Individual Assignment (25% of the total module marks)

In this part of the assignment, you are expected to carry out **practical text pre-processing, basic processing, and text normalization** in Python using Natural Language Toolkit (NLTK) or other techniques such as Regular Expression (re), and textblob.

Work on the following questions and write suitable answers along with Python codes. Share the relevant codes in the appropriate position to support each answer in the report.

Text Data

Use **Data\_1.txt** to answer Q1 to Q3 and use **Data\_2.txt** to answer Q4 given in Individual Assignment Data.zip.

Q1. Form tokenization (25 marks)

1. Demonstrate sentence segmentation and report the output. (6 marks)
2. Demonstrate word tokenisation using the *split function*, *Regular Expression and* *NLTK packages* separately and report the output. (10 marks)
3. Explain the differences of the tokenisation operations performed in Q1.2 using the reported output. (4 marks)
4. Justify the most suitable tokenisation operation for text analytics. Support your answer using obtained outputs. (5 marks)

Q2. Form word stemming (25 marks)

1. Explain the importance of stemming in text analytics (4 marks)
2. Explain the differences among *Regular Expression stemmer and Porter Stemmer* use for word stemming. (6 marks)
3. Demonstrate and report the output for word stemming using any techniques such as *Regular Expression, or Porter Stemmer*. (12 marks)
4. Justify the most suitable stemming operation for text analytics. Support your answer using the obtained output. (3 marks)

Q3. Filter stop words and punctuation (25 marks)

1. Demonstrate *stop words and punctuations* removal from the given text corpus and report the output suitably. (12 marks)
2. Report the stop words found in the given text corpus. (6 marks)
3. Explain the importance of filtering the stop words and punctuations in text analytics. (7 marks)

Q4. Form Parts of Speech (POS) taggers & Syntactic Analysers (25 marks)

1. Demonstrate POS tagging using *NLTK POS tagger,* the R*egular Expression tagger* and report the output. (9 marks)
2. Explain the differences of the POS taggers using the output obtained in the above question. (6 marks)
3. Justify the most suitable POS tagger for text analytics. Support your answer using the output obtained. (5 marks)
4. Draw possible parse trees for the given sentences using suitable python codes and report the Parse Trees along with the Python code. (5 marks)

Deliverables

**Report:**

Word count: 2000 words

The report must be prepared in a professional manner following the proper documentation aspects.

**Codes:** Python code should run without any arguments. It should read files in the same directory. The output must be as specified in respective questions. Suitable comments must be inserted in proper places in the code.

**Softcopy:** The relevant softcopies such as the report *(.doc or .docx or .pdf*) and the python code files *(.py or. ipynb)* must be uploaded via the specified link in the MOODLE.

Academic Integrity

Copying or paraphrasing someone's work (code included) or permitting your own work to be copied or paraphrased, even if only in part, is not allowed, and will result in disciplinary action. Your grade should reflect your own work.

Basically, 'plagiarism' means representing someone else's work as if it is your own. This is a very serious academic offence for all students within the University regulations and is particularly reprehensible for a researcher. Please do not even consider it. Remember that accidental plagiarism (or the appearance of it) may be avoided by referencing your work properly. This gains you credit, not loses it! The simple rule is that you must not represent the ideas of other people (whether they are published works or the work of other students) as your own.

Carefully read APU guidelines and policies on the Webspace. The golden rule on plagiarism is **DO NOT DO IT**.