```
# import libraries
import pandas as pd
!pip install tensorflow-datasets
import tensorflow_datasets as tfds
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.text import one_hot
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Flatten, Embedding, Dense
from tensorflow.keras.callbacks import EarlyStopping
```

```
Requirement already satisfied: tensorflow-datasets in /usr/local/lib/python3.10/dist-packages (4.9.6)
Requirement already satisfied: absl-py in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (1.4.0)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (8.1.7)
Requirement already satisfied: dm-tree in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (0.1.8)
Requirement already satisfied: immutabledict in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (4.2.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (1.26.4)
Requirement already satisfied: promise in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (2.3)
Requirement already satisfied: protobuf>=3.20 in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (3.20.3)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (5.9.5)
Requirement already satisfied: pyarrow in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (16.1.0)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (2.32.3)
Requirement already satisfied: simple-parsing in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (0.1.6)
Requirement already satisfied: tensorflow-metadata in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (1.16.1)
Requirement already satisfied: termcolor in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (2.5.0)
Requirement already satisfied: toml in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (0.10.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (4.66.5)
Requirement already satisfied: wrapt in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (1.16.0)
Requirement already satisfied: array-record>=0.5.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow-datasets) (0.5.1)
Requirement already satisfied: etils>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from etils[enp,epath,epy,etree]>=1.6.0; python_v
Requirement already satisfied: typing_extensions in /usr/local/lib/python3.10/dist-packages (from etils[enp,epath,epy,etree]>=1.6.0; pyt
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from etils[enp,epath,epy,etree]>=1.6.0; python_version
Requirement already satisfied: importlib_resources in /usr/local/lib/python3.10/dist-packages (from etils[enp,epath,epy,etree]>=1.6.0; p
Requirement already satisfied: zipp in /usr/local/lib/python3.10/dist-packages (from etils[enp,epath,epy,etree]>=1.6.0; python_version <
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->tensorflow-da
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->tensorflow-datasets) (3.1
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->tensorflow-datasets
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.19.0->tensorflow-datasets
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from promise->tensorflow-datasets) (1.16.0)
Requirement already satisfied: docstring-parser<1.0,>=0.15 in /usr/local/lib/python3.10/dist-packages (from simple-parsing->tensorflow-d
```

```
# get data files
!wget https://cdn.freecodecamp.org/project-data/sms/train-data.tsv
!wget https://cdn.freecodecamp.org/project-data/sms/valid-data.tsv

train_file_path = "train-data.tsv"
test_file_path = "valid-data.tsv"
```

```
--2024-10-31 15:46:59--  https://cdn.freecodecamp.org/project-data/sms/train-data.tsv
Resolving cdn.freecodecamp.org (cdn.freecodecamp.org)... 172.67.70.149, 104.26.2.33, 104.26.3.33, ...
Connecting to cdn.freecodecamp.org (cdn.freecodecamp.org)|172.67.70.149|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 358233 (350K) [text/tab-separated-values]
Saving to: ‘train-data.tsv.8’

train-data.tsv.8    100%[===================>] 349.84K  --.-KB/s    in 0.03s

2024-10-31 15:46:59 (13.5 MB/s) - ‘train-data.tsv.8’ saved [358233/358233]

--2024-10-31 15:46:59--  https://cdn.freecodecamp.org/project-data/sms/valid-data.tsv
Resolving cdn.freecodecamp.org (cdn.freecodecamp.org)... 172.67.70.149, 104.26.2.33, 104.26.3.33, ...
Connecting to cdn.freecodecamp.org (cdn.freecodecamp.org)|172.67.70.149|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 118774 (116K) [text/tab-separated-values]
Saving to: ‘valid-data.tsv.8’

valid-data.tsv.8    100%[===================>] 115.99K  --.-KB/s    in 0.01s

2024-10-31 15:46:59 (8.10 MB/s) - ‘valid-data.tsv.8’ saved [118774/118774]
```

```
names = ["class", "message"]
```

```
train_file = pd.read_csv(train_file_path, sep='\t', names=names)
train_file
```

	class	message
0	ham	ahhhh...just woken up!had a bad dream about u ...
1	ham	you can never do nothing
2	ham	now u sound like manky scouse boy steve,like! ...
3	ham	mum say we wan to go then go... then she can s...
4	ham	never y lei... i v lazy... got wat? dat day ü ...
...	...	...
4174	ham	just woke up. yeesh its late. but i didn't fal...
4175	ham	what do u reckon as need 2 arrange transport i...
4176	spam	free entry into our £250 weekly competition ju...
4177	spam	-pls stop bootydelious (32/f) is inviting you ...
4178	ham	tell my bad character which u dnt lik in me. ...

4179 rows × 2 columns

Next steps: Generate code with train_file | View recommended plots | New interactive sheet

```
test_file = pd.read_csv(test_file_path, sep='\t', names=names)
test_file
```

	class	message
0	ham	i am in hospital da. . i will return home in e...
1	ham	not much, just some textin'. how bout you?
2	ham	i probably won't eat at all today. i think i'm...
3	ham	don't give a flying monkeys wot they think and...
4	ham	who are you seeing?
...	...	...
1387	ham	true dear..i sat to pray evening and felt so.s...
1388	ham	what will we do in the shower, baby?
1389	ham	where are you ? what are you doing ? are yuou ...
1390	spam	ur cash-balance is currently 500 pounds - to m...
1391	spam	not heard from u4 a while. call 4 rude chat pr...

1392 rows × 2 columns

Next steps: Generate code with test_file | View recommended plots | New interactive sheet

```
train_message = train_file["message"].values.tolist()
train_label = np.array([0 if x=="ham" else 1 for x in train_file['class'].values.tolist()])
test_message = test_file["message"].values.tolist()
test_label = np.array([0 if x=="ham" else 1 for x in test_file['class'].values.tolist()])
```

```
vocabulary_dict = {}
for messgae in train_message:
  for vocabulary in messgae.split():
    if vocabulary not in vocabulary_dict:
      vocabulary_dict[vocabulary] = 1
    else:
      vocabulary_dict[vocabulary] += 1
```

```
VOCAB_SIZE = len(vocabulary_dict)
MAX_LENGTH = len(max(train_message, key=lambda p: len(p.split())).split())
```

```
encoded_train_message = [one_hot(d, VOCAB_SIZE) for d in train_message]
padded_train_message = pad_sequences(encoded_train_message, maxlen=MAX_LENGTH, padding='post')
encoded_test_message = [one_hot(d, VOCAB_SIZE) for d in test_message]
padded_test_message = pad_sequences(encoded_test_message, maxlen=MAX_LENGTH, padding='post')
```

```
model = Sequential()
embedding_layer = Embedding(VOCAB_SIZE, 100, input_length=MAX_LENGTH)
model.add(embedding_layer)
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
monitor = EarlyStopping(monitor='val_acc', min_delta=1e-4, patience=25, verbose=1, mode='max', restore_best_weights=True)
model.fit(padded_train_message, train_label, validation_data=(padded_test_message, test_label), callbacks=[monitor], epochs=1000, verbose=2)
```

```
Epoch 14/1000
131/131 - 3s - 24ms/step - acc: 1.0000 - loss: 5.7181e-04 - val_acc: 0.9864 - val_loss: 0.0478
Epoch 15/1000
131/131 - 2s - 18ms/step - acc: 1.0000 - loss: 5.4573e-04 - val_acc: 0.9878 - val_loss: 0.0459
Epoch 16/1000
131/131 - 2s - 13ms/step - acc: 1.0000 - loss: 3.8771e-04 - val_acc: 0.9878 - val_loss: 0.0455
Epoch 17/1000
131/131 - 2s - 13ms/step - acc: 1.0000 - loss: 3.0095e-04 - val_acc: 0.9864 - val_loss: 0.0496
Epoch 18/1000
131/131 - 2s - 13ms/step - acc: 1.0000 - loss: 2.7559e-04 - val_acc: 0.9864 - val_loss: 0.0511
Epoch 19/1000
131/131 - 3s - 19ms/step - acc: 1.0000 - loss: 2.3664e-04 - val_acc: 0.9864 - val_loss: 0.0507
Epoch 20/1000
131/131 - 3s - 24ms/step - acc: 1.0000 - loss: 1.9690e-04 - val_acc: 0.9871 - val_loss: 0.0480
Epoch 21/1000
131/131 - 2s - 19ms/step - acc: 1.0000 - loss: 1.6410e-04 - val_acc: 0.9864 - val_loss: 0.0518
Epoch 22/1000
131/131 - 2s - 13ms/step - acc: 1.0000 - loss: 1.5212e-04 - val_acc: 0.9864 - val_loss: 0.0516
Epoch 23/1000
131/131 - 3s - 20ms/step - acc: 1.0000 - loss: 1.3042e-04 - val_acc: 0.9871 - val_loss: 0.0512
Epoch 24/1000
131/131 - 3s - 20ms/step - acc: 1.0000 - loss: 1.1798e-04 - val_acc: 0.9864 - val_loss: 0.0521
Epoch 25/1000
131/131 - 2s - 13ms/step - acc: 1.0000 - loss: 1.0355e-04 - val_acc: 0.9864 - val_loss: 0.0521
Epoch 26/1000
131/131 - 3s - 25ms/step - acc: 1.0000 - loss: 9.2725e-05 - val_acc: 0.9864 - val_loss: 0.0535
Epoch 27/1000
131/131 - 4s - 32ms/step - acc: 1.0000 - loss: 8.1342e-05 - val_acc: 0.9864 - val_loss: 0.0550
Epoch 28/1000
131/131 - 3s - 20ms/step - acc: 1.0000 - loss: 7.4071e-05 - val_acc: 0.9864 - val_loss: 0.0527
Epoch 29/1000
131/131 - 2s - 19ms/step - acc: 1.0000 - loss: 6.6748e-05 - val_acc: 0.9864 - val_loss: 0.0556
Epoch 30/1000
131/131 - 2s - 17ms/step - acc: 1.0000 - loss: 6.0522e-05 - val_acc: 0.9864 - val_loss: 0.0543
Epoch 31/1000
131/131 - 3s - 20ms/step - acc: 1.0000 - loss: 5.4127e-05 - val_acc: 0.9864 - val_loss: 0.0565
Epoch 32/1000
131/131 - 2s - 16ms/step - acc: 1.0000 - loss: 5.0350e-05 - val_acc: 0.9864 - val_loss: 0.0563
Epoch 33/1000
131/131 - 2s - 18ms/step - acc: 1.0000 - loss: 4.6001e-05 - val_acc: 0.9864 - val_loss: 0.0552
Epoch 34/1000
131/131 - 3s - 20ms/step - acc: 1.0000 - loss: 4.0289e-05 - val_acc: 0.9864 - val_loss: 0.0577
Epoch 35/1000
131/131 - 3s - 20ms/step - acc: 1.0000 - loss: 3.7984e-05 - val_acc: 0.9864 - val_loss: 0.0584
Epoch 36/1000
131/131 - 4s - 27ms/step - acc: 1.0000 - loss: 3.4764e-05 - val_acc: 0.9864 - val_loss: 0.0570
Epoch 37/1000
131/131 - 4s - 30ms/step - acc: 1.0000 - loss: 3.1077e-05 - val_acc: 0.9864 - val_loss: 0.0589
Epoch 38/1000
131/131 - 2s - 13ms/step - acc: 1.0000 - loss: 2.9050e-05 - val_acc: 0.9864 - val_loss: 0.0585
Epoch 39/1000
131/131 - 3s - 20ms/step - acc: 1.0000 - loss: 2.6780e-05 - val_acc: 0.9864 - val_loss: 0.0577
Epoch 40/1000
131/131 - 2s - 13ms/step - acc: 1.0000 - loss: 2.4374e-05 - val_acc: 0.9864 - val_loss: 0.0591
Epoch 40: early stopping
Restoring model weights from the end of the best epoch: 15.
<keras.src.callbacks.history.History at 0x7a5294352800>
```

```
# function to predict messages based on model
def predict_message(pred_text):
  class_dict = {
      0 : "ham",
      1 : "spam",
      }
  encoded_message = [one_hot(pred_text, VOCAB_SIZE)]
  padded_message = pad_sequences(encoded_message, maxlen=MAX_LENGTH, padding='post')
  prediction = [model.predict(padded_message)[0][0], class_dict[np.round(model.predict(padded_message)[0][0])]]
  return prediction

pred_text = "how are you doing today?"
```

```
prediction = predict_message(pred_text)
print(prediction)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
[0.00012277569, 'ham']
```

```
def test_predictions():
  test_messages = ["how are you doing today",
                   "sale today! to stop texts call 98912460324",
                   "i dont want to go. can we try it a different day? available sat",
                   "our new mobile video service is live. just install on your phone to start watching.",
                   "you have won £1000 cash! call to claim your prize.",
                   "i'll bring it tomorrow. don't forget the milk.",
                   "wow, is your arm alright. that happened to me one time too"
                  ]

  test_answers = ["ham", "spam", "ham", "spam", "spam", "ham", "ham"]
  passed = True

  for msg, ans in zip(test_messages, test_answers):
    prediction = predict_message(msg)
    if prediction[1] != ans:
      passed = False

  if passed:
    print("You passed the challenge. Great job!")
  else:
    print("You haven't passed yet. Keep trying.")

test_predictions()
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
You haven't passed yet. Keep trying.
```