==========
===README==========

Step 1 : Download Tweets using Twitter API

bash step1_DownloadTweets.sh $OUPUTFILENAME

$OUTPUTFILENAME : Name of the Output JSON file where the tweets need to be downloaded using twitter API

Example Usage :

bash step1_DownloadTweets.sh 'output.json'

NOTE : The twitter API gets the stream of tweets with particular keywords specified in the keyword files of each data directory
Update 'data/keywords/en-keywords.txt'
'data/keywords/es-keywords.txt'
'data/keywords/ru-keywords.txt'
'data/keywords/fa-keywords.txt'
for adding or removing keyword search terms

Step 2 : Parse the JSON File and extracts the metadata about each tweets

bash step1_ParseJSONTweets.py $inputJSONFile $outputParseFile

Gets the downloaded JSON as input and parses metadata about each tweet

Example Usage :

bash step1_ParseJSONTweets.sh 'output.json' 'outputjsonparse.txt'

Step 3 : Filter tweets and make it ready for feature Extraction

bash step2_FilterTweets.sh $language_option $inputFileStep1 $test_train_cutoffrange $data_path

$language_option - 'en' for english, 'es' for spanish, 'ru' for russian and 'fa' for farsi
$inputFileStep1 - output tweetmetadata metadata file from the previous step
$test_train_cutoffrange - integer value which divides train and test set. First $test_train_cutoffrange tweets are going to be training set. The rest are going to be test set.
$data_path - path to the directory structure under which all the data for this particular experiment is going to reside

Example Usage:

bash step2_FilterTweets.sh 'en' 'outputjsonparse.txt' '10000' 'data/'

specifies first 10000 tweets to be in train set and the rest to be in test

bash step2_FilterTweets.sh 'en' 'outputjsonparse.txt' '0' 'data/'

All files are going to be a single set. Doesnt split them into train and test sets.

Step 4 : Extract Features from the dataset :

bash step3_FindFeatureValues.sh $test_or_train_option $inputfilename $tweetCutOffRange $language_option $data_path

$test_or_train_option - '-t' for train set or '-T' for test set
$inputfilename - '-x' uses outputfilename from previous script.
- any other file whose features need to be extracted
NOTE : Train should always be run before the test file in order to generate features.
If you use same data folder for different train and test pairs, it is mandatory to run training set before test set
$tweetCutOffRange - We may sometime need specific number of tweets for each class. Say uniform distribution of 5000 tweets for each class. This argument specifies that range of tweets.
language_option - 'en','es','ru','fa'
data_path - path to the directory where the data files for this particular experiment resides

Example Usage:

bash step3_FindFeatureValues.sh '-t' '-x' 3000 'es' 'data1/'
Specifies Train set Feature Extraction which uses previous output file

bash step3_FindFeatureValues.sh '-T' '/data/input_step5-T.txt' 3000 'es' 'data1/'
Specifies Test set Feature Extraction with a user selected file

bash step3_FindFeatureValues.sh '-t' '-x' 0 'es' 'data1/'
Specifies Train Set feature extraction selecting all the tweets for training (number of samples in each class is not specified)

NOTE : When the argument is '-t' certain features are generated.
But when the argument is '-T' it would just use the features but it would not generate those features.
So it is always better to run the train and test back to back until unless there is a data folder for each experiment

Step 5: Run SVM and generate models :

bash step4_RunSVM.sh $test_or_train_option $class_option $language_option $data_path

$test_or_train_option - '-t' for test, '-T' for train
$class_option - 1 to 8 for running binary classifier against that class and rest
- -x for running default multiclass classifier

$language_option - 'en','es'
data_path - directory path where the data folder of this experiment resides
- -x for default data path 'data/'

Example Usage :

bash step4_RunSVM.sh '-t' '1' 'es' 'data1/'

Happy vs Rest Binary classifier Training for Spanish language with the data_path as 'data1/' folder

bash step4_RunSVM.sh '-T' '-x' 'en' '-x'

Multiclass classification Testing for English Language tweets with default data path 'data/'s

NOTE :

Model will be generated only when running with '-t' option which should be used immediately followed by '-T' option for testing

Feature selection should be run before this process so that required features would have been calculated and SVM input could be prepared

mergeFeatures.py file contain some arguments which needs to be modified to choose appropriate features to be combined