Movie Recommendation System

A movie recommendation system using Neural Collaborative Filtering (NCF) to predict movie ratings and recommend personalized movies to users based on their preferences and historical interactions. Dataset used is MovieLens 20M dataset. The approach combines traditional collaborative filtering with deep learning techniques to capture both user-item interactions and item features like genres and tags.

Approach

NCF is a recommendation algorithm that blends collaborative filtering and deep learning. Traditional collaborative filtering methods rely on user-item interactions (e.g., user ratings), whereas NCF introduces non-linear transformations of user and item features using neural networks, enabling it to capture complex patterns in the data.

Feature Engineering:

-User and Item Embeddings: Embeddings transform categorical features (user IDs, movie IDs) into continuous vectors. In this code, both users and movies are embedded into 64-dimensional vectors.

-Genre and Tag Features: Movie genres and tags are represented as binary vectors and TF-IDF weighted vectors, respectively. Dimensionality reduction using Truncated Singular Value Decomposition (SVD) is applied to reduce the feature space.

Data Preparation:

-Three datasets - ratings, movies, and tags - are loaded and merged based on movie and user IDs. Categorical user and movie IDs are mapped to continuous indices. Missing tags are filled with empty strings. MultiLabelBinarizer and TfidfVectorizer are utilized to transform genre and tag features into binary and TF-IDF weighted vectors, respectively.

Data Generator:

-A generator function is defined to generate batches of data for training and testing the NCF model to accommodate large size of the data.

Model Architecture:

1)Input Layers: Four input layers for user ID, movie ID, genre features, and tag features.

2)Embedding Layers: Embedding layers for user and movie IDs.

3)Concatenation: Concatenation of flattened embeddings and feature vectors.

4)Fully Connected Layers: Several dense layers with ReLU activation functions.

5)Output Layer: A dense layer with a linear activation function to predict ratings.

Reasons for Selecting This Approach :

-Flexibility: NCF can handle various types of data, including explicit feedback (ratings) and implicit feedback (user interactions).

-Scalability: Deep learning models like NCF can efficiently process large-scale datasets with high-dimensional features.

Evaluation Metric:

-RMSE serves as a reliable metric in recommender system competitions so have used it to be able to compare it with other algorithms. Model has 0.81 RMSE which is comparable to notable competitions like the Netflix Prize, Kaggle, and RecSys Challenges.

Mathematical Concepts Used

Matrix Factorization

Given a user-item interaction matrix A of dimensions m×n (where m is the number of users and n is the number of items), matrix factorization aims to find two smaller matrices U (user matrix) and V (item matrix) such that:

$A \approx U \times V^T$

Purpose The primary purpose of matrix factorization in recommender systems is to capture latent features or factors that influence users' preferences and item characteristics. By decomposing the original matrix into user and item matrices, we can generate personalized recommendations by identifying similar users and items based on their latent features.

Truncated Singular Value Decomposition (SVD)

Given a matrix A of dimensions m×n, Truncated SVD decomposes A into three matrices U, Σ , and V^T such that:

$A \approx U \times \Sigma \times V^T$

The "Truncated" aspect involves retaining only the top-k singular values and corresponding singular vectors, effectively reducing the dimensionality and noise in the data.

The primary purpose of Truncated SVD in recommender systems is to simplify the user-item interaction matrix by retaining only the most important latent factors or features. By reducing dimensionality and noise, Truncated SVD enables more efficient and effective modeling of user preferences and item characteristics, leading to improved recommendation quality and computational efficiency.

Trade-offs to consider

Accuracy vs. Coverage :

Accuracy : NCF, being a deep learning-based model, excels in capturing intricate user-item interactions and latent features, leading to high prediction accuracy and personalized recommendations

Coverage: Emphasizing coverage in NCF ensures a broader reach and promotes the exploration of a diverse range of items, enriching the user experience and fostering engagement

The focus on accuracy in NCF may result in a narrower item selection, predominantly recommending popular or similar items to users with comparable preferences, potentially limiting diversity and serendipitous discovery of less-known or niche items.

Timeliness:

NCF's ability to leverage deep learning architectures and techniques enables it to capture and adapt to evolving user preferences and market dynamics quickly, facilitating the generation of timely and context-aware recommendations based on real-time user interactions.

But,

achieving timeliness in NCF necessitates significant computational resources, memory, and expertise to train, fine-tune, and deploy deep learning models efficiently, especially as the dataset grows and the model complexity increases.

Maintainability:

Ensuring maintainability in NCF involves adopting modular, transparent, and well-documented design principles, practices, and workflows to facilitate collaboration, troubleshooting, and updates by developers, data scientists, and stakeholders over time.

However,

maintaining NCF models and complex recommender systems requires ongoing efforts, expertise, and resources to address issues such as model drift, data quality, algorithmic biases, and performance degradation due to evolving user preferences, market dynamics, and technological advancements.

Reference

https://towardsdatascience.com/neural-collaborative-filtering-96cef1009401