Dr. Dimitrios Letsios
Department of Informatics
King's College London

Semester 2
Friday 5 March 2021

7CCSMDM1 Data Mining

Coursework 2

Due: Friday 26 March 2021 (23:59 UK time)

This coursework assignment is relevant to mining text and image data. The coursework is worth 10% of the overall module mark and will be marked out of 100 points. The distribution of points is (i) 60 points for the first part on **text mining** and (ii) 40 points for the second part on **image processing**. The data required for this coursework are provided in the KEATS page of the module. No need to download them from their original sources. The links to their origins are provided for referencing purposes. The **submission instructions** are given in the last part of this document.

1 Text Mining

This part uses the **Coronavirus Tweets NLP** data set from Kaggle `https://www.kaggle.com/datatattle/covid-19-nlp-text-classification` to predict the sentiment of Tweets relevant to Covid. The data set (**Corona_NLP_test.csv** file) contains 6 attributes:

Attribute	**Description**
UserName	Anonymized attribute.
ScreenName	Anonymized attribute.
Location	Location of the person having made the tweet.
TweetAt	Date.
OriginalTweet	Textual content of the tweet.
Sentiment	Emotion of the tweet.

Because this is a quite big data set, use vectorized (e.g. pandas / numpy) built-in functions to effectively perform the various tasks with a typical personal computer. In this way, you will be able to run your code in few seconds. Otherwise, running your code might require a significant amount of time, e.g. in the case where *for loops* are used for accessing all elements of the data set. Marks will be reduced if your code does not use vectorization. Further, you are expected to use raw Python string functions for text processing operations.

1. *[20 points]* Compute the possible sentiments that a tweet may have, the second most popular sentiment in the tweets, and the date with the greatest number of extremely positive tweets. Next, convert the messages to lower case, replace non-alphabetical characters with whitespaces and ensure that the words of a message are separated by a single whitespace.

2. *[20 points]* Tokenize the tweets (i.e. convert each into a list of words), count the total number of all words (including repetitions), the number of all distinct words and the 10 most frequent words in the corpus. Remove stop words, words with ≤ 2 characters and recalculate the number of all words (including repetitions) and the 10 most frequent words in the modified corpus. What do you observe?

3. *[10 points]* Plot a histogram with word frequencies, where the horizontal axis corresponds to words, while the vertical axis indicates the fraction of documents in a which a word appears. The words should be sorted in increasing order of their frequencies. Because the size of the data set is quite big, use a line chart for this, instead of a histogram. In what way this plot can be useful for deciding the size of the term document matrix? How many terms would you add in a term-document matrix for this data set?

4. *[10 points]* This task can be done individually from the previous three. Produce a Multinomial Naive Bayes classifier for the Coronavirus Tweets NLP data set using scikit-learn. For this, store the corpus in a numpy array, produce a sparse representation of the term-document matrix with a CountVectorizer and build the model using this term-document matrix. What is the error rate of the classifier? You may want to check the scikit-learn documentation for performing this task.

2 Image Processing

Use the provided image data for performing image processing operations with **skimage** and **scipy**. The data set consists of the following 4 images:

File	**Source**
avengers_imdb.jpg	`https://www.imdb.com/`
bush_house_wikipedia.jpg	`https://en.wikipedia.org/`
forestry_commission_gov_uk.jpg	`https://www.gov.uk/`
rolland_garros_tv5monde.jpg	`http://www.tv5monde.com/`

Each of the following questions requires producing one or more new images. Every image that you produce (and has been requested) should be stored in a folder that you will name **outputs**. For each question, briefly explain what has been achieved and how it could be useful in the context of the corresponding image.

1. *[8 points]* Determine the size of the *avengers_imdb.jpg* image. Produce a grayscale and a black-and-white representation of it.

2. *[12 points]* Add Gaussian random noise in *bush_house_wikipedia.jpg* (with variance 0.1) and filter the perturbed image with a Gaussian mask (sigma equal to 1) and a uniform smoothing mask (the latter of size 9x9).

3. *[8 points]* Divide *forestry_commission_gov_uk.jpg* into 5 segments using k-means segmentation.

4. *[12 points]* Perform Canny edge detection and apply Hough transform on *rolland_garros_tv5monde.jpg*.

Instructions

- Implement this coursework on your own. You may discuss solution strategies with classmates, but you must individually write your own code and report. Violation of this rule will be considered as an act of misconduct.

- Submit a zip file with a (i) **report** in pdf format answering the questions posed in each part of the coursework, (ii) **Python code** in .py format (**not** .ipynb iPython Notebooks) for generating the answers (one .py file for each part of the coursework), and (iii) a **readme** plain text file briefly explaining what your source code does. Do **not** add any source code in the report.

- Submit the zip file via KEATS submission link.