Comp6940 Assignment 4

Date Due: Friday 13th April, 11:59pm

Warm Up (20 marks)

You are given the following list of "Documents" where there exist only 3 words: "Apple", "Orange", and "Banana". Every sentence or document is made up of these words and they become the basis of a 3 dimensional vector space. The "sentence" or "document" is simply a linear combination of these vectors where the number of appearances of the words is the coefficient along that dimension:

```
corpus = ['Apple Orange Orange Apple',\ # [ 2.  0.  2.]
  'Apple Banana Apple Banana',\ # [ 2.  2.  0.]
  'Banana Apple Banana Banana Banana Apple',\ # [ 2.  4.  0.]
  'Banana Orange Banana Banana Orange Banana',\ #[ 0.  4.  2.]
  'Banana Apple Banana Banana Orange Banana'] # [ 1.  4.  1.]
```

Write a Python function (from scratch) which takes the corpus and creates a vector representation of the corpus. The `pandas` or `numpy` structure may be used.
The comment at the end of each line represent the corresponding vector representation.

The function should output the following based on the corpus given above:

```
[[ 2.  0.  2.]
 [ 2.  2.  0.]
 [ 2.  4.  0.]
 [ 0.  4.  2.]
 [ 1.  4.  1.]]
```

Preprocessing and Data Organization (20 marks)

The dataset can be downloaded here

Tasks:

1. Create a new column in the dataframe called 'sentiment'. Using the like and dislike counts, populate the new column with 0's and 1's where 0 refers to a negative sentiment and 1 refers to a positive sentiment.
2. Clean the subtitles data and store the cleaned text in a new column 'subtitle_clean'.
    1. For each step of your text cleaning give a brief explanation of why you chose to perform that method on the text.
3. Use `TFIDFVectorizer` and `CountVectorizer` to encode the clean subtitle.s

Text Classification (30 marks)

1. When choosing a metric to access the performance of your classifier provide a brief explanation

of why you chose that metric.
2. Perform the following classification experiments keeping track of the performance of each classification task for future use:
    1. Logistic regression model on word count
    2. Logistic regression model on TFIDF
    3. Logistic regression model on TFIDF + ngram
    4. Support Vector Machine model on word count
    5. Support Vector Machine model on TFIDF
    6. Support Vector Machine model on TFIDF + ngram

    You may use the SVM classifier from sklearn
3. Plot a bar graph showing the performance of each of the experiments.

Topic Modeling (20 marks)

1. Using TFIDF and Count Vectorizer models imported for `sklearn`, perform topic modelling using the following topic modeling algorithms:
    1. NMF
    2. LDA
    3. SVD.
2. When choosing the number of topics give a brief explanation of why that number was chosen.
3. Discuss based on the top 10 words each of the algorithms choose for each topic cluster what category the topics fall under.

Visualization (10 marks)

1. Choose the clusters obtained from a topic model algorithm from above and plot a word cloud for each of the clusters. For example, if the number of topics chosen was 10 and the topics were obtained from the SVD algorithm, 10 word clouds should be plotted.

Submission Details:

Students should submit either the following:

- An exported Jupyter notebook of the assignment with notes

or

- A python script file with the assignment and appropriate comments

Email submissions to:

nchamansingh@gmail.com

With the following subject format: *< StudentID > COMP6940 Assignment 4*

Late Submissions:

A daily penalty for late submissions of 15% per day or partial day