**Movie Recommendations**

**Participants: Musabai Ali**,

**Dzholdybayev Zhan**,

**Aitzhanuly Abylaikhan**

1. Introduction

1.1 Problem

Recommendation systems play a crucial role in helping users discover content that aligns with their preferences. In this report, we explore the development of a movie recommendation system using deep learning techniques.



1.2 Literature Review

Collaborative Filtering:

Collaborative filtering is based on the assumption that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past. The system generates recommendations using only information about rating profiles for different users or items. By locating peer users/items with a rating history similar to the current user or item, they generate recommendations using this neighborhood. This approach builds a model from a user’s past behaviors (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in. Collaborative filtering methods are classified as memory-based and model-based.

Collaborative filtering is a widely used technique in recommendation systems. Research papers such as "[Matrix Factorization Techniques for Recommender Systems](https://datajobs.com/data-science-repo/Recommender-Systems-%5bNetflix%5d.pdf)" by Koren et al. and "[Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model](https://people.engr.tamu.edu/huangrh/Spring16/papers_course/matrix_factorization.pdf)" by Koren et al. provide insights into matrix factorization methods.

1.3 Current Work

In this work, we build a deep learning-based recommendation system using TensorFlow/Keras. The system predicts movie ratings for users based on their historical preferences and suggests top-rated movies they have not yet seen.

2. Data and Methods

Data Description:

The dataset used is the MovieLens 100k dataset, containing user ratings for movies.

It includes information such as user IDs, movie IDs, ratings, timestamps, movie titles, and genres.

Exploratory Data Analysis:

Visualizations showing:

* Distribution of movie ratings.
* Most rated movies.
* Genres distribution.
* User activity over time.

Machine Learning/DL Models:

Embedding-Based Collaborative Filtering:

* Utilizes embeddings to capture latent factors for users and movies.
* Combines user and movie embeddings to predict ratings.

Deep Learning Model:

* Two embedding layers for users and movies.
* Dense layers with dropout for learning complex patterns.
* Output layer with softmax activation for rating prediction.

3. Results:

Model Training and Evaluation:

* The model is trained using Sparse Categorical Cross entropy loss.
* Training and validation accuracy and loss curves.

Movie Recommendations:

Top-rated movie recommendations for selected users.

* Example: User ID 894:
  + Movie seen by the User:
    - [List of seen movies]

Top 10 Movie Recommendations:

* [Recommended Movies]

4. Discussion:

Critical Review of Results:

* The model performs well in predicting ratings and suggesting movies.
* Effectiveness of deep learning in capturing user preferences.
* Potential biases or limitations in the dataset affecting recommendations.

Next Steps:

* Incorporate user feedback for better personalization.
* Explore advanced deep learning architectures.
* Implement A/B testing for recommendation strategies.
* Enhance model interpretability for user trust.

5. Conclusion

In conclusion, we have developed a movie recommendation system using deep learning techniques. The system effectively predicts movie ratings and provides personalized recommendations to users based on their historical preferences. Through further enhancements and optimizations, this system can offer valuable content discovery experiences for users.

6. References

Link to Youtube Video: [link](https://www.youtube.com/watch?v=iSn6vXira10)

MovieLens dataset: [link](https://grouplens.org/datasets/movielens/100k/)