```python
# This Python 3 environment comes with many helpful analytics
libraries installed
# It is defined by the kaggle/python Docker image:
https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/"
directory
# For example, running this (by clicking run or pressing Shift+Enter)
will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/)
that gets preserved as output when you create a version using "Save &
Run All"
# You can also write temporary files to /kaggle/temp/, but they won't
be saved outside of the current session
```

```
/kaggle/input/sentiment140/training.1600000.processed.noemoticon.csv
```

```python
import tensorflow as tf
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np


import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder


import re

print("Tensorflow Version",tf.__version__)
```

```
[nltk_data] Downloading package stopwords to /usr/share/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Tensorflow Version 2.15.0
```

```
# data preprocessing
df =
pd.read_csv('/kaggle/input/sentiment140/training.1600000.processed.noe
moticon.csv',
                 encoding = 'latin',header=None)
df.head()
```

```
   0           1                             2         3
4  \
0  0  1467810369  Mon Apr 06 22:19:45 PDT 2009  NO_QUERY
_TheSpecialOne_
1  0  1467810672  Mon Apr 06 22:19:49 PDT 2009  NO_QUERY
scotthamilton
2  0  1467810917  Mon Apr 06 22:19:53 PDT 2009  NO_QUERY
mattycus
3  0  1467811184  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY
ElleCTF
4  0  1467811193  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY
Karoli

                                                   5
0  @switchfoot http://twitpic.com/2y1zl - Awww, t...
1  is upset that he can't update his Facebook by ...
2  @Kenichan I dived many times for the ball. Man...
3    my whole body feels itchy and like its on fire
4  @nationwideclass no, it's not behaving at all....
```

```
df.columns = ['sentiment', 'id', 'date', 'query', 'user_id', 'text']
df.head()
```

```
   sentiment          id                          date     query  \
0          0  1467810369  Mon Apr 06 22:19:45 PDT 2009  NO_QUERY
1          0  1467810672  Mon Apr 06 22:19:49 PDT 2009  NO_QUERY
2          0  1467810917  Mon Apr 06 22:19:53 PDT 2009  NO_QUERY
3          0  1467811184  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY
4          0  1467811193  Mon Apr 06 22:19:57 PDT 2009  NO_QUERY

           user_id                                               text

0  _TheSpecialOne_  @switchfoot http://twitpic.com/2y1zl - Awww, t...

1    scotthamilton  is upset that he can't update his Facebook by ...

2         mattycus  @Kenichan I dived many times for the ball. Man...

3          ElleCTF    my whole body feels itchy and like its on fire

4           Karoli  @nationwideclass no, it's not behaving at all....
```

```
df = df.drop(['id', 'date', 'query', 'user_id'], axis=1)
```

```
lab_to_sentiment = {0:"Negative", 4:"Positive"}

def label_decoder(label):
    return lab_to_sentiment[label]

df['sentiment'] = df['sentiment'].map(label_decoder)
df.head()
```

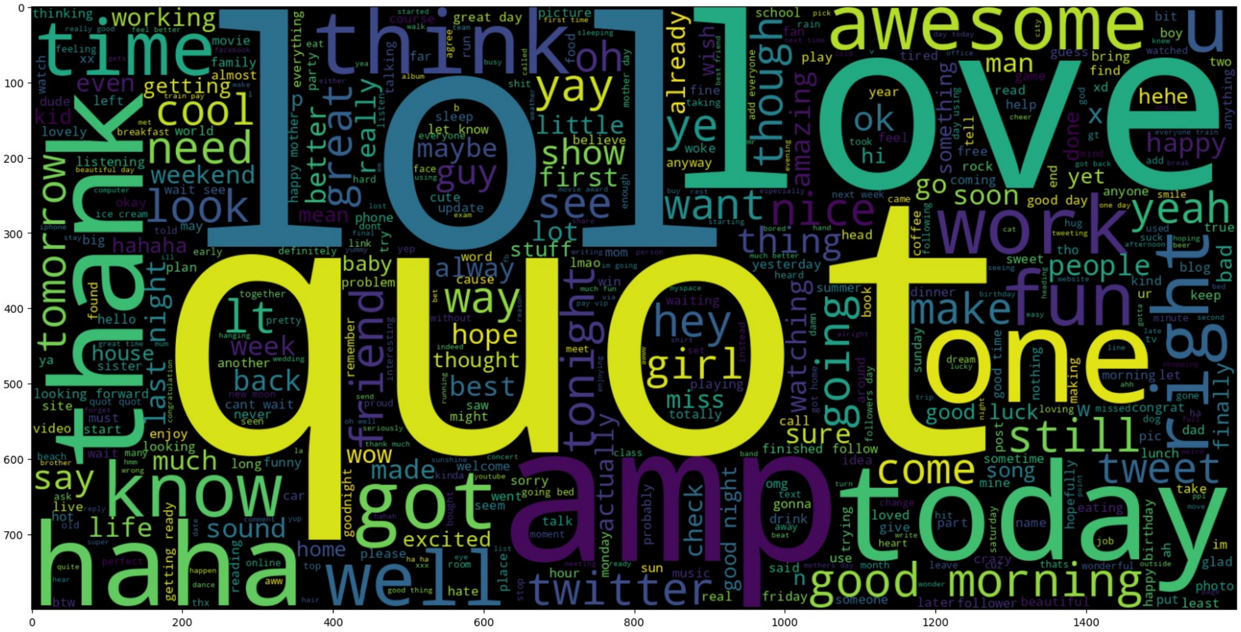
```
  sentiment                                               text
0  Negative  @switchfoot http://twitpic.com/2y1zl - Awww, t...
1  Negative  is upset that he can't update his Facebook by ...
2  Negative  @Kenichan I dived many times for the ball. Man...
3  Negative    my whole body feels itchy and like its on fire 
4  Negative  @nationwideclass no, it's not behaving at all....
```

```
df.sample()
```

```
        sentiment                                               text
436198   Negative  @Ratchyl but you're sitting next to the most a...
1406385  Positive                       Brazil no trending topics!!! 
1374642  Positive  twitter is taking over my life! for example i'...
2095     Negative                          holy shindigs.  thats HOT.
162920   Negative  @nikhilbelsare exactly the same problem i am h...
```

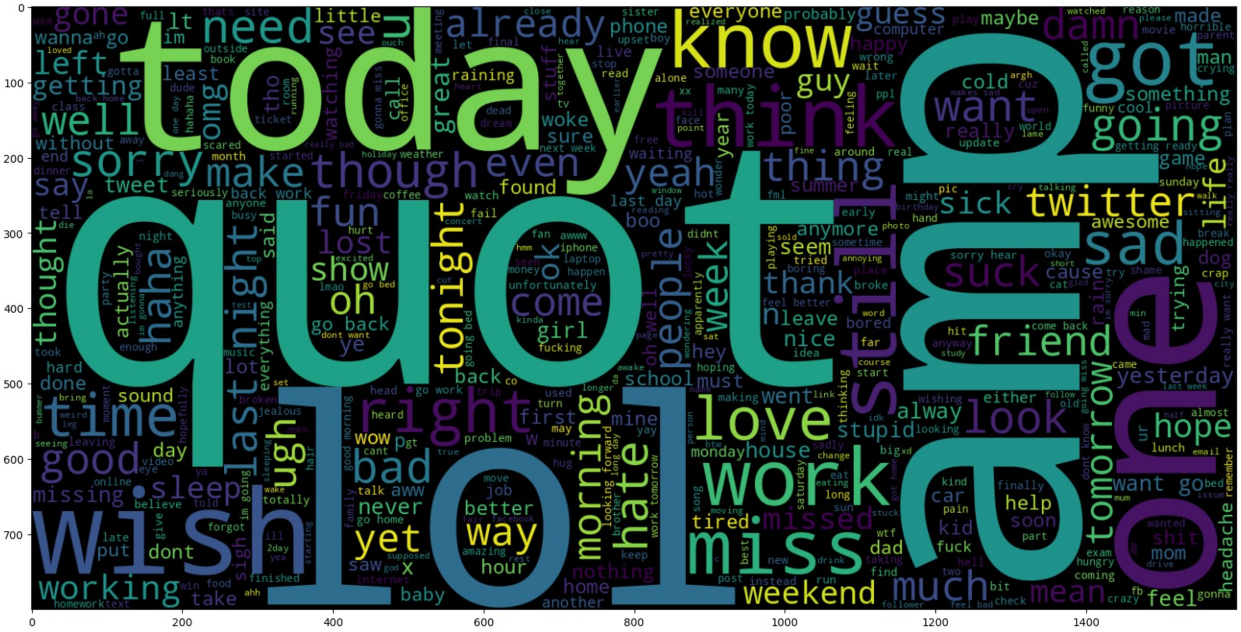
```
# text preprocessing

stop_words = stopwords.words('english')
stemmer = SnowballStemmer('english')

text_cleaning_re = "@\S+|https?:\S+|http?:\S|[^A-Za-z0-9]+"

def preprocess(text):
    text = re.sub(text_cleaning_re, ' ', str(text).lower()).strip()
    tokens = []
    for token in text.split():
        if token not in stop_words:
            tokens.append(token)
    return " ".join(tokens)

df['text'] = df['text'].map(preprocess)

from wordcloud import WordCloud

# positive texts
plt.figure(figsize = (20,20))
wc = WordCloud(max_words = 500 , width = 1600 , height =
800).generate(" ".join(df[df.sentiment == 'Positive'].text))
plt.imshow(wc , interpolation = 'bilinear')
```

```
<matplotlib.image.AxesImage at 0x7d14795ba6b0>
```



```
from wordcloud import WordCloud

# negative texts
plt.figure(figsize = (20,20))
wc = WordCloud(max_words = 500 , width = 1600 , height =
800).generate(" ".join(df[df.sentiment == 'Negative'].text))
plt.imshow(wc , interpolation = 'bilinear')
```

```
<matplotlib.image.AxesImage at 0x7d14795c1cf0>
```



```
# train test splitting

MAX_NB_WORDS = 100000
MAX_SEQUENCE_LENGTH = 30

train_data, test_data = train_test_split(df, test_size=0.2,
random_state=42)
print("Train Data size:", len(train_data))
print("Test Data size", len(test_data))
```

```
Train Data size: 1280000
Test Data size 320000
```

```
train_data.head()
```

```
        sentiment                                               text
1374558  Positive  ya quot like palm pre touchstone charger ready...
1389115  Positive          felt earthquake afternoon seems epicenter
1137831  Positive                          ruffles shirts like likey
790714   Negative  pretty bad night crappy morning fml buttface d...
1117911  Positive                                    yeah clear view
```

```
from tensorflow.keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_data.text)

word_index = tokenizer.word_index
vocab_size = len(tokenizer.word_index) + 1
print("Vocabulary Size :", vocab_size)
```

```
Vocabulary Size : 290419
```

```
from tensorflow.keras.preprocessing.sequence import pad_sequences

x_train = pad_sequences(tokenizer.texts_to_sequences(train_data.text),
                        maxlen = MAX_SEQUENCE_LENGTH)
x_test = pad_sequences(tokenizer.texts_to_sequences(test_data.text),
                       maxlen = MAX_SEQUENCE_LENGTH)

print("Training X Shape:",x_train.shape)
print("Testing X Shape:",x_test.shape)
```

```
Training X Shape: (1280000, 30)
Testing X Shape: (320000, 30)
```

```
labels = train_data.sentiment.unique().tolist()

encoder = LabelEncoder()
encoder.fit(train_data.sentiment.to_list())

y_train = encoder.transform(train_data.sentiment.to_list())
```

```
y_test = encoder.transform(test_data.sentiment.to_list())

y_train = y_train.reshape(-1,1)
y_test = y_test.reshape(-1,1)

print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
y_train shape: (1120000, 1)
y_test shape: (480000, 1)
```

```
# download pretrained GloVe word
!wget http://nlp.stanford.edu/data/glove.6B.zip
!unzip glove.6B.zip
```

```
/opt/conda/lib/python3.10/pty.py:89: RuntimeWarning: os.fork() was
called. os.fork() is incompatible with multithreaded code, and JAX is
multithreaded, so this will likely lead to a deadlock.
  pid, fd = os.forkpty()
```

```
--2024-03-31 09:15:00--  http://nlp.stanford.edu/data/glove.6B.zip
Resolving nlp.stanford.edu (nlp.stanford.edu)... 171.64.67.140
Connecting to nlp.stanford.edu (nlp.stanford.edu)|171.64.67.140|:80...
connected.
HTTP request sent, awaiting response... 302 Found
Location: https://nlp.stanford.edu/data/glove.6B.zip [following]
--2024-03-31 09:15:00--  https://nlp.stanford.edu/data/glove.6B.zip
Connecting to nlp.stanford.edu (nlp.stanford.edu)|
171.64.67.140|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://downloads.cs.stanford.edu/nlp/data/glove.6B.zip
[following]
--2024-03-31 09:15:01--
https://downloads.cs.stanford.edu/nlp/data/glove.6B.zip
Resolving downloads.cs.stanford.edu (downloads.cs.stanford.edu)...
171.64.64.22
Connecting to downloads.cs.stanford.edu (downloads.cs.stanford.edu)|
171.64.64.22|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 862182613 (822M) [application/zip]
Saving to: 'glove.6B.zip'

glove.6B.zip        100%[===================>] 822.24M  5.13MB/s    in
2m 42s

2024-03-31 09:17:43 (5.08 MB/s) - 'glove.6B.zip' saved
[862182613/862182613]

Archive:  glove.6B.zip
  inflating: glove.6B.50d.txt
  inflating: glove.6B.100d.txt
  inflating: glove.6B.200d.txt
  inflating: glove.6B.300d.txt
```

```
GLOVE_EMB = r"/kaggle/working/glove.6B.300d.txt"
EMBEDDING_DIM = 300
LR = 1e-3
BATCH_SIZE = 1024
EPOCHS = 10
MODEL_PATH = '.../output/kaggle/working/best_model.hdf5'

embeddings_index = {}
f = open(GLOVE_EMB)
for line in f:
    values = line.split() # splits each line into a list of values
    word = value = values[0] # extracts the first value as the word
itself
    coefs = np.asarray(values[1:], dtype='float32') # converts the
remaining values into a NumPy array of floating-point numbers,
representing the word's embedding vector
    embeddings_index[word] = coefs
f.close()

print('Found %s word vectors.' %len(embeddings_index))
```

```
Found 400000 word vectors.
```