Movie Recommendation System: Machine Learning Perspective

Gaurab Baral
Dr. Junxiu Zhou
School of Computing and Analytics
Northern Kentucky University

Abstract

Movie recommendation systems aim to enhance user engagement and satisfaction by providing relevant movie options which plays a pivot role in current movie industry. Machine learning techniques have shown promise in revolutionizing various industries. This research not only provides an overview of movie recommendation from the machine learning perspective but also presents a performance comparative study about them. Specifically, we reviewed the trend of current movie recommendation systems and identified and implemented six widely used machine learning models to further explore their recommendation performance. This exploration helped us understand how various approaches work to provide more precise and diverse movie recommendations.

Movie Dataset

The movie dataset utilized in this research work is MovieLens20M, sourced from the MovieLens website. There is 20 million ratings from a total of 20231 user on 31983 movies. A sample of user and movie tables is shown below(Table 1 and 2):

Table1: A sample of movie data

movieId	title	genres
2	Jumangi (1995)	Adventure|Children|Fantasty
131258	The Pirates (2014)	Adventure

Table2: A sample of user data

userId	movieId	rating	timestamp
1	296	5	1.147880e+09
1	306	3.5	1.147869e+09

Overview Existing Works

To get a better sense of movie recommendation systems from the machine learning perspective, we searched "movie recommendation machine learning" on Google Scholar webpage and analyzed the main trend (see Fig. 3). Among them, we focused six widely used machine learning models, i.e., matrix decomposition, K-NN, SVD, SVM, MLP, and Autoencoder.

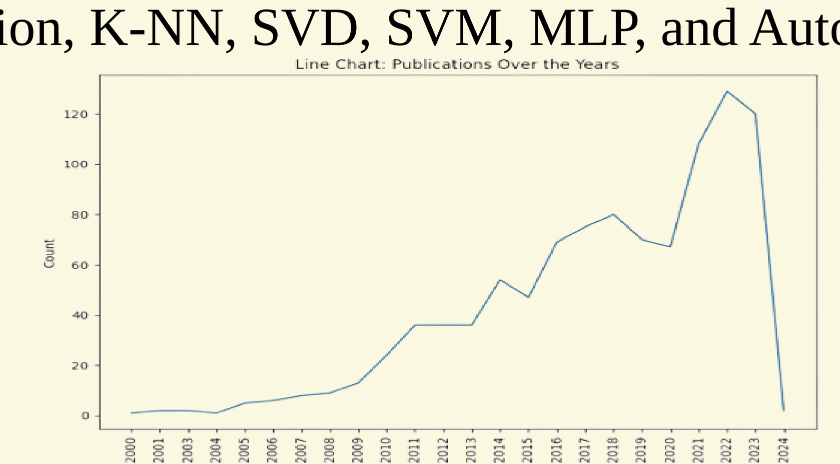
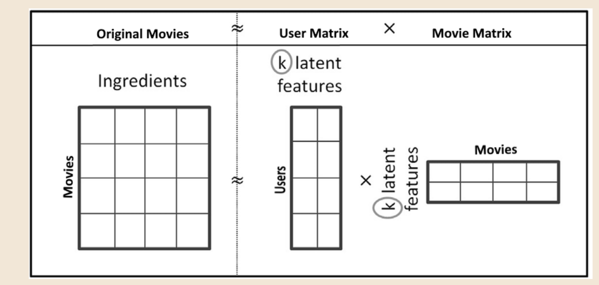


Figure 3. Publication trends on movie recommendation related to ML

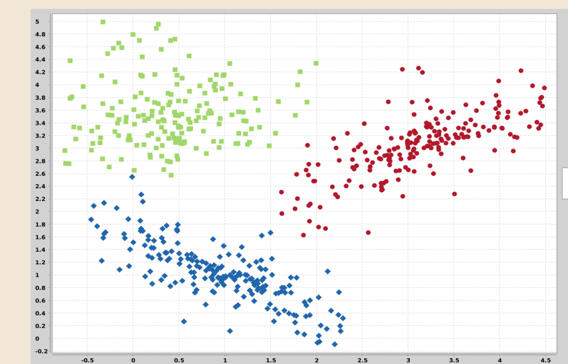
Machine Learning Models

To gain a better insights on how ML models contribute to movie recommendation systems. We selected and implemented six widely used ML models:

- **Matrix Decomposition:** This technique breaks down a matrix into its constituent parts. It is often used in collaborative filtering to discover latent features between users and items.



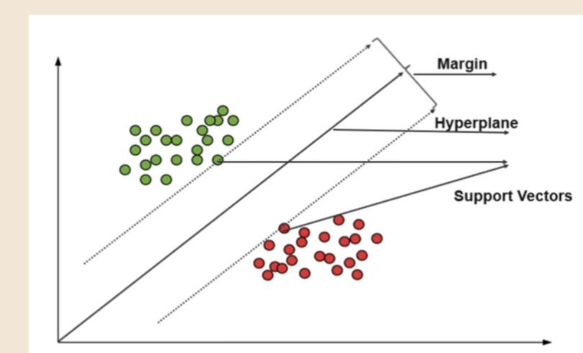
- **KNN (K-Nearest Neighbors):** This is an instance-based learning algorithm. KNN classifies a new instance based on the majority class of its 'k' nearest instances in the feature space.



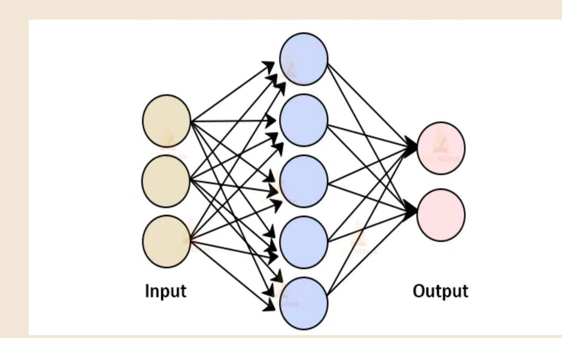
- **SVD (Singular Value Decomposition):** This is a factorization method used to reduce the dimensionality of data. It enhances interpretability and reduces noise.

$$A = U\Sigma V^T$$

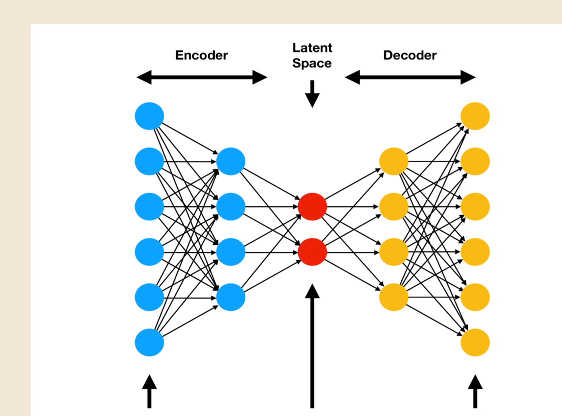
- **SVM (Support Vector Machine):** This is a supervised learning model that performs classification by finding the hyperplane that best separates classes of data with maximum margin.



- **MLP- ANN (Multi-Layer Perceptron Artificial Neural Network):** This is a class of feedforward artificial neural network. It consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one.



- **Auto-Encoders:** These are supervised neural networks used for dimensionality reduction or feature learning. They encode input data as an internal fixed-size representation in reduced dimensionality



Numerical Results

We implanted all six ML models on a Intel i7 CPU@2.6GHz 16GB RAM desktop using Python and Sci-kit Learn package. The performance of the models are measured using root mean squared error (RMSE). The test time also recorded and presented in Table 3. The Auto Encoder has the best RMSE performance, and ANN built with MLPREGRESSOR has the fastest test time.

Table3: Model performance comparison in terms of RMSE and test time

Algorithms	RMSE	Test time (s)
ANN-1	0.81145	47
ANN-2(MLPRegressor)	1.1045	**12**
KNN	13.5096	15
SVD	0.8149	25
SVM(kernel = linear)	1.58	47
Auto Encoders	**0.62351**	22

Recommendation Results Comparison

To analyze the effectiveness of the models, we compare the recommendation results with the recommendation generated from ChatGPT, i.e., Psycho (1960), Rebecca (1940), Rope (1948), North by Northwest (1959), Felicia's Journey (1999) for the movie "*Vertigo (1958)*" which is a 1958 American psychological thriller movie directed and produced by Alfred Hitchcock.

Table 4: Model recommendations for the movie "Vertigo (1958)"

Simple ANN
The Secret of Roan Inish (1994)
The Crow: City of Angels, (1996)
Mina Tannenbaum (1994)
Cinema Paradiso (1989)
Strangers on a Train (1951)

KNN
The Big Sleep (1946)
Notorious (1946)
Rebecca (1940)
Dial M for Murder(1954)
North by Northwest (1959)

SVD
Black Dog (1998)
Mighty The Mighty (1998)
Beefcake (1999)
Felicia's Journey (1999)
1969 (1988)

ANN(MLP Regressor)
Rope (1948)
Topaz (1969)
Trouble with Harry, The (1955)
I Confess (1953)
Stranger on a Train (1951)

AutoEncoder
Strangers on a Train(1951)
Rebecca (1940)
Dial M for Murder(1954)
Notorious (1946)
The third Man(1949)

SVM
A Study in Terror (1965)
From Hell (2001)
The Lodger (1944)
Mystery of Edwin Drood (1935)
The Hound of the Baskervilles (1959)

As can be observed above,

- The six algorithms present somewhat comparable recommendations, with autoencoders and KNN yielding the closest matches to those generated by ChatGPT.
- Meanwhile, the ANN algorithms rank towards the lower end in terms of their recommendations, albeit still providing some useful suggestions to users.
- KNN although has a worse RMSE is useful in recommendation engines.

In addition, different models may have different focus in learning to recommend movies, that could provide diversity to recommendation systems, indicating hybrid model based recommendation system may contribute more to a more personalized recommendation result.

Future Prospects:

For future work, we plan to deploy these recommendation systems in our movie recommendation site and use the following business metrics to better evaluate the performance of the systems.

1. **Click-Through Rates**: $\frac{Ratio\ of\ Users\ who\ click\ on\ the\ movie}{Ratio\ of\ Total\ Users\ who\ watched\ the\ movie}$
2. **A and B testing**: Compares different versions of the recommendation algorithm to find the most effective one.
3. **User Engagement**: Tracks user actions like rating movies, adding to watch list, or sharing recommendations.
4. **Take Rate:** Measures the percentage of recommended movies or series that users actually watch, indicating the effectiveness of recommendations (inspired by Netflix).

Hybrid models are always next in recommendation systems which combine different recommendation techniques to deliver more accurate and personalized recommendations.

I would like to express my sincere gratitude to the Local Initiative For Excellence LIFE Foundation for their generous support and funding of this research project.