# DSBA/MBAD 6211: Homework Assignment 3 (50 pts)

**Instructions:** This is an individual assignment. The submitted solution and answers should be your own. The data file for this homework is **book\_ratings.csv**, which is to be downloaded from Canvas. You are asked to use Python to perform Text Mining tasks and answer the given questions. Create a new Word document and save it as TextMining\_xxxx (where xxxx is your ninernet login name). Write your full name on the first page of the Word document. Where required, write your answers or paste screenshots in this Word document. You need to submit both the Word document and Python Code file. **Your Python code should run correctly for your assignment to be graded. Code that generates error will result in loss of points (up to a maximum of 20%)**

# Variables and models naming requirements:

* Include your ***name initials*** to the data frame names as well as model names in your Python coding. This is required for your work to be graded.
* For instance, my initials are **CS**, and in my coding, I would name the data frames as ***dfCS, dfCS.train***, and ***dfCS.test.*** I would also name the models as ***SVDCS, topicsCS***, etc.

**Problem description and questions:** The dataset boo\_ratings.csv has reviews of books, along with other information, submitted on Amazon. The descriptions of the columns are given below.

* Id: Unique ID for the book
* Title: Book’s title
* Price: The price of Book
* User\_id: Unique ID of the user who rates the book
* profileName: Name of the user who rates the book
* review/helpfulness: Helpfulness rating (normalized to between 0 and 1)
* review/score: Rating from 0 to 5 for the book
* review/time: Time of given review
* review/summary: The summary of a text review
* review/text: The full text of a review

Tasks to complete and questions to answer (support your answers with data/analysis output):

[**Important Note:** You may find it useful to follow the code posted on Canvas for this assignment. However, the assignment context is different from the federalist papers example, and some steps have may need to be modified or dropped. Do not copy all the steps in the posted federalist papers example. In particular, don’t blindly copy the comments from the posted code. Choose which steps and code are needed for the assignment. If code not necessary for solving the questions is present in your solution, it will be penalized.]

* 1. (10 pts) Preprocess the text data in each of the "review/summary" and "review/text" columns. Preprocessing should include tokenization, lowercasing, stop word removal, stemming and any other necessary steps. Describe each of the above step in the Word document.

First you check for nulls and drop any in the dataset, as it could potentially skew the models down the line, the only column with nulls is review/helpfulness.

You define the dimensions for each of the columns:

A screenshot of a computer program

Description automatically generated

Remove the punctuation:

A screenshot of a computer program

Description automatically generated

Convert all the letters to lowercase and remove the stopwords:

A screenshot of a computer program

Description automatically generated`

Then you stem the words to get the main factor of the word:

A screenshot of a computer program

Description automatically generated

* 1. (7 pts) Create word clouds for each of the "review/summary" and "review/text" columns, after preprocessing. Copy and paste the word clouds outputs in the Word document. Write one-two sentences for what you can interpret from each of the two word clouds.

A close up of words

Description automatically generated

This word cloud seems to have a lot of positive indicators that go into multiple subjects, we see words like great/good or excel, resource or stori. I think this indicates that most of these books have a postive light thatt span multiple genres and the summary recommend the book typically in their summary.

A screenshot of a computer screen

Description automatically generated

Like the other one, we see a lot of positive indicators with words "like" and "love". We also see other words like "one", "make", "differ" which could indicate that there is only need for one of these books from the review, making a slightly negative disposition.

* 1. (8 pts) Perform text mining for each of the "review/summary" and "review/text" columns to estimate the similarity between documents. Show the similarity output tables for the first 5 documents for the summary and text columns. Paste the appropriate screenshots in the Word document.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

SUMMARY SIMUL:

A screenshot of a computer program

Description automatically generated

TEXT SIMUL:

A screenshot of a computer program

Description automatically generated

TEXT TOP 5:

A screenshot of a computer program

Description automatically generated

SUMMARY TOP 5:

A computer screen shot of a program code

Description automatically generated

* 1. (10 pts) Perform topic modeling on “review/text” using LDA model and generate 6 topics. In the word document, show the topic model output for each of the 6 topics as the linear combination of the terms. For 2 of the topics, write a short description in 2-3 sentences for each topic.

A screenshot of a computer

Description automatically generated

[(0, '0.007\*"book" + 0.005\*"present" + 0.004\*"make" + 0.004\*"cover" + 0.004\*"great" + 0.004\*"like" + 0.004\*"academ" + 0.004\*"modern" + 0.004\*"well" + 0.004\*"author"'), (1, '0.044\*"book" + 0.009\*"read" + 0.007\*"love" + 0.007\*"mani" + 0.006\*"one" + 0.005\*"make" + 0.005\*"found" + 0.005\*"would" + 0.005\*"time" + 0.005\*"much"'), (2, '0.024\*"book" + 0.007\*"would" + 0.007\*"read" + 0.007\*"one" + 0.006\*"mani" + 0.005\*"like" + 0.005\*"work" + 0.005\*"use" + 0.005\*"learn" + 0.004\*"get"'), (3, '0.015\*"book" + 0.012\*"read" + 0.010\*"like" + 0.010\*"stori" + 0.006\*"one" + 0.006\*"love" + 0.005\*"get" + 0.005\*"time" + 0.005\*"make" + 0.004\*"mani"'), (4, '0.022\*"book" + 0.014\*"read" + 0.007\*"good" + 0.007\*"one" + 0.007\*"get" + 0.006\*"like" + 0.006\*"make" + 0.006\*"want" + 0.005\*"way" + 0.005\*"book,"'), (5, '0.026\*"book" + 0.011\*"one" + 0.009\*"read" + 0.005\*"author" + 0.005\*"like" + 0.004\*"make" + 0.004\*"great" + 0.004\*"-" + 0.004\*"charact" + 0.004\*"also"')]

Topic 1: Based on the saliency of the terms and topic 1 being the largest of all the other ones, it will encompass more of the words than any thing, with the most salient words having the top 30 most of which have postive indicators. I believe this is the best representive because the most sparse amount of words that go in the topic, taking about 25.8% of all the tokens created.

Topic 6: I chose this one becuase its the opposite of topic one with the least amount of frequency of terms, only having 9.9% of tokens make up the topic. Looking at its relevance only, we can see that most words have a small amount in it, making it not as suggestive comapred to the others.

* 1. (15 pts) Build predictive models for review/score, as follows:
     1. Apply SVD to extract 5 components from the “review/text” column.

A computer screen shot of text

Description automatically generated

A screenshot of a computer

Description automatically generated

* + 1. Combine two non-text columns with the 5 extracted components and build a

decision tree model to predict the “review/score”. Call it Model-1.

A screenshot of a computer program

Description automatically generated

* + 1. Report the confusion matrix of Model 1.

A screenshot of a computer screen

Description automatically generated

* + 1. Repeat tasks 5a-c with SVD to extract 8 components and build a predictive model. Call it Model-2. Report the confusion matrix of Model-2.

A computer screen shot of text

Description automatically generated

A screenshot of a graph

Description automatically generated

* + 1. Which model performs better? Explain your answer

I would choose model 2 as most of time when ran with its predicted, it would spread to the other components rather than be consolidated into one. I look for the distribution among the labels and try to see which is more meaningful. If it would be for accuracy, then model 1 would be better but I think the second one gives more information. Both models are similar in nature though.