```
# import libraries
try:
  # %tensorflow_version only exists in Colab.
  !pip install tf-nightly
except Exception:
  pass
import tensorflow as tf
import pandas as pd
from tensorflow import keras
!pip install tensorflow-datasets
import tensorflow_datasets as tfds
import numpy as np
import matplotlib.pyplot as plt

print(tf.__version__)
```

```
Requirement already satisfied: tf-nightly in /usr/local/lib/python3.11/dist-packages (2.20.0.dev20250410)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (25.2.10)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (0.6.
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (24.2)
Requirement already satisfied: protobuf<6.0.0dev,>=4.21.6 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (5.29.4)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (75.2.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (3.0.1)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (4.13.1)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (1.71.0)
Requirement already satisfied: tb-nightly~=2.19.0.a in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (2.19.0a20250218)
Requirement already satisfied: keras-nightly>=3.6.0.dev in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (3.9.0.dev202503
Requirement already satisfied: numpy<2.2.0,>=1.26.0 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (2.0.2)
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (3.13.0)
Requirement already satisfied: ml-dtypes<1.0.0,>=0.5.1 in /usr/local/lib/python3.11/dist-packages (from tf-nightly) (0.5.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0->tf-nightly) (0.
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras-nightly>=3.6.0.dev->tf-nightly) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras-nightly>=3.6.0.dev->tf-nightly) (0.0.8)
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras-nightly>=3.6.0.dev->tf-nightly) (0.14.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tf-nigh
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tf-nightly) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tf-nightly) (
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tf-nightly) (
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-packages (from tb-nightly~=2.19.0.a->tf-nightly) (3.
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from tb-nightly~=2.1
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tb-nightly~=2.19.0.a->tf-nightly) (3.
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1->tb-nightly~=2.19.0
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras-nightly>=3.6.0.dev-
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras-nightly>=3.6.0.de
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich->keras-nightly
Requirement already satisfied: tensorflow-datasets in /usr/local/lib/python3.11/dist-packages (4.9.8)
Requirement already satisfied: absl-py in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (1.4.0)
Requirement already satisfied: array_record>=0.5.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (0.7.1)
Requirement already satisfied: dm-tree in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (0.1.9)
Requirement already satisfied: etils>=1.9.1 in /usr/local/lib/python3.11/dist-packages (from etils[edc,enp,epath,epy,etree]>=1.9.1; p
Requirement already satisfied: immutabledict in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (4.2.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (2.0.2)
Requirement already satisfied: promise in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (2.3)
Requirement already satisfied: protobuf>=3.20 in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (5.29.4)
Requirement already satisfied: psutil in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (5.9.5)
Requirement already satisfied: pyarrow in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (18.1.0)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (2.32.3)
Requirement already satisfied: simple_parsing in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (0.1.7)
Requirement already satisfied: tensorflow-metadata in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (1.17.0)
Requirement already satisfied: termcolor in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (3.0.1)
Requirement already satisfied: toml in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (0.10.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (4.67.1)
Requirement already satisfied: wrapt in /usr/local/lib/python3.11/dist-packages (from tensorflow-datasets) (1.17.2)
Requirement already satisfied: einops in /usr/local/lib/python3.11/dist-packages (from etils[edc,enp,epath,epy,etree]>=1.9.1; python_
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from etils[edc,enp,epath,epy,etree]>=1.9.1; python_
```

```
# 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"
```

```
--2025-04-10 21:24:05--  https://cdn.freecodecamp.org/project-data/sms/train-data.tsv
Resolving cdn.freecodecamp.org (cdn.freecodecamp.org)... 104.26.2.33, 172.67.70.149, 104.26.3.33, ...
Connecting to cdn.freecodecamp.org (cdn.freecodecamp.org)|104.26.2.33|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 358233 (350K) [text/tab-separated-values]
Saving to: ‘train-data.tsv.6’

train-data.tsv.6    100%[===================>] 349.84K  --.-KB/s    in 0.04s

2025-04-10 21:24:05 (9.65 MB/s) - ‘train-data.tsv.6’ saved [358233/358233]

--2025-04-10 21:24:05--  https://cdn.freecodecamp.org/project-data/sms/valid-data.tsv
Resolving cdn.freecodecamp.org (cdn.freecodecamp.org)... 104.26.2.33, 172.67.70.149, 104.26.3.33, ...
Connecting to cdn.freecodecamp.org (cdn.freecodecamp.org)|104.26.2.33|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 118774 (116K) [text/tab-separated-values]
Saving to: ‘valid-data.tsv.6’

valid-data.tsv.6    100%[===================>] 115.99K  --.-KB/s    in 0.02s

2025-04-10 21:24:06 (5.14 MB/s) - ‘valid-data.tsv.6’ saved [118774/118774]
```

```
train_data = pd.read_csv(train_file_path, sep="\t", header=None, names=["label", "message"])
test_data = pd.read_csv(test_file_path, sep="\t", header=None, names=["label", "message"])
train_data.head()
```

	label	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 ...

Étapes suivantes : Générer du code avec `train_data` | Afficher les graphiques recommandés | New interactive sheet

```
train_data['label'] = train_data['label'].map({'ham': 0, 'spam': 1})
test_data['label'] = test_data['label'].map({'ham': 0, 'spam': 1})

X_train = train_data['message'].values
y_train = train_data['label'].values

X_test = test_data['message'].values
y_test = test_data['label'].values

max_features = 5000
sequence_length = 100

vectorize_layer = tf.keras.layers.TextVectorization(
    max_tokens=max_features,
    output_mode='int',
    output_sequence_length=sequence_length,
    standardize='lower_and_strip_punctuation'
    )

vectorize_layer.adapt(X_train)
```

```
model = tf.keras.Sequential([
    vectorize_layer,
    tf.keras.layers.Embedding(max_features + 1, 16),
    tf.keras.layers.GlobalAveragePooling1D(),
    tf.keras.layers.Dense(16, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

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

```
history = model.fit(
    X_train,
    y_train,
    epochs=30,
    validation_data=(X_test, y_test)
)
```

```
Epoch 2/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.8714 - loss: 0.3553 - val_accuracy: 0.8657 - val_loss: 0.3602
Epoch 3/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.8736 - loss: 0.3450 - val_accuracy: 0.8657 - val_loss: 0.3486
Epoch 4/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.8718 - loss: 0.3367 - val_accuracy: 0.8657 - val_loss: 0.3304
Epoch 5/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.8663 - loss: 0.3202 - val_accuracy: 0.8678 - val_loss: 0.2652
Epoch 6/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.8695 - loss: 0.2501 - val_accuracy: 0.9440 - val_loss: 0.1772
Epoch 7/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9477 - loss: 0.1544 - val_accuracy: 0.9612 - val_loss: 0.1224
Epoch 8/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9700 - loss: 0.1013 - val_accuracy: 0.9605 - val_loss: 0.1193
Epoch 9/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9795 - loss: 0.0815 - val_accuracy: 0.9713 - val_loss: 0.0863
Epoch 10/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9753 - loss: 0.0785 - val_accuracy: 0.9720 - val_loss: 0.0828
Epoch 11/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 2s 9ms/step - accuracy: 0.9833 - loss: 0.0590 - val_accuracy: 0.9720 - val_loss: 0.0731
Epoch 12/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - accuracy: 0.9836 - loss: 0.0496 - val_accuracy: 0.9741 - val_loss: 0.0682
Epoch 13/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9858 - loss: 0.0537 - val_accuracy: 0.9763 - val_loss: 0.0660
Epoch 14/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9879 - loss: 0.0475 - val_accuracy: 0.9784 - val_loss: 0.0620
Epoch 15/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9863 - loss: 0.0453 - val_accuracy: 0.9799 - val_loss: 0.0598
Epoch 16/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9879 - loss: 0.0378 - val_accuracy: 0.9784 - val_loss: 0.0585
Epoch 17/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9881 - loss: 0.0373 - val_accuracy: 0.9784 - val_loss: 0.0571
Epoch 18/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9926 - loss: 0.0288 - val_accuracy: 0.9799 - val_loss: 0.0594
Epoch 19/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9914 - loss: 0.0258 - val_accuracy: 0.9777 - val_loss: 0.0616
Epoch 20/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9940 - loss: 0.0252 - val_accuracy: 0.9784 - val_loss: 0.0661
Epoch 21/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9954 - loss: 0.0221 - val_accuracy: 0.9777 - val_loss: 0.0674
Epoch 22/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.9924 - loss: 0.0273 - val_accuracy: 0.9813 - val_loss: 0.0518
Epoch 23/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - accuracy: 0.9935 - loss: 0.0219 - val_accuracy: 0.9820 - val_loss: 0.0515
Epoch 24/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 2s 6ms/step - accuracy: 0.9956 - loss: 0.0213 - val_accuracy: 0.9799 - val_loss: 0.0567
Epoch 25/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9916 - loss: 0.0222 - val_accuracy: 0.9828 - val_loss: 0.0535
Epoch 26/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9936 - loss: 0.0221 - val_accuracy: 0.9835 - val_loss: 0.0505
Epoch 27/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9963 - loss: 0.0166 - val_accuracy: 0.9820 - val_loss: 0.0495
Epoch 28/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9955 - loss: 0.0167 - val_accuracy: 0.9792 - val_loss: 0.0586
Epoch 29/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 2s 10ms/step - accuracy: 0.9927 - loss: 0.0214 - val_accuracy: 0.9835 - val_loss: 0.0499
Epoch 30/30
131/131 ━━━━━━━━━━━━━━━━━━━━ 3s 12ms/step - accuracy: 0.9970 - loss: 0.0155 - val_accuracy: 0.9741 - val_loss: 0.0699
```

```
def predict_message(msg):
    input_text = tf.convert_to_tensor([msg])
    prediction = model.predict(input_text)[0][0]
    label = "spam" if prediction > 0.5 else "ham"
    return [float(prediction), label]


# Run this cell to test your function and model. Do not modify contents.
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 140ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
You passed the challenge. Great job!
```