Lab 1: The objective of this lab is to make use of the Bag-of-Words model in NLP and observe its effect in text classification

Learning objectives:

- Utilize existing NLP libraries (e.g. spacy) to:
    - Preprocess the text data and tokenize it
    - Create a Bag-of-Words model
    - Make use of the specified machine learning models for text classification
- Observe the difference of the text classification performance with and without the Bag-of-Model

Procedure:

- Download the zip file called *Lab1_Week4_empty.py* and extract it. This zip file contains:
    - An empty Python code for you to fill in
    - The dataset to be tested on
    - This manual for the lab
- The Python code consists of several comments that indicate where you should fill in
    - First, make use of the preprocessing and cleaning steps on the given text. You will pass this preprocessed data as input the specified machine learning technique
    - Then, create the Bag-of-Words model. You will pass this model as input into the specified machine learning technique.
        - You will compare the performance of the ML models with and without Bag-of-Words
    - Then, implement Support Vector Machine (i.e. SVM(...)) and Logistic Regression (i.e. LR(...))
        - To reiterate, you will compare the performance of both these ML models with and without Bag-of-Words.

Question:

1. Please describe whether the performance of SVM and LR change when you consider the Bag-of-Words. If there is a difference in performance, then please explain.
2. Please explain the reason behind the change of performance (or lack thereof).

Deliverables:

- Your code that fills in the blanks as indicated by the comments in *Lab1_Week2_empty.py*
- A report that details the results you have found. It should contain the following information:
    - Introduction to the Bag-of-Words model
    - Answers to the above questions. Include the results from your code (as specified by the questions)

    - Concluding remark of what you have learned from this lab

Your implemented code should have the following (you will be marked for both):

- Correctness: The code should compile without error
- Style: There should be comments and proper formatting