Report on Creating a Movie Recommendation System

Introduction

The development of a movie recommendation system involves several key stages, from initial data exploration to the building and fine-tuning of predictive models.

Initial Data Exploration

The first notebook focuses on initial data exploration, which is a critical step in understanding the dataset's characteristics and preparing it for effective model training.

Key Activities:

> Data Loading and Cleaning; Exploratory Data Analysis (EDA); Feature Engineering.

Model Building

The second notebook focuses on the construction and evaluation of the recommendation model. This approach addresses the challenge of developing a movie recommendation application by adapting it to a link prediction task, which is one of the three main types of tasks typically addressed in graph machine learning or graph neural networks, alongside node classification and graph classification. Initially, I constructed a bipartite graph consisting of nodes representing users and movies, with edges indicating the rating (ranging from 1 to 5) a user has assigned to a particular movie. Objective is to identify potential links between users and movies they haven't watched yet.

Key Activities:

> Model Selection: The choice of model is influenced by “Build Recommendation Systems with PyTorch Geometric and ArangoDB”[1], hence in this article the author uses a similar dataset.
> Training and Evaluation: The selected model is trained using the prepared dataset.
> Result Analysis: After training, the model's performance is evaluated using RMSE and results printed for the exact user for further personal analysis.
> Predictions and Recommendations: The final part of the notebook demonstrates how the trained model can be used to make movie recommendations. This includes generating personalized recommendations for specific users based on their historical preferences.

Second Approach:

My second approach to developing a movie recommendation system was about using embedding spaces. This method relies on deep learning techniques to create a nuanced model that understands user preferences and item (movie) characteristics.

Creation of Embedding Spaces:

- The core of this approach is to represent users and movies in an embedding space. This method involves creating dense, low-dimensional representations of high-dimensional data.
- Embeddings help in capturing the underlying features of users and movies, facilitating better recommendation accuracy.

Conclusion

The use of embedding spaces in this recommendation system allows for a more nuanced understanding of user preferences and movie attributes, leading to more accurate and satisfying recommendations. It shows RMSE 0.9781 instead of 1.0752 in GNN, but I personally find results from GNN more personalized and predictable.

References

[1]
https://medium.com/arangodb/integrate-arangodb-with-pytorch-geometric-to-build-recommendation-systems-dd69db688465

[2]
https://medium.com/coinmonks/how-to-implement-a-recommendation-system-with-deep-learning-and-pytorch-2d40476590f9