MOVIE RECOMMENDATION SYSTEM USING CONTENT-BASED FILTERING ON GRADIO FRAMEWORK

[1]Muhammad Irvan Arfirza, [2]Dr. Dina Indarti, S.Si., M.Si., M.T.

[1.2] Jurusan Informatika Fakultas Teknologi Industri Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat

[1]firzzairvn@gmail.com, [2] dina_indarti@staff.gunadarma.ac.id

Abstract

Film is a moving visual communication medium that contains elements such as Images, sound, and text to convey stories or information to the audience. The ease of accessibility to movies allows viewers to have a variety of viewing preferences. However, this can also lead to confusion in selecting the desired movie, often requiring a search for reviews or consultation with friends. Another challenge is the bias of recommendations provided by some digital media that may collaborate with movie owners or publishers for promotional purposes, reducing the validity of such recommendations. To address these issues, this research develops a recommendation system based on movie similarity. The method used is content-based filtering with TF-IDF and cosine similarity algorithms, which rely on the content of movies to recommend similar movies. This research uses a dataset from Grouplens with the dataset name MovieLens 25M. This dataset includes 25 million ratings for movies from 1995 to 2019 that provide rating data, movie titles, and tags. This recommendation model is able to provide recommendations for 10 movies with the highest similarity based on movie content. This research also tests the recommendation model using the Gradio framework to validate the performance of the developed recommendation system. The results show that the developed movie recommendation system can run well on the Gradio framework.

Keywords*: Movies, Recommendation System, Content-based Filtering, Gradio*

INTRODUCTION

The movie industry has come a long way since its inception in the 19th century. Every year, thousands of movies are produced and released around the world, with trends from movies every year changing such as horror, superhero, comedy and others so that the audience will have more choices of movies. With the development of existing technology, the audience is increasingly facilitated in accessing a film, for example, in the past to watch a film, the audience had to come directly

to the cinema but now the audience can access the film using only a smartphone.

With the ease with which viewers can access a movie, viewers will have more and more viewing preferences that they can choose from. But with that, a new problem arises, namely how when users like a movie and want to watch a movie that is similar to the movie the audience likes, so with these problems a recommendation system needs to be developed. The recommendation system serves to assist users in getting a recommendation based on the similarity of the items or content of a movie.

It is known that sometimes as a user is quite confused about the desired movie recommendation, so usually a search for reviews or asking friends to get recommendations. By looking for reviews, watching movie recommendations on the internet or asking friends, sometimes a new problem is found, namely biased recommendations. Sometimes some digital media that provide movie recommendations have collaborated with the owner or publisher of the film to promote the film, so that people who review the movie will try to recommend the movie, which means that the recommendations given can be said to be less valid.

Based on these problems, a recommendation system is needed that provides a recommendation to users based on the similarity of the movie they want. The recommendation model must also provide recommendations based on the data owned from each movie. By creating a recommendation model that analyses data from each content, this model can also be developed to various needs such as e-commerce, book recommendations, vehicle recommendations and so on provided that the processed data must explain the content of each item.

This recommendation model is built using the Content-Based Filtering algorithm. According to Bhatt (2014), the Content-Based Filtering system algorithm analyses documents or preferences given by certain users, and tries to build a model based on this data. This system utilizes the user's special interests and tries to match the user's profile with the attributes possessed by various content objects to be recommended [4]. For example, if a user likes comedy movies, then this algorithm will recommend other movies with the same genre.

This research was developed based on references from research conducted previously with the same method, namely Content-Based Filtering (CBF) with the title "Machine Learning Journal Article Recommendation System Using Content Based Filtering" [5] and research with the title "Job Recommendation System Using Content and Collaborative Filtering Based Techniques" [2]. Based on research conducted by Rianti et al., that the recommendation system carried out calculates the similarity of the journal dataset owned, with an average precision of 98.33%. For research conducted by Dhemeliya & Desai discusses the recommendation system applied to job recommendations in the e-recruitment process with recall evaluation results of 63.97%.

Therefore, this research develops a movie recommendation system using Content-Based Filtering. The Content-Based Filtering method uses TF-IDF algorithm and cosine similarity to calculate the similarity between movies. The developed recommendation system is then integrated using the Gradio framework.

RESEARCH METHODS

Problem Analysis

At this stage, an analysis of how a problem can be solved using a content-based filtering system, so that the objectives of this paper will be adjusted to the method used.

To create a model that can solve the problems previously described, several stages in modelling will be carried out starting from preparing data until the model is successfully integrated into the Gradio framework. All stages can be seen in Image 1.

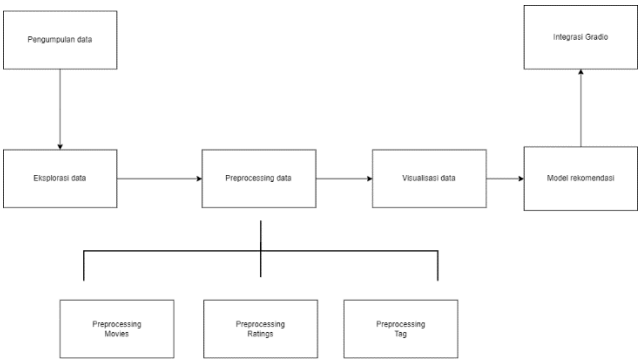

Image 1. Modelling Flow of Recommendation System

As seen in Image, the first step is to collect data that suits the modelling needs. In this paper, the Movielens 25M dataset is used. After the data is collected, it is necessary to explore it to find out the amount of data, parameters, and columns available.

The next step is data preprocessing, which varies depending on the needs of each dataset. This stage is important to ensure that the model created gives the right results. After preprocessing, the data needs to be visualized to make it easier to read. Good visualization is important to avoid data misinterpretation.

Finally, a recommendation model is created using a content-based method by calculating similarity using cosine similarity. This model is then integrated into the Gradio framework.

Data Collection

Data plays an important role in making training models, which can be likened to data is the fuel for making good models, with data that has quality data, the information presented will also be of quality.

The data source used to create the recommendation model comes from the dataset provided by the Grouplens website with the name MovieLens 25M Data dataset, which consists of genome-score.csv, genome-tags.csv, links.csv, movies.csv, ratings.csv, and tags.csv datasets. But in this scientific writing, only movies.csv, ratings.csv, and tags.csv datasets are used and will be combined into one dataset, the contents of these datasets can be seen in Table 1.

Table 1. Dataset Structure

Dataset	Variable	Description
Movies	movieId	Identity of each movie
	title	Title of each movie
	genres	Existing genres of each movie
Ratings	userId	Identity of the user who did the movie review
	movieId	Identity of each movie
	rating	The value of each movie according to users
	timestamp	The length of airtime of each movie
Tag	userId	Identity of the user who 'tagged' each movie
	movieId	Identity of each movie
	Tag	The impression the user gives of the movie
	Timestamp	The length of airtime of each movie

Data Preparation

This stage aims to prepare the data through manipulations such as data preprocessing. This stage is very important to produce quality data that will be used in modelling. The data preprocessing stage is very important in modelling because at this stage all data will be processed in various ways to produce a dataset that is suitable for modelling. This process ensures that the data is feasible, good, and in accordance with the modelling needs to be carried out, and determines whether the model

created will produce the appropriate output or not for the preprocessing stage can be seen in Image 2. In Image 2, various stages are carried out such as removing the timestamp column, calculating the average rating, normalizing the data in the rating dataset and so on, all of which need to be done to produce data that is suitable for making good models in providing recommendations.

Image 2. Pre-processing stage

Preprocessing Dataset Ratings

At this stage, some preprocessing steps must be done such as eliminating the timestamp column because the column is not used. The next preprocessing is to calculate the average rating on the dataset, the rating is used for the calculation of each available movie. In the initial dataset, each movie has several ratings, therefore the ratings must be combined into one so that each movie has only one rating and finally because the results of the average rating calculation have a format that is quite difficult to read, it is necessary to normalize the data by formatting only 2 numbers behind the comma to make it easier to read the data.

Preprocessing Dataset Tag

Some of the steps to be taken at this preprocessing stage include the removal of userid and timestamp columns, which are considered irrelevant for further analysis, as well as the merging of multiple tag columns into one more comprehensive column. After that, data normalization was performed on the tag column with the aim of cleaning the data from unnecessary symbols, such as punctuation marks or special characters, which could interfere with the analysis. In addition, all text in the tag column was converted to lowercase to ensure consistency and simplify the further analysis process.

Preprocessing Dataset Movies

In the movie dataset, some of the preprocessing steps that will be performed include the extraction of the year in the movie title column to separate the release year from the main title, as well as the removal of the year information from the movie title itself to

make it cleaner and suitable for the desired analysis format.

In addition, various data cleaning techniques will be applied to the genre column to ensure the movie categories are well-defined and there is no irrelevant or redundant data. The final step in this preprocessing is to filter the data, selecting movies released between 1995 and 2019, so that the resulting dataset is more focused and relevant for the analysis to be conducted.

Modeling

In the model building stage here will process the data owned by the content-based filtering method, the model will measure how often a sentence appears using TF-IDF and will calculate the similarity of each movie array using cosine similarity and finally the model will be evaluated. Content-Based Filtering (CBF) provides recommendations based on user preferences with given categories [3].

By using a content-based recommendation system, this system can give users the freedom to give ratings exclusively according to the preferences of each user. This rating will later be used to build a user profile that becomes the basis for more personalized recommendations.

Term Frequency - Inverse Document Frequency is a statistical method used in information retrieval and natural language processing to measure the importance of a word in a document. Term frequency states the number of words/terms/tokens that appear in a document [1]. To calculate Term frequency using equation (1) below:

$$TF(t,d) = \frac{Jumlah\ kemuculan\ term\ t\ dalam\ dokumen\ d}{Jumlah\ total\ term\ dalam\ dokumen\ d} \quad (1)$$

The IDF calculation itself uses equation (2) and the TF-IDF calculation uses equation (3) below:

$$IDF(t,D) = log\left(\frac{N}{1+Jumlah\ dokumen\ yang\ mengandung\ t}\right) \quad (2)$$

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D) \quad (3)$$

Cosine similarity is one method that is often applied to the process of information retrieval and clustering where the level of similarity of the distance between two documents will be measured [1]. The cosine similarity value ranges between 0 and 1, with 0 indicating dissimilarity and 1 indicating similarity. Mathematically, the cosine similarity formula between two vectors

A and B can be seen in equation (4) below:

$$Cosine\ Similarity(A,B) = \frac{A \cdot B}{||A||x||B||} \qquad (4)$$

Which one:

- A . B is the result of the dot product of vectors A and B.
- ||A|| is the magnitude (norm) of vector A, which is calculated as

$$\sqrt{a\frac{2}{1} + a\frac{2}{2} + .. + a\frac{2}{n}}$$

- ||B|| is the magnitude (norm) of the vector B, which is calculated as

$$\sqrt{b\frac{2}{1} + b\frac{2}{2} + .. + b\frac{2}{n}}$$

Model Evaluation

To see how the model provides actual recommendations, the model must be evaluated. The evaluation will be calculated based on the comparison between the two movies based on the cosine similarity calculation. The result of the calculation will be the percentage of similarity of the comparison and the result will be matched with the recommendations given.

Then the data must be validated whether the recommendations given really have the highest similarity. Then the model must be evaluated by comparing samples from the similarity list in Image 3.

```
# Test the recommendation function with user input
recommendation_input = input('Masukkan judul film yang kamu inginkan: ')
recommendations_system = get_recommendations(recommendation_input)
print(recommendations_system)
```

```
Masukkan judul film yang kamu inginkan:  cars 2
Toy Story 2 (Rating: 3.81)
Bug's Life, A (Rating: 3.57)
Cars (Rating: 3.33)
Monsters, Inc. (Rating: 3.85)
Toy Story (Rating: 3.89)
Monsters University (Rating: 3.5)
Finding Dory (Rating: 3.62)
Finding Nemo (Rating: 3.83)
Ratatouille (Rating: 3.81)
Toy Story 3 (Rating: 3.86)
```

Image 3. Movie Recommendations

In Image 4 below, it is known that the result of calculating the similarity using cosine similarity between the movie Cars 2 and the movie Toy Story 2 is 59.20%.

```
    sim_per = sim_score * 100

    return f"Film '{title1}' dan '{title2}' memiliki tingkat kemiripan sebesar {sim_per:.2f}%"

# Test dengan input manual
title1 = input('Masukkan judul film pertama: ')
title2 = input('Masukkan judul film kedua: ')
similarity_result = hitung_kemiripan(title1, title2)
print(similarity_result)
```

```
Masukkan judul film pertama:  cars 2
Masukkan judul film kedua:  toy story 2
Film 'cars 2' dan 'toy story 2' memiliki tingkat kemiripan sebesar 59.20%
```

Image 4. Evaluation 1

In Image 5 below will compare the movie Cars 2 with the movie Bug's Life, A has a similarity of 58.87% and from the bottom 10 recommendations tried to compare the movie Cars 2 with the movie Toy Story 3 has a similarity of 46.93% in Image 6. So from this data it can be seen that the recommendation model is able to calculate the similarity between movies accurately which in turn

can reduce errors in providing recommendations.

```
# Test dengan input manual
title1 = input('Masukkan judul film pertama: ')
title2 = input('Masukkan judul film kedua: ')
similarity_result = hitung_kemiripan(title1, title2)
print(similarity_result)
```

```
Masukkan judul film pertama:  cars 2
Masukkan judul film kedua:  Bug's Life, A
Film 'cars 2' dan 'Bug's Life, A' memiliki tingkat kemiripan sebesar 58.87%
```

Image 5. Evaluation 2

```
# Test dengan input manual
title1 = input('Masukkan judul film pertama: ')
title2 = input('Masukkan judul film kedua: ')
similarity_result = hitung_kemiripan(title1, title2)
print(similarity_result)
```

```
Masukkan judul film pertama:  cars 2
Masukkan judul film kedua:  Toy Story 3
Film 'cars 2' dan 'Toy Story 3' memiliki tingkat kemiripan sebesar 46.93%
```

Image 6. Evaluation 3

RESULTS AND DISCUSSION

Dataset

The dataset used consists of 25 million ratings datasets and consists of 1 million tags applied to 62 thousand movies that have been rated by 162 thousand people. After the whole series of preprocessing the available data is 23,681 rows of data grouped with 6 columns, the amount of each data can be seen in Table 2.

Table 2: Number of data in the Dataset

Dataset	Quantity	Column
Movies	62.423	3
Ratings	25.000.095	4
Tag	1.093.360	4
Merge	23.681	6

Data Visualization

Data needs to be visualized to make it easier to read the data. Some techniques that can be used include WordCloud to see words whose frequency appears frequently, the word will get a bigger picture, or another basic technique is to use a bar chart as in Image 8 which is a diagram showing the 10 most frequently occurring sentences.

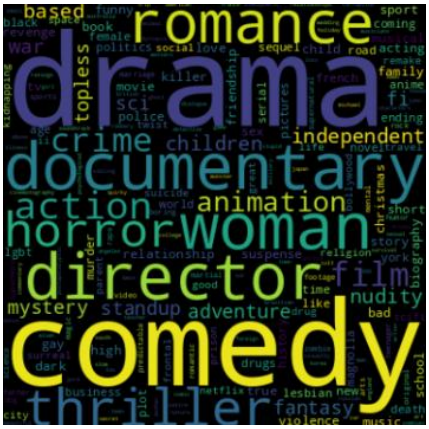

Image 7. Wordcloud

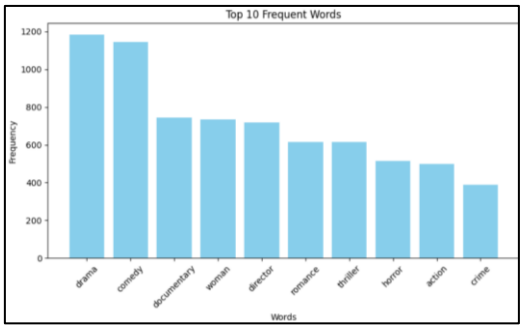

Image 8. Frequency of Many Sentences

Model Integration

After the model has been developed, the next step is to integrate the model into various applications. At this stage, the integration is done using the Gradio framework.

In addition to testing the model, the use of the Gradio framework aims to facilitate users in interacting with the system that has been built, thanks to its ability to provide an intuitive and interactive interface. The integration can be seen in Image 9 and the result of the integration is in Image 10 below:

```
#interface gradio
iface = gr.Interface(
    fn=recommendation,
    inputs=gr.Textbox(label="Masukkan Judul Film"),
    outputs=gr.Textbox(label="Rekomendasi Film"),
    title = 'Rekomendasi Film'
)

# Jalankan antarmuka
iface.launch()
```

Image 9. Model Integration to Gradio

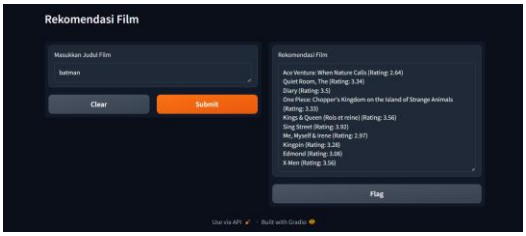

Image 10. Model Integration Results

CLOSING

In this research, a content-based movie recommendation system has been successfully developed that is able to consider the unique characteristics of each movie to provide more targeted recommendations.

The model has been well tested using the Gradio framework, which allows interactive visualization of recommendation results and demonstrates the effectiveness of the model in various situations. With the success of this development, the resulting recommendation model has great potential to be integrated into various platforms, such as websites or movie streaming services, so as to provide more personalized and relevant recommendations for users.

The use of this model is expected to not only improve the user experience in finding movies that match their preferences, but also become the

foundation for further development of recommendation systems in the future. Suggestions for further research development include:

1. Implement this model across various digital platforms, including websites and mobile apps, to improve the quality of user experience through more relevant and personalized recommendations. This integration is expected to reduce bias in recommendations and provide results that are more in line with each user's preferences.
2. Further develop the recommender system by exploring alternative methods, such as collaborative filtering and hybrid filtering, which can potentially lead to more accurate recommendations. In addition, integration with AI technologies, such as chatbots, can enable deeper personalization through proactive recommendations based on the context of the conversation.

LITERATURE

[1] Alimah, N. L., Putra, P. A., & Sigit, A. 2019. "Rekomendasi Film Berdasarkan Sinopsis Menggunakan Metode Word2Vec". Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol.3. Universitas Brawijaya. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5623 [Diakses pada 02 Mei 2024],

[2] Juhi Dhameliya & Nikita Desai. 2019. "Job Recommendation System using Content and Collaborative Filtering based Techniques". International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-9. DOI:10.35940/ijsce.C3266.099319 [Diakses pada 16 Juni 2024].

[3] Pramarta, A., & Baizal, A. 2022. "Hybrid Recommender System Using Singular Value Decomposition and Support Vector Machine in Bali Tourism." JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), vol. 7, no. 2, pp. 408–418.https://doi.org/10.29100/jipi.v

7i2.2770 [Diakses pada 16 Juni 2024].

[4] Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S., & Venkatesh, B. 2019. “Content-Based Movie Recommendation System Using Genre Correlation.” In Smart Intelligent Computing and Applications, Smart Innovation, Systems and Technologies (pp. 419-427). Springer. https://doi.org/10.1007/978-981-13-1927-3_42 [Diakses pada 24 April 2024].

[5] Rianti, A., Majid, N. W. A., & Fauzi, A. 2024. “MACHINE LEARNING JOURNAL ARTICLE RECOMMENDATION SYSTEM USING CONTENT BASED FILTERING.” JUTI: Jurnal Ilmiah Teknologi Informasi, Volume 22. http://juti.if.its.ac.id/index.php/juti/article/view/1193/500 [Diakses pada 16 Juni 2024].