This week's Peer Review has two parts.

**Part 1**:

**For this week’s mini-project, you will participate in this Kaggle competition:**

[**Kaggle Competition: BBC News Classification**](https://www.kaggle.com/c/learn-ai-bbc/overview) **[80 pts]**

This Kaggle competition is about categorizing news articles. You will use matrix factorization to predict the category and submit your notebook for peer evaluation.

The part has 80 points. The instructions summarize the criteria you will use to guide your submission and review others’ submissions.

**You will submit one deliverable for Part 1:**

A Jupyter notebook with exploratory data analysis (EDA) procedure, model building and training, and comparison with supervised learning.

Suppose your work becomes so large that it doesn’t fit into one notebook (or you think it will be less readable by having one large notebook). In that case, you can make several notebooks or scripts in a GitHub repository (as deliverable 3) and submit a report-style notebook or pdf instead.

**Part 2:**

**Limitation(s) of sklearn’s non-negative matrix factorization library. [20 pts]**

1. Load the movie ratings data (as in the HW3-recommender-system) and use matrix factorization technique(s) and predict the missing ratings from the test data. Measure the RMSE. You should use sklearn library. [10 pts]

2. Discuss the results and why sklearn's non-negative matrix facorization library did not work well compared to simple baseline or similarity-based methods we’ve done in Module 3. Can you suggest a way(s) to fix it? [10 pts]

Three of your peers will review each of your three deliverables (Jupyter notebook or pdf report, GitHub repository, and challenge leaderboard screenshot) based on the rubrics for each deliverable.

Use the rubrics to guide your project to include all parts for the grade you want to achieve. The project has 100 total points.

**Extracting word features and show** **Exploratory Data Analysis (EDA) — Inspect, Visualize and Clean the Data (15 pts)**

Show a few visualizations like histograms. Describe any data cleaning procedures. Based on your EDA, what is your plan of analysis?

Please feel free to look at online resources on processing raw texts to feature vectors. Many methods process texts to matrix form (word embedding), including TF-IDF, GloVe, Word2Vec, etc. Pick a method and process the raw texts to word embedding. Briefly explain the method(s) and how they work in your own words. Also, do exploratory data analysis such as word statistics and/or visualization.

As we did not learn natural language processing (NLP) specific techniques such as word embeddings in the lectures, we recommend reading discussions and example codes from others in the Kaggle and/or doing some research online to make sure you understand. You can refer to any resource as needed, but make sure you “demonstrate” your understanding- please include explaining in your own words, discussions, and your interpretation. Also importantly, please have a reference list at the end of the report.

**Building and training models. [35 pts]**

In the Kaggle competition, the training data has labels (category). Thus, it can be solved using supervised learning. In general, the more labeled data we have, the more accurate the supervised learning model will be. But unsupervised learning can be powerful even when there is a small number of labels or no labels. ***This assignment will apply an unsupervised approach, especially the matrix factorization method,*** to discover topics in the news articles and use the labels to check the accuracy.

Here are some steps to guide this section: 1) Think about this and answer: when you train the unsupervised model for matrix factorization, should you include texts (word features) from the test dataset or not as the input matrix? Why or why not? 2) Build a model using the matrix factorization method(s) and predict the train and test data labels. Choose any hyperparameter (e.g., number of word features) to begin with. 3) Measure the performances on predictions from both train and test datasets. You can use accuracy, confusion matrix, etc., to inspect the performance. You can get accuracy for the test data by submitting the result to Kaggle. 4) Change hyperparameter(s) and record the results. We recommend including a summary table and/or graphs. 5) Improve the model performance if you can- some ideas may include but are not limited to; using different feature extraction methods, fit models in different subsets of data, ensemble the model prediction results, etc.

**Compare with supervised learning [30 pts]**

Use the following steps to guide your work:

1) Pick and train a supervised learning method(s) and compare the results (train and test performance) 2) Discuss comparison with the unsupervised approach. You may try changing the train data size (e.g., Include only 10%, 20%, 50% of labels, and observe train/test performance changes). Which methods are data-efficient (require a smaller amount of data to achieve similar results)? What about overfitting?

**Produce Deliverable: High-Quality, Organized Jupyter Notebook Report**

This project gives you hands-on experience on Kaggle, a famous platform for data science competitions. To further your professional development, think of your peers as work colleagues. An essential part of this project is the Peer Review.

**Complete Part 2**.

Limitation(s) of sklearn’s non-negative matrix factorization library. [20 pts]

1. Load the movie ratings data (as in the HW3-recommender-system) and use matrix factorization technique(s) and predict the missing ratings from the test data. Measure the RMSE. You should use sklearn library. [10 pts]

2. Discuss the results and why sklearn's non-negative matrix facorization library did not work well compared to simple baseline or similarity-based methods we’ve done in Module 3. Can you suggest a way(s) to fix it? [10 pts]