Homework 2

You should submit a single file containing your code ("homework2.py") to the autograder

Diagnostics (week 2):

We'll start by building a classifier that predicts whether a beer is highly alcoholic (ABV greater than 7 percent). First, shuffle the data and split it into 50%/25/%/25% train/validation/test fractions (this is already done by the runner/autograder).

1. We'll use the style of the beer to predict its ABV. First we construct a one-hot encoding of the beer style, for those categories that appear in more than 1,000 reviews (already done in the runner). Train a logistic regressor using this one-hot encoding to predict whether beers have an ABV greater than 7 percent (i.e., `d['beer/ABV'] > 7`). Train the classifier on the training set and report its performance in terms of the accuracy and Balanced Error Rate (BER) on the validation and test sets, using a regularization constant of C = 10. For all experiments use the class weight='balanced' option.
2. Extend your model to include two additional features: (1) a vector of five ratings (`review/aroma`, `review/overall`, etc.); and (2) the review length (in characters). Scale the 'length' feature to be between 0 and 1 by dividing by the maximum length seen during training. Using the same value of C from the previous question, report the validation and test BER of the new classifier.
3. Implement a complete regularization pipeline with the balanced classifier. Split your data from above in half so that you have 50%/25%/25% train/validation/test fractions (using code from the stub). Consider values of C in the range `{0.001, 0.01, 0.1, 1, 10}`. Report the model's validation and test performance for the value of C that works best on the validation set.
4. An *ablation study* measures the marginal benefit of various features by re-training the model with one feature "ablated" (i.e., deleted) at a time. Considering each of the three features in your classifier above (i.e., beer style, ratings, and length), and setting C = 1, report the test BER with only the other two features and the third deleted (2 marks).

Rating Prediction:

For these questions we'll use the Amazon Musical Instruments data. Code to read the dataset is included in the runner.

5. Implement a function to retrieve the most similar items to a given query item. Report both similarities and item ID.
6. Implement a rating prediction model based on the similarity function

$$r(u,i) = \bar{R}_i + \frac{\sum_{j \in I_u \setminus \{i\}} (R_{u,j} - \bar{R}_j) \cdot \mathrm{Sim}(i,j)}{\sum_{j \in I_u \setminus \{i\}} \mathrm{Sim}(i,j)}$$

(there is already a prediction function similar to this in the textbook code, you can either start from scratch or modify the solution there). Split the data into 90% train and 10% testing portions (data to do so is provided in the runner). When computing similarities return the item's average rating if no similar items exist (i.e., if the denominator is zero), or the global average rating if that item hasn't been seen before. All averages should be computed on the training set only.

7. Implement a predictor that works better than either the predictor from Q6 (or is better than a naive solution in the event that the solution from Q6 works poorly)