ReadMe file: Data Mining Assignment 1 (KNN Implementation)

Authors: Dhanvi Athmakuri (athmakuri.1), Harsh Gupta (gupta.749)

Date: 09/20/2016

KNN has been implemented in two different ways. Implementation-I uses **sklearn,** whereas Implementation-II implements the following research paper – ‘**Zhang, Min-Ling, and Zhi-Hua Zhou. "ML-KNN: A lazy learning approach to multi-label learning." Pattern recognition 40.7 (2007):\\ 2038-2048**’

1. All the relevant source code is stored in the following directory : **/home/8/athmakuri.1/DMLab2Submission/**
2. All the required data is stored in : **/home/8/athmakuri.1/athmakuri.1\_DM\_Lab\_2/**
3. The code is compatible with python 2

**Running Implementation – I** (**faster code**):

* file containing source code : **knn\_tester.py**
* parent folder : **/home/8/athmakuri.1/DMLab2Submission/**
* command to run the code : **python ./knn\_tester.py <command\_line\_arguments>**
  + **command line arguments** : The program takes a maximum of **seven** command line arguments as follows :
  + **<doc|title|tfidf> <file\_count > <k-value> <-a|-c|-ac> <-s> <test set\_fraction>**
    - **<doc|title|tfidf>** : the type of feature vector for which you want the results. Enter ‘**doc**’, for **bag of words file (raw frequency)**, enter ‘**title**’ for **feature vector with title words alone**, enter ‘**tfidf**’ for **bag of words file (tfidf measure).**
    - **<file size >** : The only possible values you can enter are **1,2,3,22.** Enter 1 if you want to run the code for 500 articles (1 .sgm file), 2 for 1000 articles, 3 for 1500 articles and 22 for all the 10,377 articles (all the .sgm files complete reuters dataset)
    - **<k-value>** : Enter the desired k value
    - **<-a|-c|-ac>** : Choice of performance measure. Enter **-a** for accuracy alone. Enter **-c** for coverage error alone. Enter **-ac** for both the measures. Note that if you enter -ac, it will take twice the time to report the results. If you enter **-c** or **-ac**, the predictions will be written to **predictions\_file.csv** in the folder : **/home/8/athmakuri.1/athmakuri.1\_DM\_Lab\_2**
    - **<-s>** : indicates that user wants to enter the choice of split. The **only** value that can be entered here is **-s**
    - **<test\_fraction**> : After entering -s, enter your desired fraction of testing dataset.
* **Example** : The following command :
* **python ./knn\_tester.py doc 3 5 -a -s 0.3**, would run the code on the bag of words vector matrix, on **3** .sgm files(1500 articles), with a k-value of **5**. The code would report only the accuracy (-a) and would use 70% of the data for training and 30% for testing. Note that you have to **‘-s’** in that position before you enter the desired test set fraction.
* If you do not wish to enter all the options, just enter **python ./knn\_tester.py** . It will run the code for all the 22 .sgm (complete reuters dataset) files for bag of words vector matrix, with a value of k, reporting only the accuracy and using a split of 70% - 30%. Estimated time would be 780 seconds.
* **If you choose to give command line arguments, please see to it that they are in the proper order, lest it should cause run time error, as erroneous command line input is not checked.**

**Running Implementation-II (Slower code. Run only on small datasets (<file size> = 1,2,3))**

* file containing source code : **ml\_knn.py, offline\_processor.py**
* parent folder : **/home/8/athmakuri.1/DMLab2Submission/**
* **offline\_processor.py :** Computes the k nearest neighbours of all the training instances, so that ml\_knn.py can compute the posterior probabilities.
* **ml\_knn.py :** runs the actual knn algorithm as presented in the above research paper.

The offline\_processor.py creates the **training\_data\_file.csv, testing\_data\_file.csv and neighbours.csv** files which are used by **ml\_knn.py** for further computation. The offline\_processor.py takes a lot of time to run, there fore there are pre computed files that knn.py can use to predict the testing instances.

**Running ml\_knn.py :**

* command : **python ./ml\_knn.py <file size>**
  + <file size> is a command line argument, which can only be **1** or **2** or **‘-n**’. Enter 1 if you want to run the algorithm for 1 .sgm file (500 articles) and 2 for two .sgm files (1000 articles).
  + **-n** indicates that you have run offline\_processor.py again and hence ml\_knn.py should use the new files generated by offline\_processor.py and not the pre computed files. Enter -n **only** if you run offline\_processor.py
  + If you do not give any command line arguments, <file\_size> would be taken to be 1.
  + All the predictions, by knn.py will be stored in **ml\_knn\_predictions.csv** file, in the folder : **/home/8/athmakuri.1/athmakuri.1\_DM\_Lab\_2**
  + Note that **ml\_knn.py** is a slow algorithm. It takes atleast over 200 seconds for the smallest dataset.

**Running offline\_processor.py :**

* Command : **python ./offline\_processor.py <doc|title|tfidf> <file size> <test\_set\_fraction>**. Here the command line arguments mean the same as in the previous cases. Note that <file size> can only be 1,2,3 or 22.
* If you do not give any command line arguments, bag of words file would be used, with a file size of 1 and a test set fraction of 0.3.