Lab3 VSM

Due Sep 30, 2018 by 11:59pm **Points** 10 **Submitting** a text entry box or a file upload
File Types txt, py, and csv

Files we'll use:

Lab_3_VSM_Exercise.pdf (https://canvas.ust.hk/courses/19691/files/1636054/download?wrap=1) 
(https://canvas.ust.hk/courses/19691/files/1636054/download?wrap=1)

lab3_skeleton.py (https://canvas.ust.hk/courses/19691/files/1636053/download?wrap=1) 
(https://canvas.ust.hk/courses/19691/files/1636053/download?wrap=1)

You need to submit the program output and python script.

Answer:

lab3_ans.py (https://canvas.ust.hk/courses/19691/files/1718729/download?wrap=1) 
(https://canvas.ust.hk/courses/19691/files/1718729/download?wrap=1)

Oct 2 Update:

We compared performance of sparse and dense matrix representation using following script,
sparse.py (https://canvas.ust.hk/courses/19691/files/1651873/download?wrap=1) 
(https://canvas.ust.hk/courses/19691/files/1651873/download?wrap=1)

Make sure you can understand the code. If you want to run the script, you can use smaller data instead of reuters, as the dense representation could take upto 6Gb RAM . The elapsed time is listed here:

TfIdf mat shape for SparseVec : (10788, 41600)
Load sparse: 50.976316928863525
TfIdf mat shape for DensVec : (10788, 41600)
Load dense: 49.63896584510803
TfIdf mat shape for SKLearnVec : (10788, 41420)
Load sklearn: 17.090452194213867

Profiling for 1 runs
Sparse matrix: 8.935842602979392
Dense matrix: 14.186886073788628
SKLearn matrix: 7.622824081918225