**Neural Collaborative Filtering Movie Recommendation System**

**Overview**

This project implements a **Neural Collaborative Filtering (NCF)** based movie recommendation system using TensorFlow and the MovieLens dataset. The system predicts movie ratings for users based on their past interactions and provides personalized movie recommendations. The project includes data preprocessing, model training, evaluation, and a user-friendly interface built with Gradio for interactive predictions.

**Dataset**

The system uses the **MovieLens dataset**, which consists of:

* **movies.csv**: Contains movie details such as movieId, title, and genres.
* **ratings.csv**: Contains user ratings with columns userId, movieId, rating, and timestamp.

The datasets are merged to create a single DataFrame with columns: movieId, title, genres, userId, and rating.

**Prerequisites**

To run the notebook, ensure you have the following Python packages installed:

pip install pandas scipy sklearn numpy tensorflow gradio matplotlib

**Project Structure**

The Jupyter Notebook (NCF\_Movie\_Recommendation\_System.ipynb) is organized as follows:

1. **Imports**: Libraries such as pandas, scipy, sklearn, numpy, tensorflow, and gradio are imported.
2. **Data Loading and Preprocessing**:
   * Load and merge movies.csv and ratings.csv.
   * Drop the timestamp column and handle missing values.
   * Create a mapping of movie titles to movie IDs.
3. **Model Training**:
   * Implements a Neural Collaborative Filtering model using TensorFlow.
   * Trains the model on a subset of the data to predict user ratings.
4. **Evaluation**:
   * Plots the distribution of prediction errors using a histogram to visualize model performance.
5. **Recommendation Function**:
   * A function recommend\_movie(user\_id, movie\_title) predicts the rating for a given user and movie.
   * Handles errors for invalid user IDs or movies not in the dataset.
6. **Gradio Interface**:
   * Provides an interactive interface for users to input a userId and movie\_title to get predicted ratings.

**How to Run**

1. **Clone the Repository**:
2. git clone <repository-url>
3. cd <repository-directory>
4. **Install Dependencies**: Run the following command to install required packages:
5. pip install -r requirements.txt

Alternatively, install the packages listed in the Prerequisites section.

1. **Download the Dataset**:
   * Download the MovieLens dataset (e.g., ml-latest-small) from GroupLens.
   * Place movies.csv and ratings.csv in the project directory.
2. **Run the Notebook**:
   * Open the NCF\_Movie\_Recommendation\_System.ipynb in Jupyter Notebook or Google Colab.
   * Execute the cells sequentially to load data, train the model, and launch the Gradio interface.
3. **Launch the Gradio Interface**:
   * The final cell launches a web-based interface where you can input a user ID and movie title to get a predicted rating.
   * If running on Colab, a public URL will be generated for accessing the interface.

**Usage**

* **Input**: Provide a numerical userId (e.g., 1, 2, 3, ...) and a movie\_title (e.g., "Toy Story (1995)") in the Gradio interface.
* **Output**: The system returns the predicted rating for the specified movie and user, or an error message if the input is invalid.

**Notes**

* The dataset contains some missing values in the title and genres columns (414 entries). These are not handled explicitly in the provided code but can be addressed by filtering or imputing missing data.
* The model assumes that the movie ID exists in the training data (train\_movie\_ids). If a movie is not found, an error message is returned.
* The Gradio interface may require a public URL when running on Colab. Ensure you have a stable internet connection.
* The prediction error distribution is visualized using a histogram, which helps evaluate the model's performance on a 20% sample of the data.

**Future Improvements**

* Handle missing values in the dataset more robustly (e.g., by filtering or imputing).
* Add support for recommending top-N movies for a user instead of single predictions.
* Incorporate additional features such as movie genres or user demographics into the model.
* Optimize the model architecture and hyperparameters for better accuracy.
* Deploy the Gradio interface as a standalone web application for easier access.