Name of the course: Machine Learning for Trading
Project Name: Project 1 Part B: Compute tf-idf vector for example files
Student Name: Akshat Harit
GTID: 903090915

tfidf.py

The program can be abstracted as the following steps:

1. **Input the file**
   This is done using the fileinput module. The file is read twice with the first pass being used to calculate the number of files and the unique words. A processing function is used that strips numbers, punctuation, other non-printable characters and converts text to lower case. Words are put into a set data structure, which allows the calculation of unique words easily. This allows us to create the matrix precisely in one go without wasting space.
2. **Make a matrix for words and files, detailing what frequency a word occurs in a particular file**
   The matrix is created based on the number of files and number of words. It is initialized with zeros. Then the files are read again, with the matrix being populated this time with the word frequency. A dictionary of file names and words is used to facilitate the communication between matrix elements that are referenced by numbers i.e. row number and column number and files and words that are referred by text. Numpy is used in the creation of matrix. This allows us to directly use matrix operators for the upcoming calculations
3. **Compute idf**
   idf of a word=log(Number of files/ Word occurs in the file)
   The idf is calculated by using a scalar operation on the matrix derived from step 2. The difference being that all frequencies other than zero are set to 1 to facilitate the sum operation along the axis of files. Float typecasting is done to avoid integer division truncation, though it is probably handled by numpy. This is to be investigated further.
4. **Compute tf and tf-idf in a loop**
   tf and tfidf are calculated in a single loop that iterates over another matrix that is of the same size as in step 2.
   tf of a word in a file=word count/ max(word count in that file)
   tfidf=tf of word in a file*idf of a word in sets of file
5. **Print the results to the console**
   A preliminary step is done to make a list of file dictionary. As dictionary is an unordered structure, it gives no guaranteed order of traversal. A list is made to ensure an order in the ensuing printing. This could also be done for word dictionary, but isn't as order doesn't appear to be an inherent property when larger files will be used.. The results are then printed corresponding to each word and file, with a header line printed of different file names.

The output of the following line of code is as follows(python2 used for Python 2.7)

```
$python2 ballmer*.txt > output.csv
$cat output.csv
```

```
------------------------------------OUTPUT STARTS------------------------------------------------------------
term, ballmer01.txt,ballmer02.txt,ballmer03.txt,ballmer04.txt,ballmer05.txt,ballmer06.txt,ballmer07.txt
named,0.0,0.0,0.0,1.94591014906,0.0,0.0,0.0
retired,0.0,0.0,0.972955074528,0.0,0.0,0.0,0.0
give,0.0,0.0,0.0,0.0,1.94591014906,0.0,0.0
```

```
nadella,0.0,0.0,0.0,1.94591014906,0.0,0.0,0.0
it,0.0,0.0,0.0,0.0,1.94591014906,0.0,0.0
steve,0.0,1.94591014906,0.0,0.0,0.0,0.0,0.0
as,0.0,0.0,0.972955074528,0.0,0.0,0.0,0.0
ballmers,0.0,0.0,0.0,0.0,0.0,0.0,1.94591014906
resigned,0.0,1.94591014906,0.0,0.0,0.0,0.0,0.0
its,0.0,1.94591014906,0.0,0.0,0.0,0.0,0.0
satya,0.0,0.0,0.0,1.94591014906,0.0,0.0,0.0
from,0.0,1.2527629685,0.626381484248,0.0,0.0,0.0,0.0
would,0.0,0.0,0.0,0.0,1.94591014906,0.0,0.0
acquired,0.0,0.0,0.0,0.0,0.0,1.94591014906,0.0
threefold,0.0,0.0,0.0,0.0,0.0,0.0,1.94591014906
board,0.0,1.94591014906,0.0,0.0,0.0,0.0,0.0
largest,1.94591014906,0.0,0.0,0.0,0.0,0.0,0.0
new,0.0,0.0,0.0,1.94591014906,0.0,0.0,0.0
was,0.0,0.0,0.0,1.94591014906,0.0,0.0,0.0
more,0.0,0.0,0.0,0.0,1.94591014906,0.0,0.0
billion,0.0,0.0,0.0,0.0,0.0,1.94591014906,0.0
his,0.0,0.0,0.972955074528,0.0,0.0,0.0,0.0
recently,0.0,0.0,0.0,0.0,0.0,1.94591014906,0.0
increased,0.0,0.0,0.0,0.0,0.0,0.0,1.94591014906
tenure,0.0,0.0,0.0,0.0,0.0,0.0,1.94591014906
ceo,0.0,0.847297860387,0.423648930194,0.847297860387,0.0,0.0,0.0
team,0.0,0.0,0.0,0.0,0.0,1.94591014906,0.0
clippers,0.0,0.0,0.0,0.0,0.0,1.94591014906,0.0
said,0.0,0.0,0.0,0.0,1.94591014906,0.0,0.0
during,0.0,0.0,0.0,0.0,0.0,0.0,1.94591014906
him,0.0,0.0,0.0,0.0,1.94591014906,0.0,0.0
basketball,0.0,0.0,0.0,0.0,0.0,1.94591014906,0.0
he,0.0,0.0,0.0,0.0,0.0,1.94591014906,0.0
a,0.0,0.0,0.0,0.0,0.0,1.94591014906,0.0
worlds,1.94591014906,0.0,0.0,0.0,0.0,0.0,0.0
for,0.0,0.0,0.0,0.0,0.0,1.94591014906,0.0
yearold,0.0,0.0,0.626381484248,1.2527629685,0.0,0.0,0.0
ballmer,0.0,0.847297860387,0.423648930194,0.0,0.847297860387,0.0,0.0
microsoft,1.94591014906,0.0,0.0,0.0,0.0,0.0,0.0
time,0.0,0.0,0.0,0.0,1.94591014906,0.0,0.0
position,0.0,0.0,0.972955074528,0.0,0.0,0.0,0.0
the,0.559615787935,0.0,0.559615787935,0.559615787935,0.0,0.559615787935,0.0
revenues,0.0,0.0,0.0,0.0,0.0,0.0,1.94591014906
maker,1.94591014906,0.0,0.0,0.0,0.0,0.0,0.0
software,1.94591014906,0.0,0.0,0.0,0.0,0.0,0.0
--------------------------------------OUTPUT ENDS---------------------------------------------------------------
```