
📝 Description

This project aims to build a **Collaborative Filtering-based Recommendation System** that suggests items (e.g., movies, products) to users based on their past interactions and preferences, as well as those of similar users. Collaborative filtering relies on the assumption that users with similar behaviors and preferences will enjoy similar items.

We will use the **MovieLens 100K dataset**, a widely-used dataset in the recommender systems community, which contains 100,000 ratings from 943 users on 1,682 movies.

✅ Requirements

Functional Requirements:

- Load and preprocess the MovieLens dataset.
- Implement **User-Based Collaborative Filtering**.
- Implement **Item-Based Collaborative Filtering**.
- Evaluate the model using **RMSE** and **precision/recall at K**.
- Provide top-N movie recommendations for a user.
- Optional: Visualize user similarities or recommendation performance.

Technical Requirements:

- Programming Language: Python 3.x
- Libraries:
    - `pandas`, `numpy`
    - `scikit-learn`

    - `surprise` (for collaborative filtering algorithms)
    - `matplotlib` / `seaborn` (for optional visualizations)
- Dataset: MovieLens 100K: https://www.kaggle.com/datasets/prajitdatta/movielens-100k-dataset
-
- Execution environment: Jupyter Notebook or Python scripts

📦 Deliverables

- `README.md`: Project description, setup, and usage instructions.
- `data/`: Folder containing the MovieLens dataset.
- `notebooks/`:
    - `01_data_preprocessing.ipynb`: Load and prepare data.
    - `02_user_based_cf.ipynb`: User-based collaborative filtering implementation.
    - `03_item_based_cf.ipynb`: Item-based collaborative filtering implementation.
    - `04_evaluation_and_visualization.ipynb`: Model evaluation and plots.
- `src/`:
    - `data_loader.py`: Functions to load and preprocess data.
    - `recommender.py`: Core collaborative filtering logic.
    - `evaluation.py`: RMSE, precision, recall computations.
- `results/`: Folder for saving evaluation metrics and recommendation examples.
- `requirements.txt`: List of Python dependencies.

📁 File Structure

kotlin

CopyEdit

```
collaborative-filtering-recommender/
│
├── README.md
├── requirements.txt
│
├── data/
│   └── u.data
│   └── u.item
│   └── u.user
│
├── notebooks/
│   ├── 01_data_preprocessing.ipynb
│   ├── 02_user_based_cf.ipynb
│   ├── 03_item_based_cf.ipynb
│   └── 04_evaluation_and_visualization.ipynb
│
├── src/
│   ├── __init__.py
│   ├── data_loader.py
│   ├── recommender.py
│   └── evaluation.py
│
└── results/
    ├── top_n_recommendations.csv
    └── evaluation_metrics.json
```
