

Content-Based Filtering for Personalized Article Recommendations System

Mahmoud M.Alakrimi⁽¹⁾

Computer Department, College of
Education Abu Issa, Zawia University,
Libya, E-mail: M.Elakrami@zu.edu.ly
GSM: +218918791453

Abstract— This research investigates the development and evaluation of an article recommendation system, based on content-based filtering. The system utilizes natural language processing techniques to extract meaningful features from article text, such as keywords. These features are then used to calculate similarity between articles, enabling the system to recommend articles with similar content to users based on their reading history. The performance of the Content-Based Filtering Algorithm is assessed by evaluating its effectiveness in providing relevant and personalized article recommendations to users. For instance, the model successfully identified "Google shares data center security and design..." as the most similar article to the query "Google Data Center 360° Tour" based on the lowest Euclidean distance (1.18).

Keywords— recommendation system, natural language processing, Content-Based filtering, machine learning, TF-IDF, Cosine similarity.

I. INTRODUCTION

"I figured that reading Articles has its advantages. for example, they mostly have the latest information and are much more agile. If there is a breakthrough or reinvention of certain things, it reflects more efficiently in articles than books. Another thing that really works for me is reading on the topics I might not be interested in, only to find out it is actually interesting. I would not do that with a book, as reading a book needs commitment in terms of time and attention. Some Articles have often exposed me to new topics, information, and authors. This sentiment aligns with the adage, "Internet is the world's largest library. It's just that all the books are on the floor" [2]. In the contemporary digital age, the sheer volume of information available online has reached unprecedented levels, creating an information overload for users. Sifting through this vast sea of content to discover relevant and engaging articles has become a significant challenge. Traditional search engines, while powerful in their own right, often fall short in providing truly personalized recommendations. They primarily rely on keyword matching and may not adequately capture the nuanced interests and preferences of individual users. To address this challenge, this research focuses on the development and evaluation of a content-based article recommendation system. Content-based filtering is a prominent approach that leverages the inherent characteristics of articles to suggest relevant content to users [1]. By analyzing the content of articles, such as their topics, keywords, the system can identify patterns and similarities, enabling it to recommend articles that are likely to be of interest to a particular user. This research will employ a combination of data science techniques to build an effective recommendation system. Data collection will involve using

dataset of articles from Github website, encompassing a wide range of topics and including crucial metadata such as article titles, language, authors and full text. The collected data were Accuracy checked and preprocessed to handle missing values, inconsistencies, and noise. This step is crucial to ensure data quality and improve the accuracy of subsequent analysis. Feature engineering plays a vital role in this research. The TF-IDF (Term Frequency-Inverse Document Frequency) technique was employed to extract meaningful representations from the text of each article. These extracted features are then used to calculate the similarity between articles. Cosine similarity, a widely used metric in text analysis, was employed to measure the angular distance between the vector representations of articles, thereby quantifying their similarity. Building upon the extracted features and similarity measures, the system utilizes content-based filtering algorithms to generate personalized recommendations. These algorithms analyses user's reading history, identifying articles that exhibit high similarity to those previously read by the user. By leveraging the inherent characteristics of the articles themselves, the system aims to provide users with relevant and engaging recommendations that align with their individual interests and preferences. This research will contribute to the development of more effective and personalized information retrieval systems, enabling users to navigate the vast expanse of online content with greater ease and efficiency. By providing users with targeted recommendations, the system aims to enhance their online reading experience, improve information discovery, and ultimately save them valuable time and effort."

II. RELATED WORK

Recommender systems are trained to suggest fast and relevant results to users. In the existing literature, many works have tried to produce efficient recommendation engines using Term Frequency - Inverse Document Frequency and Cosine Similarity.

(Afika., R,2024) employed the CRISP-DM methodology and utilized TF-IDF and Cosine Similarity algorithms. A dataset of 100 machine learning journal articles used for evaluation. Content-Based Filtering demonstrated promising results, achieving a precision score of 76%. However, the limited dataset size impacted the system's performance, particularly in recommending diverse articles.

(Mohammed J., Ayat F.,2024) developed Academic Article Recommendation Systems (ARSs) to assist researchers by suggesting relevant articles. Algorithms employed in ARSs facilitate navigation through vast scholarly literature. This review explores existing research

on ARS, encompassing recommendation algorithms, data sources, evaluation metrics, and user interfaces. Furthermore, the review examines emerging trends, including the application of advanced machine learning techniques and semantic analysis in enhancing the performance of ARSs.

(Bram B., Indra,2024) in their study they aimed to enhance an existing article recommendation widget by incorporating reader interest data. Item-Based Collaborative Filtering was employed, analyzing reading time and article selections to personalize recommendations. A simulation using reader data demonstrated a preference for sports news with a score of 0.743210. The goal is to improve user engagement by providing highly relevant article recommendations that align with individual interests.

(Gisela Y., Dade N., Selly M.,2022). This study leverages TF-IDF to recommend news articles. By weighting words in news titles and calculating cosine similarity, the system identifies related articles. To evaluate accuracy, a hit-rate metric was used to compare recommendations with actual user clicks on Microsoft News. The study achieved a hit-rate of 80.77%, demonstrating the effectiveness of the TF-IDF approach in recommending relevant news articles.

III. METHODOLOGY

A. Research procedure

The research employs a combination of data science techniques, including data collection and preprocessing, feature engineering, model development and evaluation as shown in figure (1). Dataset will collected from Github website and preprocessed to handle missing values and inconsistencies, feature extraction TF-IDF technique are employed to extract meaningful representations from the text. Utilizing these extracted features, cosine similarity is computed between articles. Finally, the system recommends articles to users based on their similarity to previously read articles.

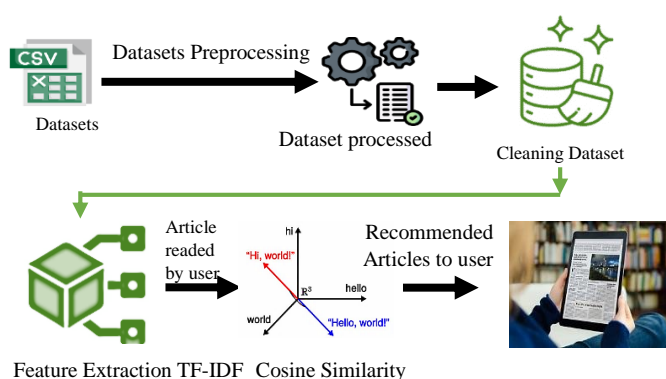


Figure 1: System Development Flow using TF-IDF & Cosine

B. Recommendation System over the web

Article Recommendation Systems personalize the web reading experience. By analyzing user behavior and article content, they suggest relevant articles, helping readers discover new content and save time. These systems are not limited to online reading; they are widely used in e-

commerce, job portals, and streaming services to personalize recommendations and enhance user experiences across various platforms.

C. Content-Based recommendation

In this paper, a plan to use the content-based recommendation method to create a recommendation system for articles over the web is planned. These articles could be in HTML format or Video format, or Rich Text format.

The procedure Scenario

1) pre-processing a dataset that provides the attributes of articles and the target user.

2) The idea is to assume the user is reading one of the articles from the articles dataset and providing that input.

3) Then the recommender system will return a list of articles with similar topics that the user might want to read next.

This model is beneficial when a user researches a particular subject/topic and subsequently might want to read similar articles.

Tools: python is used as programming language for development.

Dataset: In this research, shared_articles.csv Dataset from the Github website is used. We only used 2211 record English titles from 3122 articles .The number of datasets used is limited due to limited TF-IDF method which can only work on a small corpus.

Preprocessing: The dataset has the following Features:

(Timestamp, eventType, contentId, authorPersonId, authorSessionId, authorUserAgent, authorRegion, authorCountry, contentType, url, title, text, lang).

eventType: Article shared or article removed at a particular timestamp. We have 3047 Content shared and 75 Content Removed .

authorCountry: figure 2 display the distribution Country for the author of the articles.

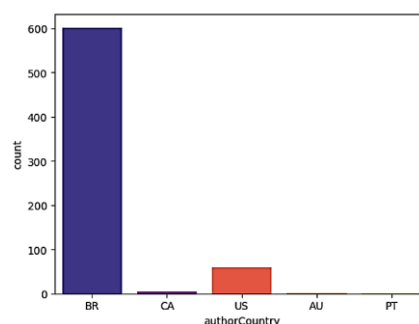


Figure 2: Author Country Distribution

Content Type: The formats of type ['HTML' 'RICH' 'VIDEO'] articles are shared but this system will limit the recommendations to one format only.

URL: URL of the articles; It will be useful for the users to navigate to the article directly.

Title: Figure 3 shows Title/headline of the articles.

```

1  Ethereum, a Virtual Currency, Enables Transact...
2  Bitcoin Future: When GBPcoin of Branson Wins O...
3  Google Data Center 360° Tour
4  IBM Wants to "Evolve the Internet" With Blockc...
5  IEEE to Talk Blockchain at Cloud Computing Oxf...
Name: title, dtype: object

```

Figure 3: Title Column

The good news here is there's no record with the title as null or blank. This is used to identify the recommendation and also as input to the recommendation system.

Text: Figure 4 displays the content of the articles.

```

1  All of this work is still very early. The firs...
2  The alarm clock wakes me at 8:00 with stream o...
3  We're excited to share the Google Data Center ...
4  The Aite Group projects the blockchain market ...
5  One of the largest and oldest organizations fo...
Name: text, dtype: object

```

Figure 4: Text Column

This is the most critical column in the analysis since we will use a content-based recommendation system. we used this field to create a TF-IDF matrix for the analysis.

Lang — Figure 5 display language in which the article is written.

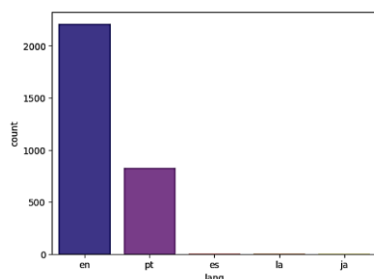


Figure 5: Language Distribution.

So, the most used language is English.

Though people might read in different languages, for simplicity, we will restrict the language to English.

IV. EXPLORATORY DATA ANALYSIS

Now that we have the data fields identified and cleaned up, we will start the analysis, leading to the recommendation system creation.

First, we will create a data frame included only the relevant columns.

```
Articles_df=pd.DataFrame(Articles_df,columns=[
'contentID','authorPersonId','content','title','text'])
```

Let's try to find the articles which are similar to the article user is reading, which is basically the input to my recommendation system. We can derive a pairwise cosine similarity and recommend articles with a similar threshold score to the input articles. While doing so, our recommendation system will face a known issue called the natural language processing problem. This means since the articles have different oratory and styles of sentence framing, finding similarity will be next to impossible. So, we will have to extract some features out of these texts. The feature is similar to the index, which focuses on the highly used words and creates an index.

we will first create a column to dataframe and call it 'Plate.' Plate is nothing but a concatenated version of all the feature fields. In this case, it will only contain the values in the column 'text.' The ' '.join(x['text']) part is used to combine a list of words into a single string with spaces between each word.

Readability: When we concatenate words without spaces, the resulting string becomes a single, long word, which is difficult to read and understand. For example:['This', 'is', 'an', 'example'] becomes "Thisisanexample" (unreadable). ' '.join(['This', 'is', 'an', 'example']) becomes "This is an example" (readable).

Natural Language Processing: Most natural language processing (NLP) tasks, such as tokenization, stemming, and lemmatization, assume that words are separated by spaces. By joining the words with spaces, you ensure that the resulting string can be processed correctly by NLP algorithms.

Term Frequency – Inverse Document Frequency (TF-IDF) is a method of weighting each word by calculating the frequency of occurrence of words in each document and the frequency of occurrence of words in all documents (Chiny, M.,2022 [4]). TF the greater the number of occurrences of a term in the document, the greater its weight . (IDF) aims to reduce the weight of the term if it exists in all documents. In contrast to TF, the less the frequency with which words appear in the document, the greater the value [10]. TF-IDF is done to change news titles in textual form to numeric ones so that they can be understood and processed by computers.

Several methods are available to create the vectorized form of the values in the 'text' column. (TF-IDF) vectors is used for each article. This will produce a matrix where each column represents a word in the overview vocabulary (all the words that appear in at least one article). Each column represents an article(title).

The TF-IDF score is the frequency of a word occurring in an article, down-weighted by the number of documents in which it occurs. This is done to reduce the importance of words that frequently occur in plot overviews and, therefore, their significance in computing the final similarity score.

TF-IDF assigns a weight to each term(word) in a document based on (TF - IDF) as the next Mathematical formula's.

$$\begin{aligned}
 \text{TF}(i,j) &= (\# \text{ times word } i \text{ appears in document } j) / (\# \text{ words in document } j) \\
 \text{IDF}(i,D) &= \log_e(\# \text{ documents in the corpus } D) / (\# \text{ documents containing word } i) \\
 \text{weight}(i,j) &= \text{TF}(i,j) \times \text{IDF}(i,D)
 \end{aligned}$$

So if a word occurs more often in an article but fewer times in all other articles, its TF-IDF value will be high.

scikit-learn is used to give a built-in TfidfVectorizer class that produces the TF-IDF matrix. We define a TF-IDF Vectorizer Object by removing all english stop words such as "the", "a", "is", "and" ,then Construct the required TF-IDF matrix by fitting and transforming the data. we selected the 'English' language to stop words like 'the', as we already have filtered the dataset to include articles published in English only.

The output shape of `tfidf_matrix` clearly says that there are about 45496 words shared among 2211 articles.

Cosine Similarity:

As mentioned earlier, it is independent of magnitude and is relatively easy and fast to calculate (especially when used in conjunction with TF-IDF scores). Mathematically, it is defined as follows:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}^T}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^n x_i \cdot y_i}{\sqrt{\sum_{i=1}^n (x_i)^2} \sqrt{\sum_{i=1}^n (y_i)^2}}$$

Figure 6 displays the results of using the `cosine_similarity()` function.

```
[[1.         0.03003382 0.01607362 ... 0.05093481 0.08979393 0.02016659]
 [0.03003382 1.         0.0234537 ... 0.02699924 0.03326086 0.0113887 ]
 [0.01607362 0.0234537 1.         ... 0.02257124 0.04506441 0.         ]
 ...
 [0.05093481 0.02699924 0.02257124 ... 1.         0.05944899 0.07901536]
 [0.08979393 0.03326086 0.04506441 ... 0.05944899 1.         0.04326837]
 [0.02016659 0.0113887 0.         ... 0.07901536 0.04326837 1.         ]]
```

Figure 6: Cosine similarity Matrix

Cosine similarity checks each pair of elements vector and finds the cosine angle between them. The less the angle, the more similar the elements are to each other. In this case, it's a 3k by 3 matrices between 0 & 1. The similarity vector is ready to create a reverse map of indices using the indices of the field 'title' also removing the duplicate titles, if any. In essence, creating a reverse map using the 'title' field is essential for translating the results of a similarity calculation into a human-readable and usable format.

Figure 7 display the Reset index of main DataFrame and construct reverse mapping as before.

```
title
Ethereum, a Virtual Currency, Enables Transactions That Rival Bitcoin's    0
Bitcoin Future: When GBCoin of Branson Wins Over USDcoin of Trump        1
Google Data Center 360° Tour                                             2
IBM Wants to "Evolve the Internet" With Blockchain Technology            3
IEEE to Talk Blockchain at Cloud Computing Oxford-Con - CoinDesk         4
Banks Need to Collaborate With Bitcoin and Fintech Developers            5
Blockchain Technology Could Put Bank Auditors Out of Work                6
Why Decentralized Conglomerates Will Scale Better than Bitcoin - Interview with OpenLedger CEO - Bitcoin News 7
The Rise And Growth of Ethereum Gets Mainstream Coverage                8
Situação financeira ruim de varejistas pressiona shoppings e eleva renegociações - Home - iG 9
dtype: int64
```

Figure 7:Reset Title Index

Now the indices are ready to create the recommender system using these maps and matrices. By using function to takes the article title as input, `indices(` having the titles and their indices) and outputs most similar articles. First, it finds the index of the input title. Since we have to remove it later, it iterates over the length of the cosine similarity matrix and checks for the distance of the articles from the input article we passed.

The result of Cosine Similarity enumerated list

```
((0,0.2674),(1,0.2619),(2,0.06862),(3,0.02637))
```

Now it will sort the articles based on their similarity values. The highest come first.

```
((1009,1.0),(1113,0.4222),(1636,0.4289),(1183,0.3939),(257,0.3478))
```

Then it picks only the top 10 similar articles having the highest similarity thresholds and creates a list of indices that are top 10 similar articles. Finally, it returns the list

with article titles matching the indices as derived in the earlier step.

V. RESULTS AND DISCUSSION

To use the recommendation system with various input titles. the input means a user has read or reading that article. So, the recommender will recommend based on the similarity of that article.

```
print(get_recommendations('Intel's internal IoT platform for
real-time enterprise analytics', indices, cosine_sim, metadata))
```

```
1422 Comparing IoT Platforms: Compare 4 IoT platfor...
2213 Bring a dinosaur to life with Watson IoT Platf...
1526 Decentralizing IoT networks through blockchain
302 IoT Day: A timeline of how IoT is changing the...
2879 Relating a Problem Definition to IoT Architect...
535 How IoT security can benefit from machine lear...
142 Is the Internet of Things in Your Home? Or on ...
2495 IoT Insurance: Trends in Home, Life & Auto Ins...
2074 Popular Internet of Things Forecast of 50 Bill...
2286 The Internet of Things is looking for its Visi...
Name: title, dtype: object
```

Figure 8: The result of first Title Test

Notice that in Figure 8 all the articles are based on IoT since the input article title had IoT in it. It is also imperative that the articles have IoT in their titles and have a similar frequency of usage for more than one word. Also, notice the variant IoT and Internet of Things have also been detected as similar, which is our intention.

Let's try in with the second title as input.

```
print(get_recommendations('The Rise And Growth of
Ethereum Gets Mainstream Coverage', indices,
cosine_sim,metadata))
```

```
0 Ethereum, a Virtual Currency, Enables Transact...
505 For Blockchain VCs, the Time for Ethereum Inve...
178 Ethereum and Bitcoin Are Market Leaders But No...
106 Solidity Available in Visual Studio - Ethereum...
80 Microsoft Adds Ethereum to Windows Platform Fo...
181 Microsoft Continues to Embrace Ethereum & Bitc...
109 Cashila Announces Convenient Buy and Sell Feat...
125 Eyeing Volume, Asian Exchanges Add Support for...
113 Decentralized Options Exchange Etheropt Uses A...
17 Five Bitcoin and Ethereum Based Projects to Wa...
Name: title, dtype: object
```

Figure 9: The result of second Title Test

So, maintaining its reputation so far, recommender system results in Figure 9 have returned a list of articles which is a range of articles related to Ethereum and Bitcoin. Also, it has identified articles that talk about policies related to these financial instruments across the globe.

we will use one final input to conclude this test.

```
print(get_recommendations('Google Data Center 360° Tour',
indices, cosine_sim,metadata))
```

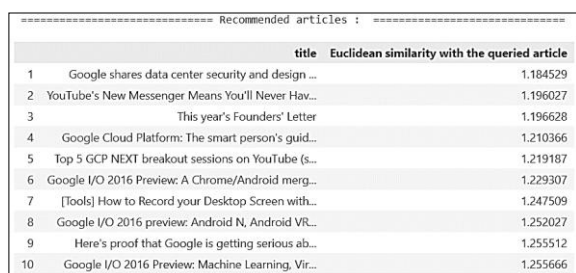
```
136 Google shares data center security and design ...
650 This year's Founders' Letter
240 Google Cloud Platform: The smart person's guid...
871 YouTube's New Messenger Means You'll Never Hav...
526 Top 5 GCP NEXT breakout sessions on YouTube (s...
849 Google I/O 2016 Preview: A Chrome/Android merg...
918 Google I/O 2016 preview: Android N, Android VR...
2813 [Tools] How to Record your Desktop Screen with...
922 Google I/O 2016 Preview: Machine Learning, Vir...
76 Google admits original enterprise cloud strate...
Name: title, dtype: object
```

Figure 10: The result of third Title Test

Notice that in Figure 10, the system found similar articles to google and I/O. It has also sensed an article that is related to youtube and doesn't have google in it. So evidently, our recommender system can detect the variants of a company's products. This is encouraging.

Euclidean distance

This metric measures the distance between two points in a multi-dimensional space. Applying Euclidean similarity for the title 'Google Data Center 360° Tour' and retrieve the top 10 similar titles as displayed in Figure 11.



	title	Euclidean similarity with the queried article
1	Google shares data center security and design ...	1.184529
2	YouTube's New Messenger Means You'll Never Hav...	1.196027
3	This year's Founders' Letter	1.196628
4	Google Cloud Platform: The smart person's guid...	1.210366
5	Top 5 GCP NEXT breakout sessions on YouTube (s...	1.219187
6	Google I/O 2016 Preview: A Chrome/Android merg...	1.229307
7	[Tools] How to Record your Desktop Screen with...	1.247509
8	Google I/O 2016 preview: Android N, Android VR...	1.252027
9	Here's proof that Google is getting serious ab...	1.255512
10	Google I/O 2016 Preview: Machine Learning, Vir...	1.255666

Figure 11: Euclidean similarity Results

In the context of Euclidean distance in a TF-IDF vector space, a distance in Figure 11 between 1.18 and 1.25 generally indicates that the two documents are fairly dissimilar. Understanding the domain of the documents can help in interpreting the distance more meaningfully. For example, a distance more than (1) might be considered small for documents in a highly specialized field, but large for documents in a general domain.

VI. CONCLUSION

Initially, we developed a simple recommendation system using content-based filtering. This system utilized a limited dataset of articles with basic attributes. It effectively suggests subsequent articles to users based on the content of the currently viewed article. This approach is particularly advantageous when user history or publication data is scarce, addressing the 'cold start' problem. However, as the system evolves and gathers more data on user behaviour and preferences, incorporating collaborative filtering techniques will be crucial for enhanced accuracy and personalization. Collaborative filtering can even be used to dynamically identify and utilize the most relevant features for recommendation [3],[7],[8]. Nonetheless, content-based filtering serves as a valuable initial approach, providing a solid foundation for a more sophisticated recommendation system.

VII. LIMITATION AND FUTURE WORK

Limitation

Article titles are long and do not focus on including keywords, which reduces the accuracy of the recommendation system results. Furthermore, the uneven distribution of data in certain columns, such as "userCountry" and "authorPersonId," prevents them from being effectively used as features. Incorporating more evenly distributed features would greatly improve the system's ability to recommend articles across diverse

topics, thereby increasing the range and diversity of recommendations.

Future Work

To enhance the model's understanding of word relationships, future work will investigate the integration of Word Embeddings, which can effectively capture semantic and syntactic similarities beyond the simple term frequency approach of TF-IDF.

VIII. REFERENCE

- [1]. Afika., R. Nuur., W. Abdul., M., and Ahmad., F. Jan 2024. "Machine Learning Journal Article Recommendation System using Content based Filtering A ". JUTI: Jurnal Ilmiah Teknologi Informasi - Volume 22, Number 1, January 2024: 1 – 10.
- [2] Akash Bhajantri , Nagesh K.1 , R H. Goudar., Dhananjaya G M , Rohit. B Kaliwal Vijayalaxmi Rathod , Anjanabhargavi Kulkarni and Govindaraja K , " Personalized Book Recommendations: A Hybrid Approach Leveraging Collaborative Filtering, Association Rule Mining, and Content-Based Filtering ", EAI Endorsed Transactions on Internet of Things | Volume 10 | 2024 |, doi: 10.4108/eetiot.6996.
- [3] Bram Bravo , Indra," Article Recommendations with Item-Based Collaborative Filtering on Online News Portals ",2024, Journal of Information Systems and Informatics Vol. 6, No.3, September 2024 e-ISSN: 2656-4882 p-ISSN: 2656-5935, DOI: 10.51519/journalisi.v6i3.851.
- [4]. Chiny, M., Chihab, M., Bencharef, O. and Chihab, Y. Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms. DOI: 10.5220/0010727500003101 In Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning (BML 2021), pages 15-20 ISBN: 978-989-758-559-3
- [5] Gisela Yunanda , Dade Nurjanah, Selly Meliana," Recommendation System from Microsoft News Data using TF-IDF and Cosine Similarity Methods ", Building of Informatics, Technology and Science (BITS) Volume 4, No 1, Juni 2022 Page: 277–284 ISSN 2684-8910 (media cetak) ISSN 2685-3310 (media online) DOI 10.47065/bits.v4i1.1670.
- [6]. Mohammed Jabardi, Ayat Abbas Fadhil,," A Web-Based Academic Article Recommendation System: Survey ". Article in Al-Furat Journal of Innovations in Electronics and Computer Engineering · July 2024 DOI: 10.46649/fjiece.v3.2.22a.1.6.2024
- [7] Ms. Tejashri Sharad Phalle , Prof. Shivendu Bhushan," Content Based Filtering And Collaborative Filtering: A Comparative Study ", Journal of Advanced Zoology ISSN: 0253-7214 Volume 45 Issue S-4 Year 2024 Page 96-100.
- [8] R. Glauber and A. Loula, "Collaborative Filtering vs. Content-Based Filtering: differences and similarities," 2019, [Online]. Available: <http://arxiv.org/abs/1912.08932>
- [9] Reetu Singh, Pragya Dwivedi," Food Recommendation Systems Based On Content-based and Collaborative Filtering Techniques ", 14th ICCNT IEEE Conference July 6-8, 2023 IIT - Delhi, Delhi, India.

[10] Shuhao Jiang , Yizi Lu , Haoran Song, Zihong Lu and Yong Zhang," A Hybrid News Recommendation Approach Based on Title–Content Matching " , . Mathematics 2024, 12, 2125. [https://doi.org/ 10.3390/math12132125](https://doi.org/10.3390/math12132125).

[11]. vikashraj luhaniwal,Jan-2022," Recommending news articles based on already read articles Content based recommendation in Python from scratch". Published in Towards Data Science. <https://towardsdatascience.com/>.

[12]. vikashrajluhaniwal," Recommending news articles based on read articles ",<https://www.kaggle.com/>.