Heather Hanwacker

MET CS 777 – Spring O2 2022

March 30, 2022

**Assignment 2 – Read Me**

kNN classifier (20 points)

A screenshot of a computer

Description automatically generated with medium confidenceGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text

Description automatically generated

The overall code created ran on the small data set consistently in less than 2 minutes. Moving to the larger dataset became a challenge where over time the worker nodes would lose their connection permanently and the job would not become successful. To be able to complete each task and understand the run time. I ran task 1 and task 3 separately to achieve results for the larger dataset. Then went back and optimized task 2 based on suggestions in the class discussion. To confirm the data set and results were working, I printed in the output the size of the data and words from the corpus. Below are the results

1000000

Top Words in Corpus: [('the', 26451056), ('of', 12507151), ('in', 10807932), ('and', 10758007), ('a', 7988396), ('to', 7681799), ('was', 4588136), ('is', 3746524), ('for', 3145951), ('as', 3066049)]

Word Positions in our Feature Matrix. Last 20 words in 20k positions: [('sizeable', 19999), ('unlocked', 19998), ('dent', 19997), ('sadler', 19996), ('lockout', 19995), ('minerva', 19994), ('ub', 19993), ('ericsson', 19992), ('ami', 19991), ('erica', 19990), ('yarmouth', 19989), ('divert', 19988), ('overt', 19987), ('csa', 19986), ('melancholy', 19985), ('madurai', 19984), ('rida', 19983), ('corbin', 19982), ('estadio', 19981), ('ethnographic', 19980)]Graphical user interface, text

Description automatically generatedText, letter

Description automatically generated

Task 1 Large Dataset output:

allDocsAsNumpyArrays.take(3)

A screenshot of a computer

Description automatically generated with medium confidence

allDocsAsNumpyArraysTFidf.take(2)

Text

Description automatically generated

Task 1 - Large Dataset Spark Screenshot:

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Text

Description automatically generated

Task 2 Small Dataset output:

print(getPrediction('Sport Basketball Volleyball Soccer', 10))

Text

Description automatically generated

print(getPrediction('What is the capital city of Australia?', 10))

Text

Description automatically generated

print(getPrediction('How many goals Vancouver score last year?', 10))

Text

Description automatically generated

Task 2 Small Dataset Spark Screenshot:

A picture containing table

Description automatically generatedTable

Description automatically generatedA computer screen capture

Description automatically generated with medium confidence

Task 2 Large Dataset output:

print(getPrediction('Sport Basketball Volleyball Soccer', 10))

Text

Description automatically generated

print(getPrediction('What is the capital city of Australia?', 10))

Text

Description automatically generated

print(getPrediction('How many goals Vancouver score last year?', 10))

Text

Description automatically generated

Task 2 Larger Dataset Spark Screenshot:

Timeline

Description automatically generated Graphical user interface, application, table

Description automatically generated

For the large data set I changed the location of the join to later in the code. This was based on a suggestion in the class discussion. By doing this I was able to reduce the number of categories being joined and speed up the code drastically. This helped to ensure that the code would not fail.

Text

Description automatically generated

Task 3 Large Dataset output:

3.1 – count, mean, stddev, min, max

Text

Description automatically generated

3.1 – median



3.2 – Top 10

Text

Description automatically generated

Task 3 Large Dataset Spark Screenshot:

Timeline

Description automatically generatedGraphical user interface, table

Description automatically generated

Graphical user interface, text, application

Description automatically generated with medium confidence Graphical user interface, text, application

Description automatically generated with medium confidence

Removing stop words would be beneficial when it comes to viewing the top words in the corpus and may speed up the computations because there will be a smaller number of features. However, the results using TF-IDF with the final kNN results would not be changed heavily after removing stop words. The reason for this is stop words are very common in all documents. It is very unlikely that a stop word would be influential in the outcome unless it was an uncommon stop word which goes against what the definition of a stop word is. Thinking back to the professor’s class example of calculating TF-IDF from word frequencies. Stop words are going to be common in most of the documents which results in giving less weight to those words when doing TF-IDF calculations. This is one of the main goals of IDF, to remove words with no semantic value from the corpus.

Graphical user interface, text, application

Description automatically generated with medium confidence

Stemming could change the outcome of the final kNN results. It may or may not heavily change the outcome, that would depend on the corpus and type of documents you are using. Stemming essentially would result in grouping all words that are derived from the same stem. Grouping these words would increase the occurrences of the stem which can either cause a stem word to be more frequent in a document or overall, more frequent in the corpus. For example, if you are looking at a document about a video game tournament. The word game, games, gamed, gaming, etc. would most likely show up multiple times. By grouping these occurrences together, it will make the stem “game” more dominant to the classification if we try to classify a document with the same stem word of game. This is useful if you are trying to classify between a video game tournament document and cat video document. However, if you are trying to classify between different types of gaming documents (board games, video games, sports games, etc.). By using stemming, it will ensure that the word game is not as meaningful in the classification.

Files Included in submission:

* **readme\_assignment2**– current file that explains assignment and results
* **Screenshot\_main\_\_1\_and\_3\_spark\_job**– screenshot of page with 2 job runs with start time and elapsed time (task 1 and task 3)
* **Screenshot\_main\_\_2\_sm\_lg\_spark\_job** – screenshot of run time of spark job for task 2 on small and large data
* **Screenshot\_main\_\_small\_spark\_job** – screenshot of run time of spark job for small data set on entire code/task
* **main\_assignment2.py** – py file that has all the tasks combined
* Task 1: Generate the Top 20K dictionary and Create the TF-IDF Array
  + **main\_assignment2\_task1.py**
  + Input – first argument is data set Wikipedia pages, second argument is data set wiki category links, third and fourth arguments is where to store output
  + Output
    - **assignment2\_1.1\_lg\_part-00000**– allDocsAsNumpyArrays.take(3)
    - **assignment2\_1.2\_lg\_part-00000**– allDocsAsNumpyArraysTFidf.take(2)
  + **Screenshot\_main\_\_task1\_spark\_job**– Screenshot of run time of spark job
* Task 2:
  + **main\_assignment2\_task2.py**
  + Input – first argument is data set Wikipedia pages, second argument is data set wiki category links, third, fourth, and fifth arguments is where to store output
  + Output
    - **assignment2\_2.1\_lg\_part-00000** – print(getPrediction('Sport Basketball Volleyball Soccer', 10))
    - **assignment2\_2.2\_lg\_part-00000** – print(getPrediction('What is the capital city of Australia?', 10))
    - **assignment2\_2.3\_lg\_part-00000** – print(getPrediction('How many goals Vancouver score last year?', 10))
  + **Screenshot\_main\_\_task2\_spark\_job** – Screenshot of run time of spark job
* Task 3:
  + **main\_assignment2\_task3.py**
  + Input – first argument is data set Wikipedia pages, second argument is data set wiki category links, third, fourth, and fifth arguments is where to store output
  + Output
    - **assignment2\_3.1\_lg\_part-00000** – summary statistics
    - **assignment2\_3.2\_lg\_part-00000** – summary statistics (median)
    - **assignment2\_3.3\_lg\_part-00000** – file with top 10 most used Wikipedia categories output
  + **Screenshot\_main\_\_task3\_spark\_job** – Screenshot of run time of spark job
* Task 4:
  + None – conceptual questions