### Movie Recommendation System: An IEEE Report

**Abstract**

This project presents a movie recommendation system utilizing two distinct implementations: one using Deep Belief Networks (DBN) with PyTorch, and the other leveraging Restricted Boltzmann Machines (RBM) with TensorFlow. The aim is to predict user ratings for movies and provide personalized recommendations based on the user's rating history. The report details the project inception, data preprocessing, model training, evaluation, and key findings. The models are evaluated on the MovieLens dataset, demonstrating effective prediction and recommendation capabilities.

**Introduction**

With the exponential growth of online content, recommendation systems have become essential tools for helping users discover relevant items such as movies, books, and products. This project focuses on developing a movie recommendation system using two different machine learning approaches: Deep Belief Networks (DBN) and Restricted Boltzmann Machines (RBM). The MovieLens dataset, which contains user ratings for a wide range of movies, is used to train and evaluate the models.

**Data Preprocessing**

The dataset used in this project is the MovieLens dataset, which includes user information, movie information, and user ratings. The preprocessing steps involved merging user and movie data, handling missing values, creating age group features, extracting the year from movie titles, and one-hot encoding categorical variables. These steps ensure that the data is in a suitable format for model training.

**Implementation 1: DBN with PyTorch**

The first implementation uses Deep Belief Networks (DBN) with PyTorch. The steps involved are:

1. **Loading the Data:** Importing and preparing the MovieLens dataset.
2. **Data Preprocessing:** Handling missing values, feature extraction, and encoding.
3. **Feature Engineering:** Creating new features such as age groups and one-hot encoding.
4. **Model Development:** Building and training the DBN model.
5. **Model Optimization:** Fine-tuning hyperparameters and evaluating model performance.

The DBN model is trained using the MovieLens dataset, and its performance is evaluated using metrics such as Mean Squared Error (MSE). The results show that the DBN model effectively captures user preferences and provides accurate movie ratings predictions.

**Implementation 2: RBM with TensorFlow**

The second implementation leverages Restricted Boltzmann Machines (RBM) with TensorFlow. The key steps are:

1. **Initializing Parameters:** Setting up the initial weights and biases.
2. **Gibbs Sampling:** Performing Gibbs sampling for training the RBM.
3. **Training the RBM:** Optimizing the model parameters.
4. **Predicting with the RBM:** Generating movie rating predictions.

The RBM model is trained and evaluated on the same MovieLens dataset. The results indicate that the RBM model also performs well in predicting movie ratings and making recommendations based on user history.

**Results and Findings**

The models were evaluated based on their ability to predict movie ratings and provide personalized recommendations. Key findings include:

1. Both DBN and RBM models showed consistent improvement in predicting movie ratings over epochs.
2. Users tend to rate movies in specific genres consistently higher, indicating strong genre preferences.
3. Popular genres like Comedy, Drama, and Action receive a high number of ratings, while niche genres have more dedicated viewers.
4. Different age groups exhibit distinct preferences, with teens preferring Animation and Sci-Fi, while seniors favor Classics and Westerns.
5. High-rated movies often feature well-known actors and directors, highlighting the influence of star power.

**Conclusion**

The project successfully implemented and evaluated two different machine learning models for movie recommendation. Both models demonstrated their effectiveness in capturing user preferences and providing accurate recommendations. Future work could involve exploring additional features, refining model architectures, and incorporating more advanced techniques to further improve recommendation accuracy.

**References**

1. G. Hinton, S. Osindero, and Y. Teh, "A Fast Learning Algorithm for Deep Belief Nets," Neural Computation, vol. 18, no. 7, pp. 1527-1554, 2006.
2. R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," in Advances in Neural Information Processing Systems 20 (NIPS 2007), 2008.
3. J. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer, vol. 42, no. 8, pp. 30-37, 2009.
4. X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, Article ID 421425, 2009.
5. P. Resnick and H. R. Varian, "Recommender Systems," Communications of the ACM, vol. 40, no. 3, pp. 56-58, 1997.