heathlikethecandybar / phase_4 Public

<> Code   Issues   Pull requests   Actions   Projects   Security   Insigh

main   1 branch   0 tags   Go to file   Code   ıt

...	51a55e1 yesterday	8 commits
.ipyn...	Finishing Touc...	yesterday
data	adding comme...	2 weeks ago
imag...	Finishing Touc...	yesterday
.DS_...	updates to repo	yesterday
REA...	updates to repo	yesterday
phas...	updates to repo	yesterday
phas...	updates to repo	yesterday
phas...	updates to repo	yesterday
phas...	add repo pdf	yesterday

No description, website, or topics provided.

Readme

0 stars

2 watching

0 forks

Report repository

Releases

No releases published

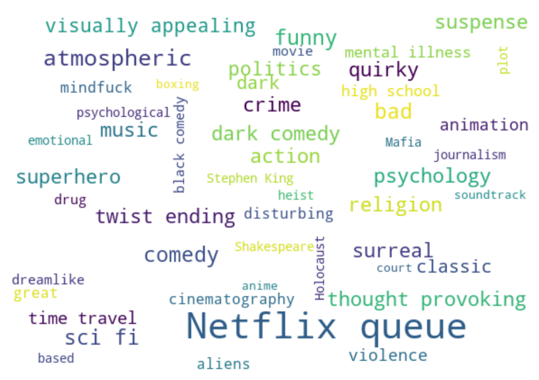
Packages

No packages published

Languages

● **Jupyter Notebook** 100.0%

README.md



Recommendation System

Author: Heath Rittler

Overview

You are working for an online streaming platform that offers a vast collection of movies to its subscribers. The platform wants to improve the movie recommendation system to enhance user engagement, increase user satisfaction, and ultimately drive more subscriptions. Currently, the platform provides basic recommendations based on popular movies, but they want to implement a more personalized recommendation system.

Business Problem

Develop a recommendation system that combines collaborative filtering techniques with content-based approaches. Utilize user-item rating data to identify similar users and recommend movies that have been positively rated by similar users. Additionally, leverage movie metadata such as genre, director, actors, and plot summary to provide content-based recommendations that match user preferences.

Data

This dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains 100836 ratings and 3683 tag applications across 9742 movies. These data were created by 610 users between March 29, 1996 and September 24, 2018. This dataset was generated on September 26, 2018.

Users were selected at random for inclusion. No demographic information is included. Each user is represented by an id, and no other information is provided.

The data are contained in the files

- `links.csv`

- `movies.csv`
- `ratings.csv`
- `tags.csv`

For our project we will be using primarily the ratings and movies data files:

ratings.csv

- `userId`
- `movieId`
- `rating`
- `timestamp`

Ratings are made on a 5-star scale, with half-star increments (0.0 stars - 5.0 stars). Timestamps represent seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970.

Ratings visual

movies.csv

- `movieId`
- `title`
- `genres`

genres visual

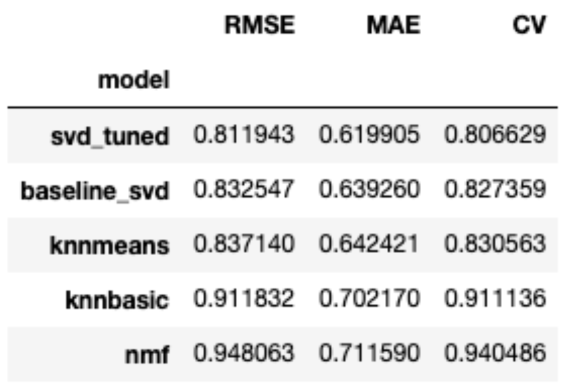
Additional information about this dataset can be found on the MovieLens website.

Methods

After our exploratory analysis, we looked at different collaborative based models to see if we could accurately predict/ recommend new movie titles within our data. We trained our model on 80% of the dataset, while saving the remaining 20% to test our assumptions in what our algorithms learned. We leveraged a SVD model as our baseline, and then iterated and tuned our SVD model to optimize performance.

That model was then iterated on, leveraging multiple models such as kNN, and NMF to evaluate the best model. Each approach was modeled with and without tuned hyperparameters. We chose our tuned SVD model, as it performed the best across our metrics of interest, RMSE, MAE, and Cross Validation.

For our final model, we were able to reduce our RMSE to 0.81. For example, our predictions are off roughly 0.83 on a scale of 0-5.

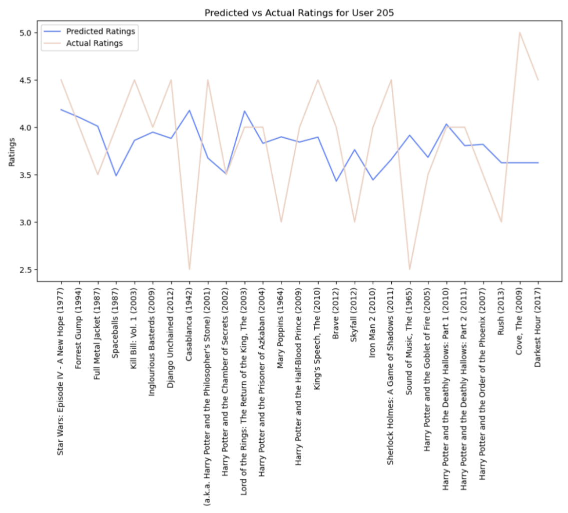


model	RMSE	MAE	CV
svd_tuned	0.811943	0.619905	0.806629
baseline_svd	0.832547	0.639260	0.827359
knnmeans	0.837140	0.642421	0.830563
knnbasic	0.911832	0.702170	0.911136
nmf	0.948063	0.711590	0.940486

Results

Our tuned SVD model has the lowest RMSE and MAE, indicating that it performs the best in terms of accuracy among the evaluated models. When evaluating our 5-fold cross validated results, our tuned SVD model again has the lowest CV, indicating consistent and good performance across different iterations. The RMSE measures the average difference between the predicted ratings and the actual ratings in the test set. With a rating scale of 0 to 5, an RMSE of 0.80 indicates that, on average, the predicted ratings are off by approximately 0.80 units from the actual ratings. This suggests that the recommendation system is making reasonably accurate predictions.

With that being said, and in addition to our scores, and comparisons, it is also important to get feedback from the users to indicate whether or not the predictions are providing value.



Conclusions & Recommendations

In conclusion, we were able to create a recommendation model that will accurately recommend movie titles to an end user. By leveraging this model, the online streaming company will have an engagement tool they can use, and measure to gain additional users, and build loyalty within their existing user base. Moving forward, integrating and testing the recommendation algorithm will be key and are summarized below.

- **A/B testing between users without the recommendations, and users with the recommendations to measure engagement.**
- **Continue to evaluate content recommendation model to improve hybrid approach (tags/ descriptions)**
- **Work with software dev/engineering to implement initial rating capture for new users**
- **Consider larger datasets and better compute resources**

For More Information

The full analysis is located in the Jupyter Notebook or review this summary presentation.

For additional info, contact Heath Rittler at hrittler@gmail.com

Repository Structure

```
├── data
├── images
├── README.md
├── phase_4_presentation.pdf
├── phase_4_notebook.pdf
└── phase_4_notebook.ipynb
```