CSE 190, Fall 2015: Homework 3

Please submit your solution **at the beginning of the Monday week 7 lecture (Nov 9)** or outside of CSE 4102 beforehand. Please complete homework **individually**.

These homework exercises are intended to help you get started on potential solutions to Assignment 1. We'll work directly with the Assignment 1 dataset to complete them, which is available here: `http://jmcauley.ucsd.edu/data/assignment1.tar.gz`

You'll probably want to implement your solution by modifying the baseline code provided.

Note that you should be able to join the competitions using a UCSD e-mail. The competition pages can be found here:
`https://inclass.kaggle.com/c/cse-190-255-fa15-assignment-1-task-1-helpfulness-prediction/`
`https://inclass.kaggle.com/c/cse-190-fa15-assignment-1-task-2-purchase-prediction/`

Tasks (Helpfulness prediction)

First, since the data is quite large, when prototyping solutions it may be too time-consuming to work with all 1 million training examples. Also, since we don't have access to the test labels, we'll need to simulate validation/test sets of our own.

So, let's split the training data ('train.json.gz') as follows:
(1) Reviews 1-100,000 for training
(2) Reviews 900,001-1,000,000 for validation
(3) Upload to Kaggle for testing only when you have a good model on the validation set. This will save you time (since Kaggle can take several minutes to return results), and also will stop us from crashing their website...

1. Fitting the 'nHelpful' variable directly may not make sense, since its scale depends on the total number of votes received. Instead, let's try to fit $\frac{\text{nHelpful}}{\text{outOf}}$ (which ranges between 0 and 1). Start by fitting a simple model of the form
$$\frac{\text{nHelpful}}{\text{outOf}} \simeq \alpha.$$
What is the value of α (1 mark)?

2. What is the performance of this trivial predictor on the validation set? Recall that this should be measured in terms of the *mean absolute error* (`https://www.kaggle.com/wiki/AbsoluteError`) (1 mark).

3. To fit the same quantity, train a predictor of the form
$$\frac{\text{nHelpful}}{\text{outOf}} \simeq \alpha + \beta_1(\#\text{ words in review}) + \beta_2(\text{review's rating in stars}).$$
Report the fitted parameters and the MAE on the validation set (1 mark).

4. To run our model on the *test* set, we'll have to use the files 'pairs_Helpful.txt' to find the userID/itemID pairs about which we have to make predictions, and 'helpful.json.gz' to get the review data for those pairs. Using that data, run the above model and upload your solution to Kaggle. Tell us your Kaggle user name (1 mark). If you've already uploaded a better solution to Kaggle, that's fine too!

Tasks (Purchase prediction)

We need to think harder to generate a validation set for this data. Instead of just taking the last 100k reviews for validation, let's instead take the last 50,000, but also *randomly* select 50,000 'non-purchases' by randomly selecting a user and an item (and checking that they don't already show up together in the training set). We'll use this 50,000 purchased + 50,000 non-purchased dataset to validate our model.

5. The baseline provided for this task simply ranks products by popularity, and returns 'true' for popular products, by finding the most popular products that account for 50% of purchases. What is the performance of this baseline on the test set you built (1 mark)?

6. There is nothing particularly principled about this threshold of 50%—let's select a better one. Write down how you could use the popularity as a feature in a logistic regressor to set a threshold (e.g. what would the features and the parameters be?) (1 mark).

7. Can you find a better threshold (using logistic regression or otherwise)? Report the threshold you found and its performance (1 mark).

8. We could also set a *user* threshold in the same way. Report a user threshold that works better than a simple 50% threshold, and its performance on the test set. Run one of these methods on the test data and upload your solution to Kaggle (1 mark).