MoviesLens.org movie recommendation algorithm

Matthew Hale

Jan-2024

1. Introduction

Describe the dataset and summarizes the goal of the project and key steps that were performed

1a) Project Objective

MovieLens.org is a website that asks users to rate movies they've watched from 0.5 to 5 stars, and based on that information will provide recommendations of other movies that user might like to watch

To provide these recommendations a Machine Learning algorithm has been trained to predict the star rating a user will give to a specific movie, and return the movies with the highest predicted star ratings as recommendations.

The goal of this project is to train a new Machine Learning algorithm on a large dataset of movie ratings collected from a wide range of MovieLens.org users to predict the star rating each user might give a new film, and do so as accurately as possible.

1b) Initial Datasets

The MovieLens data has been sourced from the grouplens.org site here

This data contains over 10 million movie ratings of 10,000 movies by 72,000 users and was released in 2009. Per the ReadMe file at the link above, users were selected at random for inclusion amongst those who had rated at least 20 movies.

1c) Processed/Cleansed Datasets

Initial processing/cleansing has been done to join the datasets together into a single flat file, reformat columns where required, and randomly split the data into a training set called **edx** (90% of ratings on which to train the new ML algorithm) and a test set called **final_holdout_test** (10% of ratings which won't be involved in model development and will be used to test the performance of the final model produced).

Please see MovieLensProjectCode.R for the initial cleansing/processing script.

The rowcounts for the two sets are below, showing the ~90/10 split.

```
## Number of rows in training set edx = 9000055
```

```
## Number of rows in test set final_holdout_test = 999999
```

The 6 columns in the training and test datasets are per the below, listing the data type in each column and some sample entries:

Variable	Type	Examples
userId	int	50805 42011 45985 19660 46073 13268 6910 30265 65174 18227
movieId	int	1095 6303 593 4210 186 485 596 2402 913 6238
rating	num	4 3.5 4 4 4 2 2.5 2 3.5 3
timestamp	int	952366935 1121891132 854490846 1207988321 1123689113 923662409 1190501210 974970828 1159816589
title	chr	"Glengarry Glen Ross (1992)" "Andromeda Strain, The (1971)" "Silence of the Lambs, The (1991)"
genres	chr	"Drama" "Mystery|Sci-Fi" "Crime|Horror|Thriller" "Action|Crime|Drama|Horror|Thriller" …

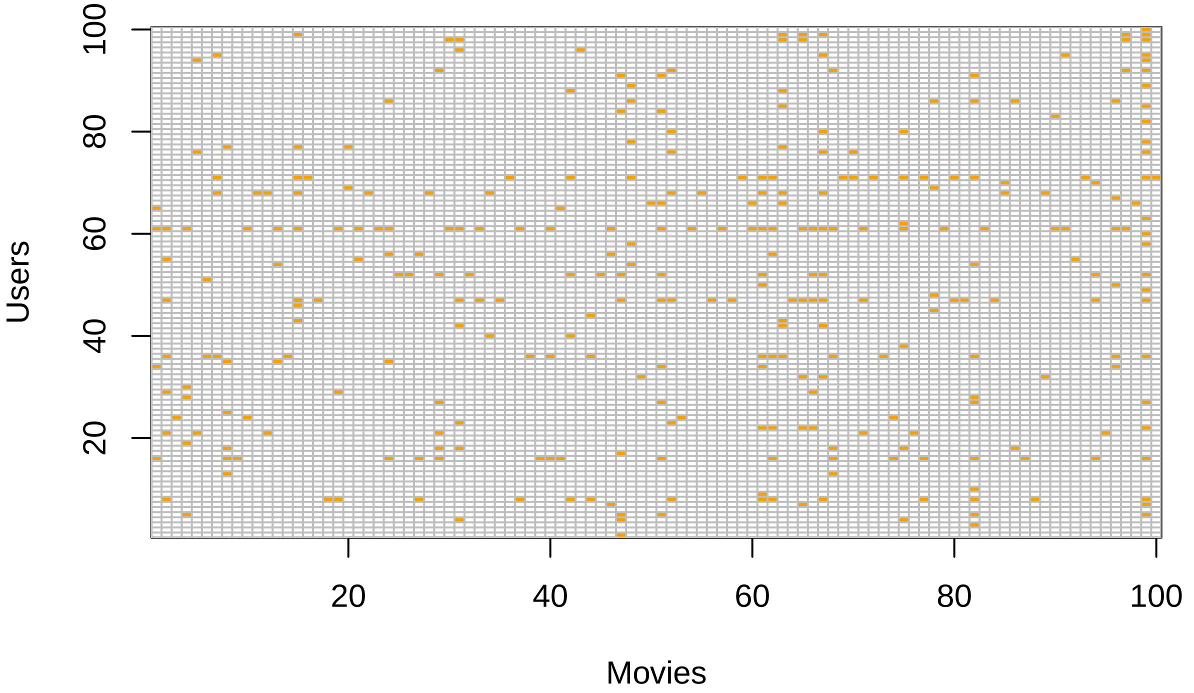
1d) Methodology

After initial exploratory data analysis, the algorithm will be developed using the edx set, with a number of Machine Learning methods evaluated using that dataset to try and produce an accurate model that is 'generalisable' to new data.

The final, selected model will then be applied to the final_holdout_test set, and evaluated primarily using root mean squared error (RMSE) - a measure of the deviation between the ML algorithm's predictions and the actual responses of users in the test set. For more see wikipedia here

2. Analysis

explains the process and techniques used, including data cleaning, data exploration and visualization, insights gained, and your modeling approach



```
## Selecting by n
```

userId	Forrest Gump (1994)	Jurassic Park (1993)	Pulp Fiction (1994)	Shawshank Redemption, The (1994)	Silence of the Lambs, The (1991)
13	NA	NA	4	NA	NA
16	NA	3	NA	NA	NA
17	NA	NA	NA	NA	5
18	NA	3	5	4.5	5
19	4	1	NA	4.0	NA

print(keep)

3. Results

presents the modeling results and discusses the model performance

4. Conclusion

a brief summary of the report, its limitations and future work

Tips - DELETE BEFORE SUBMISSION

- It is recommended to submit a GitHub or similar link in addition to or instead of uploading your files to the edX platform. Please make sure your repository is not set to private.

*Can I just use the same code/approach from course 8 for the MovieLens project?

You are certainly more than welcome to use the code from the previous course as a starting point. Note that you are expected to go beyond the models previously shown though; it is not enough to simply copy the code or follow the same approach and submit your report without building upon it. It's up to you to include the previous code or not, but most learners do and it helps provide context and a starting point for your report. Additions to the previously shown models can include extra biases (e.g., genre information) or techniques such as matrix factorization, or even completely different modeling techniques (i.e., not linear regression).

- What is the goal of your project? Why is this project important to do and why should people care about it?
- Have clear report sections/headings.
- Give enough background about the subject matter and dataset for people to understand. I recommend having a list or table of the dataset variables and their definitions.
- It's not enough to just show code and output without any text explanations. Explain any insights gained and your thought process throughout.
- Why did you choose to take X approach to handle Y task/issue? What are the pros and cons of it? What alternatives exist?
- Graders do not have the opportunity to discuss your project with you, and can only use what you provide in your report. I recommend providing more detail and explanations than risk providing too little and being penalized.
- Consider hiding some code chunks or messages, as some are not helpful and make the report unnecessarily long. An example is the messages from loading packages.
- Readers should not have to go back and forth between your PDF report and the Rmd/R files to understand what is happening and why.
- Be mindful of data leakage. Do not use information from the outcome or from the test set when processing your data and training your models. Check that the variables you want to use are actually available at the time of prediction for your use case. Some values are only known after the event happens and thus shouldn't be used.
- It can be nice to have a baseline to compare your models to, such as randomly guessing, the mean value from the training set, or a simple rule based system based on domain knowledge/EDA.
- It is recommended to start with simple models before moving onto more complex ones.
- What metric(s) did you choose and why? What does the metric mean in the context of your project (e.g. what does a RMSE of 5 mean, and can that be considered good or bad)? Does your model perform better/worse on certain subgroups? For classification projects, is the "cost" of a false positive vs false negative the same? What if you adjusted the decision threshold (not using the default of 0.5)?
- It can be nice to have a table of your models and their metrics at the end of your modeling section so the results are together in one place for easy comparison.
- What kind of impact or value can this project bring (e.g. increase revenue, decrease costs, save time, improve user/customer satisfaction, etc.)?
- What would you do differently or add onto your project if you had more or better data, more time, more computational power, etc.?