

MSCI 641 Assignments

All assignments will be done using the corpus of Amazon reviews available for download at:
<https://github.com/fuzhenxin/textstyletransferdata/tree/master/sentiment>

This dataset contains two classes of consumer product reviews: positive and negative.

You must use python 3 for your assignments. It is important that you do all assignments, as each subsequent assignment builds upon the work you have done in all previous assignments.

Create a private GitHub repository for all your assignments and share it only with the TA. Upload each of your assignments into a separate folder in your repository by the due date. If an assignment requires a report, place the report in the README file in the main folder for each Assignment. By the due date of Assignment 0, submit the URL of your Github account to “Assignments” dropbox on Learn. Additional instructions on how to upload code for automatic verification will be posted later.

All assignments are due by **11:59pm EST** on the due day. You **may not** edit/update your assignment files on Github after the due date.

You must use your own computational resources for these assignments.

Assignment 0 (no grade). Due date: May 23

Create a development environment for your subsequent assignments:

1. Create Python 3 virtual environment on your local machine
2. Install the following libraries: keras, NumPy, SciPy and gensim

Assignment 1 (code only, 3%) Due date: May 30

Write a python script to perform the following data preparation activities:

1. Tokenize the corpus
2. Remove the following special characters: !"#\$%&()*+/,;<=>@[\\]^_`{|}~\t\n
3. Create two versions of your dataset: (1) with stopwords and (2) without stopwords. Stopword lists are available online.
4. Randomly split your data into training (80%), validation (10%) and test (10%) sets.

Assignment 2 (code + short report, 7%). Due date: June 6

Write a python script using SciPy library to perform the following:

1. Train Multinomial Naïve Bayes (MNB) classifier to classify the documents in the Amazon corpus into positive and negative classes. Conduct experiments with the following conditions and report classification accuracy in the following table:

Stopwords removed	text features	Accuracy (test set)
yes	unigrams	
yes	bigrams	
yes	unigrams+bigrams	
no	unigrams	
no	bigrams	
no	unigrams+bigrams	

For this assignment, you must use your training/validation/test data splits from Assignment 1. Train your models on the **training** set. You may only tune your models on your **validation** set. Once the development is complete, run your classifier on your **test** set.

2. Answer the following two questions:
 - a. Which condition performed better: with or without stopwords? Write a brief paragraph (5-6 sentences) discussing why you think there is a difference in performance.
 - b. Which condition performed better: unigrams, bigrams or unigrams+bigrams? Briefly (in 5-6 sentences) discuss why you think there is a difference?

Assignment 3 (code + short report, 6%). Due date: June 13

1. Write a python script using genism library to train a **Word2Vec model** on the Amazon corpus.
2. Use genism library to get the most similar words to a given word. Find 20 most similar words to “good” and “bad”. Are the words most similar to “good” positive, and words most similar to “bad” negative? Why this **is** or **isn’t** the case? Explain your intuition briefly (in 5-6 sentences).

Assignment 4 (code + short report, 9%). Due date: June 27

Write a python script using keras to train a **fully-connected feed-forward neural network classifier** to **classify** documents in the Amazon corpus into **positive and negative classes**. Your network must consist of:

1. Input layer of the word2vec embeddings you prepared in Assignment 3.
2. One hidden layer. For the hidden layer, try the following activation functions: ReLU, sigmoid and tanh.
3. Final layer with softmax activation function.
4. Use cross-entropy as the loss function.
5. Add L2-norm regularization.
6. Add dropout. Try a few different dropout rates.

For this assignment, you must use your training/validation/test data splits from Assignment 1. Train your models on the **training** set. You may only tune your models on your **validation** set. Once the development is complete, run your classifier on your **test** set.

Report your classification **accuracy** results in a table with three different activation functions in the hidden layer (**ReLU, sigmoid and tanh**). What effect do activation functions have on your results? What effect does addition of **L2-norm regularization** have on the results? What effect does **dropout** have on the results? Explain your intuitions briefly (up to 10 sentences).