INFO7390 Report: Anime Recommender System

Zhao Zhang 001533447

Liu Bo 001057068

Source

https://www.kaggle.com/hernan4444/anime-recommendation-database-2020?select=anime_with_synopsis.csv

Background

With the rise of eWorld , e-commerce and online advertisement are widely used. The recommender system becomes unavoidable in our daily life. Many famous websites, such as Amazon, Netflix and Youtube, all have their own unique recommender systems. Accurately recommending the items that users want can generate a huge amount of profit for those companies. Nowadays, there are different kinds of recommender systems, the most famous ones are collaborative filtering and content-based filtering. Collaborative filtering finds similar items based on the user's past behaviors, and recommends possible items that the user may like based on history. Content-based recommender is more NLP based and focuses on user's profile or item' description and recommender similar items. There are also other ways of recommending and many industries are using hybrid recommending which combines multiple ways of recommender systems.

Objective

This project aims to have experiments with different recommender systems. The datasets we used are from Kaggle, which are anime and rating information on MyAnimeList until the year of 2020. We managed to implement several styles of anime recommendations.

The focus of this project is to implement collaborative filtering algorithms from scratch to understand the underlying logic, then apply with surprise library with different modules and algorithms, and eventually evaluate the performances given by all the modules implemented by surprise library and also one nerual network recommender.

Except for the main task which focuses on collaborative filtering, we also had a taste of other algorithms. Those tasks are optional tasks for us, and they are implemented for the purpose of further exploring and personal interests.

Datasets

There are totally 5 datasets on Kaggle and in this project, we use 3 of them. The anime file contains all information about an anime except the synopsis. There are a total of 34 columns and 48.5k rows. This file is used to read anime's basic information when output such as anime's name. To the contrary, the anime_with_synopsis file contains only basic anime information and its synopsis, there are 5 columns and 48.5 rows. This file is only used in content-based filtering and no other algorithms.

The most important files are rating_complete.csv. This one contains all users' rating information, and that rating is only valid and included in this file when the user has watched an anime completely (not want to watch or in watching, but watched). There are 3 columns, user_id, anime_id, rating, and there are 57 million ratings in total.

Methodology

The main task is the mandatory part and we strictly follow the project requirements. We tried 7 different collaborative recommender systems. One of them is written a system from scratch with the Cosine Similarity algorithm, the other 6 algorithms are implemented by surprise library and they are KNN Basic, KNN with Z score, SVD, SVD++, NMF, and Coclustering. Each of them use a different statistical and mathematical method to estimate the scores and get the recommended anime. We also tune hyper parameters by grid cross validation and choose the best parameter for each algorithm and compare the results of 6 different algorithms by visualization.

Sub tasks are optional, mainly for personal interest or further exploring. They include other recommender algorithms. The recommender algorithm includes the most simple recommender: weighted rating, which ranks all anime by not just averaging the scores, but also weighted each anime based on the number of votes for that anime. Second algorithm in this part is content based filtering, which builds a tf-idf matrix based on anime's genres and synopsis and calculates cosine similarities between all anime. Recommendation is item based and will recommend similar anime based one anime' cosine similarities scores. We will try a neural network to fully connect users and anime and calculate scores then recommend.

Evaluation

In the recommender system, the error is the difference between predicted scores and real scores, therefore the metrics that suit the regression method can also be applied to the recommender system. Here we will choose two evaluation metrics for our models:

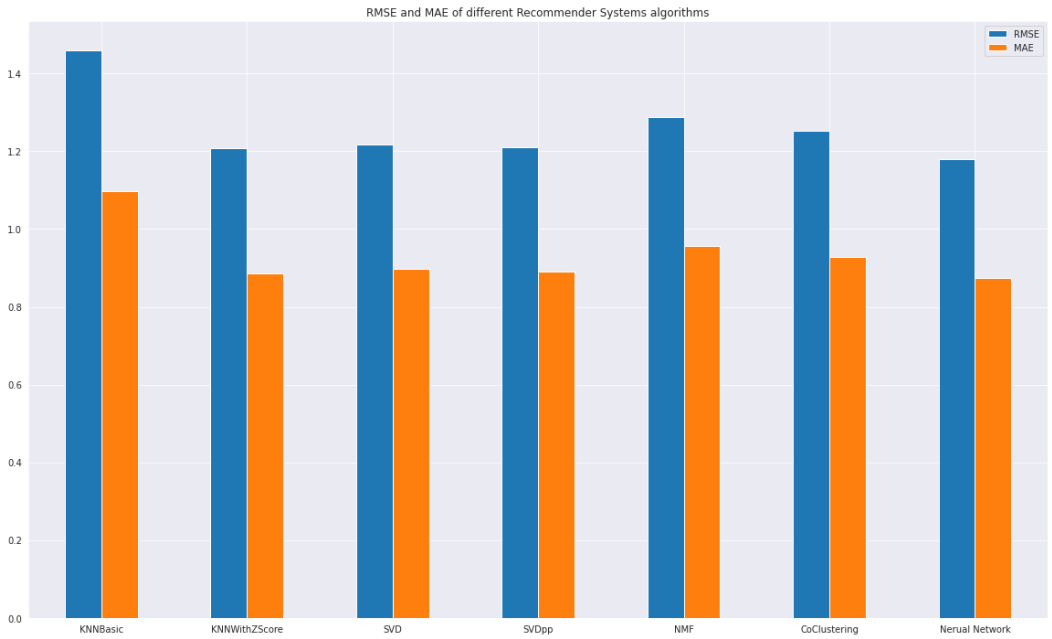
MAE(Mean Average Error) and RMSE(Root Mean Squared Error)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{r}_{ui} - r_{ui})^2}{n}}$$

$$MAE = \frac{\sum_{i=1}^{n} |\hat{r}_{ui} - r_{ui}|}{n}$$

MAE simply averages the error and RMSE uses the squared and root form of the average error.

Results and Finding



This visualization compares the resulting MAE, RMSE for 7 different models(6 of them are surprise collaborative filtering, the other one is neural network) with best parameters for each of them.

From the graph it can be found that the neural network has best performance, followed by KNNwithZScore, SVD, SVDpp. The results of these 4 models are actually very close.

Code and Running Instruction

Online Running On Kaggle: if run on Kaggle, uncomment the code which imports datasets in Kaggle(and you do not need to download data to local and store it) you can click run all then simply run all notebooks. The kaggle notebook should be created based on the datasets “Anime Recommendation Database 2020” in Kaggle(see source link)

Running locally: Make sure your datasets are in the same directory with your code(or change the path in pd.read_csv to the local path where you store your datasets) then run all, if should run successfully.

Library used in Code: numpy, pandas,sklearn, surprise, re, nltk, keras, matplotlib,seaborn