MIE 223 Data Science
Lab and Assignment 04:
Dealing with Natural Language

In this lab and assignment, you will learn and gain hands-on experience on the use of the nltk library and some concepts in natural language processing (NLP).

- Programming language: Python (Google Colab Environment)
- Due Date: Posted in Syllabus

Marking scheme and requirements: You are required to use the Jupyter notebook named `Assignment_04.ipynb` to provide appropriate answers to the questions in this assignment. All changes to the `Assignment_04.ipynb` notebook should be committed and pushed to your assignment repository on Github.

This assignment has *4 points* in total and the point allocation is shown below:

- Solution to Questions (1 point):
  - Q1: 0.1 point
  - Q2: 0.1 point
  - Q3: 0.1 point
  - Q4(a): 0.2 point
  - Q4(b): 0.1 point
  - Q5(a): 0.2 point
  - Q5(b): 0.1 point
  - Q5(c): 0.1 point
- Code review (1 point)
- Quiz (2 points)

What/how to submit your work:

- All your code should be included in the provided notebook named `Assignment_04.ipynb`.
- Commit and push your work to your github repository in order to submit it. Your last commit and push before the assignment deadline will be considered to be your submission. You can check your repository online to make sure that all required files have actually been committed and pushed to your repository.
- A link to create a personal repository for this assignment is posted on QUERCUS.

Notes that you should pay attention to

1. All your code should be written in the provided notebook named `Assignment_04.ipynb`.
2. All functions must *return* the specified return type.
3. Commit and push your work to your GitHub repository in order to submit it. **Your last commit and push will be considered to be your submission.** You can check your repository online to make sure that all required code has actually been committed and pushed to your repository.
4. **Your code should have output for all assignment questions unless otherwise stated**. If any of the assignment questions asks you to explain your outputs, please use markdown cells to clearly explain them.
5. As a backup plan, also upload a PDF copy of your assignment notebook to GitHub. This is useful especially when GitHub cannot render some images or outputs.
6. During the code review, TAs will go over your answer submissions to assess your understanding of the concepts relating to the assignment and the logic behind your codes. Please, ensure your codes are readable and well commented using python comments and markdown cells when necessary. Not have readable code may impact your code review score.
7. Please note the plagiarism policy in the syllabus.

1 Main Assignment

Please answer the questions below in the provided notebook named `Assignment_04.ipynb` using Google Colab. **Numpy** and **Pandas** functions should be used for all computations. Unless otherwise stated, the Pandas DataFrame '.plot()' function or **Matplotlib** should be used to generate plots. **Clearly give titles and labels to all plots**.

You are provided with a news dataset with topic classifications, called BABE-v2. This dataset contains news articles from various US news platforms, covering 22 controversial topics. The topics have been cleaned for you and the dataset is stored in a Pandas DataFrame as **"babe_df_cleaned"**. As a data scientist exploring this field, you want to be able to extract insights from these texts. Some questions you might want to consider are:

- How can we extract relevant, meaningful content from the texts?
- Can the words in the articles help us determine the topic?
- What can we learn from the text?

Using the **"babe_df_cleaned"** DataFrame, complete the following questions. Note: for **Q1-Q3**, fill in the provided functions. Modify the input **"df"** and return the modified **"df"**.

Q1. Text Cleaning

Using NLTK's Punkt tokenizer, tokenize all the cells in the 'text' column in the dataset into lowercase words. Save your results for each text cell as a Python list into a new column in **"babe_df_cleaned"** and name it 'tokens'. Run the second code block to print the tokens (as a Python list) for the first text cell (the one starting with "They are, she believes...").

Q2. Stop Words and Punctuation

Remove all stop words and punctuation from the tokens in the 'tokens' columns in **"babe_df_cleaned"** that you created in **Q1**. Overwrite the 'tokens' column in **"babe_df_cleaned"** with the new lists with stop words and punctuation removed. Run the second code block to print the tokens (as a Python list) for the first text cell (the one starting with "They are, she believes...").

Q3. Stemming

Using NLTK's Snowball Stemmer for "english", stem all of the tokens in the 'tokens' columns in **"babe_df_cleaned"**. Save the stemmed tokens in a new column called 'tokens_stemmed'. Do **NOT** overwrite the 'tokens' columns with the stemmed results since we will need the non-stemmed tokens later. Include any necessary imports in your code cell for the question. Run the second code block to print the stemmed tokens (as a Python list) for the first text cell (the one starting with "They are, she believes...").

Q4. Token Frequency

Use the 'tokens' column (non-stemmed tokens) in **"babe_df_cleaned"** to answer the following questions:

(a) Find the frequency of each token across all tokens in the dataset and plot the frequency of each token, sorted in descending order by frequency. Ensure you label the axes and title the plot. What is this distribution called?

(b) Print a DataFrame showing the top 10 words by frequency. Your first column should contain the top 10 tokens. Your second column should be labelled ‘Frequency’ and have the frequency for each of the top 10 tokens.

Q5. Mutual Information

(a) Using the ‘tokens’ column (non-stemmed tokens) in **“babe_df_cleaned”**, calculate the mutual information (MI) of the top 500 most common tokens with all the topics. Print the 10 tokens with the highest MI scores for each topic in a single DataFrame. Ensure the DataFrame contains ‘Topic’, ‘Word’, and ‘MI Score’ columns and that it is grouped by topic. Within each block of 10 consecutive rows for each topic, the words should be sorted in descending order by MI Score. Your output should look similar to the **“mi_scores”** DataFrane in the lab.

(b) Repeat **Q5a** using the stemmed tokens (‘tokens_stemmed’ column in **“babe_df_cleaned”**) instead of the non-stemmed tokens.

(c) What can the MI score help us learn about a given topic? Which table (**Q5a** or **Q5b**) was more useful in drawing this conclusion? Why do you think so? Answer in 1-2 sentences.