*11/2/*22, 10:22 AM

Assignment_4

Double-click (or enter) to edit

1. Download the dataset link

- Label - Ham or Spam
- Message - Message

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

2. Importing Required Library

Double-click (or enter) to edit

```
import re import nltk import pandas as pd import numpy as np import matplotlib.pyplot as plt from nltk.stem import
```

```
WordNetLemmatizer from nltk.corpus import stopwords from
wordcloud import Wordcloud, STOPWORDS,ImageColorGenerator
```

~ 3. Read dataset and do Preprocessing

```
df = pd.read_csv ("/content/spam.csv", encoding='ISO-8859-1')
```

```
df = df.iloc[:,:2]
df.columns=['label', 'message']
df.head()
```


	label	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2		onom Cron antime in 2 wlumomn to win CA Cin fino

```
df.info()
```

<class 'pandas.core.frame.DataFrame'> RangeIndex: 5572 entries, 0 to 5571 Data columns (total 2 columns):
Column Non-Null Count Dtype
--- o label 5572 non-null object 1 message 5572 non-null object
dtypes: object(2) memory usage: 87.2+ KB
--


```
ms1 = pd.Series((df.loc[df['label']=='ham',
'message']).tolist()).astype (str) wordcloud = WordCloud(
stopwords=STOPWORDS,width=800,height=600,
background_color='black'). plt.figure(figsize=(20,10))
plt.imshow(wordcloud) plt.axis('off')
```

11/2/22, 10:22 AM

Assignment_4

*lu*as 700 E 500 E
asl

```
ms2 = pd. Series((df.loc[df['label']=='spam',
'message']).tolist()).astype(str) wordcloud =
WordCloud(stopwords=STOPWORDS,
width=1000,height=400,
background_color='black').
plt.figure(figsize=(20,10)) plt.imshow(wordcloud)
plt.axis('off')
```

o
customer
number
msg
wanna know
contact
mates
word

cash

entry opt

pound •

pound

Spoly Weash Ext NOKIA

Know
ur cash.txt NOKIA
year Double dins
reward BT national

(-0.5, 999.5, 399.5, -0.5)

orize t% opt new a

SMS Itone

uk

nalne

Enjoy

dating service

M points per min

next service

sexy

TIEJ Tegas

prize GUARANTEED

gift voucher)

Today

latest

Mobile number

trying

Nokia tone

Valid 12hrs

land line

line Claim

met Bonus

Thank

ЖКХ.

Lumeur

every week new video

Free entry

Suite342 2Lands

1st

U

Co uko

see

Camcorder Reply

PRIVATE FreeMsg national rate

E Nokia

callessage

chance selected

send friend offer

nne Tental

winner

unsubscribe

auction

Code Expires

Jend 150p msg

HG Suite 342

yr

eivel

SMS ac

draw shows

phone

anten.o

love a

masz on

content

Claim ur

ücredit

love Ly Stor e

we

wille sue

WAP reply STOP

Xmas

service representative

Cash await

wk TXT tried

camera phone

every wk

date getzed co

Charged

back secret admirer

waiting 2Lands Row

Bonus Caller 15 Ls

wear

pic

exting landlineå award •

valued Orange

week'

someone

How Wl JUHL

orange W

O

Club

N

await collection

club

BY
Choose ur mob

lub cdWineweeki

Claim code

W in Cost_choose to mob

collect ŐL
Cost

2

award

PO Box

chat:

call2optout
game
shows un

find.contact U
day

awarded guaranteed å I reveal

TITT Identifier Code
ur awarded - - send STOP live operato

join min

Account Statement 2nd attempt
hot

prize Call Identifier GUARANTEED Call Call MobileUpds

customer service in to none Holiday T

Please cal

ist week

Can now

top

want habe

from nltk.stem.wordnet import WordNetLemmatizer
lemmatizer =WordNet Lemmatizer() corpus = []

import nltk from nltk.corpus import **stopwords**
nltk.download('all')

for i in range(len(df)):
review = re. sub('[^a-zA-Z]', '
',df['message'][i]) **review = review.lower() review = review**.split() **review** = [lemmatizer.lemmatize(i) for i in review if not i in set(stopwords.words('enę

review = ''.join(review) corpus.append(review)

LMILK_uala] i roul/ MILK_ualde.. [nltk_data] | Unzipping grammars/spanish_grammars.zip. [nltk_data] Downloading package state_union to /root/nltk_data... [nltk_data] Unzipping corpora/state_union.zip. [nltk_data] | Downloading package stopwords to /root/nltk_data... [nltk_data] | Unzipping corpora/stopwords.zip. [nltk_data] | Downloading package subjectivity to [nltk_data] /root/nitk_data... Inl+k datal

Unzinning corporalcubiectivity zin

["en_uacaj [nltk_data] [nltk datal

 VIILIPP1116 cui pui a, suv youuviuy.c?y. | Downloading package swadesh to /root/nltk_data... | Unzipping corpora/swadesh.zip.

Downloading package switchboard to /root/nltk_data...
Unzipping corpora/switchboard.zip. Downloading package tagsets to /root/nltk_data... | Unzipping help/tagsets.zip. | Downloading package timit to /root/nltk_data...
Unzipping corpora/timit.zip. | **Downloading package** toolbox to /root/nltk_data...
Unzipping corpora/toolbox.zip. Downloading package treebank to /root/nltk_data... | Unzipping corpora/treebank.zip.
Downloading package twitter_samples to
/root/nltk_data...
Unzipping corpora/twitter_samples.zip. | Downloading package udhr to /root/nltk_data... | Unzipping corpora/udhr.zip. | Downloading package udhr2 to /root/nltk_data...
Unzipping corpora/udhr2.zip. Downloading package unicode_samples to L /root/nltk_data...
Unzipping corpora/unicode_samples.zip. | Downloading package universal_tagset to L /root/nltk_data...
Unzipping taggers/universal_tagset.zip. | Downloading **package universal_treebanks_v20 to**
/root/nltk_data... **Downloading package vader_lexicon to**
/root/nltk_data... | Downloading package verbnet to /root/nltk_data... | Unzipping corpora/verbnet.zip. | Downloading package verbnet3 to /root/nltk_data...
Unzipping corpora/verbnet3.zip. **Downloading package webtext to**
/root/nltk_data...
Unzipping corpora/webtext.zip. | Downloading package wmt15_eval to /root/nltk_data...
Unzipping models/wmt15_eval.zip. Downloading package word2vec_sample to /root/nltk_data.... Unzipping models/word2vec_sample.zip. Downloading **package wordnet to** /root/nltk_data... Downloading package wordnet2021 to /root/nltk_data.. **Downloading** package wordnet31 to /root/nltk_data... **Downloading package wordnet_ic to** /root/nltk_data...
Unzipping corpora/wordnet_ic.zip. Downloading package words to /root/nltk_data... | Unzipping corpora/words.zip.
Downloading **package ycoe to** /root/nltk_data...
Unzipping corpora/ycoe.zip.

- 4. Create Model

from keras.preprocessing. text import Tokenizer from keras_preprocessing.sequence import pad_sequences

from keras.layers import Dense, Dropout, LSTM, Embedding from bonne models immont Connontillond modal token = Tokenizer() token.fit_on_texts(corpus) text_to_seq = token.texts_to_sequences(corpus)

max_length_sequence = max([len(i) for i in text_to_seq])

padded_seq = pad_sequences (text_to_seq, maxlen=max_length_sequence, padding="pre")

padded_sea

```
array([[    0,
        [
        [
            0,
            0,
            0,
0,..., 16, 3551, 70], 0, ..., 359, 1, 1610], 0,
                    ..., 218, 29, 293],
```

```
        0,
    [
    [
    [
       0,
       0,
       0,
            0,
                0, ..., 7042, 1095, 3547], 0, ...,
                842, 1, 10], 0, ..., 2198, 347,
                152]], dtype=int32)
            0,
```

```
from sklearn.preprocessing import
LabelEncoder le = LabelEncoder() y =
le.fit_transform(df['label'])
```

```
from sklearn.model_selection import train_test_split X_train,x_test,y_train,y_test =
train_test_split(padded_seq,y,test_size=0.25, random_state
```

```
X_train.shape
```

*(4179,
77)*

5. Add Layers

```
TOT_SIZE = len(token.word_index) + 1 model = Sequential() #IP
Layer model.add(Embedding(TOT_SIZE, 32,
input_length=max_length_sequence)) model.add(LSTM(units=50,
activation = 'relu', return_sequences=True))
```

model.add(Dropout(0.2))

#Layer2 model.add(LSTM(units=60, activation = 'relu')) model.add(Dropout(0.3)) #output layer model.add(Dense(units=1, activation='sigmoid'))

WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the c WARNING:tensorflow:Layer lstm_1 will not use cuDNN kernels since it doesn't meet the

model.summary()

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 77, 32)	225408
lstm (LSTM)	(None, 77, 50)	16600
dropout (Dropout)	(None, 77, 50)	0
lstm_1 (LSTM)		

(None, 60)
26640

dropout_1 (Dropout)
(None, 60)

dense (Dense)
(None, 1)

=====
===
=====
======
==
==========
Total params: 268,709 **Trainable params: 268, 709** Non-trainable params: 0

- 6 Compile the model

model.compile(optimizer='adam', **loss='binary_crossentropy',** metrics=['accuracy'])

- 7 Fit the model

model.fit(x_train, y_train, validation_data=(x_test,y_test), epochs=10)

Epoch 1/10 **131/131 [==========] - 33$** 252ms/step - loss: 0.1533 - accuracy: Epoch 2/10 **131/131 [==========]** - 335 251ms/step - **loss: 0.3151 -**

accuracy: Epoch 3/10 131/131 [==========] - 32s 247ms/step - loss: 0.1197 - accuracy: Epoch 4/10 131/131 [==========] - 36s 278ms/step - loss: 0.0955 - accuracy: Epoch 5/10 131/131 [==========] - 365 271ms/step - loss: 0.0788 - accuracy: Epoch 6/10 131/131 [==========] - 34s 261ms/step - loss: 0.0663 - accuracy: Epoch 7/10 131/131 [==========] - 335 248ms/step - loss: 0.0559 - accuracy: Epoch 8/10 131/131 [==========] - 32s 243ms/step - loss: 0.0477 - accuracy: Epoch 9/10 131/131 [==========] - 32s 247ms/step - loss: 0.0413 - accuracy: Epoch 10/10 131/131 [==========] - 32s 245ms/step - loss: 0.0384 - accuracy: <keras.callbacks.History at 0x7f035858c9d0>

accurac
4

https://colab.research.google.com/drive/1_XZBWvdLmxiVlwQIDEJSIB3OsE3FTNJH#scrollTo=bc381cc7&printMode=true


```
model.evaluate(x_test,y_test)
```

```
44/44 [==========] - 1s 20ms/step - loss: 0.0987 - accuracy: 0.9
[0.09865640848875046, 0.9777458906173706]
```

- 8. Save the Model

```
from pickle import dump,
```

```
load tfid = 'tfid.sav' lstm =
'1stm.sav'


    dump(token, open(tfid,
'wb')) model.save('nlp.h5')
```

9. Test the Model

```
def preprocess(raw_mess):
    review = re.sub ('[^a-zA-Z]',' ', raw_mess) review = review.lower() review =
    review.split() review = [lemmatizer.lemmatize(i) for i in review if not i in
    set(stopwords.words('eng review = ' '.join(review) return review


def predict(mess):
    vect = load(open(tfid, 'rb')) classifier = load_model('nlp.h5')
    clean = preprocess(mess) text_to_seq = token
    texts_to_sequences ([mess]) padded_seq = pad_sequences
    (text_to_seq, maxlen=77, padding="pre") pred =
    classifier.predict(padded_seq) return pred


msg = input("Enter a message:
") predi = predict(msg) if predi >=
0.6:
        print("It is a
spam") else:
    print("Not a spam")

     Enter a message:
```

```
msg = input("Enter a message: ")
predi = predict(msg)
if predi >= 0.6:
print("It is a spam")
else:
print("Not a spam")
```

Colab paid products - Cancel contracts here

Executing (5m 24s) Cell > raw_input
> _input_request() > select()
. . .
X