CHAPTER V
CONCLUSIONS AND RECOMMENDATIONS

5.1. Conclusions

Based on the research that was conducted, a movie recommendation system based on user rating similarity using the Matrix Factorization algorithm was successfully developed and implemented as a Flask-based web application. The system was designed to help users obtain movie recommendations based on a target rating and a preferred movie genre. Through this mechanism, users were not required to have a personal rating history within the application because the recommendations were generated using the rating patterns of historical users contained in the MovieLens dataset.

The Matrix Factorization model used in this study was able to learn latent representations of users and movies from rating data. The dataset consisted of 100,836 rating records, 610 unique users, and 9,724 unique movies, with a global average rating of 3.5016. The dataset had a sparsity level of 98.30%, indicating that most user–movie pairs did not contain ratings. Therefore, Matrix Factorization was appropriate because it could learn user–movie interaction patterns from an incomplete rating matrix.

The developed recommendation system utilized rating and genre inputs as the basis for generating recommendations. The target rating was used to identify historical users whose average ratings were close to the user's desired rating within the selected genre. Genre information was used to filter candidate movies so that the recommendation results remained within the category preferred by the user. After the reference users and candidate movies had been identified, the Matrix Factorization model was used to predict ratings for the candidate movies. The final recommendation score was calculated by considering the compatibility between the predicted rating and the target rating, as well as the normalized predicted-rating value.

The model-training results showed that the Training MSE decreased consistently throughout the training process, while the Validation MSE achieved its best value of 0.7243. Training was stopped at epoch 19 using an early-stopping mechanism because the validation performance no longer demonstrated stable improvement. These results indicate that the model successfully learned rating patterns from the training data while maintaining consideration for its ability to generalize to the validation data. The final evaluation on the test set produced an RMSE of 0.8657 and an MAE of 0.6617.

These values indicate that the average rating-prediction error was below one point on the MovieLens rating scale. Therefore, the Matrix Factorization model used in this study was sufficiently reliable to support a recommendation process based on predicted ratings.

From the application-implementation perspective, the system successfully displayed movie recommendations as Top-N lists, with available recommendation quantities of 5, 10, 15, or 20. The recommendation results were displayed as movie cards containing the movie title, genre, predicted rating, rating-match percentage, recommendation score, poster or placeholder image, and a visual score bar. This presentation made the recommendation results more informative and easier for users to interpret.

The application also successfully provided an advanced exploration feature through recommendation-card clicks. When a user clicked one of the recommended movie cards, the selected movie was used as an additional reference in the subsequent recommendation process. In this mode, the system retained the user’s selected rating and genre while also considering the embedding similarity between the candidate movies and the selected reference movie. This feature allowed users to explore recommendations more interactively without changing the system’s primary inputs.

Movie-poster integration was implemented as a supporting interface feature. Posters were retrieved optionally from an external service, while placeholders were provided when posters were unavailable. The application also used a caching mechanism through the poster_cache.json file to reduce repeated poster requests. Therefore, the poster feature improved the visual presentation of the application without interfering with its primary recommendation functionality.

Overall, this study successfully developed a Matrix Factorization-based movie recommendation system that utilized user rating similarity and genre filtering. The system was not only able to generate recommendations based on predicted ratings but also considered movie-category relevance through genre filtering. In addition, the advanced exploration feature improved the interactivity of the application by allowing users to generate subsequent recommendations based on a selected reference movie.

5.2. Recommendations

Based on the research results and the identified system limitations, several recommendations are proposed for future development.

1. The system evaluation can be expanded by incorporating Top-N ranking metrics such as Precision@K, Recall@K, Hit Rate, or NDCG. These metrics are important because the developed system produces an ordered recommendation list, whereas RMSE and MAE primarily measure rating-prediction errors.
2. The system can be developed by incorporating additional content attributes beyond genre, such as release year, tags, synopsis, actors, directors, or movie keywords. These additional attributes may enable the system to generate richer recommendations that more closely reflect the characteristics of individual movies.
3. The recommendation-score weighting mechanism can be investigated further. In this study, the final recommendation score was calculated using a combination of rating compatibility, predicted rating, and movie similarity in exploration mode. Future studies may evaluate different weighting combinations to identify the configuration that produces the most relevant recommendations.
4. Movie-poster integration can be improved by using cross-database movie identifiers, such as IMDb IDs or TMDb IDs. An identifier-based approach may produce more accurate poster retrieval than title-based searches.
5. The application can be extended by adding user-account functionality so that the system can directly store individual user preferences. With access to user histories, recommendations can become more personalized because the system would not rely solely on temporary rating and genre inputs but could also consider previous user interactions.
6. System testing can be expanded by conducting user evaluations to measure usability, user satisfaction, and perceived recommendation relevance. User testing is required so that the quality of the system can be assessed not only in terms of model performance and application functionality but also from the perspective of user experience.

7. The system can be optimized for larger datasets by applying techniques such as precomputation, vector-index storage, or more efficient candidate-retrieval methods. These improvements would be necessary if the system were implemented using substantially larger numbers of users and movies.

By applying these recommendations, the developed movie recommendation system can be further improved in terms of recommendation quality, evaluation completeness, computational efficiency, personalization, and overall user experience.