

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____
(повна назва)

Кафедра _____ програмної інженерії _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

Дослідження методів машинного навчання для підвищення ефективності SEO-
оптимізації веб-сайтів
(тема)

Виконав:
здобувач _____ 2 _____ року навчання
групи _____ ПЗМ-23-2 _____

_____ Андрій МАРТИНЕНКО _____
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність _____ 121 – Інженерія програмного
забезпечення _____
(код і повна назва спеціальності)

Тип програми _____ освітньо-наукова _____

Керівник _____ доц. Роксана МЕЛЬНІКОВА _____
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри

_____ Кирило СМЕЛЯКОВ _____
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____
 Кафедра _____ програмної інженерії _____
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 121 – Інженерія програмного забезпечення _____
 Тип програми _____ освітньо-наукова програма _____
 Освітня програма _____ Інженерія програмного забезпечення _____
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Мартиненку Андрію Олексійовичу _____
 (прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів машинного навчання для підвищення ефективності SEO-оптимізації веб-сайтів»

Затверджена наказом по університету від 15.04. 2025р. № 290 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 18.06.2025

3. Вихідні дані до роботи опис досліджуваних SEO-інструментів, що використовують машинне навчання для сканування та аналізу веб-сайтів, проведення експериментів, спрямованих на їх порівняння, розробка програми з інтеграцією методів машинного навчання для автоматизації та пришвидшення виконання рутинних процесів SEO-спеціаліста, опис застосування на практиці.

4. Перелік питань, що потрібно опрацювати в роботі

вступ, аналіз предметної галузі, огляд й аналіз літературних, наукових джерел, постановка задачі, теоретичне дослідження, проведення експерименту, висновки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	16.04.2025	<i>виконано</i>
2	Аналіз предметної галузі і постановка задачі	16.04.2025 – 20.04.2025	<i>виконано</i>
3	Огляд й аналіз літературних, наукових джерел	20.04.2025 – 22.04.2025	<i>виконано</i>
4	Теоретичне дослідження	24.04.2025 – 30.04.2025	<i>виконано</i>
5	Підготовка до апробації результатів дослідження. Публікація матеріалів	30.05.2025 – 10.05.2025	<i>виконано</i>
6	Проведення експерименту	10.05.2025 – 26.05.2025	<i>виконано</i>
7	Підготовка пояснювальної записки	26.05.2025 – 29.05.2025	<i>виконано</i>
8	Підготовка презентації та доповіді	29.05.2025 – 31.05.2025	<i>виконано</i>
9	Перевірка на плагіат	12.06.2025	<i>виконано</i>
10	Нормоконтроль	13.06.2025	<i>виконано</i>
11	Рецензування	14.06.2025	<i>виконано</i>
12	Попередній захист	16.06.2025	<i>виконано</i>
13	Занесення диплома в електронний архів	16.06.2025	<i>виконано</i>
14	Допуск до захисту у зав. кафедри	16.06.2025	<i>виконано</i>

Дата видачі завдання 16 квітня 2025р.

Студент _____
(підпис)

Андрій МАРТИНЕНКО

Керівник роботи _____
(підпис)

доц. Роксана МЕЛЬНІКОВА
(посада, Власне ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ / ABSTRACT

Робота містить: 76 с., 10 рис., 4 табл., 23 джерела.

АНАЛІЗ, ОПТИМІЗАЦІЯ, СКАНУВАННЯ, AHREFS, GOOGLE, MAJESTIC, MOZ, SCREAMING FROG, SEO, SERANKING, SERPSTAT.

Об'єктом дослідження є методи сканування веб-сайтів пошуковими системами та методи машинного навчання для кластеризації текстових даних у задачі пошукової оптимізації веб-сайтів.

Метою роботи є дослідження методів сканування веб-сайтів пошуковими системами та методів машинного навчання для кластеризації контенту, щоб інтегрувати у програмний продукт та автоматизувати рутинні процеси SEO-оптимізації.

Методами аналізу є вивчення проблемної області SEO-оптимізації, вибір інструментів для дослідження шляхом вирішення багатокритеріальної задачі прийняття рішень, порівняння їх продуктивності на основі встановлених критеріїв та дослідження методів машинного навчання, що можуть автоматизувати та пришвидшити рутинні процеси при аналізі даних перед початком ведення внутрішньої SEO-оптимізації веб-сайтів.

У результаті кваліфікаційної роботи було здійснено детальний аналіз обраних інструментів для SEO-оптимізації, зокрема розглянуто їх функціональні можливості, переваги та обмеження. Було досліджено теоретичні аспекти їх застосування для сканування та аналізу веб-сайтів, визначено ключові критерії ефективності та розроблене програмне забезпечення з використанням методів кластеризації для автоматизації процесу створення семантичного ядра та описано практичне застосування отриманих результатів.

ANALYSIS, OPTIMIZATION, CRAWLING, AHREFS, GOOGLE, MAJESTIC, MOZ, SCREAMING FROG, SEO, SERANKING, SERPSTAT.

The object of research is the methods of website crawling by search engines and machine learning methods for clustering text data in the task of search engine optimization of websites.

The purpose of the study is to investigate the methods of website crawling by search engines and machine learning methods for integration into a software product to automate routine SEO optimization processes.

The methods of analysis include exploring the problem domain of SEO optimization, selecting tools for the study by solving a multi-criteria decision-making task, comparing their performance based on established criteria, and researching machine learning methods that can automate and accelerate routine data analysis processes before initiating on-site SEO optimization.

As a result of the qualification work, a detailed analysis of the selected SEO optimization tools was carried out, including their functionality, advantages, and limitations. The theoretical aspects of their use for website crawling and analysis were investigated, key performance criteria were identified, and software was developed using clustering methods to automate the process of creating a semantic kernel, and the practical application of the results was described.

Умови публікації звіту: заява щодо самостійного виконання кваліфікаційної роботи та можливості її публікації в електронному архіві відкритого доступу EIAr KhNURE.

Завідувачу кафедри

П

(скорочена назва кафедри)

проф. Кирилу СМЕЛЯКОВУ

(вчене звання, сласне ім'я, прізвище)

ЗАЯВА

щодо самостійності виконання кваліфікаційної роботи та можливості її публікації (та/або публікації анотації кваліфікаційної роботи) в електронному архіві відкритого доступу EIAr KhNURE

Я, Мартиненко Андрій Олексійович

(прізвище, ім'я, по батькові)

здобувач вищої освіти на другому (магістерському) рівні вищої освіти академічної групи ІПЗМ-23-2

кафедра програмної інженерії,
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему

«Дослідження методів машинного навчання для підвищення ефективності SEO-оптимізації веб-сайтів»

(назва роботи)

що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в репозиторії "EIArKhNURE". Погоджуюся з авторським договором, відповідно до Положення про репозиторій ХНУРЕ "EIArKhNURE". Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з вимогами академічної доброчесності, згідно з якими виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

Дата 09.06.2025

Підпис

ЗМІСТ

Перелік скорочень	9
Вступ.....	10
1 Аналіз предметної галузі	12
1.1 Актуальність теми дослідження	13
1.2 Масштаб проблеми та значущість веб-сканування	14
1.3 Залежність бізнесу від пошукової видимості	14
1.4 Важливість адаптації до алгоритмів Google	15
1.5 Багатокритеріальний вибір SEO-інструментів	16
1.6 Опис множини альтернатив для задачі вибору	16
1.7 Опис множини критеріїв та шкал оцінок за ними	17
1.7.1 Ефективність сканування	18
1.7.2 Інтеграція машинного навчання	18
1.7.3 Швидкість аналізу	19
1.7.4 Зручність інтерфейсу.....	19
1.7.5 Вартість	19
1.8 Опис моделі згортання.....	20
1.9 Визначення вагових коефіцієнтів.....	20
1.10 Розрахунок корисності альтернатив.....	21
2 Огляд й аналіз літературних, наукових джерел.....	24
2.1 Google Search Central.....	24
2.2 Ahrefs	25
2.3 Serpstat	26
2.4 Screaming Frog	26
3 Постановка задачі.....	28
4 Теоретичне дослідження.....	29

4.1 Дослідження методів сканування та індексації сайтів	29
4.2 Методи машинного навчання	32
4.2.1 Кластеризація контенту.....	32
4.2.3 Інтеграція методів кластеризації в програмний продукт	35
4.3 Архітектура та проектування програмного забезпечення.....	36
4.3.1 Архітектурний підхід	36
4.3.2 Use-Case-діаграма.....	38
4.3.4 UML-Діаграма класів	39
5 Проведення експерименту	41
5.1 Знімки екрану та приклади коду	41
5.2 Перевірка результату кластеризації в пошуковій видачі	46
5.3 Порівняння розробленої програми з існуючими SEO-інструментами.....	48
5.4 Практичне застосування	50
Висновки.....	52
Перелік джерел посилання	53
Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії.....	56
Додаток А Апробація результатів роботи. Тези 1	57
Додаток Б Апробація результатів роботи. Тези 2	61
Додаток В Слайди презентації	64
Додаток Г Звіт з результатами перевірки на унікальність тексту в базі ХНУРЕ	74
Додаток І Експертний висновок нормоконтроль	76

ПЕРЕЛІК СКОРОЧЕНЬ

SEO – Search Engine Optimization

ML – Machine Learning

NLP – Natural Language Processing

CMS – Content Management System

CSR – Client -Side Rendering

SSR – Server -Side Rendering

CTR – Click-Through Rate

HTML – HyperText Markup Language

GUI – Graphic User Interface

CSV – Comma-Comma-Separated Values

PCA – Principal Component Analysis

ВСТУП

Світ сучасних технологій непинно розвивається, і важливе місце в цьому розвитку займають інструменти для сканування та аналізу веб-сайтів. У сфері SEO (Search Engine Optimization) такі інструменти, як Ahrefs, Serpstat, Seranking, Majestic, MOZ, Screaming Frog, а також методи сканування, які використовує Google, є невід'ємною частиною процесу оптимізації. Вони дозволяють виявляти технічні та контентні помилки, аналізувати структуру веб-сайту, а також надавати рекомендації щодо їхнього усунення. Це дозволяє підвищувати якість сайтів, їхню швидкість завантаження та зручність використання, що є ключовими аспектами конкурентоспроможності у цифровому середовищі.

Сучасні інструменти для сканування веб-сайтів дедалі частіше використовують методи машинного навчання, що дозволяє значно покращити якість аналізу, виявляти приховані закономірності у даних та робити прогнози на основі отриманих результатів. Наприклад, машинне навчання допомагає автоматизувати обробку великих обсягів даних, підвищуючи швидкість і точність аналізу. Це відкриває нові можливості для автоматизації, персоналізації підходів до оптимізації та підвищення ефективності SEO-процесів.

Дана робота присвячена дослідженню методів машинного навчання для підвищення ефективності SEO-оптимізації веб-сайтів, а саме аналізу методів сканування, які використовує Google-бот, розгляду методів кластеризації для їх впровадження в програмне забезпечення для автоматизації рутинних процесів з обробки великих об'ємів даних. Кожен із методів має свої особливості та застосування, що робить їх незамінними для кластеризації різних обсягів даних.

Також проведено порівняння популярних SEO-інструментів, шляхом вирішення задачі багатокритеріального вибору. Галузь застосування цих інструментів є широкою, охоплюючи аудит сайтів, що проводиться для створення стратегії підвищення видимості в пошукових системах, оптимізацію контенту та структури, а також аналіз зворотних посилань. Вивчення особливостей роботи цих інструментів дозволить розробити рекомендації для їх покращення з впровадженням методів машинного навчання. Крім того, інтеграція методів машинного навчання в

роботу SEO-інструментів відкриває перспективи для подальшого вдосконалення процесів оптимізації та створення більш інтелектуальних рішень.

Актуальність дослідження полягає в постійному вдосконаленні алгоритмів пошукових систем та необхідності адаптації інструментів SEO до цих змін. Застосування машинного навчання у SEO створює нові можливості для аналізу даних та розробки ефективних стратегій оптимізації. В умовах зростаючої конкуренції на ринку цифрових послуг це є критично важливим для забезпечення успіху та конкурентоспроможності бізнесу. Дослідження також сприятиме кращому розумінню сучасних тенденцій у скануванні та аналізі веб-сайтів, таких як автоматизація процесів, впровадження інноваційних методів та інтелектуалізація систем аналізу.

Робота також включає експериментальне дослідження, у ході якого будуть оцінені середні показники продуктивності кожного з інструментів за визначеними критеріями, включаючи швидкість аналізу, зручність використання та вартість.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

Спостерігаючи інтенсивну діджиталізацію усіх аспектів людської діяльності, можна прийти до висновку, що кожен бізнес обов'язково повинен мати власний веб-сайт для підвищення впізнаваності бренду та залучення вагомому потоку клієнтів. Метою дослідження є аналіз існуючих та проектування ефективного рішення для аналізу проіндексованих веб-сайтів пошуковою системою Google, яке базується на головних та вагомих факторах ранжування [1], враховуючи індивідуальні особливості різних ніш веб-сайтів (медичної, юридичної, електронної комерції, тощо) із залученням машинного навчання. Тому, дане дослідження є затребуваним та актуальним для будь-якого виду бізнесу, оскільки його результатами є покращення ранжування веб-сайтів з метою їх якісного та стрімкого просування до ТОП-3 у пошукової видачі Google.

Останнім часом зростає кількість досліджень, які зосереджені на аналізі SEO-інструментів і їхній здатності сканувати та оцінювати веб-сайти. У сучасній науковій літературі особлива увага приділяється ролі методів машинного навчання, що впроваджуються у ці інструменти. Вони забезпечують автоматизацію аналізу великих обсягів даних, дозволяють ідентифікувати приховані закономірності та підвищують точність оцінок, що сприяє більш глибокому розумінню стану веб-сайту і прийняттю обґрунтованих рішень для його оптимізації.

Одним із найпоширеніших напрямків є вивчення ефективності таких інструментів, як Ahrefs, Serpstat, Screaming Frog та Majestic. Вони спеціалізуються на аналізі різних аспектів SEO, включаючи аудит технічного стану сайту, аналіз зворотних посилань і виявлення контентних проблем. Наприклад, Ahrefs широко використовується для вивчення профілю посилань, що є критично важливим для визначення авторитетності веб-сайту.

У свою чергу, Google-бот є еталоном у сфері сканування сайтів, оскільки його алгоритми безпосередньо впливають на ранжування сторінок у пошуковій видачі. Однак у наукових дослідженнях часто звертається увага на те, що Google-бот не надає інструментів для детального аналізу, залишаючи цей простір для спеціалізованих SEO-інструментів.

Ключовою тенденцією є зростання використання машинного навчання в SEO. Методи ML використовуються для прогнозування змін у ранжуванні, аналізу поведінкових факторів користувачів і автоматизації процесів оптимізації. Це дозволяє створювати більш персоналізовані підходи до SEO, які враховують унікальні потреби кожного сайту.

Однак із широкими можливостями використання SEO-інструментів також виникають виклики. Одним із них є забезпечення точності даних та прозорості роботи алгоритмів. Важливо, щоб користувачі могли розуміти, як саме інструмент отримує ті чи інші результати, що підвищує довіру до цих технологій. Також у літературі наголошується на важливості врахування етичних аспектів використання даних для аналізу, що включає конфіденційність інформації та захист персональних даних.

Загалом, розвиток SEO-інструментів та інтеграція машинного навчання створюють нові можливості для аналізу та оптимізації веб-сайтів. Це не лише спрощує роботу SEO-спеціалістів, але й дозволяє досягати більш високих результатів завдяки точності, швидкості та автоматизації процесів.

1.1 Актуальність теми дослідження

SEO-інструменти для сканування та аналізу веб-сайтів, такі як Ahrefs, Serpstat, Seranking, Majestic, MOZ та Screaming Frog, займають ключове місце в процесах оптимізації веб-ресурсів для пошукових систем. Ці інструменти забезпечують розробників і маркетологів засобами для виявлення технічних проблем, аналізу конкурентів, підвищення видимості сайтів та покращення їхньої роботи.

Актуальність дослідження пов'язана з тим, що SEO-інструменти постійно вдосконалюються, інтегруючи інноваційні технології, зокрема машинне навчання. Це дозволяє автоматизувати аналіз, підвищити точність прогнозів та ефективність рекомендацій. Однак розробники стикаються з викликами, пов'язаними з частими змінами алгоритмів Google, який задає правила оцінки сайтів.

Дослідження порівняння підходів SEO-інструментів із методами роботи Google-бота є важливим для оцінки їхньої відповідності сучасним вимогам,

визначення поточних обмежень і пошуку шляхів їх подолання. Це дозволить розробникам і власникам веб-сайтів адаптуватися до швидкозмінних умов цифрового середовища.

1.2 Масштаб проблеми та значущість веб-сканування

Веб-сканування є фундаментальним процесом для технічного SEO, що забезпечує повний аналіз веб-сайту на предмет технічних помилок, структурних недоліків і оптимізації контенту. У сучасних умовах зростання конкуренції на ринку цифрового маркетингу, ефективне сканування сайтів стає необхідним для досягнення високих позицій у пошуковій видачі.

Із розвитком технологій розробки веб-сайтів, зокрема використанням фреймворків, таких як React, Vue, Angular, та поширенням динамічного контенту, структури сайтів стали значно складнішими. Це створює додаткові виклики для їх сканування, адже інструменти повинні враховувати специфіку JavaScript, SPA (односторінкових додатків), адаптивних версій сайтів, а також технічні помилки, як відсутність файлів robots.txt, дублювання сторінок, неправильно налаштовані редиректи або повільна швидкість завантаження, значно знижують видимість сайтів у пошукових системах.

1.3 Залежність бізнесу від пошукової видимості

У сучасному цифровому середовищі пошукова видимість відіграє вирішальну роль у залученні потенційних клієнтів та забезпеченні успіху бізнесу. Високі позиції у пошуковій видачі безпосередньо впливають на кількість відвідувачів сайту, що, у свою чергу, відображається на конверсіях та доходах компанії. Згідно дослідження про вплив позиції в пошуковій видачі на CTR (відношення кількості переходів на сайт до кількості показів у результатах пошуку) було визначено ресурсом Web-promo [2] :

- перша позиція отримує приблизно 28,5% кліків;
- друга позиція близько 15,7% кліків;
- третя позиція приблизно 11% кліків.

Ці показники свідчать про те, що перші три позиції в пошуковій видачі Google отримують понад 50% всіх кліків, що підкреслює важливість високого ранжування для бізнесу.

1.4 Важливість адаптації до алгоритмів Google

Алгоритми Google регулярно оновлюються, щоб забезпечити користувачам найрелевантніші й якісні результати пошуку, що робить їх врахування критично важливим для бізнесу. Наприклад, алгоритм RankBrain [3] використовує машинне навчання для розуміння складних пошукових запитів і надання найкращих відповідей, тоді як BERT допомагає Google краще розуміти контекст і значення слів у запитах. Core Web Vitals зосереджені на трьох ключових аспектах досвіду користувачів:

- швидкість завантаження (Largest Contentful Paint);
- інтерактивність (First Input Delay);
- візуальна стабільність (Cumulative Layout Shift).

Кожне оновлення алгоритмів спрямоване на покращення пошукової видачі, і їх ігнорування може призвести до значних втрат. Наприклад, після оновлення Panda (2011) [3], яке карало сайти з низькоякісним контентом, і Penguin (2012) [3], що знижував позиції сайтів із неякісними зворотними посиланнями, багато компаній втратили позиції та трафік.

Щоб залишатися конкурентоспроможними, бізнесам необхідно адаптуватися до змін алгоритмів. Це включає постійний моніторинг оновлень через офіційні джерела, технічну оптимізацію сайтів, впровадження, структурованих даних для поліпшення видимості в пошуковій видачі, та створення якісного, релевантного контенту.

Особливу увагу слід приділяти алгоритмам, що враховують авторитетність сайту, як-от E-E-A-T (Experience, Expertise, Authoritativeness, Trustworthiness) [4], де важливо публікувати експертний контент і працювати над побудовою якісних зворотних посилань.

Адаптація до алгоритмів Google не лише забезпечує стабільний органічний трафік, а й допомагає зберегти конкурентну перевагу в умовах динамічного цифрового середовища.

1.5 Багатокритеріальний вибір SEO-інструментів

Необхідно провести вибір найкращих інструментів, що дозволяють сканувати та аналізувати веб-сайти, схожим принципом, як це робить Google, які формують звіти, базуючись на яких будується стратегія SEO-оптимізації.

Для вирішення задачі багатокритеріального вибору [5] обрано 6 варіантів та наступні критичні фактори: ефективність сканування, інтеграція машинного навчання, швидкість аналізу, зручність інтерфейсу та вартість інструментів.

Для цього потрібно оцінити кожен інструмент за зазначеними критеріями та здійснити порівняння з урахуванням ваги кожного критерію.

1.6 Опис множини альтернатив для задачі вибору

У задачі багатокритеріального вибору для вибору найкращого інструмента сканування та аналізу даних представлено 6 альтернатив, кожна з яких має свої особливості та характеристики:

- Ahrefs. Один із найбільш популярних інструментів для SEO-аналізу, відомий своєю потужною базою даних і можливістю ефективно відслідковувати зворотні посилання, аналізувати конкуренцію, а також надавати детальні звіти про SEO-стратегії. Може бути інтегрований з різними інструментами машинного навчання для глибшого аналізу;
- Serpstat. Всеосяжний інструмент для аналізу SEO, який охоплює численні функції для моніторингу позицій в результатах пошукової видачі, дослідження ключових слів та аналізу зворотних посилань. Має хорошу інтеграцію з машинним навчанням для підвищення точності результатів;
- Seranking. Інструмент, орієнтований на більш вузькі завдання SEO-аналізу, зокрема зворотних посилань та детального дослідження ефективності

контенту. Має високу швидкість аналізу, але з менш інтуїтивно зрозумілим інтерфейсом;

- Majestic. Спеціалізується на зворотних посиланнях та посиланнях на сайти, має потужний інструмент для дослідження посилальної маси, але менш універсальний в інших аспектах SEO-аналізу;
- MOZ. Відомий своїм інтуїтивно зрозумілим інтерфейсом та чудовими результатами в SEO-аналізі, MOZ забезпечує різноманітні функції для аналізу ключових слів, зворотних посилань та загальної стратегії SEO. Також має інтеграцію з машинним навчанням для покращення результатів;
- Screaming Frog. Забезпечує глибокий аналіз сайтів через сканування структур даних. Особливо корисний для великих сайтів, маючи високу ефективність сканування та інтеграцію з іншими платформами машинного навчання.

Ці інструменти є провідними для SEO-спеціалістів, тому будуть розглядатися в даному дослідженні.

1.7 Опис множини критеріїв та шкал оцінок за ними

Спочатку визначимо головні критерії оцінки інструментів, які сканують та аналізують показники веб-сайтів, спираючись на потреби SEO-спеціалістів:

- ефективність сканування;
- інтеграція машинного навчання;
- швидкість аналізу;
- зручність інтерфейсу;
- вартість.

Перелічені критерії розташовані в порядку спадання за їх пріоритетом.

Джерела оцінок:

- дані зі офіційних веб-сайтів інструментів;
- досвід та відгуки SEO-спеціалістів.

Ефективність сканування – це головний критерій, бо від нього залежить наскільки успішно буде проаналізований сайт, від чого залежить повнота та зміст отриманих звітів. Далі наведено більш детальний опис кожного з обраних критеріїв.

1.7.1 Ефективність сканування

Даний критерій оцінює здатність кожного інструмента виявляти технічні та контентні помилки на обраному веб-сайті для отримання звіту, ґрунтуючись на якому формується стратегія оптимізації та просування SEO-спеціалістом.

Шкала – порядкова шкала, від 0 до 5, де 0 – мінімальна кількість знайдених помилок, 5 – максимальне покриття можливих проблем.

Спосіб нормування – з урахуванням мінімуму та максимуму. Розраховується за формулою 1.1:

$$f = \frac{f_{\text{вимір}} - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}} = \frac{f_{\text{вимір}}}{5} \quad (1.1)$$

де $f_{\text{вимір}}$ – значення, що вимірюється,

f_{min} – мінімальне значення, що дорівнює 0,

f_{max} – максимальне значення, що дорівнює 5.

1.7.2 Інтеграція машинного навчання

Цей критерій оцінює підтримку методів машинного навчання для прогнозування та аналізу. Інструменти з високою інтеграцією машинного навчання також здатні виявляти патерни у даних, надавати інтелектуальні рекомендації та автоматично оптимізувати стратегії SEO.

Шкала – порядкова шкала, від 0 до 5, де 0 – відсутність інтеграції машинного навчання, 5 – повна підтримка функцій машинного навчання.

Спосіб нормування – з урахуванням мінімуму та максимуму. Розраховується за формулою 1.2:

$$f = \frac{f_{\text{вимір}} - f_{\text{min}}}{f_{\text{max}} - f_{\text{min}}} = \frac{f_{\text{вимір}}}{5} \quad (1.2)$$

де $f_{\text{вимір}}$ – значення, що вимірюється,

f_{min} – мінімальне значення, що дорівнює 0,

f_{max} – максимальне значення, що дорівнює 5.

1.7.3 Швидкість аналізу

Вимірюється середній час сканування та обробки веб-сайту (у хвиликах). Швидкість аналізу є важливим фактором, оскільки від цього залежить час отримання звіту, зважаючи на кількість сторінок та технології з використанням якої розроблено веб-сайт.

Шкала – метрична шкала, час в хвиликах.

Спосіб нормування – з урахуванням мінімуму та максимуму, де мінімум – найшвидший час формування звіту, максимум – найдовший час. Тому, чим більший час формування звіту – тим менша нормована оцінка.

1.7.4 Зручність інтерфейсу

Визначається середньою оцінкою користувачів на платформах. Цей критерій включає інтуїтивність інтерфейсу, легкість у налаштуванні та доступність наявної документації.

Шкала – порядкова шкала, від 0 до 5, де 0 – найменший рівень зручності, 5 – найвищий.

Спосіб нормування – з урахуванням мінімуму та максимуму, де мінімум – найменша оцінка користувачів, максимум найбільша оцінка.

1.7.5 Вартість

Оцінюється ціна інструмента за місяць використання. Низька вартість робить інструмент більш доступним для широкого кола користувачів, особливо для малих та середніх бізнесів.

Шкала – метрична шкала, ціна тарифу в USD на місяць.

Спосіб нормування – з урахуванням мінімуму та максимуму, чим більша вартість – тим менша нормована оцінка.

1.8 Опис моделі згортання

Для обчислення коефіцієнта корисності обраних сервісів буде використано лінійне адаптивне згортання з ваговими коефіцієнтами. Розраховується за формулою 1.3:

$$Z = \sum_{i=1}^n \alpha_i \beta_i a_i \quad (1.3)$$

де α_i – нормуючі множники,

β_i – вагові коефіцієнти,

a_i – значення i -го критерія,

Z – результат згортання.

1.9 Визначення вагових коефіцієнтів

Було обрано 5 критеріїв, які мають пріоритет від 1 до 5. Тепер визначимо вагові коефіцієнти для кожного з критеріїв, шляхом простого ранжування, враховуючи пріоритетність:

- ефективність сканування – 1;
- інтеграція машинного навчання – 2;
- швидкість аналізу – 3;
- зручність інтерфейсу – 4;
- вартість інструментів – 5.

Сума пріоритетів критеріїв дорівнює 15 та розраховується за формулою 1.4:

$$S_p = \sum_{i=1}^n p_i \quad (1.4)$$

де S_p - сума пріоритетів,

p_i – пріоритет i -го критерія.

Розрахуємо коефіцієнт для кожного критерія за формулою 1.5:

$$\beta_i = \frac{p_{max} + p_{min} - p_i}{S_p} \quad (1.5)$$

де β_i - ваговий коефіцієнт i -го критерія,

p_{max} – пріоритет найважливішого критерія (1),

p_{min} – пріоритет найменш важливого критерія (5).

Нижче наведено пріоритет та відповідний ваговий коефіцієнт для кожного з критеріїв (див. табл. 1.1).

Таблиця 1.1 – Вагові коефіцієнти (Таблиця виконана самостійно)

Критерій	Пріоритет	Ваговий коефіцієнт
Ефективність сканування	1	0,33
Інтеграція машинного навчання	2	0,26
Швидкість аналізу	3	0,20
Зручність інтерфейсу	4	0,13
Вартість інструментів	5	0,06

Таким чином, критерії розташовані в порядку спадання.

1.10 Розрахунок корисності альтернатив

Для розрахунку коефіцієнту корисності кожного з наведених інструментів, необхідно нормувати кожну оцінку.

Нижче наведені нормовані значення критеріїв для кожної альтернативи разом з результатом згортання – коефіцієнт корисності (див. табл. 1.2).

Таблиця 1.2 – Коефіцієнти корисності (Таблиця виконана самостійно)

Інструмент	Ефективність сканування	Інтеграція ML	Швидкість аналізу, хв	Зручність інтерфейсу	Вартість інструментів (USD/місяць)	Корисність
Ahrefs	0.9	0.8	0.7	0.9	0.4	0.80
Serpstat	0.8	0.7	0.6	0.8	0.5	0.71
Sernaking	0.6	0.6	0.9	0.7	0.7	0.68
Majestic	0.7	0.5	0.5	0.6	0.9	0.61
MOZ	0.8	0.6	0.6	0.8	0.4	0.67
Screaming Frog	1.0	0.9	0.8	0.9	0.3	0.87

Кожна альтернатива описана вектором:

- Ahrefs: [0.9, 0.8, 0.7, 0.9, 0.4]
- Serpstat: [0.8, 0.7, 0.6, 0.8, 0.5]
- Seranking: [0.6, 0.6, 0.9, 0.7, 0.7]
- Majestic: [0.7, 0.5, 0.5, 0.6, 0.9]
- MOZ: [0.8, 0.6, 0.6, 0.8, 0.4]
- Screaming Frog: [1.0, 0.9, 0.8, 0.9, 0.3]

Використовуючи лінійне адаптивне згортання було вираховано корисність для кожного з обраних сервісів, що дозволяють сканувати та обробляти веб-сайти для формування звіту з показниками ефективності та виявленими проблемами. В результаті найвищий коефіцієнт корисності мають інструменти:

- Screaming Frog;
- Ahrefs;
- Serpstat.

Тому саме ці інструменти обрані для подальшого дослідження.

Варто зазначити, що вирішення даної багатокритеріальної задачі стало підготовчим етапом для основної частини, яке зосереджене на вивченні ефективності методів SEO-оптимізації. Усі наведені інструменти мають унікальні характеристики, проте детальне дослідження кожного з них є малоефективним. Шляхом застосування методу багатокритеріального вибору вдалося звужити коло об'єктів дослідження, орієнтуючись на ключовий критерій: ефективність сканування веб-сайтів та інтеграція машинного навчання. Це дозволило сфокусуватися на найперспективніших інструментах, таких як Screaming Frog, Ahrefs та Serpstat, які демонструють найкращі показники продуктивності та гнучкості. Такий підхід забезпечує більш цілеспрямоване і глибоке вивчення ефективності цих рішень у рамках подальших досліджень.

2 ОГЛЯД Й АНАЛІЗ ЛІТЕРАТУРНИХ, НАУКОВИХ ДЖЕРЕЛ

Зосередимося на детальному огляді документації SEO-інструментів, що були обрані на попередньому кроці дослідження, зокрема Google, Ahrefs, Serpstat та Screaming Frog, а також приділимо увагу зовнішнім дослідженням, які описують роль машинного навчання у SEO.

До обраних джерел відноситься офіційна документація досліджуваних інструментів та перевірена інформація із зовнішніх досліджень за останні 5 років.

2.1 Google Search Central

Основне та первинне джерело інформації про алгоритми сканування та індексації, включаючи використання Googlebot. В документації [6] зазначено, що Googlebot працює за принципами динамічного пріоритетного сканування, що дозволяє визначати найбільш релевантні сторінки для подальшого індексування.

Технології машинного навчання є основою роботи багатьох алгоритмів Google. Один із ключових компонентів — RankBrain, алгоритм, що застосовує методи обробки природної мови (NLP) для інтерпретації пошукових запитів. RankBrain використовує нейронні мережі для класифікації запитів і виявлення ключових патернів у поведінці користувачів, що допомагає більш точно відповідати на складні та багатозначні запити.

Інший важливий алгоритм, BERT (Bidirectional Encoder Representations from Transformers), дозволяє Google аналізувати контекст слів у запитах, враховуючи не лише значення окремих слів, а й їх зв'язок із сусідніми термінами. Це сприяє поліпшенню розуміння довгих запитів та пошукових фраз, зокрема тих, що включають заперечення чи складні синтаксичні конструкції.

Згідно з SEO.ai [7], Google використовує методику класифікації контенту на основі релевантності та якості. Для цього залучаються алгоритми, що аналізують унікальність, структурованість і відповідність інформації намірам користувача. Такий підхід дозволяє ранжувати контент у реальному часі, враховуючи змінні фактори, як-от трендовість теми чи регіональні відмінності.

Важливим елементом також є алгоритм PageRank, який працює у поєднанні з сучасними методами машинного навчання. PageRank оцінює цінність сторінок на основі кількості та якості зворотних посилань, а машинне навчання допомагає фільтрувати спам-посилання та оцінювати природність профілю посилань.

Також, посилаючись на Page Optimizer Pro [8], використання алгоритмів ML дозволяє Google динамічно адаптувати свої системи до нових шаблонів поведінки користувачів та змін у структурі контенту. Це забезпечує високу якість пошукової видачі та сприяє підвищенню задоволеності користувачів результатами.

2.2 Ahrefs

Однією з ключових функцій Ahrefs є AhrefsBot, який щоденно сканує понад 8 мільярдів сторінок, збираючи дані про зворотні посилання, ключові слова та їхній вплив на органічний пошук. Цей бот використовує алгоритми машинного навчання для класифікації даних, оцінки їхньої релевантності та виявлення нових SEO-трендів.

Інструмент пропонує унікальну функцію кластеризації ключових слів. Вона базується на ML-алгоритмах, які автоматично групують запити за тематичними категоріями, дозволяючи створювати структури для оптимізації контенту. Наприклад, запити про «SEO» та «аналітика» можуть бути згруповані для підготовки спільної маркетингової стратегії.

Ще одним важливим аспектом є використання прогнозових моделей для аналізу трафіку. Ahrefs здатний оцінювати потенційний приріст від SEO-стратегій, спираючись на історичні дані. У цьому контексті прогнозування базується на виявленні патернів у змінах трафіку та конкурентному аналізі.

Крім того, Ahrefs впроваджує автоматичне виявлення токсичних посилань за допомогою ML, що дозволяє уникати штрафів від пошукових систем. Це досягається шляхом аналізу профілю посилань, оцінки їхньої якості та визначення підозрілих активностей, таких як спам-лінки.

Згідно з даними Ahrefs [9], інструмент також інтегрує обробку природної мови (NLP) для аналізу контенту, оцінки його унікальності та відповідності намірам

користувача. Це забезпечує створення рекомендацій щодо оптимізації контенту, які базуються на реальних пошукових запитах і сучасних трендах.

2.3 Serpstat

Serpstat спеціалізується на аналізі великих обсягів даних, таких як частотність ключових слів та позиції у пошуковій видачі. Офіційна документація доступна на Serpstat Help Center [10]. Інструмент використовує алгоритми машинного навчання для: кластеризації ключових слів, аналізу конкурентів та визначення ефективності рекламних кампаній.

2.4 Screaming Frog

Інструмент технічного аналізу сайтів, який дозволяє виявляти критичні помилки на сайтах [11]. Офіційна документація доступна на Screaming Frog User Guide [12]. За документацією, Screaming Frog інтегрує машинне навчання для автоматичного визначення пріоритетності проблем та створення рекомендацій щодо їх вирішення.

Інструмент також підтримує інтеграцію з API Google Analytics, що дозволяє глибше аналізувати дані користувацької поведінки. За даними Go Fish Digital [13], Screaming Frog допомагає виявляти слабкі сторони у структурі сайту та автоматично генерує звіти для оптимізації.

На основі проведеного огляду можна зробити висновок, що Ahrefs значно перевершує більшість інших інструментів SEO, окрім Google, у впровадженні та використанні технологій машинного навчання. Його можливості прогнозування трафіку, кластеризації ключових слів, виявлення токсичних посилань і аналізу зворотних посилань роблять його ідеальним інструментом для SEO-фахівців, які прагнуть використовувати сучасні AI-технології для конкурентної переваги.

Google, однак, залишається незаперечним лідером у масштабі інтеграції машинного навчання, зокрема через алгоритми RankBrain, BERT і PageRank. У зв'язку з цим подальші дослідження будуть спрямовані на порівняння ефективності Ahrefs із Googlebot у контексті аналізу контенту, індексації сторінок і управління

SEO-стратегіями. Це дозволить краще зрозуміти переваги та обмеження обох підходів у реальних умовах застосування.

На основі проведеного огляду можна зробити висновок, що інструменти SEO активно інтегрують машинне навчання для оптимізації процесів аналізу та ранжування веб-сайтів. Основні технології включають кластеризацію даних, прогнозування та автоматизацію технічного аналізу.

Попри значний прогрес, дослідження все ще потребують вдосконалення, зокрема у вивченні взаємодії ML-алгоритмів з пошуковими запитами та специфічними потребами користувачів. Ці аспекти можуть стати перспективними напрямками для майбутніх досліджень.

3 ПОСТАНОВКА ЗАДАЧІ

Метою роботи є дослідження методів машинного навчання та їх інтеграція у програмний продукт для автоматизації процесів SEO-оптимізації. Особливу увагу приділено аналізу інструментів, які забезпечують високу точність та ефективність сканування, таких як Ahrefs та Googlebot, а також застосуванню методів кластеризації для підвищення якості аналізу даних. У даній магістерській роботі передбачено виконання наступних задач:

- оцінити функціональні можливості SEO-інструментів. Провести детальний аналіз функціональних можливостей інструменту Ahrefs та Google-бота для сканування та аналізу веб-сайтів. Це включає аналіз точності та повноти виявлення технічних і контентних помилок;
- проаналізувати переваги та недоліки. Вивчити плюси та мінуси кожного інструменту, зокрема їхню здатність працювати з великими сайтами, точність аналізу зворотних посилань, швидкість роботи та зручність використання;
- розглянути методи машинного навчання, що використовуються для кластеризації контенту;
- на основі отриманих даних розробити програмне забезпечення з інтеграцією методів машинного навчання для автоматизації рутинних процесів аналізу даних при веденні SEO-оптимізації;
- застосувати отримані результати кластеризації для стратегії просування веб-сайту в пошукових системах.

Задачі даного дослідження спрямовані на забезпечення глибокого розуміння можливостей та обмежень SEO-інструментів для сканування та аналізу веб-сайтів. Виконання цих задач дозволить обрати найефективніші рішення для оптимізації веб-сайтів і розробити стратегії їх використання в практичній SEO-діяльності.

4 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

SEO-оптимізація – це комплексний процес робіт, основні етапи якого: технічна оптимізація веб-сайту, створення якісного та унікального контенту, впровадження структурованих даних у форматі JSON та побудова стратегії зовнішніх посилань з релевантних ресурсів. Дані етапи узагальнені та розгалужуються на суттєву кількість підзадач, які залежать від першочергових факторів, таких як: вік домену, відсутність або наявність штрафних санкцій від пошукової системи Google, CMS або JavaScript-фреймворки за допомогою яких створено сайт, враховуючи технологію рендерингу сторінок: CSR або SSR, а також наявний посилальний профіль. Починаючи роботу з маленьких кроків та розробивши правильну стратегію просування можна отримати успішний результат з покращенням позицій в пошуковій видачі, що має значний вплив на конкурентоспроможність в умовах насиченого ринку.

4.1 Дослідження методів сканування та індексації сайтів

На даний момент штучний інтелект та його підгалузь машинне навчання мають значний вплив на SEO кількома способами [14]. Пошукові системи використовують машинне навчання для аналізу отриманих даних на відповідність до контексту пошукових запитів, завдяки чому визначають рейтинг веб-сторінок. На рисунку 1, зображено схему роботи Google-бота [15] із урахуванням більш складного сканування JavaScript-сайтів. Бот відвідує лише відкриті для сканування сторінки в індивідуальному файлі robots.txt з каталогу, де розміщено сайт. До кожного сайту з наявною позначкою User-agent: Googlebot з дозволом сканування для Google-бота, або позначкою User-agent: *, для всіх можливих ботів, відбувається наступний процес сканування.

На рисунку 4.1 наведено процес сканування сайтів, що створені засобами JavaScript-фреймворків та бібліотек.

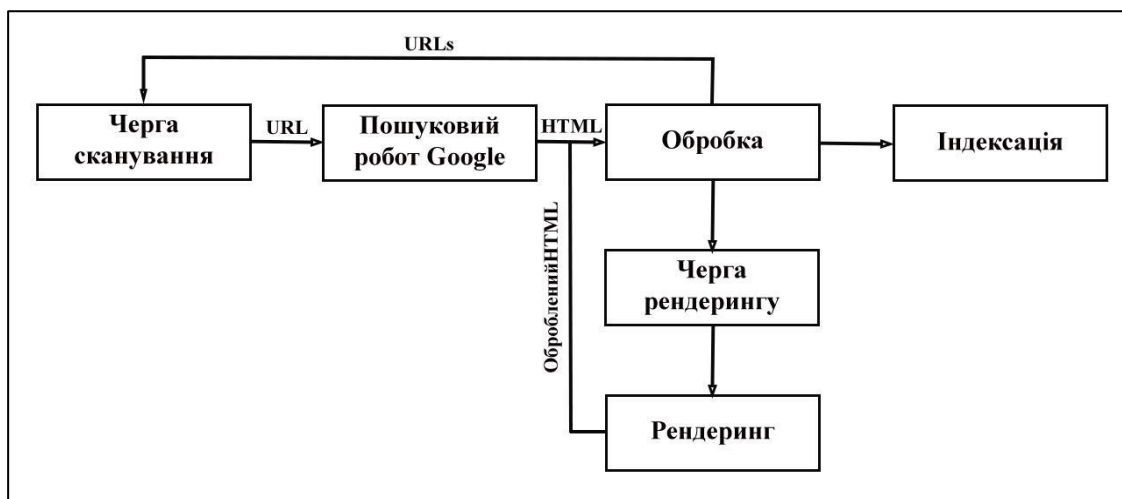


Рисунок 4.1 – Схема роботи Google-бота при скануванні сайтів (Створено самостійно)

Основна відмінність від сайтів, створених з використанням CMS, таких як: WordPress або Shopify, полягає у типі рендерингу сторінок. На даний момент лівова частка самописних сайтів створено без урахування рекомендованих умов від пошукових систем. До прикладу, веб-сайт створений з використанням стандартної бібліотеки для створення користувацьких інтерфейсів – React має технологію Client Side Rendering, в такому випадку відбувається двоетапний процес індексації. Спочатку Google-бот завантажує HTML-документ та програмний код JavaScript, після чого з використанням внутрішнього двигуна та середовища Chromium, виконує JavaScript-код, щоб отримати остаточний контент на сторінці. Така послідовність створює затримку між скануванням та індексацією веб-сайту.

Варто зазначити, що це не заперечує можливість сканування веб-сайту, але при такому сценарії витрачається в 2 рази більше краулінгового бюджету сайту [16], що призводить до зменшення ймовірності сканування, якомога більшої кількості сторінок.

Тому, перехід з наведеної бібліотеки React на JavaScript-фреймворк Next.js, який в основі використовує технологію Server Side Rendering та створений на базі React, дозволить зменшити кількість запитів до сайту та призведе до пришвидшення процесу сканування сторінок, оптимізувавши краулінговий бюджет, який визначається за двома факторами – краулінгова здатність (Crawl Capacity Limit), що

відповідає за максимальну кількість запитів, яку Google-бот може здійснити на сервер сайту, не створюючи проблем для його продуктивності та краулінгова потреба (Crawl Demand), що базується на важливості та свіжості контенту на сторінках сайту.

Також зараз дуже широко залучається машинне навчання в інструментах, що дозволяють аналізувати пошуковий трафік та дані про залучення користувача. Що в свою чергу дуже спрощує роботу для SEO-фахівця, допомагаючи визначити та зрозуміти напрямки, які варто вдосконалити. Оскільки Google ретельно приховує роботу своїх алгоритмів та ніколи не повідомляє про їх оновлення або створення нових, перед SEO-спеціалістами постає задача методом досліджень, аналізу та практичних експериментів – визначити вплив нововведень на старі опрацювання та виявити нові тренди оцінювання пошуковою системою.

До прикладу, Ahrefs надає потужний інструмент для дослідження великої кількості факторів оцінки веб-сайтів, що були досліджені за тривалий період часу та продовжують досліджуватись з постійними оновленнями пошукових алгоритмів Google. Розглянемо процес сканування пошукового бота від Ahrefs на рисунку 4.2.

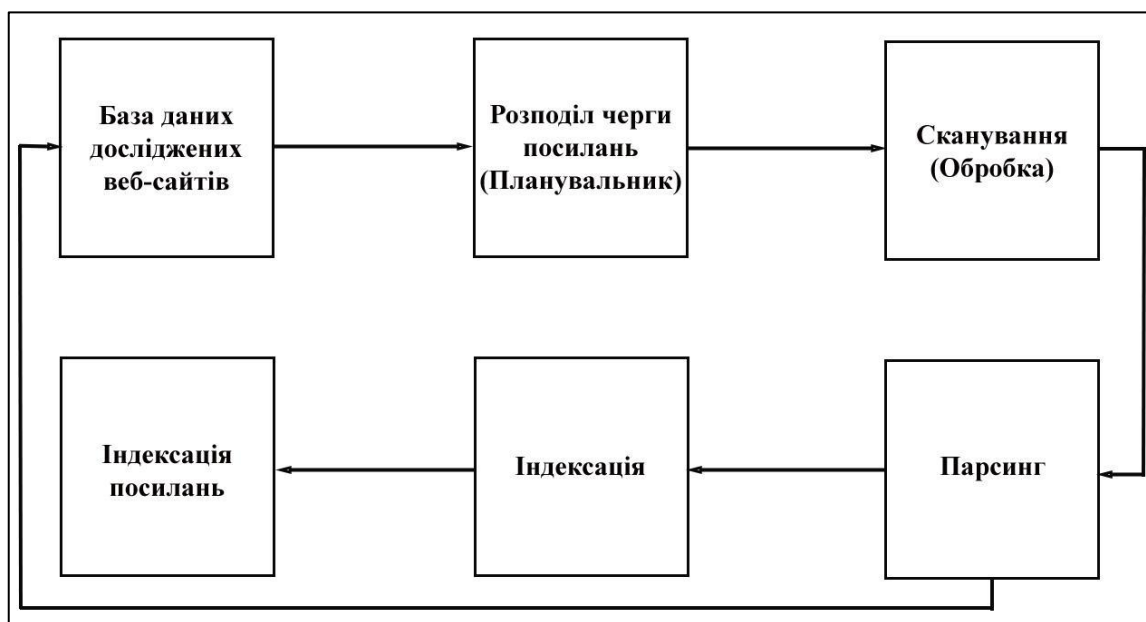


Рисунок 4.2 – Схема роботи бота Ahrefs при скануванні сайтів (Створено самостійно)

Сканування обраного користувачем сайту починається з модифікації великої бази даних, що складається із вже відомих сервісу URL, адреси сторінок сайту перенаправляються до планувальника, який створює чергу сторінок, які мають бути просканованими.

Коли URL-адреси сторінок будуть готові до аналізу, Ahrefs-бот надсилає їх до сканеру та завантажує вміст сторінок для обробки, суворо дотримуючись правил дозволу або заборони, встановлених у файлі robots.txt, який вже згадувався раніше. Робот-сканер доставляє необроблені дані до аналізатора, який робить парсинг сторінок та витягує посилання на цій сторінці, такі як: заголовок та інші відповідні метадані, після чого дані надсилаються на індексацію [17]. В результаті чого, отримані дані додаються до індексу посилань і стають доступним у різних звітах інструменту Ahrefs.

4.2 Методи машинного навчання

Метою експерименту, що проведений у цьому дослідженні, є інтеграція методів машинного навчання в програмне забезпечення для підвищення ефективності SEO-оптимізації.

Далі буде наведено розгляд методів, що використовуються для кластеризації контенту.

4.2.1 Кластеризація контенту

Кластеризація контенту базується на алгоритмах, таких як K-Means [18], які групують сторінки веб-сайту за тематичною схожістю. Це дає змогу визначати сегменти контенту, які потребують оптимізації, і покращувати їхню структуру. Наприклад, сторінки блогу можна згрупувати за ключовими словами для створення більш узгодженого змісту за формулою 4.1 .

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} (x_i - u_i)^2 \quad (4.1)$$

де k – число кластерів,

S_i – отримані кластери,

$i = 1, 2, \dots, k, u_i$ – центри мас векторів $x_j \in S_i$.

Цей метод застосовується для групування сторінок за тематичними категоріями, наприклад, для визначення контенту, який відповідає конкретним ключовим словам.

Основні методи машинного навчання на основі K-means:

- Mini-batch K-means є модифікацією базового K-means, яка працює з невеликими підвибірками даних (batch). Це дозволяє ефективно кластеризувати великі набори даних з обмеженими ресурсами. Метод характеризується високою швидкістю та оптимізацією для великих обсягів, але має трохи знижену точність порівняно зі стандартним K-means;
- Spectral Clustering базується на використанні матриці подібності для перетворення даних у простір, де вони стають лінійно роздільними. Після цього застосовується K-means. Метод ідеально підходить для кластеризації нелінійно роздільних даних. Основними перевагами є здатність обробляти складні структури, однак обчислювальна складність робить її менш ефективною для великих наборів даних;
- Deep Embedded Clustering (DEC) інтегрує нейронні мережі для зменшення розмірності даних з подальшим застосуванням K-means. Це дозволяє працювати зі складними багатовимірними даними, досягаючи найвищої точності кластеризації. Головною перевагою є здатність обробляти складні дані, хоча метод вимагає значних обчислювальних ресурсів і великої кількості даних для ефективного навчання.

Для тестування було взято датасети: Effective SEO parameters for all types of websites [19] та SEO Sample Data [20]. Дані для тестування включали інформацію про сторінки веб-сайтів, таку як ключові слова. Ці дані оброблялися через TF-IDF для векторизації текстових характеристик [21], а поведінкові сигнали та метадані слугували основою для кластеризації. Програмні засоби, зокрема Python з бібліотеками scikit-learn та tensorflow, використовувалися для реалізації методів. Метрики оцінки включали час виконання (середній час на одну ітерацію

кластеризації), точність за допомогою Silhouette Score та Adjusted Rand Index (ARI), а також масштабованість на наборах даних від 1000 до 100,000 сторінок. Результат наведено в таблиці 4.1.

Таблиця 4.1 – Порівняння методів (Таблиця виконана самостійно)

Метод	Час виконання	Точність (Silhouette Score)	Масштабованість	Примітки
K-means	2.5 с	87%	Висока	Базовий метод для кластеризації
Mini-batch K-means	1.8 с	85%	Дуже висока	Оптимізований для великих наборів даних
Spectral Clustering	4.5 с	90%	Середня	Для нелінійно роздільних даних
Deep Embedded Clustering	6.0 с	92%	Висока	Найкраща точність серед методів

Процес порівняння методів складався з трьох основних етапів. На першому етапі здійснювалася попередня обробка даних, яка включала нормалізацію даних через мінімаксне масштабування та векторизацію текстових характеристик за допомогою TF-IDF. На другому етапі виконувалося навчання моделей на двох наборах даних із однаковими параметрами. Наприклад, для Deep Embedded Clustering попередньо застосовувалися автоенкодері для зменшення розмірності даних. Третій етап включав оцінку методів, під час якої використовуввся набір даних. Крос-валідація забезпечувала перевірку стійкості методів та надавала додаткову впевненість у результатах.

Результати порівняння показали, що кожен метод має свої сильні та слабкі сторони залежно від обраного сценарію використання. Базовий K-means забезпечує швидке й точне групування сторінок, проте його ефективність знижується на великих обсягах даних. Mini-batch K-means демонструє оптимальні результати для великих наборів завдяки зменшенню обчислювальної складності, хоча його точність трохи нижча. Spectral Clustering підходить для даних із нелінійною структурою, але потребує більше ресурсів. Найкращі результати за точністю показала Deep Embedded Clustering (DEC), яка інтегрує нейронні мережі для зменшення розмірності та обробки багатовимірних даних, що робить її ідеальною для складних завдань, хоча й потребує значних обчислювальних ресурсів.

4.2.3 Інтеграція методів кластеризації в програмний продукт

Методи кластеризації було реалізовано у програмному продукті з використанням модульного підходу на основі мови Python. Архітектура передбачає створення графічного інтерфейсу користувача (GUI) за допомогою бібліотеки CustomTkinter, а також модулів для обробки текстових даних [22] із використанням TfidfVectorizer. Основний метод машинного навчання - це K-Means з бібліотеки scikit-learn, що було застосовано для кластеризації ключових слів.

Результати кластеризації було візуалізовано безпосередньо в інтерфейсі за допомогою Matplotlib. Виводитимуться графіки найпопулярніших ключових запитів та інтерактивна 2D-проекція розподілу кластерів на основі PCA. При наведенні курсору на точку графіка відобразатиметься назва відповідного кластера. У GUI також було реалізовано функції фільтрації, пошуку та копіювання кластерів у буфер обміну.

Цей програмний продукт забезпечує автоматизацію процесу SEO-аналізу: ключові слова будуть автоматично групуватись за змістом, що дозволить прискорити аналіз великих веб-сайтів та усунути потребу в ручній класифікації. Таким чином, очікується зниження часу на підготовку даних і можливість зосередитися на прийнятті стратегічних рішень.

4.3 Архітектура та проектування програмного забезпечення

Проектування програмного забезпечення для кластеризації ключових слів з метою SEO-аналізу є комплексним завданням, що вимагає поєднання сучасних методів обробки тексту, методів машинного навчання та зручного інтерфейсу користувача. Основна мета - створення ефективного інструменту, який забезпечить:

- точність: автоматичне групування ключових слів за змістом із мінімальними помилками;
- швидкість: ефективну обробку великих обсягів даних без затримок;
- масштабованість: можливість адаптації під різні обсяги даних та структури сайтів;
- інтеграцію з ML: використання методів кластеризації для виявлення тем та побудови тематичних груп.

Розробка програмного забезпечення базується на принципах модульності, що забезпечує простоту масштабування, модифікації та підтримки. Архітектура програми побудована за багаторівневим підходом, який розділяє логіку на незалежні компоненти: інтерфейс користувача на основі CustomTkinter, модуль обробки текстових даних TfidfVectorizer та модуль кластеризації K-Means.

Така архітектура дозволяє гнучко адаптувати програму під різні сценарії використання: від підготовки семантичного ядра до кластеризації сторінок для технічного SEO. Очікується, що система значно скоротить час на ручну обробку даних та підвищить якість стратегічного планування в сфері пошукової оптимізації.

4.3.1 Архітектурний підхід

Була обрана модульна архітектура, однокористувацька система з графічним інтерфейсом з реалізацією на мові програмування Python. Програма побудована за принципами багаторівневої логіки, де кожен функціональний блок відповідає за окрему частину процесу від завантаження CSV-файлу до виводу графіків кластеризації.

Користувацький рівень (GUI), реалізовано за допомогою бібліотеки CustomTkinter, що забезпечує сучасний вигляд, легку навігацію та інтерактивність. SEO-фахівець має змогу:

- завантажити CSV-файл із ключовими словами та пошуковим обсягом;
- вказати кількість кластерів;
- запустити кластеризацію;
- переглянути результати у вигляді груп (кластерів);
- копіювати окремі або всі кластери у буфер обміну;
- ознайомитися з інтерактивними графіками: ТОП ключів та PCA-візуалізацією розподілу кластерів.

Аналітичний рівень передбачає використання методу K-Means з бібліотеки scikit-learn для кластеризації ключових слів з попереднім виконанням векторизації тексту за допомогою TfidfVectorizer. Кожен кластер має отримувати відповідну назву на основі найчастіших слів, визначених через CountVectorizer.

Візуалізаційний рівень реалізовано у вигляді побудови графіків з виведенням гістограми найпопулярніших ключових запитів за обсягом пошуку та 2D-візуалізацією розподілу кластерів у просторі ознак за допомогою PCA для зменшення розмірності векторного простору, в якому представлені ключові слова після обробки. Оскільки після векторизації кожне слово описується багатовимірним числовим вектором, візуалізація таких даних є неможливою без попереднього спрощення. PCA дозволяє проєктувати ці вектори у двовимірний простір, зберігаючи при цьому максимально можливу частку інформації.

4.3.2 Use-Case-діаграма

Було спроектовано діаграму сценаріїв використання (див. рис. 4.3).



Рисунок 4.3 – Діаграма сценаріїв використання (Створено самостійно)

Основним актором системи є SEO-фахівець, який взаємодіє з програмою для кластеризації ключових слів. Користувач має змогу завантажити CSV-файл із ключовими словами, вказати необхідну кількість кластерів для аналізу, запустити процес кластеризації, переглянути отримані результати у вигляді тематичних груп, скопіювати окремі або всі кластери в буфер обміну, а також ознайомитися з візуалізацією даних через графіки.

4.3.4 UML-Діаграма класів

Спроектована діаграма класів (див.рис. 4.4).

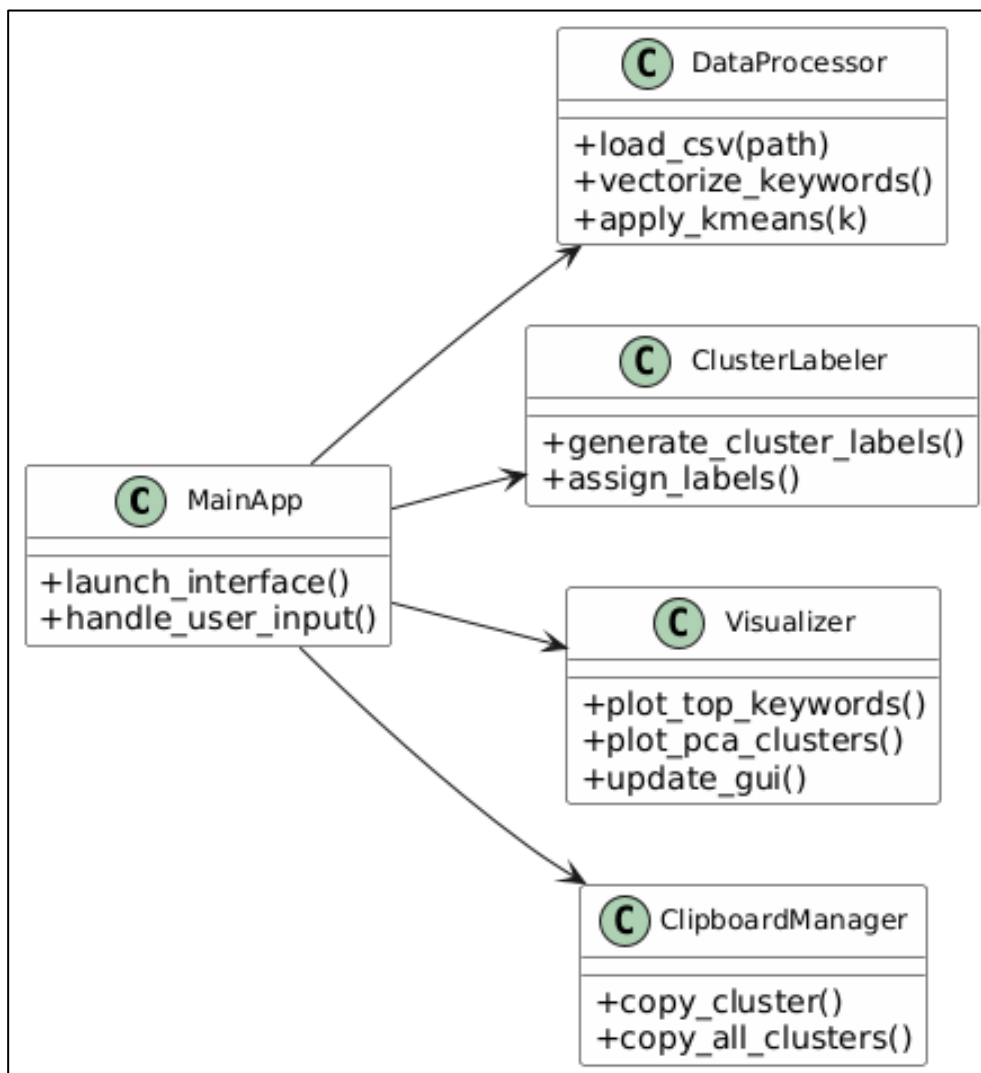


Рисунок 4.4 – Діаграма класів (Створено самостійно)

Основні класи:

- **MainApp**: відповідає за запуск програми, ініціалізацію графічного інтерфейсу користувача (GUI), обробку дій користувача, розміщення елементів керування та передавання даних іншим модулям;
- **DataProcessor**: виконує обробку вхідного CSV-файлу, зчитування ключових слів та пошукового обсягу, векторизацію тексту за допомогою `TfidfVectorizer` і кластеризацію з використанням методу K-Means;

- ClusterLabeler: реалізує логіку автоматичного створення назв кластерів. Після завершення кластеризації цей клас опрацьовує ключові слова в кожному кластері, визначає найчастотніші лексеми за допомогою аналізу частоти слів, і на основі них формує коротку назву для кожної групи. Це дозволяє користувачу швидко зорієнтуватися в тематиці кластеру;
- Visualizer: формує графіки з використанням Matplotlib: гістограму найпопулярніших ключових слів за обсягом пошуку та візуалізацію кластерів у 2D-просторі на основі PCA. Також відповідає за оновлення виводу в інтерфейсі.
- ClipboardManager: реалізує функціональність копіювання: як окремого кластеру, так і всіх кластерів разом. Це забезпечує зручність при подальшому використанні результатів, наприклад, під час створення семантичного ядра в редакторі.

Усі ці класи взаємодіють через головний модуль MainApp, який координує їхню роботу, забезпечує послідовність виконання основних операцій та відповідає за оновлення графічного інтерфейсу в режимі реального часу. MainApp виступає як зв'язувальна ланка між логікою обробки даних, кластеризацією, візуалізацією та взаємодією з користувачем, делегуючи завдання відповідним модулям. Завдяки чітко розподіленій відповідальності між класами, структура програми залишається гнучкою та масштабованою. Це дозволяє безпечно змінювати або замінювати окремі компоненти, наприклад, експериментувати з іншими методами кластеризації, типами векторизації чи форматами виводу, не впливаючи на загальну працездатність системи.

5 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

Порядок проведення експерименту:

- підготувати тестовий CSV-файл, що містить неупорядкований набір ключових слів з різними пошуковими намірами та реальними значеннями пошукового обсягу;
- запустити програму та завантажити тестовий файл через графічний інтерфейс користувача;
- провести кластеризацію та зафіксувати результати у вигляді візуалізації кластерів, графіків та нового вихідного CSV-файлу;
- оцінити точність формування кластерів шляхом ручної перевірки на відповідність кожного запита, що міститься в одному кластері до реальних пошукових намірів в результатах пошукової видачі Google.

На основі отриманих даних потрібно зробити висновки, щодо якості роботи методів машинного навчання та потенційних шляхів удосконалення.

5.1 Знімки екрану та приклади коду

Для проведення експерименту було розроблено застосунок призначений для пришвидшення підготовчих робіт SEO-оптимізації, а саме кластеризації семантичного ядра. Застосунок приймає CSV-файл з неупорядкованим набором ключових слів з реальними показниками пошукової частотності та різними пошуковими намірами.

Застосунок також формує графіки рейтингу ключових слів за обсягами пошуку та розподілу кластерів, що дуже важливо на етапі аналізу великих обсягів вхідних даних для формування стратегії та першочергових пріоритетів для просування в результатах органічної пошукової видачі Google.

Головний екран програми містить елементи керування: поле для імпорту неупорядкованого CSV-файлу, поле з кількістю сформованих кластерів, пошук кластерів та кнопки початку кластеризації та копіювання семантичного ядра.

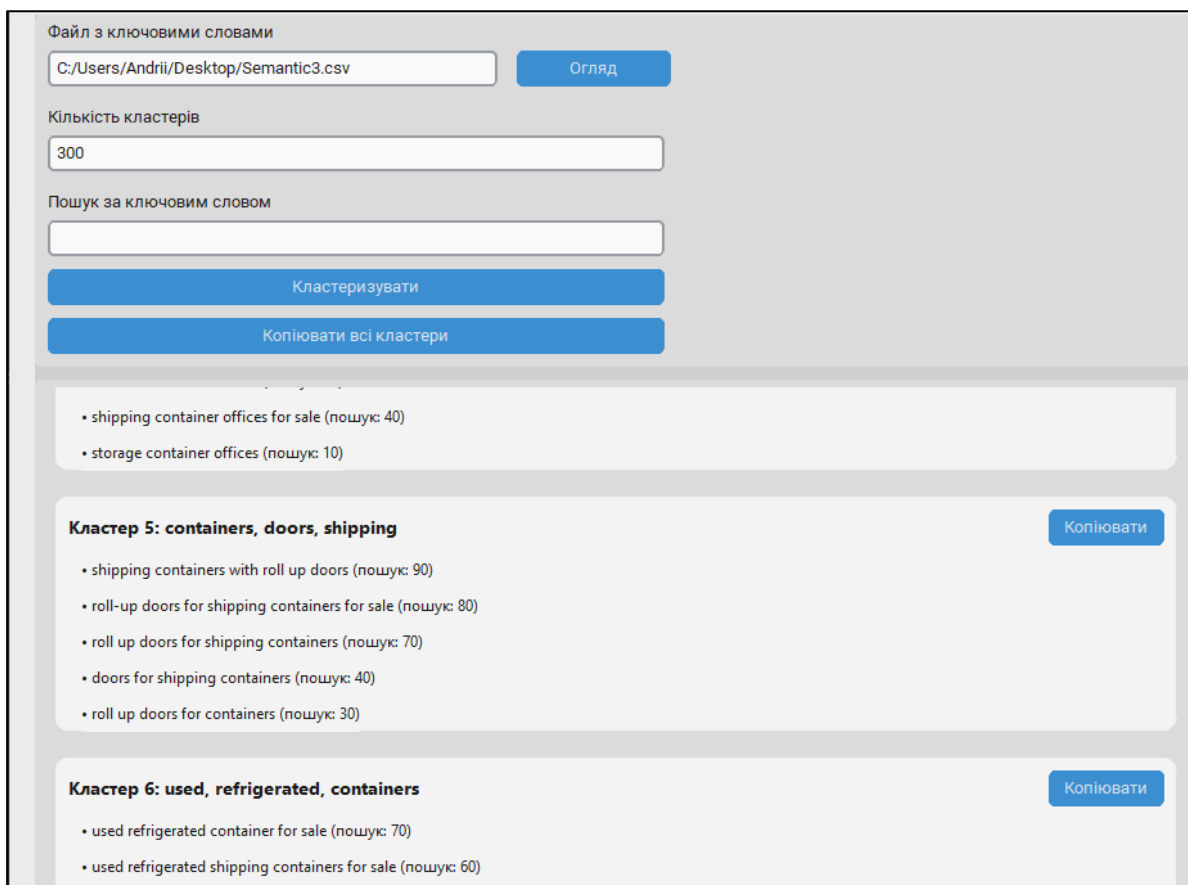


Рисунок 5.1 – Фрагмент головного екрану програми (Створено самостійно)

Для проведення експерименту було вивантажено реальний перелік ключових слів, за яким просуюються сторінки сайту з продажу морських контейнерів в ГЕО: Сполучені Штати Америки, штат Каліфорнія.

У створеній програмі кластери формуються за допомогою K-Means, який об'єднує подібні ключові слова на основі їх векторного представлення, сформованого через TfidfVectorizer. Після кластеризації для кожного кластера програма аналізує всі слова, що до нього належать, і з використанням CountVectorizer визначає три найчастотніші слова. Саме ці слова автоматично формують назву кластера, оскільки вони є найрелевантнішими для відповідної групи. Обмеження до трьох слів дозволяє зберігати назву короткою, інформативною та зручною для візуального сприйняття.

Програмний код, що обробляє вхідний CSV-файл та виконує кластеризацію семантичного ядра:

```

keywords = df['keyword'].astype(str)
tfidf = TfidfVectorizer(stop_words='english')
X = tfidf.fit_transform(keywords)

kmeans = KMeans(n_clusters=k, random_state=42)
df['cluster'] = kmeans.fit_predict(X)
counter = CountVectorizer(stop_words='english')
cluster_labels = {}
for cluster_id in sorted(df['cluster'].unique()):
    items = df[df['cluster'] == cluster_id]['keyword'].tolist()
    X_cluster = counter.fit_transform(items)
    freqs = X_cluster.sum(axis=0).A1
    words = counter.get_feature_names_out()
    top = [words[i] for i in freqs.argsort()[::-1][:3]]
    cluster_labels[cluster_id] = ", ".join(top)
df['cluster_label'] = df['cluster'].map(cluster_labels)
df.to_csv("clustered_keywords.csv", index=False)

```

Для кращого визначення пріоритетів для просування при розробці стратегії перед початком проведення SEO-оптимізації виведено графік з рейтингом найпопулярніших ключових слів за пошуковим обсягом.

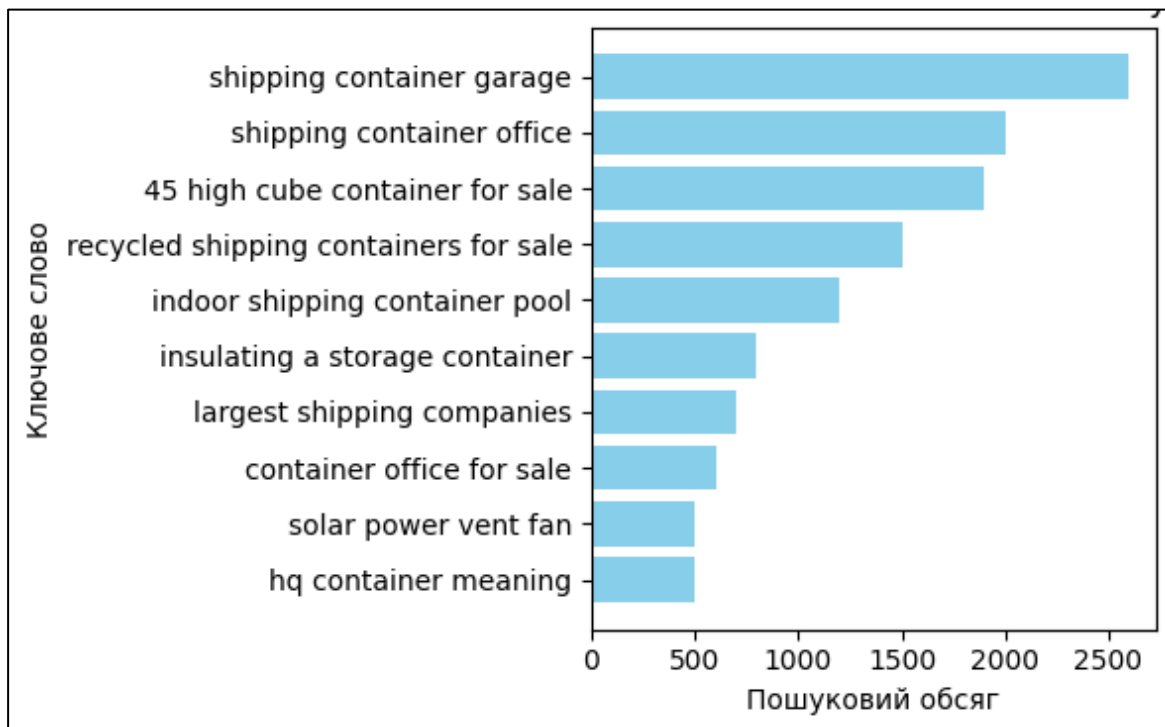


Рисунок 5.2 – Графік рейтингу ключових слів (Створено самостійно)

Код, що генерує графік рейтингу ключових слів:

```
top_keywords=df[['keyword',
'search_volume']].sort_values(by='search_volume', ascending=False).head(10)
fig1, ax1 = plt.subplots(figsize=(6, 4))
ax1.barh(top_keywords['keyword'][::-1],
top_keywords['search_volume'][::-1], color='skyblue') # [::-1]
ax1.set_title("ТОП ключових слів за обсягом пошуку")
ax1.set_xlabel("Пошуковий обсяг")
ax1.set_ylabel("Ключове слово")
fig1.tight_layout()
canvas1 = FigureCanvasTkAgg(fig1, master=chart_frame)
canvas1.draw()
canvas1.get_tk_widget().pack(fill="x", padx=10, pady=(0, 10))
```

Графік ключових слів генерується шляхом вибору десяти ключових запитів із найвищим пошуковим обсягом із вхідного CSV-файлу. Ці дані візуалізуються у вигляді горизонтальної гистограми за допомогою бібліотеки Matplotlib. По осі Y розміщено ключові слова, а по осі X відповідні значення їх пошукової частотності. Для зручності сприйняття графік відображає найпопулярніші слова зверху. Побудований графік інтегрується безпосередньо в графічний інтерфейс програми, що дозволяє користувачу швидко оцінити, які ключові запити є найважливішими з точки зору популярності.

Другий графік будується з використанням методу PCA (Principal Component Analysis) та призначений для візуалізації розподілу кластерів у двовимірному просторі. Оскільки машинне навчання оперує багатовимірними векторними уявленнями слів, PCA дозволяє зменшити ці вектори до двох координат, зберігаючи при цьому максимально можливу інформативність.

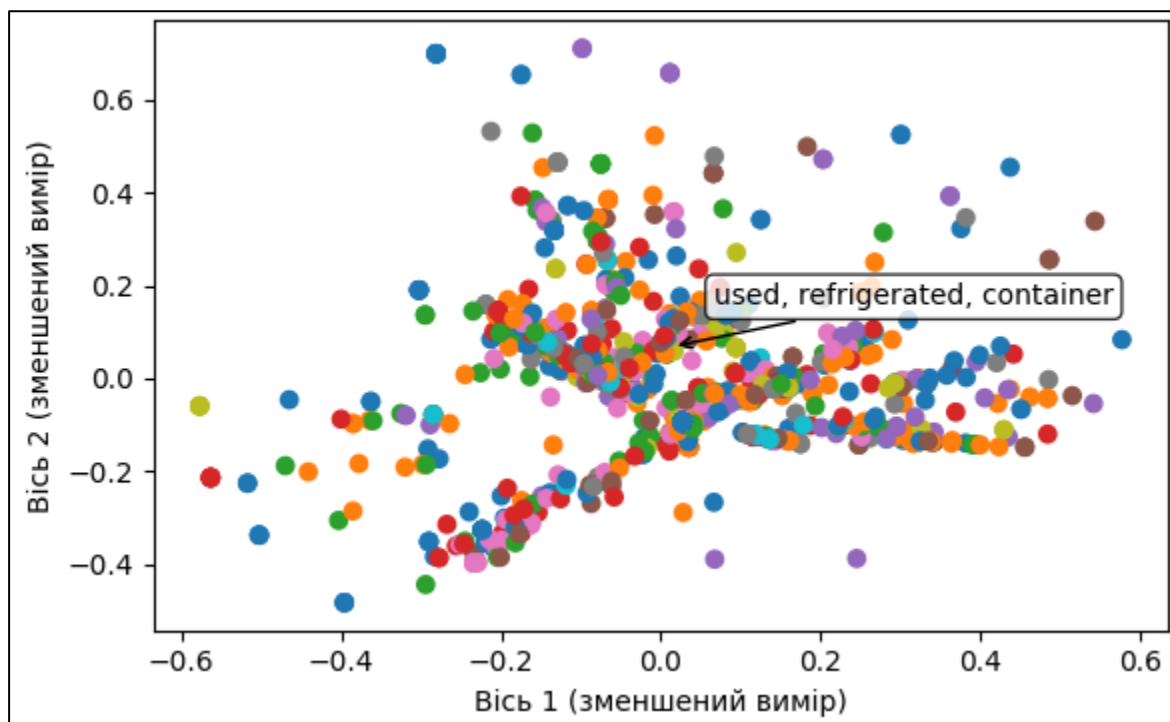


Рисунок 5.3 – Графік розподілу кластерів у 2D просторі (PCA) (Створено самостійно)

Код, що відповідає за побудову графіку PCA:

```
pca = PCA(n_components=2)
reduced = pca.fit_transform(X.toarray())
fig2, ax2 = plt.subplots(figsize=(6, 4))
scatter = ax2.scatter(reduced[:, 0], reduced[:, 1], c=df['cluster'],
map='tab10')
ax2.set_title("Розподіл кластерів у 2D-просторі (PCA)")
ax2.set_xlabel("Вісь 1 (зменшений вимір)")
ax2.set_ylabel("Вісь 2 (зменшений вимір)")
annot = ax2.annotate("", xy=(0, 0), xytext=(15, 15), textcoords="offset
points",
bbox=dict(boxstyle="round", fc="w"),
arrowprops=dict(arrowstyle="->"))
annot.set_visible(False)

def update_annot(ind):
    idx = ind["ind"][0]
    pos = scatter.get_offsets()[idx]
```

```

annot.xy = pos
text = df.iloc[idx]['cluster_label']
annot.set_text(text)
annot.get_bbox_patch().set_alpha(0.8)

def hover(event):
    vis = annot.get_visible()
    if event.inaxes == ax2:
        cont, ind = scatter.contains(event)
        if cont:
            update_annot(ind)
            annot.set_visible(True)
            fig2.canvas.draw_idle()
        elif vis:
            annot.set_visible(False)
            fig2.canvas.draw_idle()
fig2.canvas.mpl_connect("motion_notify_event", hover)

```

Таким чином, графік демонструє, як ключові слова згруповані у простір, де кожна точка представляє окреме слово, а колір вказує на його належність до певного кластера. Це дозволяє користувачу візуально оцінити якість кластеризації, зокрема, чи дійсно групи слів утворюють окремі області, а також виявити можливо неправильне об'єднання або перетин тем. Графік інтерактивний: при наведенні відображається назва кластера, до якого належить точка.

5.2 Перевірка результату кластеризації в пошуковій видачі

Для оцінки точності кластеризації семантичного ядра доцільно провести ручну перевірку результатів. Це передбачає порівняння запитів у межах кластерів з фактичною пошуковою видачею Google. Такий підхід дозволяє визначити, чи дійсно запити, згруповані разом, мають спільний пошуковий намір (інтент) та ведуть до схожих типів сторінок. Ручна перевірка особливо важлива для виявлення змішаних або помилково об'єднаних кластерів, які могли виникнути через лексичну або структурну схожість, але різний контекст використання.

Для виконання перевірки кластеризації було навмання взято кластер та перші 5 результатів пошукової видачі по кожному запиту (див. рис. 5.4).

Кластер 40: opening, shipping, container
• side opening containers for sale (пошук: 100)
• side opening shipping containers (пошук: 80)
• shipping container side opening (пошук: 50)
• side opening storage container (пошук: 20)

Рисунок 5.4 – Навмання взятий кластер (Створено самостійно)

Також було налаштовано анонімний перегляд та використано ГЕО: Сполучені Штати Америки, штат Каліфорнія для запобігання показу персоналізованих результатів від попередніх запитів, а також показу україномовних сторінок.

В результаті було вивантажено наступні результати (див.табл. 5.1):

Таблиця 5.1 – Результати пошукової видачі (Таблиця виконана самостійно)

Запит	Мета-теги Title в результатах пошуку (ТОП-5)
side opening container for sale	Buy a 20ft Open Side Container
	20FT Open Side New (One Trip) Shipping Container
	Open-Side Shipping Containers for Sale
	Open sided shipping containers for sale
	Open Side Containers for Sale or Lease
side opening containers	Buy a 20ft Open Side Container
	Open-Side Shipping Containers for Sale
	Open sided shipping containers for sale
	Side Opening Shipping Containers
	20FT Open Side New (One Trip) Shipping Container
shipping container side opening	Buy a 20ft Open Side Container
	20FT Open Side New (One Trip) Shipping Container

Кінець таблиці 5.1

Запит	Мета-теги Title в результатах пошуку (ТОП-5)
	Side-Door Shipping Containers for Sale
	Modular Open-Sided Shipping Containers Features MMPS
	Open sided shipping containers for sale
side opening storage container	Buy a 20ft Open Side Container
	Open-Side Shipping Containers for Sale
	Modular Open-Sided Shipping Containers Features MMPS
	20FT Open Side New (One Trip) Shipping Container
	Open sided shipping containers for sale

Усі 4 запити стосуються придбання чи пошуку бічного (side opening) контейнера. Незалежно від формулювання: «for sale», «storage», або «shipping container side opening», результати пошукової видачі Google ведуть до однакових або схожих сторінок продажу контейнерів із бічними дверима, зазвичай формату 20ft.

У результаті проведеного експерименту щодо перевірки релевантності кластеру пошукових запитів було встановлено, що всі запити ведуть до тематично однорідних результатів у пошуковій видачі Google. Більшість Title-посилань повторюються або мають семантичну відповідність, що свідчить про спільний пошуковий намір: придбання або вивчення характеристик контейнерів із боковим відкриванням. Такий рівень збіжності підтверджує коректність сформованого кластеру. Варто зазначити, що запити мають різну пошукову частотність: від 20 до 100 пошуків на місяць. Завдяки цьому є можливість оптимізації цільової сторінки під кілька запитів різної частотності. Це дозволяє не лише зберегти релевантність, але й забезпечити більш ефективне охоплення цільової аудиторії.

5.3 Порівняння розробленої програми з існуючими SEO-інструментами

У процесі аналізу та вибору найкращого SEO-інструменту було визначено, що до найефективніших сучасних інструментів для SEO-аналізу належать Ahrefs, Serpstat та Screaming Frog. Ці платформи активно розвиваються та поступово

імплементують методи машинного навчання для покращення точності аналізу, прогнозування та автоматизації рутинних завдань.

З-поміж зазначених систем Serpstat є найбільш наближеним до логіки кластеризації, реалізованої у розробленій програмі. Serpstat дозволяє об'єднувати ключові слова в кластери на основі спільних результатів пошукової видачі та обирати рівень «жорсткості» кластеризації (soft/hard clustering). Проте Serpstat не дає змоги змінювати модель векторизації або метод кластеризації, та не передбачає візуального відображення результатів у вигляді 2D-графіків, що обмежує інтерпретованість кластерів.

В свою чергу Ahrefs надає широкі можливості для збору ключових слів, аналізу конкурентів і побудови SEO-стратегії, однак не має функції автоматичної кластеризації ключових слів за пошуковим наміром. Інструмент може показати Parent Topic або схожі запити, проте подальше групування доводиться здійснювати вручну або через сторонні сервіси.

Програма Screaming Frog насамперед технічний сканер сайтів, який спеціалізується на аудиті структури сторінок, мета-даних, помилок, редиректів тощо. Він не підтримує кластеризацію ключових слів як частину свого функціоналу.

Натомість розроблена програма пропонує відкриту та гнучку структуру кластеризації на базі методів машинного навчання, де використовується TF-IDF векторизація текстів, а результати подаються в текстовому вигляді та у вигляді графіків. Програма також дає змогу переглядати, копіювати кластери, шукати запити за ключовим словом та аналізувати найпопулярніші запити за частотністю. Завдяки цьому, програма забезпечує глибший контроль над процесом обробки та інтерпретації семантичного ядра.

Таким чином, розроблена програма виконує конкретну задачу автоматизованої кластеризації семантичного ядра, яку або не реалізовано, або реалізовано частково в інших популярних SEO-системах.

5.4 Практичне застосування

Одним із найбільш перспективних напрямів застосування такої кластеризації ключових слів є реалізація стратегії Programmatic SEO. Дана стратегія передбачає масове генерування сторінок на основі шаблонів та кластеризованих даних.

Завдяки сформованому семантичному ядру, де кожен кластер репрезентує певну тему або підкатегорію, можна створювати цільові посадкові сторінки, релевантні до конкретного набору запитів. Наприклад, кластер із запитами типу «roll up door kit for shipping container», «container roller door» може бути в основі однієї сторінки оптимізованої саме під цей сегмент попиту або розділеними на різні сторінки окремо під кожний запит.

Однією з ключових можливостей якісного просування в результатах пошуку є створення посадкових сторінок на основі кластерів із низькочастотними, але деталізованими запитами.

Це дозволяє сторінкам швидко зайняти високі позиції в пошуку навіть без значного авторитету сайту або великої кількості зворотних посилань. Такі запити часто мають чіткий комерційний або інформаційний намір, що підвищує кількість переходів на сайт та ймовірність конверсії, про що свідчить графік показників ефективності в Google Search Console (див. рис. 5.5).



Рисунок 5.5 – Графік показників ефективності з персонального акаунту Google Search Console за весь доступний період відслідковування сайту (Створено самостійно)

У межах програматичної SEO-стратегії використання кластеризованого семантичного ядра дає змогу масово охопити тисячі таких запитів, що в сумі формує істотний обсяг цільового трафіку з високим потенціалом залучення та монетизації.

Для детальнішого відслідковування отриманих результатів наведено порівняння графіків до та після початку контентної оптимізації сайту, а саме масового створення посадкових сторінок під кластери отриманих запитів, що були отримані ході використання розробленого застосунку (див .рис. 5.7).



Рисунок 5.6 – Порівняння графіків на персональному акаунті Google Search Console до та після початку контентної оптимізації (Створено самостійно)

Після впровадження розробленої програми для практичного застосування, за період останніх 4 місяців проведення оптимізації кількість переходів на сайт з результатів органічного пошуку стабільно зросла на 130% з 2,41 тисячі до 5,54 тисяч. Водночас загальна кількість показів в пошуковій видачі зросла на 138% з 124 тисяч до 295 тисяч.

Якісно сформоване семантичне ядро є одним із ключових елементів успішної SEO-стратегії, при роботі з великими масивами ключових слів, ручне групування стає складним та надзвичайно трудомістким. Саме тому застосування методів машинного навчання для автоматизованої кластеризації набуває стратегічного значення. Це не лише економить час та зусилля SEO-фахівців, але й забезпечує вищу точність структурування семантики, що безпосередньо впливає на якість створених сторінок, їх релевантність та ефективність у пошуковій видачі.

ВИСНОВКИ

В результаті роботи було виконано низку завдань, поставлених на початку дослідження. Основною метою було дослідити методи для аналізу індексації веб-сайтів пошуковими системами, а також створити теоретичну базу для розробки програмного забезпечення, яке враховує сучасні виклики SEO та використовує методи машинного навчання.

В межах роботи також було розроблено програмне забезпечення для кластеризації ключових слів з метою підвищення ефективності семантичного аналізу в SEO. Програма дозволяє автоматично обробляти великі списки запитів, групуючи їх за змістом, що значно спрощує та пришвидшує формування семантичного ядра.

У ході роботи було також проведено порівняння популярних методів кластеризації. У результаті аналізу обрано метод K-Means, який продемонстрував найкраще співвідношення точності, простоти реалізації та швидкодії для задачі групування текстових даних.

Проведений експериментальний аналіз із використанням реальних пошукових запитів, за яким ранжуються сайти-конкуренти підтвердив релевантність сформованих кластерів та відповідність їх пошуковому наміру. Це демонструє доцільність використання автоматизованої кластеризації в процесі SEO-оптимізації.

У порівнянні з провідними інструментами для роботи з ключовими словами, такими як Serpstat або Ahrefs, розроблений продукт забезпечує прозорість логіки кластеризації та можливість її адаптації під конкретні задачі. Крім того, автоматизація процесу дозволяє суттєво заощадити час, який зазвичай витрачається на ручну обробку, та зосередитись на стратегічних аспектах пошукової оптимізації.

Після практичного застосування отриманих даних було зафіксовано значний приріст ефективності у веденні SEO-оптимізації, а саме зріст переходів на сайт на 130% та так зріст кількості показів в результатах пошуку на 138%. Отже, отримані результати свідчать про практичну цінність розробленого продукту та його перспективність для подальшого вдосконалення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Google Ranking Factors. Semrush. 2024. URL: <https://go.semrush.com/Ranking-Factors.html#form> (Дата звернення: 05.05.2025).
2. Як позиція в пошуковій видачі Google впливає на CTR. web-promo. 2020. URL: <https://web-promo.ua/ua/blog/kak-pozicziya-v-poiskovoj-vydache-google-vliyaet-na-ctr-aktualnaya-statistika-2020/> (Дата звернення: 05.05.2025).
3. Фільтри Google: види, діагностика, відновлення. Wezom. 2023. Режим доступу до ресурсу: <https://wezom.com.ua/ua/blog/yak-ne-potrapiti-pid-filtri-google> (Дата звернення: 05.05.2025).
4. Що таке E-E-A-T і чому він важливий для SEO. 2023. URL: <https://wordfactory.ua/ua/eeat/> (Дата звернення: 05.05.2025).
5. Triantaphyllou E. Multi-criteria Decision Making Methods: A Comparative Study. Springer Science & Business Media, 2000. 290 p.
6. How to get your website on Google Search. Google. 2024. URL: <https://developers.google.com/search> (Дата звернення: 05.05.2025).
7. Google's Machine Learning in Content Ranking. Seo.ai. 2024. URL: <https://seo.ai/blog/googles-machine-learning-in-content-ranking> (Дата звернення: 05.05.2025).
8. The Role of Machine Learning in SEO: A Comprehensive Guide. Pageoptimizer.pro. 2024. URL: <https://www.pageoptimizer.pro/blog/the-role-of-machine-learning-in-seo-a-comprehensive-guide> (Дата звернення: 09.05.2025).
9. Natural Language Understanding (NLU). Ahrefs. 2024. URL: <https://ahrefs.com/seo/glossary/natural-language-understanding> (Дата звернення: 09.05.2025).
10. Загальна документація по API (з чого почати роботу). Serpstat. 2024. URL: <https://serpstat.com/uk/api/> (Дата звернення: 09.05.2025).
11. Лозиченко А. В., Мельнікова Р. В. Дослідження методів виявлення помилок проектування сайтів // Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління : тез. доп. дванадцятої міжнародної

науково-технічної конференції, 27–28 квітня 2022 р. – Т. 2. – Баку–Харків–Жиліна, 2022. – С. 180.

12. Screaming Frog Guide For The SEO Spider. Screaming Frog. 2024. URL: <https://www.screamingfrog.co.uk/seo-spider/user-guide/> (Дата звернення: 09.05.2025).

13. Google Describes a Machine Learning Model For a Searchable Index. Go Fish Digital. 2021. URL: <https://gofishdigital.com/blog/machine-learning-model/> (Дата звернення: 09.05.2025).

14. The Impact of Machine Learning on SEO. Marketbrew. 2024. URL: <https://marketbrew.ai/the-impact-of-machine-learning-on-seo> (Дата звернення: 09.05.2025).

15. Understand the JavaScript SEO basics. Google. 2024. URL: <https://developers.google.com/search/docs/crawling-indexing/javascript/javascript-seo-basics> (Дата звернення: 09.05.2025).

16. What is Crawl Budget? Ahrefs. 2024. URL: <https://ahrefs.com/seo/glossary/crawl-budget> (Дата звернення: 09.05.2025).

17. Мартиненко А. О., Мельнікова Р. В. Дослідження методів сканування та індексації веб-сайтів пошуковою системою Google. Сучасні комп'ютерні та інформаційні системи і технології: матеріали IV Всеукраїнської наук.-практ. інтернет-конф. (Запоріжжя, 9-20 грудня 2024 р.). Запоріжжя: ТДАТУ, 2024. С -78-81.

18. Bishop C. M. Pattern Recognition and Machine Learning. Springer Science & Business Media, 2006. 758 p.

19. Effective SEO parameters for all types of websites. Kaggle. 2024. URL: <https://www.kaggle.com/datasets/ashkangoharfar/sites-information-data-from-alexa-com-dataset> (Дата звернення: 09.05.2025).

20. SEO Sample Data. Kaggle. 2024. URL: <https://www.kaggle.com/datasets/muhammetvarl/seo-sample-data> (Дата звернення: 09.05.2025).

21. Мартиненко А.О. Кластеризація контенту з використанням алгоритмів машинного навчання. 29-1 Міжнародний молодіжний форум «Радіоелектроніка та

молодь у XXI столітті». (Харків, 16-19 квітня 2025 року.). 3б. матеріалів форуму. Т.6., - Харків: ХНУРЕ. 2025. – 630 с.

22. Sharonova, N., Kyrychenko, I., Gruzdo, I., Tereshchenko, G. Generalized Semantic Analysis Algorithm of Natural Language Texts for Various Functional Style Types CEUR Workshop Proceedings, 2022, 3171, pp. 16–26.

23. Github - Keyword-Clustering-Tool-for-SEO-Optimization. URL: https://github.com/awdrallyracer/Keyword-Clustering-Tool-for-SEO-Optimization/2025_M_III_IP3-23-2_Мартиненко_А_О (Дата звернення: 09.05.2025).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

11. Лозиченко А. В., Мельнікова Р. В. Дослідження методів виявлення помилок проектування сайтів // Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління : тез. доп. дванадцятій міжнародній науково-технічній конференції, 27–28 квітня 2022 р. – Т. 2. – Баку–Харків–Жиліна, 2022. – С. 180.

17. Мартиненко А. О., Мельнікова Р. В. Дослідження методів сканування та індексації веб-сайтів пошуковою системою Google. Сучасні комп'ютерні та інформаційні системи і технології: матеріали IV Всеукраїнської наук.-практ. інтернет-конф. (Запоріжжя, 9-20 грудня 2024 р.). Запоріжжя: ТДАТУ, 2024. С -78-81.

21. Мартиненко А.О. Кластеризація контенту з використанням алгоритмів машинного навчання. 29-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». (Харків, 16-19 квітня 2025 року.). 3б. матеріалів форуму. Т.6., - Харків: ХНУРЕ. 2025. – 630 с.

22. Sharonova, N., Kyrychenko, I., Gruzdo, I., Tereshchenko, G. Generalized Semantic Analysis Algorithm of Natural Language Texts for Various Functional Style Types CEUR Workshop Proceedings, 2022, 3171, pp. 16–26.