International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I4P235

Volume 71 Issue 4, 403-415, April 2023
© 2023 Seventh Sense Research Group®

Original Article

Potential Web Content Identification and Classification System using NLP and Machine Learning Techniques

T. B. Lalitha[1], P. S. Sreeja[2]

[1,2]*Hindustan Institute of Technology and Science, Chennai, India.*

[1]*Corresponding Author : lalitha.srm@gmail.com*

Received: 10 January 2023 Revised: 17 April 2023 Accepted: 21 April 2023 Published: 25 April 2023

Abstract *- Nowadays, the volume of educational content on the world wide web is surging rapidly, challenging users with numerous options for e-Learning content in various areas of interest. This transition paves the way for web data mining and classification for identifying the most relevant content according to the user's interests and needs. Web mining is a technique to automatically track down and extract patterns from the data on WWW. The purpose of this paper was to analyze and classify web content based on keyword inputs resulting in a database facilitating a new way of data content recommendation for the users. The proposed work aims to scrape the freely accessible unstructured text content on the search engine and preprocess it to structured data using NLP methods. The extracted structured data undergoes an unsupervised learning algorithm for clustering them to obtain the three classified clustered sets of highly impacted, average, and low impacted data contents, which will be further stored in the database for the future recommendation of classified web content pages to the users.*

Keywords *- e-Learning, Web content mining, Unsupervised learning, NLP, k-means algorithm, Classification, PageRank algorithm.*

1. Introduction

The web page content classification is part of an information retrieval application that brings forth required advantageous information. This process aims to quickly categorize the web contents that yield relevant information from the vast world wide web [1]. This machine learning problem is getting crucial with the enormous rise of millions of websites gradually. In the recent two or three years, because of the epidemic situation, the world has undergone drastic changes in learning strategies. It almost depended on Online or Hybrid mode. So compared to the prior era, a lot and a lot of very useful informational content were shared over the web apart from the MOOC content, which is very cost-effective and, without time constraints, the users can learn accordingly. Many researchers and academicians are highly active in sharing their vast knowledge over the web, which can be useful for interested peers worldwide. So, to access these highly impacted freely accessible web contents from the extensive and scattered online content, we need some comparison and evaluation for the web contents that are mined and classify them according to the rank impact factor so that the users can be able to access the highly impacted web page contents according to their interests and needs.

For this, we start with web mining plays a vital part in locating useful pattern information or logs to help in classification. The web mining approach is an application of a data mining technique or algorithm and knowledge discovery process which focuses on extracting information patterns directly from the web data, which helps to enhance the search engine's capability in classifying the contents and predicting its user's behaviors [2]. Web mining extracts features for a purpose like news monitoring, cluster financial data, tracking price, consumer sentiment analysis, etc., from the various fields of websites like eCommerce, travel & tourism, marketing, research, social media sites, sports analytics etc. Web mining is majorly used for predicting user performance. Web mining is mainly divided into web content mining, web structure mining and web usage mining [3]. Web mining gathers information through structured, semi-structured and unstructured web content. The tools like Scrapy, beautiful soup, PageRank, Selenium, Apache logs etc., are used for web mining. The below (figure .1) represents the web mining taxonomy.

1.1. Web Content Mining

Web content mining is widely used in the education sector to retrieve valuable and relevant information from web documents or sites based on user interests and needs [4]. Web content is a collection of webpages which includes several types of data like text, images, videos, and animation connected through hyperlinks for navigation, then creating logfiles on every visit to that site [5]. There are different types of approaches for web content mining – structured mining, unstructured mining, semi-structured mining, and multimedia mining.

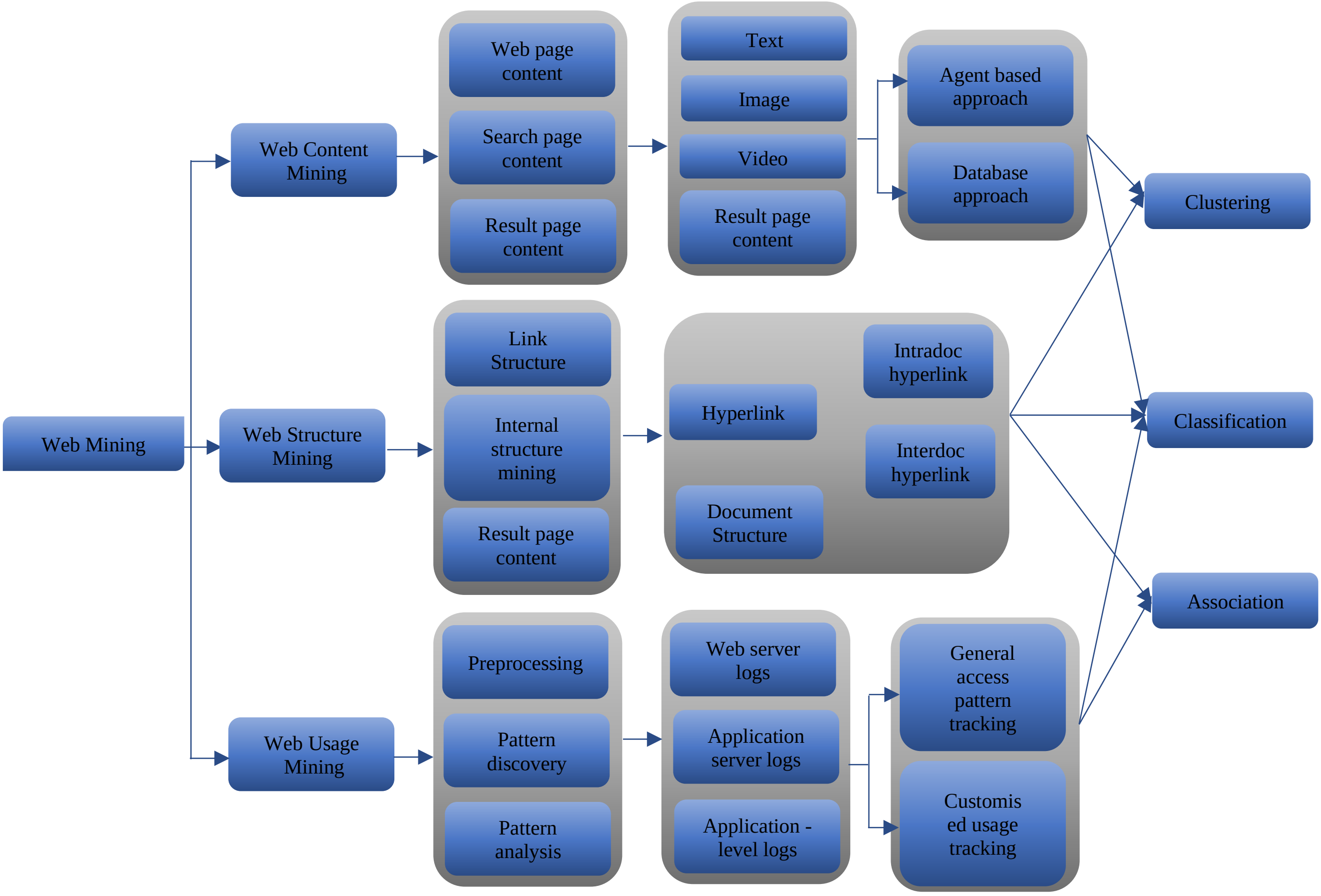


Fig. 1 Taxonomy of web mining

Website pages, Unstructured Data
Web data scraping
Structured Data (CSV, XSL, SQL, XML)

Fig. 2 Mining of web contents

The mining of text documents is a form of unstructured mining involving text, images and clusters of the web page based on input keywords or content with the help of machine learning and natural language processing (NLP), known as text mining [6]. The below (figure .2.) depicts the mining process of web contents that are scraped and stored in the database.

Web content mining has two approaches: agent-based and data-based approaches.

- The agent-based approach focuses on enhancing information detection and filtering. The three categories involved are information filtering/categorization, intelligent search agents, and personalized web agents.
- Data-based approaches are utilized to classify semi-structured data existing over the web as structured data.

1.2. Web Structure Mining

Web structure mining [7], [8] is a process of finding structure information as a structured summary of the specific website, like web pages and hyperlinks as nodes and edges from the web. The discovered web structure information identifies the relationship between direct link connections or the information linked by web pages [53]. Web structure mining can be categorized into two types extracting the patterns from the hyperlinks and mining the data structure. Web structure mining is mainly useful to determine the links between two commercial websites.

1.3. Web Usage Mining

Web usage mining [4], [9] is an application for extracting different varieties of interesting data patterns and useful information from the logs obtained from browsing the web server that is readily accessible on the web. It assists in analyzing the log files of users created when users previously interacted with web content [37]. The web server and application server logs are the prominent source of data collected. The three major types of log data are categorized into Server-side, Client-site, and Proxy-side. Then the other sources of data collected are cookies, demographics, etc. The data normally includes user profiles, browser logs, IP addresses, proxy server logs, past records of events, etc.

Up to limited knowledge gained from the survey of various works done by many researchers, the web usage mining is vastly concentrated on classifying contents and recommendations by majorly retrieving the users' weblogs. Whereas web content mining is used scantily, the mere focus on web content is very rare, so the study was chosen to work in this area.

In this work, we use the web content mining technique for scrapping the text contents using an agent-based and database approach. Then the scraped contents are preprocessed as unstructured text contents to obtain the clean scarped data. Then the data will be stored in the local database in the format of csv. Files or Excel. Using the machine learning algorithm, these data contents are clustered and classified into high, medium, and low-impacted contents.

This paper mainly covers the study of the proposed web content classification model for web content mining of unstructured text content using NLP and machine learning techniques and its respective results. The structure of the paper consists of existing literature research works in the section. II, then the internal process of the framework is explained in the section. III, whereas the experimental results and process work are discussed in the section. IV, and section. V as the conclusion of the work.

2. Background and Related Works

The web content mining approach is to mine the data contents from websites or web pages, which largely consists of text, tables, graphics, images, audio, videos, data records and blocks, ranging from unstructured data to semi-structured data, which are typically in any one of the forms of HTML, XML, XHMTL content pages [44] [51]. The web content mining approach is also called as agent base approach, where it has three types of agents, namely intelligent search engines [48], personalized web agents and information or data filtering. Unstructured content mining is also known as knowledge discovery in text. It carries out the functionalities like text detection, mining and preprocessing of the resulting contents using the techniques like NLP, text categorization or information retrieval [54]. The results that are extracted will be

- Units
- Topic Tracking
- Summarization
- Web page categorization
- Web site categorization
- Web page clustering
- Web site clustering
- Web object clustering
- Information Virtualization

2.1. Challenges in Web Content Mining

The major challenges or problems that arise in web content mining [39] when extracting information from web search engines are given as

- Data extraction [46]: It involves obtaining structured data from web pages, namely search results and product services. Machine learning and automatic data extraction techniques are used to solve this issue.
- Web Information Integration and Schema Matching [42]: World Wide Web incorporates a huge amount of data of the similar type of information in different ways for each website/ page. The major problem is identifying semantically similar data and matching it to the query.
- Opinion extraction from online sources [43] (surveys, chats, forums, etc.)
- Knowledge synthesis
- Segmenting Web pages and detecting noise [45]: the problem is to automatically extract the segment of the main content page without advertisements, copyright notices, or navigation links.

Web content mining is an effort of multi-disciplinary techniques which is drawn out from various fields like informational retrieval, NLP, Machine learning, statistics, etc., for better, specific, and accurate extraction of results. The scraped web contents undergo data cleaning then processed data will be clustered and classified by using machine learning algorithms to get highly impacted contents. The Web page classification is the procedure of allocating a web content page to different predefined datapoint classifications, which plays an important part in analysing the web's topical structure, scraping, crawling, topic-specific weblink analysis, and cultivating web directories, etc., [10]. Here, we discuss some related works of web page content classification below.

Kenan Enes Aydin et al. [49] have proposed a model to classify website contents to solve the complicated issue of parental control for children to use safe, secure internet. Dynamic classification of websites is modelled using natural language processing, machine learning techniques like support vector machines and text categorization. The classification success rate reached 0.8756 for the results obtained using SVM. Cavalieri. A et al. [50] have presented an intelligent system aimed at supporting political science scholars by facilitating the automatic categorization of specific political documents containing the parliamentary questions, which are collected at the Chamber of Deputies of the Italian Republic during the weekly Question Times using machine learning and deep learning text classifications techniques.

Utiu. N et al. [12] have proposed a framework for separating the textual content from the whole web page while excluding the template and other decorative elements. The process uses Machine learning classification techniques over renowned datasets like the Cleaneval dataset and Dragnet dataset to get the main contents of the web pages and obtained 0.96 as an f1 score in performance evaluation without going for exhaustive preprocessing procedures. Karthikeyan. T et al. [11] have proposed a model for the classification of documents that are extracted with a better accuracy rate using effective web scraping techniques and machine learning methods for recursive feature elimination to select better feature subsets.

Ali. F et al. [13] proposed a work using fuzzy ontology-based semantic knowledge and an SVM classifier to filter and differentiate whether the URL is adult, normal or medical and then detect and block pornographic content automatically. Dimitrovski. A et al. [14] framed a work to classify the educational content by mapping course descriptions based on academic topics and disciplines using information retrieval techniques over the data like CIP and Wikipedia for better reach of student requests.

Shinde. S et al. [15] intended work to classify web documents using a machine learning technique as a support vector machine classifier for the available data contents related to freelancing and remote job over the search engine. Dilip Patel. A et al. [52] proposed a method for article categorization by identifying the category of the article and its hierarchical file structure, thus helping the users to access the appropriate article content and to locate the related article titles. The classification of web pages to form catalog generation for child content and parenting context content is available over the web.

Jiménez. L. R. [17] proposed a content classification method for grouping web classes into 6 different classes based on an unsupervised learning clustering technique as a K-means algorithm. Where the dataset generated based on Multipurpose Internet Mail Extensions (MIME) content breakdown and the external subdomain connections, through PC running Webpage Test tool are used to obtain the various grouped contents. Deng. L et al. [18] presented a web page classification method based on heterogeneous textual features and tree-like structural features, which are further converted to vectors and then fused, are used to propose combined multiple-classifiers. From there, the accuracy of the web page classification has been increased using the combined multiple classifiers over the DMOZ dataset, Amazon dataset, and 7-web-genres dataset.

3. Proposed Model and Methodology

The proposed work describes the experimental study on web content mining and classifying unstructured web content or pages using machine learning techniques. Overall, the model developed within the scope of this study is designed on text mining the web contents freely available over the web in the field of computer science. Focusing on one language course as "Core java programming" and its prescribed syllabus content that is information collected through academic course content and MOOC syllabus contents. Thus, further aimed to expand the territory to many more subjects for classifying in the later stage or future process.

The modular architecture of web content classification is given in (figure 3.) where the feature extraction of e-Learning contents over the internet henceforth provides the classified levels of impact clusters involved in the specified web contents [19].

In this data collection phase, the web scraping process is used to get the required web content documents from the search engine site. Web Scraping is also called web harvesting or extraction. It is a specialized and automatic tool to acquire or extract large amounts of accurate data from web pages [20]. The web scraper uses bots to obtain content from websites directly accessing the world wide web by utilizing the hypertext transfer protocol or a web browser. The retrieved data is further exported into a comfortable format such as JSON, CSV or Excel spreadsheet etc. The workflow model of the web scraping is represented in (figure .4) given below.

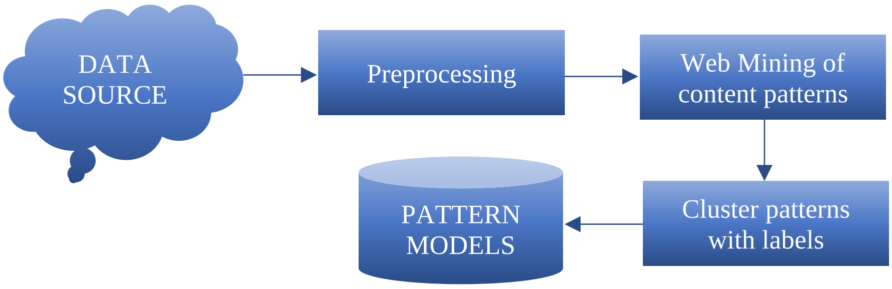


Fig. 3 Modular architecture of Classification (Source: [19])

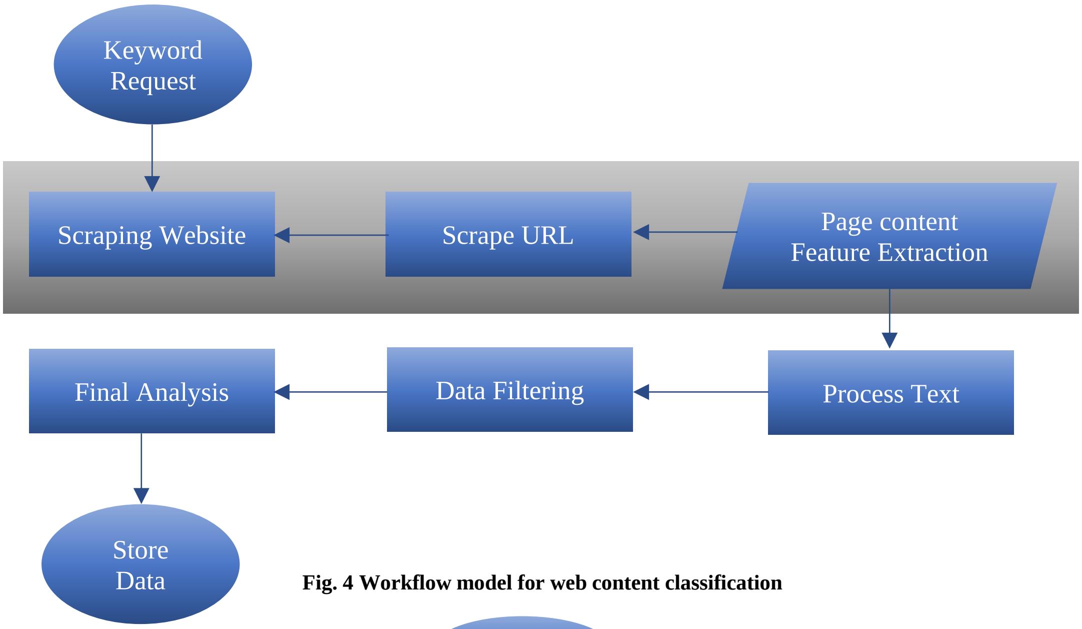


Fig. 4 Workflow model for web content classification

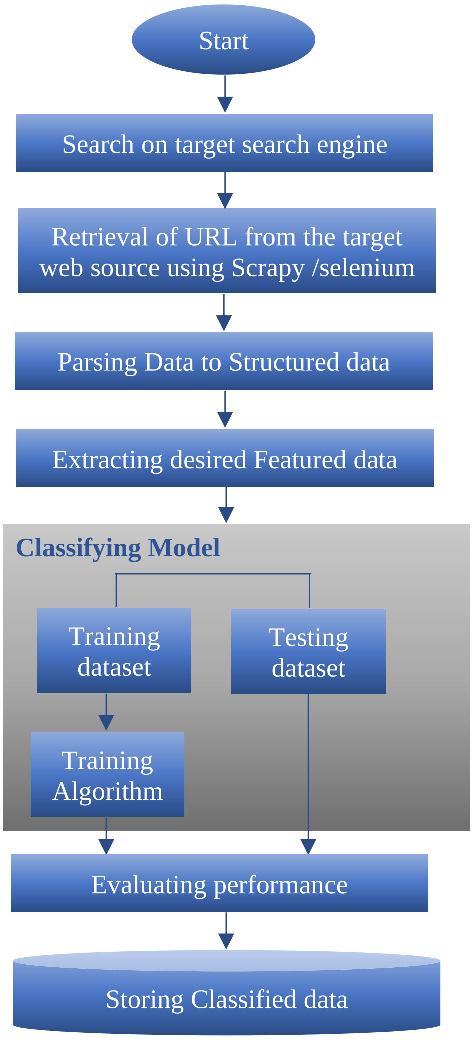

Start
Search on target search engine
Retrieval of URL from the target web source using Scrapy /selenium
Parsing Data to Structured data
Extracting desired Featured data
Classifying Model
Training dataset
Testing dataset
Training Algorithm
Evaluating performance
Storing Classified data

Fig. 5 Flow chart of the classification model

Web scraping includes three major steps, which consist of scraping the website on a search engine, scraping URL, and Page content [21]. First, the input keyword is sent to the search engine as a GET request. Here around 220 input keywords are created with human intervention covering multiple prescribed and well know Java textbook keywords and referring to MOOC Coursera online syllabus content. As requested, sending the input dataset to the server will obtain the output web page content as a response. Then, the website's HTML code will be parsed, and a tree structure path will be formed. In the end, the parse tree, which is an ordered tree representing the structured syntactic data, is searched using the Python library [22]. Selenium is used to scrape the web data content from the targeted search engine [23], [24] Google. There are several search engines in the world, and a few of them, namely Google, Bing, Baidu, Yahoo!, Yandex, Ask, AOL, DuckDuckGo, Excite, Naver, Seznam, Ecosia, Lycos, etc. According to the worldwide search engine market share, the most popular search engine is Google, with a whopping 92.26% share compared to others. So Google is considered for this work.

The flow chart of the web content classification model is given in (figure 5.) below. Where the whole process represents starts from scrapping the web for the web contents, then preprocessing the retrieved data passing through the classifying model algorithms for the final desired clustered dataset sent into the database storage section.

The Google search engine is taken as the target search engine for this work, and then the web scraping using Selenium acquires and retrieves the desired URL from the target web source. The retrieved unstructured data are parsed to get structured data which can be used further to extract features from the web contents. The retrieved feature data is taken for training and testing data. Where the split percentage of the dataset is sent for training as a training dataset, and the rest is used for testing as a testing dataset. This classifying model determines the impact of the collected web contents and clusters according to the level of impact of those contents. Three classes of rating the content pages are 'high impact', 'average impact' and 'low impact'. Then by evaluating the performance, the classified web contents are stored in the database for the future recommendation process.

This process involves web scraping of contents over the search engine for the given input keyword, then searching and scraping all the results (website URLs, title, the word count of keyword, ranking, the total number of words, etc.) for the keyword. The preferred clusters are formed and stored as the classified contents in the database using the results obtained through scarping.

The web scraping procedure uses Selenium which automates the search engines. Selenium is an open-source automated tool which provides a single interface to write test coding scripts using the programming languages [22]. Here python libraries and applet viewers are used to build the automated tool, which acts as an automated web bot. Selenium uses Chrome driver [25], where the keywords are sent to the Google search engine by searching and opening the URL, then gets and loads the search results into the csv file. The process of an automated tool for searching and loading the results for each keyword into separate csv files.

To obtain the desired results, the contents are scarped from Google for the given keyword primarily based on subjectivity classification of sentiment analysis for collecting relevant pages [26]. Subjectivity classification is the process of differentiating the contents, whether objective or subjective [27]. Here, subjectivity refers to the quality and the relevancy of the pages collected for the given keyword.

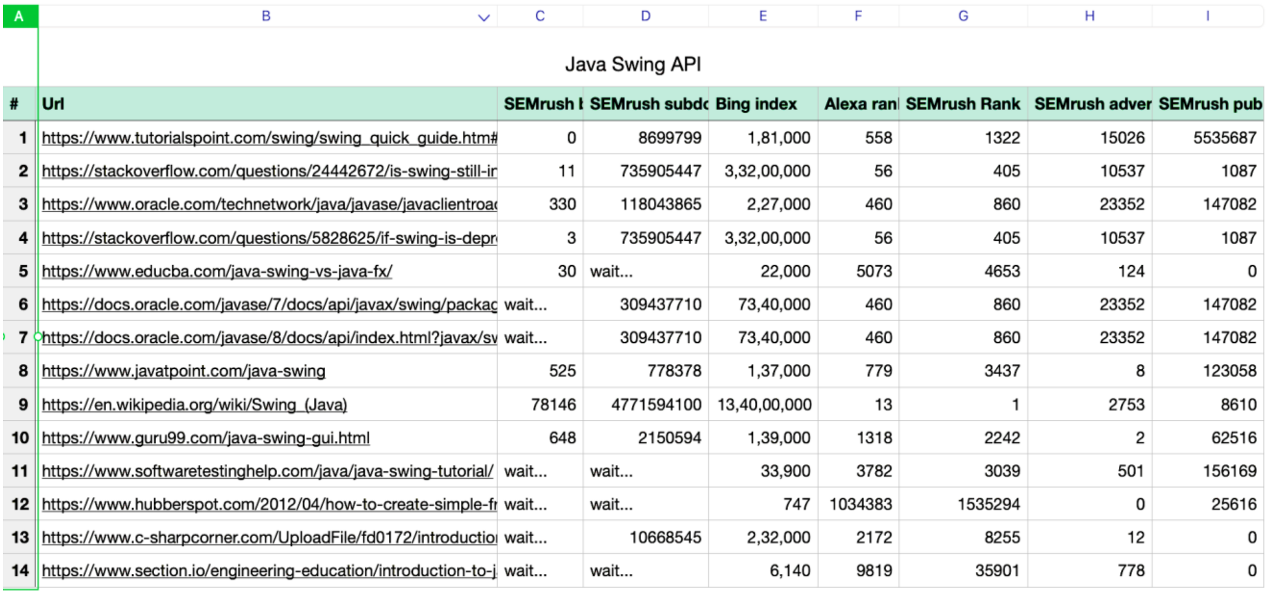
- 0 – page content is not subjective at all, which is written objectively.
- 1 – Page content is fully subjective and is written like an opinion.

Algorithm of Web Scraper:

```
Step 1:  Input user query keyword
Step 2:  Get request website_URL
Step 3:  Select the web contents using Selenium
Step 4:  Scrape Website information from a given URL
Step 5:  Scrape Unstructured page content feature
         extraction from a given URL.
Step 6:  Parse and extract information text from the web
         document
Step 7:  Filter the scraped data by the specific query
         keywords
Step 8:  Save the scraped data into a structured data format
         as .CSV file
Step 9:  Consolidate all the scraped .CSV files into one
         cumulative .CSV file
Step 10: Store the cumulative .CSV file (scraped data) in
         the local database.
Step 11: End process.
```

Then, the data like the URL of the page, published date, the word count of the keyword on that page content, the total number of words on the page, google ranking of the page, title, and text content are scraped for the searching keyword and certain data like google ranking, the word count of keyword, the total number of words are further processed for finding the appropriate cluster of impact groups of the obtain data contents.

The below (figure. 6)shows the extraction of the web contents that have been identified and abstracted details like URL, various rank lists in the different search engines, for the specific keyword that was provided through coding script, and then the extracted data have been stored into excel sheet as shown above. Thus, the high-level diagram of classified results after processing will be given in (figure 7.) below.



Java Swing API

#	Url	SEMrush I	SEMrush subdc	Bing index	Alexa ran	SEMrush Rank	SEMrush adver	SEMrush pub
1	https://www.tutorialspoint.com/swing/swing_quick_guide.htm#	0	8699799	1,81,000	558	1322	15026	5535687
2	https://stackoverflow.com/questions/24442672/is-swing-still-ir	11	735905447	3,32,00,000	56	405	10537	1087
3	https://www.oracle.com/technetwork/java/javase/javaclientroa	330	118043865	2,27,000	460	860	23352	147082
4	https://stackoverflow.com/questions/5828625/if-swing-is-depr	3	735905447	3,32,00,000	56	405	10537	1087
5	https://www.educba.com/java-swing-vs-java-fx/	30	wait...	22,000	5073	4653	124	0
6	https://docs.oracle.com/javase/7/docs/api/javax/swing/packag	wait...	309437710	73,40,000	460	860	23352	147082
7	https://docs.oracle.com/javase/8/docs/api/index.html?javax/sv	wait...	309437710	73,40,000	460	860	23352	147082
8	https://www.javatpoint.com/java-swing	525	778378	1,37,000	779	3437	8	123058
9	https://en.wikipedia.org/wiki/Swing_(Java)	78146	4771594100	13,40,00,000	13	1	2753	8610
10	https://www.guru99.com/java-swing-gui.html	648	2150594	1,39,000	1318	2242	2	62516
11	https://www.softwaretestinghelp.com/java/java-swing-tutorial/	wait...	wait...	33,900	3782	3039	501	156169
12	https://www.hubberspot.com/2012/04/how-to-create-simple-fr	wait...	wait...	747	1034383	1535294	0	25616
13	https://www.c-sharpcorner.com/UploadFile/fd0172/introductio	wait...	10668545	2,32,000	2172	8255	12	0
14	https://www.section.io/engineering-education/introduction-to-j	wait...	wait...	6,140	9819	35901	778	0

Fig. 6 Screenshot of data contents abstracted into Excel

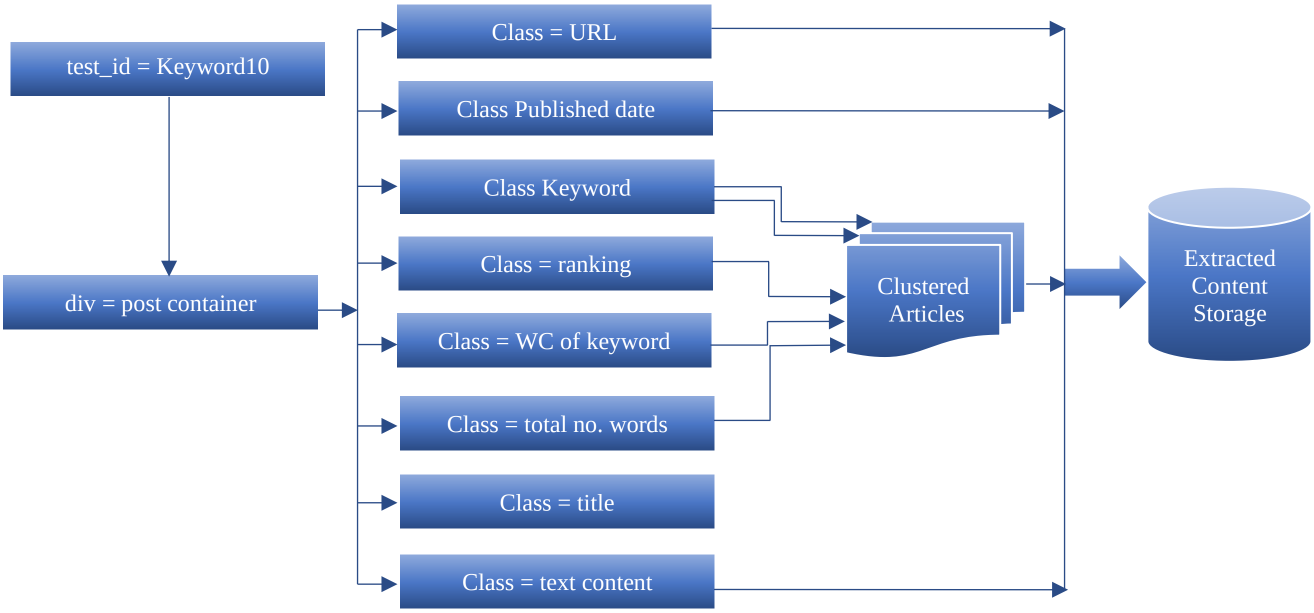


Fig. 7 Diagram of classified content result

The Google listing or ranking of a page is obtained directly from the search engine Google list position of the web content page [28]. The Google listing uses the algorithm called PageRank algorithm for Google, which is used to rank web pages in their results by counting the number of links and link quality of the content page to measure the estimated value for regulating the importance of the web content page [29]. The page rank of A is calculated as in equation (1) below.

$$PR(A) \; = (1-d) + d(PR(T_1)/C(T_1) + \cdots + \; PR(T_n)/C(T_n)) \quad (1)$$

Where d is the damping factor, T1 to Tn are pages, C(T1…Tn) is the number of links to the page. This determines the process of the most important web pages that are likely to get more links from other pages.

Here Google's link position is taken from 1 to 15 links of the web content pages and then fetches the content listings from Google as a result of obtaining the Google listing rank data. Thus, for each search keyword, a minimum of 10 to 15 URL links and positions are taken as data, forming a dataset of around 3000 samples.

The fastest way to acquire the desired materials is through uniquely finding the associated keywords. So that the challenges like bulk search volume and keyword difficulty can be driven out or minimized using the most appropriate informational keywords to target the contents [40], the high volume of search content can be controlled. To get the most relevant content using the long-tail informational keywords [41], which reduces the bulk traffic of any irrelevant content gathering when commonly used keywords are used as query search. Thus, the long-tail keywords have low conflict in targeting and acquiring the main topic while searching. The below (figure. 8) shows how the several relevant search contents vary with the detailing of keywords like long-tail keywords to target more specific suggestions.

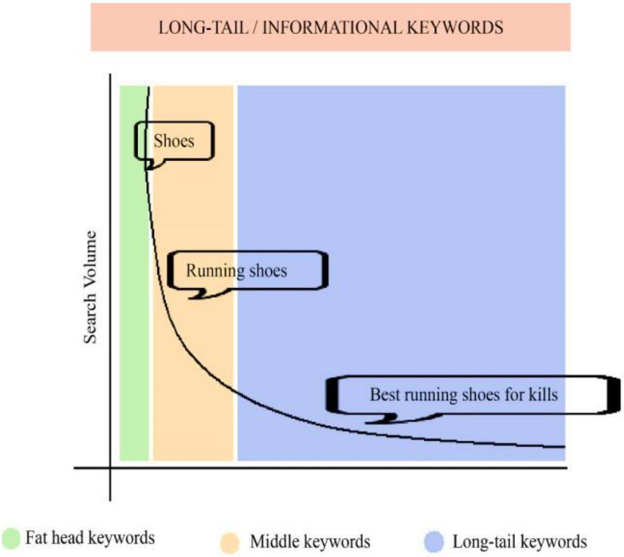


Fig. 8 Long-Tail Keywords

Since this leads to having a unique search keyword as a key phrase, it becomes an effective way of optimizing for informational queries and further strengthening relevant topics by giving out smaller search volumes with accurate content to those who are looking for very specific materials than by just browsing all around the related terms.

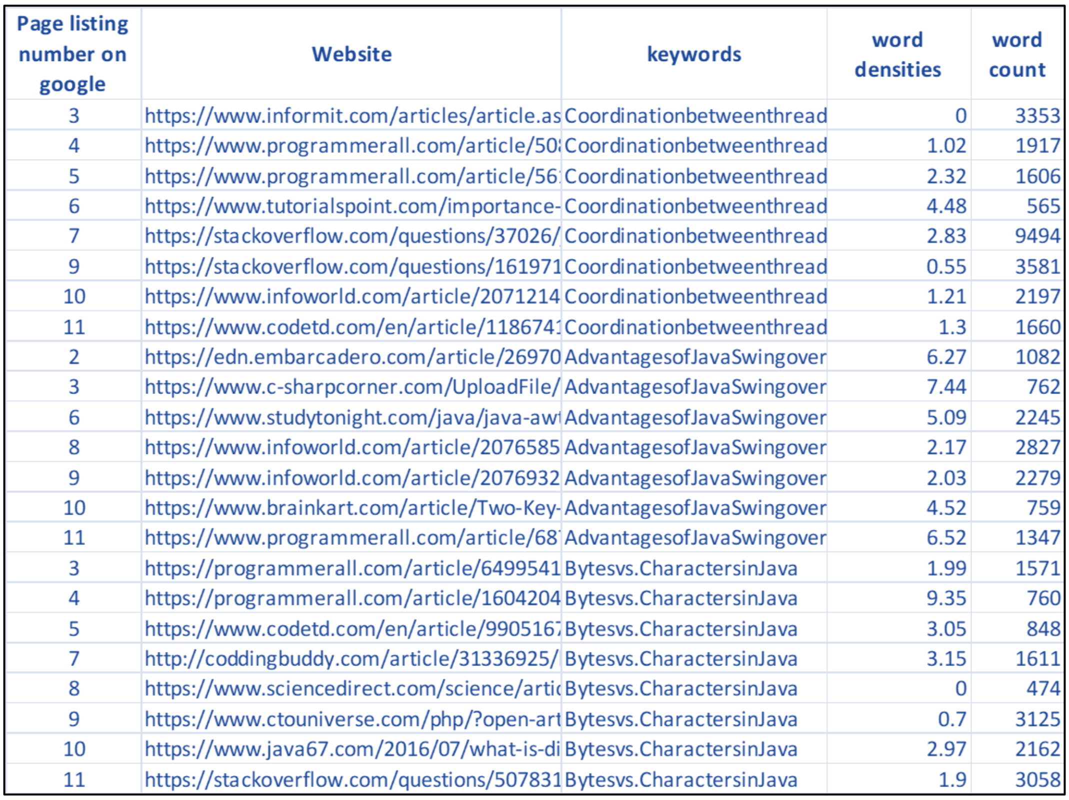
The word count of keywords on the page and the total number of words on the page data are gathered using the Python Split() function, where the word count represents the keyword's number of occurrences on that specific page content. Then, we will calculate the keyword density by using the word count of the keyword and the total number of words data. Keyword density refers to the percentage obtained through a simple division of the number of times the keywords occurred by the total number of words on that content [30]—the keyword density used to improve the webpage visibility on search engines [31]. The keyword density formula is given in equation (2) below.

$$Keyword\ Density = (NK_r/TK_n) * 100 \tag{2}$$

Where Nkr is the word count of keywords on the page, and TKn is the total number of words on the page. Here the keyword density is calculated by taking all the words in the search word as long-tail keywords into account, excluding the prepositions and then by finding the respective densities of each word and finally summing them up to get the destined keyword density percentage.

Ex: The Search word is "Memory management in Java."

Where the word densities of the words' Memory', 'java', 'management' are taken respectively, excluding the preposition 'in'. Then all the values are summed up to get the final value. That final value is taken as keyword density. The retrieved data are cumulated and processed to form a single dataset, then stored in the database as an Excel file which is shown below (figure. 9)



Page listing number on google	Website	keywords	word densities	word count
3	https://www.informit.com/articles/article.as	Coordinationbetweenthread	0	3353
4	https://www.programmerall.com/article/50	Coordinationbetweenthread	1.02	1917
5	https://www.programmerall.com/article/56	Coordinationbetweenthread	2.32	1606
6	https://www.tutorialspoint.com/importance-	Coordinationbetweenthread	4.48	565
7	https://stackoverflow.com/questions/37026/	Coordinationbetweenthread	2.83	9494
9	https://stackoverflow.com/questions/161971	Coordinationbetweenthread	0.55	3581
10	https://www.infoworld.com/article/2071214	Coordinationbetweenthread	1.21	2197
11	https://www.codetd.com/en/article/118674	Coordinationbetweenthread	1.3	1660
2	https://edn.embarcadero.com/article/26970	AdvantagesofJavaSwingover	6.27	1082
3	https://www.c-sharpcorner.com/UploadFile/	AdvantagesofJavaSwingover	7.44	762
6	https://www.studytonight.com/java/java-aw	AdvantagesofJavaSwingover	5.09	2245
8	https://www.infoworld.com/article/2076585	AdvantagesofJavaSwingover	2.17	2827
9	https://www.infoworld.com/article/2076932	AdvantagesofJavaSwingover	2.03	2279
10	https://www.brainkart.com/article/Two-Key-	AdvantagesofJavaSwingover	4.52	759
11	https://www.programmerall.com/article/68	AdvantagesofJavaSwingover	6.52	1347
3	https://programmerall.com/article/6499541	Bytesvs.CharactersinJava	1.99	1571
4	https://programmerall.com/article/1604204	Bytesvs.CharactersinJava	9.35	760
5	https://www.codetd.com/en/article/990516	Bytesvs.CharactersinJava	3.05	848
7	http://coddingbuddy.com/article/31336925/	Bytesvs.CharactersinJava	3.15	1611
8	https://www.sciencedirect.com/science/arti	Bytesvs.CharactersinJava	0	474
9	https://www.ctouniverse.com/php/?open-art	Bytesvs.CharactersinJava	0.7	3125
10	https://www.java67.com/2016/07/what-is-di	Bytesvs.CharactersinJava	2.97	2162
11	https://stackoverflow.com/questions/507831	Bytesvs.CharactersinJava	1.9	3058

Fig. 9 Screenshot of the dataset

4. Results and Discussion

Through the obtained data, try to find the level of impact of the collected content pages. There are no preassigned labels for the datasets, so the unsupervised learning algorithm is used to classify data contents. Clustering is an exploratory data analysis technique that plays a vital role in unsupervised learning classification, where the assumed data points are clustered with respect to the similarity of those data to form various groups [32]. Numerous unsupervised learning algorithm prevails to work out the clustering or classifying issues in machine learning or data science that actively functions along with the Python implementation. So, at present many machine learning algorithms are available for clustering techniques, in that for the initial process, the k-means algorithm is taken for clustering the data points.

The k-means algorithm is a partitioning approach where all the data points are formed into different fixed clusters based on similarities [33]. This algorithm is applied for unlabeled quantitative data variables [34]. Where the unlabeled dataset is clustered or grouped into different collections of clusters. In K-means, the K refers to the number of fixed clusters that are to be produced in the process,

For ex., if K=3, then there will be 3 clusters

if K=n, then there will be n clusters

Where n is the number of clusters created.

K-means clustering is an iterative algorithm that splits the unlabeled dataset into K distinct clusters, where each cluster consists of similar properties of the dataset belonging to one group. K-means clustering is a centroid-based algorithm where it aims in such a way to reduce the sum of distances between the specific clusters and their data points so that each cluster is connected to the centroid. The main task performed by the K-means algorithm is

- Finds out the best value for K using an iterative process.
- Data points are allocated to the nearest K-centroid, thus creating a classified cluster.

Algorithm of K-mean Clustering

```
Step 1: Input
        D = {$t_1$, $t_2$, …., $t_n$} // set of data points
        K is no. of clusters
Step 2: Assign initial values for the model as $m_1$, $m_2$, … $m_k$
Step 3: Randomly choose k data points from D as initial centroids
Step 4: Repeat
        Assign each data $d_i$ to the nearest cluster centroid and calculate the new mean for each cluster.
Step 5: Re-assign each data point in the dataset to the nearest cluster using Euclidean distance as a measure of distance until the convergence criteria are met.
Step 6: Output
        Set of K clusters.
```

For this process, 70% of the dataset is taken as a training dataset. Then, the two-dimensional space is created using Google listing number as X-axis and Word density sum as Y-axis. The data points are clustered based on three types of meaningful intuition assumptions made for grouping the data, fixing the exact number of groups, and getting the clusters' centroid through continuous iterations. The three datapoint cluster assumption is taken as below.

- Data points with low word density and yet one of the first links in the search.
- Data points with high word density and yet somehow are in the middle of the search.
- Data points with low word density and yet are at the bottom of the search page.

Based on the assumed data points, the resulting clusters diagram for the data points is shown in (figure .10) below.

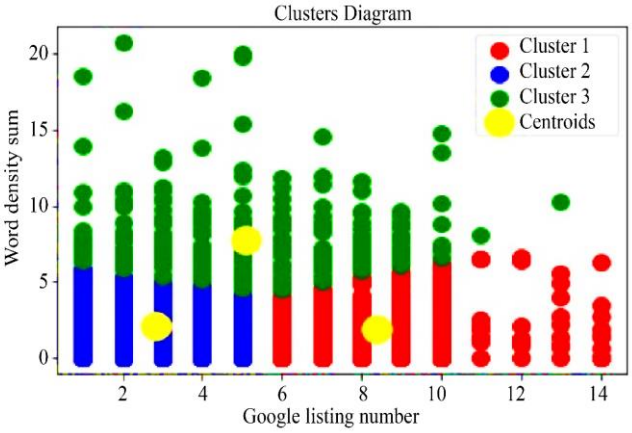


Fig. 10 K-means cluster diagram

The three clusters show the level of impact of the contents based on the data points meaningful intuition, where

- blue-cluster2 gives out a high-level impact of content.
- green-cluster3 gives out the average level of impact content
- red-cluster1 gives out a low level of impact content.

The Elbow method is a metric used to evaluate the K-means clusters and find the optimal number of groups. The elbow method uses the within-cluster sum of squares (WCSS) value to determine the overall variations inside a cluster. The formula to calculate WCSS value is given in equation (3) below,

$$WCSS = \sum_{P_{i\ in\ cluster\ 1}} distance(P_i\ C_1)^2 + \sum_{P_{i\ in\ cluster\ 2}} distance(P_i\ C_2)^2 + \cdots + \sum_{P_{i\ in\ cluster\ n}} distance(P_i\ C_n)^2 \quad (3)$$

Where $P_{i\ in\ cluster\ 1}$, is the data points within cluster1. C$_i$ is the centroid of the cluster. $\sum_{P_{i\ in\ cluster\ 1}} distance(P_i\ C_1)^2$ is the sum of the square distance between the centroid and their respective data points inside cluster 1

The WCSS is calculated to measure the average squared distance of all data points within the cluster to the centroid [35]. The Euclidian distance or Manhattan distance is used to measure the values to determine the distance between the centroid and the data points. The formula to calculate the Euclidian distance of two-dimensional space is given in equation (4) below.

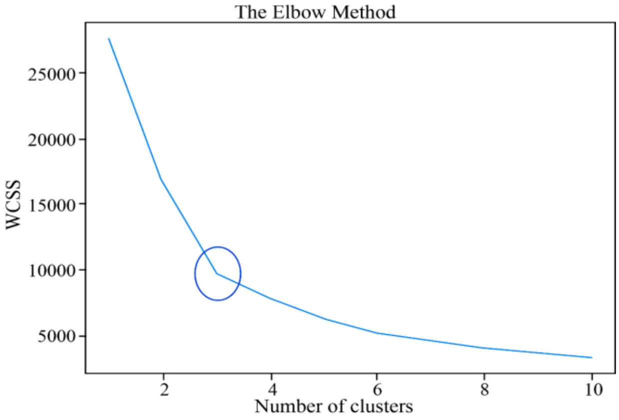


Fig. 11 Elbow Method of Clustering

$$\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \quad (4)$$

The elbow method is used to get the optimal K value of the k-mean algorithm, representing the number of clusters that can choose to fix the required groups [36]. The above (figure .11) shows the dataset's exact number of clusters.

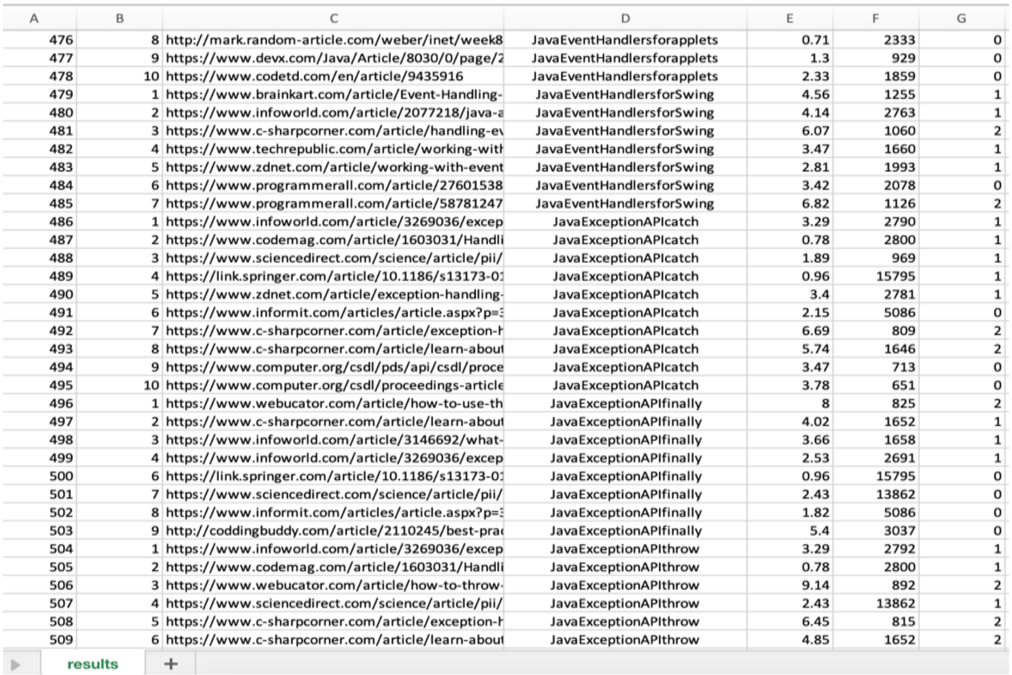
The graph depicts the sharp bend, which appears to be an elbow; that sharp point of bend or knee of the curve determines the best value of K. By implementing the Python line of codes and executing it, the data extracted are independent variables dataset in the excel database, will look like above (figure. 10.) Thus, the knee of the curve shows the exact clustering group as 3 as the cutoff point.

The data clustered through the above method are processed to form the final desired result and then stored in the database as an Excel file. Here are the details of the obtained and stored headers, given below in Table 1.

Thus, finally processing, the extracted data for every keyword were combined into a single dataset are stored in the Excel database as the clustered and classified page contents. The below (figure. 12) shows the final dataset acquired through the process.

Table. 1 Details of the stored data heading in the database

Columns	Headers
A	Number of data
B	Google rank list
C	URL
D	Keyword
E	Keyword density
F	Total Number of words
G	Cluster group



A	B	C	D	E	F	G
476	8	http://mark.random-article.com/weber/inet/week8	JavaEventHandlersforapplets	0.71	2333	0
477	9	https://www.devx.com/Java/Article/8030/0/page/2	JavaEventHandlersforapplets	1.3	929	0
478	10	https://www.codetd.com/en/article/9435916	JavaEventHandlersforapplets	2.33	1859	0
479	1	https://www.brainkart.com/article/Event-Handling-	JavaEventHandlersforSwing	4.56	1255	1
480	2	https://www.infoworld.com/article/2077218/java-a	JavaEventHandlersforSwing	4.14	2763	1
481	3	https://www.c-sharpcorner.com/article/handling-ev	JavaEventHandlersforSwing	6.07	1060	2
482	4	https://www.techrepublic.com/article/working-witl	JavaEventHandlersforSwing	3.47	1660	1
483	5	https://www.zdnet.com/article/working-with-event	JavaEventHandlersforSwing	2.81	1993	1
484	6	https://www.programmerall.com/article/27601538	JavaEventHandlersforSwing	3.42	2078	0
485	7	https://www.programmerall.com/article/58781247	JavaEventHandlersforSwing	6.82	1126	2
486	1	https://www.infoworld.com/article/3269036/excep	JavaExceptionAPIcatch	3.29	2790	1
487	2	https://www.codemag.com/article/1603031/Handli	JavaExceptionAPIcatch	0.78	2800	1
488	3	https://www.sciencedirect.com/science/article/pii/	JavaExceptionAPIcatch	1.89	969	1
489	4	https://link.springer.com/article/10.1186/s13173-01	JavaExceptionAPIcatch	0.96	15795	1
490	5	https://www.zdnet.com/article/exception-handling-	JavaExceptionAPIcatch	3.4	2781	1
491	6	https://www.informit.com/articles/article.aspx?p=3	JavaExceptionAPIcatch	2.15	5086	0
492	7	https://www.c-sharpcorner.com/article/exception-h	JavaExceptionAPIcatch	6.69	809	2
493	8	https://www.c-sharpcorner.com/article/learn-about	JavaExceptionAPIcatch	5.74	1646	2
494	9	https://www.computer.org/csdl/pds/api/csdl/proce	JavaExceptionAPIcatch	3.47	713	0
495	10	https://www.computer.org/csdl/proceedings-article	JavaExceptionAPIcatch	3.78	651	0
496	1	https://www.webucator.com/article/how-to-use-th	JavaExceptionAPIfinally	8	825	2
497	2	https://www.c-sharpcorner.com/article/learn-about	JavaExceptionAPIfinally	4.02	1652	1
498	3	https://www.infoworld.com/article/3146692/what-	JavaExceptionAPIfinally	3.66	1658	1
499	4	https://www.infoworld.com/article/3269036/excep	JavaExceptionAPIfinally	2.53	2691	1
500	6	https://link.springer.com/article/10.1186/s13173-01	JavaExceptionAPIfinally	0.96	15795	0
501	7	https://www.sciencedirect.com/science/article/pii/	JavaExceptionAPIfinally	2.43	13862	0
502	8	https://www.informit.com/articles/article.aspx?p=3	JavaExceptionAPIfinally	1.82	5086	0
503	9	http://coddingbuddy.com/article/2110245/best-prac	JavaExceptionAPIfinally	5.4	3037	0
504	1	https://www.infoworld.com/article/3269036/excep	JavaExceptionAPIthrow	3.29	2792	1
505	2	https://www.codemag.com/article/1603031/Handli	JavaExceptionAPIthrow	0.78	2800	1
506	3	https://www.webucator.com/article/how-to-throw-	JavaExceptionAPIthrow	9.14	892	2
507	4	https://www.sciencedirect.com/science/article/pii/	JavaExceptionAPIthrow	2.43	13862	1
508	5	https://www.c-sharpcorner.com/article/exception-h	JavaExceptionAPIthrow	6.45	815	2
509	6	https://www.c-sharpcorner.com/article/learn-about	JavaExceptionAPIthrow	4.85	1652	2

Fig. 12 Screenshot of Final Result of clustered contents

5. Conclusion

The proportion of educational content accessible on the internet is increasing, which affects the users to find a very relevant option for retrieving content from a huge collection of data. A web content classification system is proposed to optimize web pages and get the most quality impacted web content to the users. In this work, the web scraping application uses Selenium tools and Python libraries to identify the relevant web pages for the given 220 keywords of the specific language. Then, the machine learning algorithm is approached to spot the degree of impact of the web content. The data obtained through scraping are unlabeled, so an unsupervised learning algorithm is used to classify contents. The clustering is trained by taking around 2000 records as a sample train dataset. The three different clusters are formed by taking meaningful intuition datapoint similarities to provide the degree of impact of the web page content as 'high', 'average', and 'low'. The result of this approach demonstrates the success of potential content classification from raw data to an effective end product. The proposed method is a novel process and efficient in categorizing the highly impacted web contents for the particular query request posed by the user for the recommendation. Future work depends on these datasets of the obtained results in the database, where the personalized search query is posed. The desired relevant search results are sent to the users as actual recommendations for accurate, convenient, and flexible learning.

References

[1] Xiaoguang Qi, and Brian D. Davisona, “Web Page Classification: Features and Algorithms,” *ACM Computing Surveys*, vol. 41, no. 2, pp 1-31. 2009. [CrossRef] [Google Scholar] [Publisher Link]

[2] Kavita Sharma, Gulshan Shrivastava, and Vikas Kumar, "Web mining: Today and tomorrow," *3rd International Conference on Electronics Computer Technology*, pp. 399-403, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[3] Yeqing Li, "Research on Technology, Algorithm and Application of Web Mining," *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), IEEE*, vol. 1, pp. 772-775, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[4] R. Cooley, B. Mobasher, and J. Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web," *Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence*, pp. 558-567, 1997. [CrossRef] [Google Scholar] [Publisher Link]

[5] Guandong Xu, Yanchun Zhang, and Lin Li, “Web Content Mining,” *Web Mining and Social Networking*, Springer, Boston, MA. pp. 71-87, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[6] Anthony Scime, *Web Mining: Applications and Techniques*, IGI Global, 2005.

[7] M.G. da Costa, and Zhiguo Gong, “Web Structure Mining: An Introduction,” *2005 IEEE International Conference on Information Acquisition, IEEE*, p. 6, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[8] P Ravi Kumar, and Ashutosh Kumar Singh, “Web Structure Mining: Exploring Hyperlinks and Algorithms for Information Retrieval,” *American Journal of Applied Sciences*, vol. 7, no. 6, p. 840, 2010. [Google Scholar]

[9] Mahendra Pratap Singh Dohare, Premnarayan Arya, and Aruna Bajpai, “Novel Web Usage Mining for Web Mining Techniques,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 1, pp. 253-262. 2012. [Google Scholar]

[10] Mahdi Hashemi, “Web Page Classification: A Survey of Perspectives, Gaps, and Future Directions,” *Multimed Tools Applications*, vol. 79, pp. 11921–11945, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] T. Karthikeya et al., "Personalized Content Extraction and Text Classification Using Effective Web Scraping Techniques," *International Journal of Web Portals*, vol.11, no. 2, pp. 41-52, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Nichita Utiu, and Vlad-Sebastian Ionescu, "Learning Web Content Extraction with DOM Features," *IEEE 14th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 5-11, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[13] Farman Ali et al., "A Fuzzy Ontology and SVM–Based Web Content Classification System," *IEEE Access*, vol. 5, pp. 25781-25797, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[14] Atanas Dimitrovski, Ana Gjorgjevikj, and Dimitar Trajanov, “Courses Content Classification Based on Wikipedia and CIP Taxonomy,” *ICT Innovations 2017, Communications in Computer and Information Science*, Springer, Cham, vol. 778, pp. 140-153, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[15] Sharmila Shinde, Prasanna Joeg, and Sandeep Vanjale, "Web Document Classification using Support Vector Machine," *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication*, pp. 688-691, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[16] M.Vanathi, "Web Content Mining-A Study," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 1, no. 1, pp. 23-27, 2014. [CrossRef] [Publisher Link]

[17] Luis Roberto Jiménez, "Web Page Classification based on Unsupervised Learning using MIME type Analysis," *International Conference on COMmunication Systems & NETworkS*, pp. 375-377, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Li Deng, Xin Du, and Ji-zhong Shen, “Web Page Classification Based on Heterogeneous Features and a Combination of Multiple Classifiers,” *Frontiers of Information Technology & Electronic Engineering*, vol. 21, pp. 995–1004, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] T.B. Lalitha, and P.S. Sreeja, “Personalised Self-Directed Learning Recommendation,” *Procedia Computer Science*, vol. 171, pp. 583-592, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] Bo Zhao, “Web Scraping,” Encyclopedia of Big Data, pp. 1-3, 2017. [Google Scholar]

[21] Anand V. Saurkar, Kedar G. Pathare, and Shweta A. Gode, “An Overview on Web Scraping Techniques and Tools,” *International Journal on Future Revolution in Computer Science & Communication Engineering,* vol. 4, no. 4, pp. 363-367, 2018. [Google Scholar] [Publisher Link]

[22] Richard Lawson, *Web Scraping with Python*, Packt Publishing Ltd, 2015. [Google Scholar] [Publisher Link]

[23] Simon Munzert et al., *Automated Data Collection with R: A Practical Guide to Web Scraping and Text Mining*, John Wiley & Sons, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[24] A. Chapagain, *Hands-On Web Scraping with Python: Perform Advanced Scraping Operations using Various Python Libraries and Tools Such as Selenium, Regex, and Others*, Packt Publishing Ltd, 2019. [Google Scholar] [Publisher Link]

[25] Elior Vila, Galia Novakova, and Diana Todorova, “Automation Testing Framework for Web Applications with Selenium Webdriver: Opportunities and Threats,” *Proceedings of the International Conference on Advances in Image Processing,* pp. 144-150, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[26] Bing Liu, “Sentiment Analysis and Subjectivity,” *Handbook of Natural Language Processing,* vol. 2, pp. 627-666, 2010. [Google Scholar]

[27] Abdul-Mageed, Muhammad, Mona Diab, and Mohammed Korayem. "Subjectivity and Sentiment Analysis of Modern Standard Arabic," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,* pp. 587-591. 2011. [Google Scholar] [Publisher Link]

[28] Ian Rogers, "The Google Pagerank Algorithm and How it Works," 2002. [Google Scholar]

[29] Amy N. Langville, and Carl D. Meyer, *Google's PageRank and Beyond*, Princeton University Press, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[30] Meng Cui, and Songyun Hu, “Search Engine Optimization Research for Website Promotion,” *2011 International Conference of Information Technology, Computer Engineering and Management Sciences, IEEE*, vol. 4, pp. 100-103, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[31] Meenakshi Bansal, and Deepak Sharma, “Improving Webpage Visibility in Search Engines by Enhancing Keyword Density using Improved on-Page Optimization Technique,” *International Journal of Computer Science and Information Technologies*, vol. 6, no. 6, pp. 5347-5352, 2015. [Google Scholar]

[32] Sanghamitra Bandyopadhyay, and Sriparna Saha, *Unsupervised Classification: Similarity Measures, Classical and Metaheuristic Approaches, and Applications*, Springer Science & Business Media, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[33] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek, “The Global k-Means Clustering Algorithm,” *Pattern recognition*, vol. 36, no. 2, pp. 451-461, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[34] Kristina P. Sinaga, and Miin-Shen Yang, "Unsupervised k-Means Clustering Algorithm," *IEEE Access*, vol. 8, pp. 80716-80727, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[35] Joonas Hämäläinen, Susanne Jauhiainen, and Tommi Kärkkäinen, “Comparison of Internal Clustering Validation Indices for Prototype-Based Clustering,” *Algorithms*, vol. 10, no. 3, p. 105, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[36] Purnima Bholowalia, and Arvind Kumar, “EBK-Means: A Clustering Technique Based on Elbow Method and K-Means in WSN,” *International Journal of Computer Applications,* vol. 105, no. 9, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[37] Shiva Asadianfam, Hoshang Kolivand, and Sima Asadianfam “A New Approach for Web Usage Mining Using Case Based Reasoning,” *SN Applied Sciences, Springer,* vol. 2, p. 1251, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[38] Karan Sukhija, "Semantic Web Mining: An Amalgamation for Knowledge Extraction," *SSRG International Journal of Computer Science and Engineering,* vol. 2, no. 8, pp. 14-17, 2015. [CrossRef] [Publisher Link]

[39] Makinde Opeyemi Samuel, Afolabi Ibukun Tolulope, and Oladipupo Olufunke Oyejoke, “A Systematic Review of Current Trends in Web Content Mining,” *Journal of Physics: Conference Series*, vol. 1299, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[40] Huiran Li, and Yanwu Yang, “Keyword Targeting Optimization in Sponsored Search Advertising: Combining Selection and Matching,” *Electronic Commerce Research and Applications*, vol. 56, p. 101209, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[41] Mayank Nagpal, and Andrew Petersen, “Keyword Selection Strategies in Search Engine Optimization: How Relevant is Relevance?,” *Journal of Retailing*, vol. 97, no. 4, pp. 746-763, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[42] Binbin Gu et al., "The Interaction Between Schema Matching and Record Matching in Data Integration," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 186-199, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[43] Shishir K. Shandilya, and Suresh Jain, "Opinion Extraction & Classification of Reviews from Web Documents," *IEEE International Advance Computing Conference*, pp. 924-927, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[44] Fatima Almatrooshi et al., “Text and Web Content Mining: A Systematic Review,” *Proceedings of International Conference on Emerging Technologies and Intelligent Systems,* vol. 299, no. 79-87, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[45] Derar Alassi, and Reda Alhajj, “Effectiveness of Template Detection on Noise Reduction and Websites Summarization,” *Information Sciences, Elsevier,* vol. 219, pp. 41-72, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[46] Shinde Santaji Krishna, and Joshi Shashank Dattatraya, "Schema Inference and Data Extraction from Templatized Web Pages," *International Conference on Pervasive Computing (ICPC)*, pp. 1-6, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[47] Ms. Anushree Negi, "A Brief Survey on Text Mining, Its Techniques, and Applications," *SSRG International Journal of Mobile Computing and Application*, vol. 8, no. 1, pp. 1-6, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[48] Faizan Shaikh et al., "SWISE: Semantic Web Based Intelligent Search Engine," *2010 International Conference on Information and Emerging Technologies*, pp. 1-5, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[49] Kenan Enes Aydın, and Sefer Baday, "Machine Learning for Web Content Classification," *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1-7, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[50] A. Cavalieri et al., "An Intelligent System for the Categorization of Question Time Official Documents of the Italian Chamber of Deputies," *Journal of Information Technology & Politics*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[51] Sandeep Sirsat, "Extraction of Core Contents from Web Pages," *International Journal of Engineering Trends and Technology*, vol. 8, no. 9, pp. 484-489, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[52] Ankit Dilip Patel, and Vimal N. Pandya, "Web Page Classification Based on Context to the Content Extraction of Articles," *2nd International Conference for Convergence in Technology*, pp. 539-541, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[53] Neha Tyagi, and Santosh Kumar Gupta, "Web Structure Mining Algorithms: A Survey," *Big Data Analytics. Advances in Intelligent Systems and Computing*, Springer, Singapore, vol 654, pp. 305-317, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[54] Manjunath Pujar, and Monica R Mundada, "A Systematic Review Web Content Mining Tools and its Applications," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, 2021. [CrossRef] [Google Scholar] [Publisher Link]