```
# utilities
import re
import numpy as np
import pandas as pd
# plotting
import seaborn as sns
from wordcloud import WordCloud
import matplotlib.pyplot as plt
# nltk
from nltk.stem import WordNetLemmatizer
# sklearn
from sklearn.svm import LinearSVC
from sklearn.naive_bayes import BernoulliNB
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix, classification_report
```

```
! gdown --id 1kuzlzcpA-Vd27Qp7Ir7yxJ10bkcE-gUQ
```

```
/usr/local/lib/python3.7/dist-packages/gdown/cli.py:131: FutureWarning: Option `--id` wa
  category=FutureWarning,
Downloading...
From: https://drive.google.com/uc?id=1kuzlzcpA-Vd27Qp7Ir7yxJ10bkcE-gUQ
To: /content/training.1600000.processed.noemoticon.csv
100% 239M/239M [00:02<00:00, 108MB/s]
```

```
# Importing the dataset
DATASET_COLUMNS=['target','ids','date','flag','user','text']
DATASET_ENCODING = "ISO-8859-1"
df = pd.read_csv('training.1600000.processed.noemoticon.csv', encoding=DATASET_ENCODING, name
df.sample(5)
```

| | target | ids | date | flag | user | text |
|---|---|---|---|---|---|---|
| **1437510** | 4 | 2061182738 | Sat Jun 06 20:20:44 PDT 2009 | NO_QUERY | Stephenpj | Had a really nice Birthday Party today. Glad y... |
| **785594** | 0 | 2324460846 | Thu Jun 25 03:12:49 PDT 2009 | NO_QUERY | joelduggan | @acedtect To bad the Logitech MX1100 is left h... |
| | | | Mon Jun 15 | | | Damn it #masterchef not |

```
df.head()
```

| | target | ids | date | flag | user | |
|---|---|---|---|---|---|---|
| **0** | 0 | 1467810369 | Mon Apr 06 22:19:45 PDT 2009 | NO_QUERY | _TheSpecialOne_ | @switc |
| **1** | 0 | 1467810672 | Mon Apr 06 22:19:49 PDT 2009 | NO_QUERY | scotthamilton | is upse |
| **2** | 0 | 1467810917 | Mon Apr 06 22:19:53 PDT 2009 | NO_QUERY | mattycus | @Kenich |
| **3** | 0 | 1467811184 | Mon Apr 06 22:19:57 PDT 2009 | NO_QUERY | ElleCTF | my |
| **4** | 0 | 1467811193 | Mon Apr 06 22:19:57 PDT 2009 | NO_QUERY | Karoli | @nati |

```
df.columns
```

```
Index(['target', 'ids', 'date', 'flag', 'user', 'text'], dtype='object')
```

```
print('length of data is', len(df))
```

```
length of data is 1600000
```

```
df. shape
```

```
(1600000, 6)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1600000 entries, 0 to 1599999
Data columns (total 6 columns):
 #   Column  Non-Null Count    Dtype  
---  ------  --------------    -----  
 0   target  1600000 non-null  int64  
 1   ids     1600000 non-null  int64  
 2   date    1600000 non-null  object 
 3   flag    1600000 non-null  object 
 4   user    1600000 non-null  object 
 5   text    1600000 non-null  object 
dtypes: int64(2), object(4)
memory usage: 73.2+ MB
```

```
df.dtypes
```

```
target     int64
ids        int64
date      object
flag      object
user      object
text      object
dtype: object
```

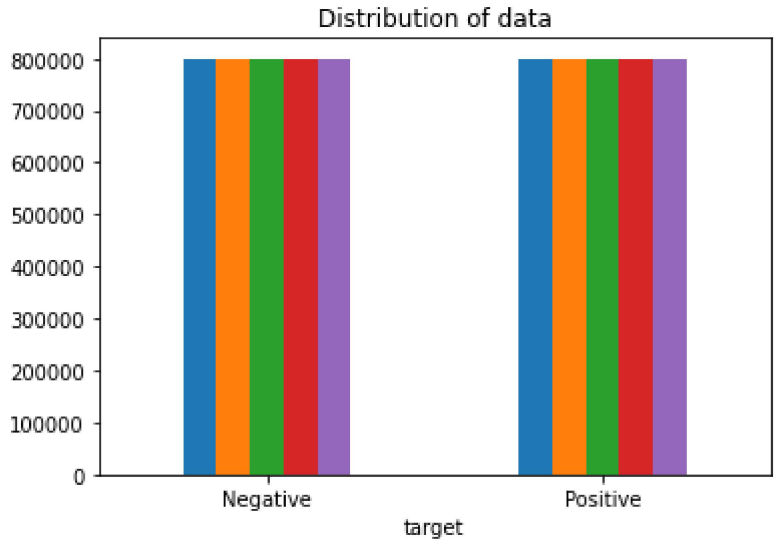
```
print('Count of columns in the data is:  ', len(df.columns))
print('Count of rows in the data is:  ', len(df))
```

```
Count of columns in the data is:   6
Count of rows in the data is:   1600000
```

```
np.sum(df.isnull().any(axis=1))
```

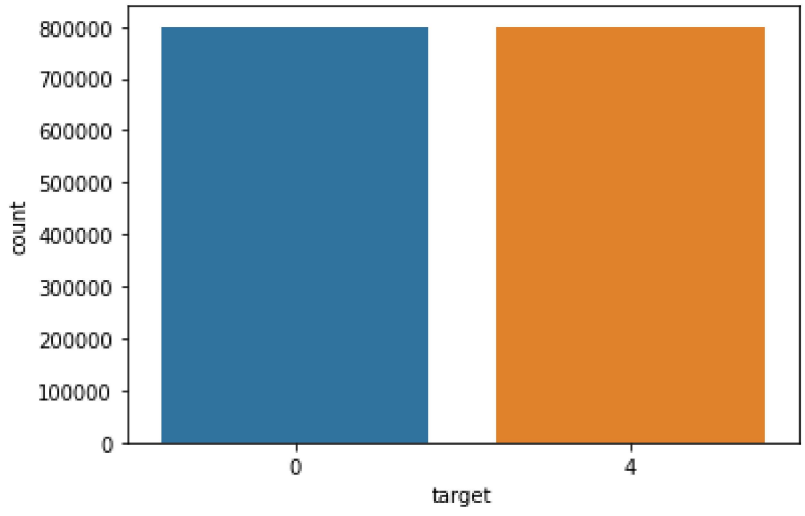
```
0
```

```
# Plotting the distribution for dataset.
ax = df.groupby('target').count().plot(kind='bar', title='Distribution of data',legend=False)
ax.set_xticklabels(['Negative','Positive'], rotation=0)
# Storing data in lists.
text, sentiment = list(df['text']), list(df['target'])
```




```
import seaborn as sns
sns.countplot(x='target', data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f93a3780c50>
```




```
data=df[['text','target']]
```

```
#Replacing the values to ease understanding. (Assigning 1 to Positive sentiment 4)

data['target'] = data['target'].replace(4,1)
data['target'].unique()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user
  This is separate from the ipykernel package so we can avoid doing imports until
array([0, 1])
```

```
data['target'].unique()
```

```
array([0, 1])
```

```
#Separating positive and negative tweets

data_pos = data[data['target'] == 1]
data_neg = data[data['target'] == 0]
```

```
#Combining positive and negative tweets

dataset = pd.concat([data_pos, data_neg])
```

```
#Making statement text in lower case

dataset['text']=dataset['text'].str.lower()
dataset['text'].tail()
```

```
799995    sick  spending my day laying in bed listening ...
799996                                  gmail is down?
799997                        rest in peace farrah! so sad
799998    @eric_urbane sounds like a rival is flagging y...
799999    has to resit exams over summer...  wishes he w...
Name: text, dtype: object
```

```
stopwordlist = ['a', 'about', 'above', 'after', 'again', 'ain', 'all', 'am', 'an',
             'and','any','are', 'as', 'at', 'be', 'because', 'been', 'before',
             'being', 'below', 'between','both', 'by', 'can', 'd', 'did', 'do',
             'does', 'doing', 'down', 'during', 'each','few', 'for', 'from',
             'further', 'had', 'has', 'have', 'having', 'he', 'her', 'here',
             'hers', 'herself', 'him', 'himself', 'his', 'how', 'i', 'if', 'in',
             'into','is', 'it', 'its', 'itself', 'just', 'll', 'm', 'ma',
             'me', 'more', 'most','my', 'myself', 'now', 'o', 'of', 'on', 'once',
```

```
              'only', 'or', 'other', 'our', 'ours','ourselves', 'out', 'own', 're','s', 'same'
              't', 'than', 'that', "thatll", 'the', 'their', 'theirs', 'them',
              'themselves', 'then', 'there', 'these', 'they', 'this', 'those',
              'through', 'to', 'too','under', 'until', 'up', 've', 'very', 'was',
              'we', 'were', 'what', 'when', 'where','which','while', 'who', 'whom',
              'why', 'will', 'with', 'won', 'y', 'you', "youd","youll", "youre",
              "youve", 'your', 'yours', 'yourself', 'yourselves']
```

```
import string
english_punctuations = string.punctuation
punctuations_list = english_punctuations
def cleaning_punctuations(text):
    translator = str.maketrans('', '', punctuations_list)
    return text.translate(translator)
dataset['text']= dataset['text'].apply(lambda x: cleaning_punctuations(x))
dataset['text'].tail()
```

```
     799995    sick  spending my day laying in bed listening ...
     799996                                      gmail is down
     799997                        rest in peace farrah so sad
     799998    ericurbane sounds like a rival is flagging you...
     799999    has to resit exams over summer  wishes he work...
     Name: text, dtype: object
```

```
#Cleaning and removing repeating characters

def cleaning_repeating_char(text):
    return re.sub(r'(.)1+', r'1', text)
dataset['text'] = dataset['text'].apply(lambda x: cleaning_repeating_char(x))
dataset['text'].tail()
```

```
     799995    sick  spending my day laying in bed listening ...
     799996                                      gmail is down
     799997                        rest in peace farrah so sad
     799998    ericurbane sounds like a rival is flagging you...
     799999    has to resit exams over summer  wishes he work...
     Name: text, dtype: object
```

```
#Cleaning and removing URL’s

def cleaning_URLs(data):
    return re.sub('((www.[^s]+)|(https?://[^s]+))',' ',data)
dataset['text'] = dataset['text'].apply(lambda x: cleaning_URLs(x))
dataset['text'].tail()
```

```
     799995    sick  spending my day laying in bed listening ...
     799996                                      gmail is down
     799997                        rest in peace farrah so sad
     799998    ericurbane sounds like a rival is flagging you...
```

```
799999    has to resit exams over summer  wishes he work...
Name: text, dtype: object
```

```
#Cleaning and removing Numeric numbers

def cleaning_numbers(data):
    return re.sub('[0-9]+', '', data)
dataset['text'] = dataset['text'].apply(lambda x: cleaning_numbers(x))
dataset['text'].tail()
```

```
799995    sick  spending my day laying in bed listening ...
799996                                      gmail is down
799997                          rest in peace farrah so sad
799998    ericurbane sounds like a rival is flagging you...
799999    has to resit exams over summer  wishes he work...
Name: text, dtype: object
```

```
#Applying Stemming

import nltk
st = nltk.PorterStemmer()
def stemming_on_text(data):
    text = [st.stem(word) for word in data]
    return data
dataset['text']= dataset['text'].apply(lambda x: stemming_on_text(x))
dataset['text'].head()
```

```
800000             i love healthuandpets u guys r the best
800001    im meeting up with one of my besties tonight c...
800002    darealsunisakim thanks for the twitter add sun...
800003    being sick can be really cheap when it hurts t...
800004        lovesbrooklyn he has that effect on everyone
Name: text, dtype: object
```

```
#Applying Lemmatizer

import nltk
nltk.download('wordnet')
nltk.download('omw-1.4')
lm = nltk.WordNetLemmatizer()
def lemmatizer_on_text(data):
    text = [lm.lemmatize(word) for word in data]
    return data
dataset['text'] = dataset['text'].apply(lambda x: lemmatizer_on_text(x))
dataset['text'].head()
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
800000             i love healthuandpets u guys r the best
800001    im meeting up with one of my besties tonight c...
800002    darealsunisakim thanks for the twitter add sun...
800003    being sick can be really cheap when it hurts t...
```

```
800004        lovesbrooklyn he has that effect on everyone
Name: text, dtype: object
```

```
!pip install transformers
import transformers
from transformers import BertTokenizer, TFBertModel

from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

import warnings
warnings.filterwarnings("ignore")
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/pub
Collecting transformers
  Downloading transformers-4.22.2-py3-none-any.whl (4.9 MB)
     |████████████████████████████████| 4.9 MB 2.1 MB/s
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-packages (from
Collecting huggingface-hub<1.0,>=0.9.0
  Downloading huggingface_hub-0.10.0-py3-none-any.whl (163 kB)
     |████████████████████████████████| 163 kB 61.8 MB/s
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packa
Collecting tokenizers!=0.11.3,<0.13,>=0.11.1
  Downloading tokenizers-0.12.1-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.wh
     |████████████████████████████████| 6.6 MB 38.1 MB/s
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.7/di
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
Installing collected packages: tokenizers, huggingface-hub, transformers
Successfully installed huggingface-hub-0.10.0 tokenizers-0.12.1 transformers-4.22.2
```

```
tokenizer = BertTokenizer.from_pretrained('bert-large-uncased')
tokenizer
```

```
Downloading: 100%  232k/232k [00:00<00:00, 293kB/s]

Downloading: 100%  28.0/28.0 [00:00<00:00, 777B/s]

Downloading: 100%  571/571 [00:00<00:00, 17.3kB/s]

PreTrainedTokenizer(name_or_path='bert-large-uncased', vocab_size=30522,
model_max_len=512, is_fast=False, padding_side='right', truncation_side='right',
special_tokens={'unk_token': '[UNK]', 'sep_token': '[SEP]', 'pad_token': '[PAD]'
```

```
bert_model = TFBertModel.from_pretrained('bert-base-uncased')
```

```
Downloading: 100%  570/570 [00:00<00:00, 15.0kB/s]

Downloading: 100%  536M/536M [00:16<00:00, 21.5MB/s]

Some layers from the model checkpoint at bert-base-uncased were not used when initializi
- This IS expected if you are initializing TFBertModel from the checkpoint of a model tr
- This IS NOT expected if you are initializing TFBertModel from the checkpoint of a mode
All the layers of TFBertModel were initialized from the model checkpoint at bert-base-ur
If your task is similar to the task the model of the checkpoint was trained on, you can
```

```
X = data.text
y = data.target


X_train,X_test,y_train,y_test = train_test_split(X, y, test_size=0.2, random_state = 0)


def encode(text, maxlen):
  input_ids=[]
  attention_masks=[]

  for row in text:
    encoded = tokenizer.encode_plus(
        row,
        add_special_tokens=True,
        max_length=maxlen,
        pad_to_max_length=True,
        return_attention_mask=True,
    )
    input_ids.append(encoded['input_ids'])
    attention_masks.append(encoded['attention_mask'])

  return np.array(input_ids),np.array(attention_masks)



X_train_input_ids, X_train_attention_masks = encode(X_train.values, maxlen=68)
X_test_input_ids, X_test_attention_masks = encode(X_test.values, maxlen=68)
```

```
Truncation was not explicitly activated but `max_length` is provided a specific value, p
```

```
def build_model(bert_model):
   input_word_ids = tf.keras.Input(shape=(68,),dtype='int32')
   attention_masks = tf.keras.Input(shape=(68,),dtype='int32')

   sequence_output = bert_model([input_word_ids,attention_masks])
   output = sequence_output[1]
   output = tf.keras.layers.Dense(3200,activation='relu')(output)
```

```
    output = tf.keras.layers.Dropout(0.2)(output)
    output = tf.keras.layers.Dense(1,activation='sigmoid')(output)

    model = tf.keras.models.Model(inputs = [input_word_ids,attention_masks], outputs = output)
    model.compile(Adam(lr=1e-5), loss='binary_crossentropy', metrics=['accuracy'])

    return model


import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Dense, Input
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.models import Model


model = build_model(bert_model)
model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 68)]         0           []

 input_2 (InputLayer)           [(None, 68)]         0           []

 tf_bert_model (TFBertModel)    TFBaseModelOutputWi  109482240   ['input_1[0][0]',
                                thPoolingAndCrossAt               'input_2[0][0]']
                                tentions(last_hidde
                                n_state=(None, 68,
                                768),
                                 pooler_output=(Non
                                e, 768),
                                 past_key_values=No
                                ne, hidden_states=N
                                one, attentions=Non
                                e, cross_attentions
                                =None)

 dense (Dense)                  (None, 32)           24608       ['tf_bert_model[0][1]']

 dropout_37 (Dropout)           (None, 32)           0           ['dense[0][0]']

 dense_1 (Dense)                (None, 1)            33          ['dropout_37[0][0]']

==================================================================================================
Total params: 109,506,881
Trainable params: 109,506,881
Non-trainable params: 0
__________________________________________________________________________________________________
```

```
class_weight = {0: 1, 1: 8}

history = model.fit(
    [X_train_input_ids, X_train_attention_masks],
    y_train,
    batch_size=3200,
    epochs=1,
    validation_data=([X_test_input_ids, X_test_attention_masks], y_test))
```

```
40000/40000 [==============================] - 11171s 279ms/step - loss: 0.3322 - accura
```

```
loss, accuracy = model.evaluate([X_test_input_ids, X_test_attention_masks], y_test)
print('Test accuracy :', accuracy)
```

```
10000/10000 [==============================] - 851s 85ms/step - loss: 0.3022 - accuracy
Test accuracy : 0.871681272983551
```

```
#save model
model.save_weights('bert_model')
```