```
!unzip dataset.zip
```

```
Archive:  dataset.zip
replace imbalanced_data.csv? [y]es, [n]o, [A]ll, [N]one, [r]ename: ^C
```

```
import pandas as pd
```

```
imbalance_data = pd.read_csv("imbalanced_data.csv")
```

```
imbalance_data.head()
```

index	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. #disap
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

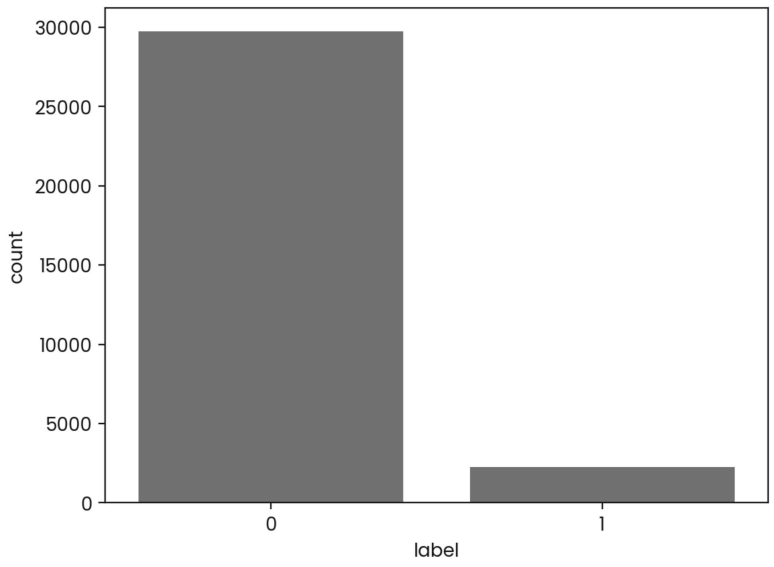
Rows: 5

Exploratory Data Analysis (EDA)

```
import seaborn as sns
```

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

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



- 0 ---> No hate
- 1 ---> Hate

```
imbalance_data.shape
```

```
(31962, 3)
```

```
imbalance_data.isnull().sum()
```

index	0
id	
label	
tweet	

Rows: 3

```
imbalance_data.drop("id", axis=1, inplace=True)
```

```
imbalance_data.head()
```

index	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 #getthanke
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

Rows: 5

```
raw_data = pd.read_csv("raw_data.csv")
```

```
raw_data.head()
```

	U...			hat...		offensive_lang...				tweet
0	0	3		0		0	3	2		!!! RT @mayasolovely: As a woman you shoul...
1	1	3		0		3	0	1		!!!!! RT @mleew17: boy dats cold...tyga dwn b...
2	2	3		0		3	0	1		!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sba...
3	3	3		0		2	1	1		!!!!!!!!! RT @C_G_Anderson: @viva_based she l...
4	4	6		0		6	0	1		!!!!!!!!!!!! RT @ShenikaRoberts: The shit you he...

Rows: 5

```
raw_data.shape
```

```
(24783, 7)
```

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

index	
Unnamed: 0	0
count	0
hate_speech	0
offensive_language	0
neither	0
class	0
tweet	0

Rows: 7

```
# 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)
```

```
raw_data.head()
```

		tweet
0	2	!!! RT @mayasolovely: As a woman you shoul...
1	1	!!!!! RT @mleew17: boy dats cold...tyga dwn b...
2	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sba...
3	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she l...
4	1	!!!!!!!!!!!! RT @ShenikaRoberts: The shit you he...

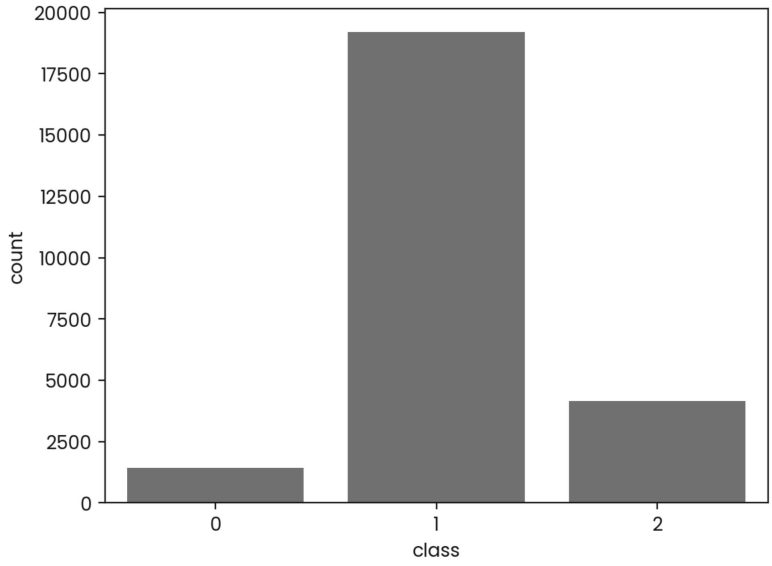
Rows: 5

```
# 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'>
```



- 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
```

```
raw_data.head()
```

		tweet
0	2	!!! RT @mayasolovely: As a woman you shoul...
1	1	!!!!! RT @mleew17: boy dats cold...tyga dwn b...
2	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sba...
3	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she l...
4	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you he...

Rows: 5

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

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

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

		tweet
85	0	"@Blackman38Tide: @WhaleLookyHere @Ho...
89	0	"@CB_Baby24: @white_thunduh alsarabsss" ...
110	0	"@DevilGrimz: @VigxRArts you're fucking gay...
184	0	"@MarkRoundtreeJr: LMFAOOOO I HATE BLA...
202	0	"@NoChillPaz: "At least I'm not a nigger" http:...
204	0	"@NotoriousBM95: @_WhitePonyJr_ Ariza is ...
219	0	"@RTNBA: Drakes new shoes that will be rele...
260	0	"@TheoMaxximus: #GerrysHalloweenParty h...
312	0	"@ashlingwilde: @ItsNotAdam is bored suppo...
315	0	"@bigbootybishopp: @white_thunduh lassen ...
349	0	"@jayswaggkillah: Jackies a retard #blondep...
352	0	"@jgabsss: Stacey Dash won 💦 http:/...
437	0	"Don't worry about the nigga you see, worry ...
459	0	"Hey go look at that video of the man that fo...
519	0	"Let's kill cracker babies!". WTF did I just hear...
526	0	"My grandma used to call me a porch monk...

Rows: 1,430

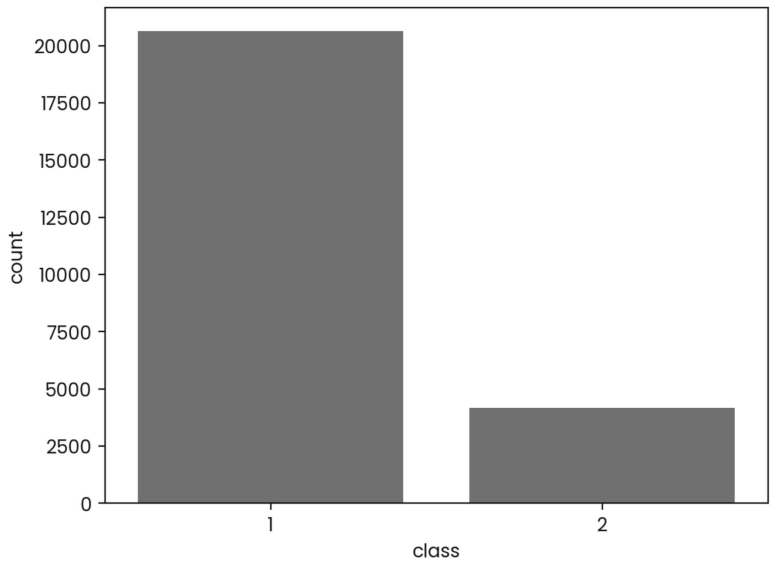
```
# replace the value of 0 to 1
raw_data["class"].replace({0:1},inplace=True)
```

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

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

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

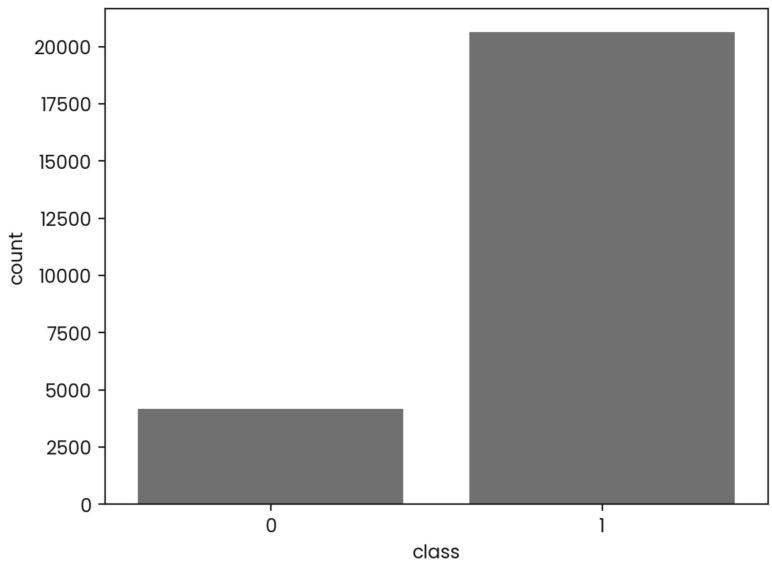
```
<Axes: xlabel='class', ylabel='count'>
```




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

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

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




```
imbalance_data.head()
```

		tweet
0	0	@user when a father is dysfunctional and is s...
1	0	@user @user thanks for #lyft credit i can't us...
2	0	bihday your majesty
3	0	#model i love u take with u all the time in ur...
4	0	factsguide: society now #motivation

Rows: 5

```
raw_data.head()
```

		tweet
0	0	!!! RT @mayasolovely: As a woman you shoul...
1	1	!!!!! RT @mleew17: boy dats cold...tyga dwn b...
2	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sba...
3	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she l...
4	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you he...

Rows: 5

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

```
raw_data.head()
```

		tweet
0	0	!!! RT @mayasolovely: As a woman you shoul...
1	1	!!!!! RT @mleew17: boy dats cold...tyga dwn b...
2	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sba...
3	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she l...
4	1	!!!!!!!!!!!!! RT @ShenikaRoberts: The shit you he...

Rows: 5

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

```
df.head()
```

		tweet
0	0	@user when a father is dysfunctional and is s...
1	0	@user @user thanks for #lyft credit i can't us...
2	0	bihday your majesty
3	0	#model i love u take with u all the time in ur...
4	0	factsguide: society now #motivation

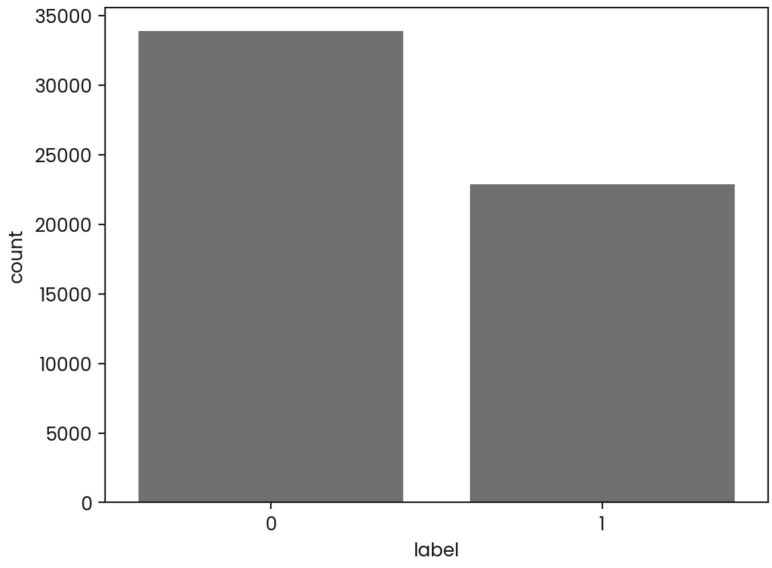
Rows: 5

```
df.shape
```

```
(56745, 2)
```

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

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



Preprocessing

```
import re
import nltk
import string
```

```
from nltk.corpus import stopwords
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /home/repl/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
True
```

```
# 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
```

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

	tweet
1	@user @user thanks for #lyft credit i can't us…
1	!!!!! RT @mleew17: boy dats cold...tyga dwn b…

Rows: 2

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

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

	tweet
1	user user thanks for lyft credit i cant use cau…
1	rt boy dats coldtyga dwn bad for cuffin dat …

Rows: 2

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

```
from sklearn.model_selection import train_test_split
```

```
# 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
```

Feature engineering

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

```
2024-12-05 23:58:17.142482: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
```

```
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)
```

```
sequences_matrix.shape
```

```
(42558, 300)
```

```
from keras.models import Sequential
from keras.layers import LSTM, Activation, Dense, Dropout, Input, Embedding, SpatialDropout1D
from keras.optimizers import RMSprop
```

```
# 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: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	?	0 (unbuilt)
spatial_dropout1d (SpatialDropout1D)	?	0 (unbuilt)
lstm (LSTM)	?	0 (unbuilt)
dense (Dense)	?	0 (unbuilt)

Total params: 0 (0.00 B)

Trainable params: 0 (0.00 B)

Non-trainable params: 0 (0.00 B)

```
model.compile(loss='binary_crossentropy',optimizer=RMSprop(),metrics=['accuracy'])
```

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

```
266/266 ━━━━━━━━━━━━━━━━━━━━ 201s 749ms/step - accuracy: 0.8290 - loss: 0.3823 - val_accuracy: 0.9402 - val_loss: 0.1740
```

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

```
test_sequences_matrix
```

```
array([[   29,  1856,  1260, ...,    13,    11,    29],
       [  471,   192,    31, ...,    16,     3, 17681],
       [    0,     0,     0, ...,   261,   331,  1505],
       ...,
       [    0,     0,     0, ...,    62,    10,   456],
       [    0,     0,     0, ...,     2,     4,     4],
       [    0,     0,     0, ...,     8,    88,  3776]], dtype=int32)
```

```
test_sequences_matrix.shape
```

```
(14187, 300)
```

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

```
444/444 ━━━━━━━━━━━━━━━━━━━━ 20s 46ms/step - accuracy: 0.9335 - loss: 0.1907
```

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

```
444/444 ━━━━━━━━━━━━━━━━━━━━ 20s 46ms/step
```

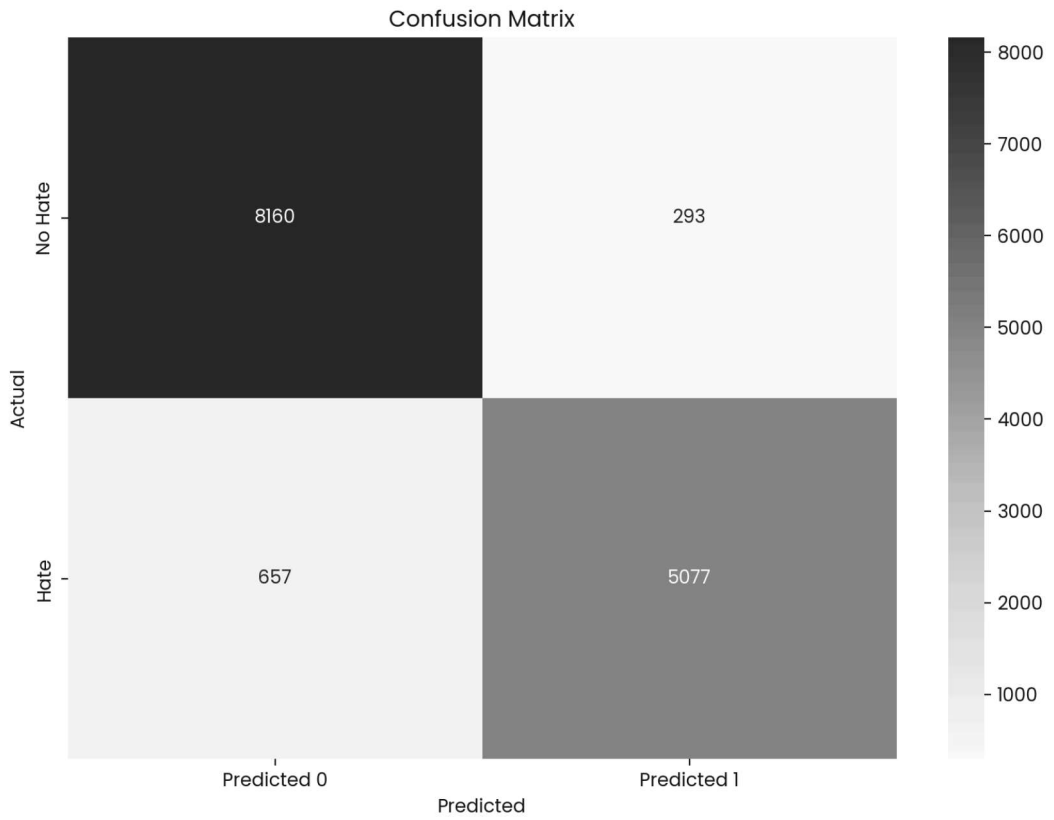
```
res = []
for prediction in lstm_prediction:
    if prediction[0] < 0.5:
        res.append(0)
    else:
        res.append(1)
```

```
from sklearn.metrics import confusion_matrix
```

```
import seaborn as sns
import matplotlib.pyplot as plt

# Generate the confusion matrix
cm = confusion_matrix(y_test, res)

# Plot the confusion matrix using seaborn
plt.figure(figsize=(10,7))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Predicted 0', 'Predicted 1'], yticklabels=['No Hate', 'Hate'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```




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

```
# Let's save the mdoel.
model.save("model.h5")
```

```
import keras
```

```
load_model=keras.models.load_model("model.h5")
```

```
with open('tokenizer.pickle', 'rb') as handle:
    load_tokenizer = pickle.load(handle)
```

```
# Let's test our model on custom data.
test = 'i love this movie'

def clean_text(text):
    print(text)
    text = str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    print(text)
    text = [word for word in text.split(' ') if word not in stopword]
    text=" ".join(text)
    text = [stemmer.stem(word) for word in text.split(' ')]
    text=" ".join(text)
    return text

test=[clean_text(test)]
print(test)

seq = load_tokenizer.texts_to_sequences(test)
padded = pad_sequences(seq, maxlen=300)
print(seq)
```