CSCI - 5901 - The Process of Data Science - Summer 2019
Assignment 2

Due date: July 17th, 2019 11:59:59 pm.
The submission must be done through brightspace.
Teams of 2 students

- Cite any and all resources used.
  - books, websites (other than documentation) like stackoverflow.
- I will use plagiarism tools to detect any type of cheating and copying (your code and PDF).
- Write all of your comments and explanations in the code as a text cell.
- Two sides of Data Science (and your mark):
  - Technical // does it work?
    - Quality of code (documentation, naming)
    - Does it work
  - Conceptual // what did you find from the data?
    - Quality of the data insights
    - Quality of the discussion text
  - Questions will be marked individually. Their weights are shown in parentheses after the question number.
- Your submission is a single Jupyter notebook and a PDF (With the compiled results generated by your Jupyter notebook). Filename should be **A2-<your_name1>-<your_name2>.jpynb** and **A2-<your_name1 >-<your_name2>.pdf**. Please upload to Brightspace. Please include your B# in your Jupyter notebook and PDF.
- **Forgetting to submit both files results in 0 markings for both students.**

Q1. Collocation extraction. (**35 marks**)
In this assignment, we will use part of the 20 newsgroups dataset, which is part of Scikit learn.
https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html
The part of interest referred to as the corpus consists of the following four newsgroups: consider alt.atheism, talk.religion.misc, comp.graphics, sci.space
a. Tokenize this corpus and perform part-of-speech tagging on it. (**5 marks**)
b. Apply the techniques described in Tutorial 6 (Frequency with filter, PMI, T-test with filter, Chi-Sq test) to extract bigram collocations from the corpus. Show the top 20 results of each technique. (**15 marks**)
c. How much overlap is there among the techniques? Do you think it makes sense to consider the union of the results? (**15 marks**)

Q2. SVM and NB for Text Classification (75 marks)

In this part, you will play around with SVM and Naive Bayes for text classification on the corpus of Q1.

a. Clean the text: (**10 marks**)

- Remove stop words
- Remove numbers and other non-letter characters
- Stem the words

b. Study the section on feature extraction in Scikit Learn, https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction and convert the corpus into a bag-of-words tf-idf weighted vector representation. (**10 marks**)

c. Split the data randomly into training and testing set (70-30 %). (**5 marks**)
Train SVM and report confusion matrix. (**5 marks**)
Train Multinomial NB and report confusion matrix. (**5 marks**)
Which algorithm has higher accuracy and why? (**5 marks**)
Does changing the kernel of the SVM change the accuracy or decrease confusion between classes? (**15 marks**)

d. Perform part-of-speech tagging on the raw data (i.e. prior to cleaning it), clean as in part (a) above, and extract the nouns only to obtain a bag-of-words tf-idf weighted vector representation using only the nouns. Repeat question (c). How does this accuracy compare with that of part (c)? How does the size of the vocabulary compare with that of part (c)? (**20 marks**)

More tips for reaching full marks in this assignment.

- Be selective in the experimental results you present. You don't need to present every single experiment you carried out. It is best to present only the interesting results, where the behaviour of the ML model behaves differently.
- In your answer, you want to demonstrate skill in using the sklearn library to experiment with ML models, and good understanding/interpretation of the results. Make it easy for the marker to see your skill and your understanding.
- Justify your hyperparameter choices.
- Minimize the code you write from scratch. If there is a procedure in sklearn that does the task, find it, read its documentation and use it.
- Add liberally markdown cells to explain your code and experimental results. Make good use of the formatting capabilities of markdown to make your notebook highly readable.