**MMAI 891**

**Natural Language Processing**

**Dr. Stephen W. Thomas**

**Individual Assignment 1**

**Click Here and Input Due Date**

**Click Here and Enter Your Name**

# Assignment Instructions

This assignment contains two mandatory questions and one optional bonus question. Each question has multiple tasks. There are two types of tasks: tasks that require you to write code; and tasks that require you to write text responses. A grading rubric is provided for each task of each question.

For tasks that require **code**:

* Use Python (preferred) or R to complete the task.
* Use the template file provided (Assignment1\_891.py for Python or Assignment1\_891.R for R] and rename it to contain the question number and your full name without spaces (e.g., Assignment1\_891\_Q1\_StephenThomas.py). Do not submit Jupyter Notebooks (.ipynb) or RMarkdown (.Rmd) files.
* Submit code that runs without errors.
* Submit code that is reproducible. E.g., set random number seeds as appropriate. You should be able to run you code again and again and again, from the top of the file to the bottom of the file, and get the exact same results each time. I should be able to run your code, on my machine, from scratch, again and again, and get the exact same results that you get.
* Submit code that is organized. Make your code readable. Provide comments to describe what the code is doing and why. Don’t leave “old” code laying around. Overall, if your code is clear and easy to read, then we will be happy. When we are happy, we give better marks.

For tasks that require **text responses**:

* Type your response in this Word document.
* Be clear about which task you are responding to.
* Use English. Use proper grammar, spelling, and punctuation. Be professional and clear. Be complete, but not overly-verbose.
* You may refer to your code. Please do so very clearly. E.g., “As can be seen in on line X of file Y…”

Your assignment submission should contain the following files:

* Assignment1\_891\_Q1\_FirstLast.py or Assignment1\_891\_Q1\_FirstLast.R
* Assignment1\_891\_Q2\_FirstLast.py or Assignment1\_891\_Q2\_FirstLast.R
* (Optional) Assignment1\_891\_Q3\_FirstLast.py or Assignment1\_891\_Q3\_FirstLast.R
* Assignment1\_891\_FirstLast.docx

# Sentiment Analysis via Lexicon-Based Approach

## Preamble

Download the “Product Sentiment” dataset: *sentiment\_train.csv* and *sentiment\_test.csv.*

Note: this dataset is originally from <http://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences>

## Tasks

1. [Code] Perform sentiment analysis on the dataset using the lexicon-based approach. If using R, using the qdap library. If using Python, use the nltk.sentiment.vader library.
   1. Load, clean, and preprocess the data as you find necessary.
   2. Use the training data to tune the hyperparameters of the function.
      1. In qdap, there are four hyperparameters: *question.weight*, *amplifier.weight*, *n.before*, and *n.after*.
      2. In vader, there are no hyperparameter to tune ☺
   3. Use the testing data to measure the accuracy and F1-score of your model.
2. [Text] Given the accuracy and F1-score of your model, are you satisfied with the results? Explain.
3. [Text] Show five example instances in which your model was incorrect. Describe why the model was wrong.

## Grading Rubric

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Points** | **Satisfactory (full points)** | **Unsatisfactory (no points)** |
| Task 1: Code Execution | 1 | Code runs without errors. | Errors prohibit code from running. |
| Task 1: Code Reproducibility | 1 | Code fully reproducible. | Code not fully reproducible: missing random number seeds, code order woes, etc. |
| Task 1: Code Organization | 1 | Code well organized. | Code mostly organized. Comments missing or are unclear/unhelpful. Minor issues in clarity around code. Code not easy to follow. |
| Task 1.a | 1 | Code clearly and fully performs the task specified. | Code does not perform the task completely. |
| Task 1.b | 2 | Code clearly and fully performs the task specified. | Code does not perform the task completely. |
| Task 1.c | 1 | Code clearly and fully performs the task specified. | Code does not perform the task completely. |
| Task 2 | 1 | Response clearly and fully answers the question. Response is well-justified and convincing. | Response does not clearly and fully answer the question. Response unclear, missing information, or incomplete. Response not fully justified or convincing. |
| Task 3 | 1 | Response clearly and fully answers the question. Response is well-justified and convincing. | Response does not clearly and fully answer the question. Response unclear, missing information, or incomplete. Response not fully justified or convincing. |
| Tasks 2- 3: Grammar, Usage, and Mechanics | 1 | Virtually no spelling, punctuation, or grammatical errors. Response is highly professional, clear, and well-formatted. | A number of spelling, punctuation, or grammatical errors. Response could be more clear, professional, and/or concise. Formatting could be improved. |

# Sentiment Analysis via ML-Based Approach

## Preamble

Download the “Product Sentiment” dataset: *sentiment\_train.csv* and *sentiment\_test.csv.*

## Tasks

1. [Code] Perform sentiment analysis on the dataset using the ML-based approach.
   1. Load, clean, and preprocess the data as you find necessary.
   2. Using the training data, extract features from the text (i.e., vectorization using BOW and/or Bag of N-Grams and/or topics and/or lexical features and/or doc2vec).
   3. Use your favorite ML algorithm to train a classification model. Don’t forget everything that we’ve learned in our ML course: hyperparameter tuning, cross validation, handling imbalanced data, etc. Make reasonable decisions and try to create the best-performing classifier that you can.
   4. Use the testing data to measure the accuracy and F1-score of your model.
2. [Text] Given the accuracy and F1-score of your model, are you satisfied with the results? Explain.
3. [Text] Show five example instances in which your model was incorrect. Describe why the model was wrong.
4. [Text] Compare and contrast the performance of the lexicon-based approach from Q2 with the ML-based approach here.

## Grading Rubric

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Points** | **Satisfactory (full points)** | **Unsatisfactory (no points)** |
| Task 1: Code Execution | 1 | Code runs without errors. | Errors prohibit code from running. |
| Task 1: Code Reproducibility | 1 | Code fully reproducible. | Code not fully reproducible: missing random number seeds, code order woes, etc. |
| Task 1: Code Organization | 1 | Code well organized. | Code mostly organized. Comments missing or are unclear/unhelpful. Minor issues in clarity around code. Code not easy to follow. |
| Task 1.a | 1 | Code clearly and fully performs the task specified. | Code does not perform the task completely. |
| Task 1.b | 2 | Code clearly and fully performs the task specified. | Code does not perform the task completely. |
| Task 1.c | 2 | Code clearly and fully performs the task specified. | Code does not perform the task completely. |
| Task 1.d | 1 | Code clearly and fully performs the task specified. | Code does not perform the task completely. |
| Task 2 | 1 | Response clearly and fully answers the question. Response is well-justified and convincing. | Response does not clearly and fully answer the question. Response unclear, missing information, or incomplete. Response not fully justified or convincing. |
| Task 3 | 1 | Response clearly and fully answers the question. Response is well-justified and convincing. | Response does not clearly and fully answer the question. Response unclear, missing information, or incomplete. Response not fully justified or convincing. |
| Task 4 | 1 | Response clearly and fully answers the question. Response is well-justified and convincing. | Response does not clearly and fully answer the question. Response unclear, missing information, or incomplete. Response not fully justified or convincing. |
| Tasks 2- 4: Grammar, Usage, and Mechanics | 1 | Virtually no spelling, punctuation, or grammatical errors. Response is highly professional, clear, and well-formatted. | A number of spelling, punctuation, or grammatical errors. Response could be more clear, professional, and/or concise. Formatting could be improved. |

# (Optional) Sentiment Analysis via Deep ML-Based Approach

As this is an optional bonus question, a maximum of 10% will be added to your mark.

## Preamble

Download the “Product Sentiment” dataset: *sentiment\_train.csv* and *sentiment\_test.csv.*

## Tasks

1. [Code] Perform sentiment analysis on the dataset using the ML-based approach, using a deep learning algorithm.
   1. Load, clean, and preprocess the data as you find necessary.
   2. Transform the data into an embedding appropriate for deep learning approaches.
   3. Use a deep learning algorithm to train a classification model. Don’t forget everything that we’ve learned in our ML course: hyperparameter tuning, cross validation, handling imbalanced data, etc. Make reasonable decisions and try to create the best-performing classifier that you can.
   4. Use the testing data to measure the accuracy and F1-score of your model.
2. [Text] Given the accuracy and F1-score of your model, are you satisfied with the results? Explain.
3. [Text] Show five example instances in which your model was incorrect. Describe why the model was wrong.
4. [Text] Compare and contrast the performance of the lexicon-based approach from Q2 with the ML-based approach here.

## Grading Rubric

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **Points** | **Satisfactory (full points)** | **Unsatisfactory (no points)** |
| Task 1: Code Execution | 1 | Code runs without errors. | Errors prohibit code from running. |
| Task 1: Code Reproducibility | 1 | Code fully reproducible. | Code not fully reproducible: missing random number seeds, code order woes, etc. |
| Task 1: Code Organization | 1 | Code well organized. | Code mostly organized. Comments missing or are unclear/unhelpful. Minor issues in clarity around code. Code not easy to follow. |
| Task 1.a | 1 | Code clearly and fully performs the task specified. | Code does not perform the task completely. |
| Task 1.b | 2 | Code clearly and fully performs the task specified. | Code does not perform the task completely. |
| Task 1.c | 2 | Code clearly and fully performs the task specified. | Code does not perform the task completely. |
| Task 1.d | 1 | Code clearly and fully performs the task specified. | Code does not perform the task completely. |
| Task 2 | 1 | Response clearly and fully answers the question. Response is well-justified and convincing. | Response does not clearly and fully answer the question. Response unclear, missing information, or incomplete. Response not fully justified or convincing. |
| Task 3 | 1 | Response clearly and fully answers the question. Response is well-justified and convincing. | Response does not clearly and fully answer the question. Response unclear, missing information, or incomplete. Response not fully justified or convincing. |
| Task 4 | 1 | Response clearly and fully answers the question. Response is well-justified and convincing. | Response does not clearly and fully answer the question. Response unclear, missing information, or incomplete. Response not fully justified or convincing. |
| Tasks 2- 4: Grammar, Usage, and Mechanics | 1 | Virtually no spelling, punctuation, or grammatical errors. Response is highly professional, clear, and well-formatted. | A number of spelling, punctuation, or grammatical errors. Response could be more clear, professional, and/or concise. Formatting could be improved. |