Edwin Huang

CSE 158 Assignment 1

Due 11/19/2019

Task 1:

First, I created two separate sets of data (training, validation) and added negative labels to them by randomly selecting books that the user has not read for a user - book pair.

The approach that gave me the highest score is similar to the approach that I did in homework 3, which is to use the Jaccard similarity score and combine it with the top 40 most popular books. Given a user and book ID pair, if the book's maximum similarity score is higher than 0.013 and is popular, then the predictor will return 1. I also changed from using the jaccard similarity to using the cosine similarity, but this method did not score higher than before so I decided to go back to Jaccard. Some other approaches that I took is also training an SVM SVC learning model, as well as a logistic regression model. However, these models scored less than the predictor in hw3. The features that I chose for the SVM and logistic regression model were: [maximum similarity score, rating, book count, read-or-not]. With this, I then used Sklearn's standard scalar model to normalize the training data so that they can be better compared using the prediction models. The parameters that I chose for the SVM were C = 1, and gamma = 0.1. For the logistic regressor, I picked C = 1. I trained the data on part of the dataset that I made with negative labels. The score for the logistic regressor on Kaggle scored around a 0.64, while the SVM scored around a 0.53.

Task 2:

The approach that I took for this is to use my solution from homework 3 and combined it with the homework 3 solutions. In one of the examples given in lecture, PorterStemmer() was used to stem and reduce the size of the dictionary/bag of words that we used to create the dataset. I felt that this method would remove some of the meaning of what the reviews meant, so I just stuck with removing the punctuations and lower-casing everything. In the beginning, I had a small train length for the bag of words with a length of 1000. I then created a list of possible lengths that we could use and tested on the valid set. I found out that the bigger the bag of words is, the higher the score would be. In the end, I picked a length of 15000 which gave me my highest score on Kaggle. I wanted to try using the whole length of the count of words in the whole dataset, but I felt that this would destroy my computer. I then split the dataset into training and validation datasets, and put it into a logistic regression with parameter C = 1, and fitted on the training data.