ZHAOKAI XU (KAGGLE NAME: ZhaokaiXu)

CSE 158 Assignment 1

Prof Julian McAuley

Report of 158 Assignment 1

**Task 1 Purchase Prediction**

Purchase prediction Predict given **a (user,item) pair** from `pairs Purchase.txt' **whether the user purchased the item** (really, whether it was one of the products they reviewed). 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 purchased items and the other 50% do not.

I applied the idea of **decision tree** by using several predictors. Since the noise of the data is relatively high, I applied several filters/predictors as the node of the decision tree to obtain a final accuracy of 0.70 (ranking 121overall).

After trying different combinations of predictors and different features for prediction to build the decision tree, finally I **built a two-layers decision tree** as my overall predictor:

**The first predictor is to filter data by popularity(frequency).** By running data on the training and validation data set, I found **a good threshold is 20**. If the popularity of the item is over 20, I treated this item as a popular item and the user has a very high chance to buy it, so I marked it True. This is based on the common sense that a user is very likely to buy an item if the item is popular.

I did also try to build another predictor using **the average rating of an item**. My previous assumption is that if an item has a very low rating, then the probability to buy it should be low. However, after running the predictor on training and validation set, I found that predictor based on rating does not work well because there are very few items have a very low average rating.

**The second predictor is to filter data by the category similarity of previous purchased history.** Since the category (a set of words to describe an item) of an item is unique, and by adding all words related to the items that a user purchased before to the set of words for a user, we can calculate the **Jaccard similarity** of an item and a user using the two set of words. I found the best threshold is by setting the similarity to **0.17.**

To further improve the accuracy by loosing the predictor a little bit, I also calculated the **number of common words** in the user set of works and the set of words in the item by setting **the threshold of common words to be 5.**

**Task 2 Category Prediction**

Category prediction Predict the category of a business from a review. Five categories are used for this task, which can be seen in the baseline program, namely Women's, Men's, Girls', Boys', and Baby clothing. Performance will be measured in terms of the fraction of correct classi\_cations.

I applied **sklearn Linear SVC** **to train five Linear SVC classifiers** and apply each of them to predict one categories, whether it predicts True OR False. If True predicted by one predictor, then we predict the item to be its category. **The linear SVC is running on the high dimension mapping of review Text.**

Before running Linear SVC, I mapped words into a high dimension space using some sklearn API from sklearn.feature\_extraction.text including **CountVectorizer** and **TfidfTransformer**.

With **CountVectorizer,** I calculated the counts of words from reviewText.

With **TfidfTransformer,** I mapped words into a high dimension space to feed into a linear SVC.

I obtained a final accuracy of 0.85 (ranking 77 overall).

Overall Summary:

From the above two task, I learned some techniques to train classifiers for recommendation system such as calculating the Jaccard similarity and apply decision tree to further improve accuracy, and I obtained some skills to do data mining such as vector2word and Tfidf, which are popular used in data pre-processing in NLP (neural network processing )