STAT4609 Big Data Analytics Final Project Report

Sua Kim (alixkim@connect.hku.hk)
Department of Statistics and Actuarial Science
The University of Hong Kong
Pok Fu Lam, Hong Kong

April 23, 2024

Overview

The project aims to develop a movie recommender system using the *MovieLens* 1M Dataset, which comprises approximately 1 million ratings of various movies provided by users of MovieLens. This dataset not only contains user ratings but also includes user demographic information and detailed movie metadata. The recommender system's primary objective is to analyse this extensive data to understand individual user preferences and predict movies that would appeal to them, enhancing their viewing experience.

The system will employ machine learning techniques to predict user ratings for movies that they have not yet watched. By doing so, it can suggest new movies to users based on their unique features and viewing history. This approach involves several steps: data preparation, exploratory data analysis, model selection, model training and validation, and implementation.

The end goal is to create an adaptive system that not only understands the preferences of its users but also evolves with their changing tastes, thereby sustaining user engagement and satisfaction. By leveraging the potential of advanced machine learning techniques and the comprehensive data provided by *MovieLens*, this project seeks to set a benchmark in personalised movie recommendation systems.

Data and Preprocessing

The dataset comprises three CSV files: Users, Movies, and Ratings. The Users table contains columns on user information, such as UserID, Gender, Age, Occupation and Zip-code. The Movies table contains columns on movie information, including MovieID, Title and Genres. There can be multiple genres for a single movie, and they are saved as a single string separated by vertical bars (|). For example, the genres for the movie *Toy Story (1995)* are saved as “Animation|Children’s|Comedy”. Finally, the Ratings table consists of the columns

UserID, MovieID, Rating, and Timestamp. For ease of computation, the three tables are joined by UserID and MovieID, which denote distinct users and movies.

For data preprocessing, we first reindex the following columns using the label encoder: UserID, MovieID, Gender and Occupation. Label encoding converts categorical variables, such as Gender and Occupation, into numerical values by assigning a unique number to each category, enabling algorithms to process and understand the data. Although the MovieID and UserID columns are already represented as numerical values, it is still beneficial to apply a label encoder on them for the contiguity and consistency in scale. Numerical ID values may not be contiguous, especially if some movies or users are removed from the dataset or if the IDs were assigned non-sequentially. Moreover, some algorithms, especially those based on distance computations like K-Nearest Neighbors or clustering algorithms, might misinterpret the numerical scale of IDs as a form of magnitude or importance (Pedregosa et al., 2011).

Then, we performed one-hot encoding on the Genres columns by splitting the genres by the vertical bars, then splitting them into separate columns. If the movie is included in a genre, the genre will have a value of 1, and if the movie is not included in a specific genre, the genre will have a value of 0. By using these dummy variables, we were able to convert the genres into separate binary features that are much easier for machine learning algorithms to process. After one-hot encoding, there are 18 columns of genres: Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, and Western.

Furthermore, we added the columns AverageRating and NumRatings, representing average ratings and the number of ratings, for each movie. This is to enhance the model's performance and the quality of the recommendations since they provide contextual and statistical insights that help the model better understand the general perception and popularity of movies among all users. In addition, it also allows the model to understand how the user features such as Age, Gender, and Occupation interact with the movie features AverageRating, NumRatings, and Genre.

For numeric variables, we normalised the values for uniformity in scale and numeric stability and to avoid biases. Finally, the dataset was split into 80% training and 20% testing datasets.

The dataframe was left with 29 features after the preprocessing. The user-related features were UserID, Gender, Age, Occupation and Zip-code. The movie-related features were Title, AverageRating, NumRatings, and 18 genre columns after the one-hot encoding process. Finally, the ratings-related features were Rating and Timestamp. The final processed data frame is shown in Figure 1.

	UserID	MovieID	Rating	Timestamp	Gender	Age	Occupation	Zip-code	Title	Action	...	Horror	Musical	Mystery	Romance	Sci-Fi	Thriller	War	Western	AverageRating	NumRatings
37222	252	3186	3	976688190	0	0.436364	11	97370	Thelma & Louise (1991)	1	...	0	0	0	0	0	0	0	0	3.680311	1417
145484	934	1774	5	976669675	1	0.618182	14	60538	Rocky (1976)	1	...	0	0	0	0	0	0	0	0	3.943313	1129
345747	2028	2820	4	974670594	0	0.436364	12	10011	Robin Hood (1973)	0	...	0	0	0	0	0	0	0	0	3.743440	343
358642	2097	346	4	974652753	1	0.618182	20	48067	Forrest Gump (1994)	0	...	0	0	0	1	0	0	1	0	4.087967	2194
562909	3462	1148	4	967174968	1	0.436364	15	80301	Terminator, The (1984)	1	...	0	0	0	0	1	1	0	0	4.152050	2098

Figure 1. Dataset table after joining and pre-processing

Exploratory Data Analysis

To gain a deeper understanding of the dataset and to choose an appropriate model for rating prediction, we conducted an exploratory data analysis (EDA). The EDA involved visualising various aspects of the data, including the distribution of ratings, user activity, movie popularity, and user demographics.

First, we examined the rating distribution across the 1-5 scale. The analysis revealed that the highest number of ratings were given at a score of 4, indicating a left-skewed distribution, as shown below in Figure 2. This suggests that users generally tend to rate movies positively, with a bias towards higher ratings.

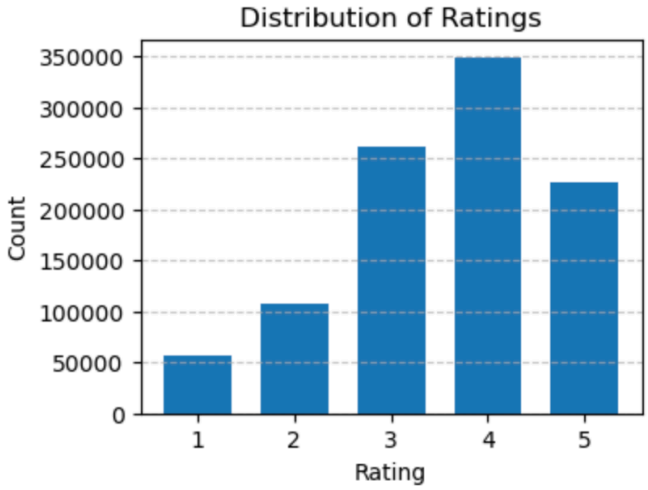

Figure 2. A bar chart that illustrates the distribution of ratings.

Next, we investigated the distribution of ratings per user and per movie. According to Figures 3 and 4, both distributions exhibited long-tailed and extremely right-skewed characteristics, which is a common trait in ratings datasets. This implies that a small number of users and movies account for a disproportionately large share of the total ratings. Such skewness can

pose challenges for recommender systems, as it may lead to biases towards popular movies or active users (Park, 2008).

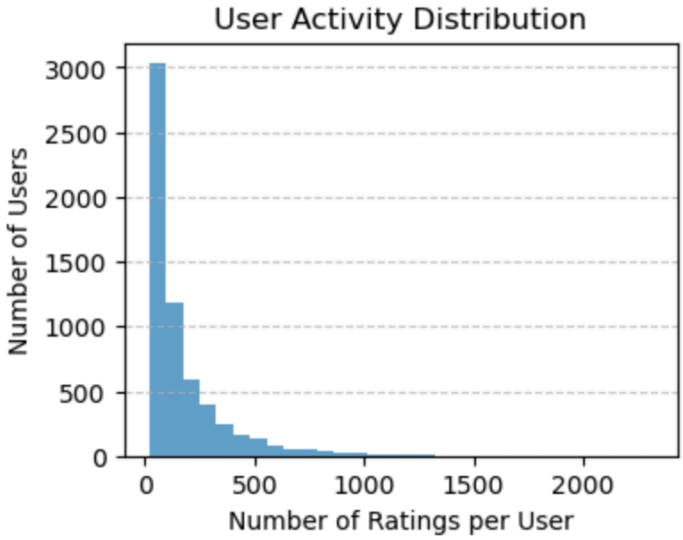

Figure 3. A bar plot chart that illustrates the number of ratings per user.

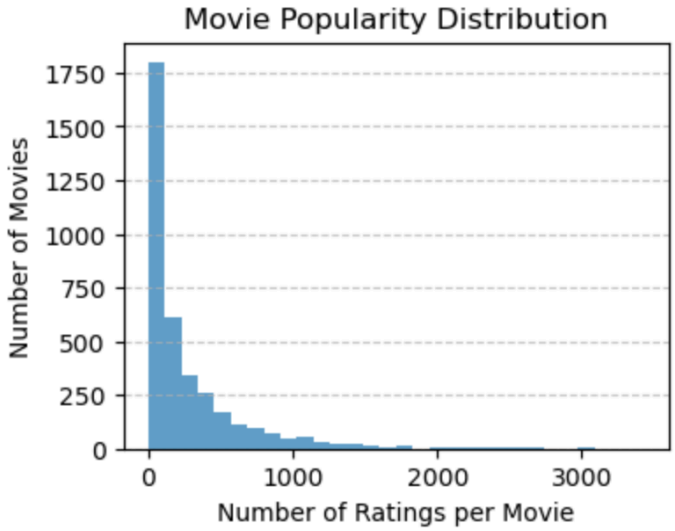

Figure 4. A bar plot chart that illustrates the number of ratings per movie.

Regarding user demographics, we analysed the gender and age distributions. The gender distribution in Figure 5 shows that there were more than twice as many male users as female users. This gender imbalance could potentially influence the overall rating patterns and preferences captured in the dataset. For the age distribution, Figure 6 shows that the 25-34 age range had the highest user count, indicating a right-skewed distribution. This suggests that the dataset predominantly represents the preferences of younger adults.

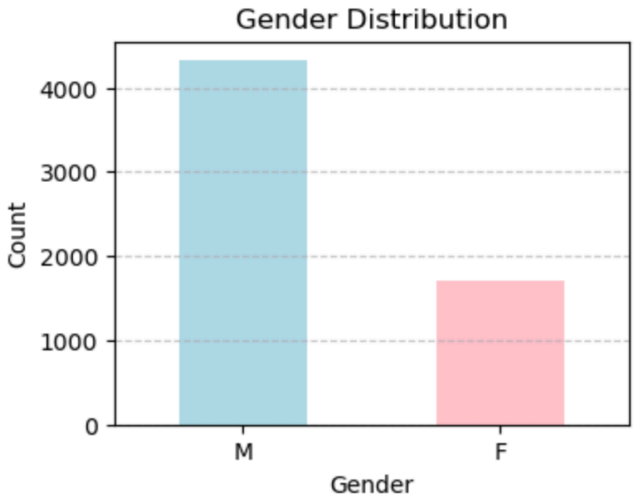

Figure 5. A bar plot chart of the number of users by gender.

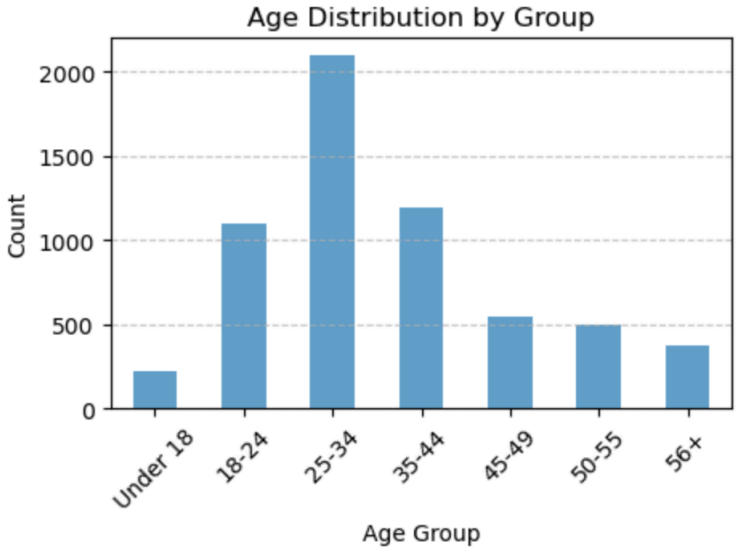

Figure 6. A bar plot chart of the number of users by age group.

Based on the insights gained from the EDA, we decided to build a predictive model using neural networks. Neural networks are well-suited for handling complex, non-linear relationships between variables, making them a promising choice for capturing the intricate patterns in the ratings data. The proposed neural network architecture consists of repeating layers of linear transformations, ReLU activations, and dropout regularisation.

Linear layers allow the model to learn weighted combinations of the input features, enabling it to capture the relationships between user and movie characteristics and the corresponding ratings.

Furthermore, ReLU activations introduce non-linearity into the model, allowing it to learn more complex and expressive representations of data. This is particularly important given the non-linear nature of user preferences and movie attributes.

Finally, Dropout regularisation helps prevent overfitting by randomly dropping out a portion of the neurons during training. This encourages the model to learn more robust and generalisable patterns, reducing its reliance on specific features or combinations of features.

By leveraging the power of neural networks and incorporating the insights from the EDA, we aim to build a recommender system that can effectively capture the nuances in user preferences and provide accurate movie recommendations. The proposed architecture, combining linear layers, ReLU activations, and dropout regularisation, is designed to handle the complexities and challenges identified in the dataset, such as the skewed rating distribution and the long-tailed user and movie activity distributions.

Model

Based on the insights gained from the exploratory data analysis and the characteristics of the dataset, we developed a neural network-based model class called MovieRecommender for predicting user ratings. The model architecture is designed to leverage the user and movie features, as well as the interactions between them, to make accurate rating predictions.

For user and movie embeddings, the model uses concatenated features to learn dense vector representations of users and movies. These features capture the latent factors and preferences associated with each user and movie, allowing the model to understand the underlying patterns and similarities. In addition to the user and movie embeddings, the model incorporates user and movie features, such as age, gender, occupation, and movie genres. These features provide additional context and information to enhance the prediction accuracy. The user and movie features are concatenated with their respective embeddings to form a comprehensive representation.

The concatenated features are passed through a series of fully connected layers. These layers learn the complex non-linear relationships between the user and movie features and their impact on the rating prediction (NVDIA, 2020). The model employs ReLU activation functions to introduce non-linearity and capture the intricate patterns in the data. To prevent overfitting and improve generalisation, the model incorporates dropout regularisation. Dropout randomly drops out a portion of the neurons during training, encouraging the model to learn more robust and generalisable patterns. This helps in reducing the model's reliance on specific features and enhances its ability to handle unseen data.

The model is trained using an MSE loss function and optimised using the Adam optimiser, which is an adaptive learning rate optimization algorithm, set up with a learning rate of 0.001. Adam combines the advantages of two other extensions of stochastic gradient descent: Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp)

(Kingma et al., 2014). Adam is particularly suited for problems with large amounts of data and parameters.

As shown in figure 7, the training process loops to perform training and evaluation over 10 epochs. Each epoch consists of a training phase where the model learns from the batched data from *train_loader*, and an evaluation phase using *test_loader* to assess the model's performance on unseen data. The loss values are printed each epoch, providing insight into the learning progress and the model's effectiveness at reducing prediction error. Finally, after training, we evaluated the final model performance using the test dataset and calculated the RMSE from the test loss, which gives a more interpretable measure of the average error.

```
Using cpu device
Epoch 1: Train Loss = 3.9154, Test Loss = 1.2427
Epoch 2: Train Loss = 1.5992, Test Loss = 0.9651
Epoch 3: Train Loss = 1.1386, Test Loss = 0.8785
Epoch 4: Train Loss = 0.9618, Test Loss = 0.8462
Epoch 5: Train Loss = 0.8916, Test Loss = 0.8400
Epoch 6: Train Loss = 0.8706, Test Loss = 0.8338
Epoch 7: Train Loss = 0.8613, Test Loss = 0.8406
Epoch 8: Train Loss = 0.8552, Test Loss = 0.8291
Epoch 9: Train Loss = 0.8521, Test Loss = 0.8234
Epoch 10: Train Loss = 0.8503, Test Loss = 0.8200
Test RMSE: 0.9055
```

Figure 7. The model training process and the final test MRSE.

Results and Application

The test set RMSE, as reported in Figure 7, is 0.9055. Having an MRSE of 0.9055 indicates that**,** on average, the model's predictions deviate from the actual ratings by about 0.9055 points on the 1 to 5 rating scale. Considering that the entire range of ratings spans 4 points (from 1 to 5), an RMSE of approximately 0.91 is about 22.6% of this range. This suggests that moderate prediction errors exist relative to the rating scale, and there can still be room for improvement.

However, it's noteworthy that the model achieves an RMSE of 0.9055, which is only slightly higher than the 0.86 RMSE achieved by the previous *Bellkor* solution of the Netflix Grand Prize. Considering the relative simplicity of this project's recommender model, achieving an RMSE close to such a benchmark shows the model's effectiveness despite the room for further improvement in accuracy.

Using the predicted ratings, we further built a function that recommends the top 10 movies to a user based on their user features. Figure 8 shows the top 10 recommended movies for a randomly selected user, ordered by descending predicted rating.

```
Top 10 Movie Recommendations:
MovieID: 2698, Predicted Rating: 4.45
MovieID: 708, Predicted Rating: 4.45
MovieID: 1066, Predicted Rating: 4.38
MovieID: 1839, Predicted Rating: 4.36
MovieID: 2309, Predicted Rating: 4.34
MovieID: 689, Predicted Rating: 4.30
MovieID: 1133, Predicted Rating: 4.29
MovieID: 1122, Predicted Rating: 4.29
MovieID: 843, Predicted Rating: 4.28
MovieID: 2816, Predicted Rating: 4.26
```

Figure 8. Top 10 movie recommendations for a randomly selected user.

Limitations and Conclusion

Although the model performs decently compared to the benchmarks of more complex models, its limitation is that it only considers the user and movie features and not the watch history of the user. A possible future improvement could include using models such as the RNN of LSTM after ordering each user's watch history by timestamp. However, with the current dataset, it wasn't easy to do so since the rating occurrence is long-tailed, which means that most users had left very few ratings. Instead of the rating record, a watch record that contains all the movies or shows watched could make it easier to build a recommendation model based on the user history.

In conclusion, the project developed a neural-network-based movie recommender system using the *MovieLens* 1M Dataset that includes different user and movie features. After 10 epochs of training, the model shows an MRSE of 0.9055, which is acceptable compared to benchmarks from similar projects, but there still exists room for improvement in the future, such as incorporating recurrent models by analysing the users' watch history.

References

1. Lecture notes from STAT4609 Big Data Analytics, The University of Hong Kong.
2. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
3. Park, Y., Tuzhilin, A. (2008). The Long Tail of recommender systems and how to leverage it. *Proceedings of the 2008 ACM Conference on Recommender Systems. ACM Conferences.* https://dl.acm.org/doi/10.1145/1454008.1454012.
4. NVIDIA Docs Hub. (2020). Linear/Fully-Connected Layers User's Guide. from https://docs.nvidia.com/deeplearning/performance/dl-performance-fully-connected/index.html.
5. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.