**import** os

**import** csv

**import** re

**import** sample\_package.commonFunction **as** basefunc

**import** sample\_package.filterMethod **as** filter\_TFID

**from** nltk.corpus **import** stopwords

**import** re

**from** datetime **import** datetime

**from** nltk.sentiment.vader **import** SentimentIntensityAnalyzer

**from** numpy **import** array

**import** numpy **as** np

**from** scipy.misc **import** toimage

sid **=** SentimentIntensityAnalyzer()

foundCnt **=** 0;

foundMax **=** 1000;

tweetIn**=**[];

labelIn**=**[];

nrow**=**'';

text **=** []

target **=** []

*# get Game Infomation*

print("--- programming start---")

print("Collecting tweets...")

*#let's try team "Thunder" far example*

*# if you want to try multiple teams, append team names into list.*

teamNameList **=** ["Thunder"] *# , "Blazers","Grizzlies"*

**for** teamName **in** teamNameList:

gameInfo **=** basefunc**.**getGameInfo(teamName);

**for** eachGameInfo **in** gameInfo:

*# Game data and time*

gameDate **=** eachGameInfo**.**Date

gameTime **=** eachGameInfo**.**Time

*# print("Team: {} Date: {} {}".format(teamName, gameDate, eachGameInfo.result))*

fileName **=** basefunc**.**getCvsPathByGameData(teamName, gameDate)

foundCnt**+=**1

**if** (len(fileName) **==** 0):

**continue**;

fileName **=** ''**.**join(fileName)

nrow**=**'';

file**=**open(fileName, 'r', encoding**=**'utf-8')

csvCursor**=**csv**.**reader(file)

**for** idx, row **in** enumerate(csvCursor):

**if** idx**>**0:

*#For here,if you want to run with sentiment, please comment block 1. and open block 2.*

*#-------------block 1. without sentiment-------------#*

*# remove http link*

tpt **=** re**.**sub(r'(https|http)?:\/\/(\w|\.|\/|\?|\=|\&|\%)\*\b', '', row[0], flags**=**re**.**MULTILINE)

*# remove team's name*

tpt **=** tpt**.**replace(teamName, '')

nrow**+=**tpt

*#------------- end of block 1. -----------#*

*#-------------block 2. sentiment-------------*

*# remove*

*# tpt = re.sub(r'(https|http)?:\/\/(\w|\.|\/|\?|\=|\&|\%)\*\b', '', row[0], flags=re.MULTILINE)*

*# tpresult = sid.polarity\_scores(tpt)*

*# if(tpresult["compound"] != 0):*

*# tpt = tpt.replace(teamName, '')*

*# nrow += tpt*

*#------------- end of block 2. ------------#*

tweetIn**.**append(nrow)

**if**(eachGameInfo**.**result**==**'W'):

labelIn**.**append(eachGameInfo**.**result)

**elif**(eachGameInfo**.**result**==**'L'):

labelIn**.**append(eachGameInfo**.**result)

**if** foundCnt **>=** foundMax:

**break**;

print("--- programming end---")

Check for this output - but ignore the same path, the path and keep your path

--- programming start---

Collecting tweets...

Get result from /Users/cosoet/SIT/BIA660-WebAnalytics/FinalProject/PythonCode/GameResult/Thunder.csv

--- programming end---

**from** sklearn.cross\_validation **import** train\_test\_split

x\_train, x\_test, y\_train, y\_test **=** train\_test\_split(\

tweetIn, labelIn, test\_size**=**0.1, random\_state**=None**)

Check for this output - but ignore the same path, the path and keep your path

/Users/cosoet/anaconda3/lib/python3.6/site-packages/sklearn/cross\_validation.py:41: DeprecationWarning: This module was deprecated in version 0.18 in favor of the model\_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are different from that of this module. This module will be removed in 0.20.

"This module will be removed in 0.20.", DeprecationWarning)

*#for here, we are running Naive Bayes now.*

**from** sklearn.pipeline **import** Pipeline

**from** sklearn.naive\_bayes **import** MultinomialNB

**from** sklearn.feature\_extraction.text **import** TfidfVectorizer

**from** sklearn **import** metrics

*#If you want to run Naive Bayes, please open block NV and comment block SVM.*

*#---------- NV --------------#*

*#running Naive Bayes*

pipeline **=** Pipeline([

('vect', TfidfVectorizer(stop\_words **=** 'english', min\_df**=**0.05, max\_df**=**0.75)),

('clf', MultinomialNB()),

])

labels**=**sorted(list(set(labelIn)))

pipeline**.**fit(x\_train, y\_train)

y\_predicted **=** pipeline**.**predict(x\_test)

print("Running Naive Bayes: \n")

print(metrics**.**classification\_report(y\_test, y\_predicted, target\_names **=** labels))

*#----------end of NV --------------#*

Output

Running Naive Bayes:

precision recall f1-score support

L 0.00 0.00 0.00 4

W 0.50 0.80 0.62 5

avg / total 0.28 0.44 0.34 9

*#* Running SVM Algorithm in Python

*#---------- SVM --------------#*

**from** sklearn.multiclass **import** OneVsRestClassifier

**from** sklearn.svm **import** LinearSVC

pipeline **=** Pipeline([

('vect', TfidfVectorizer(stop\_words **=** 'english', min\_df**=**0.05, max\_df**=**0.75)),

('clf', OneVsRestClassifier(LinearSVC())),

])

labels**=**sorted(list(set(labelIn)))

pipeline**.**fit(x\_train, y\_train)

y\_predicted **=** pipeline**.**predict(x\_test)

print("Running SVM: \n")

print(metrics**.**classification\_report(y\_test, y\_predicted, target\_names **=** labels))

*#----------end of SVM --------#*

Running SVM:

precision recall f1-score support

L 0.00 0.00 0.00 4

W 0.50 0.80 0.62 5

avg / total 0.28 0.44 0.34 9