CSE 158/258, Fall 2020: Assignment 1

Instructions

In this assignment you will build **recommender systems** to make predictions related to user/recipe interactions from *Food.com*.

Solutions will be graded on Kaggle (see below), with the competition closing at **5pm, Monday November 15** (note that the time reported on the competition webpage is in UTC!).

You will also be graded on a brief report, to be submitted electronically on gradescope by the following day. Your grades will be determined by your performance on the predictive tasks as well as your written report about the approaches you took.

This assignment should be completed **individually**. To begin, download the files for this assignment from: `http://cseweb.ucsd.edu/classes/fa21/cse258-b/files/assignment1.tar.gz`

Files

trainInteractions.csv.gz 500,000 instances (recipe ratings) to be used for training. This data should be used for the 'cooking prediction' (both classes) and 'rating prediction' **(CSE258 only)** tasks. It is not necessary to use *all* observations for training, for example if doing so proves too computationally intensive.

user_id The ID of the user. This is a hashed user identifier from Food.com.

recipe_id The ID of the recipe. This is a hashed recipe identifier from Food.com.

date Date when the rating was entered.

rating The star rating.

train_Recipes.json.gz Training data for the cook-time prediction task **(CSE158 only)**, though the metadata could also be used for other tasks (since the recipe IDs match those in the interaction data). This file is json formatted, and contains the following fields:

name Name of the recipe.

minutes Cook time in minutes (i.e., the target variable).

contributor_id User that contributed the recipe.

submitted When was the recipe uploaded?

steps The recipe (steps are tab-separated).

description Short description of the recipe.

ingredients List of ingredients.

recipe_id ID of the recipe (same as in the interaction data).

test_Recipes.json.gz Test data associated with the cook-time prediction task. This data has the same format as above, with the 'minutes' (cook time) field removed.

stub_Made.txt Entries on which you are to predict whether a recipe would be made by a user (both classes).

stub_Rated.txt Entries (user_id and recipe_id) on which you are to predict user ratings **(CSE258 only)**.

stub_Minutes.txt Recipe IDs on which you are to predict cook time (these have the same order as the entries in test_Recipes, above).

baselines.py A simple baseline for each task, described below.

Please do not try to collect these reviews from the Web, or to reverse-engineer the hashing function I used to anonymize the data. Doing so will not be easier than successfully completing the assignment! **We will request working code for any solution suspected of violating the competition rules**.

Tasks

You are expected to complete the following tasks:

Cook prediction (both classes) Predict given a (user,recipe) pair from 'stub_Made.txt' whether the user would make a recipe (0 or 1). Accuracy will be measured in terms of the *categorization accuracy* (fraction of correct predictions). The test set has been constructed such that exactly 50% of the pairs correspond to cooked recipes and the other 50% do not.

Cook-time prediction (CSE158 only) Predict how long, in minutes, would be required to cook a recipe. Accuracy will be measured in terms of the *mean-squared error* (MSE).

Rating prediction (CSE258 only) Predict what rating a user would give to a recipe. Accuracy will be measured in terms of the *mean-squared error* (MSE).

A competition page has been set up on Kaggle to keep track of your results compared to those of other members of the class. The leaderboard will show your results on *half of* the test data, but your ultimate score will depend on your predictions across the *whole* dataset.

Grading and Evaluation

This assignment is worth 25% of your grade. You will be graded on the following aspects. Each of the two tasks is worth 10 marks (i.e., 10% of your grade), plus 5 marks for the written report.

- Your ability to obtain a solution which outperforms the leaderboard baselines on *the unseen portion of* the test data (6 marks for each task). Obtaining full marks requires a solution which is substantially better than baseline performance.
- Your ranking for each of the tasks compared to other students in the class (2 marks for each task).
- Obtain a solution which outperforms the baselines on *the seen portion of* the test data (i.e., the leaderboard). This is a consolation prize in case you overfit to the leaderboard. (2 mark for each task).

Finally, your written report should describe the approaches you took to each of the tasks. To obtain good performance, you should not need to invent new approaches (though you are more than welcome to!) but rather you will be graded based on your decision to apply reasonable approaches to each of the given tasks (5 marks total). The report is mostly a sanity check on the methods you applied, and is not usually a significant factor in grading (i.e., if you scored poorly on some task, we won't penalize you based on a poor selection of methods); just aim for a report detailed enough that somebody who had taken the class could re-implempent something like what you describe. 1-2 pages is fine.

Baselines

Simple baselines have been provided for each of the tasks. These are included in 'baselines.py' among the files above. **They are mostly intended to demonstrate how the data is processed and prepared for submission to Kaggle.** These baselines operate as follows:

Cook prediction Find the most popular recipes that account for 50% of interactions in the training data. Return '1' whenever such a recipe is seen at test time, '0' otherwise.

Cook-time prediction A simple linear regressor based on the length of the instructions.

Rating prediction Return the user mean if we've seen the user before, or the global mean otherwise.

Running 'baselines.py' produces files containing predicted outputs (these outputs can be uploaded to Kaggle). Your submission files should have the same format.

Kaggle

We have set up a Kaggle page to help you evaluate your solution. You should be able to access the competition via public links. The Kaggle pages for each of the tasks are:
`https://www.kaggle.com/c/cse158258-cooking-prediction/`
`https://www.kaggle.com/c/cse158-cook-time-prediction/`
`https://www.kaggle.com/c/cse258-recipe-rating-prediction/`

You are welcome to attempt the tasks from either class, but will only be graded on the tasks from your own class.