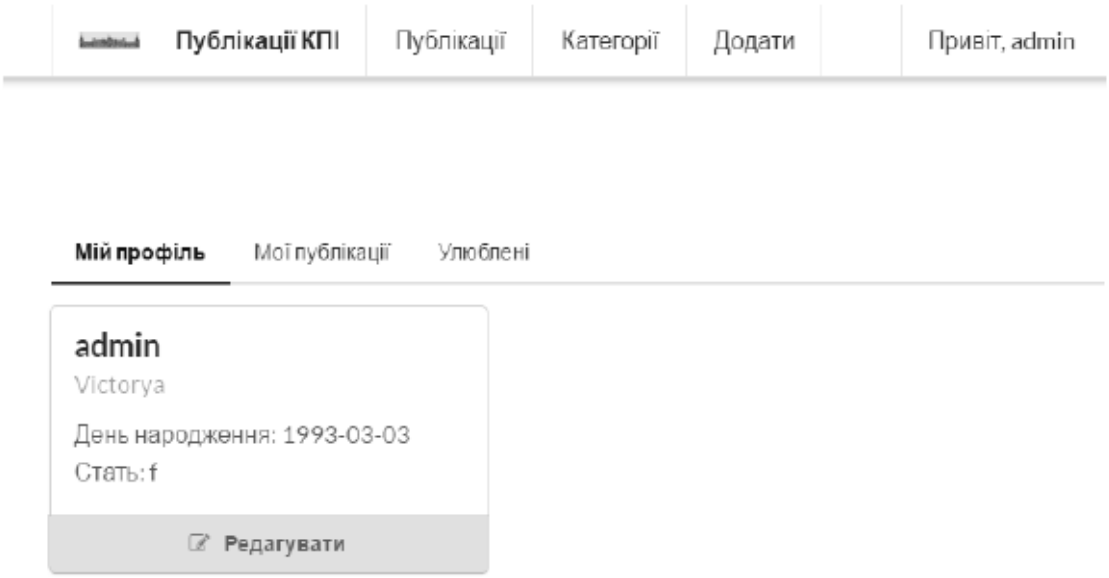

Рисунок 4.15 – Перегляд профілю користувача

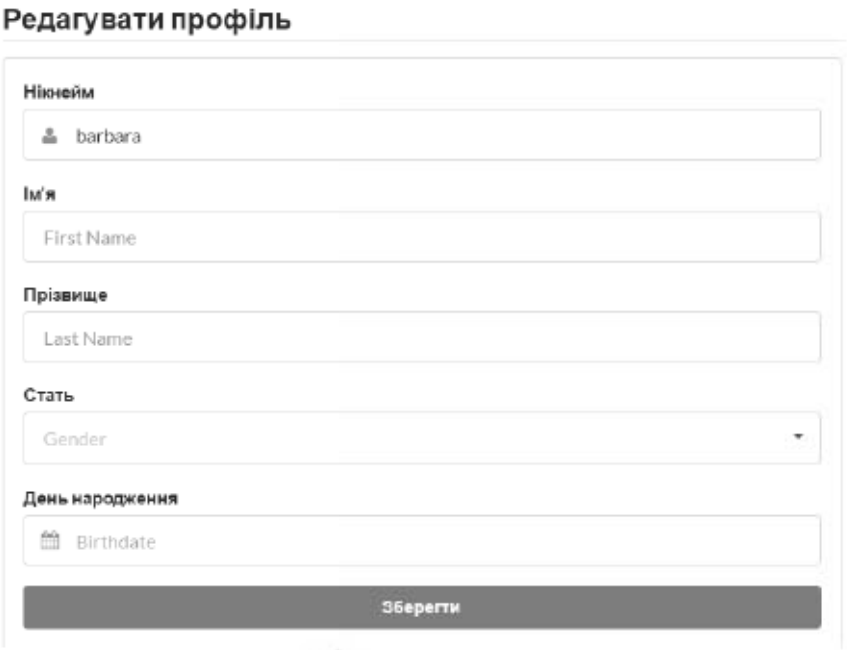

Рисунок 4.16 – Редагування профілю користувача

4.4 Випробування компонентів підсистеми

Покажемо працездатність класифікатора на декількох контрольних прикладах. Дану підсистему було протестовано на 101 публікації.

4.4.1 Контрольний приклад № 1

Якщо знову звернемось до рисунку 4.14, що був продемонстрований вище, можемо побачити результати роботи навчання байєсового класифікатора.

У якості вхідних даних була обрана публікація, що має свій унікальний десятковий код 519.6. Байєсовий класифікатор визначив дану публікацію до коду

519.0, що становить 80% правильності результату, що є позитивним результатом класифікації, оскільки мінімальна придатна точність класифікації є 70%.

4.4.2 Контрольний приклад № 2

Розглянемо наступний приклад (рисунок 4.17), де в якості вхідних даних була публікація, що має свій універсальний десятковий код 658.

Статистика

Публікація	Категорії		Оцінка, %
	Фактичні	За класифікатором	
МЕТОДИКА ОЦІНЮВАННЯ ЗАХИЩЕНОСТІ ДАНИХ КОРИСТУВАЧА В «ХМАРНИХ» ТЕХНОЛОГІЯХ	658	658	100
МЕТОДИКА ОЦІНЮВАННЯ ЗАХИЩЕНОСТІ ДАНИХ КОРИСТУВАЧА В «ХМАРНИХ» ТЕХНОЛОГІЯХ	004	658	0

Рисунок 4.17 – Контрольний приклад № 2

Підсистема визначила публікацію також до номеру 658, що складає 100% правильності роботи класифікатора. Якщо ми введемо знову таку саму публікацію, але скажемо, для прикладу, що її номер був 004, то класифікатор все рівно віднесе до номеру 658, що є насправді правильним. А отже, дані результату свідчать про те, що класифікатор працює вірно.

4.4.3 Контрольний приклад № 3

Візьмемо наступний приклад для тестування правильності роботи класифікатора (рисунок 4.18).

Публікація	Категорії		Оцінка, %
	Фактичні	За класифікатором	
Бібліотечна справа	3 001.1	001.1	50
Бібліотечна справа	3	001.1	0
Бібліотечна справа	001.1	001.1	100

Рисунок 4.18 – Контрольний приклад № 3

Дана публікація має 2 універсальні коди 3 та 001.1. Наш же класифікатор відніс публікацію лише до коду 001.1, що становить лише 50 % правильності. Це можна пояснити тим, що в базі зберігається, на даний момент, недостатня вибірка навчаючих даних за даною тематикою, тобто, не достатня кількість публікацій з якими порівнюються нові надходженні публікації. Дану проблему можна вирішити шляхом збільшенню кількості навчаючих даних для класифікатора за даною тематикою.

4.5 Висновки до розділу

В даному розділі було представлено архітектуру автоматизованої підсистеми, розглянуто її основні компоненти. Для даної підсистеми розроблено базу даних,

реалізацію якої розглянуто на концептуальному рівні. Структурно змодельована послідовність всіх процесів в підсистемі, розглянуто основні життєві цикли об'єктів.

Детально описано керівництво користувача з представленням всіх вікон програми та можливих повідомлень користувачу даної підсистеми. Було окремо протестовано математичну модель для класифікації публікацій з використанням універсального десяткового коду, а саме модель на основі байєсових мереж довіри.

Даний байєсовий класифікатор показав доволі гарний результат класифікації публікацій – 84,2 % правильності результату на 101 публікації.

ВИСНОВКИ

У даній роботі було розроблено автоматизовану підсистему класифікації публікацій із використанням універсального десяткового коду. У якості предметної області та набору даних для навчання були обрані публікації конференцій на різну тематику, кожна з яких має свій унікальний десятковий код. При побудові підсистеми була розроблена база даних, підібрана математична модель за допомогою якої був побудований класифікатор для класифікації публікацій, логічно правильно сформований пошук публікацій у базі даних, та протестовано правильність отриманих результатів. У результаті проведеного порівняльного аналізу за наперед визначеними критеріями для вирішення поставленої задачі обрано модель, що основана на байєсових системах довіри – наївний байєсовий класифікатор.

Для реалізації даної задачі, в якості головної мови програмування було використано PHP версії 5.5 для генерації сторінок на сервері, з підтримкою СКБД MySQL версії 5.7. В якості мови програмування, що виконується на стороні користувача, використовувався JavaScript. MySQL Workbench було використано як середовище розробки, що містить кейс-засоби і засоби адміністрування БД.

Отримані результати класифікації публікацій дають доволі гарний результат, якщо в базі достатня кількість навчаючих даних – 84,2% правильності класифікації на 101 публікації. Цей результат класифікатора є доволі позитивним серед існуючих рішень.

Таким чином, було спроектовано автоматизовану підсистему, що має зручний інтерфейс взаємодії програми із користувачем. Розроблений сервіс, що зможе знайти своє застосування, оскільки така підсистема дозволяє не тільки шукати, але і створює бібліотечний довідник по ключовим словам та змісту публікацій.

Представлена розробка підсистеми проста у використанні, так що освоїти принципи роботи з нею не складе труднощів навіть користувачеві, який володіє невеликими навичками роботи за комп'ютером.

ПЕРЕЛІК ПОСИЛАНЬ

1. Sebastiani F. Machine learning in automated text categorization / F. Sebastiani // ACM Computing Surveys (CSUR). — 2002. — Vol. 34, No. 1. — P. 1–47.

2. Sebastiani F. Text Categorization / F. Sebastiani // Text Mining and Its Applications. — 2005. — P. 109–129.

3. Hull D. A. Improving text retrieval for the routing problem using latent semantic indexing / D. A. Hull // Pro ceding's of SIGIR-94, 17th A CM International Conference on Research and Development in Information Retrieval. — Dublin, Ireland, 1994. — P. 282–289.

4. Joachims T. Text categorization with sup ort vector machines: learning with many relevant features / T. Joachims // Pro ceding's of ECML-98, 10th European Conference on Machine Learning. — Chemnitz, Germany, 1998. — P. 137–142.

5. Quinlan J. Induction of decision trees / J. Quinlan // Machine Learning. — 1998. — Vol. 1, No. 1. — P. 81–106.

6. Quinlan J. Programs for Machine Learning / J. Quinlan, M. Kaufmann. — 1993. — P. 302.

7. Dagan Ido, Karov Yael, Roth Dan. Mistake-driven learning in text categorization / Ido Dagan, Yael Karov, Dan Roth // In The second conference on empirical methods in natural language processing. — 1997.— P. 55–63.

8. Hwee Tou Ng, c Boon Goh, Kok Leong Low. Feature selection, perceptron learning, and a usability case study for text categorization / Tou Ng Hwee, Boon Goh Boon Goh, Leong Low Kok // SIGIR Forum.— 1997.— P. 67–73.

9. Wiener Erik, Pedersen Jan O, Weigend Andreas. A neural network approach to topic spotting / Erik Wiener, Jan O Pedersen, Andreas Weigend. — 1995.

10. Ruiz Miguel, Srinivasan Padmini. Hierarchical neural networks for text categorization / Miguel Ruiz, Padmini Srinivasan // In Proceedings of the 22 Annual

International ACM SIGIR Conference on Research and Development in Information Retrieval. — 1999.— P. 281–282.