Group Assignment 2
Introduction to Python for Data Science, AM02

Use **`AM02_Group_Assignment_2.py`** template to answer the Qns.

Q1. Twitter Data Analysis

I have collected some Twitter data for the period 25/03/2020 to 25/04/2020 and stored in the file `Tesla_Share_Tweets_1Mth` pandas dataframe (we will read it into a list). It contains tweets with words 'Tesla Share'.

This is a guided question designed to take you through an applied real-life problem of tweeter data analysis with the aim of showing you the importance of data cleaning, pre-processing, and obtaining overall tweet sentiment score with Natural Language Processing tools. Are people talking positively or negatively about Tesla? Such information often forms input into trading models.

Part I – Tweets Preprocessing:

a) [Code provided] Load the provided libraries. If you do not have some of them use the installation instructions at the top of the .py file.

b) [Code provided] Load the tweets data into a dataframe called `df_all`

c) Examine `df_all` by double clicking on the variable name from the Variable explorer.

d) [Code provided] Extract just the text part of the tweets and store it inside a list (each tweet's text should be an individual element within a list called `tweets`).

e) Find number of obtained tweets and store it into variable N.

f) Print out the 5th tweet.

g) Remove duplicates from the list `tweets` (these may occur due to retweets / news repeats etc). You must think of the fastest way of doing this in Python i.e. on a single line with very little effort. Store results into a list called `twsUnclean`.
Hint: this could involve a `set()` operation.

h) Report number of tweets obtained in `twsUnclean`.

i) [Code provided] Examine how `preprocesor` library's function `clean()` works using the provided code.

j) Given this information write a list comprehension to clean all tweets of hashtags, emojis and links. Store result into variable called `tws1` and print out the first 5 tweets.

k) [Code provided] There are some non-English characters and numbers, punctuation etc that is present inside tweets. These can be removed in one go using the regular expression library's function `re.sub()`.
The code provided uses list comprehension to replace any characters in a string that are not either lower-case or upper-case letters with a space.
Print out the first 5 tweets of `tws2`.

l) Use list comprehension to strip away any start/end whitespaces and convert all letters to lower case for each tweet. Store result into variable `tws3` and print out first 5 tweets.
Hint: you can chain methods using dot notation, e.g.: `s.method1().method2()`

m) As the last step, delete empty tweets using list comprehension. Save result to `tws4`.
Hint: you can think of the qn as keeping only tweets which do not have zero length.

Part II – Sentiment Analysis:

a) [Code provided] Familiarise yourself with the Natural Language Processing library in Python called `nltk` by reading through the provided code, comments and examples.

b) Write code which would provide an **average 'compound' score** for all clean Tesla tweets stored in `tws4`, i.e. you should obtain a single number.
Note: 'compound' score for each tweet is obtained from the result of `polarity_score()` function's returned value, for example:
`score = sid.polarity_scores('something neurtral')`
we see a compound score of 0.0:
`{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound': 0.0}`
Use this information to obtain the average of all compound scores obtained for tweets.

c) Comment on the final score that you have found and state whether you believe this has negative or positive sentiment. State your intuition on the strength of this value.

Part III – Wordcloud in Python:

a) [Code provided] This is a fun demonstration to show you how wordclouds could be created in Python (you have already seen how this was done in R), both the standard way, and using an image to fill it with the most frequent words. Here I used an image of the Tesla Cybertruck and filled it with Tesla's most frequent words in tweets.

This is what you should obtain if you manage to run the code correctly:

Q2. Federal Reserve Statement Analysis

The Federal Reserve decides on the base interest rates in the USA at regular intervals throughout the year and their decision depends on the economic outlook and inflation rate (among other things). Following each Fed interest rate decision (released at 7.00pm UK time), an accompanying document with information on the decision called the "Statement" is also released. You have already had some experience with one such statement which you scrapped from the FOMC website. Each statement is closely watched by the financial markets for the occurrence of certain words which **signal what the Fed plan to do with interest rates in the near future**, such as: raising rates / pausing / lowering the base rate. As you can image, as such new information arrives into the market it tends to move it if a surprise wording in the statement is encountered.

Due to the cov-19 the Fed have dropped the interest rates to 0%, yet they have been hiking (increasing) rates for over the previous 2.5 years from 0.50% Dec 2016 to recent level of 2.5%.

A closely watched Fed phrase at the time of the hikes was: "The Committee **expects** that **further gradual increases** in the target range for the federal funds rate will be consistent with sustained expansion…" Meeting dates: Jan, Mar, Jun and Sep 2018 statements. For e.g.: Sept 26th, 2018. (Rate decision raised from 2% to 2.25%). The expects word was observed each month and kept telling the markets that the Fed plan to keep raising rates.

In their Dec 19th 2018 meeting, this phrase was amended to "The Committee **judges** that **some further gradual increases** in the target range for the federal funds rate will be consistent with sustained expansion…" (Rate decision: rates raised from 2.25 to 2.5%). The markets expected the word "further" to be dropped entirely thus indicating that Fed will substantially ease off the further expected tightening (rate rises) due to recession concerns at the time. However, the Fed used the word 'judges' to indicate to the markets that the Fed will be more data driven, and the word 'some' to indicate that they were indeed slowing down the pace of raises. The stock markets however sold off on the release of the statement because they considered it to be a lot more hawkish (in favour of interest rate rises) than the dovish (interest rate cuts) tone that the markets expected.

Event driven / intraday traders could gain execution advantage over traders relying on reading the news on Bloomberg / hearing it on the squawk, by using software such as Python to analyse the FOMC statement in real time for the words contained in the text.

The two sentences from the Sept and Dec statements are already provided inside the template .py file, these are called `fomc_sep2018` and `fomc_dec2018`.

a) What is the type of each variable?
b) How are the 2 sentences represented inside these variables? i.e. is the sentence one long string or are words separated out as elements?
c) Check how many elements are in each container (i.e inside `fomc_sep2018` and `fomc_dec2018`).

Your task is to write a Python function which takes as input 2 strings (in the order: older statement sentence as its first argument, and later statement sentence as the second argument) and returns as output the following information:

d) All words occurring in two sentences.
e) Words which were common for both sentences.
f) Words which were NOT used in the latest sentence.
g) Words which were NEW in the latest sentence.

Note: I am purposefully giving you the flexibility to design code in a way you think is best to solve this real-life task. So long as code works efficiently and is within the parameters of the qn (e.g. I am asking you to solve this using a function), the answer will be accepted in full.

h) Make a function call with appropriate arguments and store outputs from the function into variables with names of your choice.
i) Report each of the returned variable contents inside a docstring (this is for the ease of grading your work and seeing if you have correct results).