In [1]:

```
import pickle
import re
import numpy as np
import pandas as pd
from tqdm import tqdm
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\91874\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[1]:

```
True
```

In [2]:

```
def preprocess_tweet(text):
    text = re.sub('<[^>]*>', '', text)
    emoticons = re.findall('(?::|;|=)(?:-)?(?:\)|\(|D|P)', text)
    lowercase_text = re.sub('[\W]+', ' ', text.lower())
    text = lowercase_text+' '.join(emoticons).replace('-', '')
    return text
```

In [3]:

```
tqdm.pandas()
df = pd.read_csv('Data.csv')
df['text'] = df['text'].progress_apply(preprocess_tweet)
```

```
D:\Users\91874\anaconda3\lib\site-packages\tqdm\std.py:668: FutureWarning: T
he Panel class is removed from pandas. Accessing it from the top-level names
pace will also be removed in the next version
  from pandas import Panel
100%|██████████| 115/115 [00:00<00:00, 23269.09it/s]
```

In [4]:

```
from nltk.stem.porter import PorterStemmer
porter = PorterStemmer()
def tokenizer_porter(text):
    return [porter.stem(word) for word in text.split()]
```

In [5]:

```
from nltk.corpus import stopwords
stop = stopwords.words('english')
```

In [6]:

```
[w for w in tokenizer_porter('a runner likes running and runs a lot') if w not in stop]
```

Out[6]:

```
['runner', 'like', 'run', 'run', 'lot']
```

In [7]:

```
def tokenizer(text):
    text = re.sub('<[^>]*>', '', text)
    emoticons = re.findall('(?::|;|=)(?:-)?(?:\(|D|P)',text.lower())
    text = re.sub('[\W]+', ' ', text.lower())
    text += ' '.join(emoticons).replace('-', '')
    tokenized = [w for w in tokenizer_porter(text) if w not in stop]
    return tokenized
```

In [8]:

```
from sklearn.feature_extraction.text import HashingVectorizer
vect = HashingVectorizer(decode_error='ignore', n_features=2**21,
                         preprocessor=None,tokenizer=tokenizer)
```

In [9]:

```
from sklearn.linear_model import SGDClassifier
clf = SGDClassifier(loss='log', random_state=1)
```

In [11]:

```
X = df["text"].to_list()
y = df['labels']
```

In [12]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,
                                                 y,
                                                 test_size=0.20,
                                                 random_state=0)
```

In [13]:

```
X_train = vect.transform(X_train)
X_test = vect.transform(X_test)
```

In [14]:

```
classes = np.array([0, 1])
clf.partial_fit(X_train, y_train,classes=classes)
```

Out[14]:

```
SGDClassifier(loss='log', random_state=1)
```

In [15]:

```
print('Accuracy: %.3f' % clf.score(X_test, y_test))
```

```
Accuracy: 0.783
```

In [16]:

```
label = {0:'negative', 1:'positive'}
example = ["I'll kill myself am tired of living depressed and alone"]
X = vect.transform(example)
print('Prediction: %s\nProbability: %.2f%%'
      %(label[clf.predict(X)[0]],np.max(clf.predict_proba(X))*100))
```

```
Prediction: positive
Probability: 95.93%
```

In [ ]:

```

```

In [ ]:

```

```