Homework 1 - Due February 1st, 2021 at 11:59 Eastern Standard Time

Objective: The purpose of this homework is for you to gauge how comfortable you are with regular expressions and text classification.

How to do this assignment: Using Google Colab (https://colab.research.google.com/) please answer the following questions in a new notebook. You are to write the question on a Text field, and your programming answer on a Code field. Please write your name and student ID number on a Text field at the very beginning of this notebook. Note that when you are testing your code, the output needs to be shown

What to submit: Via iCollege you are to download a copy of the executed Colab notebook in PDF format, and submit this file ALONGSIDE a link (in the comments/description section) to your own GitHub repository where this Colab notebook is stored. In other words, you are expected to keep your own personal GitHub repository for this class and place ALL your assignments and project code in that location.

Questions (150 total points): *NOTE: You can only use Python's built in regular expressions (re) and scikit-learn as your ML library for these exercises. Any code using any other libraries will not be graded.*

TEXT: Using Shakespeare's "The Life and Death of Julius Caesar" found as hw_1-text.txt under this iCollege assignment. Write the following pieces of code:

1. Write a piece of code using a regular expression in Python to count how many times the following name appears: CICERO, this regex needs to be case-sensitive. (10 points)

2. Write a regular expression in Python that finds all 6 character words found in this play. (10 points)

3. Write a piece of code that uses a regular expression in Python to count how many words are capitalized in the first letter only. (20 points)

4. Write a regular expression using groups in Python that extracts all phrases that have "son of " and extract/output the answer in a case-insensitive way. For example, for the following "I am the son of earth", should find "son of earth" and extract/output "earth". (20 points)

5. Plays are divided in acts, write a regular expression using groups in Python that extracts the number of each act found in this play in a case-sensitive way. (20 points)

6. Write a Python function that takes the file as an input and outputs how many lines end with: ? , that is a question mark at the end of the line. This function needs to use a regular expression. (20 points)

7. Using the code from Class 04, train a Random Forest classification model for the same two newsgroups, generate the confusion matrix, and all the same performance metrics. (50 points)

Bonus (15 points): Using the code from Class 04. Write a function to find the best performing model for the TF-IDF vectorization. The requirements for this are: 1) Pass all metrics for each model in a single dataframe (each row has a metric value, each column is a model type). 2) Make the decision of the best model by counting which column has the highest F1-score and accuracy.