Sheffield Hallam University

MSc Artificial Intelligent

Nguyen Quang Anh – Student ID: c3005993

Applied Artificial Intelligence

Movie Recommendation

# I. Introduction

## 1. Recommendation System

Most Internet users have come across a recommendation system in one way or another. In the entertainment industry, selecting suitable content has become a challenge for many due to the diverse and abundant sources of films today. For entertainment service companies like YouTube or Netflix, recommendation systems are crucial as they can enhance user engagement and retention.

The entertainment service companies scenario is useful for discussing several aspects of such software systems. First, notice that we are talking about personalized recommendations – in other words, every user sees a different list depending on his or her tastes. In contrast, many other portals may simply inform you of their top-selling items. Theoretically, we could interpret this information as a sort of impersonal watching recommendation as well and, in fact, very popular movies will suit the interests and preferences of many users. Still, there will be also many people who do not like to watch Harry Potter despite its strongly popular – in other words, for these people, recommending top items is not very helpful. (Jannach et al., 2010, p. 1)

So, the movie recommendation system not only helps users save time when searching for suitable movie content, but also provides a personalized entertainment experience. And in this assessment, I will introduce how the movie recommendation system works to meet the above requirements.

## 2. Dataset

The dataset I use for this project is Movie and Rating dataset. The ‘movies.csv’ contains 10325 movies, each movie has a movieId, a title and genres. The ‘ratings.csv’ contains 668 user, each user has a userId, and creata a rating connect with movie through movieId key.

A screen shot of a movie list

Description automatically generated A screen shot of a computer

Description automatically generated

A graph with blue lines

Description automatically generated

There are total 105339 rating had been create. On average, 1 user creates 157 ratings and the graph descripes the ratings each user creates.

# II. Technology

## 1. Content-based Recommendation

As the name suggests, content-based recommender systems make use of the content of an item’s description to predict its utility based on a user’s profile. Content-based recommender systems aim to recommend items that are similar to items that have previously interested in a specific user. Different item properties are extracted from documents/descriptions. For instance, a movie can be represented by attributes such as genre, the director, writer, actors, storyline, etc. (Qian Zhang · Jie Lu · Yaochu Jin, 2020, p.440)

In my dataset, movie has a title and associated with the key movieID, so only genre is an attribute that I can use for content-based recommendation.

A screen shot of a computer program

Description automatically generatedA screenshot of a computer screen

Description automatically generated A screenshot of a computer

Description automatically generated

There are a total of 20 different types of genres, and a movie can have multiple genres, so I divided them into separate columns, assigning the value 1 if the movie has that genre, 0 for otherwise and use them for feature space.

## 2. Collaborative filtering-based Recommendation

In contrast to content-based recommender systems, which are independent of other users but dependent on a user’s personal historical records, CF-based recommender systems infer the utility of an item according to other users’ ratings. Today, CF is still the most popular technique applied in recommender systems. The basic assumption underpinning the CF technique is that users who share similar interests will consume similar items, so a system using the CF technique relies on information provided by users who have similar preferences to the given user. (Qian Zhang et al, 2020, p.441)

A classic scenario in CF is to predict a user’s ratings on unconsumed items from a user-item rating matrix (Qian Zhang et al, 2020, p.441).

A screenshot of a computer

Description automatically generated

After creating the matrix as above, I calculate the average value of each column (the value will later be added to the final prediction result to standardize on a 5-point rating scale) and take them as central values, fill NaN result with 0.

A screenshot of a computer

Description automatically generated

( Challenge:

According to the user-item rating matrix, the sparsity is high (98,47%). In other words, the number of empty cells over the number of cells with data and this may affect the calculation process and result).

A screen shot of a computer program

Description automatically generated

## 3. Difference Between Content-Based Filtering and Collaborative Filtering

Content-based filtering is like recommending content based on the content of the movies you like. Collaborative filtering is like recommending content based on what other people with similar preferences have liked. (KHARWAL, 2023)

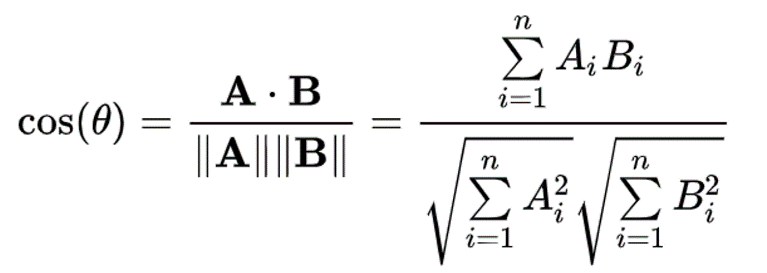
A screenshot of a computer

Description automatically generated

## 4. Cosine Similarity

Cosine similarity is a similarity rate the calculation obtained from the cosine angle multiplication of two vectors being compared, because the cosine 0 degree is 1 and less than 1 to the value of another angle, then the value of the similarity of the two vectors are said to be similar when the value of the cosine similarity is (Lahitani, Alfirna Rizqi et al, 2016, p.3)

Cosine similarity formula between two vectors A and B is:



The result closer to 1 implies high similarity; closer to 0 implies low similarity and closer to -1 implies high dissimilarity (opposite directions).

# III. Implementation

## 1. Content-based Recommendations

### a. Find similar movies

Goal: Given a movieID as input, the system will return movies similar to that movie based on their genres as features.

The steps are as follows:

Split genres into separate columns and use them as features.A screenshot of a computer

Description automatically generated

Calculate the similarity between movies by applying the cosine similarity formula using genres as features.A group of text boxes

Description automatically generated with medium confidence

Input movieID, retrieve the top k movies with the highest scores from the similarity table as suggestions.

### b. Suggest for user base on content

Input: userId, Output k suggested movies for the user.

Goal: Based on the top-rated movies of the given userID, create a unique profile for that userID. Then, calculate the cosine similarity between this profile and all the remaining unseen movies. Finally, suggest the top k movies with the highest similarity scores.

The steps are as follows:

Similarly, calculate the similarity between movies as described earlier.

Create the user profile:

Retrieve the top-rated movies by the user:

- If the user has rated more than 30 movies, select the 15 highest-rated movies.  
- If the user has rated fewer than 30 movies, select the top 30% highest-rated movies.

A screenshot of a computer

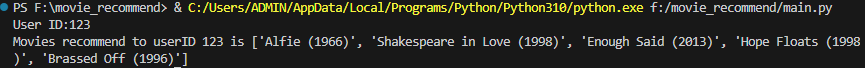
Description automatically generated

Compute the mean similarity score between these top-rated movies and all other movies to create the user profile.

A screenshot of a computer

Description automatically generated

Calculate the cosine similarity between all unseen movies and the user profile. Retrieve the top k movies that are most suitable for the user based on this similarity.



## 2. Collaborative filtering-based Recommendations

The steps are as follows:

Create a user-item rating matrix based on the rating user had made.

A screenshot of a computer

Description automatically generated

Calculate the mean value and center it around the mean, then fill NaN values with 0.

A screenshot of a computer

Description automatically generated

Compute the cosine similarity between users, taking ratings for movies as the feature. So, each user has 10327 features (the number of movies in dataset), and use them to calculate the similarities using cosine similarity.

A screenshot of a computer

Description automatically generated

After that, with input is userID:

Find the k neighbors with the highest similarity to the input userID.

For a movie not been seen by the user, calculate the mean score of the k neighbors and add it to the initial mean of the userID (because I had calculate the mean value and center it around the mean) to obtain the predicted score of the user for that movie on 5-rating scale.

A screenshot of a computer program

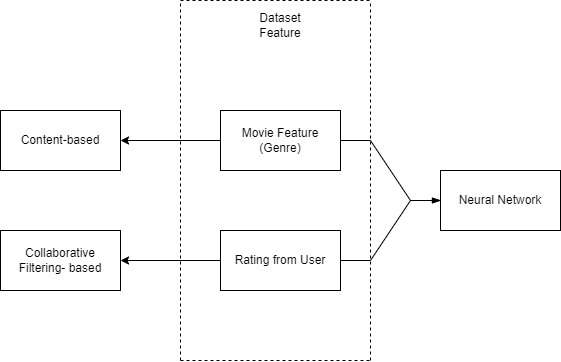
Description automatically generated

Loop for all unseen movie ID, then top movies with the highest scores will be suggested to the user.

# IV. Neural Network

In the two approaches mentioned, with the content-based approach, I only utilize movie features – genre – to find similarities between movies and provide recommendations based on that. On the other hand, with the collaborative filtering approach, I use user ratings for movies to build a similarity matrix between users, and then provide recommendations from other similar users.

So, I want use hybrid approach by using both content-base (movie feature – genre) and rating by user.



To do that, I use a neural network built on TensorFlow Recommenders (TFRS).

## 1. TensorFlow Recommenders (TFRS)

TensorFlow Recommenders (TFRS) is a library for building recommender system models. It helps with the full workflow of building a recommender system: data preparation, model formulation, training, evaluation, and deployment.

It's built on Keras and aims to have a gentle learning curve while still giving you the flexibility to build complex models. (TensorFlow Recommenders, n.d.)

Advantages:

* Ability to combine features between user ratings and movie attributes.
* Avoids data sparsity issues.

Disadvantages:

* Limited to training with the provided dataset; requires retraining when new movies or new ratings are introduced to be applicable.

## 2. Feature Extraction

Combining all available information from the dataset, I use ratings and genres to create features. There are a total of 105339 ratings from 668 users for 10325 movies.

A screen shot of a computer

Description automatically generated

For each rating, it will include the userID of the rater, movieID of the rated movie, rating score, title, and genre of the movie. Excluding the movie title as its information is not efficient, other attributes will be used for the Neural Network.

A screenshot of a computer

Description automatically generated

## 3. Create and Train Model

Initially, I followed the process proposed by Muhammad Faarisul Ilmi and train the model. (Ilmi, 2023)

The train/test split is 95339/10000 (around 90/10) using the features mentioned above.

Result model:

A screen shot of a computer

Description automatically generated

After training, the model will provide rating scores for each movie for a user input, and then select the top 5 movies with the highest scores as recommendations for that user.

A screenshot of a computer

Description automatically generated

The model predicts a rating score for a specific movie, and then scales it back to a 5-point rating scale.

A screenshot of a computer

Description automatically generated

As we can see, for example, for User 123, who watches many movies in the Comedy genre, so I would expect the prediction for their rating of the movie Toy Story, which also the Comedy genre, to be high.

## 4. Fine-tuning

Using an Embedding layer from the TensorFlow library to create an embedding layer in a neural network model. Specifically, this Embedding layer is used to generate embedding vectors for features in the dataset to serve as inputs for the Neural network.

I build a simple Neural network for the task:

Input 1 dense layer with a size of 256, using the ReLU activation function.

Hidden layer consists of 1 dense layer with a size of 128, using the ReLU activation function.

Output is a scalar layer, and it serves as the predicted score.

Optimizers: Adagrad - Learning rate: 0.1

With epoch number is 10, the model took 20 minutes to train.

## 5. Evaluation

My output evaluation for this model as follows:

A grey background with white text

Description automatically generated

Accuracy: Accuracy is a metric that measures how often a machine learning model correctly predicts the outcome.

Top-k parameters: The K parameter is the evaluation cutoff point. It represents the number of top-ranked items to evaluate. (Evidently AI Team, n.d.)

The dataset contains 10327 movies, so I would take 100 for k parameters (about top 1%).

RMSE (Root Mean Squared Error): It shows how far predictions fall from measured true values using Euclidean distance. The lower the RMSE, the more accurate the model.

# V. Ethical Implications

Ethical implications for a movie recommendation system can involve several moral and ethical considerations, as well as issues related to privacy and fairness. Here are some ethical issues that may arise:

Biased Recommendations: One of the ethical issues of recommender systems is the potential for bias and discrimination. Bias can arise from the data, the algorithms, or the users themselves, and can lead to unfair or inaccurate recommendations that favor or exclude certain groups, opinions, or values. (Linkedin, n.d.).

Privacy Concerns: Collecting personal data from users to generate movie recommendations can raise privacy concerns. The use of personal data without user consent or in violation of privacy regulations can be controversial.

Legal issues: A issue of recommender systems is the legal implications they have in different jurisdictions and contexts. Recommender systems may be subject to various laws and regulations that govern the collection, processing, storage, and sharing of personal data (Linkedin, n.d.).

# VI. Conclusion

In this project, I have successfully implemented a basic movie recommendation system with content-based, collaborative filtering based approaches, and create a simple Neural Network for the task.

While the project may not have achieved its maximum potential, it has provided me with invaluable insights into applying machine learning and AI techniques to address real-world challenges. I am deeply appreciative of the opportunity to explore and gain knowledge from Applied Artificial Intelligence module and other University's modules.

# VII. References

(n.d.). Retrieved from TensorFlow Recommenders: https://www.tensorflow.org/recommenders

Baker, K. (2005). Singular value decomposition tutorial. *The Ohio State University Volume 24*, 22.

Evidently AI Team. (n.d.). Retrieved from https://www.evidentlyai.com/: https://www.evidentlyai.com/ranking-metrics/evaluating-recommender-systems#evaluation-principles

Ilmi, M. F. (2023, 12 27). Retrieved from Kaggle: https://www.kaggle.com/code/mfaaris/content-based-and-tensorflow-recommender-system/notebook?scriptVersionId=156678721

Jannach, D. and Zanker, M. and Felfernig, A. and Friedrich, G. (2010). *Recommender Systems: An Introduction.* Cambridge University Press. Retrieved from https://books.google.co.uk/books?id=eygTJBd\_U2cC

KHARWAL, A. (2023, April 20). Retrieved from https://thecleverprogrammer.com/: https://thecleverprogrammer.com/2023/04/20/content-based-filtering-and-collaborative-filtering-difference/#:~:text=Content%2Dbased%20filtering%20is%20like,with%20similar%20preferences%20have%20liked.

Lahitani, Alfirna Rizqi and Permanasari, Adhistya Erna and Setiawan, Noor Akhmad. (2016). 2016 4th International Conference on Cyber and IT Service Management. *Cosine similarity to determine similarity measure: Study case in online essay assessment*, 1-6. doi:10.1109/CITSM.2016.7577578

*Linkedin*. (n.d.). Retrieved from What are the potential risks and challenges of recommender systems in different domains and contexts?: https://www.linkedin.com/advice/3/what-potential-risks-challenges-recommender#:~:text=One%20of%20the%20ethical%20issues,groups%2C%20opinions%2C%20or%20values.

Qian Zhang · Jie Lu · Yaochu Jin. (2020). Complex & Intelligent Systems. *Artificial intelligence in recommender systems*, 440. Retrieved from https://link.springer.com/article/10.1007/s40747-020-00212-w?fromPaywallRec=true