EMOTION CLASSIFIER

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

Read File

```
file = open('emotionClassifier.txt','r')
data = file.readlines()
```

```
data
```

```
['i didnt feel humiliated;sadness\n',
 'i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake;sadness\n',
 'im grabbing a minute to post i feel greedy wrong;anger\n',
 'i am ever feeling nostalgic about the fireplace i will know that it is still on the property;love\n',
 'i am feeling grouchy;anger\n',
 'ive been feeling a little burdened lately wasnt sure why that was;sadness\n',
 'ive been taking or milligrams or times recommended amount and ive fallen asleep a lot faster but i also feel like so funny;surprise\n',
 'i feel as confused about life as a teenager or as jaded as a year old man;fear\n',
 'i have been with petronas for years i feel that petronas has performed well and made a huge profit;joy\n',
 'i feel romantic too;love\n',
 'i feel like i have to make the suffering i m seeing mean something;sadness\n',
 'i do feel that running is a divine experience and that i can expect to have some type of spiritual encounter;joy\n',
 'i think it s the easiest time of year to feel dissatisfied;anger\n',
 'i feel low energy i m just thirsty;sadness\n',
 'i have immense sympathy with the general point but as a possible proto writer trying to find time to write in the corners of life and with no sign of an agent let alone a publishing contract this feels a little precious;joy\n',
 'i do not feel reassured anxiety is on each side;joy\n',
 'i didnt really feel that embarrassed;sadness\n',
 'i feel pretty pathetic most of the time;sadness\n',
 'i started feeling sentimental about dolls i had as a child and so began a collection of vintage barbie dolls from the sixties;sadness\n',
 'i now feel compromised and skeptical of the value of every unit of work i put in;fear\n',
 'i feel irritated and rejected without anyone doing anything or saying anything;anger\n',
 'i am feeling completely overwhelmed i have two strategies that help me to feel grounded pour my heart out in my journal in the form of a letter to god and then end with a list of five things i am most grateful for;fear\n',
 'i have the feeling she was amused and delighted;joy\n',
 'i was able to help chai lifeline with your support and encouragement is a great feeling and i am so glad you were able to help me;joy\n',
 'i already feel like i fucked up though because i dont usually eat at all in the morning;anger\n',
 'i still love my so and wish the best for him i can no longer tolerate the effect that bm has on our lives and the fact that is has turned my so into a bitter angry person who is not always particularly kind to the people around him when he is feeling stressed;sadness\n',
 'i feel so inhibited in someone elses kitchen like im painting on someone elses picture;sadness\n',
 'i become overwhelmed and feel defeated;sadness\n',
 'i feel kinda appalled that she feels like she needs to explain in wide and lenghth her body measures etc pp;anger\n',
 'i feel more superior dead chicken or grieving child;joy\n',
 'i get giddy over feeling elegant in a perfectly fitted pencil skirt;joy\n',
 'i remember feeling acutely distressed for a few days;fear\n',
 'i have seen heard and read over the past couple of days i am left feeling impressed by more than a few companies;surprise\n',
 'i climbed the hill feeling frustrated that id pretty much paced entirely wrong for this course and that a factor that has never ever hampered me had made such a dent in the day;anger\n',
 'i can t imagine a real life scenario where i would be emotionally connected enough with someone to feel totally accepted and safe where it it morally acceptable for me to have close and prolonged physical contact and where sex won t be expected subsequently;joy\n',
 'i am not sure what would make me feel content if anything;joy\n',
 'i have been feeling the need to be creative;joy\n',
 'i do however want you to know that if something someone is causing you to feel less then your splendid self step away from them;joy\n',
 'i feel a bit rude writing to an elderly gentleman to ask for gifts because i feel a bit greedy but what is
```

```
        christmas about if not mild greed;anger\n',
         'i need you i need someone i need to be protected and feel safe i am small now i find myself in a season of no
        words;joy\n',
         'i plan to share my everyday life stories traveling adventures inspirations and handmade creations with you and
```

```
c1 = []
c2 = []
```

```
for i in data:
  s = i.split(';')
  c1.append(s[0])
  c2.append(s[1])
```

```
c1
```

```
...]
```

```
c2
```

```
'joy\n',
'sadness\n',
'anger\n',
'joy\n',
'joy\n',
'sadness\n',
'joy\n',
'love\n',
'joy\n',
'joy\n',
'joy\n',
'anger\n',
'fear\n',
'sadness\n',
'joy\n',
'sadness\n',
'sadness\n',
'love\n',
'anger\n',
'joy\n',
'love\n',
'sadness\n',
'joy\n',
'fear\n',
'sadness\n',
'joy\n',
'joy\n',
'sadness\n',
'sadness\n',
'love\n',
'joy\n',
'sadness\n',
'sadness\n',
'sadness\n',
'sadness\n',
'anger\n',
'sadness\n',
'joy\n',
'sadness\n',
'joy\n',
'sadness\n',
'love\n',
'anger\n',
'joy\n',
'fear\n',
'sadness\n',
'sadness\n',
'joy\n',
'joy\n',
'anger\n',
'fear\n',
'love\n',
'joy\n',
'sadness\n',
'anger\n',
'sadness\n',
'joy\n',
...]
```

Dataset

```
w = {'Statement' : c1 ,'Emotion' : c2}
```

```
df = pd.DataFrame(w)
```

```
df
```

	Statement	Emotion
0	i didnt feel humiliated	sadness\n
1	i can go from feeling so hopeless to so damned...	sadness\n
2	im grabbing a minute to post i feel greedy wrong	anger\n
3	i am ever feeling nostalgic about the fireplac...	love\n
4	i am feeling grouchy	anger\n
...	...	...
15995	i just had a very brief time in the beanbag an...	sadness\n
15996	i am now turning and i feel pathetic that i am...	sadness\n
15997	i feel strong and good overall	joy\n
15998	i feel like this was such a rude comment and i...	anger\n
15999	i know a lot but i feel so stupid because i ca...	sadness\n

16000 rows × 2 columns

```
df['Emotion'] = df['Emotion'].str.replace('\n','')
```

```
df
```

	Statement	Emotion
0	i didnt feel humiliated	sadness
1	i can go from feeling so hopeless to so damned...	sadness
2	im grabbing a minute to post i feel greedy wrong	anger
3	i am ever feeling nostalgic about the fireplac...	love
4	i am feeling grouchy	anger
...	...	...
15995	i just had a very brief time in the beanbag an...	sadness
15996	i am now turning and i feel pathetic that i am...	sadness
15997	i feel strong and good overall	joy
15998	i feel like this was such a rude comment and i...	anger
15999	i know a lot but i feel so stupid because i ca...	sadness

16000 rows × 2 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16000 entries, 0 to 15999
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   Statement  16000 non-null  object
 1   Emotion    16000 non-null  object
dtypes: object(2)
memory usage: 250.1+ KB
```

```
df
```

	Statement	Emotion
0	i didnt feel humiliated	sadness
1	i can go from feeling so hopeless to so damned...	sadness
2	im grabbing a minute to post i feel greedy wrong	anger
3	i am ever feeling nostalgic about the fireplac...	love
4	i am feeling grouchy	anger
...	...	...
15995	i just had a very brief time in the beanbag an...	sadness
15996	i am now turning and i feel pathetic that i am...	sadness
15997	i feel strong and good overall	joy
15998	i feel like this was such a rude comment and i...	anger

Importing Nlp modules for text preprocessing

16000 rows × 2 columns

```
import nltk
nltk.download("punkt")
nltk.download("wordnet")
nltk.download("stopwords")

from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
def cleantext(text):
  tokens = word_tokenize(text.lower())
  ftoken = [t for t in tokens if(t.isalpha())]
  stop = stopwords.words("english")
  ctoken = [t for t in ftoken if(t not in stop)]
  lemma = WordNetLemmatizer()
  ltoken = [lemma.lemmatize(t) for t in ctoken]
  return " ".join(ltoken)


df["cleantext"]=df["Statement"].apply(cleantext)


df
```

	Statement	Emotion	cleantext
0	i didnt feel humiliated	sadness	didnt feel humiliated
1	i can go from feeling so hopeless to so damned...	sadness	go feeling hopeless damned hopeful around some...

Data Separation

```
x = df.cleantext
y = df.Emotion
```

4	i am feeling grouchy	anger	feeling grouchy

Creating new column of sentence count

```
s = []
for i in df['cleantext']:
  s.append(len(word_tokenize(i)))


df['Sentence'] = s
```

```
df.head()
```

	Statement	Emotion	cleantext	Sentence
0	i didnt feel humiliated	sadness	didnt feel humiliated	3
1	i can go from feeling so hopeless to so damned...	sadness	go feeling hopeless damned hopeful around some...	9
2	im grabbing a minute to post i feel greedy wrong	anger	im grabbing minute post feel greedy wrong	7
3	i am ever feeling nostalgic about the fireplac...	love	ever feeling nostalgic fireplace know still pr...	7

```
max(df.Sentence)
```

```
35
```

```
np.quantile(s, 0.95)
```

```
20.0
```

```
max_len = np.quantile(s, 0.95)
```

Tokenizer Class

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


tok = Tokenizer(char_level=False, split=" ")

tok.fit_on_texts(x)
tok.index_word
```

```
 774: 'company',
 775: 'suppose',
 776: 'seriously',
 777: 'late',
 778: 'utterly',
 779: 'cut',
 780: 'attention',
 781: 'spirit',
 782: 'community',
 783: 'challenge',
 784: 'lie',
 785: 'appreciative',
 786: 'possible',
 787: 'sentimental',
 788: 'skeptical',
 789: 'strongly',
 790: 'worry',
 791: 'finished',
 792: 'view',
 793: 'leg',
 794: 'water',
 795: 'across',
 796: 'envious',
 797: 'tragic',
 798: 'color',
 799: 'damn',
 800: 'brain',
 801: 'news',
 802: 'amused',
 803: 'heard',
 804: 'tree',
 805: 'dad',
 806: 'perhaps',
 807: 'fit',
 808: 'voice',
 809: 'tomorrow',
 810: 'quickly',
 811: 'known',
 812: 'wait',
 813: 'spending',
 814: 'recently',
 815: 'connection',
 816: 'finish',
 817: 'top',
 818: 'film',
 819: 'fucking',
 820: 'expect',
 821: 'season',
 822: 'unpleasant',
 823: 'theyre'
```

Length of vocabulary

```
vocab_len = len(tok.index_word)
vocab_len
```

```
13476
```

Sequence of text

```
seq = tok.texts_to_sequences(x)
seq
```

```
 [24, 105, 133, 305, 1, 364, 62, 228, 801],
 [1, 584, 3, 7111, 5006],
 [3, 3, 2, 308],
 [166, 124, 744, 89, 335, 1, 3, 458],
 [2629, 2922, 2838, 47, 5007, 709, 1, 262, 1980],
 [7, 1, 7112, 537, 966, 7113],
 [1, 10, 60, 478, 10, 501, 10, 75, 1852, 181, 18, 873, 22],
 [335, 784, 4, 10, 752, 233, 1, 583, 193, 111, 195, 209, 2630, 742, 11],
 [7114, 7115, 7116, 5008, 194, 3958, 1359, 165, 1, 670, 305, 485, 181, 12],
 [20, 944, 133, 33, 679, 944, 4, 2, 6, 61, 13, 38, 246, 83, 98, 5009],
 [259, 756, 7117],
 [439,
  196,
  464,
  1593,
  1493,
  1522,
  1593,
  706,
  1593,
  5,
  67,
  111,
  113,
  813,
  11,
  1,
  470,
  7118],
 [992,
  1483,
  1482,
  131,
  3959,
  3960,
  1053,
  7119,
  992,
  3306,
  121,
  1,
  649,
  172,
  43],
 [261, 80, 2, 348],
 [1, 3, 62, 14, 675, 80, 17, 34, 411, 53, 492],
 [8, 70, 7120, 44, 5010, 2, 955],
 [2, 34, 424, 450],
 [94, 2, 6, 501, 2126, 845, 1781, 949, 452, 1574, 1319, 22, 1765],
 ...]
```

Padding

```
seqmat = sequence.pad_sequences(seq, maxlen= int(max_len))
seqmat
```

```
array([[   0,    0,    0, ...,   51,    1,  565],
       [   0,    0,    0, ...,   58,  172, 1139],
       [   0,    0,    0, ...,    1,  382,  321],
       ...,
       [   0,    0,    0, ...,  226,   33, 1206],
       [   0,    0,    0, ...,  467,    4,  236],
       [   0,    0,    0, ...,    1,  178, 3301]], dtype=int32)
```

```
vocab_len
```

```
13476
```

Label Encoding

```
from sklearn.preprocessing import LabelEncoder
```

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

Data Splitting

```
from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest = train_test_split(seqmat,y,test_size=0.3,random_state=1)
```

Neccessary Classes for Rnn model buliding

```
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense,SimpleRNN,Embedding,Dropout
```

Creating rnn model

```
rnn = Sequential()
```

Adding Embedding layer

```
rnn.add(Embedding(vocab_len+1,400, input_length=int(max_len), mask_zero=True))
```

RNN

```
rnn.add(SimpleRNN(units=32, activation="tanh"))
rnn.add(Dropout(0.2))
```

Hidden layer

```
rnn.add(Dense(units=32, activation="relu"))
rnn.add(Dropout(0.2))

rnn.add(Dense(units=10, activation="softmax"))

rnn.compile(optimizer="adam", loss="sparse_categorical_crossentropy",metrics=['accuracy'])
```

Fitting training data

```
rnn.fit(xtrain, ytrain, batch_size=70, epochs=15)
```

```
Epoch 1/15
160/160 [==============================] - 14s 73ms/step - loss: 1.6664 - accuracy: 0.3602
Epoch 2/15
160/160 [==============================] - 7s 43ms/step - loss: 0.7761 - accuracy: 0.7361
Epoch 3/15
160/160 [==============================] - 8s 48ms/step - loss: 0.3371 - accuracy: 0.8943
Epoch 4/15
160/160 [==============================] - 7s 46ms/step - loss: 0.1651 - accuracy: 0.9507
Epoch 5/15
160/160 [==============================] - 7s 44ms/step - loss: 0.1044 - accuracy: 0.9691
Epoch 6/15
160/160 [==============================] - 7s 47ms/step - loss: 0.0789 - accuracy: 0.9772
Epoch 7/15
160/160 [==============================] - 7s 43ms/step - loss: 0.0652 - accuracy: 0.9807
```

```
Epoch 8/15
160/160 [==============================] - 7s 46ms/step - loss: 0.0501 - accuracy: 0.9866
Epoch 9/15
160/160 [==============================] - 7s 45ms/step - loss: 0.0473 - accuracy: 0.9859
Epoch 10/15
160/160 [==============================] - 7s 44ms/step - loss: 0.0510 - accuracy: 0.9854
Epoch 11/15
160/160 [==============================] - 7s 46ms/step - loss: 0.0530 - accuracy: 0.9853
Epoch 12/15
160/160 [==============================] - 7s 46ms/step - loss: 0.0337 - accuracy: 0.9906
Epoch 13/15
160/160 [==============================] - 7s 45ms/step - loss: 0.0306 - accuracy: 0.9913
Epoch 14/15
160/160 [==============================] - 8s 47ms/step - loss: 0.0235 - accuracy: 0.9930
Epoch 15/15
160/160 [==============================] - 7s 43ms/step - loss: 0.0179 - accuracy: 0.9948
<keras.callbacks.History at 0x7f879813ec20>
```

Prediction

```
ypred = rnn.predict(xtest)
```

```
150/150 [==============================] - 1s 2ms/step
```

```
ypred
```

```
array([[2.09644233e-04, 1.59265808e-04, 9.62256551e-01, ...,
        1.72385649e-06, 2.08368147e-06, 2.48952051e-06],
       [5.60573459e-01, 3.71696160e-06, 4.32633251e-01, ...,
        1.65433590e-07, 1.62220264e-08, 3.33736736e-08],
       [2.41624111e-05, 1.13915057e-05, 8.51128561e-07, ...,
        3.00974690e-10, 8.35495475e-12, 4.23869342e-11],
       ...,
       [6.43403328e-04, 7.68921757e-03, 1.02396743e-05, ...,
        2.44416628e-06, 1.78035941e-06, 1.38450264e-06],
       [2.14815799e-07, 1.85985520e-08, 9.99998271e-01, ...,
        4.31119767e-12, 1.08810496e-13, 1.75216775e-12],
       [9.99904573e-01, 8.19484558e-05, 8.95880839e-06, ...,
        2.68467426e-09, 4.77047235e-10, 8.75070905e-10]], dtype=float32)
```

```
yp = ypred.argmax(axis=1)
```

```
from sklearn.metrics import classification_report
```

```
print(classification_report(ytest,yp))
```

```
                  precision    recall  f1-score   support

               0       0.83      0.80      0.82       664
               1       0.77      0.76      0.77       549
               2       0.85      0.88      0.86      1646
               3       0.72      0.67      0.69       403
               4       0.86      0.88      0.87      1363
               5       0.76      0.59      0.66       175

        accuracy                           0.83      4800
       macro avg       0.80      0.76      0.78      4800
    weighted avg       0.83      0.83      0.83      4800
```

```
# Accuracy of dataset is 83%
```