

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

ЗАЯЦЬ Ірина Зіновіївна

**«Алгоритми оцінки ефективності веб - сайту
засобами машинного навчання / Algorithms for
evaluating the effectiveness of a website by machine
learning means»**

спеціальність: 123 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерна інженерія
Кваліфікаційна робота

Виконав студент групи КІм-21
І.З. Заяць

Науковий керівник:
к.т.н., доц. О. Й. Піцун

Кваліфікаційну роботу допущено
до захисту:

"__" _____ 20__ р.

Завідувач кафедри
_____ Л. О. Дубчак

ТЕРНОПІЛЬ – 2024

АНОТАЦІЯ

Заяць І. З. Алгоритми оцінки ефективності веб-сайту засобами машинного навчання. – Рукопис.

Кваліфікаційна робота на здобуття освітнього ступеня «магістр» за спеціальністю 123 «Комп'ютерна інженерія», освітньо-професійна програма. Західноукраїнський національний університет, Тернопіль, 2024.

Робота написана обсягом 70 сторінок і містить 27 ілюстрацій, 2 таблиці, 3 додатки та 32 джерела за переліком посилань. Метою роботи є розроблення алгоритмів для оцінки ефективності веб-сайтів на основі алгоритмів кластеризації даних.

Методи досліджень. Для розв'язання поставлених задач у кваліфікаційній роботі використано: алгоритми машинного навчання, зокрема методів кластеризації, які дозволяють групувати веб-сайти за схожими характеристиками, виявляючи тенденції, корисні для подальшого виконання.

Результати роботи призначені для використання в навчальному процесі та для проведення наукових досліджень в галузі опрацювання даних.

Орієнтовні напрямки розвитку досліджень: проаналізувати засоби оцінки ефективності веб-сайтів, проаналізувати алгоритми машинного навчання для опрацювання даних, розробити узагальнений алгоритм оцінки продуктивності веб-сайтів, розробити алгоритм кластеризації показників ефективності сайтів, програмно реалізувати графічний інтерфейс для задання даних для подальшого опрацювання засобами машинного навчання.

КЛЮЧОВІ СЛОВА: АЛГОРИТМ, КЛАСТЕРИЗАЦІЯ, МАШИННЕ НАВЧАННЯ, АНАЛІЗ, ОПТИМІЗАЦІЯ.

ANNOTATION

Zaiats I. Z. Algorithms for evaluating the effectiveness of a website by machine learning means. – Manuscript.

Qualification work for obtaining the degree of "Master" in specialty 123 "Computer Engineering", educational and professional program. Western Ukrainian National University, Ternopil, 2024.

The work is written in 70 pages and contains 27 illustrations, 2 tables, 3 appendices and 32 sources according to the list of references. The purpose of the work is to develop algorithms for assessing the effectiveness of websites based on data clustering algorithms.

Research methods. To solve the tasks set in the qualification work, the following were used: machine learning algorithms, in particular clustering methods, which allow grouping websites by similar characteristics, identifying trends that are useful for further implementation.

The results of the work are intended for use in the educational process and for conducting scientific research in the field of data processing.

Approximate directions of research development: analyze tools for assessing website performance, analyze machine learning algorithms for data processing, develop a generalized algorithm for assessing website performance, develop an algorithm for clustering website performance indicators, programmatically implement a graphical interface for specifying data for further processing by machine learning tools.

KEYWORDS: ALGORITHM, CLUSTERIZATION, MACHINE LEARNING, ANALYSIS, OPTIMIZATION.

3MICT

Вступ.....	9
1. Аналіз засобів оцінювання продуктивності веб-сайтів.....	12
1.1 Засоби оцінки ефективності веб-сайтів	12
1.2. Алгоритми навчання без вчителя.....	18
1.3 Існуючі аналоги засобів оцінки продуктивності сайтів	22
1.4 Аналіз завдання магістерської роботи та постановка задач дослідження	25
2. Алгоритми оцінки ефективності веб-сайтів	27
2.1 Узагальнений алгоритм оцінки ефективності веб-сайтів	27
2.2 Алгоритм кластеризації показників продуктивності веб-сайтів.....	31
2.3 Структура бази даних	39
2.4 Висновки до розділу	41
3. Програмна реалізація алгоритмів оцінки ефективності веб-сайтів.....	43
3.1 Графічний інтерфейс розробленого модулю	43
3.2 Формування набору даних	46
3.3 REST API	50
3.4 Порівняльний аналіз отриманих результатів	53
3.5 Висновки до розділу 3	58
Висновки	59
Список використаних джерел	60
Додаток А Лістинг коду генерування та візуалізація даних	Помилка! Закладку не визначено.
Додаток Б Лістинг коду генерування та візуалізація даних	Помилка! Закладку не визначено.
Додаток В Світлокопії виданих публікацій	Помилка! Закладку не визначено.

ВСТУП

У сучасному цифровому світі веб-сайти стали важливою складовою будь-якої організації або бізнесу. Вони служать не тільки як основний канал комунікації з клієнтами, але й визначають успішність підприємства в умовах глобалізації та цифрової трансформації. З цієї причини ефективність веб-сайтів є критичним чинником для їхньої конкурентоспроможності. Оцінка ефективності веб-сайтів стає необхідною задачею для бізнесів, маркетологів та розробників, адже це дозволяє своєчасно виявляти слабкі місця, покращувати користувацький досвід і оптимізувати веб-ресурси.

В останні роки набули популярності різноманітні сервіси для автоматичної оцінки веб-сайтів, які вимірюють показники їх продуктивності, доступності та зручності для користувачів. Одним із найпоширеніших інструментів для аналізу веб-сайтів є сервіси, такі як Google PageSpeed Insights, GTmetrix, Pingdom та інші. Вони надають різноманітні метрики, зокрема, час завантаження сторінки, перший контентний малюнок (FCP), час до взаємодії (TBT), найбільший контентний малюнок (LCP) тощо. Однак, для точного аналізу ефективності та виявлення потенційних проблем, окрім традиційних метрик, необхідно застосовувати більш глибокі методи обробки та класифікації даних.

Одним із таких методів є використання алгоритмів машинного навчання, зокрема, методів кластеризації, які дозволяють групувати веб-сайти за схожими характеристиками, таким чином виявляючи тенденції, корисні для подальшого вдосконалення. Кластеризація дає змогу виявляти схожі групи веб-ресурсів, що мають спільні проблеми з продуктивністю або зручністю користувача, що дозволяє оптимізувати їх на основі отриманих результатів [1].

У роботі будуть розглянуті основні метрики для оцінки ефективності веб-сайтів, а також розроблено модель кластеризації для автоматичної категоризації веб-ресурсів за їх характеристиками.

Метою кваліфікаційної роботи є розробка алгоритмів для оцінки ефективності веб-сайтів на основі алгоритмів кластеризації даних. Результати цього дослідження можуть стати основою для покращення якості веб-ресурсів та надання рекомендацій щодо їх оптимізації.

Для виконання мети потрібно вирішити наступні задачі:

- проаналізувати засоби оцінки ефективності веб-сайтів;
- проаналізувати алгоритми машинного навчання для опрацювання даних;
- розробити узагальнений алгоритм оцінки продуктивності веб-сайтів;
- розробити алгоритм кластеризації показників ефективності сайтів;
- програмно реалізувати графічний інтерфейс для задання даних для подальшого опрацювання засобами машинного навчання;
- провести порівняльний аналіз розробленого програмного модуля на основі кластеризації з аналогами;

Об'єкт дослідження – процеси оцінювання ефективності веб-сайтів за критеріями серверної та клієнтської частини.

Предмет дослідження – методи кластеризації для виділення даних в режимі навчання без вчителя.

Наукова новизна полягає у розробці алгоритму оцінки ефективності веб-сайтів на основі кластеризації даних показників веб-сайтів.

Результати кваліфікаційної роботи призначені для використання в навчальному процесі та для проведення наукових досліджень в галузі опрацювання даних.

Кваліфікаційна робота складається із трьох розділів, висновків, списку використаної літератури та додатків.

У першому розділі було проаналізовано засоби оцінки ефективності веб-сайтів, алгоритми навчання без вчителя та існуючі засоби оцінки продуктивності на серверному та клієнтському рівнях.

У другому розділі були наведені алгоритмічні розробки оцінки ефективності веб-сайтів на основі показників від сервісів статистичних оцінок сайтів.

У третьому розділі були програмно реалізований модуль графічного інтерфейсу для збору даних для їх подальшого опрацювання засобами машинного навчання та проведено порівняльний аналіз отриманих результатів.

Результати роботи знайшли своє відображення у роботі Всеукраїнської науково-практичної конференції молодих вчених, аспірантів і студентів «Інтелектуальні комп'ютерні системи та мережі» (Тернопіль, Україна, 5 листопада 2024 року).

1. АНАЛІЗ ЗАСОБІВ ОЦІНЮВАННЯ ПРОДУКТИВНОСТІ ВЕБ-САЙТІВ

1.1 Засоби оцінки ефективності веб-сайтів

Оцінка сайту є важливим етапом у забезпеченні його ефективності та конкурентоспроможності в інтернеті. Оцінка сайту — це процес аналізу й оцінювання якості, ефективності та загальної корисності вебсайту з різних точок зору [2]. Періодичний аналіз допомагає виявити проблеми, оптимізувати сайт для кращого залучення аудиторії, підвищити конверсію та забезпечити успішну роботу онлайн-ресурсу.

Перед тим як розпочати процес оцінки сайту, важливо визначити основні критерії, за якими буде проводитися аналіз. Вони охоплюють як технічні аспекти, так і зручність для користувачів, оптимізацію для пошукових систем та інші фактори [3, 4, 5]. Основні категорії оцінки сайту та ключові показники, які допомагають виявити сильні та слабкі сторони веб-ресурсу наведено нижче у таблиці 1.1

Таблиця 1.1 – Критерії оцінки сайту

Категорія	Що оцінюється	Ключові показники
Юзабіліті	Зручність використання сайту	Простота навігації, чіткість інтерфейсу, адаптивність до мобільних пристроїв, зручність форм для заповнення користувачами
Технічна оцінка	Продуктивність та безпека сайту	Швидкість завантаження, сумісність з браузером, адаптивність до пристроїв, наявність SSL, відсутність технічних помилок

Продовження таблиці 1.1

SEO-аналіз	Пошукова оптимізація сайту	Відповідність вимогам пошукових систем, наявність та оптимізація мета-тегів, правильна перелінковка, використання ключових слів
Контент	Якість та актуальність інформації	Якість текстів, актуальність контенту, відповідність інтересам користувачів, регулярне оновлення, використання візуальних елементів (зображення, відео)
Аналіз конкурентів	Порівняння сайту з іншими в ніші	Переваги та недоліки у функціоналі, контенті, юзабіліті та SEO конкурентів
Аналіз аудиторії	Вивчення поведінки та характеристик користувачів	Хто відвідує сайт, поведінка на сайті (середній час перебування, кількість відвіданих сторінок), рівень конверсії (виконання цільових дій: покупка, підписка, замовлення тощо)

Швидкість завантаження сайту відіграє ключову роль у формуванні вражень користувачів та визначає їхню подальшу поведінку. У світі електронної комерції кожна секунда має значення: швидкий сайт сприяє вищій лояльності, а повільний може відштовхнути потенційних клієнтів. Дослідження показують, що користувачі очікують, що сторінка завантажиться за 1-3 секунди. Після цього часу терпіння користувачів починає зникати, і вони, швидше за все, підуть на сайт конкурентів [6].

Нижче представлена інфографіка, яка порівнює очікуваний і допустимий час завантаження веб-сторінки. Вона наочно показує, як швидкість сайту впливає на задоволеність користувачів та їхню готовність залишатися на сторінці або здійснювати покупку.

Існують різноманітні інструменти для оцінки продуктивності та якості вебсайтів, які допомагають поліпшити їхню функціональність. Вони аналізують

швидкість завантаження, SEO-оптимізацію, безпеку та користувацький досвід. Використання таких засобів дозволяє вебмайстрам виявити проблеми та впровадити рішення для підвищення ефективності сайтів.

GTmetrix — це онлайн-платформа, яка використовується для аналізу продуктивності веб-сайтів. Вона надає детальну інформацію про швидкість завантаження сторінки, її оптимізацію, а також рекомендації щодо поліпшення показників продуктивності. GTmetrix дозволяє цінювати швидкість завантаження сторінки, зокрема платформа аналізує час завантаження сайту та окремих його елементів. Також GTmetrix ідентифікує проблеми оптимізації, наприклад GTmetrix виділяє проблеми, які впливають на продуктивність (наприклад, некоректна оптимізація зображень, кешування або надмірний обсяг CSS/JavaScript). Сервіс пропонує конкретні кроки для поліпшення швидкості роботи сайту [7]. GTmetrix дозволяє відстежувати показники продуктивності у динаміці. За допомогою GTmetrix можна проводити тести для різних пристроїв, браузерів, місцезнаходжень і швидкості мережі. Переваги GTmetrix такі:

- зручний інтерфейс;
- інтеграція з Lighthouse;
- детальні звіти;
- багатомовність;
- гнучкі налаштування тестування;
- моніторинг у реальному часі.

Даний сервіс характеризується наявністю зрозумілого та доступного як для початківців, так і для досвідчених веб-розробників інтерфейсу, використанням аналітики Google Lighthouse для більш точного аналізу, розбиттям інформації на категорії з чіткими рекомендаціями (включаючи структуру сайту, швидкість, розмір сторінки тощо), підтримує декілька мов, що робить платформу доступною для різних користувачів, можливість вибору регіону для тесту, швидкості інтернету та типу пристрою, доступ до регулярного аналізу з автоматичною генерацією звітів [8].

До недоліків GTmetrix можна віднести такі:

- обмеження безкоштовної версії;
- складність для новачків;
- фокус на технічні показники;
- час тестування;
- обмежені можливості порівняння.

Графічний інтерфейс веб-сервісу наведено на рисунку 1.1.

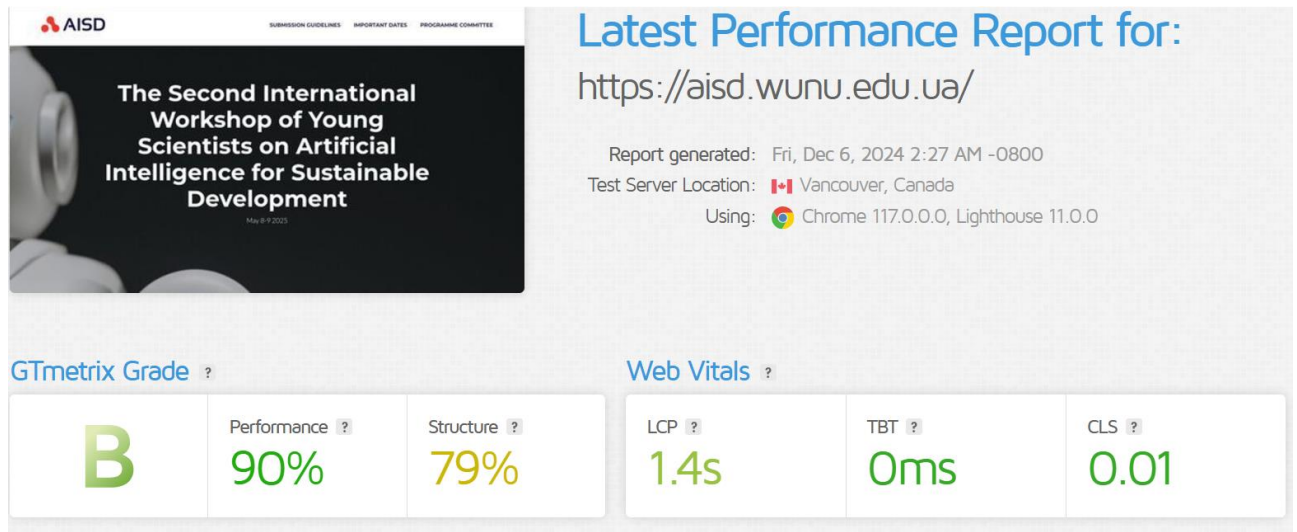


Рисунок 1.1 – Графічний інтерфейс веб-сервісу GTmetrix

Загальні показники продуктивності:

- Performance Score (Загальна оцінка продуктивності): рейтинг на основі Google Lighthouse, що показує якість завантаження сторінки;
- Structure Score (Оцінка структури): аналіз внутрішньої структури сайту та її оптимізації;
- Speed Index: час, за який стає видимою більшість контенту сторінки;
- Largest Contentful Paint (LCP): час, за який найбільший елемент сторінки стає видимим;
- Time to Interactive (TTI): час, за який сторінка стає повністю інтерактивною;
- First Contentful Paint (FCP): час, коли перший візуальний елемент відображається на сторінці;

- TT (TBT): сукупний час, протягом якого сторінка була заблокована (не реагувала на дії користувача);
- Cumulative Layout Shift (CLS): оцінка стабільності макета під час завантаження.

WebPageTest — це відкритий інструмент для оцінки продуктивності веб-сайтів, який переважно підтримується компанією Google. Цей сервіс використовує один або кілька серверів, які функціонують як автоматизовані браузерери, відвідуючи сайти та збираючи дані про користувацький досвід. Результати тестування подаються у вигляді детального звіту про продуктивність [9, 10]. WebPageTest дозволяє проводити тести з кількох геолокацій, що дає змогу оцінити, як швидкість завантаження варіюється в різних регіонах світу. Користувачі можуть вибирати між різними браузерами, такими як Chrome, Firefox, і навіть Internet Explorer, що дозволяє перевіряти сумісність та продуктивність сайту в різних умовах, що також відображено на відповідних рисунках 1.2 і 1.3.

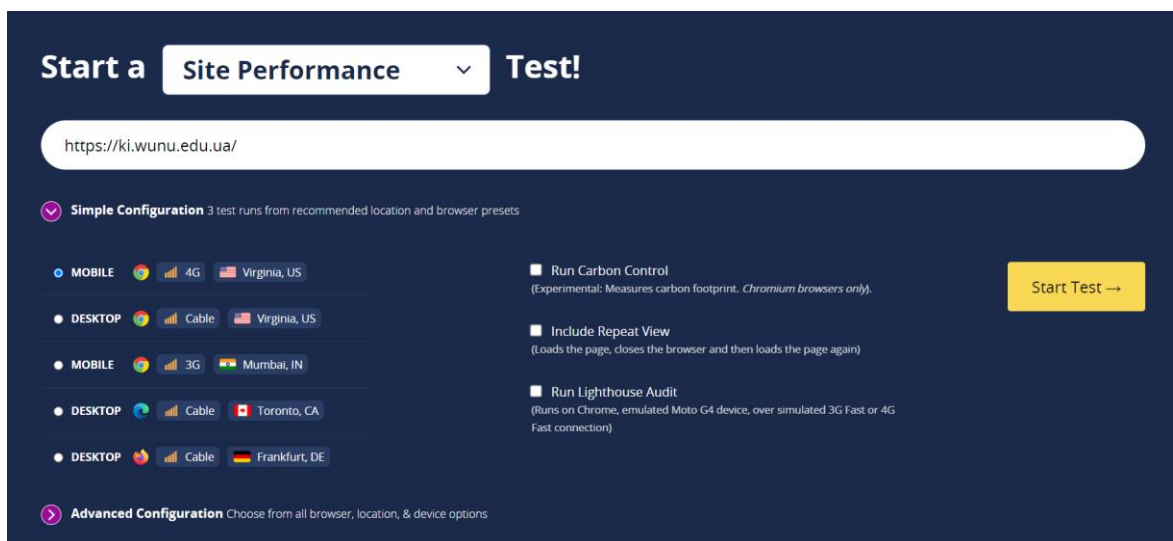


Рисунок 1.2 – Загальні налаштування WebPageTest

The screenshot shows the WebPageTest interface. At the top, there's a header with 'Start a' followed by a dropdown menu set to 'Site Performance' and a 'Test!' button. Below this is a large input field containing the URL 'https://ki.wunu.edu.ua/'. Under the URL field, there are two configuration options: 'Simple Configuration' (with a right arrow icon) and 'Advanced Configuration' (with a checkmark icon). The 'Advanced Configuration' option is selected, and its description 'Choose from all browser, location, & device options' is visible. Below these options are two dropdown menus: 'Test Location' set to 'Dulles, Virginia, USA' and 'Browser' set to 'Chrome'. A yellow 'Start Test →' button is located at the bottom right of the configuration area.

Рисунок 1.3 – Інтерфейс вибору локації WebPageTest

Після налаштування параметрів тестування необхідно натиснути кнопку "Start Test". Інструмент запускає автоматизований процес, в ході якого браузер відвідує вказану сторінку та збирає дані про швидкість завантаження. WebPageTest автоматично записує час завантаження різних елементів на сторінці, таких як зображення, CSS та JavaScript файли. Цей процес забезпечує детальне відображення всіх аспектів завантаження (рисунок 1.4).

The screenshot shows the 'Тестування...' (Testing...) results page. It displays the URL 'https://ki.wunu.edu.ua/' and the test configuration: 'Від: Даллес, Вірджинія, США - Chrome - кабель' (From: Dulles, Virginia, USA - Chrome - cable). Below this, a box contains the following information: 'Тільки перший перегляд' (Only first view), 'Тестові запуски: 3' (Test runs: 3), 'Підключення: 5000/1000 Кбіт/с, затримка 28 мс' (Connection: 5000/1000 Kbps, latency 28 ms), and a link to 'Спеціальні показники' (Special metrics). At the bottom, there is a progress bar with three stages: 'Очікування...' (Waiting...), 'Тестування...' (Testing...), and 'Готово' (Ready). The 'Тестування...' stage is currently active and highlighted in blue.

Рисунок 1.4 – Результати тестування веб-сайту

Показники ефективності веб-сайту є критично важливими для оцінки його продуктивності та впливу на користувацький досвід. Серед основних показників

можна виділити час завантаження, який визначає загальний період, необхідний для повного завантаження сторінки, а також Time to First Byte (TTFB), що показує затримку від запиту до отримання першого байта даних. Водоспадна діаграма візуалізує порядок завантаження елементів, що допомагає виявити повільні ресурси, тоді як кількість запитів і розмір сторінки впливають на швидкість завантаження.

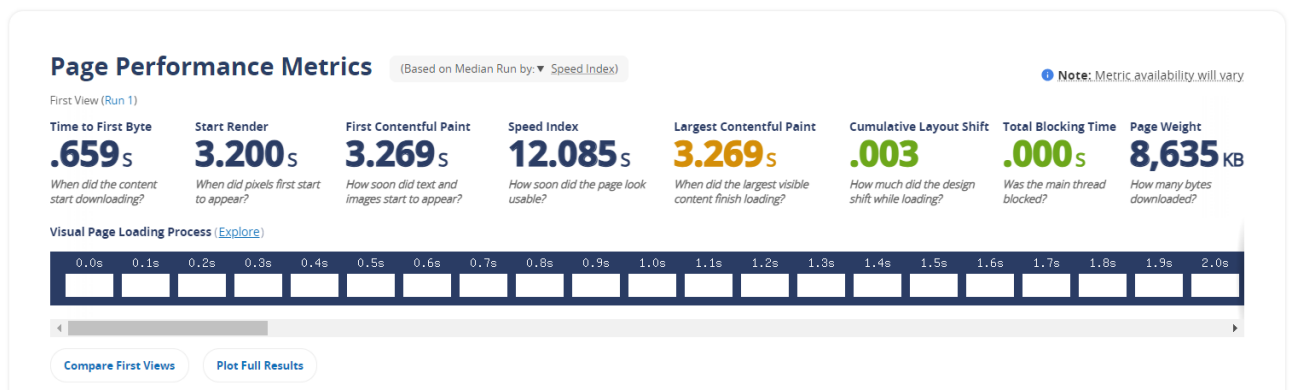


Рисунок 1.5 – Результати оцінки сайту

Інші важливі показники, такі як First Contentful Paint (FCP) та Largest Contentful Paint (LCP), показують, коли користувач вперше бачить вміст, а Cumulative Layout Shift (CLS) вимірює візуальну стабільність елементів на сторінці. Всі ці показники не тільки впливають на задоволеність користувачів, але й є важливими факторами для SEO, оскільки швидкість завантаження може вплинути на позиції сайту в пошукових системах. Регулярний моніторинг і оптимізація цих показників є ключовими для забезпечення високої продуктивності та конкурентоспроможності вебсайту [11].

1.2. Алгоритми навчання без вчителя

Машинне навчання дозволяє знаходити рішення в завданнях, які важко формалізувати традиційними методами. Наприклад, розпізнавання зображень, обробка природної мови або прогнозування складних взаємозалежностей у

даних. Замість того, щоб вручну створювати правила для розпізнавання обличчя, ML-модель сама навчиться виділяти ключові особливості для цього завдання [12]. Візуалізація процесу класеризації наведено на рисунку 1.6.

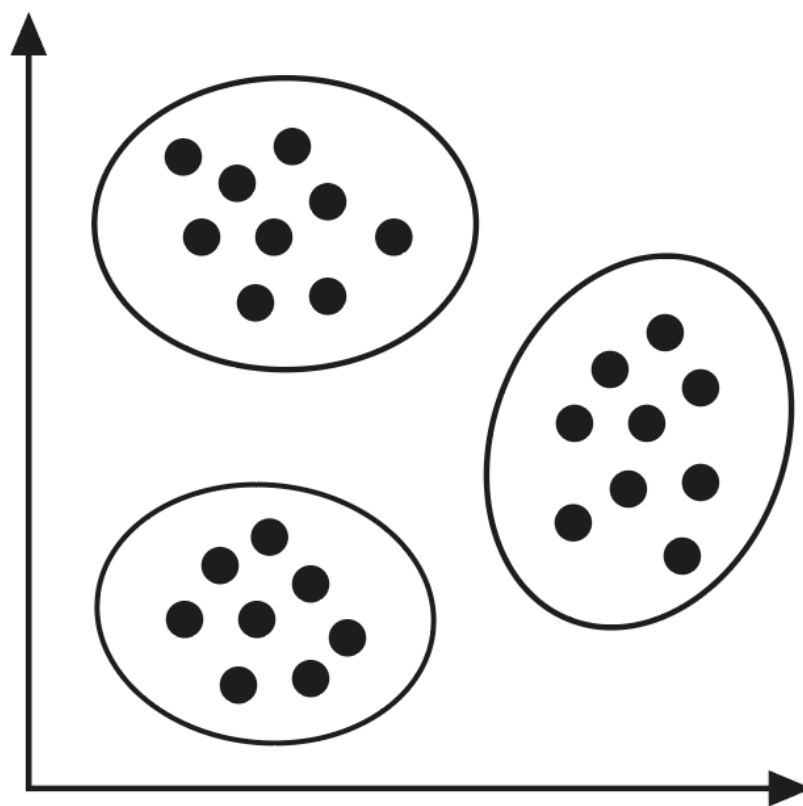


Рисунок 1.6 – Процес кластеризації

Моделі машинного навчання можуть адаптуватися до змін у даних і постійно вдосконалюватися. Це робить їх ефективними у середовищах, які швидко змінюються. ML ефективно справляється з завданнями прогнозування на основі минулих даних, а також створення нових даних, схожих на існуючі.

Машинне навчання використовується у різних галузях:

- медицина;
- фінанси;
- маркетинг;
- транспорт.

ML ефективно обробляє неструктуровані дані, такі як тексти, зображення, аудіо чи відео, тоді як традиційні алгоритми краще працюють з табличними

даними. Машинне навчання дозволяє створювати системи, які постійно покращують свої результати, використовуючи нові дані. Моделі машинного навчання можуть адаптуватися до змін у даних і постійно вдосконалюватися. Це робить їх ефективними у середовищах, які швидко змінюються [13].

Навчання без учителя знижує витрати часу і ресурсів на маркування даних, що є особливо цінним для великих та складних наборів даних. Прикладом є аналіз супутникових знімків для класифікації територій без попереднього маркування кожного знімка. Алгоритми можуть допомогти досліджувати нові домени, де очікувані результати невідомі. Наприклад, виявлення нових типів захворювань за медичними показниками.

Алгоритми навчання без учителя в машинному навчанні (unsupervised learning) відрізняються тим, що працюють з даними, які не мають міток або попередньо визначених категорій. Вони спрямовані на виявлення прихованих закономірностей, структур і взаємозв'язків у даних без необхідності вручну позначати їх. У цьому підході моделі працюють з "сирими" даними, які не мають попередньо визначених результатів чи категорій. Це корисно у випадках, коли маркування даних є складним, дорогим або неможливим [14].

Головна задача алгоритмів — знайти закономірності чи структури в даних. Це може бути корисно для розуміння складної багатовимірної інформації.

Кластеризація — це метод навчання без учителя у машинному навчанні, який використовується для групування схожих об'єктів у групи, що називаються кластерами. Об'єкти всередині одного кластеру мають більше спільних властивостей між собою, ніж із об'єктами з інших кластерів. Кластеризація використовується для виявлення структур або закономірностей у великих наборах даних, які не мають попередньо визначених міток. Її завдання — організувати дані у спосіб, який допоможе краще їх зрозуміти, сегментувати або аналізувати [15].

Основні цілі кластеризації такі:

- виявлення груп об'єктів - кластеризація клієнтів за схожими характеристиками для маркетингових цілей;

- спрощення даних - заміна великих наборів даних кількома узагальненими групами;

- підготовка для подальшого аналізу - кластеризація може використовуватися для створення міток для алгоритмів навчання з учителем.

Основними характеристиками кластерів є внутрішня схожість, де об'єкти всередині одного кластеру мають бути максимально схожими, зовнішня відмінність, коли об'єкти з різних кластерів мають бути максимально несхожими, розмір кластерів - може бути рівномірним або різним залежно від застосованого алгоритму.

Типи алгоритмів кластеризації наведено на рисунку 1.7.



Рисунок 1.7 – Типи кластеризації

K-Means кластеризація розділяє дані на фіксовану кількість k кластерів. Працює шляхом мінімізації відстані між об'єктами та центрами кластерів. Перевагою є простота та швидкість. Недолік полягає у потребі задання кількості кластерів заздалегідь.

Ієрархічна кластеризація будує дерево (дендрограму) для поступового об'єднання або розбиття кластерів. Перевагою є те, що не потребує задання кількості кластерів заздалегідь. Недолік полягає у складності для великих наборів даних.

DBSCAN групує об'єкти, які розташовані близько один до одного, і виявляє аномалії. Перевагою є те, що алгоритм ефективний для кластерів із довільними формами. Алгоритм чутливий до вибору гіперпараметрів.

Gaussian Mixture Models – це модель кластеризації, яка припускає, що дані є результатом накладання кількох нормальних розподілів. Модель враховує статистичну ймовірність приналежності до кластера. Недолік – складніший у реалізації, ніж K-Means [16, 17, 18].

1.3 Існуючі аналоги засобів оцінки продуктивності сайтів

Метрики для оцінки продуктивності сайту — це кількісні показники, які допомагають оцінити швидкість, стабільність і зручність використання веб-ресурсу. Вони визначають, наскільки ефективно сайт завантажується, реагує на дії користувача і забезпечує позитивний досвід роботи.

Google PageSpeed Insights (PSI) – це безкоштовний інструмент від Google, призначений для оцінки продуктивності вебсайтів як на мобільних пристроях, так і на комп'ютерах. Інструмент надає детальні звіти про час завантаження сторінок та продуктивність, а також пропонує рекомендації для покращення швидкості роботи сайту. Інструмент присвоює оцінку від 0 до 100 балів, де вищий показник зазвичай свідчить про те, що ваша сторінка добре оптимізована для швидкого завантаження та ефективної роботи [19].

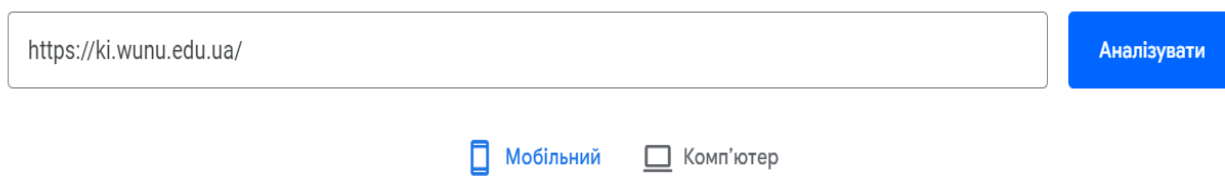
Largest Contentful Paint – час, необхідний для відображення найбільшого видимого елемента на сторінці (зображення, відео, великий блок тексту).

First Input Delay – час, що проходить між першою взаємодією користувача з сайтом (клік, натискання) та реакцією сторінки.

Cumulative Layout Shift Вимірює, наскільки макет сторінки змінюється під час завантаження (наприклад, коли кнопки чи зображення "стрибають").

При введенні URL-адреси веб-сторінки для аналізу, що представлено на рисунку. 1.8 інструмент генерує детальний звіт про її продуктивність. У ньому вказані проблемні моменти та потенційні шляхи їх вирішення, що спрощує процес оптимізації.

До того ж, Google PageSpeed Insights надає окремі оцінки продуктивності для веб-сторінки на мобільних пристроях та комп'ютерах.



https://ki.wunu.edu.ua/ Аналізувати

Мобільний Комп'ютер

Рисунок 1.8 – Введення веб-адреси сторінки для аналізу

Оптимізація інтерфейсу є важливим етапом покращення взаємодії користувачів із сайтом, особливо на мобільних пристроях. На рис. 1.9 наведено результати аналізу продуктивності, що демонструють ключові показники ефективності сайту. Далі детальніше розглянемо такі аспекти, як час завантаження контенту, блокування основних потоків, а також інші метрики, які впливають на швидкість і зручність використання сайту.

При аналізі продуктивності вебсайту важливо враховувати різні платформи, на яких він використовується. Для цього існують інструменти, що дозволяють оцінити роботу сайту як на мобільних пристроях, так і на комп'ютерах. Це допомагає розробникам виявляти та усувати можливі проблеми, які можуть впливати на користувацький досвід залежно від обраного пристрою.

Порівняльний аналіз систем оцінки продуктивності наведено у таблиці 1.2.

Таблиця 1.2 – Порівняльний аналіз систем оцінки продуктивності веб-сайтів

Критерій	GTmetrix	WebPageTest	Google PageSpeed
Зручність використання	Інтуїтивний інтерфейс з графіками, таблицями і розбивкою по метриках.	Орієнтований на досвідчених користувачів; надає детальні звіти, але менш зручний для новачків.	Дуже простий інтерфейс, адаптований для широкого кола користувачів. Лише основні метрики та поради.
Цільова аудиторія	Розробники, оптимізатори, маркетологи, які хочуть отримати як загальну, так і детальну інформацію.	Професіонали з веб-розробки, які потребують глибокого аналізу.	Початківці та власники сайтів, які хочуть отримати базову оцінку і прості рекомендації.
Переваги	Гнучкі налаштування, детальні рекомендації, інтеграція з Google Lighthouse.	Найбільша гнучкість у налаштуваннях, підтримка реальних умов завантаження.	Простота використання, акцент на мобільну оптимізацію, безкоштовний доступ.
Недоліки	Деякі функції доступні лише у платній версії.	Менш зручний інтерфейс, складність для новачків.	Мінімальна деталізація звітів, обмежені можливості налаштування.

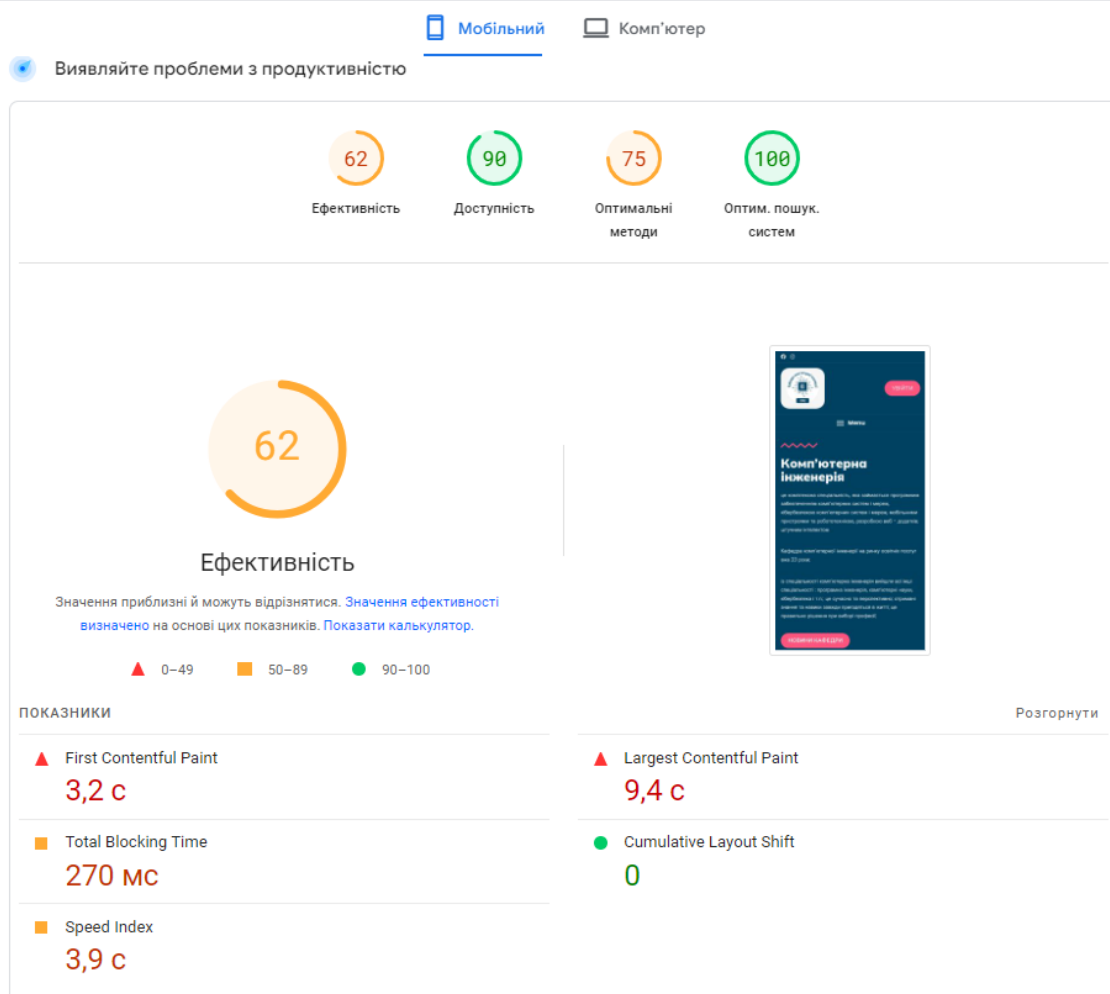


Рисунок 1.9 – Оптимізація інтерфейсу

Ці рекомендації можуть включати: оптимізацію зображень (сжаття, використання сучасних форматів), використання кешування браузера, мінімізацію JavaScript та CSS, зменшення кількості HTTP-запитів, використання асинхронного завантаження скриптів [20, 21, 22].

1.4 Аналіз завдання магістерської роботи та постановка задач дослідження

У цьому розділі було детально розглянуто ключові аспекти оцінки сайтів, включаючи засоби та методи, які дозволяють аналізувати їхню продуктивність та ефективність. Оцінка сайтів є важливим етапом у веб-розробці та маркетингу,

оскільки вона безпосередньо впливає на користувацький досвід, SEO-оптимізацію та бізнес-результати.

На основі аналітичного підходу проведено аналіз засобів ефективності веб-сайтів, що дозволило виділити основні критерії та їх актуальність для визначення ефективності сайту.

На основі аналітичного підходу проведено аналіз сучасних алгоритмів машинного навчання, що дозволило виділити алгоритми кластеризації, які найкраще підходять для аналізу заданого формату даних.

Для виконання мети потрібно вирішити наступні задачі:

- проаналізувати засоби оцінки ефективності веб-сайтів;
- проаналізувати алгоритми машинного навчання для опрацювання даних;
- розробити узагальнений алгоритм оцінки продуктивності веб-сайтів;
- розробити алгоритм кластеризації показників ефективності сайтів;
- програмно реалізувати графічний інтерфейс для задання даних для подальшого опрацювання засобами машинного навчання;
- провести порівняльний аналіз розробленого програмного модуля на основі кластеризації з аналогами;

2. АЛГОРИТМИ ОЦІНКИ ЕФЕКТИВНОСТІ ВЕБ-САЙТІВ

2.1 Узагальнений алгоритм оцінки ефективності веб-сайтів

Критерії оцінки продуктивності сайту є визначальним чинником в процесі розробки алгоритму оцінки продуктивності роботи, тому їх вибір є важливим завданням. Приклади критеріїв для опрацювання наведено нижче:

FSP вимірює час, необхідний для того, щоб браузер відобразив перший елемент вмісту сторінки (текст, зображення, SVG тощо) після початку завантаження. FSP показує, наскільки швидко сайт починає відображати видимий контент для користувача. Відображає початкове сприйняття швидкості завантаження. Впливає великий розмір CSS або JavaScript, оптимізація ресурсів (наприклад, зображень).

LCP вимірює час, коли найбільший видимий елемент на сторінці наприклад, зображення, відео або великий блок тексту стає видимим у вікні браузера. LCP показує, наскільки швидко завантажуються основний контент сторінки. Це ключовий показник для оцінки першого враження користувача. Впливають повільні сервери або хостинг, завантаження великих ресурсів (зображень або відео), затримки через JavaScript або CSS.

TBT вимірює загальний час, коли сторінка не реагує на дії користувача (наприклад, кліки, скролінг) через виконання JavaScript чи інших завдань. TBT відображає інтерактивність сайту під час завантаження. Чим нижчий показник TBT, тим швидше сайт стає інтерактивним. Впливає використання важких JavaScript-скриптів, неправильна оптимізація CSS або DOM.

CLS вимірює, наскільки часто і наскільки сильно зміщується макет сторінки під час завантаження. Це показник стабільності макета. CLS відображає, наскільки зручний сайт для перегляду. Високий CLS може дратувати користувачів, якщо вони натискають на неправильні елементи через зміщення. Впливає відсутність фіксованих розмірів для зображень і відео, завантаження

шрифтів, які змінюють макет, пізнє додавання елементів (наприклад, банерів або реклами).

SI оцінює, наскільки швидко відображається весь контент сторінки протягом її завантаження. Вимірюється в секундах та відображає загальну швидкість завантаження сторінки. Дає уявлення про те, наскільки плавно з'являється контент. Впливає низька швидкість інтернет-з'єднання, велика кількість зовнішніх запитів (ресурсів), неоптимізовані зображення та JavaScript.

FCP і LCP відображають, як швидко користувач бачить перший і основний контент сторінки. TBT визначає, як швидко користувач може взаємодіяти зі сторінкою. CLS показує, наскільки стабільним є макет під час завантаження. SI надає загальну оцінку швидкості завантаження. Разом ці показники забезпечують комплексну оцінку продуктивності сайту, відображаючи як технічні аспекти завантаження, так і зручність для користувача [23].

Алгоритм є основою вирішення задач у програмуванні, і правильне його представлення допомагає ефективно реалізувати логіку роботи системи. Обидва способи представлення — блок-схеми та описовий режим — мають свої переваги й особливості.

Блок-схема — це графічне зображення алгоритму за допомогою стандартних геометричних фігур (прямокутників, ромбів, стрілок тощо), які візуалізують послідовність дій, умови й рішення. Простий спосіб представлення алгоритму, який легко розуміється навіть нефахівцями. Відображає логіку процесу наочно, допомагаючи виявити помилки або неефективності.

В описовому режимі алгоритм описується словами як текстова послідовність дій або кроків, які потрібно виконати. Дозволяє використовувати будь-який стиль і рівень деталізації залежно від цілей [24].

Блок-схему узагальненого алгоритму оцінки продуктивності сайтів наведено на рисунку 2.1.



Рисунок 2.1 – Блок-схема узагальненого алгоритму оцінки продуктивності сайтів

Узагальнений алгоритм оцінки ефективності сайтів такий:

1. Вибір режиму роботи
 - десктопний режим;
 - мобільний режим.
2. Формування інформації про веб-ресурс. Збір даних, зокрема:
 - FCP (час до першого вмісту).
 - LCP (час до найбільшого вмісту).
 - TBT (загальний час блокування).
 - CLS (стабільність макета).
 - SI (швидкість відображення сторінки).
3. Визначення порогових значень для кожної метрики, для подальшої можливості відокремлення хороших показників від поганих.
Експериментальним шляхом підібрано такі порогові значення:

$$FCP = \begin{cases} t \leq 1.8c, 0 \\ 1.8 < t \leq 3.0c, 1 \\ t > 3.0c, 2 \end{cases}$$

$$LCP = \begin{cases} t \leq 2.5c, 0 \\ 2.5 < t \leq 4.0c, 1 \\ t > 4.0c, 2 \end{cases}$$

$$TBT = \begin{cases} t \leq 200mc, 0 \\ 200 < t \leq 600mc, 1 \\ t > 600mc, 2 \end{cases}$$

$$CLS = \begin{cases} t \leq 0.1, 0 \\ 0.1 < t \leq 0.25c, 1 \\ t > 0.25c, 2 \end{cases}$$

$$SI = \begin{cases} t \leq 4c, 0 \\ 4 < t \leq 6c, 1 \\ t > 6c, 2 \end{cases}$$

4. Підрахунок загальної кількості балів, чим нижче бал, тим кращим вважається сайт з точки зору продуктивності і навпаки, чим вищий бал, тим сайт вважається менш продуктивним.

Наприклад:

Якщо FCP = 2.0 секунд, то "Середньо" 2 бали.

Якщо CLS = 0.05, то "Добре" 3 бали.

5. Обчислення загального балу за результатами аналізу веб-сайту.
6. Формування навчальної вибірки з різними категоріями продуктивності роботи сайту.
7. Віднесення отриманих результатів до загального датасету для подальшого аналізу.
8. Етап кластеризації.
9. Віднесення результатів кластеризації до певного класу.

В алгоритмах кластеризації мітка класу – це умовне позначення, яке використовується для ідентифікації, до якого кластеру належить об'єкт. У кластеризації ці мітки зазвичай генеруються автоматично під час роботи алгоритму, на відміну від задач класифікації, де мітки визначаються заздалегідь.

2.2 Алгоритм кластеризації показників продуктивності веб-сайтів

Дані, що використовуються на етапі кластеризації, зазвичай представляються у формі структурованої таблиці або масиву числових векторів, залежно від особливостей задачі. Основна вимога – це підготовка даних у вигляді, придатному для обчислення відстаней або схожості між об'єктами.

Кожен рядок представляє окремий об'єкт або спостереження. Кожен стовпець відповідає одній числовій або категоріальній ознаці.

Якщо є n об'єктів і m ознак, дані мають розмір $n \times m$. Приклад наведено нижче:

Об'єкт Ознака_1 Ознака_2 Ознака_3

```
Obj1  2.5  1.8  3.0
Obj2  1.0  0.5  4.1
Obj3  3.2  1.3  3.8
```

У випадку використання масиву числових векторів, дані подаються у вигляді масиву або списку векторів, де кожен вектор відповідає одному об'єкту.

Приклад наведено нижче:

```
[[2.5, 1.8, 3.0],
 [1.0, 0.5, 4.1],
 [3.2, 1.3, 3.8]]
```

Узагальнений алгоритм складається з наступних кроків:

1) Приведемо значення показників ефективності веб-сайтів у $X_{n \times m}$, де m – номер показника, і n – значення показника.

2) Відцентрування та нормалізація вхідних даних відповідно до формули:

$$\frac{x_{ij} - \bar{x}_j}{\sigma_j},$$

де j – к-сть значень,

i – реалізація змінної j .

3) Розрахунок матриці кореляції (R).

4) Пошук власних значень $\lambda_1, \lambda_2, \dots, \lambda_p$ матриці R з характеристичного рівняння

$$\det(R - \lambda E) = 0,$$

де E – матриця.

5) Знаходження для кожного власного значення λ_j власний вектор із системи рівнянь:

$$(R - \lambda E) \cdot \vec{w} = 0,$$

де $\vec{w}$ – власний вектор.

6) Пошук лінійної комбінації для основних компонентів y_j :

$$y_j = w_{1j}x_1 + w_{2j}x_2 + \dots + w_{pj}x_p.$$

де w_{ij} – факторні навантаження ознак.

7) Аналіз внеску кожного основного компонента та їх рейтинг у порядку зростання. Внесок кожного компонента розраховується за формулою:

$$B_j = \frac{\lambda_j}{\lambda_1 + \lambda_2 + \dots + \lambda_p},$$

Після скорочення даних ми отримуємо багато найбільш інформативних функцій p : $X = \{X_1, X_2, \dots, X_p\}$, де $p \ll m$.

8) Для визначення кількості кластерів ми використовуємо метод Elbow, який базується на обчисленні мінімальної сумарної квадратичної помилки (MSSE)

$$K = \min_{c_1, \dots, c_k} \sum_{i=1}^n \min \left\{ \|x_i - c_1\|^2, \dots, \|x_i - c_k\|^2 \right\}.$$

9) Використовуючи метод К-середніх, знаходимо центри кластерів

$$C_k = \frac{\sum_{x_i \in C_k} x_i}{|C_k|}.$$

10) Щоб побудувати дендрограму, ми розраховуємо відстані між кластерами на основі відстані одного зв'язку:

$$d_{\min}(C_i, C_j) = \min_{\substack{x_i \in C_i \\ x_j \in C_j}} d(x_i, x_j).$$

11) За допомогою методу DBScan поділ на кластери здійснюється на основі поділу за щільністю точок за виразом:

$$N_{\varepsilon}(p) = \{q \in X | \text{dist}(p, q) \leq \varepsilon\},$$

де p – головна точки, q – ε - досяжна точка в кластері.

5) Якість кластеризації оцінюємо на основі ранд-індексу за виразом:

$$RI = \frac{TP + TN}{TP + FP + FN + TN},$$

де TP – кількість істинних спрацьовувань;

TN – кількість істинних негативів;

FP – кількість помилкових спрацьовувань;

FN – кількість помилкових негативів.

Структуру програмного модулю наведено на рисунку 2.2.



Рисунок 2.2 – Структура програмного модулю

Блок завантаження даних відповідає за отримання вхідних даних, які необхідні для кластеризації показників продуктивності веб-сайтів, та їх первинну обробку. Його завдання — забезпечити правильне зчитування, перевірку, структурування та передачу даних до наступних блоків системи.

Використовуються такі формати локальних файлів: CSV, Excel, JSON або інші поширені формати даних. Також використовується формат SQL-запитів для зчитування даних із баз даних. Також допустиме зчитування даних із зовнішніх джерел через API (наприклад, з GTmetrix, WebPageTest, або Google PageSpeed Insights). Додатково наявне ручне введення, а саме інтерфейс для додавання даних вручну. Структуру вхідних даних наведено на рисунку 2.3.

example.com	1.5	2.8	300	0.05	3.2
testsite.org	2.0	3.1	250	0.1	4.0

Рисунок 2.3 – Приклад вхідних даних

Приклад коду для опрацювання вхідних даних наведено нижче:

```
# Перевірка наявності необхідних колонок
required_columns = ['Website', 'FCP', 'LCP', 'TBT', 'CLS',
'SI']
if not all(col in data.columns for col in required_columns):
    raise ValueError("Відсутні необхідні показники в
даних.")
```

Блок завантаження даних є критичним елементом системи кластеризації, оскільки від його коректності залежить точність і ефективність наступних етапів. Він має забезпечити гнучкість роботи з різними форматами даних, а також враховувати всі особливості показників продуктивності веб-сайтів.

Блок для формування параметрів для кластеризації у програмній системі для оцінки продуктивності вебсайтів є важливим компонентом, який дозволяє

правильно налаштувати процес кластеризації даних. На першому етапі цей блок забезпечує збирання даних про продуктивність вебсайтів, таких як час завантаження сторінок, швидкість відповіді сервера, кількість помилок, розмір сторінки, кількість HTTP-запитів, час завантаження ресурсів (CSS, JS, зображень) тощо. Дані можуть бути зібрані за допомогою веб-аналітики, спеціалізованих інструментів моніторингу чи API для збору інформації про продуктивність (наприклад, Google Lighthouse, Web Vitals, або власні інструменти). В цьому блоці відбувається очищення та нормалізація даних. Наприклад, вибір аномальних значень (викидів), нормалізація метрик (перетворення значень в одноманітний масштаб) та стандартизація різних типів даних для подальшого аналізу. Вибір лише релевантних показників для кластеризації, тобто тих, які найбільше впливають на продуктивність та мають значення для кластеризації вебсайтів.

Користувач може налаштувати, які саме параметри будуть використовуватися для кластеризації. Це може включати час завантаження сторінки, кількість запитів до сервера, розмір сторінки, ступінь оптимізації зображень тощо. Можливість комбінування декількох параметрів, таких як час відповіді серверу і кількість запитів, щоб отримати більш точну картину про продуктивність.

Для кластеризації важливо вибрати метрику для вимірювання подібності між вебсайтами. Це може бути евклідова відстань, манхеттенська відстань або спеціалізовані метрики для вимірювання продуктивності (наприклад, метрика для часового аналізу).

Користувач може вибрати метод кластеризації, наприклад, k-means, DBSCAN, ієрархічну кластеризацію або інші методи, що дозволяють знаходити групи схожих вебсайтів. Налаштування кількості кластерів (для k-means) або інших параметрів, які можуть впливати на результат кластеризації [25].

Для зручності користувача, блок може надавати візуалізацію вибраних параметрів: графіки, таблиці або діаграми, що дозволяють зрозуміти, як різні

параметри впливають на продуктивність вебсайтів та допомагають у виборі оптимальних для кластеризації.

Цей блок є критичним для успішної кластеризації, оскільки правильний вибір параметрів та їх налаштування можуть значно покращити точність і ефективність класифікації вебсайтів за їх продуктивністю.

Блок адаптивної обробки для кластеризації в програмній системі оцінки продуктивності вебсайтів призначений для того, щоб забезпечити гнучке і динамічне налаштування параметрів кластеризації залежно від змін у даних, умовах роботи системи та потребах користувачів. Блок адаптивної обробки здатний виявляти та коригувати аномальні або неправильні дані, що можуть з'являтися через помилки збору інформації або випадкові відмови сервера. Наприклад, якщо час завантаження сторінки суттєво вищий за звичайний, система може ігнорувати або коригувати ці показники. Оскільки вебсайти можуть демонструвати різну продуктивність залежно від часу доби, навантаження на сервер чи інших зовнішніх факторів, блок адаптивної обробки може динамічно змінювати параметри нормалізації даних. Наприклад, якщо система виявляє, що певні вебсайти демонструють різні тренди в різний час, це може бути враховано при обробці даних.

Система може самостійно адаптувати набір параметрів для кластеризації в залежності від характеристик даних. Якщо, наприклад, виявлено, що одна з метрик (наприклад, час завантаження) має набагато більшу варіативність, ніж інші, вона може бути надана більшої ваги.

В залежності від кількості та різноманіття даних, блок адаптивної обробки може вибирати оптимальний метод кластеризації. Наприклад, при великій кількості даних (сотні тисяч вебсайтів) може бути обраний алгоритм, який працює швидше (наприклад, DBSCAN), а при меншій кількості — більш точний (наприклад, k-means або ієрархічна кластеризація).

Якщо система виявляє певну зміну у характері даних (наприклад, зміна патернів у часі завантаження вебсайтів через оновлення чи зміни у роботі серверів), вона може адаптувати кількість кластерів (для k-means) або інші

гіперпараметри (наприклад, мінімальну кількість точок для DBSCAN) для покращення точності кластеризації.

Блок адаптивної обробки здатний автоматично визначати вплив сезонних або часових змін на продуктивність вебсайтів. Це може бути корисним для виявлення змін у швидкості завантаження або часу відгуку, пов'язаних із підвищеним навантаженням на сервери в певні періоди.

Блок адаптивної обробки може надавати рекомендації користувачу щодо оптимальних параметрів кластеризації на основі аналізу поточних даних, таких як зміни у часі завантаження, сезонні тренди або інші фактори.

Після виконання кластеризації система може оцінювати ефективність результатів за допомогою метрик, таких як силует-коефіцієнт або внутрішня когерентність кластерів. Якщо результат кластеризації не відповідає очікуванням, блок адаптивної обробки може налаштувати параметри алгоритмів, щоб досягти кращих результатів.

Блок адаптивної обробки для кластеризації в системі оцінки продуктивності вебсайтів дозволяє автоматично налаштовувати та оптимізувати процес кластеризації з урахуванням змін у даних, потреб користувачів і зовнішніх факторів. Це забезпечує більш точну і гнучку кластеризацію, що дозволяє оперативно реагувати на зміни у продуктивності вебсайтів і отримувати більш коректні результати для подальшого аналізу та оптимізації.

Блок кластеризації в програмній системі оцінки продуктивності вебсайтів є основним компонентом, який виконує сам процес групування вебсайтів на основі схожості їхніх показників продуктивності. Цей блок відповідає за застосування алгоритмів кластеризації, визначення кількості та характеристик кластерів, а також за інтерпретацію результатів кластеризації для подальшого аналізу.

Блок кластеризації отримує від попередніх блоків систему підготовлених і очищених даних про продуктивність вебсайтів, таких як час завантаження сторінки, кількість запитів до серверів, час відповіді серверу, розмір сторінки, час завантаження різних ресурсів (зображень, JavaScript, CSS) тощо.

Блок кластеризації має вибір з кількох алгоритмів кластеризації, таких як: k-means: Класичний метод, який працює добре при наявності значної кількості даних і вимагає попереднього визначення кількості кластерів (k).

DBSCAN – алгоритм, який добре працює з неявними кластерами і дозволяє знаходити будь-які форми кластерів, ідеально підходить для даних з великою кількістю шуму.

Ієрархічна кластеризація побудова ієрархії кластерів, що дозволяє зрозуміти, як різні групи вебсайтів співвідносяться між собою.

В залежності від обраних параметрів даних (їх кількості, складності, шуму тощо) система може автоматично вибирати найбільш підходящий метод кластеризації для поточних умов.

Після виконання система кластеризації може оцінити якість результатів за допомогою метрики. Внутрішня когерентність вимірює, збільшує схожі елементи в середині кластерів. Розрізнення між кластерами оцінює, наскільки відмінні кластери один від одного.

Якщо виявляється, що деякі кластери мають низьку якість або містять значну кількість аномальних точок, блок кластеризації може перерозподілити елементи або змінити параметри кластеризації [26].

2.3 Структура бази даних

Актуальність та важливість використання бази даних для зберігання результатів опрацювання великих даних у машинному навчанні (МН) зростають в умовах постійного зростання обсягів даних і складності моделей. Зберігання результатів аналізу в базі даних є ключовим етапом у роботі з великими даними. Бази даних дозволяють організувати великі обсяги даних у структурованому вигляді, що спрощує доступ і аналіз. У контексті машинного навчання часто

опрацьовуються масиви даних, що включають числові значення, текстові дані, зображення чи навіть звукові сигнали. База даних забезпечує:

- індексацію даних;
- нормалізацію;
- стандартизацію.

Однією з головних переваг використання баз даних є їх здатність до масштабування при роботі з великими даними. У машинному навчанні часто виникає потреба в опрацюванні величезних обсягів інформації, і база даних може масштабуватися як вертикально (додавання потужностей до одного сервера), так і горизонтально (розподілена система з кількох серверів).

Для розробки системи оцінки швидкодії сайтів з допомогою кластеризації, яка використовує дані з таких джерел як GTMetrix, WebPageTest, Google Pagespeed, і надає оцінку швидкодії для кожного сайту (добре, середньо, погано) створено такі таблиці бази даних. Таблиця «websites» містить основну інформацію про сайти, для яких буде проводитися оцінка швидкодії. Приклад SQL коду для створення таблиці:

```
CREATE TABLE websites (  
    id INT AUTO_INCREMENT PRIMARY KEY, -- Унікальний ідентифікатор  
    сайту  
    name VARCHAR(255) NOT NULL, -- Назва сайту (наприклад,  
'example.com')  
    url VARCHAR(255) NOT NULL UNIQUE -- URL сайту  
);
```

Таблиця «performance_metrics» призначена для зберігання основних показників швидкодії сайту з різних джерел (GTMetrix, WebPageTest, Google Pagespeed).

```
CREATE TABLE performance_metrics (  
    id INT AUTO_INCREMENT PRIMARY KEY, -- Унікальний ідентифікатор  
    для кожного запису  
    website_id INT NOT NULL, -- Зовнішній ключ до таблиці  
    websites  
    gtmatrix_score DECIMAL(5, 2), -- Оцінка з GTMetrix  
    webpagetest_score DECIMAL(5, 2), -- Оцінка з WebPageTest  
    pagespeed_score DECIMAL(5, 2), -- Оцінка з Google Pagespeed
```

```

        date_recorded TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- Дата
запису результату
        FOREIGN KEY (website_id) REFERENCES websites(id) -- Зв'язок з
таблицею websites
    );

```

Використання бази даних для зберігання результатів обробки великих даних у машинному навчанні є важливим елементом інфраструктури, оскільки забезпечує ефективне зберігання, доступність, цілісність і масштабованість даних. Це дозволяє зберігати важливу інформацію, швидко виконувати аналітику, інтегрувати з іншими системами, а також забезпечувати безпеку та надійність у роботі з великими даними.

2.4 Висновки до розділу

На основі алгоритмічного підходу, розроблено узагальнений алгоритм оцінки ефективності веб-сайтів, що вміщує основні етапи опрацювання показників продуктивності сайту та етапи опрацювання даних засобами нечіткої логіки та кластеризації.

На основі алгоритмічного підходу, розроблено алгоритм на основі кластеризації для оцінки продуктивності сайту на основі вхідних показників із сервісів GTMetrix, WebPageTest, Google Pagespeed, що дозволило класифікувати сайти по критерію продуктивності.

Розроблено структуру бази даних для зберігання результатів опрацювання даних для кластеризації показників веб-сайтів. Структура складається з таблиць для зберігання вхідних параметрів та результатів опрацювання.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ АЛГОРИТМІВ ОЦІНКИ ЕФЕКТИВНОСТІ ВЕБ-САЙТІВ

3.1 Графічний інтерфейс розробленого модулю

Розробка якісного графічного інтерфейсу (GUI) для задач кластеризації має велике значення, оскільки вона допомагає користувачам краще зрозуміти результати аналізу, ефективно працювати з даними та приймати обґрунтовані рішення. Задачі кластеризації часто працюють з багатовимірними даними, які важко інтерпретувати без відповідної візуалізації. Графічний інтерфейс дозволяє представити дані у вигляді графіків, таких як розподіли, діаграми розсіювання, теплові карти та використовувати техніки зниження розмірності (наприклад, PCA, t-SNE) для відображення кластерів у 2D чи 3D просторах. Інтуїтивний графічний інтерфейс допомагає користувачам без спеціальної підготовки краще інтерпретувати результати кластеризації. Наприклад, кольорове кодування кластерів або показ відстаней між кластерами на графіках допомагають швидко зрозуміти структуру даних. Це дозволяє приймати більш обґрунтовані рішення на основі кластерних моделей.

Ефективний графічний інтерфейс робить задачі кластеризації доступними навіть для користувачів, які не є експертами у програмуванні або математичному аналізі. Графічний інтерфейс може дозволити порівнювати результати кількох алгоритмів кластеризації (наприклад, K-Means, DBSCAN, агломеративна кластеризація) для вибору найкращої моделі.

Якісний графічний інтерфейс є ключовим елементом у задачах кластеризації. Він робить аналіз зручнішим, підвищує ефективність і точність, а також забезпечує доступність технології для користувачів з різним рівнем технічної підготовки [27].

Приклад вікна графічного інтерфейсу для опрацювання вхідних параметрів розробленої системи наведено на рисунку 3.1.

Аналіз продуктивності сайту на основі показників GTMetrix

Введіть базові показники для оцінки продуктивності сайту

URL сайту:

Введіть URL сайту

Оцінка GTMetrix:

0-100

Оцінка WebPageTest:

0-100

Оцінка Google Pagespeed:

0-100

Оцінити продуктивність

© 2024 Аналіз продуктивності сайтів

Рисунок 3.1 – Приклад вікна графічного інтерфейсу для опрацювання вхідних параметрів

Форма вводу даних складається з таких полів:

URL сайту - вводиться в поле для вводу URL.

GTMetrix, WebPageTest, Google Pagespeed - вводяться оцінки продуктивності сайту для кожного з цих інструментів. Оцінки можуть варіюватися від 0 до 100.

Після заповнення форми та натискання кнопки "Оцінити продуктивність", дані відправляються на сервер для отримання результатів і відображення оцінки.

Після надсилання форми, з'являється розділ, де буде показано загальний результат оцінки на основі введених даних.

Приклад коду сторінки наведено нижче:

```
<section class="form-section">
  <h2>Введіть базові показники для оцінки продуктивності сайту</h2>

  <form id="performance-form">
    <div class="form-group">
      <label for="url">URL сайту:</label>
      <input type="url" id="url" name="url"
placeholder="Введіть URL сайту" required>
    </div>
```

Приклад css – стилів для формування сторінки наведено нижче:

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f9;
}

header {
  background-color: #007bff;
  color: white;
  text-align: center;
  padding: 20px;
}
```

Коли користувач натискає кнопку "Оцінити продуктивність", форма не надсилається, а дані з полів вводяться в змінні.

Оцінки з трьох джерел (GTMetrix, WebPageTest, Google Pagespeed) обчислюються в середнє значення.

На основі середнього балу, визначається оцінка продуктивності: "Добре" (80 і більше), "Середньо" (60-79), "Погано" (менше 60).

Після обчислення, результат відображається у спеціальному розділі.

Приклад Javascript коду для відправлення даних по REST API:

```
const url = document.getElementById('url').value;
const          gtmetrixScore          =
parseFloat(document.getElementById('gtmetrix-score').value);
const          webpagetestScore       =
parseFloat(document.getElementById('webpagetest-score').value);
const          pagespeedScore         =
parseFloat(document.getElementById('pagespeed-score').value);
```

Коли користувач натискає кнопку "Оцінити продуктивність", подія форми відміняється за допомогою `event.preventDefault()`. Далі збираються всі введені дані у вигляді об'єкта. За допомогою методу `fetch()` дані відправляються на сервер через POST запит. Серверна частина (яка знаходиться за адресою `https://example.com/api/evaluate`) повинна приймати ці дані, обробляти їх і повертати результат. Відповідь від сервера обробляється і на основі середнього балу виводиться категорія продуктивності: "Добре", "Середньо", "Погано".

3.2 Формування набору даних

Формування якісного датасету для задач кластеризації даних є критично важливим етапом, оскільки успішність кластеризації залежить від коректності, повноти та структурованості даних. Кластеризація, на відміну від задач класифікації, не має наперед визначених цільових міток, тому якість даних безпосередньо впливає на достовірність результатів. Датасет має відображати всі важливі характеристики аналізованої системи чи процесу. Нерепрезентативний набір даних може призвести до неправильного формування кластерів, що вплине на інтерпретацію результатів. Дані, що містять помилки, пропуски або аномалії, можуть викликати неправильне визначення кластерів та перекручення структури даних. Важливо проводити попередню обробку: видаляти шум, заповнювати пропуски, масштабувати значення. Датасет повинен включати лише ті параметри, які є важливими для формування кластерів. Зайві або нерелевантні фічі можуть ускладнити аналіз та призвести до переобчислення або втрати важливої інформації через високий розмірність.

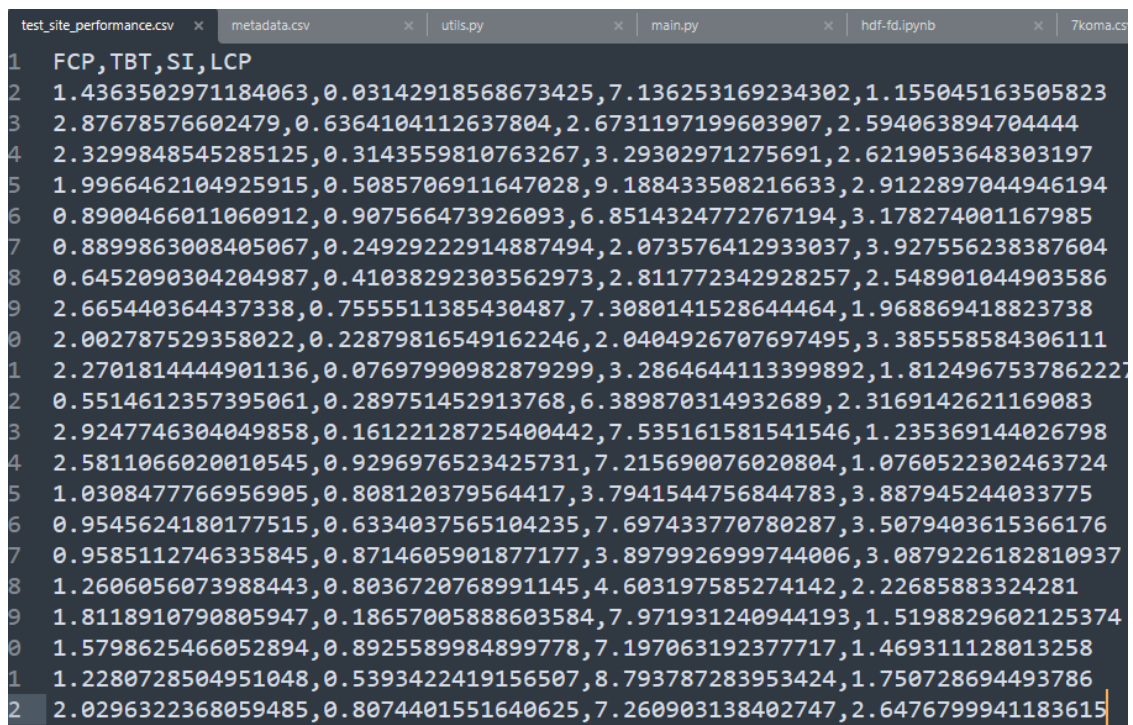
Користувачі очікують, що сайти будуть завантажуватися швидко та працювати плавно. Якщо сторінка завантажується більше 3 секунд, значна частина відвідувачів може залишити її, навіть не переглянувши вміст. Високі оцінки GTMetrix допомагають створити позитивне враження, що сприяє залученню аудиторії та підвищенню лояльності. У сучасному світі більшість користувачів заходять на сайти через мобільні пристрої. Хороші показники GTMetrix гарантують швидку та комфортну роботу сайту навіть за умов слабкого інтернет-з'єднання.

Алгоритми кластеризації, такі як K-Means або DBSCAN, краще працюють із більшими обсягами даних, оскільки вони можуть ефективніше виявляти структури. Однак, надто великий обсяг необроблених даних може призвести до перевантаження. Датасет для кластеризації повинен бути підготовлений таким

чином, щоб всі параметри мали однакову вагу в процесі обчислення. Наприклад, показники з різними одиницями вимірювання (долари, секунди, кілометри) потрібно масштабувати або нормалізувати.

Приклад сформованого датасету для даних експериментів наведено на рисунку 2.2.

Аномалії можуть суттєво вплинути на результати кластеризації, формуючи окремі "шумові" кластери. Важливо або видаляти аномальні дані, або використовувати алгоритми, стійкі до них (наприклад, DBSCAN). Якщо деякі групи в даних значно переважають інші, алгоритм може не виявити менш представлені кластери. Збалансованість забезпечує більш точний і коректний розподіл.



The screenshot shows a Jupyter Notebook interface with several tabs: 'test_site_performance.csv', 'metadata.csv', 'utils.py', 'main.py', 'hdf-fd.ipynb', and '7koma.cs'. The active tab is 'test_site_performance.csv', which displays a list of 20 rows of data. Each row contains a line number (from 1 to 20) followed by a header 'FCP, TBT, SI, LCP' and then a comma-separated list of 10 numerical values. The values are formatted with commas as thousands separators. The last row (line 20) is highlighted with a blue background.

```
1 FCP, TBT, SI, LCP
2 1.4363502971184063,0.03142918568673425,7.136253169234302,1.155045163505823
3 2.87678576602479,0.6364104112637804,2.6731197199603907,2.594063894704444
4 2.3299848545285125,0.3143559810763267,3.29302971275691,2.6219053648303197
5 1.9966462104925915,0.5085706911647028,9.188433508216633,2.9122897044946194
6 0.8900466011060912,0.907566473926093,6.8514324772767194,3.178274001167985
7 0.8899863008405067,0.24929222914887494,2.073576412933037,3.927556238387604
8 0.6452090304204987,0.41038292303562973,2.811772342928257,2.548901044903586
9 2.665440364437338,0.7555511385430487,7.3080141528644464,1.968869418823738
0 2.002787529358022,0.22879816549162246,2.0404926707697495,3.385558584306111
1 2.2701814444901136,0.07697990982879299,3.2864644113399892,1.8124967537862227
2 0.5514612357395061,0.289751452913768,6.389870314932689,2.3169142621169083
3 2.9247746304049858,0.16122128725400442,7.535161581541546,1.235369144026798
4 2.5811066020010545,0.9296976523425731,7.215690076020804,1.0760522302463724
5 1.0308477766956905,0.808120379564417,3.7941544756844783,3.887945244033775
6 0.9545624180177515,0.6334037565104235,7.697433770780287,3.5079403615366176
7 0.9585112746335845,0.8714605901877177,3.8979926999744006,3.0879226182810937
8 1.2606056073988443,0.8036720768991145,4.603197585274142,2.22685883324281
9 1.8118910790805947,0.18657005888603584,7.971931240944193,1.5198829602125374
0 1.5798625466052894,0.8925589984899778,7.197063192377717,1.469311128013258
1 1.2280728504951048,0.5393422419156507,8.793787283953424,1.750728694493786
2 2.0296322368059485,0.8074401551640625,7.260903138402747,2.6476799941183615
```

Рисунок 2.2 – Приклад сформованого датасету

Приклад показників для «швидкого» сайту наведено на рисунку 2.2.

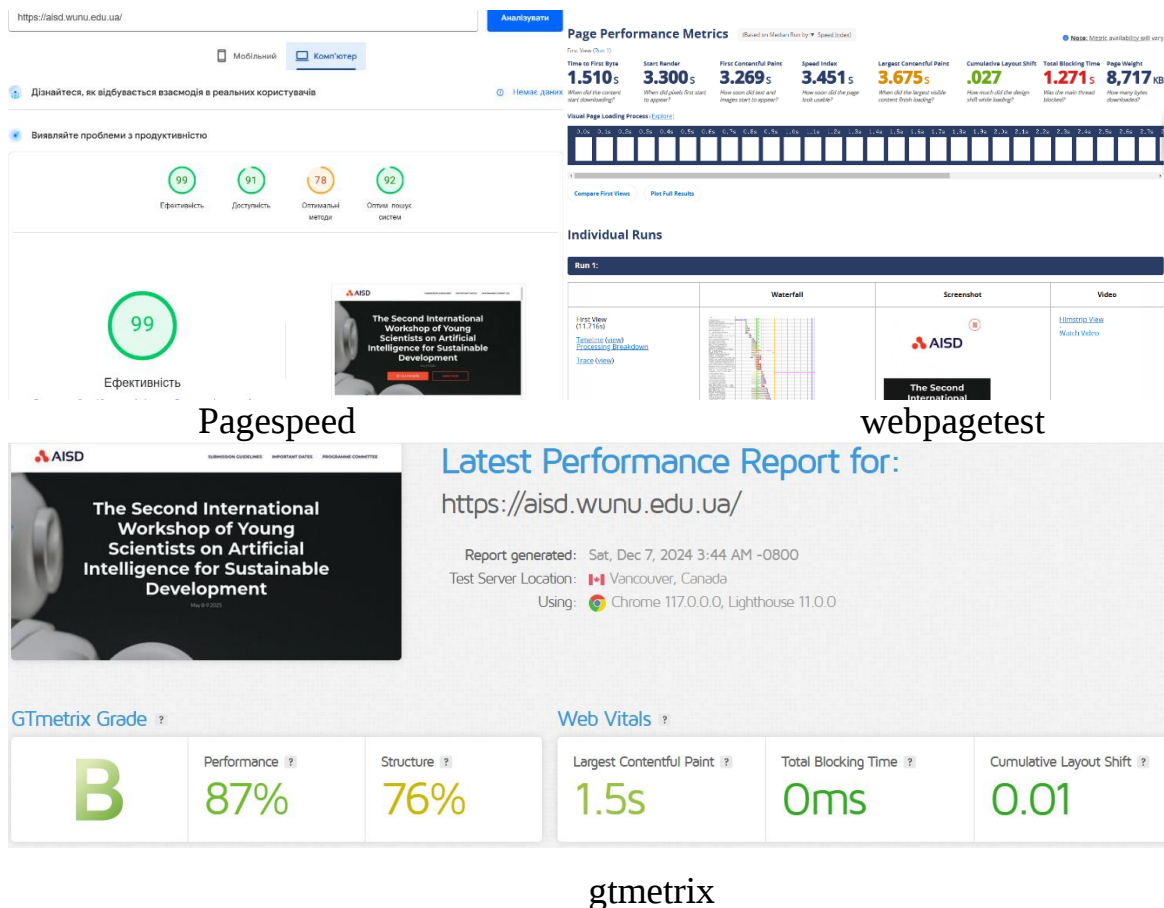


Рисунок 2.2 – Приклад показників для «швидкого» сайту

Швидкість завантаження сайту – це важливий фактор ранжування в пошукових системах, таких як Google. Хороші показники за GTmetrix допомагають забезпечити кращі позиції в пошуковій видачі, що безпосередньо впливає на трафік та потенційні конверсії.

Якщо сайт завантажується повільно, користувачі можуть покинути його, навіть не ознайомившись зі змістом. Це збільшує показник відмов (bounce rate), що негативно впливає на сприйняття сайту пошуковими системами. Погано оптимізований сайт може створювати зайве навантаження на сервери, що призводить до підвищених витрат на хостинг. Поліпшення продуктивності допомагає оптимізувати споживання ресурсів, зменшуючи витрати [28].

Приклад показників для «повільного» сайту наведено на рисунку 2.3.

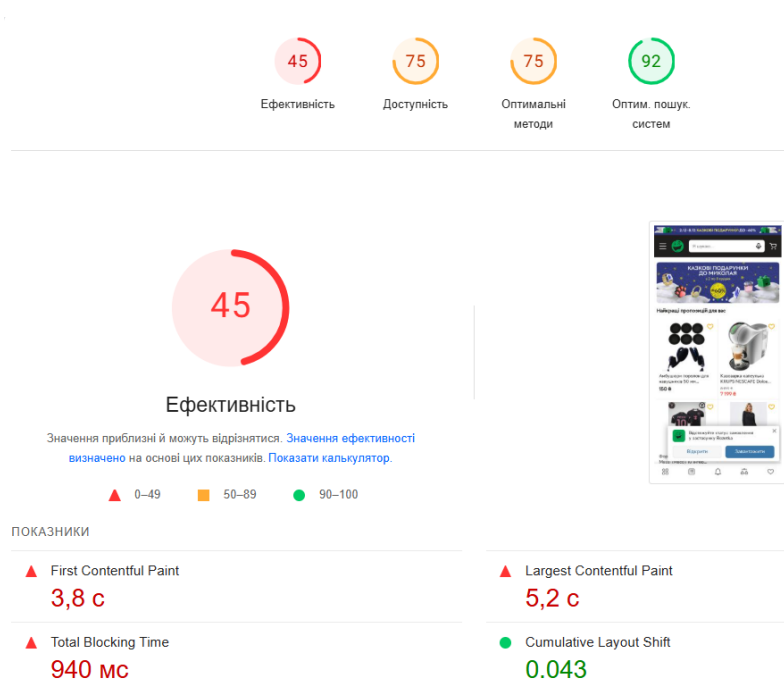


Рисунок 2.3 – Приклад показників для «повільного» сайту

Оптимізація зображень – використання сучасних форматів (WebP) та стиснення. Використання кешування зменшує час завантаження сторінки для повторних відвідувачів. Мінімізація файлів скорочуйте CSS, JavaScript і HTML. Зменшення кількості запитів об'єднуйте файли та використовуйте CDN (мережу доставки контенту).

CSV (Comma-Separated Values) файли є популярним форматом для зберігання та обміну даними у задачах машинного навчання (ML). Вони мають низку особливостей, які роблять їх зручними для роботи, але також накладають певні обмеження та вимагають дотримання правил. CSV-файли зберігають дані у вигляді тексту, що робить їх простими для читання людиною та сумісними з багатьма програмами та мовами програмування. Вони підтримуються майже всіма інструментами для аналізу даних та ML, включаючи Python, R, Excel, і бази даних. Кома (,) зазвичай використовується як розділювач, але можуть застосовуватись інші символи, такі як ; \t (табуляція), або пробіл.

Формування датасету для задач кластеризації є основою успішного аналізу даних. Якісно підготовлений набір даних забезпечує:

- виявлення прихованих структур.

- коректне групування об'єктів.
- застосування результатів кластеризації для прийняття обґрунтованих рішень.

Усі значення зберігаються як текст, тому їх потрібно конвертувати у відповідні типи (числові, категоріальні) перед використанням в ML. У дному форматі можуть бути пусті клітинки, які потрібно обробити (заповнити, видалити або ігнорувати). ані можуть містити помилки (наприклад, текст у числових полях). Можливі повтори записів, які можуть викривити аналіз та використання різних розділювачів у одному файлі може спричинити проблеми під час обробки.

CSV-файли легко завантажувати та обробляти за допомогою бібліотек, таких як pandas (Python). Вони компактні, оскільки не містять зайвого форматування. Можуть зберігати як числові, так і категоріальні дані.

CSV-файли не містять інформації про типи даних або структуру, що може викликати труднощі при їх обробці. Для дуже великих датасетів ефективніше використовувати спеціалізовані формати, наприклад, Parquet або HDF5. CSV призначений для табличних даних, тому не може зберігати вкладені чи ієрархічні структури.

CSV-файли є потужним інструментом для роботи з табличними даними у машинному навчанні завдяки їхній простоті та універсальності. Проте, для їх ефективного використання важливо правильно підготувати дані, враховуючи їх можливі недоліки та обмеження [29].

3.3 REST API

REST API є стандартом для побудови веб-сервісів, які дозволяють обмінюватися даними між додатками. У контексті машинного навчання (ML) та розробки загалом REST API є важливим інструментом для інтеграції,

масштабування та доступності рішень. REST API дозволяє легко інтегрувати ML-моделі в різні системи. Модель можна викликати з будь-якої мови програмування або платформи, яка підтримує HTTP-запити. Моделі доступні в реальному часі через мережу, що дозволяє отримувати прогнози для користувачів. Легко інтегрується з хмарними платформами для масштабування запитів.

У сучасних додатках все частіше використовують мікросервісну архітектуру, де кожен компонент (наприклад, ML-модель, база даних, інтерфейс користувача) є окремим сервісом. REST API забезпечує:

- комунікацію між сервісами: Наприклад, ML-моделі можуть працювати окремо як сервіс, до якого звертаються інші компоненти.
- зручність підтримки: Кожен сервіс можна оновлювати чи змінювати без впливу на інші.

API можна використовувати для отримання даних із зовнішніх джерел (соціальні мережі, біржі, відкриті API). втоматизація цих етапів через REST API спрощує CI/CD процеси. API надає доступ до метрик продуктивності або логів роботи моделі [30].

Принцип роботи REST API у поєднанні із Flask наведено на рисунку 3.2.

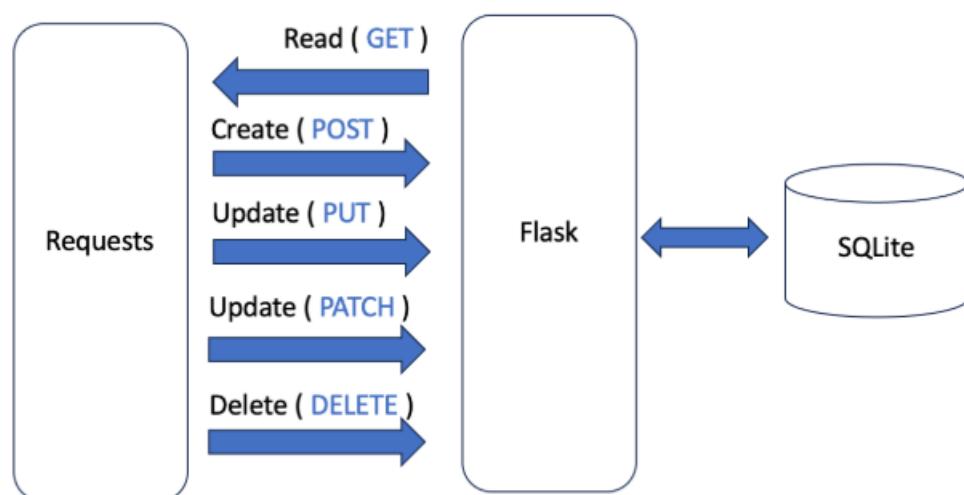


Рисунок 3.2 – Принцип роботи REST API у поєднанні із Flask

`fetch()` відправляє JSON-дані через POST-запит. Сервер обробляє запит і повертає результат у форматі JSON. Дані повертаються і відображаються на сторінці у секції результатів.

Для реалізації серверної частини необхідно налаштувати API на сервері, який буде приймати POST-запити, обробляти їх і повертати відповідь. Це можна зробити за допомогою будь-якої серверної технології, наприклад, Node.js, Python (Flask/Django), PHP тощо.

Щоб обробити вхідні дані через REST API на сервері за допомогою Python, зазвичай використовують фреймворки, такі як Flask або FastAPI. Функція `calculate_performance` отримує три оцінки: GTMetrix, WebPageTest і Google Pagespeed. Обчислює середній бал і на його основі визначає загальну оцінку продуктивності:

```
"Добре" (якщо середній бал ≥ 80)
"Середньо" (якщо середній бал від 60 до 79)
"Погано" (якщо середній бал < 60)
```

Маршрут `«/api/evaluate»` приймає POST-запит з даними в форматі JSON. Запит містить URL сайту і оцінки з трьох інструментів. Якщо всі необхідні дані передані, викликається функція для обчислення оцінки продуктивності і результат повертається у вигляді JSON.

Приклад POST-запиту:

```
{
  "url": "https://example.com",
  "gtmetrix_score": 85,
  "webpagetest_score": 75,
  "pagespeed_score": 80
}
```

Приклад відповіді:

```
{
  "url": "https://example.com",
  "performance_rating": "Добре"
}
```

Таблиця `performance_categories` визначає категорії оцінок для швидкодії сайту. Приклад коду для додавання записів:

```
INSERT INTO performance_categories (name, min_score, max_score)
VALUES ('Добре', 80.00, 100.00),
       ('Середньо', 60.00, 79.99),
       ('Погано', 0.00, 59.99);
```

Таблиця `website_ratings` зберігає оцінки для кожного сайту на основі отриманих результатів. Приклад коду для додавання записів:

```
INSERT INTO website_ratings (website_id, category_id)
VALUES (1, 1), -- 'Example Site 1' отримує оцінку "Добре"
       (2, 2), -- 'Example Site 2' отримує оцінку "Середньо"
       (3, 1); -- 'Example Site 3' отримує оцінку "Добре"
```

Таблиця `cluster_results` зберігає результати кластеризації, вказуючи до якого кластеру належить кожен сайт. Приклад коду для додавання записів:

```
INSERT INTO cluster_results (website_id, cluster_id)
VALUES (1, 1), -- 'Example Site 1' належить до кластеру 1
       (2, 2), -- 'Example Site 2' належить до кластеру 2
       (3, 1); -- 'Example Site 3' належить до кластеру 1
```

Розроблені таблиці дозволяють зберігати інформацію про дослідження та зберігати результати для подальшого аналізу і повторних досліджень.

3.4 Порівняльний аналіз отриманих результатів

Elbow метод – це техніка, яка використовується для визначення оптимальної кількості кластерів у задачах кластеризації, наприклад, у методі K-Means. Його назва походить від форми графіка, яка нагадує лікоть ("elbow"). У задачах із шумними даними або неправильною структурою Elbow метод може дати неоднозначний результат. В складних випадках використовують додаткові

методи, наприклад, силуетний аналіз чи метод пропорційної зміни похідної. Elbow метод є ефективним підходом для визначення оптимальної кількості кластерів у багатьох практичних задачах [31]. Характеристику датасету, використану для аналізу наведено на рисунку 3.2.

	FCP	TBT	SI	LCP
count	100.000000	100.000000	100.000000	100.000000
mean	1.675452	0.497832	6.140811	2.473447
std	0.743724	0.293111	2.347410	0.880357
min	0.513805	0.006952	2.040493	1.043180
25%	0.983002	0.242005	4.215039	1.748845
50%	1.660356	0.505625	6.500439	2.529155
75%	2.325508	0.766184	8.018936	3.207333
max	2.967217	0.985650	9.920431	3.971515

Рисунок 3.2 – Характеристика датасету

Elbow метод наведено на рисунку 3.3.

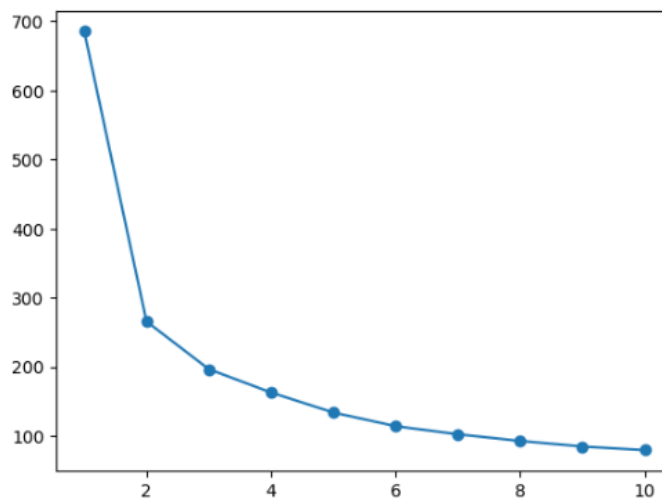


Рисунок 3.3 – Elbow метод

Виходячи з аналізу, можна зробити висновок, що наоптимальнішою кількістю класів – це 2 або 3.

Результат кластеризації за критеріями LCP та SI наведено на рисунку 3.4.

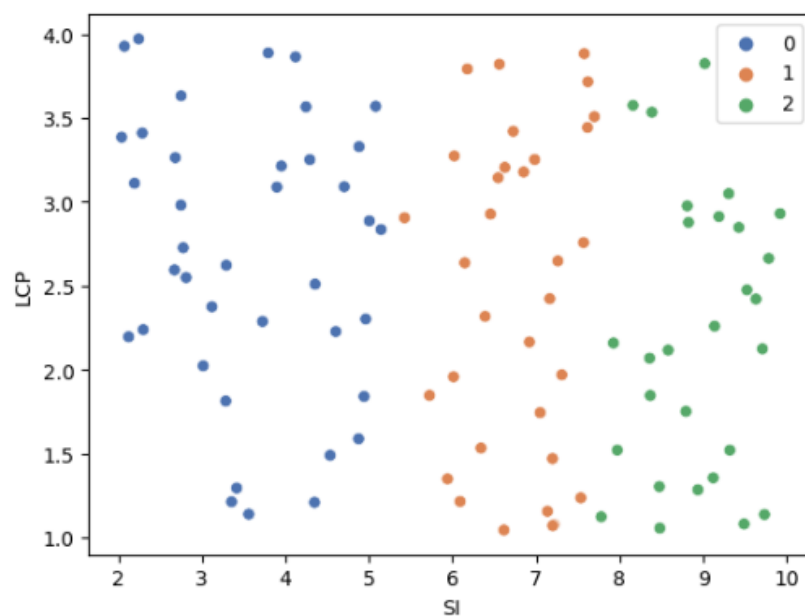


Рисунок 3.4 – Кластеризація за критеріями (LCP та SI) методом K-means

Результат кластеризації за критеріями LCP та SI методом Agglomerative Clustering наведено на рисунку 3.5.

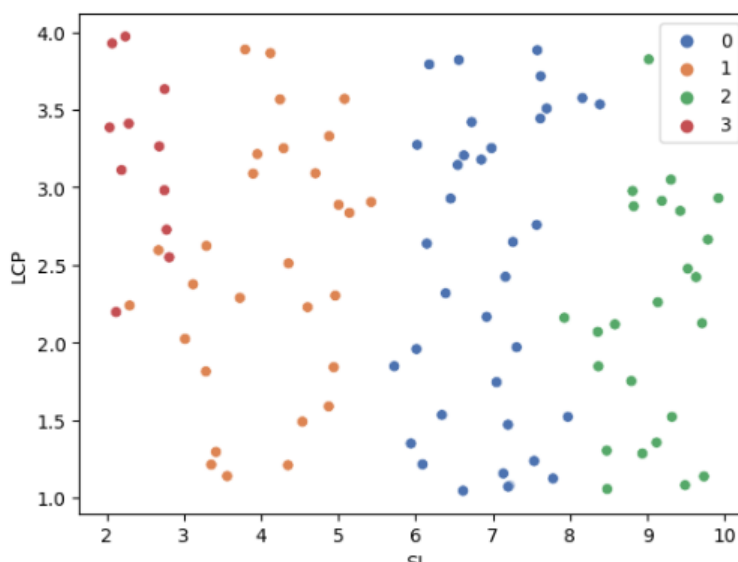


Рисунок 3.5 – Результат кластеризації за критеріями LCP та SI методом Agglomerative Clustering

Результат кластеризації за критеріями LCP та SI методом Birch Clustering наведено на рисунку 3.6.

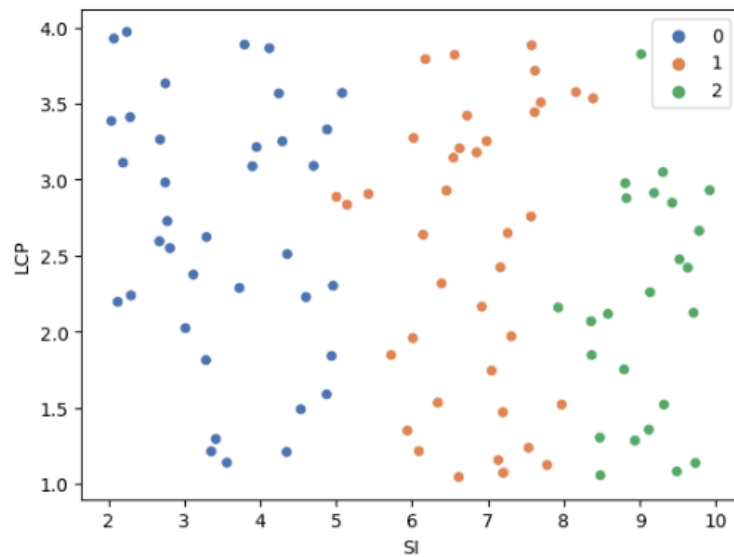


Рисунок 3.6 – Результат кластеризації за критеріями LCP та SI методом Birch Clustering

Результат кластеризації за критеріями LCP та SI методом Mean Shift наведено на рисунку 3.6.

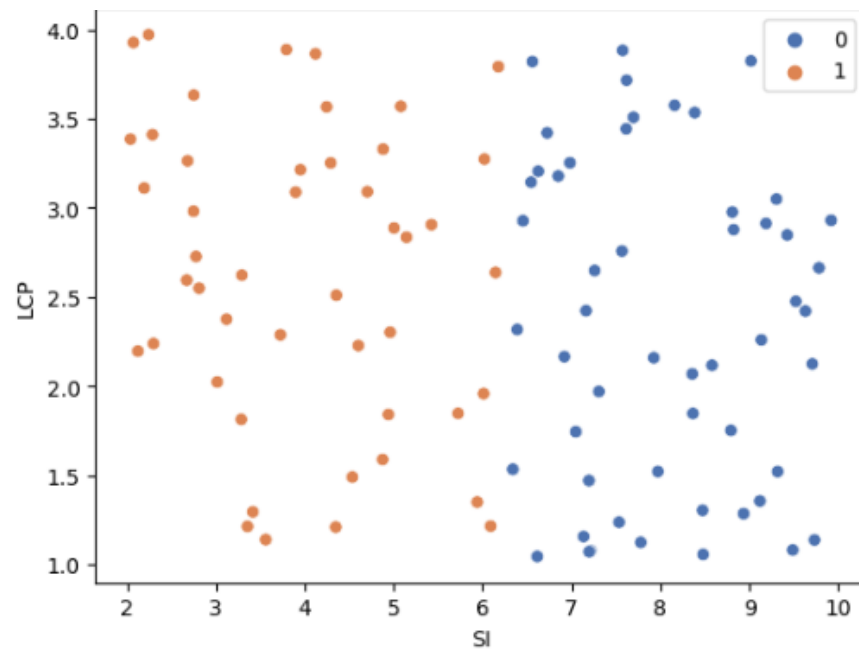


Рисунок 3.6 – Результат кластеризації за критеріями LCP та SI методом Mean Shift

Результат кластеризації за критеріями LCP та SI методом Affinity Propagation наведено на рисунку 3.7.

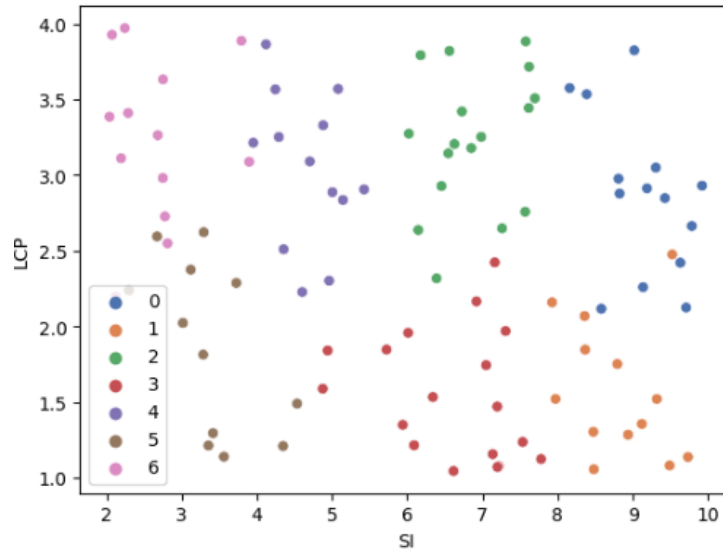


Рисунок 3.7 – Результат кластеризації за критеріями LCP та SI методом Affinity Propagation

Узагальнені результати опрацювання даних методом кластеризації наведено на рисунку 3.8.

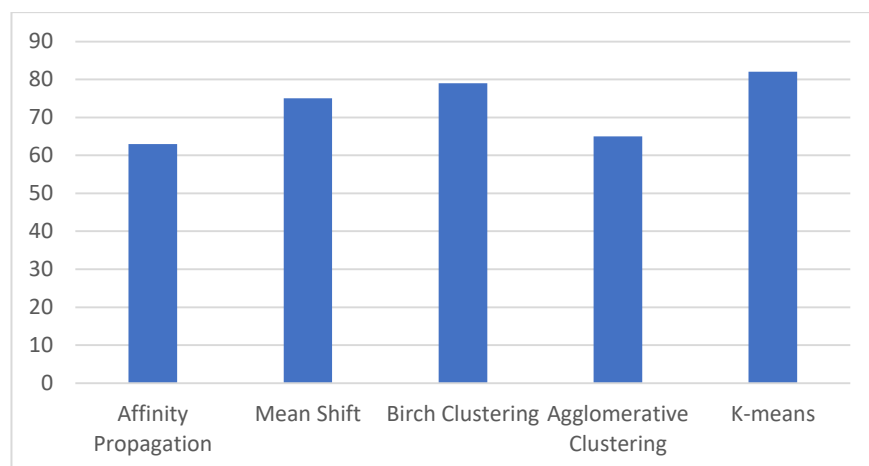


Рисунок 3.8 – Узагальнені результати опрацювання даних методом кластеризації

Виходячи із отриманих результатів можна зробити висновок, що алгоритм K-means для даних показав найкращий результат.

3.5 Висновки до розділу 3

Розроблено графічний інтерфейс розробленого програмного модулю для можливості задання параметрів у веб-режимі та для подальшого аналізу засобами штучного інтелекту, що дозволяє забезпечити зручний режим роботи з системою.

Розроблено датасет для кластеризації даних за критеріями FBT, SI, LCP та іншими для подальшого використання в задачах кластеризації, класифікації даних. Додатково розроблено REST API для можливості розширення функціоналу та взаємодії із третьосторонніми засобами.

Проведено порівняльний аналіз отриманих результатів шляхом оцінювання з допомогою Silhouette коефіцієнту. Аналіз дозволив встановити, що метод K-means показав найкращі результати для даного набору даних.

ВИСНОВКИ

1. На основі аналітичного підходу проведено аналіз засобів ефективності веб-сайтів, що дозволило виділити основні критерії та їх актуальність для визначення ефективності сайту.
2. На основі аналітичного підходу проведено аналіз сучасних алгоритмів машинного навчання, що дозволило виділити алгоритми кластеризації, які найкраще підходять для аналізу заданого формату даних.
3. На основі алгоритмічного підходу, розроблено узагальнений алгоритм оцінки ефективності веб-сайтів, що вміщує основні етапи опрацювання показників продуктивності сайту та етапи опрацювання даних засобами нечіткої логіки та кластеризації.
4. На основі алгоритмічного підходу, розроблено алгоритм на основі кластеризації для оцінки продуктивності сайту на основі вхідних показників із сервісів GTMetrix, WebPageTest, Google Pagespeed, що дозволило класифікувати сайти по критерію продуктивності.
5. Розроблено графічний інтерфейс розробленого програмного модулю для можливості задання параметрів у веб-режимі та для подальшого аналізу засобами штучного інтелекту, що дозволяє забезпечити зручний режим роботи з системою.
6. Проведено порівняльний аналіз отриманих результатів шляхом оцінювання з допомогою Silhouette коефіцієнту. Аналіз дозволив встановити, що метод K-means показав найкращі результати для даного набору даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The Evolution and Importance of Websites in the Digital Age URL: <https://www.globaltechaward.com/techbytes/detail/the-evolution-and-importance-of-websites-in-the-digital-age>
2. Badanie użyteczności stron. Dlaczego ocena strony internetowej ma znaczenie? URL: <https://www.eactive.pl/blog-o-seo/badanie-uzytecznosci-stron-dlaczego-ocena-strony-internetowej-ma-znaczenie/#:~:text=Ocena%20strony%20internetowej%20%E2%80%93%20kryteria,problemu%20porusza%C4%87%20si%C4%99%20po%20stronie.>
3. 10 Criteria to Evaluate a Website URL: <https://dougvos.com/10-criteria-evaluate-website/>
4. Юзабіліті сайту URL: <https://webtune.com.ua/statti/web-rozrobka/yuzabiliti-sajtu/>
5. Заяць І. 3. Аналіз когнітивного навантаження користувачів у їх та методи його зниження / Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі». 5 листопада 2024 р. Тернопіль. Україна-с. 115
6. Швидкість завантаження сайту URL: <https://wezom.com.ua/ua/blog/kak-uvlichit-skorost-zagruzki-sajta>
7. Performance Testing using GTMetrix URL: <https://avada.com/documentation/performance-testing-using-gtmatrix/>
8. GTmetrix: How To Use The Speed Test Tool URL: <https://www.searchenginejournal.com/gtmatrix-how-to-use-the-speed-test-tool/478269/>
9. How To Use WebPageTest and its API URL: <https://css-tricks.com/use-webpagetest-api/>
10. Website Performance testing with WebPageTest.org URL: <https://www.peakhour.io/blog/testing-website-speed-webpagetest/>

11. A Complete Guide to Using WebPageTest (and Interpreting the Results)
URL: <https://kinsta.com/blog/webpagetest/>
12. Що таке Machine Learning? URL: <https://denovo.ua/resources/what-is-machine-learning>
13. Machine Learning, ML URL: <https://www.it.ua/knowledge-base/technology-innovation/machine-learning>
14. Supervised and Unsupervised learning [Електронний ресурс]. – URL: <https://www.geeksforgeeks.org/supervised-unsupervised-learning/>
15. Кластерний аналіз URL: https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D0%B7
16. Cluster analysis URL: https://en.wikipedia.org/wiki/Cluster_analysis#Density-based_clustering
17. Hierarchical clustering URL: https://en.wikipedia.org/wiki/Hierarchical_clustering
18. Заяць І. З. Використання алгоритму k-means++ для вдосконаленого вибору початкових центрів кластерів / Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Інтелектуальні комп'ютерні системи та мережі». 5 листопада 2024 р. Тернопіль. Україна-с. 121
19. Google PageSpeed Insights: What It Is & How to Boost Your Score URL: <https://www.semrush.com/blog/google-pagespeed-insights/>
20. Швидкість завантаження сайту URL: <https://wezom.com.ua/ua/blog/kak-uvlichit-skorost-zagruzki-sajta>
21. Як перевірити і збільшити швидкість завантаження сайту URL: <https://ideadigital.agency/blog/yak-pereviriti-i-zbilshiti-shvidkist-zavantazhennya-sajtu/>
22. The Difference Between GTmetrix, PageSpeed Insights, Pingdom Tools and WebPagetest URL: <https://gtmetrix.com/blog/the-difference-between-gtmetrix-pagespeed-insights-pingdom-tools-and-webpagetest/>

