In [ ]:

```

```

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import nltk
from nltk.tokenize import TweetTokenizer
from nltk.stem.porter import PorterStemmer
import warnings
%matplotlib inline
warnings.filterwarnings('ignore')
```

In [2]:

```
df = pd.read_csv("D:/New folder/train_E6oV3lV.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	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

In [4]:
```
df.tail()
```

Out[4]:

	id	label	tweet
31957	31958	0	ate @user isz that youuu?ð□□□ð□□□ð□□□ð□□□ð□□□ð...
31958	31959	0	to see nina turner on the airwaves trying to...
31959	31960	0	listening to sad songs on a monday morning otw...
31960	31961	1	@user #sikh #temple vandalised in in #calgary,...
31961	31962	0	thank you @user for you follow

In [6]:
```
df.shape
```

Out[6]:
```
(31962, 3)
```

In [20]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31962 entries, 0 to 31961
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      31962 non-null  int64 
 1   label   31962 non-null  int64 
 2   tweet   31962 non-null  object
dtypes: int64(2), object(1)
memory usage: 749.2+ KB
```

In [21]:
```
df.isnull().sum()
```

Out[21]:
```
id       0
label    0
tweet    0
dtype: int64
```

In [22]:
```
df['label'].value_counts()
```

Out[22]:
```
0    29720
1     2242
Name: label, dtype: int64
```

Data Preprocessing

In [23]:
```
def remove_pattern(input_txt, pattern):
    r = re.findall(pattern, input_txt)
    for word in r:
        input_txt = re.sub(word, "", input_txt)
    return input_txt
```

In [24]:
```
# Remove the twitter handles @user
df['processed_tweet'] = np.vectorize(remove_pattern)(df['tweet'],"@[\w]*")
```

In [25]:
```
df.head()
```

Out[25]:

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

In [26]:
```
#remove special chars numbers and punctuations
df['processed_tweet'] = df['processed_tweet'].str.replace("[^a-zA-Z#]"," ")
```

In [27]:
```
df.head()
```

Out[27]:

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

In [ ]:

In [28]:
```
# tokenization
tt = TweetTokenizer()
tokennized_tweet = df['processed_tweet'].apply(lambda x: tt.tokenize(x))
```

In [29]:
```
print(tokennized_tweet)
```

```
0        [when, a, father, is, dysfunctional, and, is, ...
1        [thanks, for, #lyft, credit, i, can, t, use, c...
2                                  [bihday, your, majesty]
3        [#model, i, love, u, take, with, u, all, the, ...
4                  [factsguide, society, now, #motivation]
                               ...
31957                              [ate, isz, that, youuu]
31958    [to, see, nina, turner, on, the, airwaves, try...
31959    [listening, to, sad, songs, on, a, monday, mor...
31960    [#sikh, #temple, vandalised, in, in, #calgary,...
31961                       [thank, you, for, you, follow]
Name: processed_tweet, Length: 31962, dtype: object
```

In [30]:
```
stemmer = PorterStemmer()
tokennized_tweet = tokennized_tweet.apply(lambda sentence: [stemmer.stem(word) for word in sentence])
```

In [36]:
```
tokennized_tweet
```

Out[36]:
```
0        [when, a, father, is, dysfunct, and, is, so, s...
1        [thank, for, #lyft, credit, i, can, t, use, ca...
2                                  [bihday, your, majesti]
3        [#model, i, love, u, take, with, u, all, the, ...
4                        [factsguid, societi, now, #motiv]
                               ...
31957                              [ate, isz, that, youuu]
31958    [to, see, nina, turner, on, the, airwav, tri, ...
31959    [listen, to, sad, song, on, a, monday, morn, o...
31960    [#sikh, #templ, vandalis, in, in, #calgari, #w...
31961                       [thank, you, for, you, follow]
Name: processed_tweet, Length: 31962, dtype: object
```

In [37]:
```
#remove stop words

from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
```

In [38]:
```
def remove_stop_words(tokens):
    # Tokenize the text

    # Remove stop words
    stop_words = set(stopwords.words('english'))
    filtered_tokens = [token for token in tokens if token.lower() not in stop_words]
    # Return the filtered tokens as a string
    return filtered_tokens
```

In [39]:
```
tokennized_tweet = tokennized_tweet.apply(remove_stop_words)
```

In [40]:

```
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()

tokennized_tweet = tokennized_tweet.apply(lambda sentence: [stemmer.stem(word) for word in sentence])
tokennized_tweet.head()
```

Out[40]:

```
0    [father, dysfunct, selfish, drag, hi, kid, hi,...
1    [thank, #lyft, credit, use, cau, offer, wheelc...
2                                    [bihday, majesti]
3                    [#model, love, u, take, u, time, ur]
4                            [factsguid, societi, #motiv]
Name: processed_tweet, dtype: object
```

In [41]:

```
# combine words into single sentence
for i in range(len(tokennized_tweet)):
    tokennized_tweet[i] = " ".join(tokennized_tweet[i])

df['processed_tweet'] = tokennized_tweet
df.head()
```

Out[41]:

	id	label	tweet	processed_tweet
0	1	0	@user when a father is dysfunctional and is s...	father dysfunct selfish drag hi kid hi dysfunc...
1	2	0	@user @user thanks for #lyft credit i can't us...	thank #lyft credit use cau offer wheelchair va...
2	3	0	bihday your majesty	bihday majesti
3	4	0	#model i love u take with u all the time in ...	#model love u take u time ur
4	5	0	factsguide: society now #motivation	factsguid societi #motiv

In [ ]:

```
```

EDA

In [42]:
```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\users\my pc\anaconda3\lib\site-packages (1.9.1.1)
Requirement already satisfied: numpy>=1.6.1 in c:\users\my pc\anaconda3\lib\site-packages (from wordcloud)
(1.23.5)
Requirement already satisfied: matplotlib in c:\users\my pc\anaconda3\lib\site-packages (from wordcloud)
(3.7.0)
Requirement already satisfied: pillow in c:\users\my pc\anaconda3\lib\site-packages (from wordcloud) (9.5.
0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\my pc\anaconda3\lib\site-packages (from mat
plotlib->wordcloud) (2.8.2)
Requirement already satisfied: packaging>=20.0 in c:\users\my pc\anaconda3\lib\site-packages (from matplotl
ib->wordcloud) (22.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\my pc\anaconda3\lib\site-packages (from matplo
tlib->wordcloud) (4.25.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\my pc\anaconda3\lib\site-packages (from matplot
lib->wordcloud) (1.0.5)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\my pc\anaconda3\lib\site-packages (from matplot
lib->wordcloud) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\my pc\anaconda3\lib\site-packages (from matplo
tlib->wordcloud) (1.4.4)
Requirement already satisfied: cycler>=0.10 in c:\users\my pc\anaconda3\lib\site-packages (from matplotlib-
>wordcloud) (0.11.0)
Requirement already satisfied: six>=1.5 in c:\users\my pc\anaconda3\lib\site-packages (from python-dateutil
>=2.7->matplotlib->wordcloud) (1.16.0)
```

In [44]:
```
pip install --upgrade Pillow
```

```
Requirement already satisfied: Pillow in c:\users\my pc\anaconda3\lib\site-packages (9.5.0)
Note: you may need to restart the kernel to use updated packages.
```

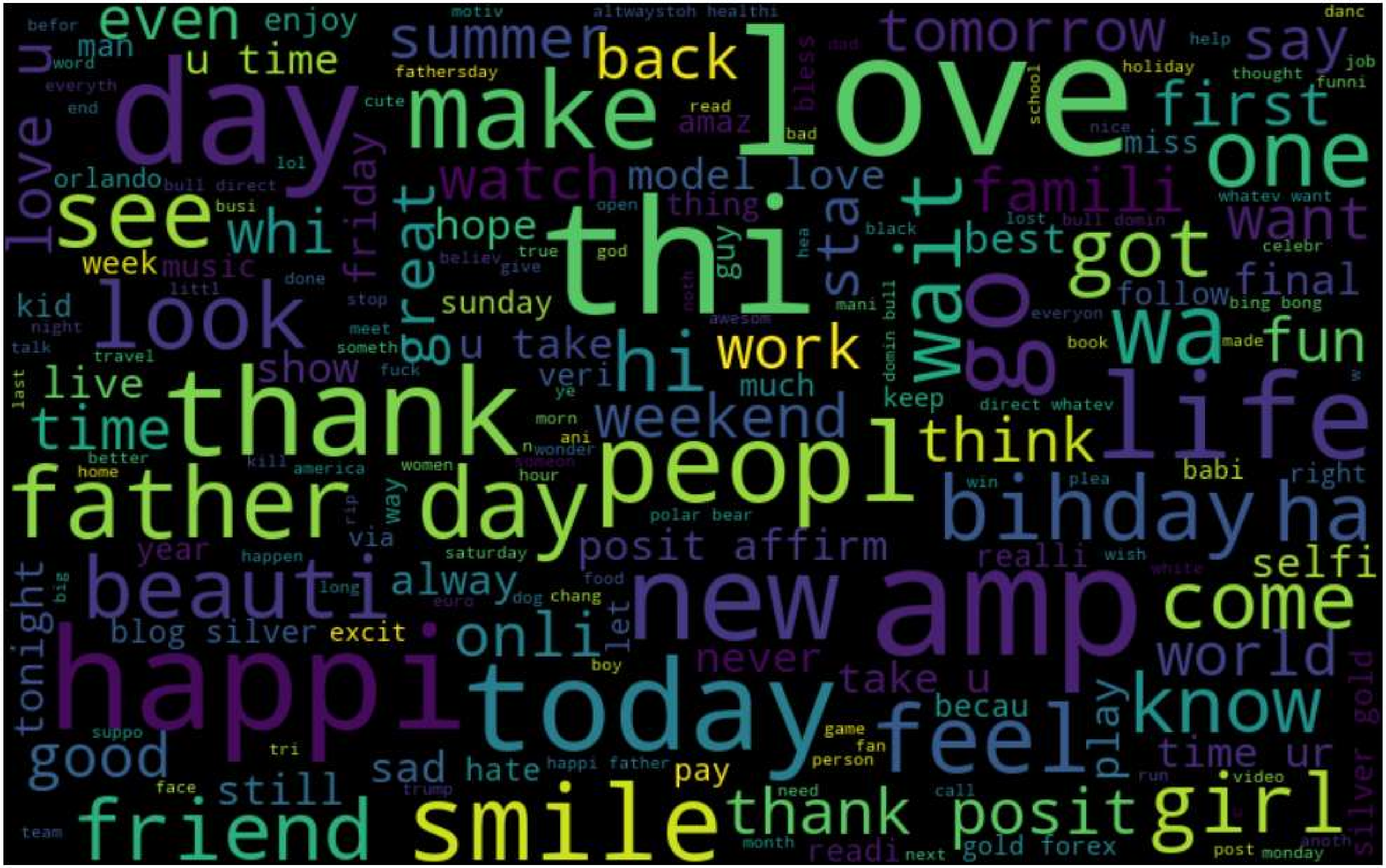
In [45]:
```
# Visualize frequant words
words_total = " ".join([sentence for sentence in df['processed_tweet']])
```

In [46]:
```
words_total
```

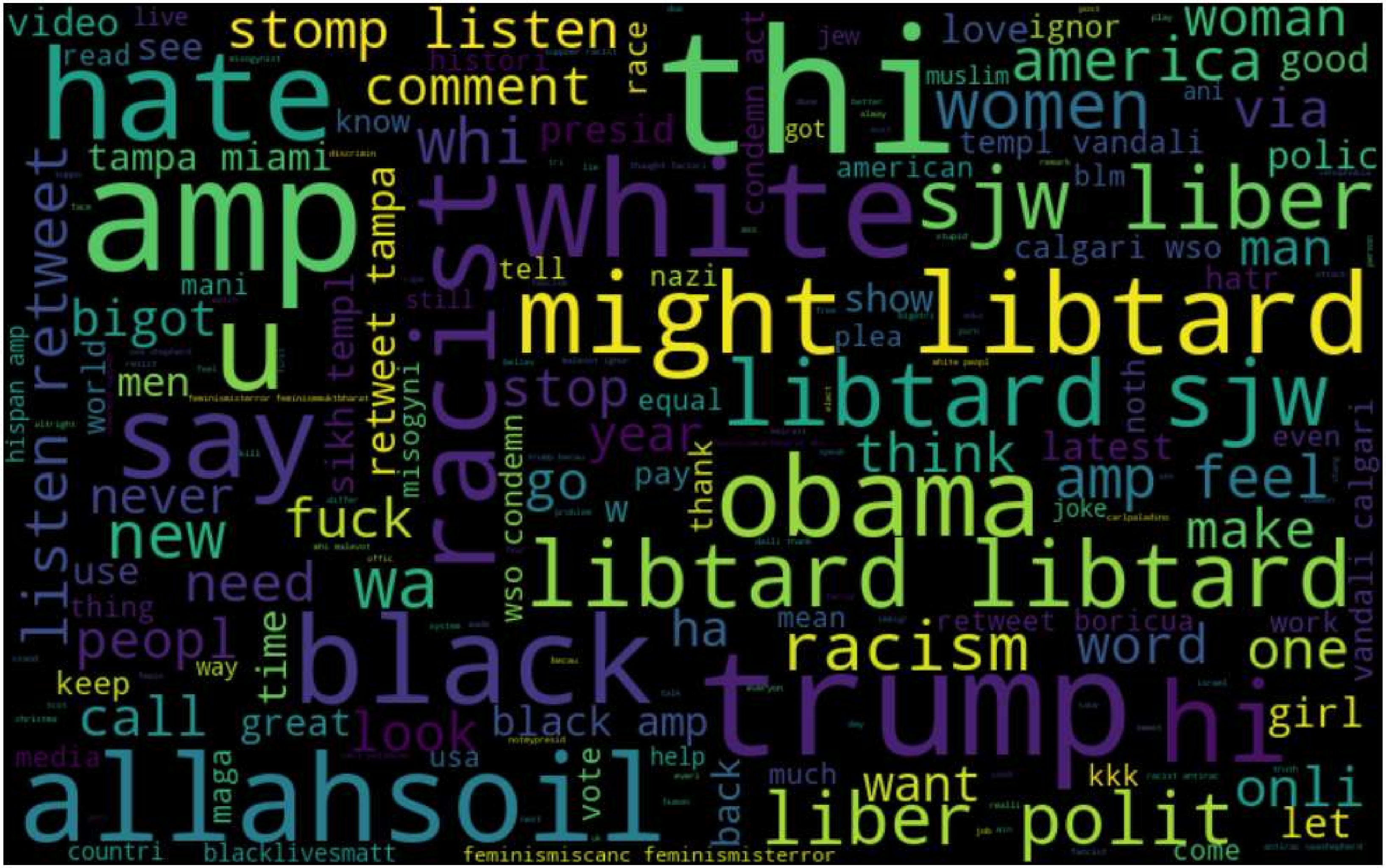
Out[46]:
```
'father dysfunct selfish drag hi kid hi dysfunct #run thank #lyft credit use cau offer wheelchair van pd
x #disapoint #getthank bihday majesti #model love u take u time ur factsguid societi #motiv huge fan far
e big talk befor leav chao pay disput get #allshowandnogo camp tomorrow danni next school year year exam
think #school #exam #hate #imagin #actorslif #revolutionschool #girl love land #allin #cav #champion #cl
eveland #clevelandcavali welcom #gr #ireland consum price index mom climb previou may #blog #silver #gol
d #forex selfish #orlando #standwithorlando #pulseshoot #orlandoshoot #biggerproblem #selfish #heabreak
#valu #love # get see daddi today # day #gettingf #cnn call #michigan middl school build wall chant #tco
t comment #australia #opkillingbay #seashepherd #helpcovedolphin #thecov #helpcovedolphin ouch junior an
gri #got #junior #yugyoem #omg thank paner #thank #posit retweet agr #friday smile around via ig user #c
ooki make peopl know essenti oil made chemic #euro peopl blame ha conc goal wa fat rooney gave away free
kick know bale hit sad littl dude #badday #coneofsham #cat #piss #funni #laugh product day happi man #wi
ne tool #weekend time open amp drink lumpi say prove lumpi #tgif #ff #gamedev #indiedev #indiegamedev #s
quad beauti sign vendor #upsideofflorida #shopalyssa #love #smile #media #pressconf #antalya #turkey sun
day #throwback love great panel mediat public servic #ica happi father day peopl went nightclub good nig
ht man action mean peopl lost famili forev #rip #orlando never chanc vote presidenti candid wa excit thi
cycl look differ #alohafriday #time doe #not #exist #positivevib #hawaiian rip fellow nohern ireland fan
sadley pass away tonight gawa forev sing cheer fire wa hard monday due cloudi weather disabl oxygen prod
uct today #goodnight #badmonday unbeliev st centuri need someth like thi #neverump #xenophobia #taylorsw
ift bull domin bull direct whatev want w morn #travelingram #dalat #ripinkylif onc onli one word tell #p
```

In [47]:

```
from wordcloud import WordCloud
wordcloud = WordCloud(width=800, height=500, random_state=42, max_font_size=100).generate(words_total)

# plot the graph
plt.figure(figsize=(15,8))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```



```
In [48]: words_total_label_0= " ".join([sentence for sentence in df['processed_tweet'][df['label']==1]])

         wordcloud = WordCloud(width=800, height=500, random_state=42, max_font_size=100).generate(words_total_label_

         # plot the graph
         plt.figure(figsize=(15,8))
         plt.imshow(wordcloud, interpolation='bilinear')
         plt.axis('off')
         plt.show()
```



In [49]:

```
# extract the hashtag
def hashtag_extract(tweets):
    hashtags = []
    # loop words in the tweet
    for tweet in tweets:
        ht = re.findall(r"#(\w+)", tweet)
        hashtags.append(ht)
    return hashtags
```

In [50]:

```
# extract hashtags from non-racist/sexist tweets
ht_positive = hashtag_extract(df['processed_tweet'][df['label']==0])

# extract hashtags from racist/sexist tweets
ht_negative = hashtag_extract(df['processed_tweet'][df['label']==1])
```

In [51]:

```
ht_positive[:5]
```

Out[51]:

```
[['run'], ['lyft', 'disapoint', 'getthank'], [], ['model'], ['motiv']]
```

In [52]:

```
ht_positive = sum(ht_positive, [])
ht_negative = sum(ht_negative, [])
```

In [53]:

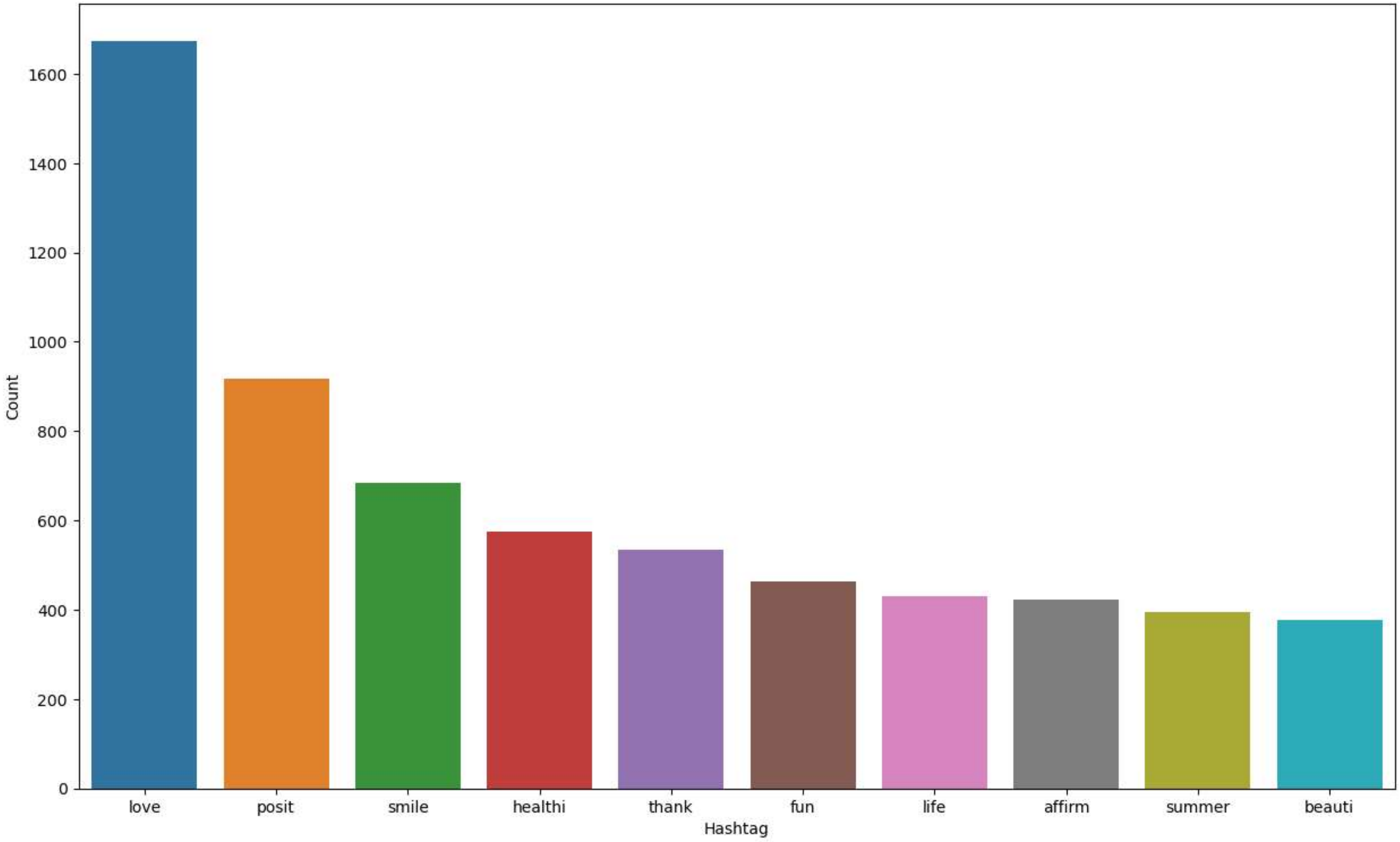
```
freq = nltk.FreqDist(ht_positive)
d = pd.DataFrame({'Hashtag': list(freq.keys()),
                  'Count': list(freq.values())})
d.head()
```

Out[53]:

	Hashtag	Count
0	run	72
1	lyft	2
2	disapoint	1
3	getthank	2
4	model	375

In [54]:

```
# select top 10 hashtags
d = d.nlargest(columns='Count', n=10)
plt.figure(figsize=(15,9))
sns.barplot(data=d, x='Hashtag', y='Count')
plt.show()
```



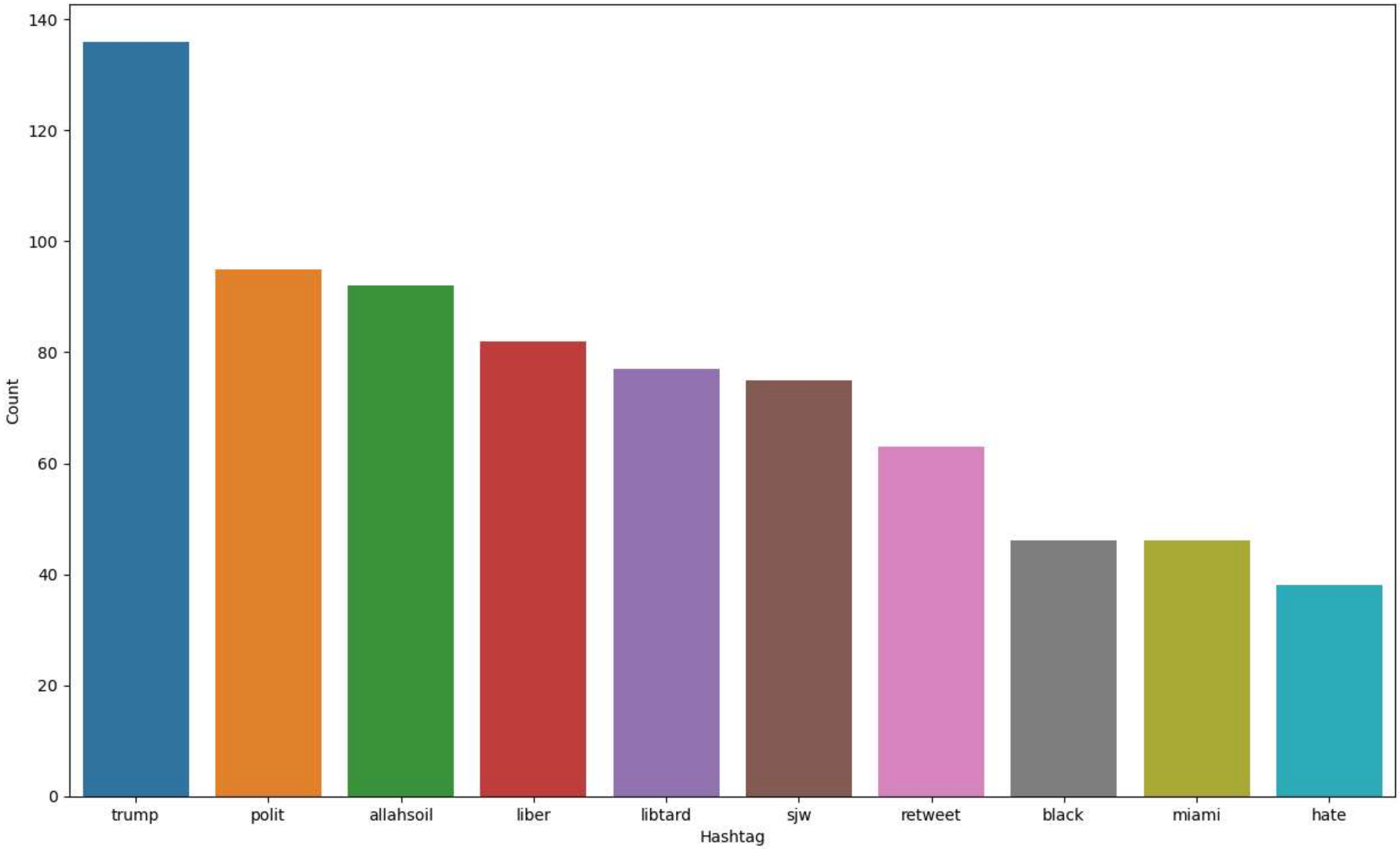
In [55]:

```
freq = nltk.FreqDist(ht_negative)
d = pd.DataFrame({'Hashtag': list(freq.keys()),
                  'Count': list(freq.values())})
d.head()
```

Out[55]:

	Hashtag	Count
0	cnn	10
1	michigan	2
2	tcot	14
3	australia	6
4	opkillingbay	5

```
In [56]: # select top 10 hashtags
         d = d.nlargest(columns='Count', n=10)
         plt.figure(figsize=(15,9))
         sns.barplot(data=d, x='Hashtag', y='Count')
         plt.show()
```



Slipt Dataset train test

In [ ]:

```
```

In [57]:

```
from sklearn.feature_extraction.text import CountVectorizer
bow_vectorizer = CountVectorizer(max_df=0.90, min_df=2, max_features=1000, stop_words='english')
bow = bow_vectorizer.fit_transform(df['processed_tweet'])
```

In [58]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(bow, df['label'], random_state=42, test_size=0.25)
```

Model Training

In [59]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score, accuracy_score
```

In [60]:

```
# training
model = LogisticRegression()
model.fit(x_train, y_train)
```

Out[60]:

```
LogisticRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [61]:

```
# testing
pred = model.predict(x_test)
f1_score(y_test, pred)
```

Out[61]:

```
0.5029655990510082
```

In [62]:

```
accuracy_score(y_test,pred)
```

Out[62]:

```
0.9475660117632336
```

In [63]:

```
pred_prob = model.predict_proba(x_test)
pred = pred_prob[:, 1] >= 0.3
pred = pred.astype(np.int)

f1_score(y_test, pred)
```

Out[63]:

```
0.566147859922179
```

In [64]:

```
from sklearn.metrics import confusion_matrix
```

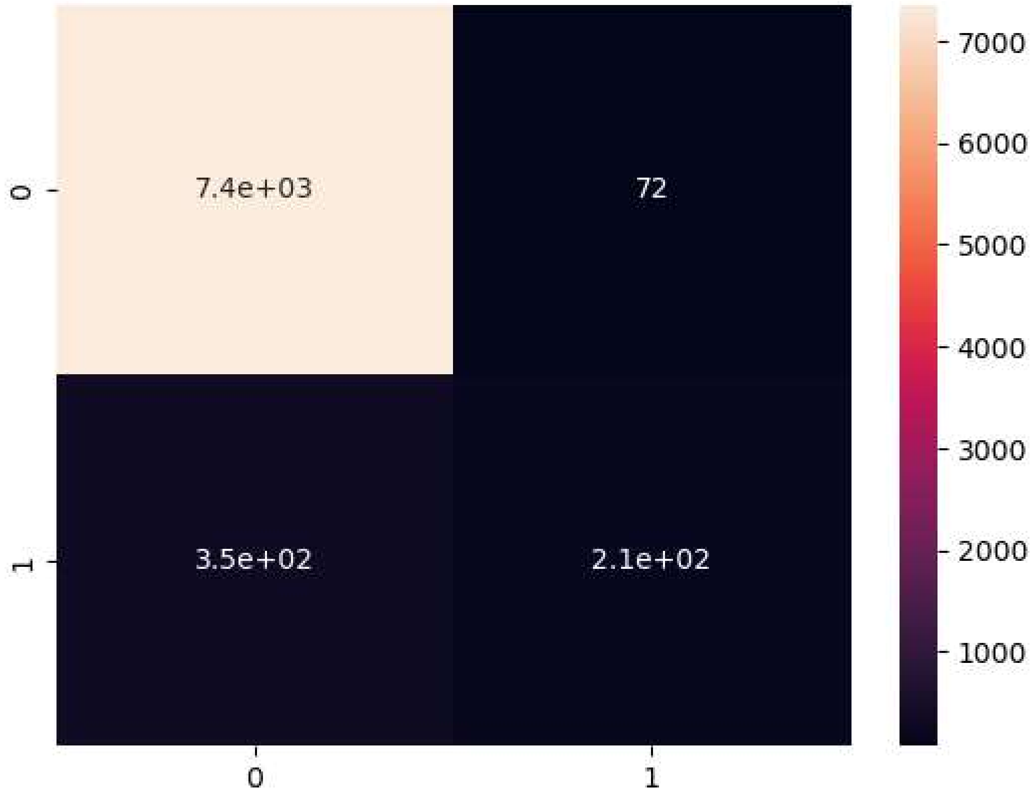
In [65]:

```
labels = [0, 1]
```

In [66]:

```
y_pred = model.predict(x_test)
```

```
In [67]: print(confusion_matrix(y_test, y_pred))
         sns.heatmap(confusion_matrix(y_test, y_pred),annot=True)

         [[7360   72]
          [ 347  212]]

Out[67]: <Axes: >
```



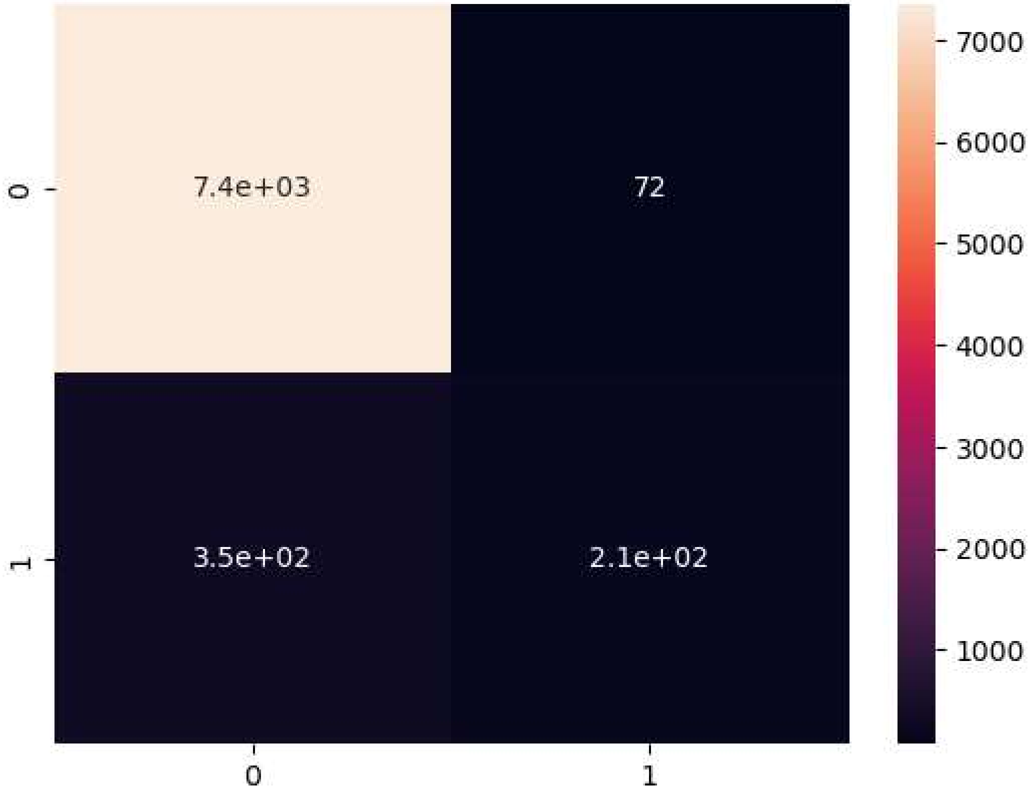
In [68]:

```
print(confusion_matrix(y_test, y_pred))
sns.heatmap(confusion_matrix(y_test, y_pred),annot=True)
```

```
[[7360   72]
 [ 347  212]]
```

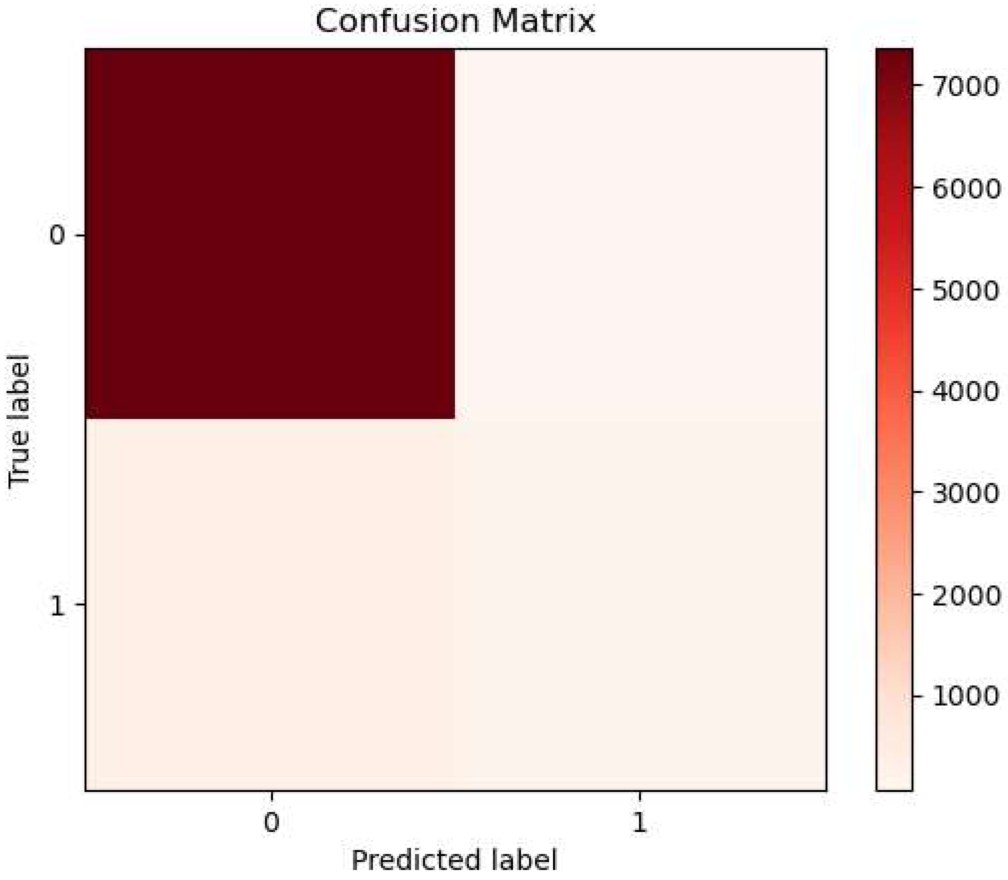
Out[68]:

```
<Axes: >
```



In [69]:

```
cm = confusion_matrix(y_test, y_pred, labels=labels)

# plot the confusion matrix as a heatmap
plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Reds)
plt.colorbar()
tick_marks = np.arange(len(labels))
plt.xticks(tick_marks, labels)
plt.yticks(tick_marks, labels)
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Confusion Matrix')
plt.show()
```



In [ ]:

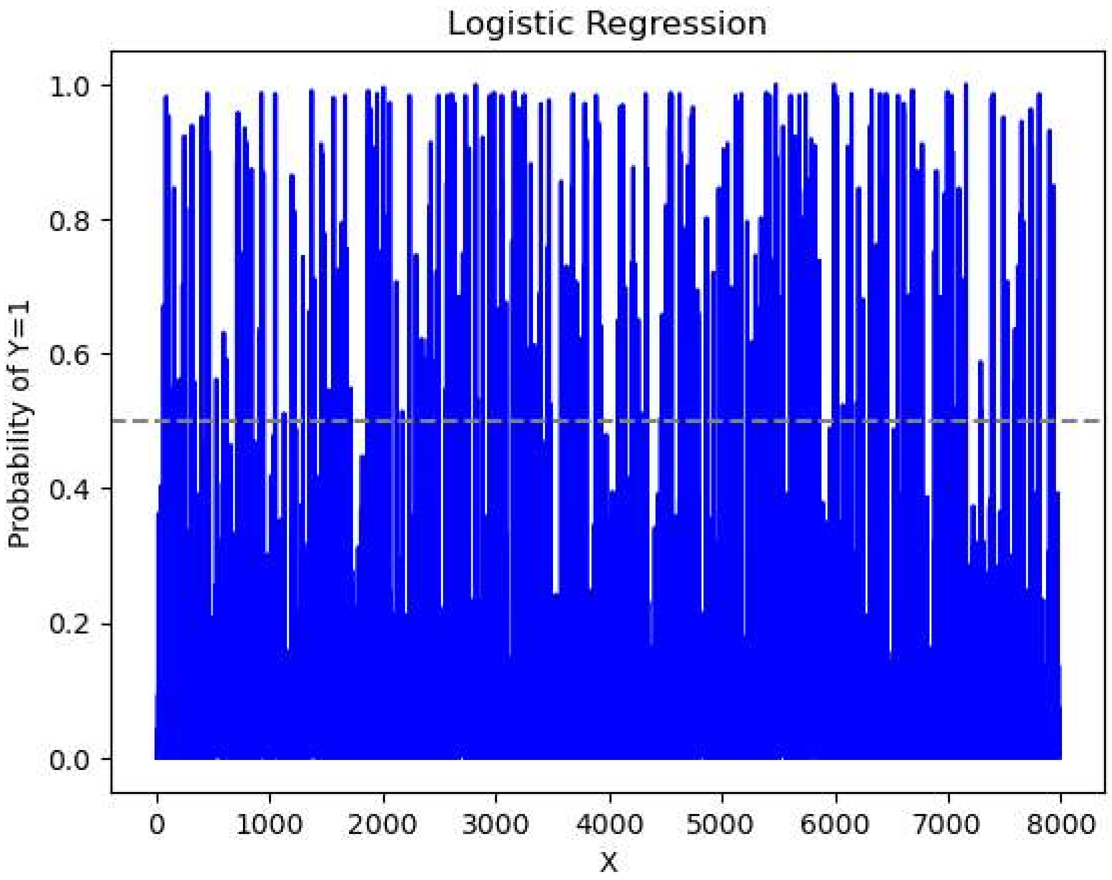
```

```

In [70]:

```
probabilities = model.predict_proba(x_test)[:, 1]
```

In [64]:

```
plt.plot(probabilities, color='blue')

# plot the threshold line at 0.5
plt.axhline(y=0.5, color='gray', linestyle='--')

# add labels and title
plt.xlabel('X')
plt.ylabel('Probability of Y=1')
plt.title('Logistic Regression')
plt.show()
```



In [66]:

```
x_train.toarray()
```

Out[66]:

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

USing Nave Bays therom

In [67]:

```
from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()
```

In [68]:

```

gnb.fit(x_train.toarray(), y_train)

#Predict the response for test dataset

y_pred_a = gnb.predict(x_test.toarray())

print(y_pred_a)
```

```
[0 0 0 ... 0 0 0]
```

In [69]:

```
from sklearn import metrics

# Model Accuracy

print("Accuracy:",metrics.accuracy_score(y_test, y_pred_a))
```

```
Accuracy: 0.5153297459642098
```

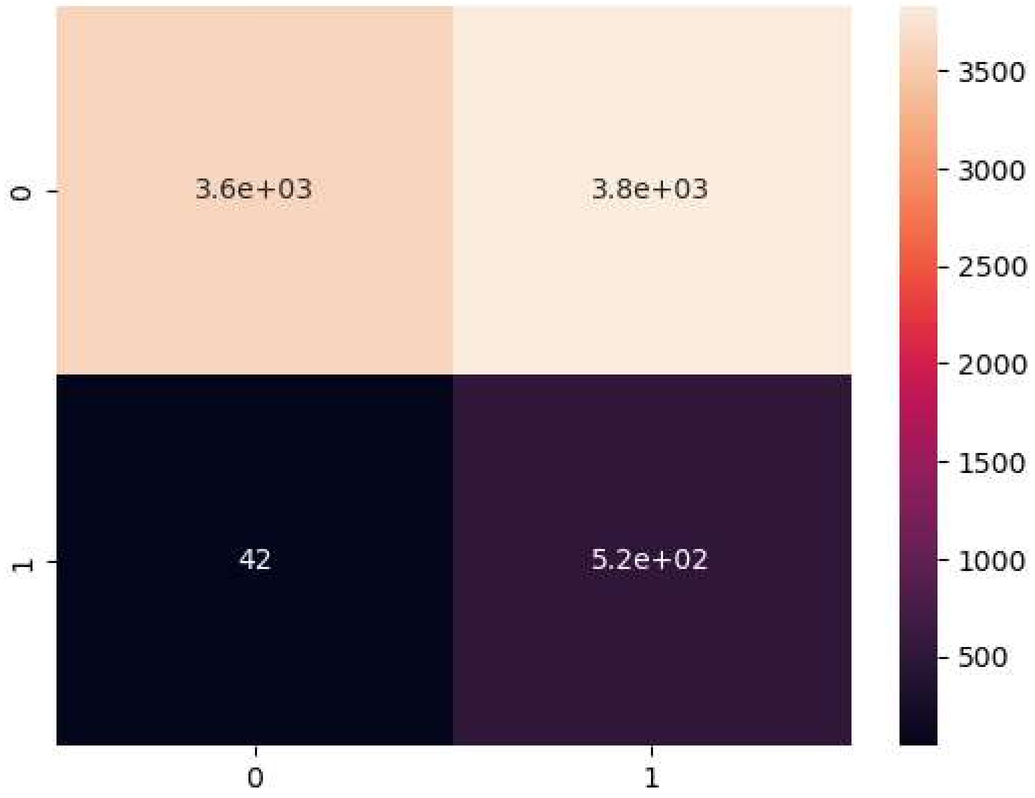
In [70]:
```
y_train
```

Out[70]:
```
19010    0
5474     0
6557     0
3617     0
5099     0
        ..
29802    0
5390     0
860      1
15795    0
23654    0
Name: label, Length: 23971, dtype: int64
```

In [71]:
```
x_train.toarray()
```

Out[71]:
```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

```
In [73]: print(confusion_matrix(y_test, y_pred_a))
         sns.heatmap(confusion_matrix(y_test, y_pred_a),annot=True)

         [[3601 3831]
          [  42  517]]

Out[73]: <Axes: >
```



```
In [ ]:

In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```