-ELINVO (Electronics, Informatics, and Vocational Education), November 2023; vol 8(2): 221-230
ISSN 2580-6424 (printed), ISSN 2477-2399 (online) DOI: https://doi.org/10.21831/elinvo.v8i2.58014

Machine Learning System Implementation of Education Podcast Recommendations on Spotify Applications Using Content-Based Filtering and TF-IDF

Muhammad Mukti Raharjo[1], Fatchul Arifin[2*)]
[1,2]Magister of Electronic and Informatic Engineering Education Universitas Negeri Yogyakarta
[*]E-mail: fatchul@uny.ac.id

ABSTRACT

Spotify, this popular music and podcast streaming service, has a fundamental problem in assisting clients in finding podcasts that fit their interests. Thus, the goal of this project is to develop a podcast recommendation system that would enhance users' capacity to identify pertinent content, particularly in the educational genre. By using content-based filtration techniques, this system analyzes the user's listening preferences and interests before recommending educational podcasts. The podcast data source is Spotify, and the suggestions are produced using the TF-IDF and Cosine Similarity techniques. The recommendations provide a list of educational podcasts catered to the user's specific interests. The Confusion Matrix Classification Report was tested to assess system performance during the review phase. Precision values show how accurate the system was at recommending educational podcasts; on average, they range from 0.52 to 0.74. Additionally, the recall value showed a mean of 0.51 and a mean of 0.79, indicating that the algorithm successfully located the relevant content. To put it briefly, this custom recommendation engine enhances the listening experience for Spotify customers by suggesting educational podcasts based on their preferences. The system's ability to match users with material that aligns with their interests was demonstrated by the metrics used to assess its performance. With more user interactions with the system, it was anticipated by Cosine Similarity, a statistic used to determine the quality of recommendations, will continue to improve. To improve user experience and personalize the podcast listening experience on Spotify, this research addresses the challenge of locating suitable podcasts.

Keywords*:* content-based filtering, Spotify, TF-IDF cosine similarity, podcast

INTRODUCTION

In this ever-evolving era of information technology, understanding how this trend affects consumer preferences in music and podcast streaming is crucial. A podcast, as explained by Phillips, is a digital audio recording created and uploaded to an online platform for sharing with other people [1]. This is a digital file distribution format for audio files that various devices, including desktop, mobile, and portable media devices like MP3 players, may access.

Based on data from the Global Web Index, podcasts and audio storytelling have become the most popular media products among Indonesian consumers. By 2021, the number of Indonesian podcast listeners will surpass the global average, with more than 35.6% of the 16–64-year-old age group listening to podcasts [2].

Based on the number of Spotify users, especially in Indonesia, this platform's capabilities are becoming recognized as a new industry standard [3]. According to data, Spotify has a substantial market share in Indonesia, with a minimal growth rate. This allows us to create educational content on a platform with a solid user base. With around a million songs released in Indonesia and about 3.9 million [4], Spotify is a beneficial medium for presenting more extensive educational content to students.

Thus, there is an excellent potential for improving educational content in the form of podcasts [5], especially if we can better understand our preferences and listener behavior about educational podcasts. However, the primary challenge faced when presenting educational content through streaming platforms like Spotify is the dominance of high-quality

content. One possible solution is to integrate with Spotify's API to create new opportunities for creating, distributing, and evaluating educational content in a more efficient manner [6]. This will enable students to comprehend the lesson's scope and reach a higher audience, especially for those who use audio learning methods that use the grand theory of audio learning. This also assists students in selecting course materials that align with their interests and needs through more individualized recommendations.

The recommendation system is a policy that systematically removes data based on user preferences [7], [8]. Term Frequency-Inverse Document Frequency, or TF-IDF for short, is one technique in the recommendation system that combines two concepts for bot detection: the frequency at which a word appears in a given document and the inverse frequency of the document that contains the word in question [9]. The frequency of word occurrences in the provided document indicates some essential terms. In the context of podcast recommendations, TF-IDF can be used to determine the relevance of the episode to the listener's preferences, where the guest commentary that appears in the attack will provide information on topics and content relevant to the listener. This enables the use of TF-IDF to suggest podcasts that are appropriate based on the listener's interest.

The use of TF-IDF (Term Frequency-Inverse Document Frequency) is a technique that has been widely applied in various research contexts. This algorithm has been used in several fields, including the Classification System for Documents of Knowledge, which assists in classifying and ranking documents based on their content and structure [10]. In addition, TF-IDF is also used in Text Categorization on Desensitized Data, which aims to classify text in data that has already undergone desensitization to reduce information deterioration [11]. Other research includes Boosting for Short-text Classification: Application to Short-messages Generated During Disasters. TF-IDF is applied to classify short texts, specifically those generated during catastrophic events [12].

It is important to note that, even in crises like a pandemic, medical professionals may assist researchers and students in quickly addressing specific questions that arise during a pandemic, such as questions about treatment, containment, and viral shedding [13]. This highlights the flexibility and applicability of the TF-IDF method, even in tight situations like pandemonium, where quick understanding and access to accurate information are crucial. The latest findings in this study moved the TF-IDF concept to a higher threshold. As of right now, this algorithm is being used in the development of an innovative educational podcast recommendation system. In this context, TF-IDF is used as a guide to recommend educational podcasts that align with listeners' interests. Not only that, but the primary difference between this system's integration with Spotify's API is that APIs (Application Programming Interfaces) are interfaces, functions, and protocols that enable programmers to interact with other operating systems and allow users to access educational content easily [14]. This makes it possible for recommendations to be more individualized and tailored to the recipient's preferences. Thus, the first research study introduces the concept of TF-IDF to the domain of recommended educational content, resulting in more effective and engaging learning experiences for the digital generation that increasingly uses streaming platforms like Spotify to access information and educational content.

Thus, the recommendation system can recommend content based on user observation results, which makes it very relevant in the context of developing a podcast-based educational platform. Utilizing machine learning and recommendation systems, we can create learning resources that are easier to use and more engaging for the digital generation, which is increasingly demanding access to educational content and information.

LITERATURE REVIEW

Machine learning is a form of artificial intelligence that can enable computers to analyze data without clearly, forcefully, and honestly communicating its meaning; instead, it must adhere to predetermined instructions [15].

Machine learning is more focused on recommendation systems in this study's implementation of an educational podcast recommendation system on Spotify, which can identify educational podcasts and predict which users will find exciting and appropriate to listen to. Creates a list of podcasts and provides educational background. Education that will be heard by users and that has been foreseen.

A subset of machine learning called recommendation systems can deal with issues more specialized than information overload. The recommendation engine facilitates user processing.

Additional information is expected to satisfy user wants and needs by offering targeted recommendations to users [16]. The recommendation system uses a variety of techniques, such as [17]:

1. Content-based: To make recommendations to users, content-based methods compare items to those already chosen.
2. Collaborative filtering: This technique uses the idea that users' preferences can be used to predict what other users will find appealing.
3. Hybrid approach: A hybrid approach, in general, combines more than one approach to produce predictions of higher quality.
4. Knowledge-based: Make recommendations by the criteria you specify for an attribute value. To make recommendations based on the item's attribute the user likes, this recommendation system explicitly asks him to enter the thing at the beginning of use.

A system known as content-based screening learns to suggest the same product to new users by comparing previous users

According to how closely user and item profiles resemble one another, Content-Based Filtering is used in this study to select and rank items. Because each item is already known to the user from podcast reflections, this method has the advantage of bringing up similarities with previously chosen things pertinent to the user. Typically, they make a decision.

Preprocessing is the process of transforming unstructured data into data that is structured to meet the needs of the following processing step. Data that has undergone preprocessing is more distorted [18].

The perceptual technique of cosine similarity is frequently used to reduce crosstalk between items [19]. Confusion Matrix is one technique for assessing system performance. In essence, this confusion matrix contains data that can be used to compare the classification conclusions reached by the system [20]. Equations (2) and (3) define precision and accuracy in the confusion matrix formula (3).

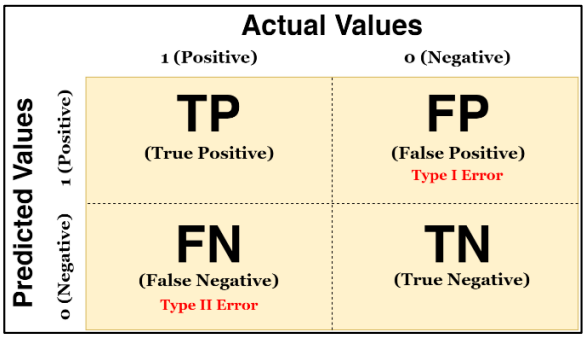

Figure 1. Confusion Matrix

There are four key terms in the confusion matrix that are relevant to classification [21]:

1. True Positive (TP):

Positive data that was correctly predicted. For instance, the patient has cancer (class 1) despite the model's prediction that they do.

2. True Negative (TN):

Harmful data that was correctly predicted. For example, the patient does not have cancer (class 2), despite the model's prediction that they do.

3. Type I Error: False Positive (FP):

Harmful data incorrectly forecasted as positive. For instance, the patient may have

cancer (class 1) despite the model's prediction that the patient does not (class 2).

4. False Negative (FN) (Type II Error):

Positive information that is incorrectly seen as unfavorable. For example, class 1 patients are predicted by the model to have cancer, whereas class 2 patients do not.

Python is an interpreted, interactive, object-oriented programming language. Python has a massive amount of power and a straightforward syntax. Has dynamic data types, dynamic typing, modules, classes, and exceptions. Python is becoming increasingly popular right now, and many new programmers prefer it because it is simple to understand and has a ton of libraries to help programmers out [1].

METHOD

The steps taken in developing a recommendation system for educational podcasts in the Spotify application can be seen in Figure 1. The following:

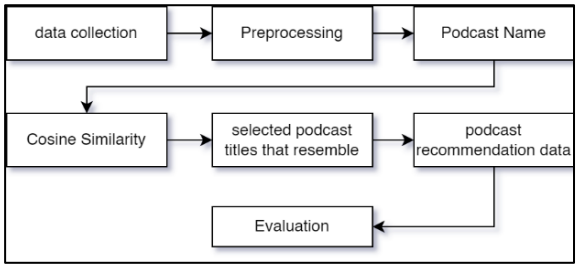

Figure 2. Flowchart of Research Methods

A. Data Collection Stage

Researchers gathered materials as research materials in the initial stages. The information gathered is text-based Spotify podcast data retrieved from the Spotify API using a client ID and client secret. Data was collected based on genre, which we used as a guide when gathering podcast data, and was then arranged alphabetically based on the volume of information gathered by the author to speed up the testing process. Following the genre-based dataset collection, the author obtained several datasets, which were then preprocessed by rearranging the data that had not yet been organized alphabetically by educational podcasts. Finally, all the data obtained were converted into test data. The SQL database will then store the lists of all the podcasts [22].

B. Preprocessing

At this stage, it will provide a change in a dataset from data collection obtained from the Spotify API to be converted into alphabetical data for educational podcasts so that it is more structured and becomes the final data. By sorting data, the dataset will become more neatly arranged and ordered alphabetically.

C. Implementation of Content-Based Filtering

The principle of the content-based filtering method is to provide recommendations based on the similarity of items. When users select an educational podcast that they like, a recommendation will be given from the system in the form of a list of podcast items that are similar to the podcast they usually like and a method of calculating similarity by comparing one podcast's data with another podcast from the Spotify podcast profile account that they like. Based on the Spotify list, the following are the flowchart stages of Content-Based Filtering Implementation:

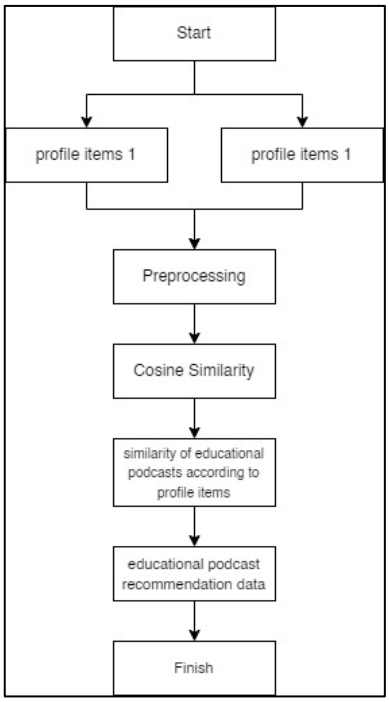

Figure 3. Conten-Based Filtering Implementation Flowchart.

In simple terms, at the cosine similarity stage, it is used to perform similarity calculations for certain documents. Here is a brief summary of the cosine equation [23] :

$$\cos(\theta) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\|\boldsymbol{A}\|\|\boldsymbol{B}\|} = \frac{\sum_{i=1}^{n} \boldsymbol{A_i B_i}}{\sqrt{\sum_{i=0}^{n} \boldsymbol{A_i^2}} \sqrt{\sum_{i=1}^{n} \boldsymbol{B_i^2}}} \quad (1)$$

where:

A : Vector A serves as the comparison item for item B,
B : which will be used to compare the two.
A • B : dot product between vectors A and B
|A| : length vector A
|B| : length vector B
|A||B| : multiplication operation based on data in vectors A and B

The two objects are considered identical if their similarity values are one and vice versa. The two objects are thought to be more similar and vice versa depending on the function similarity value.

D. Top-N Recommendation

The implementation results are cosine similarity values between one educational podcast and another. The object obtained from the similarity calculation is then used as a recommendation for the user. These items were selected so that only the five items with the highest similarity scores were included in the Top N recommendations because the higher the similarity score, the more similar the two podcasts being compared are assumed to be. The five items are then presented as a list of recommendations for the user to choose from when selecting a podcast.

E. Evaluation

Precision and accuracy calculations were used to evaluate the recommendations' quality. For this assessment, one of the categories of the chosen podcast will be compared with one of the suggested podcasts. A request was deemed accurate if it fell into the same category as the chosen one; if not, it was deemed inaccurate. After the data had been processed using TF-IDF and Cosine Similarity, this testing was one.

After that, the writers will produce a confusion matrix like the one shown in Figure 1 [24]. This matrix is a valuable tool for assessing how well the system is doing. We can compare the system's output thanks to the data structure. Ensuring consistency in mathematical formulas mitigates system stress and ambiguity throughout the categorization process.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (2)$$

$$Precision = \frac{TP}{FP+TP} * 100\% \quad (3)$$

When it comes to forecasting, True Positives (TP) and True Negatives (TN) indicate accurate projections, whereas False Positives (FP) and False Negatives (FN) indicate prediction errors that should be reduced [25].

The predictions for data with comparable degrees of similarity were analyzed using the table holding the confusion matrix data. The researcher then created a webpage on which the analysis's findings were displayed. This webpage offers viewers a list of suggested instructional podcasts simultaneously.

RESULTS AND DISCUSSIONS

A. Get Dataset

In this study, the author's data was retrieved from the Spotify API and converted into a CSV file (comma-separated values), which contained information about educational podcasts based on the genre we selected. The data was then entered into the SQL data output, which was displayed in the image results:

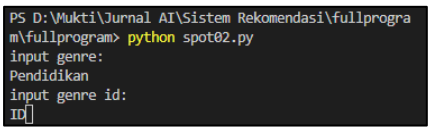

Figure 4. Perform The Genre Id Process And Enter The Genre

Figure 5. Spotify Podcast API Get-Process (Crawling data)

In these results, it was obtained that the get API podcast Spotify found an irregular dataset. Data preprocessing was done with data cleansing to make it more orderly.

Figure 6. Preprocessing Process

The following is the data that was exported to phpmyadmin.

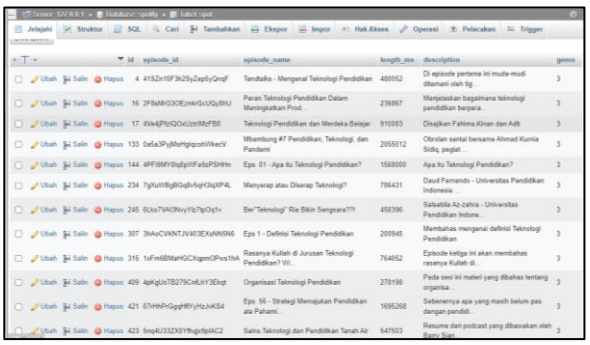

Figure 7. shows the results of sorting the educational podcasts' SQL data.

As seen in the data above, the database had sorted the Education Podcast data in SQL data, adjusting the id, episode_id, episode_name, description, and genre to the podcast's educational background.

B. System Design and Implementation

The design that is used will be presented clearly and concisely by providing a sketch that the author created using the following diagram:

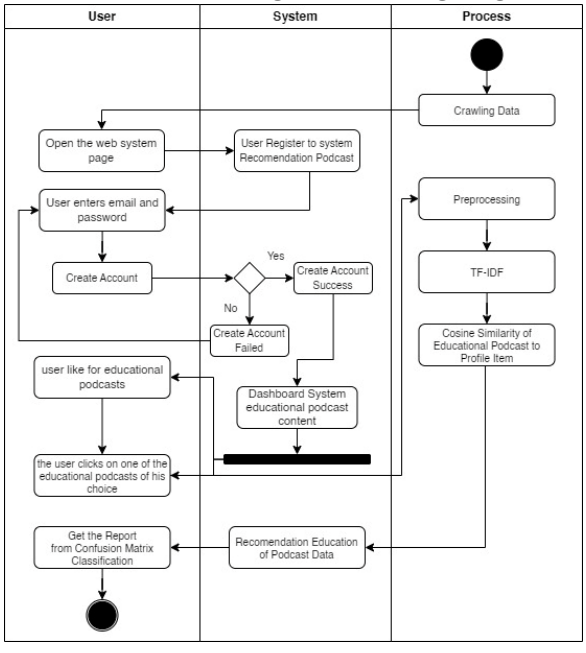

Figure 8. Diagram Activity

This step involves fetching the Spotify API, which will be crawled through the Spotify database. The above data will be generated by displaying several tabs similar to the figures in

Figure 4. In this particular chapter, the author filters podcast genre data as a filter. However, writers can filter to a few other genres using the Spotify database.

The next phase involves putting the data collection results into practice, specifically by implementing content-based filtering on the system we developed. Following the preprocessing stage, we go through several steps to implement content-based filtering, including using cosine similarity and TF-IDF to determine the similarity and height of similar numbers.

At this point, the customer will receive a list of every podcast on the website. The item's episode_name will be sent by the system once the user has chosen an item. The controller's source code to get the user's episode_name is as follows:

```
@login_required(login_url="/login/")
def podcast_detail(request):
    episode_name = request.GET.get('episode_name')
    episode_id = request.GET.get('episode_id')
    description = request.GET.get('description')
    length_ms = request.GET.get('length_ms')
```

Figure 9. Allows logged-in users to access the podcast detail page

The code is a system component that gives users who are signed in access to the podcast detail page. When a user visits that page, information about the podcast episode, including its name, ID, description, and runtime, is obtained from their request and prepared for additional processing within the application. The user will be taken to the login page if not logged in.

The following illustrates how the get API cost stage of the visual studio code application is followed by vectorization to the cosine and data sorting stages:

```
df_content = pd.DataFrame(table_rows)
print(df_content)

sepprod = spot.objects.all()

tfidf = TfidfVectorizer(stop_words='english')
df_content[2] = df_content[2].fillna('')#Construct the required TF-IDF matrix by applying the fit_tr
description_matrix = tfidf.fit_transform(df_content[2])#Output the shape of tfidf_matrix
print("proses cleaning data")
print(df_content[2])

similarity_matrix = linear_kernel(description_matrix,description_matrix)
print("matrix similaritynya")
print(similarity_matrix[0])

mapping = pd.Series(df_content.index,index = df_content[2])

podcast_index = mapping[episode_name]
print("podcast_index")
print(podcast_index)

similarity_score = list(enumerate(similarity_matrix[podcast_index]))

similarity_score = sorted(similarity_score, key = lambda name:name[0], reverse=True)

similarity_score = similarity_score[0:10]


print("cosine similaritynya")
print(similarity_score)

podcast_indices = [i[0] for i in similarity_score]

listpod = df_content[2].loc[podcast_indices]


return render(request, 'podcast_detail.html',{ 'episode_id':episode_id, 'episode_name':episode_name,
```

Figure 10. Cosine Similarity and TF-IDF Code

Cosine similarity and TF-IDF were applied at the stage mentioned above. You will ultimately receive a recommendation for an educational podcast that will be listed in the system that was created. The appraiser whose number approaches the highest number is the number that will be closest to the value of the item's similarity. The working phase entails locating the page directory in cmd using the command python manage.py run server, after which the DNS server address will appear: http://127.0.0.1:8000/.

C. Profile Items

Additionally, the interface that was created so far is the home menu, which serves only as a dummy display to give the user a first impression when they access the podcast recommendation website:

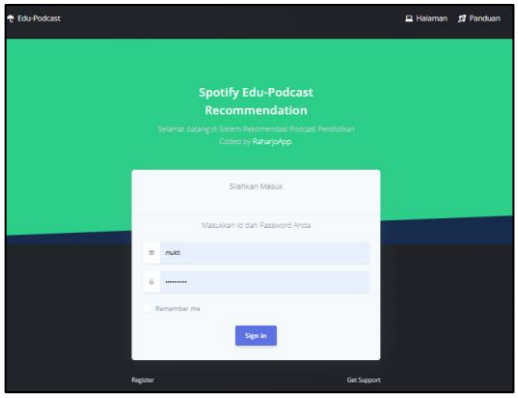

Figure 11. Initial System Profile Items

The above image shows how a user logs in to the Edu-Podcast website for the first time after entering their user ID and password. You can register first in the menu below if you have not already.

D. Dashboard of Recommendations

The next step is the Dashboard display menu on the Edu-Podcast website. The data below demonstrate that a crawling operation was performed only to display information about podcasts related to the field of education.

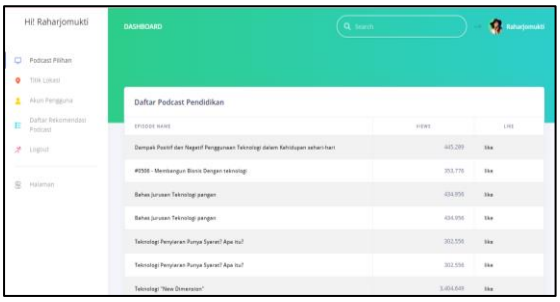

Figure 12. Display Menu

In the following step, we will attempt to enter one of the data to see the system's recommendations for choices that are similar to what we have made. The page view is shown below:

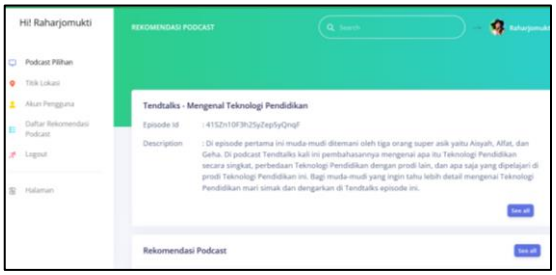

Figure 13. Display of a podcast page

Using cosine similarity and tf-idf in the display below, recommendations will be generated from a podcast after clicking on one of the data.

Figure 14. Podcast recommendations display

E. Black-box Testing System

Through functional testing, this black-box testing aims to determine whether the functionality of the recommended system application is up to par with expectations. If everything is satisfactory, then this recommendation system application is functioning correctly. Table 1. presents the results of the black-box testing using functional testing.

Table 1. Black-box Testing

Testing Components	Testing Result
Click the 'Login' button.	Valid
Click the 'Dashboard' button.	Valid
Click the 'Podcast' button.	Valid
Click the ‘Daftar Rekomendasi Podcast’ Menu	Valid
Click the 'Like' button.	Valid
Click 'See all' to See the Recommendation	Valid
Click the 'Log Out' button	Valid

F. Cosine Similarity Array Calculation

The Cosine Similarity process (similarity matrix and similarity scores) is shown in the following figure as an array of data.

Table 2. Array Data Podcast by User.

0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
.		.	.	.	.	.		.	.		.			.	.	.	.	.
0		0	6	0	0	0		0	0		0			0	0	0	0	0
3		3	1	3	3	3		2	3		3			3	2	2	2	6
4		3	4	3	4	5		4	4		5			7	8	4	9	1
3		2	7	3	3	1		5	3		1			9	8	5	4	4
8		5	5	5	8	5		5	8		3			4	5	5	7	7
3		0	1	0	3	0		8	3		0			4	8	8	9	5
2		3	6	3	2	7		5	2		7			0	6	5	4	1
		7		7		5		1			5			9	2	1		6
0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
.		.	.	.	.	.		.	.		.	.			.	.	.	.
0		0	0	0	0	0		0	0		0	0			0	0	0	0
3		3	6	3	3	3		2	3		3	3			3	2	3	3
8		7	9	7	8	9		7	8		9	6			2	7	6	3
8		6	4	6	8	6		7	8		6	9			5	7	9	2
3		6	3	6	3	9		3	3		8	4			9	3	4	9
5		9	6	9	5	0		8	5		0	4			4	8	4	7
8		2	2	2	8	1		8	8		1	0			8	8	0	
4		6	2	6	4	9		5	4		9	8			2	5	8	

Evaluation: at the data analysis stage, the author used the TF-IDF and cosine similarity methods to recommend podcasts. Using the Spotify API, which was processed into SQL data. The dataset used includes a total of more

than 10,000 podcast datasets. In the first run, the raw SQL dataset was converted using pandas. The results of the data set were processed into a TF-IDF matrix by applying the fit_transform method. The TF-IDF vectorizer was obtained by using the fit_transform method, and this value was used in matrix calculations.

The test was carried out with a comparison stage of the suitability of the podcast recommendations: Tendtalks - Getting to Know Educational Technology with datasets that users liked.

Table 3. Confusion Matrix Testing

	Positive	Negative
Positive	TP (0)	FP (1)
Negative	FN (2)	TN (7)

The confusion matrix is an essential table in evaluating binary classification systems. Consists of four critical cells: True Positive (true positive), False Positive (false positive), False Negative (false negative), and True Negative (true negative). In this example, 0 positive cases were correctly predicted, two positive points were incorrectly expected, one negative case was incorrectly expected to be positive, and seven negative cases were correctly predicted. This matrix calculates various system performance evaluation metrics in identifying positive and negative instances.

$$Precision = \frac{TP}{TP + FP} = \frac{0}{0 + 1} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} = \frac{0}{0 + 1} \quad (5)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{0 + 7}{0 + 7 + 1 + 2} \quad (6)$$

Our calculation for the accuracy value test, which yields the precision and accuracy values, is based on the results from the previous section.

Table 4. Confusion Matrix Classification Report Testing

	Precision	Recall	F1-Score	Support
1	0.00	0.00	0.00	2
0	0.82	0.88	0.84	8
Accuracy			0.82	10
Macro avg	0.52	0.51	0.49	10
Weighted avg	0.74	0.79	0.71	10

CONCLUSION

In conclusion, the content-based filtering method with TF-IDF and Cosine Similarity can offer podcast recommendations based on the user's preferred genre and episode name; the higher the similarity value, the more users like the podcast. Precision ranged between 0.52 and 0.74 in the Confusion Matrix Classification Report Testing data. We discovered that the average has a weight of 0.79 and a low value of 0.51 for recall. Since this value was so close to 1, the information aligned with the suggestions for podcasts in the educational genre.

REFERENCES

[1] M. M. Fahmy, "Confusion Matrix in Binary Classification Problems: A Step-by-Step Tutorial," *J. Eng. Res.*, vol. 6, no. 5, 2022.

[2] R. Pahlevi, "Persentase Pendengar Podcast terhadap Total Pengguna Internet Berdasarkan Negara, Kuartal III 2021," *databoks*, 2022. https://databoks.katadata.co.id/datapublish/2022/02/08/pendengar-podcast-indonesia-terbesar-ke-2-di-dunia.

[3] C. Osazuwa, "Spotify And Streaming Music Analysis," *christineosazuwa.com*, 2017. http://christineosazuwa.com/portfolio/spotify-and-streaming-music-industry-analysis/.

[4] R. Triwijanarko, "Tak Mau Disalip Kompetitor, Spotify Kembangkan Teknologi AI," *Marketeers*, 2017. https://www.marketeers.com/spotify-kembangkan-teknologi-ai/.

[5] B. Phillips, "Student-Produced Podcasts in Language Learning – Exploring Student Perceptions of Podcast Activities," *IAFOR J. Educ.*, vol. 5, no. 3, pp. 157–171, 2017, doi: 10.22492/ije.5.3.08.

[6] A. L. Fatroh, "EFEKTIVITAS PENGGUNAAN MEDIA PODCAST DALAM APLIKASI SPOTIFY UNTUK MENINGKATKAN PEMAHAMAN PESERTA DIDIK PADA PEMBELAJARAN SEJARAH KEBUDAYAAN ISLAM DI KELAS VIII SMPM 15 BRONDONG LAMONGAN,"

UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA, 2023.

[7] Y. Setiawan, A. Nurwanto, and A. Erlansari, "Implementasi Item Based Collaborative Filtering Dalam Pemberian Rekomendasi Agenda Wisata Berbasis Android," *Pseudocode*, vol. 6, no. 1 SE-Articles, pp. 13–20, Apr. 2019, doi: 10.33369/pseudocode.6.1.13-20.

[8] A. I. Putra and R. R. Santika, "Implementasi Machine Learning dalam Penentuan Rekomendasi Musik dengan Metode Content-Based Filtering," *Edumatic J. Pendidik. Inform.*, vol. 4, no. 1, pp. 121–130, 2020, doi: 10.29408/edumatic.v4i1.2162.

[9] R. T. Wahyuni, D. Prastiyanto, and E. Supraptono, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi," *J. Tek. Elektro Univ. Negeri Semarang*, vol. 9, no. 1, pp. 18–23, 2017, [Online]. Available: https://journal.unnes.ac.id/nju/index.php/jte/article/download/10955/6659.

[10] M. Yusuf and A. Cherid, "Implementasi Algoritma Cosine Similarity Dan Metode TF-IDF Berbasis PHP Untuk Menghasilkan Rekomendasi Seminar," *J. Ilm. Fak. Ilmu Komput.*, vol. 9, no. 1, pp. 8–16, 2020, [Online]. Available: https://publikasi.mercubuana.ac.id/index.php/fasilkom/article/view/8830.

[11] T. Zhang and S. S. Ge, "An Improved TF-IDF Algorithm Based on Class Discriminative Strength for Text Categorization on Desensitized Data," in *Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence*, 2019, pp. 39–44, doi: 10.1145/3319921.3319924.

[12] S. Ghosh and M. S. Desarkar, "Class Specific TF-IDF Boosting for Short-Text Classification: Application to Short-Texts Generated During Disasters," in *Companion Proceedings of The Web Conference 2018*, 2018, pp. 1629–1637, doi: 10.1145/3184558.3191621.

[13] J. Chamorro-Padial, F.-J. Rodrigo-Ginés, and R. Rodríguez-Sánchez, "Finding answers to COVID-19-specific questions: An information retrieval system based on latent keywords and adapted TF-IDF," *J. Inf. Sci.*, vol. 0, no. 0, p. 01655515221110995, doi: 10.1177/01655515221110995.

[14] S. Lubis, "Implementasi Application Programming Interface (API) Dalam Upaya Peningkatan Pengelolaan dan Pelayanan Informasi Publik Pada Kantor KPU Kabupaten Tapanuli Selatan," MAGISTER ADMINISTRASI PUBLIK, 2017.

[15] W. Budiharto, *Machine learning dan Computational Intelligence*, 1st ed. Andi Offset, 2016.

[16] W. Jepriana and S. Hanief, "ANALISIS DAN IMPLEMENTASI METODE ITEM-BASED COLLABORATIVE FILTERING UNTUK SISTEM REKOMENDASI KONSENTRASI DI STMIK STIKOM BALI," *JANAPATI*, vol. 9, no. 2, pp. 171–180, 2020.

[17] J. Pérez-Marcos, L. M. Gómez, D. M. Jiménez-Bravo, V. F. L. Batista, and M. N. M. García, "Hybrid system for video game recommendation based on implicit ratings and social networks," *J. Ambient Intell. Humans. Comput.*, pp. 1–11, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:213898591.

[18] R. V. Imbar, M. Ayub, and A. Rehatta, "Implementasi Cosine Similarity dan Algoritma Smith-Waterman untuk Mendeteksi Kemiripan Teks," *J. Inform.*, vol. 10, no. 1, pp. 31–42, 2014.

[19] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*, 1st ed., vol. 40. Cambridge University Press, 2011.

[20] E. Prasetyo, "Data mining konsep dan aplikasi menggunakan matlab," *Yogyakarta Andi*, vol. 1, 2012.

[21] R. H. Mondi and A. Wijayanto, "RECOMMENDATION SYSTEM WITH CONTENT-BASED FILTERING METHOD FOR CULINARY TOURISM IN MANGAN APPLICATION," *ITSMART J. Ilm. Teknol. Dan Inf.*, vol. 8, 2019.

[22] R. A. Wiryawan and N. R. Rosyid, "Pengembangan Aplikasi Otomatisasi Administrasi Jaringan Berbasis Website Menggunakan Bahasa Pemrograman Python," *SIMETRIS*, vol. 10, no. 2, pp. 1–12, 2019.

[23] M. Yusuf and A. Cherid, "Implementasi Algoritma Cosine Similarity Dan Metode TF-IDF Berbasis PHP Untuk Menghasilkan Rekomendasi Seminar," *J. Ilm. Fak. Ilmu Komput.*, vol. 9, no. 1, pp. 8–16, 2020.

[24] M. Nurjannah, Hamdani, and I. F. Astuti, "PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING," *J. Inform. Mulawarman*, vol. 8, no. 3, p. 110, 2013.

[25] Z. Karimi, "Confusion Matrix." 2021.