|  |
| --- |
| **Movie Recommendation System** |

**Bing Syuan WangMichelle Liu**

Department of ECE Department of ECE

University Washington University Washington

wben@uw.edu mcliu@uw.edu

**Abstract**

We create a movie recommendation system based on personal habits. We proposed serval methods, including ratings, overview, features, logistic regression, and MLP. The result is positive, and the recommended lists are credible.

**1 Short introduction**

Many OTT (Over-the-top) media services provide recommendation system services, including Netflix, HBOMAX, Disney+, Hulu, etc. The recommendation is usually based on several factors, such as release date, view, popularity genre, or personal habits. Sometimes, the system recommendation is also affected by the manufacturer's sponsorship or the company's other factors. Therefore, we want to create a system based on personal behavior, such as watching history.

**2 Technical descriptions, including algorithm**

**2.1 Recommend by Rating**

We want to use IMDB's formula to calculate the data result of TMDB. IMDB's rating is based on this formula as follows,



R = average for the movie (mean); v = number of votes for the movie; m = minimum votes required to be listed; C = the mean vote across the whole report

**2.2 Recommend by Overview**

First, we use the overview from the TMDB database. Second, using the TFI-IDF to get a vector matrix of 4803\*20987. Third, we do cosine similarity to compare the score between each movie. After we get the score, the highest one would be our recommendation.

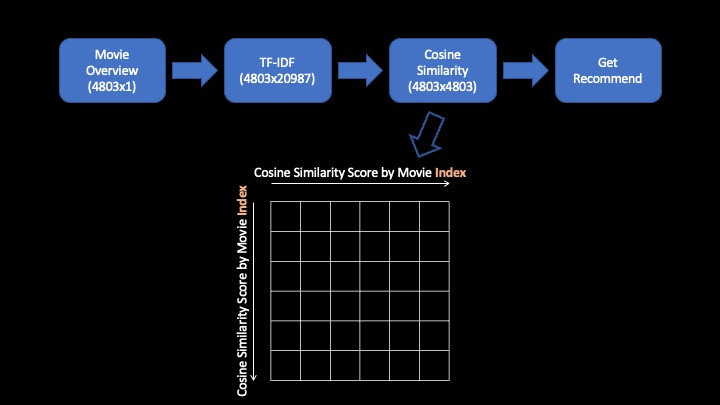


Figure 1: Recommend by Overview

**2.3 Recommend by Title, Cast, Director, Keywords, Genres, Overview** We turn Title, Cast, Director, Keywords, Genres, and Overview to an array list, do the CountVectorizer, Cosine Similarity, and recommend.

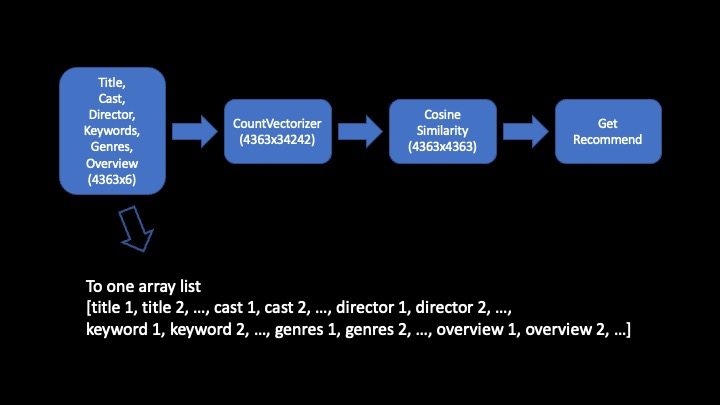


Figure 2: Recommend by Title, Cast, Director, Keywords, Genres, Overview

**2.4 Recommend by Logistic Regression**

We take CountVectorizer as our X. Since The data do not have a label. Therefore, we create Y by ourselves. Then, we take the second highest cosine similarity score as our Y.

The result of accuracy is only 0.13. Although the accuracy is not high, this method is feasible.

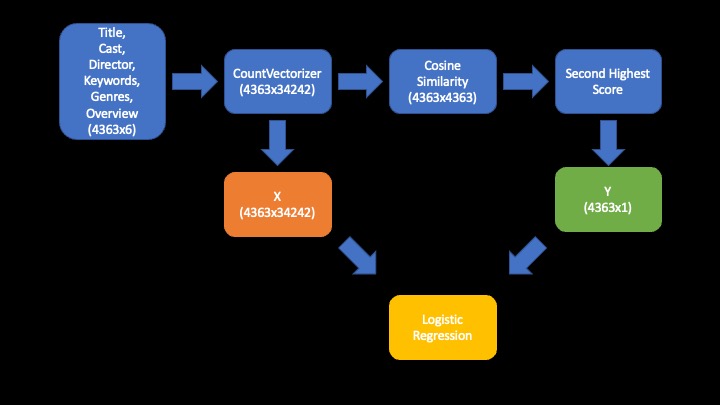


Figure 3: Recommend by Logistic Regression

**2.5 Recommend by MLP (Multi-layer perception)**

MLP (Multi-layer perception) uses neural networks to do deep learning.

The source of the first layer is CountVectorizer. It has 34242 features. There are four hidden layers in the middle, and all their activation functions are ReLU. The final layer is the output layer. Its activation function is Softmax.

The output layer has 4363 output classes from our 4363 movies, totaling 19840907 parameters.

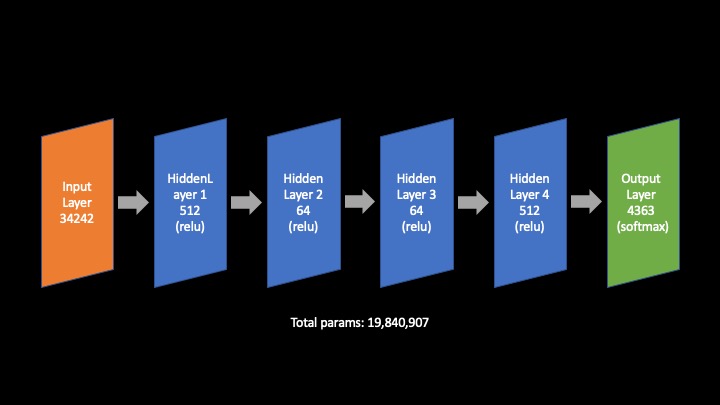


Figure 4: Recommend by MLP (Multi-layer perception)

**3 Experimental results**

**3.1 Recommend by Rating**

The result of the top ten TMDB databases which use the IMDB's formula, and at the column on the right side, which is the actual ranking of IMDB.

The two databases have repetitive ranking when using the same formula, which means the IMDB's formula is credible.

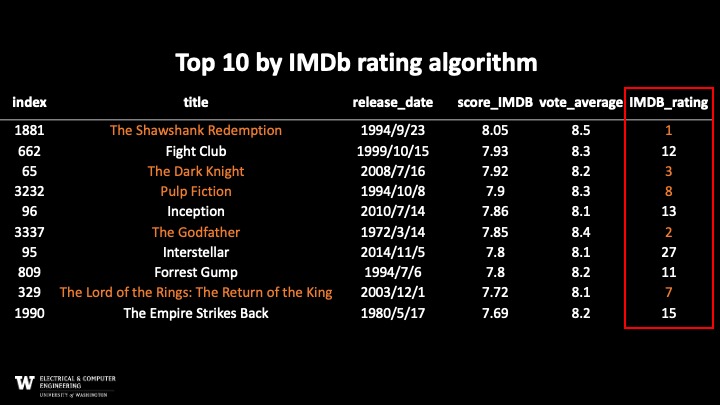


Figure 5: The result of the rating

**3.2 Recommend by Overview**

We use “The Lord of the Rings: The Fellowship of the Ring” as the original movie. The following picture shows the top ten recommendations.

The top five are “The Lord of the Rings” series, and the rest is recommended based on some keywords.

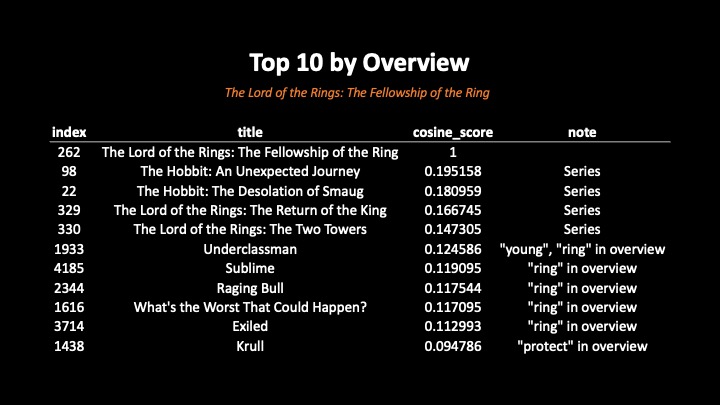


Figure 6: The result of the overview

**3.3 Recommend by Title, Cast, Director, Keywords, Genres, Overview**

The input movie is “The Lord of the Rings: The Fellowship of the Ring”. We can know the relationship from the note part that shows why it is recommended.

一張含有 文字 的圖片

自動產生的描述

Figure 7: The result of Title, Cast, Director, Keywords, Genres, Overview

**3.4 Logistic Regression**

We input two of “The Lord of the Rings” movies, and it recommends “The Lord of the Rings: The Two Towers” movie.



Figure 8: The code of logistic regression first example



Figure 9: The result of logistic regression first example

We input three random movies, and the predicted movie is based on Genres. Here, the genre is drama.



Figure 10: The code of logistic regression second example



Figure 11: The result of logistic regression second example

We input “The Matrix”, “The Matrix Reloaded”, “Speed”, and “Point Break” as four movies. It recommends “The Matrix Revolutions” because they are all performed by 'Keanu Reeves.'



Figure 10: The code of logistic regression third example



Figure 10: The code of logistic regression third example

**3.4 MLP (Multi-layer perception)**

The Loss and accuracy are shown in Figure 11. Training is getting better, but validation is not.

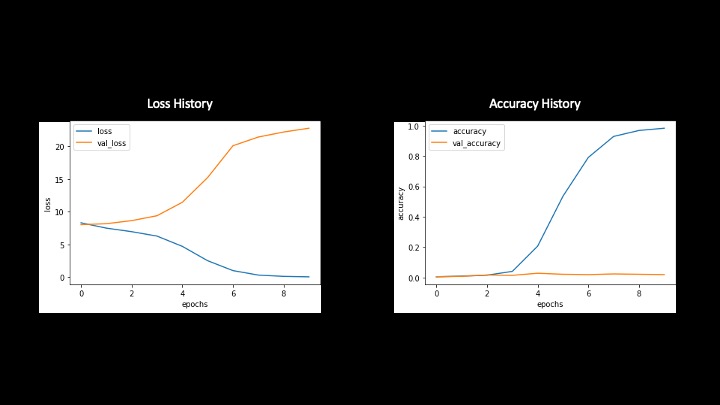


Figure 11: Loss history and accuracy history

We input two of “The Lord of the Rings” movies. The result is “The Hobbit”. However, the result is different from logic regression, but they come from the same series.



Figure 12: The code of MLP first example



Figure 13: The result of MLP's first example

We input three random movies for the MLP model, which considers not only genres but also uses the keywords of overview.



Figure 14: The code of MLP second example



Figure 12: The result of MLP's second example

We input “The Matrix”, “The Matrix Reloaded”, “Speed”, and “Point Break” as four movies. It recommends “The Matrix Revolutions”, and the same as Logistic Regression because they are all performed by 'Keanu Reeves. '



Figure 12: The code of MLP's third example



Figure 12: The result of MLP's third example

**4 Discussion of results, strengths/weaknesses, what worked, what didn't**

The advantage is that this model does not need to be labeled in advance. Our target is generated by the program. Also, others recommendation systems in the reference data are recommended by the title of the movie as an index, but our system can use all features. It can generate predicted features by itself, so it can be used more flexibly for films that are not limited to the database

The disadvantage is that because there are no labeled targets, it is difficult to judge whether our model is good or bad.

**5 Future work**

We have several expectations of our future work. First, we want to add history weight to the model, and the standard would be the closing time, the higher weight. Secondly, adding weight based on the feature affects the final score. Third, the results output of MLP are all probabilities; traditionally, we would only get the highest scores, the predicted movie, of the classification results. Therefore, we would want to add multiple recommendations in the future. Fourth, the accuracies of logistic regression and MLP are not ideal because we take cosine similarity as y, but the result would not be the same as Y. Thus, we would like to find a better solution to solve this problem.

**References**

[1] Sajal Halder & A.M. Jehad Sarkar & Young-Koo Lee. (2012) Movie Recommendation System Based on Movie Swarm. *2012 Second International Conference on Cloud and Green Computing*

[2] P. Kirubanantham & A. Saranya & D. Senthil Kumar. (2021) Convolutional Recommended Neural Network system based on user reviews for movies. *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*

[3] Yeo Chan Yoon & Jun Woo Lee. (2018) Movie Recommendation Using Metadata Based Word2Vec Algorithm. *2018 International Conference on Platform Technology and Service (PlatCon)*