Contents

1 Overview

Welcome to our Tufts MATH260 final project! Here we plan to build a recommender system for board games, using a set of 13M reviews from BoardGameGeek. Our plan is to use some type of kernel method to perform the prediction.

The project idea comes from a kaggle challenge. You can find information about the challenge here on kaggle, it is also where you will find the dataset.

2 Project Setup

2.1 Dataset

The dataset for this project is rather large, so we don't want to include it in the project repository. We have the dataset setup to be ignored by .gitignore, so to ensure that the files are in the same place for everyone, we recommend running the following command to install the dataset (or run the equivalent).

Once installed, to reduce the size we strip out all the unnecessary fields (mostly by removing the comments field) run the command:

python src/math260/data$_{\text{prep.py}}$

from the root directory of the project. (TODO learn how to make that a codeblock)

```
unzip -d data boardgamegeek-reviews.zip
```

2.2 Python Environment

To ensure everyone is running the same version of python we include a pre-configured conda environment. To install this environment, run the following command from the root of the directory:

```
conda env create -n math260 -f environment.yml
conda activate math260
```

Then everything should be setup and ready to go.

2.3 Code

The main driver for the project code is located under src/math260/main.py Make sure to run it from the root directory of the repository.

2.4 Tests

Tests are stored in the tests/ directory of the project. All tests need to have a name of the form `tests_*.py` or they will not be ran. To see an example, check out `test_basic_recommender.py`.

To run all tests, do the following:

```
./run_tests.sh
```

3 Logistics

We are all working from different timezones on different schedules. This makes collaboration difficult. To ease the burden, we hope for a couple of things:

- We comment all of our code
- We document what we've done
- We write down our tasks
- We communicate with each other :)

Hopefully, none of this looks too difficult. Now, for documenting what we've done and for writing down our tasks, I placed two files called tasks.org and notes.org in the docs/ subdirectory.

3.1 tasks.org

This file should be used for documenting tasks that we have done and tasks that we need to do. The format should be pretty self-explanatory, don't worry if you mess it up. Try to document exactly what you have done or what needs to be done here. Take a look at the file for an example.

3.2 notes.org

This file should be used for documenting important components of our project that we all need to reference. For example, if a variable named by the letter A typically denotes an adjacency matrix, document that here. Again, don't worry about the structure. It's a plain text document, the information is what is important.