

# ML Based SEO Performance Analysis and Recommendation System

Dr. Pravin Kumar D<sup>1</sup>, Preethi T V<sup>2</sup>, Reshma Sri MU<sup>3</sup>

<sup>1</sup>Assist prof, Dept of CSE

<sup>2, 3</sup> Dept of CSE

<sup>1, 2, 3</sup> K.L.N.College of Engineering, Sivagangai, India

**Abstract-** Search Engine Optimization (SEO) is important for improving the visibility of websites in search results, but checking every page of a website manually is slow and difficult for non-technical owners. To address this problem, this paper proposes an ML based SEO performance analysis and recommendation system built using Python and Flask. The user enters a website URL, and the system crawls the accessible pages using Requests and BeautifulSoup and extracts six important SEO features: content length, keyword density, internal links, external links, meta description and image alt text. An SEO score out of 100 is calculated using predefined rules, and the strengths and weaknesses of the website are identified. A Random Forest classifier trained on an SEO dataset predicts whether the ranking-improvement outcome is likely to be positive. Based on the analysis, the system recommends actions such as adding relevant content, improving internal linking and including missing meta descriptions and image alt text. The results are stored in a MySQL database and displayed through a user-friendly web interface, helping website owners to understand common SEO issues and identify possible improvements without checking every page manually.

**Keywords-** BeautifulSoup, Flask, Random Forest, Search Engine Optimization, Web Crawling

## I. INTRODUCTION

Search Engine Optimization (SEO) is the practice of improving a website so that search engines can find, understand and rank its pages in the organic results. Most users never look beyond the first page of results, so the ranking of a website directly affects its traffic and credibility. The factors that influence ranking include on-page factors such as content, meta tags, links and images, which can be measured directly from the HTML of a page.

However, many website owners, students and small businesses do not have the knowledge to check whether their websites follow SEO guidelines. Checking every page manually is time-consuming and inconsistent, and professional SEO tools are often costly or complex. This project introduces a web-based system that analyses a website automatically

from its URL, evaluates it using a rule-based score and a machine learning model, and suggests improvements in simple terms.

## II. LITERATURE SURVEY

Recent studies highlight the growing use of machine learning in SEO. Matošević et al. [1] classified web pages into low, medium and high classes according to their on-page SEO quality, with accuracy between 54.59% and 69.67%, and found the page title, meta description, H1 heading and body text to be important. Salminen et al. [2] predicted the rank of 733 pages using 30 factors, where XGBoost achieved an accuracy of 0.86 and links were among the most influential features. Su et al. [3] showed that the correct rank of seven out of ten pages can be predicted from a few factors, but the dominant factor, PageRank, is off-page and cannot be controlled by the owner.

Ghattas et al. [7] identified 59 attributes that affect web page performance and compared SVM, Logistic Regression and Random Forest, where SVM reached 89% and Random Forest 81% accuracy with feature selection; however, their attributes are technical, such as load time and page size, and not content-based SEO features. Drivas and Vraimaki [8] evaluated museum websites for accessibility, usability, SEO and speed, which shows the value of automated SEO measurement. Zhang and Dimitroff [4] reported that metadata implementation influences visibility, Ziakis et al. [5] listed 24 factors that affect Google search rank, and Zhu and Wu [6] used reverse engineering of highly ranked sites to find key factors.

Most of these studies identify ranking factors or build prediction models, but they do not provide a simple tool that analyses a website and gives practical recommendations. These studies collectively emphasize the need for a system that combines automatic crawling, transparent scoring, machine learning prediction and recommendations.

### III. PROPOSED METHODOLOGY

The proposed system integrates multiple layers to analyse the SEO performance of a website. The **frontend**, built with Flask templates using HTML and CSS, provides a simple form for entering the website URL and a result page for displaying the report. The **backend**, developed in Python with Flask, validates the URL and controls the analysis. The overall flow of the system is shown in Figure 1.

#### A. Crawling and Feature Extraction

The **crawler** uses the Requests library to download the home page and the internal pages of the same domain, and BeautifulSoup parses the HTML. From every page, six SEO features are extracted: **content length**, **keyword density**, **internal links**, **external links**, **meta description** and **image alt text**, and the page values are combined into website-level values.

#### B. SEO Score Calculation

The **scoring layer** applies predefined rules to the six features and calculates an SEO score out of 100, as shown in Table I, and then lists the strengths and weaknesses of the website.

TABLE I : SEO score rules

| SEO Feature      | Max Points | Best Range            |
|------------------|------------|-----------------------|
| Content length   | 20         | 600 words or more     |
| Keyword density  | 20         | 1% to 3%              |
| Internal links   | 15         | 10 or more            |
| External links   | 10         | 1 to 5                |
| Meta description | 20         | Present on all pages  |
| Image alt text   | 15         | 90% of images or more |
| Total            | 100        |                       |

#### C. Machine Learning Prediction and Storage

The **machine learning layer** uses a Random Forest classifier trained on an SEO dataset, which predicts whether the ranking-improvement outcome is likely to be positive. A **MySQL database** stores the analysed URL, the feature values, the score and the prediction.

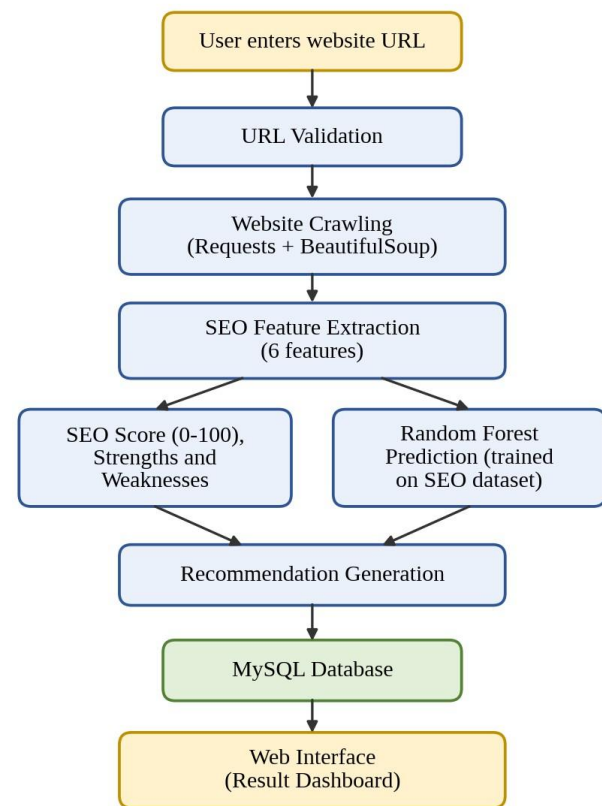


Figure 1 : Flow diagram of our proposed system

### IV. SYSTEM PROCESS FLOW

1. URL Input and Validation: The user enters the website URL, and the system checks that it has a valid scheme and domain name.
2. Website Crawling: The crawler downloads the home page and the internal pages of the same domain up to a fixed limit, skipping pages that fail.
3. SEO Feature Extraction: The six SEO features are measured on every crawled page using BeautifulSoup.
4. Feature Processing: Page-level values are averaged to form a website-level feature vector.
5. SEO Score Calculation: Predefined rules give a score out of 100, a performance level, and the strengths and weaknesses.
6. ML Prediction: The Random Forest model predicts whether the ranking-improvement outcome is likely to be positive.
7. Recommendation Generation: Each weakness is converted into a suggestion, such as adding relevant content or missing meta descriptions.
8. Result Storage and Visualization: The results are saved in MySQL and shown on the web interface as a report.

## V. RESULT AND DISCUSSIONS

The prototype system was tested with sample websites that differ in content, links and meta data. The crawler collected the accessible pages of each website, and the extracted feature values agreed with manual counts. Websites with rich content, more internal links, meta descriptions and image alt text received higher scores, whereas thin websites received lower scores and a list of weaknesses. Invalid URLs and unreachable websites were handled with clear messages.

The result page shows the SEO score, the performance level, the feature values, the strengths and weaknesses, the Random Forest prediction and the recommendations. User testing showed that the report is easy to understand, even for non-technical users, because every weakness is linked to a specific action. The score and prediction indicate the on-page SEO quality of a website but do not guarantee a search position, because ranking also depends on off-page factors. These results validate the effectiveness of combining rule-based scoring and machine learning in a web-based SEO tool.

## VI. CONCLUSION

This paper presents an ML based SEO performance analysis and recommendation system that crawls a website from its URL, extracts six on-page SEO features, calculates an SEO score out of 100, predicts the ranking-improvement outcome using a Random Forest model and provides recommendations. By using Python, Flask and MySQL, the system is simple, low-cost and suitable for beginners and small website owners. Future enhancements include broken-link detection, page-speed analysis, more SEO factors such as headings and page titles, training on a larger dataset, and AI-generated recommendations.

## VII. ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering, K.L.N. College of Engineering, Sivagangai, for the facilities and support provided for this work.

## REFERENCES

- [1] G. Matošević, J. Dobša and D. Mladenović, "Using machine learning for web page classification in search engine optimization," *Future Internet*, vol. 13, no. 1, Art. no. 9, 2021.
- [2] J. Salminen, J. Corporan, R. Marttila, T. Salenius and B. J. Jansen, "Using machine learning to predict ranking of webpages in the gift industry: Factors for search-engine optimization," in *Proc. 9th Int. Conf. Information Systems and Technologies*, Cairo, Egypt, 2019, pp. 1-8.
- [3] A.-J. Su, Y. C. Hu, A. Kuzmanovic and C.-K. Koh, "How to improve your search engine ranking: Myths and reality," *ACM Transactions on the Web*, vol. 8, pp. 1-25, 2014.
- [4] J. Zhang and A. Dimitroff, "The impact of metadata implementation on webpage visibility in search engine results (Part II)," *Information Processing & Management*, vol. 41, pp. 691-715, 2005.
- [5] C. Ziakis, M. Vlachopoulou, T. Kyrkoudis and M. Karagkiozidou, "Important factors for improving Google search rank," *Future Internet*, vol. 11, no. 2, Art. no. 32, 2019.
- [6] C. Zhu and G. Wu, "Research and analysis of search engine optimization factors based on reverse engineering," in *Proc. 3rd Int. Conf. Multimedia Information Networking and Security (MINES)*, Shanghai, China, 2011, pp. 225-228.
- [7] M. Ghattas, A. M. Mora and S. Odeh, "A novel approach for evaluating web page performance based on machine learning algorithms and optimization algorithms," *AI*, vol. 6, no. 2, Art. no. 19, Jan. 2025.
- [8] I. Drivas and E. Vraimaki, "Evaluating and enhancing museum websites: Unlocking insights for accessibility, usability, SEO, and speed," *Metrics*, vol. 2, no. 1, Art. no. 1, Jan. 2025.
- [9] L. Moreno and P. Martinez, "Overlapping factors in search engine optimization and web accessibility," *Online Information Review*, vol. 37, pp. 564-580, 2013.
- [10] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.
- [11] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.
- [12] Pallets Projects, "Flask documentation." [Online]. Available: <https://flask.palletsprojects.com/>
- [13] K. Reitz, "Requests: HTTP for humans." [Online]. Available: <https://requests.readthedocs.io/>
- [14] L. Richardson, "Beautiful Soup documentation." [Online]. Available: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- [15] Oracle Corporation, "MySQL reference manual." [Online]. Available: <https://dev.mysql.com/doc/>