TASK-7: TWITTER SENTIMENT ANALYSIS

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import string
import nltk
import warnings
%matplotlib inline

warnings.filterwarnings('ignore')
```

In [2]:
```
df = pd.read_csv('Twitter Sentiments.csv')
```

In [39]:
```
df.shape
```

Out[39]: (31962, 4)

In [42]:
```
df.describe()
```

Out[42]:

	id	label
count	31962.000000	31962.000000
mean	15981.500000	0.070146
std	9226.778988	0.255397
min	1.000000	0.000000
25%	7991.250000	0.000000
50%	15981.500000	0.000000
75%	23971.750000	0.000000
max	31962.000000	1.000000

In [43]:
```
df.tail()
```

Out[43]:

	id	label	tweet	clean_tweet
31957	31958	0	ate @user isz that youuu? ðŸ□□□ð□□□ð□□□ð□□□ð□□□ð...	that youuu
31958	31959	0	to see nina turner on the airwaves trying to...	nina turner airwav tri wrap herself mantl genu...
31959	31960	0	listening to sad songs on a monday morning otw...	listen song monday morn work

	id	label	tweet	clean_tweet
31960	31961	1	@user #sikh #temple vandalised in in #calgary,...	#sikh #templ vandalis #calgari #wso condemn
31961	31962	0	thank you @user for you follow	thank follow

In [44]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31962 entries, 0 to 31961
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   id           31962 non-null  int64 
 1   label        31962 non-null  int64 
 2   tweet        31962 non-null  object
 3   clean_tweet  31962 non-null  object
dtypes: int64(2), object(2)
memory usage: 998.9+ KB
```

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]:
```
# datatype info
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 [5]:
```
# removes pattern in the input text
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 [6]:
```
df.head()
```

Out[6]:

	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 [7]:

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

In [8]:

```
df.head()
```

Out[8]:

	id	label	tweet	clean_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 [9]:

```
# remove special characters, numbers and punctuations
df['clean_tweet'] = df['clean_tweet'].str.replace("[^a-zA-Z#]", " ")
df.head()
```

Out[9]:

	id	label	tweet	clean_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 [10]:

```
# remove short words
df['clean_tweet'] = df['clean_tweet'].apply(lambda x: " ".join([w for w in x.split() if
df.head()
```

Out[10]:

	id	label	tweet	clean_tweet
0	1	0	@user when a father is dysfunctional and is s...	when father dysfunctional selfish drags kids i...
1	2	0	@user @user thanks for #lyft credit i can't us...	thanks #lyft credit cause they offer wheelchai...

	id	label	tweet	clean_tweet
2	3	0	bihday your majesty	bihday your majesty
3	4	0	#model i love u take with u all the time in ...	#model love take with time
4	5	0	factsguide: society now #motivation	factsguide society #motivation

In [11]:
```
# individual words considered as tokens
tokenized_tweet = df['clean_tweet'].apply(lambda x: x.split())
tokenized_tweet.head()
```

Out[11]:
```
0    [when, father, dysfunctional, selfish, drags, ...
1    [thanks, #lyft, credit, cause, they, offer, wh...
2                               [bihday, your, majesty]
3                     [#model, love, take, with, time]
4                   [factsguide, society, #motivation]
Name: clean_tweet, dtype: object
```

In [12]:
```
# stem the words
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()

tokenized_tweet = tokenized_tweet.apply(lambda sentence: [stemmer.stem(word) for word i
tokenized_tweet.head()
```

Out[12]:
```
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]
Name: clean_tweet, dtype: object
```

In [13]:
```
# combine words into single sentence
for i in range(len(tokenized_tweet)):
    tokenized_tweet[i] = " ".join(tokenized_tweet[i])

df['clean_tweet'] = tokenized_tweet
df.head()
```

Out[13]:

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

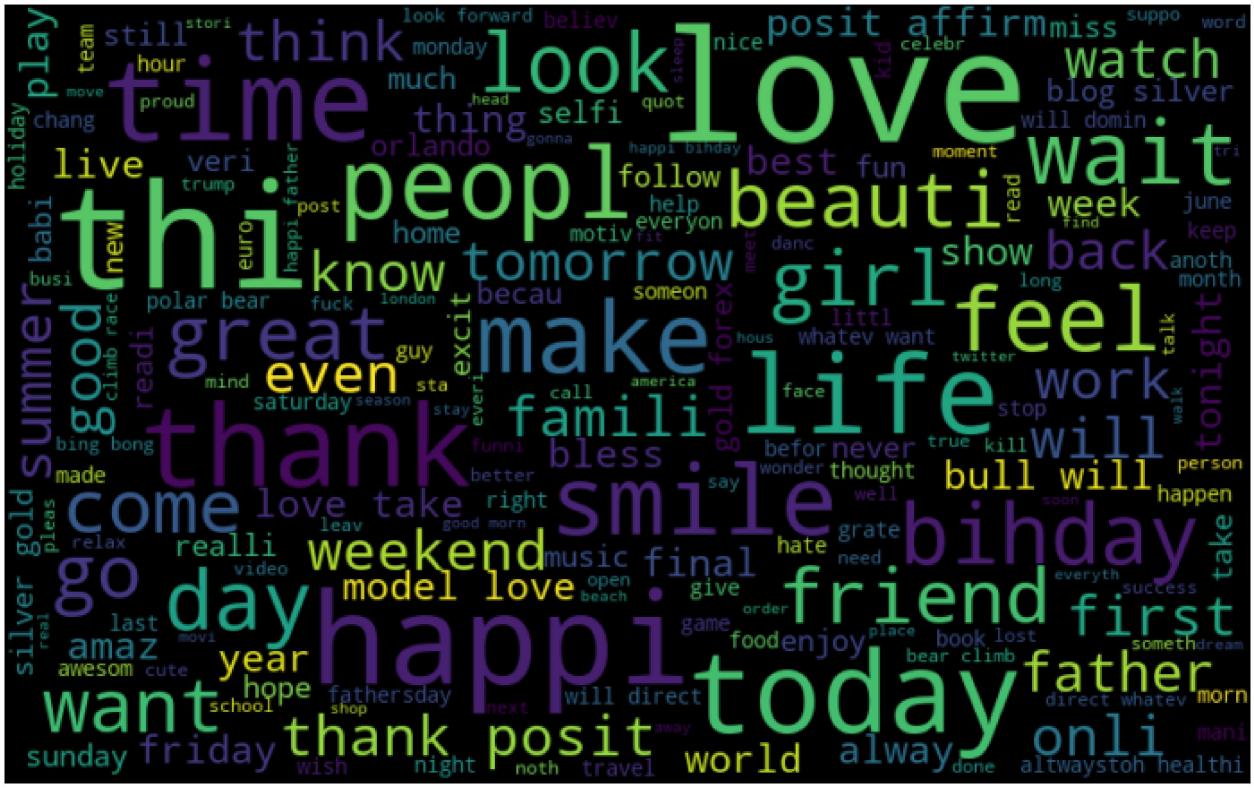
In [14]:
```
# !pip install wordcloud
```

In [15]:
```
# visualize the frequent words
all_words = " ".join([sentence for sentence in df['clean_tweet']])
```

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

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

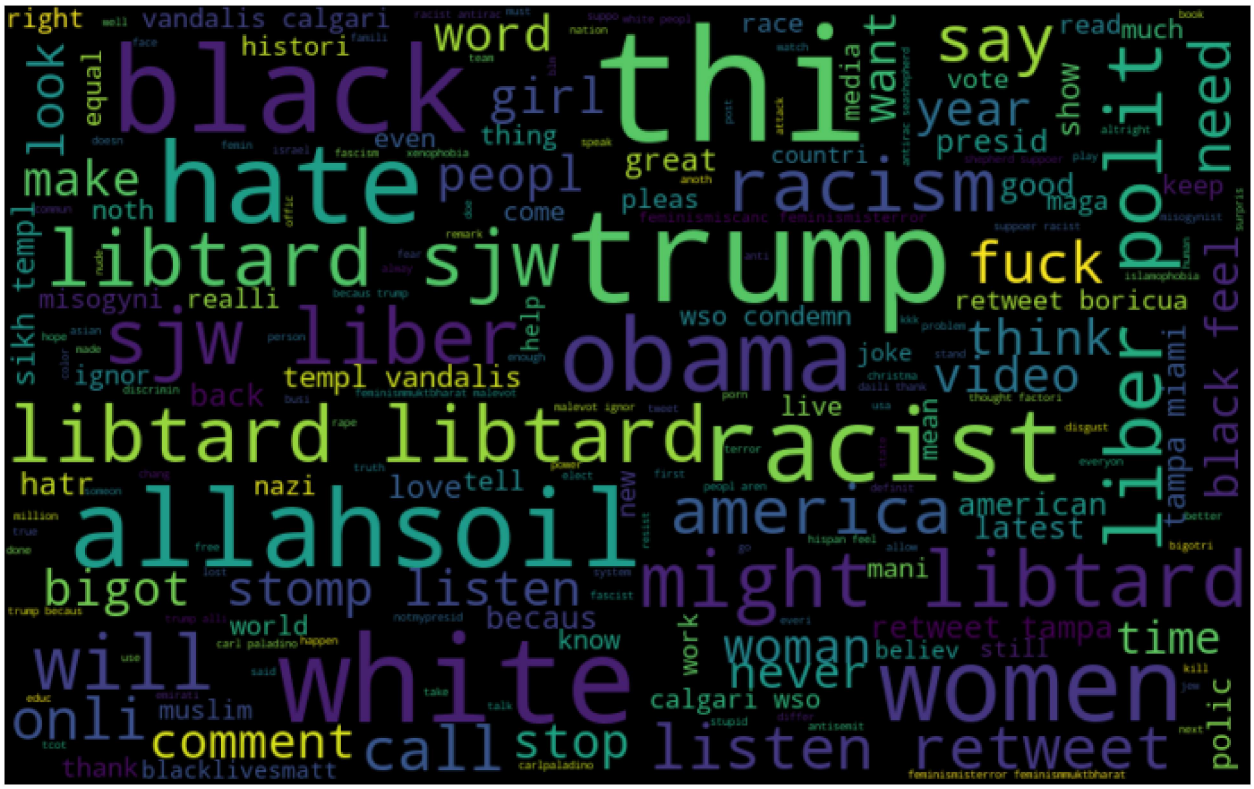
In [16]:

```
# frequent words visualization for +ve
all_words = " ".join([sentence for sentence in df['clean_tweet'][df['label']==0]])

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

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



In [17]:

```
# frequent words visualization for -ve
all_words = " ".join([sentence for sentence in df['clean_tweet'][df['label']==1]])

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

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



In [18]:
```
# 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 [19]:
```
# extract hashtags from non-racist/sexist tweets
ht_positive = hashtag_extract(df['clean_tweet'][df['label']==0])

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

In [20]:
```
ht_positive[:5]
```

Out[20]:
```
[['run'], ['lyft', 'disapoint', 'getthank'], [], ['model'], ['motiv']]
```

In [21]:
```
# unnest list
ht_positive = sum(ht_positive, [])
ht_negative = sum(ht_negative, [])
```

In [22]:
```
ht_positive[:5]
```

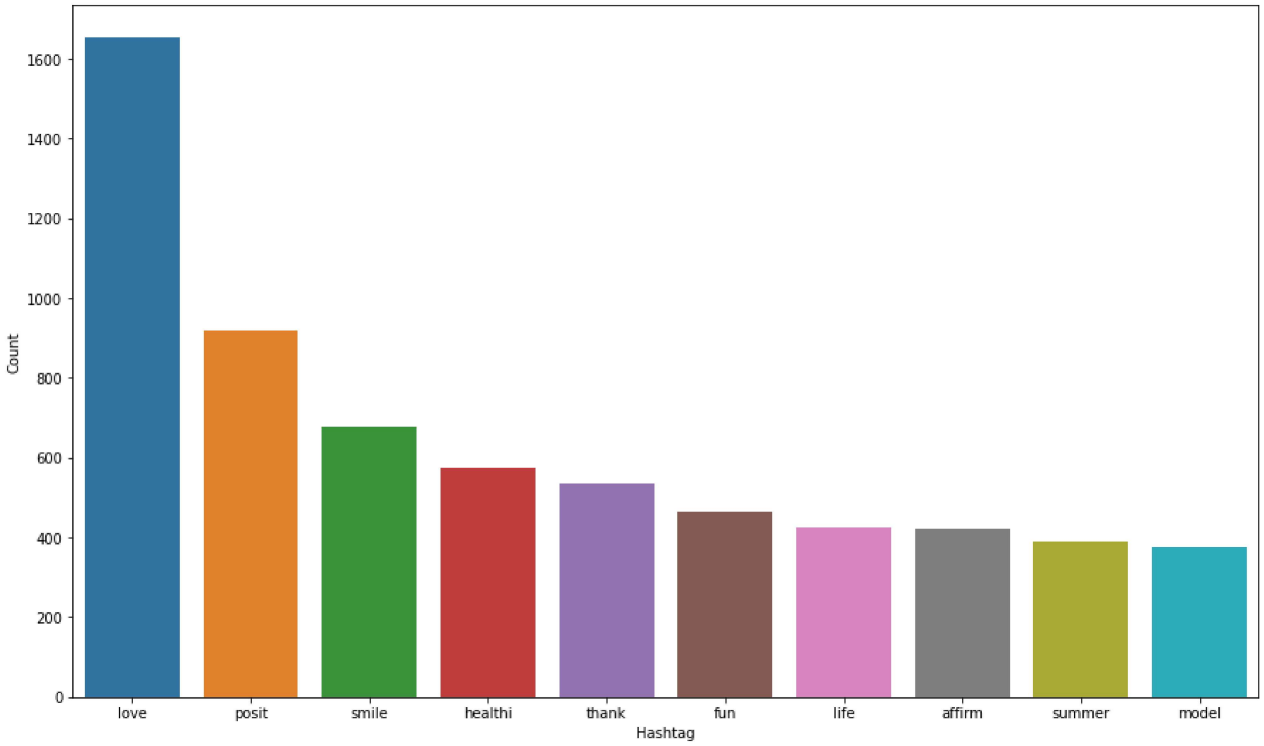
Out[22]:
```
['run', 'lyft', 'disapoint', 'getthank', 'model']
```

In [23]:
```
freq = nltk.FreqDist(ht_positive)
d = pd.DataFrame({'Hashtag': list(freq.keys()),
                  'Count': list(freq.values())})
d.head()
```

Out[23]:

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

In [24]:
```
# 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()
```



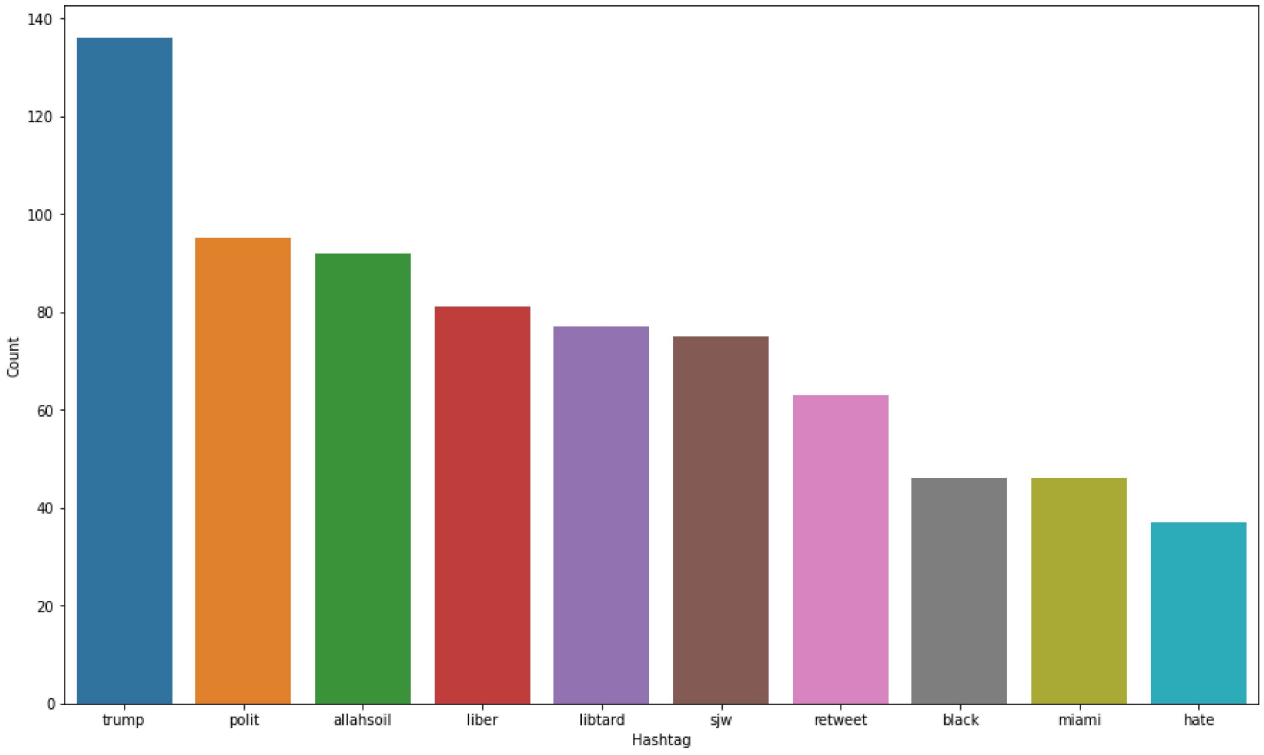
In [25]:
```
freq = nltk.FreqDist(ht_negative)
d = pd.DataFrame({'Hashtag': list(freq.keys()),
                  'Count': list(freq.values())})
d.head()
```

Out[25]:

	Hashtag	Count
0	cnn	10
1	michigan	2

	Hashtag	Count
2	tcot	14
3	australia	6
4	opkillingbay	5

In [26]:
```
# 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 [27]:
```
# feature extraction
from sklearn.feature_extraction.text import CountVectorizer
bow_vectorizer = CountVectorizer(max_df=0.90, min_df=2, max_features=1000, stop_words='
bow = bow_vectorizer.fit_transform(df['clean_tweet'])
```

In [28]:
```
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,
```

In [29]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score, accuracy_score
```

In [30]:
```
# training
model = LogisticRegression()
model.fit(x_train, y_train)
```

Out[30]:
```
LogisticRegression()
```

In [31]:
```
# testing
pred = model.predict(x_test)
f1_score(y_test, pred)
```

Out[31]:
```
0.49763033175355453
```

In [32]:
```
accuracy_score(y_test,pred)
```

Out[32]:
```
0.9469403078463271
```

In [33]:
```
# use probability to get output
pred_prob = model.predict_proba(x_test)
pred = pred_prob[:, 1] >= 0.3
pred = pred.astype(np.int)

f1_score(y_test, pred)
```

Out[33]:
```
0.5545722713864307
```

In [34]:
```
accuracy_score(y_test,pred)
```

Out[34]:
```
0.9433112251282693
```

In [35]:
```
pred_prob[0][1] >= 0.3
```

Out[35]:
```
False
```

In [ ]:
```

```