# Department of Computing

# CS471: Machine Learning

# Class: BSCS-6C

# Lab 10: Sentiment Analysis (Part 1)

# Date: 5 Mar 2019

# Time: 10:00-1:00

# Instructor: Dr. Omar Arif

# Lab 10: Sentiment Analysis (Part 1)

**Name:** Ahmad Amjad Mughal

**Reg No:**121672

**Class:** BSCS-6C

**Lab Tasks**

Download labeledTrainData.tsv, which contains 25000 IMDB movie reviews.

Use Pandas <http://pandas.pydata.org> python package to read this file. Pandas package is preinstalled in your canopy python distribution.

import pandas as pd

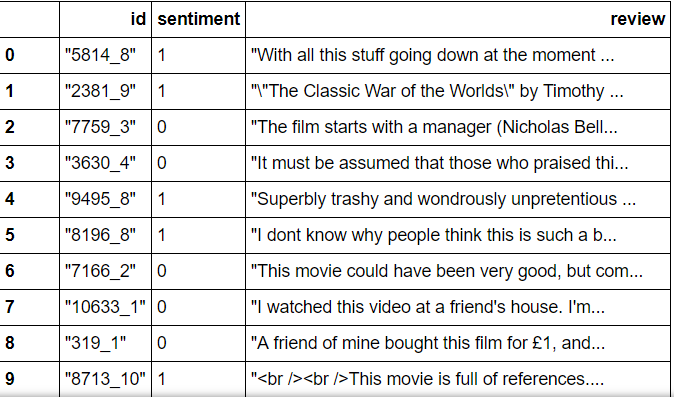
train = pd.read\_csv('labeledTrainData', header=0, delimiter='\t', quoting=3)

**Task 1:**

**How is the data stored in the variable ‘train’?**

Ans. The variable train is a 2d array and data is stored in it. Each specific row represents id, sentiment and reviews of each movie.

**Screenshot**



**What is the shape of the variable ‘train’?**

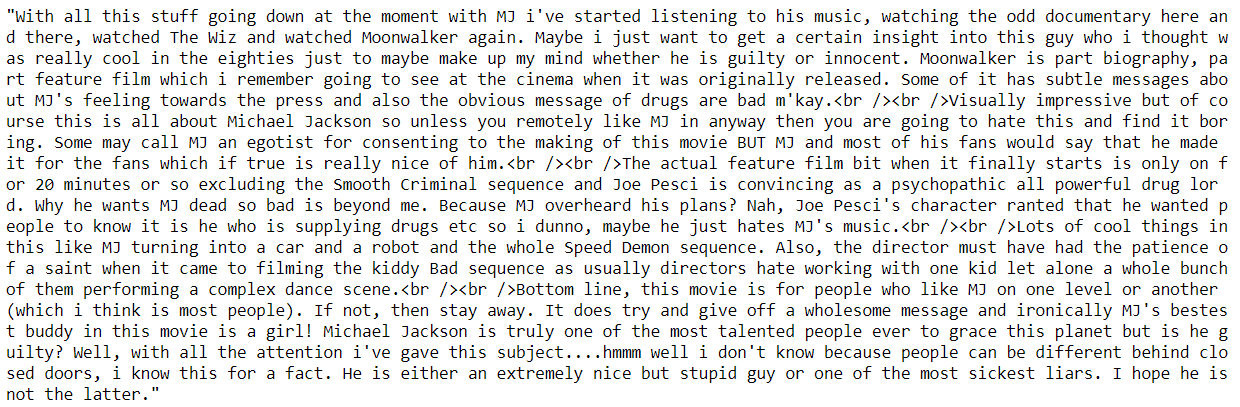
Ans. Train’s shape is in tabular form. Giving it the Serial number, id of the review, Sentiments that can be either positive or negative and the reviews itself.

**How do you read the first few reviews from the variable ‘train’?**

**Ans.** i = 0

for i in range(3):

print train["review"][i]



There are HTML tags in the review. HTML tags won’t help us in sentiment analysis. So we remove them. We will use Beautiful Soup <http://www.crummy.com/software/BeautifulSoup/bs4/doc/> package to do that. First install it in your canopy distribution using the following command

pip install BeautifulSoup4

Remember to restart the kernel after the installation. Now execute the following to remove HTML tags from the training reviews:

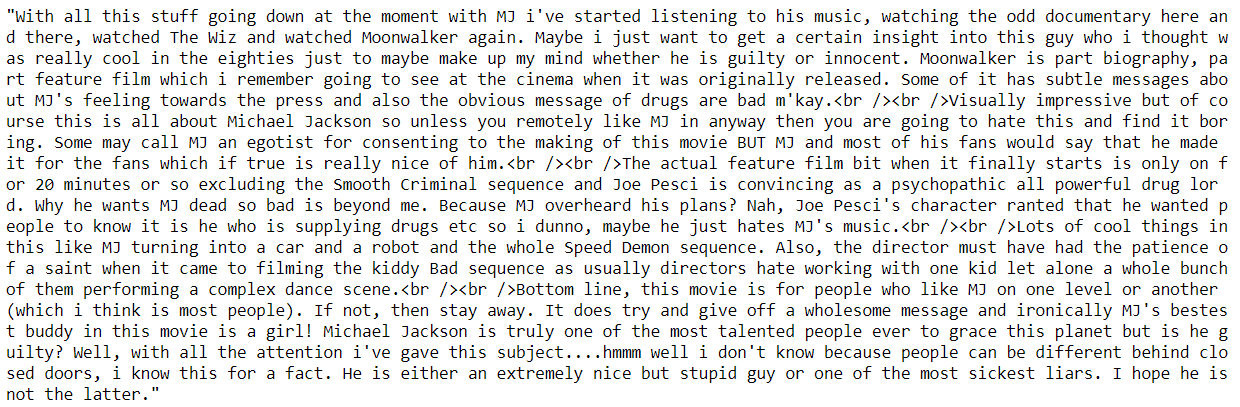
from bs4 import BeautifulSoup

# Run the BeautifulSoup object on a single movie review

example1 = BeautifulSoup(train["review"][0])

print example1.get\_text()

**Screenshot**

****

Punctuation and numbers also don’t help in deciding the sentiment of a review. We will remove them using the package **re** (regular expression). **Re** is a built in python package. See the package documentation to complete the next task.

**Task 2:**

Use re package to find every thing that is not a lowercase letter or upper case letter and replace it with a space for each review in the training data.

For example the following code finds the alphabet a and v and replaces it with b.

**Code**

import re

i = 0

for I in range(2500):

example = BeautifulSoup(train[‘’review’’][i])

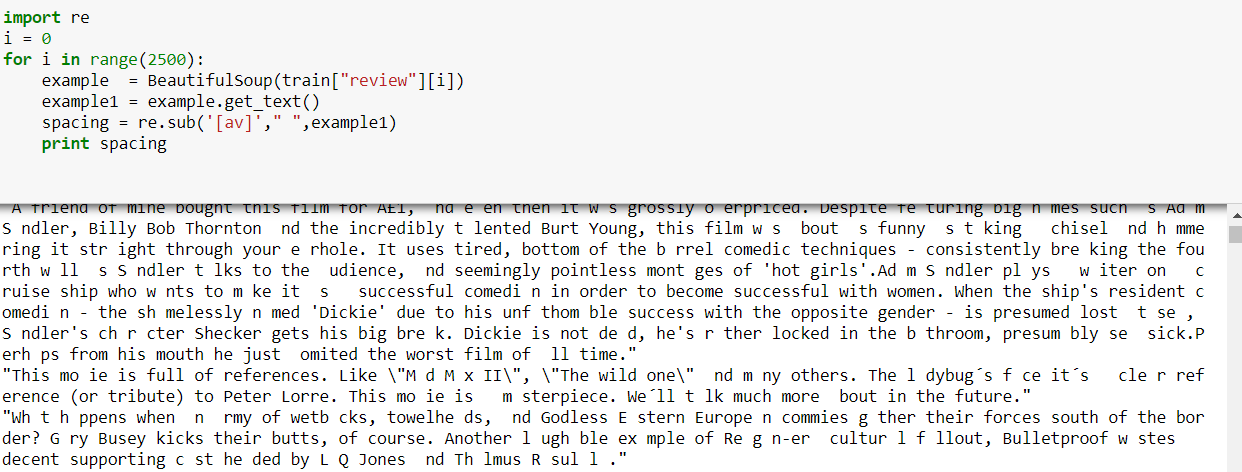
example1 = example.get\_text()

spacing = re.sub('[av]'," ",exampl1)

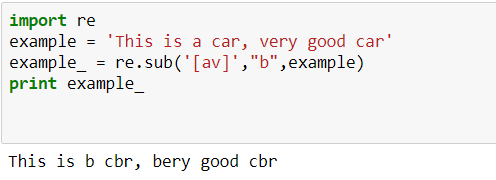
print spacing

**For all trained reviews:**

**Screenshot**

****

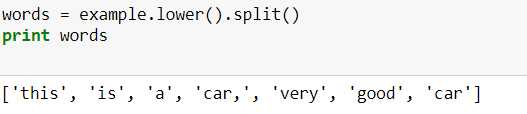
**Screenshot**

****

**TOKENIZATION:** We will also convert every thing into lower case and split the reviews into individual words using following commands

words = example.lower().split()

**Screenshot**



Finally, we need to decide how to deal with frequently occurring words that don't carry much meaning. Such words are called stop words; in English they include words such as "a", "and", "is", and "the". We will use Natural Language Toolkit (nltk) <http://www.nltk.org> package for this purpose. First install the package and download the stop word list as follows.

pip install nltk

Now execute following in the shell

import nltk

nltk.download(‘stopwords’)

from nltk import stopwords

print stopwords.words(‘english’)

Now remove the stop words from all the reviews. The following will remove the stop words from the variable ‘words’. Remember ‘words’ contains tokenized review. **Understand the syntax below (how for loop is used.)**

**Code**

import nltk

nltk.download('words')

from nltk.corpus import words

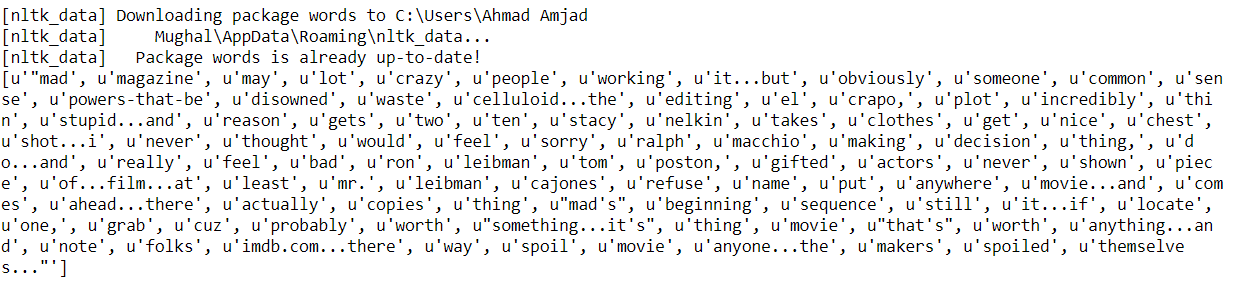
#print stopwords.words('english')

stops = set(stopwords.words('english'))

myword = [w for w in words if not w in stops]

print myword

**Screenshot**

****

**Task 4**

Multiply each element of the list A = {2, 3, 4, 5, 7, 8, 9, 2, 5} with 5 using the for loop syntax above.

Multiply each element of the list A = {2, 3, 4, 5, 7, 8, 9, 2, 5}, except 2, with 5 using the for loop syntax above.

**Code**

listA = {2, 3, 4, 5, 7, 8, 9, 2, 5}

listB = {2, 3, 4, 5, 7, 8, 9, 2, 5}

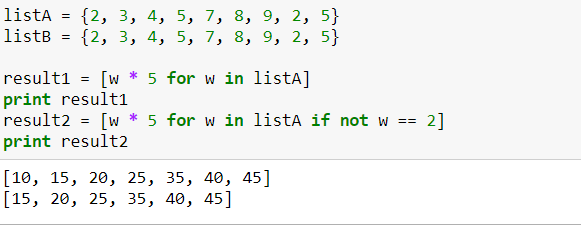
result1 = [w \* 5 for w in listA]

print result1

result2 = [w \* 5 for w in listA if not w == 2]

print result2

**Screenshot**



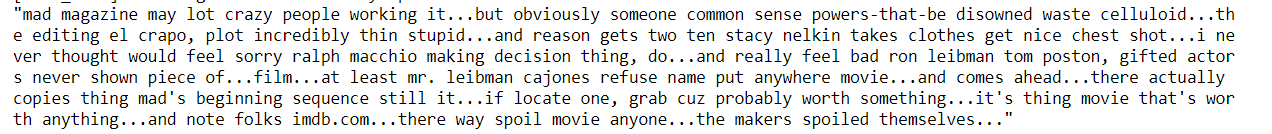
Now join the words back into one string separated by space.

**Code**

sentence = “ ”.join(words);

print sentence

**Screenshot**

****

**Task 5**

**How can you join the words back into one string separated by colon (:)?**

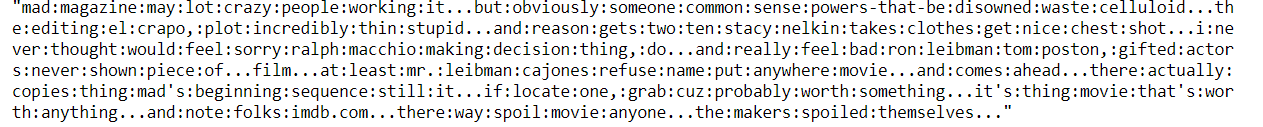
**Ans** Simply by replacing spacing with colon. Rest of the statement remains same

**code**

sentence = “:”.join(words);

print sentence

**Screenshot**



**Task 6**

You have learned how to take a review, remove HTML tags, remove punctuations, convert it to lower case, split it into words, remove stop words and finally join the words back separated by space. Write a function that combines all these steps so that you can reuse that for all the reviews.

def review\_to\_words(raw\_review)

#1. Remove HTML

#2. Remove non letters

#3. Convert to lowercase and split it into words

#4. Remove stops words

#5. Joint back and return the joined sentence

**Code**

def review\_to\_words(review):

htmlremoval = BeautifulSoup(review)

example = htmlremoval.get\_text()

example1 = re.sub(r'[^a-zA-Z]'," ",example)

splitter = example1.lower().split()

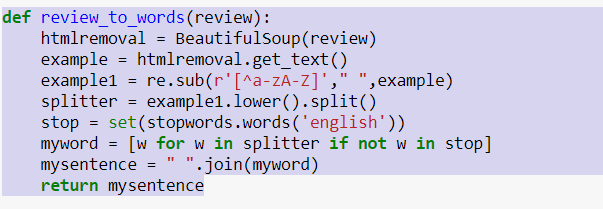
stop = set(stopwords.words('english'))

myword = [w for w in splitter if not w in stop]

mysentence = " ".join(myword)

return mysentence

Screenshot



**Task 7**

Run the above function for each review in your training data and store the output in one list.

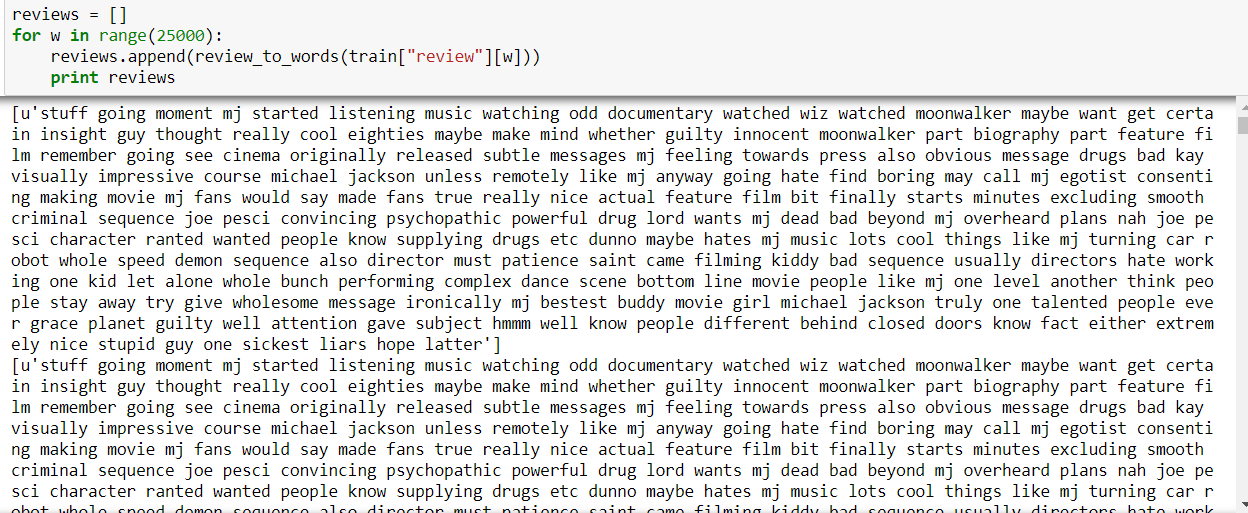
**Code**

reviews = []

for w in range(25000):

reviews.append(review\_to\_words(train["review"][w]))

print reviews

**Screenshot**