```
!nvidia-smi
```

```
NVIDIA-SMI has failed because it couldn't communicate with the NVIDIA driver. Make su
```

```
!pip install -q git+https://github.com/tensorflow/docs
```

```
  Building wheel for tensorflow-docs (setup.py) ... done
```

```
import numpy as np
import pandas as pd

import tensorflow as tf
import tensorflow_hub as hub
import tensorflow_datasets as tfds
import seaborn as sns
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (12, 8)
from  IPython import display

import pathlib
import shutil
import tempfile



import tensorflow_docs as tfdocs
import tensorflow_docs.modeling
import tensorflow_docs.plots

print("Version: ", tf.__version__)
print("Hub version: ", hub.__version__)
print("GPU is", "available" if tf.config.list_physical_devices('GPU') else "NOT AVAILABLE"

#logdir = pathlib.Path(tempfile.mkdtemp())/"tensorboard_logs"
#shutil.rmtree(logdir, ignore_errors=True)
```

```
Version:  2.8.2
Hub version:  0.12.0
GPU is NOT AVAILABLE
```

```
df = pd.read_csv("/content/train.csv.zip",
    compression='zip',
    low_memory=False
)
```

```
df.shape
```

```
(1306122, 3)
```

```
df.head()
```

```
df.target.plot(kind='hist', title="Target Distribution")
```

```
from sklearn.model_selection import train_test_split

train_df, remaning = train_test_split(df, random_state=42, train_size=0.01, stratify=df.ta

train_df.head()
```

```
remaning.head()
```

```
validation_df, _ = train_test_split(remaning, random_state=42, train_size=0.001, stratify=
```

```
print(f'train size: {train_df.shape}\nvalidation size: {validation_df.shape}')
```

```
train size: (13061, 3)
validation size: (1293, 3)
```

```
validation_df.head()
```

```
train_df.target.head(15).values
```

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

```
train_df.question_text.head(15).values
```

```
array(['What is your experience living in Venezuela in the current crisis? (2018)',
       'In which state/city the price of property is highest?',
       'Do rich blacks also call poor whites, “White Trash”?',
       'Should my 5 yr old son and 2 yr old daughter spend the summer with their
father, after a domestic violent relationship?',
       'Why do we have parents?',
       'Do we experience ghost like Murphy did in Interstellar?',
       'Are Estoniano women beautiful?',
       'There was a Funny or Die video called Sensitivity Hoedown that got pulled.
Does anyone know why?',
       'Is it a good idea to go in fully mainstream classes, even if I have
meltdowns that might disrupt people?',
       'What classifies a third world country as such?',
       'Is being a pilot safe?',
       'Who is Illiteratendra Modi? Why does he keep with him a Rs 1 lakh pen?',
       'Have modern management strategies such as Total supply Chain Management
applied to education? Can they be?',
       'Why are Lucky Charms considered good for you?',
       'How many people in India use WhatsApp, Facebook, Twitter and Instagram?'],
      dtype=object)
```

```
module_url = "https://tfhub.dev/google/tf2-preview/gnews-swivel-20dim/1" #@param ["https:/
```

module_url: "https://tfhub.dev/google"

```
import time
import os
def get_log_path(log_dir="logs/fit"):
  uniqueName = time.strftime("log_%Y_%m_%d_%H_%M_%S")
  log_path = os.path.join(log_dir, uniqueName)
  print(f"savings logs at: {log_path}")

  return log_path

log_dir = get_log_path()
```

```
savings logs at: logs/fit/log_2022_06_26_19_42_42
```

```
tensorboard_cb = tf.keras.callbacks.TensorBoard(log_dir=log_dir)
early_stopping_cb = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=2, mode=

EpochDots_cb = tfdocs.modeling.EpochDots()

CALLBACKS_LIST = [tensorboard_cb, early_stopping_cb, EpochDots_cb]
```

```
def train_and_evaluate_model(module_url, embed_size, name, trainable=False):
    hub_layer = hub.KerasLayer(module_url, input_shape=[], output_shape=[embed_size], dtyp
    model = tf.keras.models.Sequential([
        hub_layer,
        tf.keras.layers.Dense(256, activation=tf.nn.relu),
        tf.keras.layers.Dense(64, activation=tf.nn.relu),
        tf.keras.layers.Dense(1, activation=tf.nn.sigmoid)
    ])
    model.compile(
        optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001),
        loss = tf.keras.losses.BinaryCrossentropy(),
        metrics = [tf.keras.metrics.BinaryAccuracy(name='accuracy')]
    )

    model.summary()

    history = model.fit(
        train_df.question_text,
        train_df.target,
        epochs = 100,
        validation_data = (validation_df.question_text, validation_df.target),# validation
        callbacks = CALLBACKS_LIST,
        verbose=0
    )
```

```
histories = {} # will saave history object returned by train_and_evaluate_model. So I can
```

```
module_url = "https://tfhub.dev/google/tf2-preview/gnews-swivel-20dim/1" #@param ["https:/
```

module_url: https://tfhub.dev/google

```
histories['gnews-swivel-20dim'] = train_and_evaluate_model(module_url, embed_size=20, name
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer (KerasLayer)    (None, 20)                400020

 dense (Dense)               (None, 256)               5376

 dense_1 (Dense)             (None, 64)                16448

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 421,909
Trainable params: 21,889
Non-trainable params: 400,020
_________________________________________________________________

Epoch: 0, accuracy:0.9313,  loss:0.2716,  val_accuracy:0.9381,  val_loss:0.2009,
.........
```

```
module_url = "https://tfhub.dev/google/tf2-preview/nnlm-en-dim50/1" #@param ["https://tfhu
```

module_url: https://tfhub.dev/google

```
histories['nnlm-en-dim50'] = train_and_evaluate_model(
                                                                   module_url, embed_
                                                                   name='nnlm-en-dim5
                                                                   trainable=False
                                                                )
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer_1 (KerasLayer)  (None, 50)                48190600

 dense_3 (Dense)             (None, 256)               13056
 dense_4 (Dense)             (None, 64)                16448

 dense_5 (Dense)             (None, 1)                 65

=================================================================
Total params: 48,220,169
Trainable params: 29,569
Non-trainable params: 48,190,600
_________________________________________________________________

Epoch: 0, accuracy:0.9102,  loss:0.3493,  val_accuracy:0.9381,  val_loss:0.2273,
...........
```

```
module_url = "https://tfhub.dev/google/tf2-preview/nnlm-en-dim128/1" #@param ["https://tfh
```

module_url: "https://tfhub.dev/google

```
histories['nnnlm-en-dim128'] = train_and_evaluate_model(module_url, embed_size=128, name='
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer_2 (KerasLayer)  (None, 128)               124642688

 dense_6 (Dense)             (None, 256)               33024

 dense_7 (Dense)             (None, 64)                16448

 dense_8 (Dense)             (None, 1)                 65

=================================================================
Total params: 124,692,225
Trainable params: 49,537
Non-trainable params: 124,642,688
_________________________________________________________________

Epoch: 0, accuracy:0.9346,  loss:0.3229,  val_accuracy:0.9381,  val_loss:0.2105,
..............
```