HATE SPEECH CLASSIFICATION(LSTM)

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import re
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
import string
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from keras.models import Model
from keras.layers import LSTM, Activation, Dense, Dropout, Input, Embedding, SpatialDropout1D
from keras.optimizers import RMSprop
from keras.preprocessing.text import Tokenizer
from keras.preprocessing import sequence
from keras.utils import to_categorical
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.models import Sequential
from keras.utils import pad_sequences
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.metrics import confusion_matrix
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.max_colwidth', 255)
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
imbalanced_data=pd.read_csv('/content/imbalanced_data.csv')
```

```
imbalanced_data.head()
```

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
1	2	0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in urð□□±!!! ð□□□ð□□□ð□□□ð□□□ð□□¦ð□□¦ð□□¦
4	5	0	factsguide: society now #motivation

```
# EDA
```

```
sns.countplot(x='label',data =imbalanced_data)
```

```
<Axes: xlabel='label', ylabel='count'>
```

30000
25000
20000

```
# From above grpah we can observe that classes are imbalanced.we will fix it later
```

```
# Checking the shape of the data
imbalanced_data.shape
```

```
(31962, 3)
```

```
# Cheking if null values are present in the dataset or not.
imbalanced_data.isnull().sum()
```

```
id       0
label    0
tweet    0
dtype: int64
```

```
# Let's drop the 'id' column as it is not required.
imbalanced_data.drop('id',axis=1, inplace =True)
```

```
imbalanced_data.head()
```

	label	tweet
0	0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
1	0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
2	0	bihday your majesty
3	0	#model i love u take with u all the time in urð□□±!!! ð□□□ð□□□ð□□□ð□□□ð□□¦ð□□¦ð□□¦
4	0	factsguide: society now #motivation

```
# Let's load another dataset similar to our dataset, since we want to fix the imbalance data.
raw_data = pd.read_csv("/content/raw_data.csv")
```

```
raw_data.head()
```

	Unnamed: 0	count	hate_speech	offensive_language	neither	class	tweet
0	0	3	0	0	3	2	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. &amp; as a man you should always take the trash out...
1	1	3	0	3	0	1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!
2	2	3	0	3	0	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confused as shit
3	3	3	0	2	1	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny
							!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me

```
raw_data.shape
```

```
(24783, 7)
```

```
raw_data.isnull().sum()
```

```
Unnamed: 0            0
count                 0
hate_speech           0
offensive_language    0
neither               0
class                 0
tweet                 0
dtype: int64
```

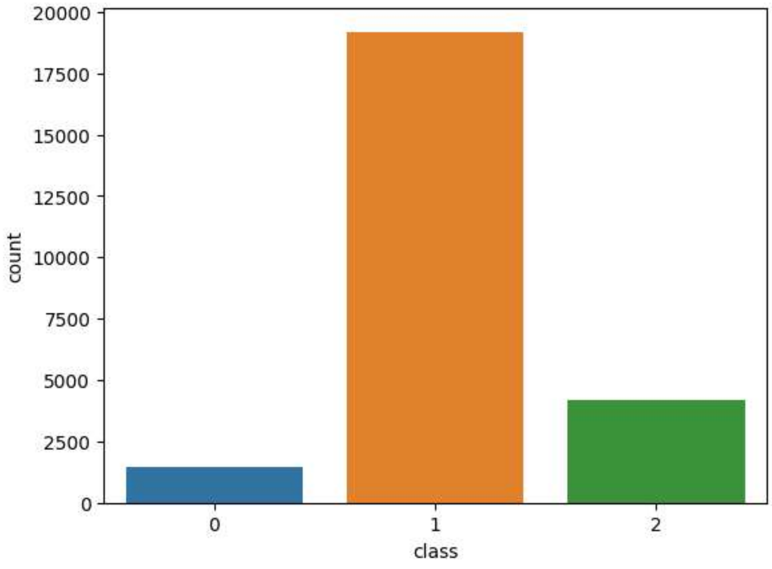
```
# Let's drop the columns which are not required for us.
raw_data.drop(['Unnamed: 0','count','hate_speech','offensive_language','neither'],axis=1,inplace =True)
```

```
# Let's check for the unique values in the dataset
raw_data['class'].unique()
```

```
array([2, 1, 0])
```

```
# Plotting the countplot for our new dataset
sns.countplot(x='class',data =raw_data)
```

```
<Axes: xlabel='class', ylabel='count'>
```



```
#above plot observation:
#  class 0: hate
# class 1: abusive
# class 2: no hate
# Let's copy the valus of the class 1 into class 0
```

```
raw_data[raw_data['class']==0]['class']=1
```

```
<ipython-input-19-55e395ab275c>:1: 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_guide/indexing.html#returning-a-view-versus-a-c
  raw_data[raw_data['class']==0]['class']=1
```

```
raw_data.head()
```

	class	tweet
0	2	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. &amp; as a man you should always take the trash out...
1	1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!
2	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confused as shit
3	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny
4	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told it to ya &#57361;

```
raw_data['class'].unique()
```

```
array([2, 1, 0])
```

```
# Let's check the values in the class 0
raw_data[raw_data['class']==0]
```

	class	tweet
85	0	"@Blackman38Tide: @WhaleLookyHere @HowdyDowdy11 queer" gaywad
89	0	"@CB_Baby24: @white_thunduh alsarabsss" hes a beaner smh you can tell hes a mexican

```
# replace the value of 0 to 1
raw_data["class"].replace({0:1},inplace=True)
```

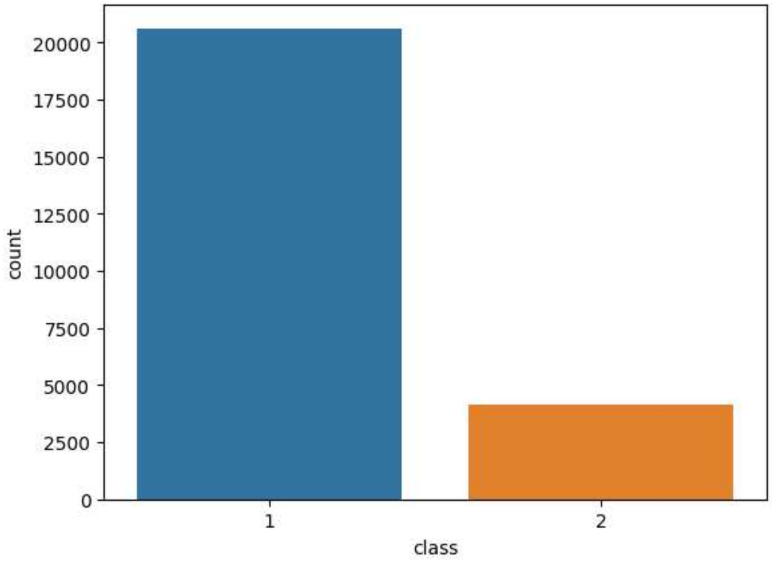
184 0 "@MarkRoundtreeJr: LMFAOOOO I HATE BLACK PEOPLE https://t.co/RNvD2nLCDR" This is why there's black people and

```
raw_data["class"].unique()
```

```
array([2, 1])
```

```
sns.countplot(x="class",data= raw_data)
```

```
<Axes: xlabel='class', ylabel='count'>
```

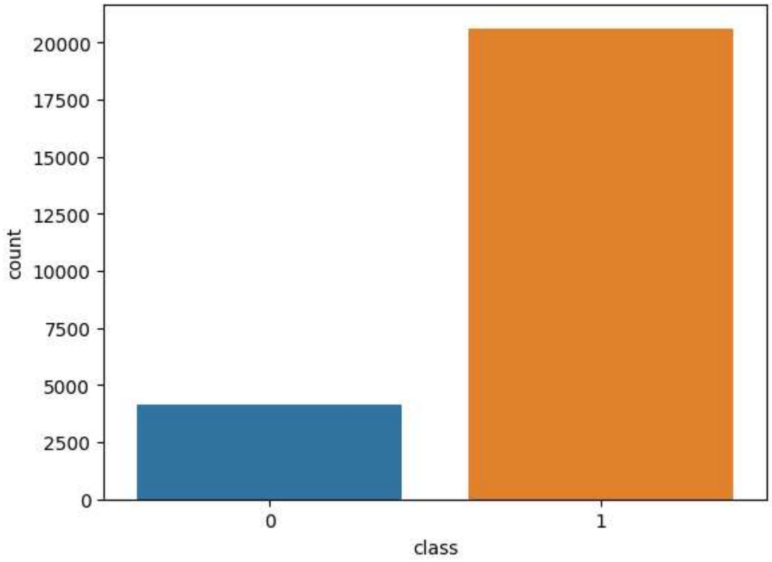


```
# Let's replace the value of 2 to 0.
raw_data["class"].replace({2:0}, inplace = True)
```

"Why people think gay marriage is okay is beyond me. Sorry I don't want my future son seeing 2 fags walking down the street

```
sns.countplot(x='class',data=raw_data)
```

```
<Axes: xlabel='class', ylabel='count'>
```



```
# Let's change the name of the 'class' to label
raw_data.rename(columns={'class':'label'},inplace =True)
```

#RebelScience ......is using an ACTUAL WOMAN as a genetic engineering lab for "all natural clones"..... or something......

```
raw_data.head()
```

	label	tweet
0	0	!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. &amp; as a man you should always take the trash out...
1	1	!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the 1st place!!
2	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a bitch and she start to cry? You be confused as shit
3	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she look like a tranny
4	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you hear about me might be true or it might be faker than the bitch who told it to ya &#57361;

```
raw_data.iloc[0]['tweet']
```

```
'!!! RT @mayasolovely: As a woman you shouldn't complain about cleaning up your house. &amp; as a man you should always ta
ke the trash out...'
```

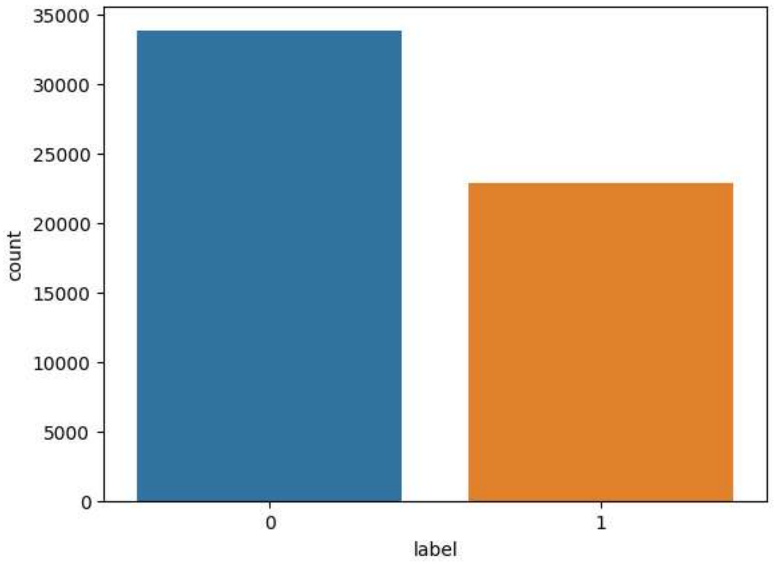
```
# Let's concatinate both the data into a single data frame.
frame = [imbalanced_data, raw_data]
df = pd.concat(frame)
```

```
df.head()
```

	label	tweet
0	0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run
1	0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked
2	0	bihday your majesty
3	0	#model i love u take with u all the time in urð□□±!!! ð□□□ð□□□ð□□□ð□□□ð□□¦ð□□¦ð□□¦
4	0	factsguide: society now #motivation

```
sns.countplot(x='label',data=df)
```

```
<Axes: xlabel='label', ylabel='count'>
```



```
# Now we can see that the problem of imbalace data has been solved.
```

```
df.shape
```

```
(56745, 2)
```

```
# Let's apply stemming and stopwords on the data
stemmer = nltk.SnowballStemmer("english")
stopword = set(stopwords.words('english'))


# Let's apply regex and do cleaning.
def data_cleaning(words):
    words = str(words).lower()
    words = re.sub('\[.*?\]', '', words)
    words = re.sub('https?://\S+|www\.\S+', '', words)
    words = re.sub('<.*?>+', '', words)
    words = re.sub('[%s]' % re.escape(string.punctuation), '', words)
    words = re.sub('\n', '', words)
    words = re.sub('\w*\d\w*', '', words)
    words = [word for word in words.split(' ') if words not in stopword]
    words=" ".join(words)
    words = [stemmer.stem(words) for word in words.split(' ')]
    words=" ".join(words)

    return words


# let's apply the data_cleaning on the data.
df['tweet']=df['tweet'].apply(data_cleaning)


df["tweet"][1]
```

```
1    user user thanks for lyft credit i cant use cause they dont offer wheelchair vans in pdx    disapointed getthank user user thanks
for lyft credit i cant use cause they dont offer wheelchair vans in pdx    disapointed getthank user user thanks for lyft...
1     rt  boy dats coldtyga dwn bad for cuffin dat hoe in the  plac  rt  boy dats coldtyga dwn bad for cuffin dat hoe in the  plac  rt
boy dats coldtyga dwn bad for cuffin dat hoe in the  plac  rt  boy dats coldtyga dwn bad for cuffin dat hoe in the  plac...
Name: tweet, dtype: object
```

```
x = df['tweet']
y = df['label']


# Let's split the data into train and test
x_train,x_test,y_train,y_test = train_test_split(x,y, random_state = 42)

print(len(x_train),len(y_train))
print(len(x_test),len(y_test))
```

```
42558 42558
14187 14187
```

```
type(x_test), type(y_test), type(x_train), type(y_train)
```

```
(pandas.core.series.Series,
 pandas.core.series.Series,
 pandas.core.series.Series,
 pandas.core.series.Series)
```

```
len(x_test)
```

```
14187
```

```
max_words = 50000
max_len = 300

tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(x_train)

sequences = tokenizer.texts_to_sequences(x_train)
sequences_matrix = pad_sequences(sequences,maxlen=max_len)


sequences_matrix
```

```
array([[    0,     0,     0, ...,   209, 13070,  4452],
       [    0,     0,     0, ...,   248,     3,   653],
       [    0,     0,     0, ...,     1,  1831, 41012],
       ...,
       [ 1126,   669,  2785, ...,   187,     1, 33462],
       [    0,     0,     0, ...,   954, 14416,   774],
       [    0,     0,     0, ...,   419,   378,    13]], dtype=int32)
```

```
# Creating model architecture.
model = Sequential()
model.add(Embedding(max_words,100,input_length=max_len))
model.add(SpatialDropout1D(0.2))
model.add(LSTM(100,dropout=0.2,recurrent_dropout=0.2))
model.add(Dense(1,activation='sigmoid'))
model.summary()
model.compile(loss='binary_crossentropy',optimizer=RMSprop(),metrics=['accuracy'])
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 embedding (Embedding)       (None, 300, 100)          5000000   
                                                                 
 spatial_dropout1d (SpatialD  (None, 300, 100)         0         
 ropout1D)                                                       
                                                                 
 lstm (LSTM)                 (None, 100)               80400     
                                                                 
 dense (Dense)               (None, 1)                 101       
                                                                 
=================================================================
Total params: 5,080,501
Trainable params: 5,080,501
Non-trainable params: 0
_________________________________________________________________
```

```
# starting model training
history = model.fit(sequences_matrix,y_train,batch_size=128,epochs = 1,validation_split=0.2)
```

```
266/266 [==============================] - 433s 2s/step - loss: 0.2612 - accuracy: 0.9062 - val_loss: 0.1686 - val_accuracy: 0.9394
```

```
test_sequences = tokenizer.texts_to_sequences(x_test)
test_sequences_matrix = pad_sequences(test_sequences,maxlen=max_len)
```

```
# Model evaluation
accr = model.evaluate(test_sequences_matrix,y_test)
```

```
444/444 [==============================] - 54s 123ms/step - loss: 0.1822 - accuracy: 0.9310
```

```
lstm_prediction = model.predict(test_sequences_matrix)
```

```
444/444 [==============================] - 53s 118ms/step
```

```
res = []
for prediction in lstm_prediction:
    if prediction[0] < 0.5:
        res.append(0)
    else:
        res.append(1)
```

```
print(confusion_matrix(y_test,res))
```

```
[[8070  383]
 [ 596 5138]]
```

```
import pickle
with open('tokenizer.pickle', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)
```

● ×