**Sentiment Analysis for Marketing**

**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')

**Loading the dataset**

df **=** pd**.**read\_csv('Twitter Sentiments.csv')

df**.**head()

df**.**info()

## Preprocessing the dataset

**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

df**.**head()

df['clean\_tweet'] **=** np**.**vectorize(remove\_pattern)(df['tweet'], "@[\w]\*")

df**.**head()

df['clean\_tweet'] **=** df['clean\_tweet']**.**str**.**replace("[^a-zA-Z#]", " ")

df**.**head()

df['clean\_tweet'] **=** df['clean\_tweet']**.**apply(**lambda** x: " "**.**join([w **for** w **in** x**.**split() **if** len(w)**>**3]))

df**.**head()

tokenized\_tweet **=** df['clean\_tweet']**.**apply(**lambda** x: x**.**split())

tokenized\_tweet**.**head()

**from** nltk.stem.porter **import** PorterStemmer

stemmer **=** PorterStemmer()

tokenized\_tweet **=** tokenized\_tweet**.**apply(**lambda** sentence: [stemmer**.**stem(word) **for** word **in** sentence])

tokenized\_tweet**.**head()

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

tokenized\_tweet[i] **=** " "**.**join(tokenized\_tweet[i])

df['clean\_tweet'] **=** tokenized\_tweet

df**.**head()

## Exploratory Data Analysis

*# !pip install wordcloud*

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)**.**generate(all\_words)

plt**.**figure(figsize**=**(15,8))

plt**.**imshow(wordcloud, interpolation**=**'bilinear')

plt**.**axis('off')

plt**.**show()

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)**.**generate(all\_words)

plt**.**figure(figsize**=**(15,8))

plt**.**imshow(wordcloud, interpolation**=**'bilinear')

plt**.**axis('off')

plt**.**show()

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)**.**generate(all\_words)

plt**.**figure(figsize**=**(15,8))

plt**.**imshow(wordcloud, interpolation**=**'bilinear')

plt**.**axis('off')

plt**.**show()

**def** hashtag\_extract(tweets):

hashtags **=** []

**for** tweet **in** tweets:

ht **=** re**.**findall(r"#(\w+)", tweet)

hashtags**.**append(ht)

**return** hashtags

ht\_positive **=** hashtag\_extract(df['clean\_tweet'][df['label']**==**0])

ht\_negative **=** hashtag\_extract(df['clean\_tweet'][df['label']**==**1])

ht\_positive[:5]

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

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

ht\_positive[:5]

freq **=** nltk**.**FreqDist(ht\_positive)

d **=** pd**.**DataFrame({'Hashtag': list(freq**.**keys()),

'Count': list(freq**.**values())})

d**.**head()

d **=** d**.**nlargest(columns**=**'Count', n**=**10)

plt**.**figure(figsize**=**(15,9))

sns**.**barplot(data**=**d, x**=**'Hashtag', y**=**'Count')

plt**.**show()

freq **=** nltk**.**FreqDist(ht\_negative)

d **=** pd**.**DataFrame({'Hashtag': list(freq**.**keys()),

'Count': list(freq**.**values())})

d**.**head()

d **=** d**.**nlargest(columns**=**'Count', n**=**10)

plt**.**figure(figsize**=**(15,9))

sns**.**barplot(data**=**d, x**=**'Hashtag', y**=**'Count')

plt**.**show()