```
from __future__ import absolute_import, division, print_function, unicode_literals

import numpy as np

try:
  # %tensorflow_version only exists in Colab.
  %tensorflow_version 2.x
except Exception:
  pass
import tensorflow as tf

!pip install tensorflow-hub
!pip install tfds-nightly
import tensorflow_hub as hub
import tensorflow_datasets as tfds

print("Version: ", tf.__version__)
print("Eager mode: ", tf.executing_eagerly())
print("Hub version: ", hub.__version__)
print("GPU is", "available" if tf.config.experimental.list_physical_devices("GPU") els
```

```
Requirement already satisfied: tensorflow-hub in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: numpy>=1.12.0 in /usr/local/lib/python3.6/dist-pac
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: protobuf>=3.4.0 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packag
Collecting tfds-nightly
  Downloading https://files.pythonhosted.org/packages/fc/09/3be889b6ef8424273d10a
     |████████████████████████████████| 3.3MB 3.5MB/s
Requirement already satisfied: tensorflow-metadata in /usr/local/lib/python3.6/di
Requirement already satisfied: termcolor in /usr/local/lib/python3.6/dist-package
Requirement already satisfied: future in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: absl-py in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: protobuf>=3.6.1 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: wrapt in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (fro
Requirement already satisfied: dill in /usr/local/lib/python3.6/dist-packages (fr
Requirement already satisfied: attrs>=18.1.0 in /usr/local/lib/python3.6/dist-pac
Requirement already satisfied: promise in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: tqdm in /usr/local/lib/python3.6/dist-packages (fr
Requirement already satisfied: googleapis-common-protos in /usr/local/lib/python3
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dis
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /usr/local/lib/python3.6/
Requirement already satisfied: idna<2.9,>=2.5 in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python3.6/
Installing collected packages: tfds-nightly
Successfully installed tfds-nightly-2.1.0.dev202003300105
Version:  2.2.0-rc1
Eager mode:  True
Hub version:  0.7.0
GPU is available
```

```
# Using google colab - this first step is for loading in the data from my personal Dri

# Login with google credentials

from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)

# Handle errors from too many requests

import logging
logging.getLogger('googleapiclient.discovery_cache').setLevel(logging.ERROR)

# The ID for my personal Drive folder is 1BVUuroPvozFxMjMIYrGOFtI4r6erSBCx
```

```
# The ID for my personal Drive folder is 1BVUuroPvozFxMjMIYrGOFtI4r6erSBCx
# I am now listing the ID numbers for the files in this folder to find the data files

#file_list = drive.ListFile({'q': "'1BVUuroPvozFxMjMIYrGOFtI4r6erSBCx' in parents and
#for file1 in file_list:
#  print('title: %s, id: %s' % (file1['title'], file1['id']))


# Now that I have the ID files, load the files

from google.colab import auth
auth.authenticate_user()
import gspread
from oauth2client.client import GoogleCredentials

gc = gspread.authorize(GoogleCredentials.get_application_default())
wb = gc.open_by_url('https://docs.google.com/spreadsheets/d/1mZ2VUPqvPujfO87V9u8_mH_uc
sheet = wb.sheet1
data = sheet.get_all_values()



# Import NTLK libraries

import numpy as np
import nltk
from nltk import sent_tokenize, word_tokenize, pos_tag

import re
from sklearn.feature_extraction.text import CountVectorizer
wpt = nltk.WordPunctTokenizer()
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
stop_words = nltk.corpus.stopwords.words('english')
```

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

```
# convert DF to Pandas, remove top row with column IDs, limit to only labelled entries

import pandas as pd
import matplotlib.pyplot as plt

df = pd.DataFrame(data)
df.columns = df.iloc[0]
df = df.iloc[1:]
df = df[:700]
#df['Label'], df['Tweet']
```

```
df = df.reset_index()

target = df['Label'].astype(int)
tweets = df['Tweet'].copy(deep=True)


target.describe()

# 34% are 0, then 66% are 1
```

```
count    700.000000
mean       0.341429
std        0.474528
min        0.000000
25%        0.000000
50%        0.000000
75%        1.000000
max        1.000000
Name: Label, dtype: float64
```

```
wn = nltk.WordNetLemmatizer()
import string

## Function to clean entries - lowercase, lemmatize, tokenize, drop stop words, recomb

def clean_articles(doc):
    for column in range(len(doc)):

        #doc[column] = doc[column].astype(str)
        doc[column] = doc[column].replace('[^\w\s]','')

        doc[column] = doc[column].lower()
        # removing punctuation worsens results a lot
        #doc[column] = ''.join([str(item) for item in doc[column] if item not in strin

        #doc[column] = doc[column].replace(np.nan, ' ', regex=True)

        doc[column] = nltk.word_tokenize(doc[column])
        doc[column] = lemmatize_text(doc[column])

        doc[column] = [token for token in doc[column] if token not in stop_words]
        doc[column] = ' '.join([str(item) for item in doc[column] ])
    return doc

def lemmatize_text(tokenized_text):
    text = [wn.lemmatize(word) for word in tokenized_text]
    return text


# Get cleaned data
data_clean = clean_articles(tweets)
```

```
# reunify cleaned data and target

merge_df = pd.concat([data_clean, target], axis=1, sort=False)

# shuffle the data

train_shuffle_0 = merge_df[merge_df["Label"] == 0].sample(frac=0.8, random_state=np.ra
train_shuffle_1 = merge_df[merge_df["Label"] == 1].sample(frac=0.8, random_state=np.ra

test_shuffle_0 = merge_df[merge_df["Label"] == 0].drop(train_shuffle_0.index)
test_shuffle_1 = merge_df[merge_df["Label"] == 1].drop(train_shuffle_1.index)

train_shuffle = pd.concat([train_shuffle_0, train_shuffle_1], axis=0, sort=False).samp
test_shuffle = pd.concat([test_shuffle_0, test_shuffle_1], axis=0, sort=False).sample(

data_clean_train = train_shuffle['Tweet']
data_clean_test = test_shuffle['Tweet']

# train test split

target_train = train_shuffle['Label']
target_test = test_shuffle['Label']



target_test.describe()
```

```
count    140.000000
mean       0.342857
std        0.476369
min        0.000000
25%        0.000000
50%        0.000000
75%        1.000000
max        1.000000
Name: Label, dtype: float64
```

```
# Convert pandas to TF data - thanks ben
train_data = (
    tf.data.Dataset.from_tensor_slices(
        (
            tf.cast(data_clean_train.values, tf.string),
            tf.cast(target_train.values, tf.int32)
        )
    )
)

test_data = (
    tf.data.Dataset.from_tensor_slices(
        (
            tf.cast(data_clean_test.values, tf.string),
            tf.cast(target_test.values, tf.int32)
        )
```

```
        ,
    )
)

train_data = train_data.shuffle(5000).batch(32)
test_data = test_data.shuffle(5000).batch(32)


from tensorflow import keras

## Transfer learning

#embedding = "https://tfhub.dev/google/tf2-preview/gnews-swivel-20dim/1"
hub_layer = hub.KerasLayer("https://tfhub.dev/google/tf2-preview/nnlm-en-dim128/1", ou
                           input_shape=[], dtype=tf.string)

## Set up model

model = tf.keras.Sequential()

model.add(hub_layer)
model.add(tf.keras.layers.Dense(1000, activation ='relu'))
model.add(tf.keras.layers.Dense(200, activation ='relu'))
keras.layers.Dropout(0.2, noise_shape=None, seed=None)
model.add(tf.keras.layers.Dense(16, activation ='relu'))
keras.layers.Dropout(0.2, noise_shape=None, seed=None)
model.add(tf.keras.layers.Dense(1, activation ='sigmoid'))

model.summary()
```

```
Model: "sequential_6"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
keras_layer_6 (KerasLayer)   (None, 128)               124642688 
_________________________________________________________________
dense (Dense)                (None, 1000)              129000    
_________________________________________________________________
dense_1 (Dense)              (None, 200)               200200    
_________________________________________________________________
dense_2 (Dense)              (None, 16)                3216      
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 17        
=================================================================
Total params: 124,975,121
Trainable params: 332,433
Non-trainable params: 124,642,688
_________________________________________________________________
```

```
model.compile(optimizer='adam', loss =tf.keras.losses.BinaryCrossentropy(from_logits=T

history = model.fit(train_data, epochs=100, verbose=1)
```

```
Epoch 2/100
18/18 [==============================] - 0s 11ms/step - loss: 0.6766 - binary_acc
Epoch 3/100
18/18 [==============================] - 0s 10ms/step - loss: 0.6598 - binary_acc
Epoch 4/100
18/18 [==============================] - 0s 11ms/step - loss: 0.6541 - binary_acc
Epoch 5/100
18/18 [==============================] - 0s 10ms/step - loss: 0.6469 - binary_acc
Epoch 6/100
18/18 [==============================] - 0s 11ms/step - loss: 0.6405 - binary_acc
Epoch 7/100
18/18 [==============================] - 0s 10ms/step - loss: 0.6198 - binary_acc
Epoch 8/100
18/18 [==============================] - 0s 10ms/step - loss: 0.6006 - binary_acc
Epoch 9/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5976 - binary_acc
Epoch 10/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5943 - binary_acc
Epoch 11/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5908 - binary_acc
Epoch 12/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5833 - binary_acc
Epoch 13/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5777 - binary_acc
Epoch 14/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5763 - binary_acc
Epoch 15/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5802 - binary_acc
Epoch 16/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5754 - binary_acc
Epoch 17/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5792 - binary_acc
Epoch 18/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5752 - binary_acc
Epoch 19/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5781 - binary_acc
Epoch 20/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5754 - binary_acc
Epoch 21/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5758 - binary_acc
Epoch 22/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5751 - binary_acc
Epoch 23/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5745 - binary_acc
Epoch 24/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5779 - binary_acc
Epoch 25/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5738 - binary_acc
Epoch 26/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5757 - binary_acc
Epoch 27/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5751 - binary_acc
Epoch 28/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5757 - binary_acc
Epoch 29/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5757 - binary_acc
Epoch 30/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5764 - binary_acc
```

```
Epoch 31/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5751 - binary_acc
Epoch 32/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5770 - binary_acc
Epoch 33/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5770 - binary_acc
Epoch 34/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5744 - binary_acc
Epoch 35/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5777 - binary_acc
Epoch 36/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5768 - binary_acc
Epoch 37/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5784 - binary_acc
Epoch 38/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5770 - binary_acc
Epoch 39/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5731 - binary_acc
Epoch 40/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5750 - binary_acc
Epoch 41/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5737 - binary_acc
Epoch 42/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5768 - binary_acc
Epoch 43/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5770 - binary_acc
Epoch 44/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5750 - binary_acc
Epoch 45/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5757 - binary_acc
Epoch 46/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5737 - binary_acc
Epoch 47/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5781 - binary_acc
Epoch 48/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5750 - binary_acc
Epoch 49/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5757 - binary_acc
Epoch 50/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5744 - binary_acc
Epoch 51/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5761 - binary_acc
Epoch 52/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5764 - binary_acc
Epoch 53/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5757 - binary_acc
Epoch 54/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5764 - binary_acc
Epoch 55/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5737 - binary_acc
Epoch 56/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5757 - binary_acc
Epoch 57/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5750 - binary_acc
Epoch 58/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5750 - binary_acc
Epoch 59/100
```

```
18/18 [==============================] - 0s 10ms/step - loss: 0.5750 - binary_acc
Epoch 60/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5744 - binary_acc
Epoch 61/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5755 - binary_acc
Epoch 62/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5750 - binary_acc
Epoch 63/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5763 - binary_acc
Epoch 64/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5782 - binary_acc
Epoch 65/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5831 - binary_acc
Epoch 66/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5835 - binary_acc
Epoch 67/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5819 - binary_acc
Epoch 68/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5815 - binary_acc
Epoch 69/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5869 - binary_acc
Epoch 70/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5823 - binary_acc
Epoch 71/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5820 - binary_acc
Epoch 72/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5838 - binary_acc
Epoch 73/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5773 - binary_acc
Epoch 74/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5801 - binary_acc
Epoch 75/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5778 - binary_acc
Epoch 76/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5745 - binary_acc
Epoch 77/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5777 - binary_acc
Epoch 78/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5737 - binary_acc
Epoch 79/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5728 - binary_acc
Epoch 80/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5744 - binary_acc
Epoch 81/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5751 - binary_acc
Epoch 82/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5751 - binary_acc
Epoch 83/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5744 - binary_acc
Epoch 84/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5757 - binary_acc
Epoch 85/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5744 - binary_acc
Epoch 86/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5731 - binary_acc
Epoch 87/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5717 - binary_acc
Epoch 88/100
```

```
18/18 [==============================] - 0s 10ms/step - loss: 0.5755 - binary_acc
Epoch 89/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5764 - binary_acc
Epoch 90/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5724 - binary_acc
Epoch 91/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5764 - binary_acc
Epoch 92/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5768 - binary_acc
Epoch 93/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5737 - binary_acc
Epoch 94/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5731 - binary_acc
Epoch 95/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5750 - binary_acc
Epoch 96/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5757 - binary_acc
Epoch 97/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5750 - binary_acc
Epoch 98/100
18/18 [==============================] - 0s 11ms/step - loss: 0.5737 - binary_acc
Epoch 99/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5737 - binary_acc
Epoch 100/100
18/18 [==============================] - 0s 10ms/step - loss: 0.5750 - binary_acc
```

```
results = model.evaluate(test_data, verbose=2)

for name, value in zip(model.metrics_names, results):
  print("%s: %.3f" % (name, value))
```

```
5/5 - 0s - loss: 0.6039 - binary_accuracy: 0.8929
loss: 0.604
binary_accuracy: 0.893
```

```
results
```

```
[0.6038973927497864, 0.8928571343421936]
```

```
model.predict(["I hated it"])
```

```
array([[2.7867325e-34]], dtype=float32)
```

```
model.predict(["loved it!"])
```

```
array([[2.8276788e-21]], dtype=float32)
```

```
model.predict(["It Was a glorious triumph"])
```

```
array([[3.9321082e-36]], dtype=float32)
```

```
model.predict(["I thought it Was a glorious piece of complete crap!"])
```