

23rd October 2023

Movie Recommendation Systems using AI

Akshan Bansal
High School Student

Shlok Khare
High School Student

Keywords

Artificial Intelligence(AI), Cosine Similarity Method(CSM), convolutional neural networks(CNN) , Artificial Neural Network(ANN), Recurrent Neural Network(RNN), Radial Basis Function Neural Networks(RBFNN), Keras, AI models

Problem Statement

In today's world, there is an ocean filled with movies. A particular viewer gets lost, given so many options, wasting their time searching for the movies they like, only if there were a recommendation system to save their time.

Research Objective

To analyze various models and determine which is the most efficient model for a movie recommendation system. Keeping the models in the most simple form with Keras. Hence giving the user their preferred movies and saving their time.

Introduction

This research analyses which model would be the most efficient for a movie recommendation system. We have compared a cosine similarity model (CSM) and 4 Neural network models - ANN, CNN, RNN, and RBFNN.

We have a cleaned dataset of 45,000 movies, however, we have used a sample of it only. Using the sample as our main dataset each run we got randomized data, we created a user dataset that stores info for user_id, movie_list, and genre-lists, ensuring it is not biased.

Using the user dataset, we compute each user's top three genres they prefer. Then using all of these data sets we train, compile, and apply the various models to recommend 10 movies and their performance. We used a different user dataset for the cosine similarity model because we realized it wouldn't be efficient. But the same ones for the neural networks.

Lastly, we compared the performances of each neural network and cosine similarity model (CSM) to determine the best model.

We did 6 runs, 1 run on a sample of 5000 movies and 5000 users each of 50 movies. To confirm the obtained best model, we ran the code 5 more times but this time with a sample of 1000 movies and 1000 users for each of 50 movies.

Significance

This research helps other recommendation systems to make them more efficient. This saves our users time searching for a movie in today's world's vast pool of movies.

Methodology

To complete this project, we used the Python Programming language and had to study how to implement AI deeply We used various AI models like CSM, CNN, ANN, RNN, RBFNN.

We used Kaggle to get our dataset which is:

<https://www.kaggle.com/datasets/ashishjangra27/imdb-movies-dataset>

We ran our code for 5,000 users once and 1,000 users 5 times to get the best accuracy. We ran various tests for each model and concluded that the ANN best fit our data set with respect to time and performance.

Code

To review the code please use this link:

<https://github.com/akshanbansal06/Movie-Recommendation-Systems-using-AI/blob/main/MLProject.ipynb>

Results

We ran the various models with different datasets to get a concrete result. We ran the models on six different datasets. We labeled the first run as 5000-m-u. This was our main run. In this run, the models were processed over datasets of 5000 movies, and 5000 users with the history of the last 50 movies they had seen. The other runs were to verify if the result found by this run was accurate. These runs were labeled 1000-m-u-x; x, here, is {1,2,3,4,5}. In this run, the models were processed over datasets of 1000 movies and 1000 users with the history of the last 50 movies they had seen.

The data that we received after running these runs gave us the accuracy of recommending the movies to the users and the time it took it to process by each model. As the cosine similarity model is purely written code that is not complying, it doesn't give us the time it took. Therefore the cell with that information is marked as NA.

Accuracy.

- **Given as percentages**

Run	CSM	CNN	RNN	ANN	RBFNN
5000-m-u	78.94	94.32	97.55	98.87	44.32
1000-m-u-1	76.58	90.35	96.4	91.85	90.88
1000-m-u-2	79.67	97.94	95.72	98.67	97.38
1000-m-u-3	76.75	91.47	96.38	95.26	91.3
1000-m-u-4	77.22	96.63	95.56	96.89	95.52
1000-m-u-5	76.87	94.1	96.2	97.34	89.82

If the average is displayed for each,

Accuracy	CSM	CNN	RNN	ANN	RBFNN
Average	77.67	94.14	96.3	96.48	84.87

Even though by average we can say ANN is the best, looking at all the data, RNN is also a model we can use to get the same accuracy.

Time Taken Table.

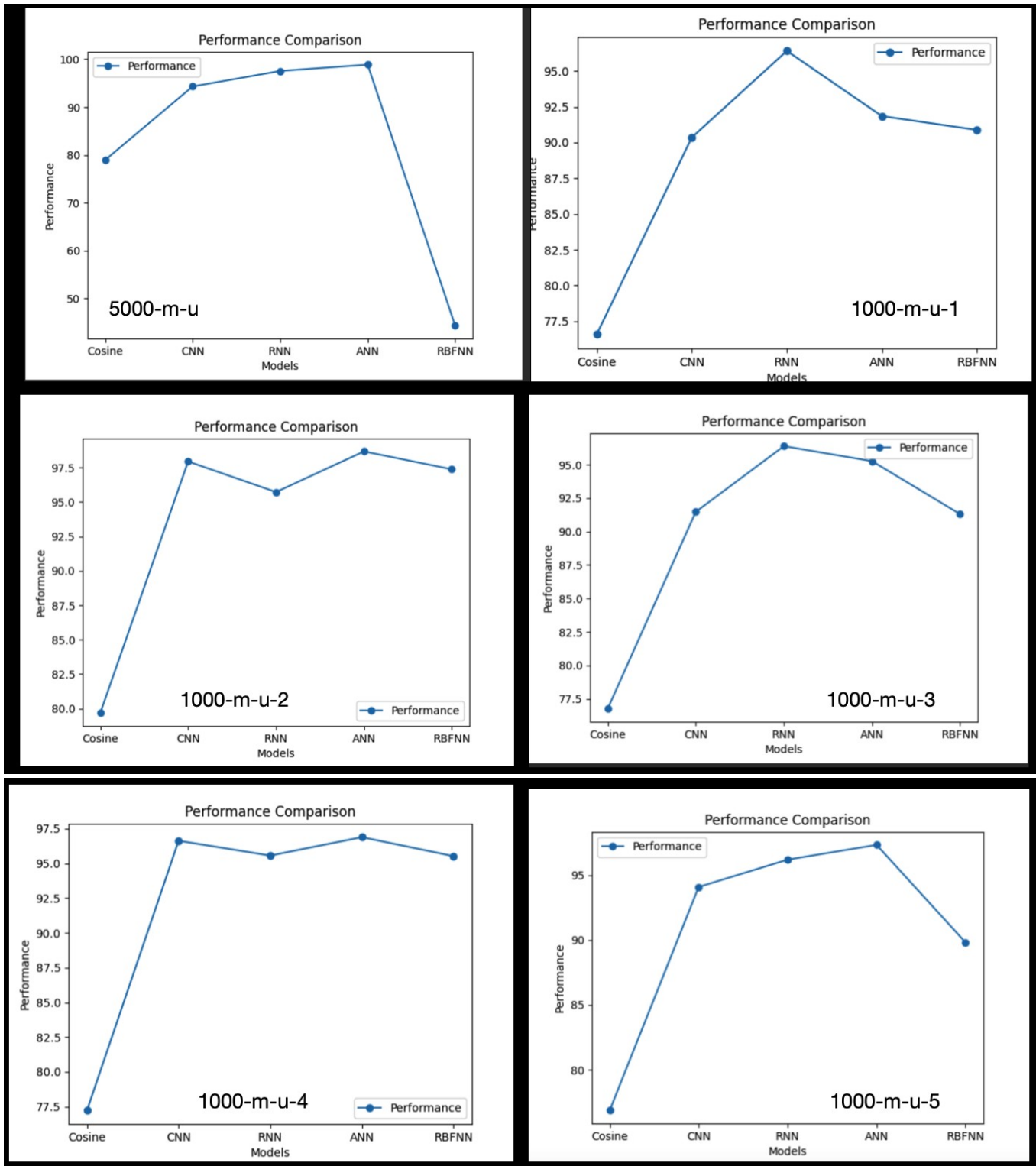
- *The values written are of units ms/step it took as average to process 10 epochs.*

<i>Run</i>	<i>CSM</i>	<i>CNN</i>	<i>RNN</i>	<i>ANN</i>	<i>RBFNN</i>
<i>5000-m-u</i>	NA	43.8	266.1	16.2	2
<i>1000-m-u-1</i>	NA	42.9	290.9	19.6	2
<i>1000-m-u-2</i>	NA	42.5	287	12.2	3.1
<i>1000-m-u-3</i>	NA	48	380.1	18.7	2.3
<i>1000-m-u-4</i>	NA	46.9	297.3	15.7	2
<i>1000-m-u-5</i>	NA	43.8	293.1	13.2	2.1

If the average is displayed for each,

<i>Time</i>	<i>CSM</i>	<i>CNN</i>	<i>RNN</i>	<i>ANN</i>	<i>RBFNN</i>
<i>Average</i>	NA	44.65	302.42	15.94	2.25

By combining both the factors, accuracy and time, ANN is the best model to use.



The first run gave us ANN as the best model, this was verified by the other five runs.

Problems Faced

- *Choosing the different AI models to complete the project*
- *Reducing our data set to use the relevant information*
- *Not being able to diversify our languages*
- *Not being able to use all the genres*
- *Difficult to explain the recommendations given to the user*

Conclusion

With a never-ending pool of movies, viewers, given so many options, waste their time searching for the movies they like, only if there were a recommendation system to save their time.

We developed a code to determine which algorithm would be the best to recommend movies. We faced basic issues of data cleaning and computer environment limitations but in the end, we were able to determine which algorithm would be the best fit. We found out that CSM and RBFNN were both time efficient but unable to recommend with high accuracy. RNN and CNN were the runners-up. Both of them were as good as ANN but RNN lacked in time efficiency and CNN wouldn't guarantee good accuracy. ANN was able to be an all-rounder and consistently gave us good accuracy with a time-efficient model.

The takeaways we got from this paper are that different AI models work in unique ways. Also, AI doesn't recommend the same thing to the same user again, it has a few differences.

References

data set used from kaggle - <https://www.kaggle.com/datasets/ashishjangra27/imdb-movies-dataset>