FPT UNIVERSITY

Film Recommendation Based on TMDB Review

Artificial Intelligence

May 16, 2024

Student:	**ID:**
Truong Dien Phu	SE171385
Nguyen Xuan Vinh	SE161978
Nguyen Phan Nhat Minh	SE171385

Supervisor:
Nguyen Quoc Trung

May 16, 2024

Contents

1 Introduction

1.1 Overview

1.1.1 Project Information

- Project name: Film recommendation
- Group name: Thap Phan

1.1.2 Project team

a.Supervisor

Full name	Email	Phone	Role
Nguyen Quoc Trung	trungnq46@fe.edu.vn trungnq46@fpt.edu.vn	0979350707	Lecturer

b.Team members

Full name	Email	Phone	Role
Nguyen Xuan Vinh	vinhnxse161978@fpt.edu.vn	0373434097	Leader
Truong Dien Phu	phutdse171385@fpt.edu.vn	0947466075	Member
Nguyen Phan Nhat Minh	minhtdse171387@fpt.edu.vn	0889140350	Member

1.2 Background and context

In the current era (21st century), with the development of the internet, everyone can go to any device to watch movies such as retail movies or TV channels. With such a large number of movies, anyone can go online to watch them at any time, but they also face the challenge of choosing the right movie for their preferences. Therefore, there is a need for an on-demand movie recommendation project to help viewers choose their favorite movies by managing personalized content recommendations. The Movie Database (TMDB) is one of the most complete sources of information related to movies, or other reality TV shows. This database has a lot of data, including all types of movies, actors as well as reviews and ratings commented by people. These comments are valuable because they summarize all of the viewers' opinions and insights The team's project aims to create a recommendation system that uses TMDB reviews and meta data to recommend movies to users. By analyzing emotions and other factors in viewer ratings, our system will suggest to viewers movies that they are more likely to enjoy watching, thereby personalizing the viewing experience. and makes movie selection more engaging and less overwhelming.

2 Related work

The previous methods that researchers used in the topic of recommending movies, they had two main methods: Collaborative Filtering method and Content-based Filtering method. Collaborative filtering can select movies based on the user's preferences in the personal information update section to see if they are similar to others, while content-based filtering can select movies by checking the history view or things that the user previously selected. However, both methods have limitations, such as the cold start problem and data sparsity, which make the results unsightly. Recently, Combined Filtering was born, it uses both of the above methods to increase accuracy in suggesting movies to viewers and reduce errors that the other two methods still have. This method is quite popular and is used by large websites such as Netflix and Amazon, especially these two websites are extremely secure and private as well as personalize users. Additionally, machine learning is now very popular and Deep Learning-based Methods have gradually become popular. It will help the model become much better at predicting than other mod-

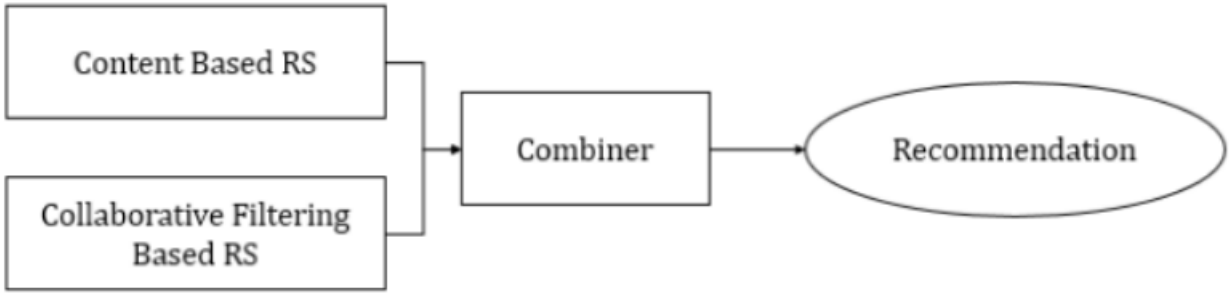


Figure 4: Hybrid Filtering

els

3 Data collection

3.1 Data sources:

TMDB (The movie database) website for movie archive and information regarding production and reviews of movie.This id q Link website: https://www.themoviedb.org/?language=vi

3.2 Data extraction method:

1. Step 1 : Accessing to the website and login to account
2. Step 2: Get the API key by filling in a sheet fill with information regarding what you are doing with said API then wait for the moderator to grant you access to the API key
3. Step 3: Send an API request
4. Step 4: Import necessary packages then create a function call getmoviedetails then also set up getreviewsdetail then initialize logging, set up your API key then proceed to use request to access the website through an URL
5. Step 5: Create a function call gettopmoviesid, this function is used mainly to get the top 10 000 most popular movie on the site and their review.
6. Step 6: Create a function call writetocsv for the extracted data to be converted into a csv file called movies.csv and reviews.csv
7. Step 7: The main function specify the main data that need to be extracted from the website and converted into csv file.During this you need to also set up a rest time for the function, the purpose of this is to respect the extraction limit and not get ban for over extraction.

3.3 Data descriptions

- Description: Data collection contains meta movie data from the top 10 000 most popular movie on the website. This allow the system to recommend the most popular movie and extract meaningful comment from the top 5 highest rated review by user and their ratings
- Format: CSV file

3.4 Features

The dataset includes the following columns, each representing different features with their respective data types:

- id (int) : id of a movie used to categorize and keep track of the film and it's data
- title (int): name of the movie
- vote average(float): The average ratings of movie by users
- : vote count (int): The total number of user who has rated the movie
- budget(float): The estimated amount of money spent on making the movie
- : overview (str): the film synopsis, summarize about the movie

- tagline (str): related one liner about the movie
- genres (str): movie genres
- production company (str): The company that is responsible for the making and publishing of that movie
- production countries (str): The country that the movie originated from
- spoken languages (str): The main language being used in the movie
- cast (str): All the actor and character that they play
- director (str): The movie director

These data give us a comprehensive look at the movie market and it most watched movie and prefer movie by movies goer.

4 Data cleaning and preprocessing

4.1 Read and clean data

An important step in the data cleaning process is checking and handling missing values. We checked the number of missing values in each column and finds that the missing value for some of the column such as ”tagline” are inconsequential and hence do not need to be drop, such rows are useful meta data that helped build the film recommendation system effectively, droping them in such cases will lead to shortage of data. After cleaning up the data we also realize that the keywords column is empty so we also proceed to drop that column.

4.2 Preprocess data

4.2.1 Convert budgets to numbers and classify them

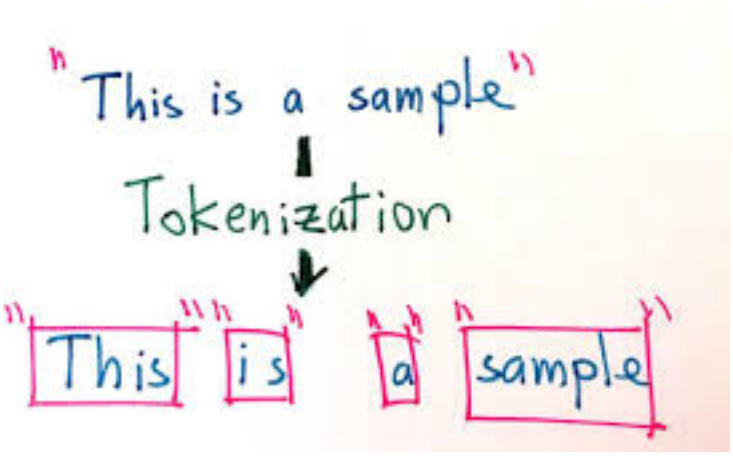
We convert budget values to numbers and handle values that cannot be converted. Invalid values are replaced with 0. We then classify budgets into groups such as ’low budget’, ’mid budget’, and ’high budget’. This allow us to have a clearer look on what said numbers budget number mean in the filming industry. Categorizing them as such also helped in learning about user prefer budget movie. High budget film meaning more studio depended an less creative.Mid budget movie tend to have more creative freedom but less money

4.2.2 Text cleaning

First we try to remove space and comma from the column that we were planning to combine into the tags column . Proceed to convert the id column and review id column into int type so we can combine the 2 dataset based on their id so we can use them as movie recommendation. After that we will make content data in the user review database and turn them into lowercase letter.

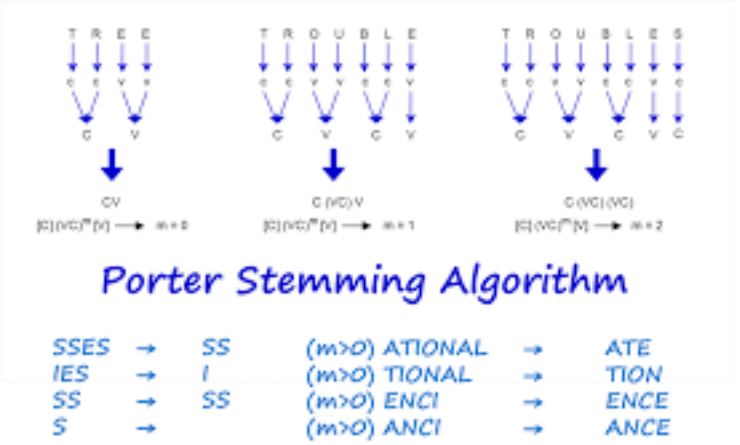
4.2.3 Tokenization

Tokenization is the process of breaking down a text into smaller units called tokens. These tokens can be words, phrases, or even characters, depending on the task at hand. Tokenization is the first step in text preprocessing and is essential for further analysis.The image below demonstrate tokenization



4.3 Stop words and Stemming

After that we will proceed to use Stop Word Removal, Stop words are commonly used words in a language that usually carry less meaningful information for text analysis. These words include articles, prepositions, conjunctions, and some pronouns (e.g., ”the,” ”is,” ”in,” ”and”). Removing stop words helps reduce the dimensionality of the data and can improve the performance of NLP models by focusing on the more meaningful words. Stemming is the process of reducing words to their base or root form. The purpose of stemming is to group together different inflected forms of a word so they can be analyzed as a single item. This can help in reducing the complexity and improving the performance of NLP tasks. The most common stemming algorithm is the Porter Stemmer which is the stemming



technique that we are using in this project.

5 Exploratory Data Analysis (EDA)

5.1 Check the data structure

1. Total number of rows and columns: The data includes X rows and Y columns.Key columns: 'id', 'title', 'overview', 'genres', 'cast', 'tagline', 'director', 'production companies','production countries', 'spoken languages', 'budget', 'sentiment '.

2. Main finding: The data has all the necessary information to perform analysis and movie recommendations.

5.2 Film budget distribution

1. Budget classification: We classify film budgets into three groups: 'low budget', 'mid budget', and 'high budget'.

2. Key findings: The distribution of film budgets is uneven, with many films having low budgets and a few films having very high budgets.

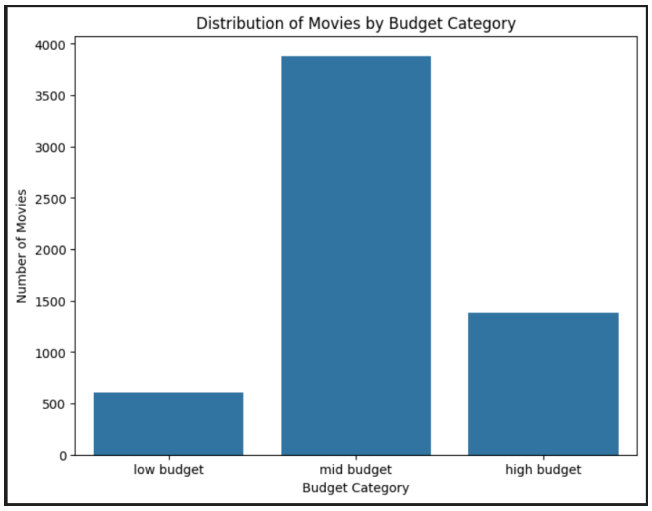


Figure 1: Budget Category

5.3 Analyze film genres

1. Distribution of movie genres: We calculate the frequency of appearance of each movie genre.

2. Key findings: Some popular movie genres such as 'Drama', 'Comedy', and 'Action' appear heavily in the data.

5.4 Calculate Sentiment Score

1. TextBlob Sentiment Analysis: Utilizing TextBlob to compute sentiment polarity for text data, which ranges from -1 (most negative) to 1 (most positive).

2. VADER (Valence Aware Dictionary and sEntiment Reasoner): Employing VADER to evaluate sentiment intensity of text, yielding a compound score that varies between -1 (extremely negative) to 1 (extremely positive).

3. Hugging Face Transformers: Utilizing pre-trained models from Hugging Face Transformers to predict sentiment scores. The input text is truncated to fit the model's maximum sequence length, typically 512 tokens.

4. Combined Sentiment Score: The final sentiment score is a weighted combination of the scores from TextBlob, VADER, and Hugging Face Transformers. Adjusting the weights allows customization to fit specific needs and enhance overall sentiment analysis accuracy.

5. Sentiment Categorization: After obtaining the combined sentiment score, it is categorized into sentiment labels such as 'Horrible Movie', 'Bad Movie', 'Fine Movie', 'Good Movie', or 'Unknown', based on predefined thresholds.

6 Methodology

6.1 TF-IDF Vectorization and Cosine Similarity:

1. TF-IDF (Term Frequency-Inverse Document Frequency) is a popular technique in natural language processing to convert text into numeric vectors, reflecting the importance of words in a document. Cosine similarity is then used to calculate the similarity between these vectors.

2. Advantages: Simple, effective and easy to deploy. TF-IDF and cosine similarity can quickly calculate the similarity between movies based on their text content.

6.2 SentenceTransformer:

1. SentenceTransformer uses advanced deep learning models (like BERT) to generate embedding vectors for text. These vectors help capture deeper semantics of text compared to traditional methods such as TF-IDF.

2. Advantages:Advantages: Ability to capture better semantics and context, helping to improve the quality of movie suggestions when using user-provided descriptions.

6.3 Text features:

1. Tags: We combine information from many different columns (like 'overview', 'genres', 'cast', 'tagline', 'director', 'production companies', 'production countries', 'spoken languages' , 'budget', 'sentiment') into a single column 'tags'. This helps centralize all important information in one place, making it easy to analyze and process.

2. This information represents important aspects of a movie and can help the model better understand the content and context of the movie.

6.4 Emotion analysis:

1. Sentiment Score: We calculate sentiment scores for movie reviews using three methods: TextBlob, VADER, and the RoBERTa model from Hugging Face Transformers. These scores are then combined to create a composite emotional score.

2. Reason for choice: Sentiment analysis helps better understand audience responses to movies,thereby improving the quality of movie recommendations.

6.5 Vectorization and text embedding:

1. TF-IDF: Use TF-IDF to convert text into numeric vectors, helping to calculate similarity between movies.

2. SentenceTransformer: Uses a deep learning model to create embedding vectors, helping tobetter capture the semantics and context of the text.

3. TF-IDF is a traditional, simple and effective method, while SentenceTransformer provides better semantic capture, suitable for detailed descriptions provided by users.

7 Model development

7.1 Model architecture

7.1.1 TF-IDF and Cosine Similarity Model:

1. TF-IDF Vectorization: We use TF-IDF (Term Frequency-Inverse Document Frequency) technique to convert text into numeric vectors. Each word in the text is assigned a weight based on its frequency in the document and its importance in the entire data set.

2. Cosine Similarity: After converting the texts into TF-IDF vectors, we use cosine similarity to calculate the similarity between the vectors. Cosine similarity measures the angle between two vectors, reflecting the degree of similarity between two movies.

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$
$$IDF(t) = log\frac{N}{1+df}$$
$$TF-IDF(t,d) = TF(t,d) * IDF(t)$$

Figure 2: TF-IDF Algorthim

7.1.2 SentenceTransformer model:

1. SentenceTransformer: We use an advanced deep learning model, specifically the all MiniLM-L12-v2 model from the SentenceTransformer library. This model is trained on multiple corpora to create embedding vectors for sentences, helping to capture semantics and context better than traditional methods.

2. Cosine Similarity: Similar to the TF-IDF model, we use cosine similarity to calculate the similarity between the embedding vectors of the movies.

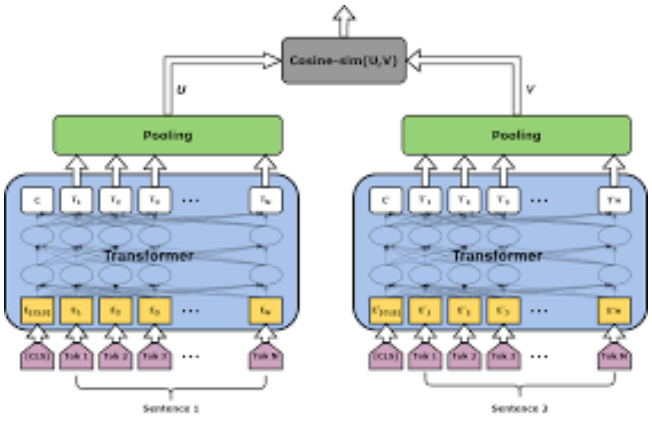


Figure 3: Sentence$_{T}ransformer_{A}lgorithm$

7.2 Training process

7.2.1 Prepare data:

1. Data cleaning: Remove missing values, convert columns to the appropriate format, and normalize text

2. Text preprocessing: Create tags column by combining many different columns, remove whitespace and perform stemming to standardize the text.

7.2.2 Create embedding vector:

1. TF-IDF Vectorization: Use "TfidfVectorizer" to convert movie tags into a TF-IDF matrix.

2. Sentence Transformer: Uses the "Sentence Transformer" model to create embedding vectors for movie tags.

3. Cosine Similarity: Calculate the cosine similarity between TF-IDF vectors and embeddingvectors to measure the similarity between movies.

7.2.3 Movie deployment and suggestions:

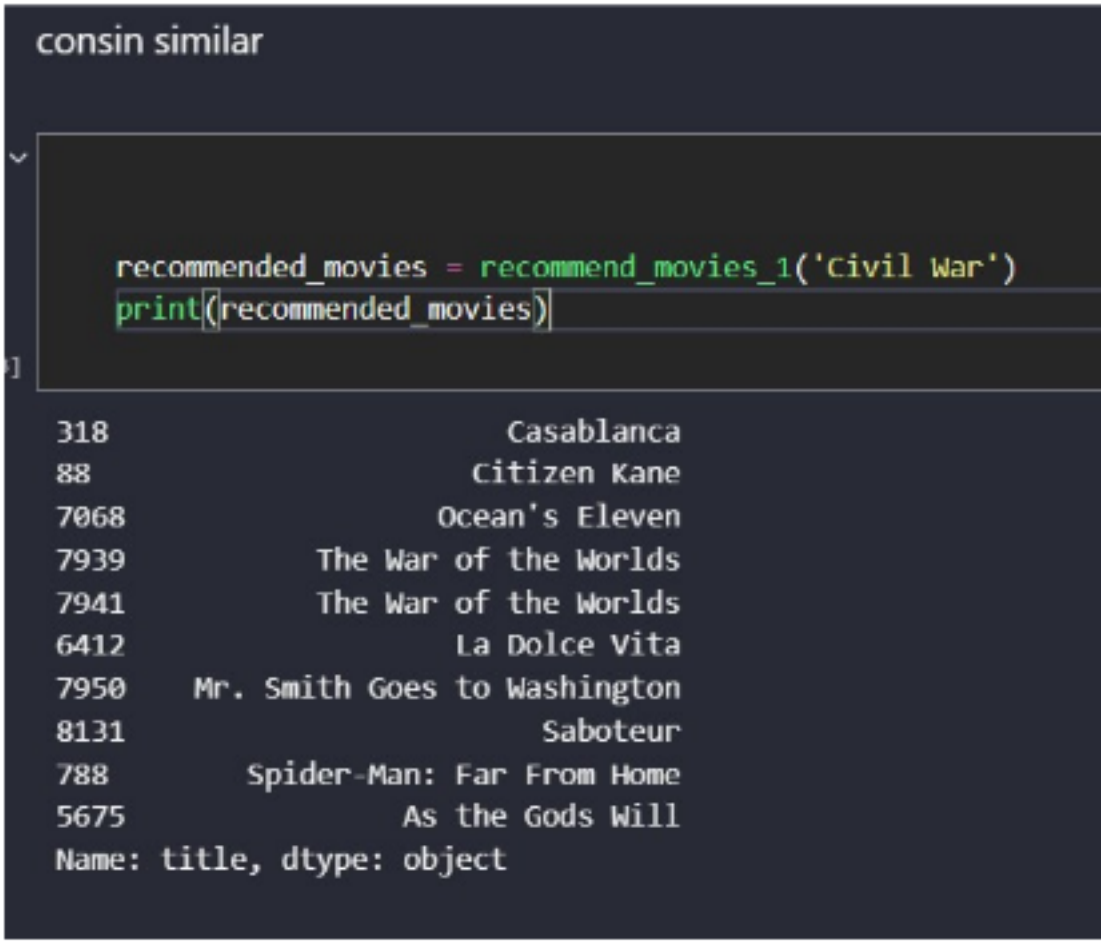
1. Recommendation function based on movie title: Uses cosine similarity to find similar movies based on known movie titles.

2. Recommendation function based on user description: Use "Sentence Transformer" to encode the user description into an embedding vector, then calculate cosine similarity to find similar movies based on this description.

8 Movel Evaluation

8.1 TF-IDF and Cosine Similarity Model

When using the TF-IDF and Cosine Similarity models to suggest movies based on the title ”Civil War”, the results include movies like ”Casablanca”, ”Citizen Kane”, ”Ocean’s Eleven”, ”The War of the Worlds”, and ”Spider-Man: Far From Home”. These movies are not explicitly related to the ”Civil War” theme, suggesting that the model may not capture the deep semantics of the movie title and only focuses on similar keywords.



8.2 SentenceTransformer model

When using the SentenceTransformer model, films are obviously more closely related to the theme of ”Civil War” or historical events and wars. This shows that the SentenceTransformer model can better capture semantics and provide movie recommendations that are more relevant to the input topic.

9 Project management

9.1 Project Schedule

Project phase	Objectives	Tim
1. Planning and designing Project	- Identify goals, ensure accuracy and effectiveness of solution methods	05/052024
2. Collecting material	- Collecting data. - Crawl data from web pages and determine properties	11/05/2024
3. Data preparation	- Delve deeper into NLP to clearly identify user trends and preferences	16/05/2024
4. Model training	- Using NLP combined with machine learning to provide movie suggestions that users want to watch	27/05/2024
5. Testing model and investigation	- To find out the cause and impact of movie on users	11/06/2024
6. Final report	- Final report on the project and give the group's perspective	17/06/2024

9.2 Project Resource Management

Platform	Description	Link
Google drive	- Dataset. -Backup Resources.	
Github	- Code management. - Final publish project.	

10 References

[1] A review of movie recommendation system: Limitations, Survey and Challenges September 2020 ELCVIA Electronic Letters on Computer Vision and Image Analysis

[2] MOVIE RECOMMENDATION SYSTEM February 2024