**LAB 3 SENTIMENTAL ANALYSIS**

https://raw.githubusercontent.com/dD2405/Twitter\_Sentiment\_Analysis/master/train.csv

import re

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import string

import nltk

import warnings

warnings.filterwarnings("ignore", category=DeprecationWarning)

%matplotlib inline

**Reading training data**

train = pd.read\_csv('https://raw.githubusercontent.com/dD2405/Twitter\_Sentiment\_Analysis/master/train.csv')

train\_original=train.copy()

**Read Testing Data**

test = pd.read\_csv('https://raw.githubusercontent.com/dD2405/Twitter\_Sentiment\_Analysis/master/test.csv')

test\_original=test.copy()

**Data Preprocessing**

**Combine train and test data**

combine = train.append(test,ignore\_index=True,sort=True)

**Data Exploration**

combine.describe()

combine.head()

combine.tail()

**Removing Twitter Handles**

def remove\_pattern(text,pattern):

    r = re.findall(pattern,text)

    for i in r:

        text = re.sub(i,"",text)

    return text

combine['Tidy\_Tweets'] = np.vectorize(remove\_pattern)(combine['tweet'], "@[\w]\*")

combine.head()

**Removing punctuation , special characters,numbers**

combine['Tidy\_Tweets'] = combine['Tidy\_Tweets'].str.replace("[^a-zA-Z#]", " ")

combine.head(10)

**Removing Short words**

combine['Tidy\_Tweets'] = combine['Tidy\_Tweets'].apply(lambda x: ' '.join([w for w in x.split() if len(w)>3]))

combine.head(10)

**Tokenization**

tokenized\_tweet = combine['Tidy\_Tweets'].apply(lambda x: x.split())

tokenized\_tweet.head()

**Stemming**

from nltk import PorterStemmer

ps = PorterStemmer()

tokenized\_tweet = tokenized\_tweet.apply(lambda x: [ps.stem(i) for i in x])

tokenized\_tweet.head()

for i in range(len(tokenized\_tweet)):

    tokenized\_tweet[i] = ' '.join(tokenized\_tweet[i])

combine['Tidy\_Tweets'] = tokenized\_tweet

combine.head()

**Visualizing Data**

**Word Cloud**

from wordcloud import WordCloud,ImageColorGenerator

from PIL import Image

import urllib

import requests

**Word Cloud for tweets with label 0**

all\_words\_positive = ' '.join(text for text in combine['Tidy\_Tweets'][combine['label']==0])

Mask = np.array(Image.open(requests.get('http://clipart-library.com/image\_gallery2/Twitter-PNG-Image.png', stream=True).raw))

image\_colors = ImageColorGenerator(Mask)

wc = WordCloud(background\_color='black', height=1500, width=4000,mask=Mask).generate(all\_words\_positive)

plt.figure(figsize=(10,20))

plt.imshow(wc.recolor(color\_func=image\_colors),interpolation="hamming")

plt.axis('off')

plt.show()

**Word Cloud for tweets with label 0**

all\_words\_negative = ' '.join(text for text in combine['Tidy\_Tweets'][combine['label']==1])

Mask = np.array(Image.open(requests.get('http://clipart-library.com/image\_gallery2/Twitter-PNG-Image.png', stream=True).raw))

image\_colors = ImageColorGenerator(Mask)

wc = WordCloud(background\_color='black', height=1500, width=4000,mask=Mask).generate(all\_words\_negative)

plt.figure(figsize=(10,20))

plt.imshow(wc.recolor(color\_func=image\_colors),interpolation="gaussian")

plt.axis('off')

plt.show()

**UnderStanding the impact of hashtags**

**Extracting hashtags**

def Hashtags\_Extract(x):

    hashtags=[]

    for i in x:

        ht = re.findall(r'#(\w+)',i)

        hashtags.append(ht)

    return hashtags

**Nested list of hashtags from positive review**

ht\_positive = Hashtags\_Extract(combine['Tidy\_Tweets'][combine['label']==0])

ht\_positive

ht\_positive\_unnest = sum(ht\_positive,[])

ht\_positive\_unnest

**Nested list of hashtags from positive review**

ht\_negative = Hashtags\_Extract(combine['Tidy\_Tweets'][combine['label']==1])

ht\_negative

ht\_negative\_unnest = sum(ht\_negative,[])

ht\_negative\_unnest

**Bar graphs for positive tweets**

word\_freq\_positive = nltk.FreqDist(ht\_positive\_unnest)

word\_freq\_positive

df\_positive = pd.DataFrame({'Hashtags':list(word\_freq\_positive.keys()),'Count':list(word\_freq\_positive.values())})

df\_positive.head(10)

df\_positive\_plot = df\_positive.nlargest(20,columns='Count')

sns.barplot(data=df\_positive\_plot,y='Hashtags',x='Count')

sns.despine()

**Bar graphs for positive tweets**

word\_freq\_negative = nltk.FreqDist(ht\_negative\_unnest)

word\_freq\_negative

df\_negative = pd.DataFrame({'Hashtags':list(word\_freq\_negative.keys()),'Count':list(word\_freq\_negative.values())})

df\_negative.head(10)

df\_negative\_plot = df\_negative.nlargest(20,columns='Count')

sns.barplot(data=df\_negative\_plot,y='Hashtags',x='Count')

sns.despine()