irushawn / **movielens_recommender__system**

Code | Pull requests | Actions | Projects | Security | Insights | Settings

0 stars | 3 forks | 0 watching | Branches | Activity | Tags

Public repository · Forked from CollinsNyatundo/movielens_recommender_system

1 Branch | 0 Tags | Go to file t | Go to file | Add file | Code

This branch is up to date with `CollinsNyatundo/movielens_recommender_system:main` . Contribute | Sync fork

CollinsNyatundo Add files via upload 367409d · 7 hours ago

data	Add files via upload	2 days ago
README.md	Update README.md	2 days ago
app.py	Add files via upload	2 days ago
index.ipynb	Add files via upload	2 days ago
presentation.pdf	Add files via upload	7 hours ago
project_proposal.md	Add files via upload	2 days ago
requirements.txt	Add files via upload	2 days ago

README

MovieLens Advanced Recommendation System

python 3.8+ | streamlit | license CC BY-NC 4.0 | GitHub Repo

Project Overview

In the era of endless streaming content, users are overwhelmed by choices. This project delivers an advanced, production-ready movie recommendation system using the MovieLens dataset. The system combines collaborative filtering, content-based filtering, and state-of-the-art ensemble machine learning models (XGBoost, LightGBM, CatBoost) to provide highly accurate, personalized movie recommendations. An interactive Streamlit web app enables users to explore recommendations, analytics, and model performance.

Key Features

- **Multiple Recommendation Approaches:**
  - Content-based filtering (genres, tags)
  - User-based and item-based collaborative filtering
  - SVD-based matrix factorization
  - Hybrid and ensemble models (XGBoost, LightGBM, CatBoost)
- **Streamlit Web App:**
  - Interactive UI for recommendations, analytics, and model performance
  - Visualizations: rating distributions, genre popularity, user activity, model comparisons
  - A/B testing simulator for business impact analysis
- **Comprehensive Evaluation:**
  - Metrics: RMSE, MAE, Precision@K, Recall@K, MAP, NDCG
  - Business-focused insights for deployment and user engagement

Data Description

- **Source:** [MovieLens ml-latest-small]
- **Size:** 100,836 ratings, 3,683 tags, 9,742 movies, 610 users
- **Files:**
  - `movies.csv` : Movie metadata (title, genres)
  - `ratings.csv` : User ratings (userId, movieId, rating, timestamp)
  - `tags.csv` : User-generated tags
  - `links.csv` : External links to IMDb and TMDb
- **Preprocessing:**
  - Merging, cleaning, and feature engineering (genres, tags, user/movie stats)
  - Removal of users/movies with <5 ratings to reduce sparsity
  - One-hot encoding, normalization, and timestamp conversion

Installation

1. **Clone the repository:**

```
git clone https://github.com/CollinsNyatundo/movielens_recommender_system.git
cd datascience
```

2. **(Optional) Create and activate a virtual environment:**

```
python -m venv ds-env
# On Windows:
ds-env\Scripts\activate
```

```
# On Unix/Mac:
source ds-env/bin/activate
```

3. **Install dependencies:**

```
pip install -r requirements.txt
```

Usage

Run the Streamlit App

```
streamlit run app.py
```

- Open the provided local URL in your browser to interact with the MovieLens Recommendation System.
- Explore recommendations, analytics, and model performance through the web interface.

Jupyter Notebook

- The full modeling pipeline, data exploration, and evaluation are documented in `index.ipynb` .
- Open the notebook in JupyterLab or VSCode to review and experiment with the code.

Methodology

1. **Data Preprocessing:**
   - Load and merge MovieLens data files
   - Clean missing values, engineer features (genres, tags, user/movie stats)
   - Normalize ratings, reduce sparsity
2. **Model Development:**
   - Content-based filtering (TF-IDF + Nearest Neighbors)
   - User-based and item-based collaborative filtering
   - SVD-based matrix factorization
   - Advanced ensemble models (XGBoost, LightGBM, CatBoost)
   - Hybrid and stacking approaches
3. **Evaluation:**
   - Metrics: RMSE, MAE, Precision@K, Recall@K, MAP, NDCG
   - Business impact: A/B testing simulator, engagement metrics

Results

- **Content-Based Filtering:**
  - Precision@5: 0.0

  - Recall@5: 0.0
- **User-Based Collaborative Filtering:**
  - Precision@5: 0.6000
  - Recall@5: 0.0154
  - RMSE: 0.9407
  - MAE: 0.7323
- **SVD-based Collaborative Filtering:**
  - Precision@5: 0.8000
  - Recall@5: 0.0205
  - RMSE: 1.9754
  - MAE: 1.5697
- **Hybrid Model (Collaborative + Content-Based):**
  - Precision@5: 0.8000
  - Recall@5: 0.0205
- **Ensemble (XGBoost, LightGBM, CatBoost):**
  - XGBoost RMSE: 0.7995
  - (See notebook for additional ensemble metrics)
- **Business Impact:**
  - Demonstrated improvements in simulated click-through, conversion, and engagement rates
- **Scalability:**
  - Efficient for large user-item matrices; handles cold-start scenarios

App Features

- **Recommendation Types:**
  - Ensemble (XGBoost + SVD), Pure SVD, Content-Based
- **User Profile Analytics:**
  - Movies rated, average rating, activity timeline
- **Analytics Dashboard:**
  - Rating distribution, genre popularity, user activity, genre correlation heatmap
- **Model Performance:**
  - Comparison table, feature importance, tree model plots
- **A/B Testing Simulator:**
  - Compare SVD vs. ensemble on simulated business metrics (CTR, conversion, engagement)
- **About Section:**
  - Technical stack, model details, performance summary

License & Citation

- **Data License:** See [MovieLens Terms of Use]
- **Citation:**

> F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872

Contributing & Contact

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Jupyter Notebook** 96.5%
- **Python** 3.5%

Suggested workflows

Based on your tech stack

Django

Build and Test a Django Project

Configure

Python Package using Anaconda

Create and test a Python package on multiple Python versions using Anaconda for package management.

Configure

Pylint

Lint a Python application with pylint.

Configure

More workflows

Dismiss suggestions