sentiment analysis in twitter

The purpose of this project was to detect whether a tweet contains hate speech or not. This project is implemented in python 2.7 and packages such as pandas, numpy, nltk, and sklearn are used.

- At first we read the training and the test csv files.

```
train  = pd.read_csv('train_E6oV3lV.csv')
test = pd.read_csv('test_tweets_anuFYb8.csv')

print(train.head())
```

```
   id  label                                              tweet
0   1      0   @user when a father is dysfunctional and is s...
1   2      0  @user @user thanks for #lyft credit i can't us...
2   3      0                                bihday your majesty
3   4      0  #model   i love u take with u all the time in ...
4   5      0             factsguide: society now    #motivation
```

- Then we combine the two datasets not to repeat the same steps twice on training test sets.

```
combi = train.append(test, ignore_index=True)
```

- Our dataset of tweets needs preprocessing. Considering that tweets may have handles such as @username we must omit them from the text of the tweets. Also we need to remove short words such as hmm, oh, etc… .

```
def remove_pattern(input_txt, pattern):
    r = re.findall(pattern, input_txt)
    for i in r:
        input_txt = re.sub(i, '', input_txt)
    return input_txt

# remove twitter handles (@user)
combi['tidy_tweet'] = np.vectorize(remove_pattern)
(combi['tweet'], "@[\w]*")

# remove special characters, numbers, punctuations
combi['tidy_tweet'] = combi['tidy_tweet'].str.replace("[^a-zA-
Z#]", " ")
```

```
# removing Short Words
combi['tidy_tweet'] = combi['tidy_tweet'].apply(lambda x: '
'.join([w for w in x.split() if len(w)>3]))
```

- The next step is to tokenize words in each tweet.

```
tokenized_tweet = combi['tidy_tweet'].apply(lambda x: x.split())
```

- We want to have the root word for each word in a tweet so now we use the nltk package to do the stemming.

```
from nltk.stem.porter import *
stemmer = PorterStemmer()
tokenized_tweet = tokenized_tweet.apply(lambda x:
[stemmer.stem(i) for i in x]) # stemming
```

```
0    [when, father, dysfunct, selfish, drag, kid, i...
1    [thank, #lyft, credit, caus, they, offer, whee...
2                                [bihday, your, majesti]
3                      [#model, love, take, with, time]
4                          [factsguid, societi, #motiv]
```

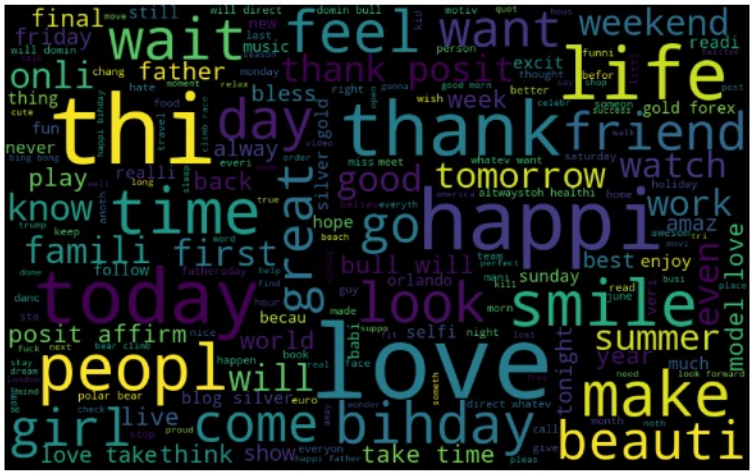
- Now we need to visualize our data. For this matter we use the word cloud method.

```
all_words = ' '.join([text for text in combi['tidy_tweet']])
from wordcloud import WordCloud
wordcloud = WordCloud(width=800, height=500, random_state=21,
max_font_size=110).generate(all_words)
plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.show()
```

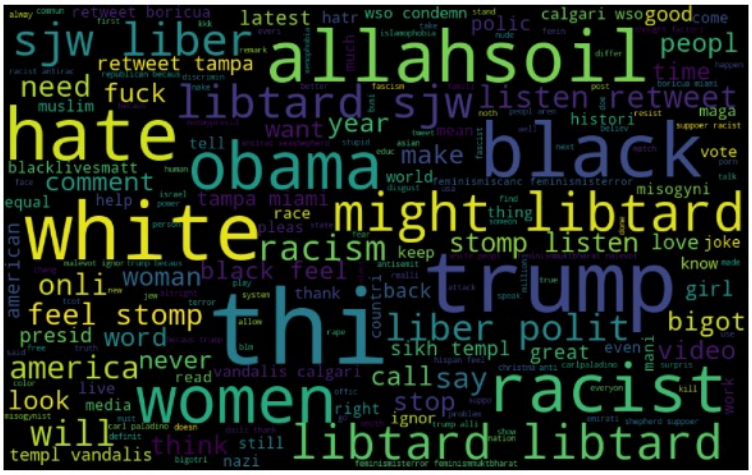
Result:

- We can see most of the words are positive or neutral. With happy and love being the most frequent ones. It doesn't give us any idea about the words associated with the racist/sexist tweets. Hence, we will plot separate word clouds for both the classes(racist/sexist or not) in our train data.

non racist/sexist tweets



racist/sexist tweets



- Hashtags in twitter are synonymous with the ongoing trends on twitter at any particular point in time. We should try to check whether these hashtags add any value to our sentiment analysis task. They help in distinguishing tweets into the different sentiments.

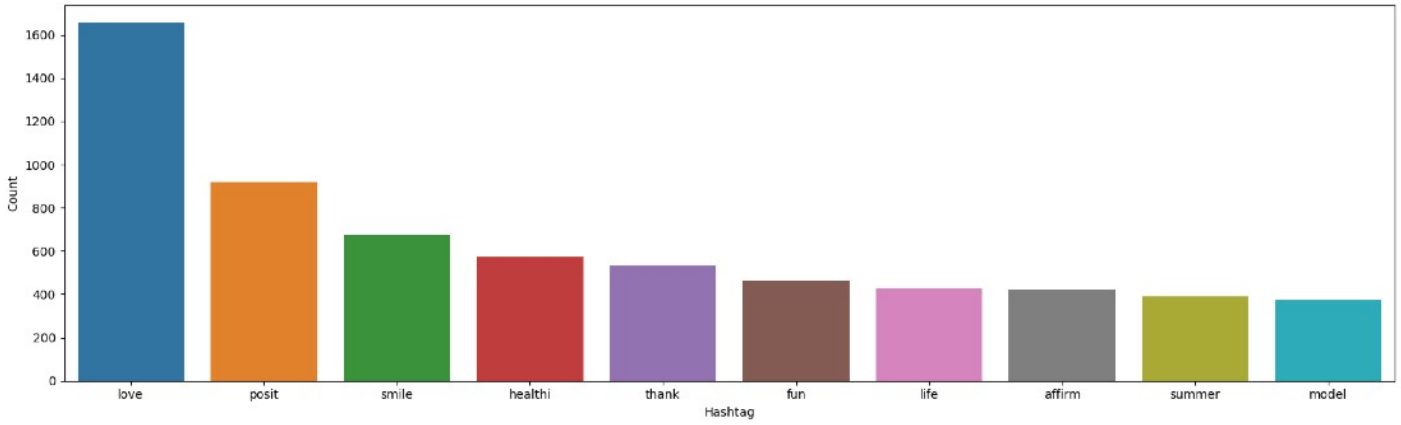
```
def hashtag_extract(x):
    hashtags = []
    # Loop over the words in the tweet
    for i in x:
        ht = re.findall(r"#(\w+)", i)
        hashtags.append(ht)
    return hashtags
```

- Then we extract the hashtags from tweets.

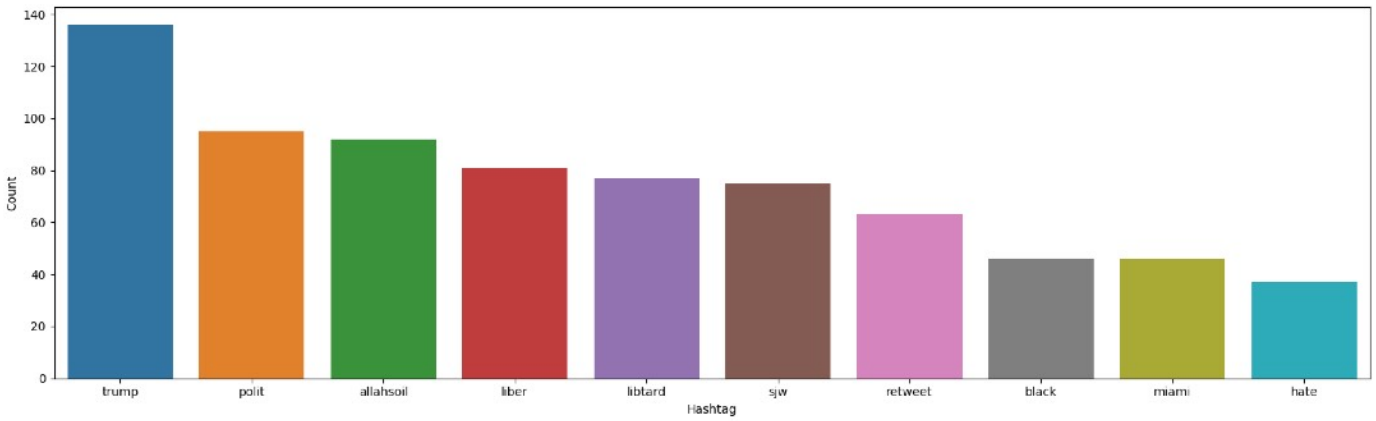
```
# extracting hashtags from non racist/sexist tweets
HT_regular = hashtag_extract(combi['tidy_tweet'][combi['label'] 
== 0])

# extracting hashtags from racist/sexist tweets
HT_negative = hashtag_extract(combi['tidy_tweet'][combi['label'] 
== 1])

# unnesting list
HT_regular = sum(HT_regular,[])
HT_negative = sum(HT_negative,[])
```

- Now we draw two plot for non-racist/sexist tweets hashtags and racist/sexist tweets hashtags.

non racist/sexist tweets



racist/sexist tweets



- We implemented two models for extracting the features of tweets. First one is called “Bag of Words”. We use sklearn package here and we set the max-feature parameter to 1000.

```
from sklearn.feature_extraction.text import CountVectorizer
bow_vectorizer = CountVectorizer(max_df=0.90, min_df=2,
max_features=1000, stop_words='english')
# bag-of-words feature matrix
bow = bow_vectorizer.fit_transform(combi[‘tidy_tweet'])
```

The accuracy that we attained from this model is: 0.5307820299500832

- The second model id TF-IDF, which is based on the frequency method but it is different to the bag-of-words approach in the sense that it takes into account, not just the occurrence of a word in a single document (or tweet) but in the entire corpus.

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(max_df=0.90, min_df=2,
max_features=1000, stop_words='english')
# TF-IDF feature matrix
tfidf = tfidf_vectorizer.fit_transform(combi['tidy_tweet'])
```

The accuracy that we attained from this model is: 0.5446507515473032

Author: Saeideh Sadeghpour Gildeh