Twitter_Sentiment_Analysis

Cyber bullying and hate speech has been a menace for quite a long time,So our objective for this task is to detect tweets associated with negative sentiments. From this dataset we classify a tweet as hate speech if it has racist or sexist tweets associated with it.

So our task here is to classify racist and sexist tweets from other tweets and filter them out. With the given twitter dataset consisting of **train.csv** and test.csv files where we have 31962 labeled tweets and 17191 unlabeled tweets where we train and validate on the train.csv file and then test our best possible model on the test.csv file.

The image file is used for super-imposing the twitter logo on the generated wordcloud.

Problem Statement

Dataset containing several tweets with positive and negative sentiment associated with it

- Cyber bullying and hate speech has been a menace for quite a long time,So our objective for this task is to detect speeches tweets associated with negative sentiments.From this dataset we classify a tweet as hate speech if it has racist or sexist tweets associated with it.
- So our task here is to classify racist and sexist tweets from other tweets and filter them out.

Dataset Description

- The data is in csv format.In computing, a comma-separated values (CSV) file stores tabular data (numbers and text) in plain text.Each line of the file is a data record. Each record consists of one or more fields, separated by commas.
- Formally, given a training sample of tweets and labels, where label ‘1’ denotes the tweet is racist/sexist and label ‘0’ denotes the tweet is not racist/sexist,our objective is to predict the labels on the given test dataset.

Attribute Information

- id : The id associated with the tweets in the given dataset
- tweets : The tweets collected from various sources and having either postive or negative sentiments associated with it
- label : A tweet with label '0' is of positive sentiment while a tweet with label '1' is of negative sentiment

PART-1

#Importing the necessary packages

```
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
```

#Train dataset used for our analysis

```
train = pd.read_csv('train.csv')
```

#We make a copy of training data so that even if we have to make any changes in this #dataset we would not lose the original dataset.

```
train_original=train.copy()

#Here we see that there are a total of 31692 tweets in the training dataset

print(train.shape)

#(31962, 3)

print(train_original)
```

#Test dataset used for our analysis

```
test = pd.read_csv('test.csv')
```

#We make a copy of test data so that even if we have to make any changes in this dataset #we would not lose the original dataset.

```
test_original=test.copy()
```

#Here we see that there are a total of 17197 tweets in the test dataset

```
print(test.shape)

#(17197, 2)

print(test_original)
combine = pd.concat([train_original, test_original])
print(combine.head())
print(combine.tail())
```

#Removing Twitter Handles (@user)

```
def remove_pattern(text,pattern):
    # re.findall() finds the pattern i.e @user and puts it in a list for further task
    r = re.findall(pattern,text)

    # re.sub() removes @user from the sentences in the dataset
    for i in r:
        text = re.sub(i,"",text)

    return text

combine['Tidy_Tweets'] = np.vectorize(remove_pattern)(combine['tweet'], "@[\w]*")
print(combine.head())
```

#Removing Punctuations, Numbers, and Special Characters

```
combine['Tidy_Tweets'] = combine['Tidy_Tweets'].str.replace("[^a-zA-Z#]", " ")
print(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]))

print(combine.head(10))
```

#Tokenization

```
tokenized_tweet = combine['Tidy_Tweets'].apply(lambda x: x.split())

print(tokenized_tweet.head())
```

#Stemming

```
from nltk import PorterStemmer

ps = PorterStemmer()

tokenized_tweet = tokenized_tweet.apply(lambda x: [ps.stem(i) for i in x])

print(tokenized_tweet.head())
```

#Visualization from Tweets

#A wordcloud is a visualization wherein the most frequent words appear in #large size and the less frequent words appear in smaller sizes

```
from wordcloud import WordCloud,ImageColorGenerator

from PIL import Image

import urllib

import requests
```

#Store all the words from the dataset which are non-racist/sexist

```
all_words_positive = ' '.join(text for text in combine['Tidy_Tweets'][combine['label']==0])
```

 # combining the image with the dataset

```
Mask = np.array(Image.open(requests.get('http://clipart-library.com/image_gallery2/Twitter-PNG-Image.png',
stream=True).raw))
```

 # We use the ImageColorGenerator library from Wordcloud

 # Here we take the color of the image and impose it over our wordcloud

```
image_colors = ImageColorGenerator(Mask)
```

 # Now we use the WordCloud function from the wordcloud library

```
wc = WordCloud(background_color='black', height=1500,
width=4000,mask=Mask).generate(all_words_positive)
```

```
# Size of the image generated
```

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

```
# Here we recolor the words from the dataset to the image's color
```

```
# recolor just recolors the default colors to the image's blue color
```

```
# interpolation is used to smooth the image generated
```

```
plt.imshow(wc.recolor(color_func=image_colors),interpolation="hamming")
```

```
plt.axis('off')
```

```
plt.show()
```