

INFS 770 – Advanced Data Mining Applications

Assignment 4: *Mining Text Data*

**Due 4/26/2021 (plus one-week grace period)**

Before you work on this assignment, please watch the lecture videos regarding text mining and make sure NLTK has been downloaded.

In this assignment, you will need to implement text mining models to analyze a sub-sample of an Amazon review data set (reference: http://snap.stanford.edu/data/web-Amazon.html). The CSV file (amazon\_review\_texts.csv) provided on D2L contains product reviews of four product categories: watch, automotive, electronic, and software. There are 1000 reviews in each category. The variables included are:

|  |  |
| --- | --- |
| pid | Product ID |
| helpful | The helpfulness rated by other user of this review |
| score | The product rating associated with this review |
| text | The text content of this review |
| category | The category of this product |

**Things to submit:** Please develop and submit an iPython notebook titled 770\_21\_a4\_*yourlastname*. In the notebook, you need to finish the tasks below, where (C) indicates that you need to write code for the task, (O) indicates that you need to show output, and (A) that you need to type your answers using Markdown text. Please use “run all” to run your code before you submit so that your iPython notebook will show the outputs of your code. You will lose 1 point if you do not “run all”. You probably want to copy and modify the code from the iPython notebook “topic modeling” posted on D2L.

In your iPython notebook, at the beginning of each cell, you need to indicate which task the cell is about. For example, in the cell related to task 1, you should first type “# Task 1: Read data”. If you do not clearly label the cells, you will lose 1-2 points (out of 18 points). Whenever you see “print” in the questions, you need write print statement to print the intended outputs.

T1. Read the dataset (amazon\_review\_texts.csv) as a Pandas data frame. Please use Pandas’s read\_csv method to read the dataset (see week 10 code to read sansers data). Print the first 5 lines of the data frame. Write code to show the distribution (value counts) of the variable “score” (i.e., review scores). Write code to show the distribution of “category”. (C)(O)

T2. Please follow the code in “newsgroup” (D2l->content->week9) and write code to tokenize the reviews. In the process, you need to 1) lowercase the tokens, 2) remove the stop words that are included in nltk.corpus.stopwords.words("english"), and 3) use nltk.stem.PorterStemmer() to do stemming. Please do this tokenization for each review in the “text” column (i.e., processed = list(map(preprocess, before\_token(df\_data1['text']))). Please write code to calculate word frequency distribution. White code to print the top 10 frequent words (hint: use fdist.tabulate(10)). (C)(O)

T3. Create a text box and tell me which of the top 10 words you obtained in T2 might not be useful in text clustering and classification, and why? (A)

T4. Write code to reconstruct the documents (i.e., processed\_doc = list(map(" ".join, processed))). Now processed\_doc is a list of strings, each of which represents a document (i.e., a review). Write code to vectorize all the documents using TfidfVectorizer (norm = ‘l2’) – you need to remove those tokens that appear in over 80% of the documents, and then do another round of stopword removal using the English stop words stored in scikit-learn. Please write code to print the numbers of features that were extracted by the TFIDF vectorizer. (C)(O)

T5. Write code to perform K-means to categorize documents into 4 clusters. Please print the top 10 representative words for each cluster (Please follow the sample code in week 9 “newsgroup” to write your code). (C)(O).

T6. Please create a text box to discuss how well you think these top 10 words describe the 4 product categories.(A)

T7. Build a topic model using Latent Dirichlet Allocation (LDA) based on the vectorized data from T4 (Please following the sample code in week 8 “topic modeling”) ( Please remember first to import genism and from gensim.models import LdaModel). Set the number of topics to 4. Print the topics (represented as linear functions of the words). (C)(O)

T8. Examine the representative words for each topic. Please create a text box and discuss how well these words describe the 4 product categories, and also tell me which unsupervised method (clustering vs LDA) you think is more effective in identifying the categories in this example and why? (A)

T9. Write code to perform a 5-fold cross validation using SGD classifier to predict the review scores using texts: Please first write code to use TFIDF to vectorize the dataset again – this time you need to remove those tokens that appear in over 80% of the documents, **remove those that appear just in 1 document**, and also remove the English stop words. Please write code to show the number of features you obtain**.** For each fold, your code needs to output the classification report (precision, recall, f1-score and support by class), as well as the average f1-score across the 5 folders. The code for running SGD using cross validation can be found in “sanders.ipynb” (d2l->Content->Week10) (C)(O)

T10. Write code to create a new variable named “satisfaction”. If a review score is 5 or 4, set the satisfaction value to 1 (the customer is satisfied). For the other review scores (3, 2, or 1), set the satisfaction value to 0 (the customer is not satisfied). **For T10, 11 and 15, the dependent variable is now “satisfaction”**. Please use the same TFIDF vectorizer as in T9 and perform a 5-fold cross validation to predict satisfaction using an SGD classifier. For each fold, your code needs to output the classification report (precision, recall, f1-score and support by class), as well as the average f1-score across the 5 folders. (C)(O)

T11. Use Bing Liu’s opinion lexicon described in the lecture as the vocabulary, and vectorize the texts again. Write code to use TFIDF to vectorize the dataset again - you need to remove those tokens that appear in over 80% of the documents, remove those that appear just in 1 document, remove the English stop words, and also remember to **change the parameter “vocabulary” in the TFIDF vectorizer**. Perform a 5-fold cross validation to predict satisfaction using an SGD classifier. For each fold, your code needs to output the classification report (precision, recall, f1-score and support by class), as well as the average f1-score across the 5 folders. (C)(O)

T12. Create a textbox and tell me if the average F1 score has increased in T11, compared with results obtained in T10? If so, why? (A)

T13. Use TFIDF to vectorize the dataset again using the same vectorizer you used in T9 and then perform PCA variable reduction (Here you need to do vectorization first to obtain a dataset that includes the TF-IDF numbers. Then you do standardization and then PCA (sample PCA code can be found in “sanders.ipynb”) to obtain a new dataset that includes a new set of variables called principle components. Each principle component now includes a vector of numbers). (C)

T14. Please create a textbox and use your own words to briefly describe PCA. (A)

T15. Please write code to determine how many principle components you obtain if you make sure that these components explain at least 90% of variance. Print the number of components**.** Perform a 5-fold cross validation to predict satisfaction using an SGD classifier. Now, the independent variables are the principle components that can explain over 90% of the total variance, and the dependent variable is “satisfaction” (Here you don’t need to do vectorization again this model fitting step since you are dealing with a 2-d numpy array of numbers). For each fold, your code needs to output the classification report (precision, recall, f1-score and support by class), as well as the average f1-score across the 5 folders. (C)(O)

T16. Create a textbox and tell me if the average F1 score has increased in T15, compared with results obtained in T10? If so, why? (A)

Please note for T15, you just need to do PCA twice. You first run PCA to determine the number of components and then run PCA again with the identified number of components to transform your data. **Please do not put PCA in the "for" loop for cross-validation.** You should use the transformed data when you do the cross-validation.