# Department of Computing

**CS471: Machine Learning**

**Class: BESE-7AB**

# Lab 06: Sentiment Analysis (Part 1)

**CL03: Apply a variety of learning algorithms to data for solution development**

**Date: 8/3/2019**

**Time: 10:00 am– 1:00 pm & 2:00 pm-5:00 pm**

**Instructor: Dr. Pakeeza Akram**

**Submitted by: Syed Waleed Hyder (189212)**

# Lab 05: Sentiment Analysis (Part 1)

**Introduction**

Sentiment Analysis, also known as opinion mining refers to the use of natural language processing, text analysis to identify and extract subjective information in source materials. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. (Wikipedia)

**Objective**

In this lab you will use IMDB database that contains 25000 movie reviews. Each movie review is labeled with a positive or a negative sentiment. This is your training dataset. You will use IMDB dataset to train a classifier. The trained classifier when presented with a new review will predict if the review is positive or negative.

In this lab you will clean the dataset so that you can use it for training purposes. In the next lab you will use the clean dataset for training.

**Tools/Software Requirement**

Python, pandas, re, BeautifulSoup4, nltk

**Lab Tasks**

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

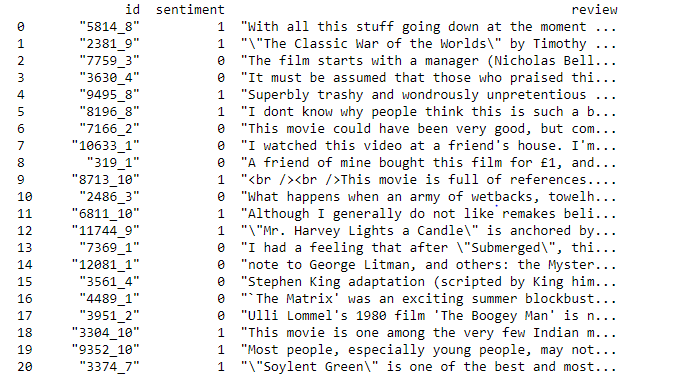
Use Pandas <http://pandas.pydata.org> python package to read this file. Pandas package is preinstalled in your canopy python distribution.

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’?



What is the shape of the variable ‘train’?



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



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()

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.

import re

example = ‘This is a car, very good car'

example\_ = re.sub('[av]',"b",example)

print example\_

#TASK1 and 2

example = "This is a car, very good car"

example\_ = re.sub('[av]',"b",example)

example = re.sub('[^A-Za-z]+',' ', example)

print(example\_)

words = example.lower().split()

# print (stopwords.words('english'))

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

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

print (words)

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

words = example.lower().split()

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.)**

stops = set(stopwords.words(‘english’))

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

print words

#TASK2

example = "This is a car, very good car"

example\_ = re.sub('[av]',"b",example)

example = re.sub('[^A-Za-z]+',' ', example)

print(example\_)

words = example.lower().split()

# print (stopwords.words('english'))

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

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

print (words)

**Task 3**

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.

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

#TASK3

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

A = {a\*5 for a in A}

print(A)

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

A = {a\*5 for a in A if a != 2}

print(A)

sentence = " ".join(words)

print(sentence)

;

**Task 4**

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

#TASK4

sentence = ":".join(words)

print(sentence)

**Task 5**

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

# TASK5

def review\_to\_words(raw\_review):

#1. Remove HTML

review\_bs\_obj = BeautifulSoup(raw\_review)

review = review\_bs\_obj.get\_text()

#2. Remove non letters

review = re.sub('[^A-Za-z]+',' ', review)

#3. Convert to lowercase and split it into words

review\_words = review.lower().split()

#4. Remove stops words

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

review\_words = [w for w in review\_words if not w in stops]

#5. Joint back and return the joined sentence

review\_sentence = " ".join(review\_words)

return review\_sentence

**Task 6**

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

In the next lab you will use this list to create a Bag of Words representation and machine learning for sentiment analysis.

for row in range(len(train)):

train.at[row, 'review'] = review\_to\_words(train.at[row, 'review'])

if(row%5 == 0):

print(row)

**Deliverables**

Upload Word file containing all the tasks.