```
import pandas as pd
import numpy as np



import tensorflow as tf

# helps in text preprocessing

tf.config.run_functions_eagerly(True)
from tensorflow import keras
from keras.preprocessing import text, sequence
from keras import utils

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

# helps in model building
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import SimpleRNN
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Embedding
from tensorflow.keras.callbacks import EarlyStopping

import nltk
!pip install nltk
nltk.download('punkt')

from nltk.corpus import stopwords

nltk.download('stopwords')
import string
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()

# split data into train and test set
from sklearn.model_selection import train_test_split
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.6)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.1)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2022.10.31)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.65.0)
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```
true_df = pd.read_csv("True.csv")
false_df = pd.read_csv("Fake.csv")
```

```
true_df
```

	title	subject
0	U.S. bill to regulate internet ads gains bipar...	politicsNews
1	Republican Senator Hatch rejects bipartisan he...	politicsNews
2	Trump to announce Fed chair decision in 'comin...	politicsNews
3	Trump does not support Alexander-Murray health...	politicsNews
4	Pentagon chief asks Congress to not hinder cyb...	politicsNews
...	...	...
10596	'Fully committed' NATO backs new U.S. approach...	worldnews
10597	LexisNexis withdrew two products from Chinese ...	worldnews
10598	Minsk cultural hub becomes haven from authorities	worldnews
10599	Vatican upbeat on possibility of Pope Francis ...	worldnews
10600	Indonesia to buy $1.14 billion worth of Russia...	worldnews

10601 rows × 2 columns

```
false_df
```

	title	subject
0	The Leaker-In-Chief Just Tweeted Classified I...	News
1	Senator Who Has Accomplished Nothing Is Very ...	News
2	Breaking: MAJOR Action Taking Place At Trump ...	News
3	Three Days Before Trump Calls For More Police...	News
4	Official GOP Twitter Account Went Fully Delus...	News
...	...	...
18301	McPain: John McCain Furious That Iran Treated ...	MiddleEast
18302	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	MiddleEast
18303	Sunnistan: US and Allied ‘Safe Zone’ Plan to T...	MiddleEast
18304	How to Blow $700 Million: Al Jazeera America F...	MiddleEast
18305	10 U.S. Navy Sailors Held by Iranian Military ...	MiddleEast

18306 rows × 2 columns

```
true_df['class'] = 1
false_df['class'] = 0
```

```
true_df
```

	title	subject	class
0	U.S. bill to regulate internet ads gains bipar...	politicsNews	1
1	Republican Senator Hatch rejects bipartisan he...	politicsNews	1
2	Trump to announce Fed chair decision in 'comin...	politicsNews	1
3	Trump does not support Alexander-Murray health...	politicsNews	1
4	Pentagon chief asks Congress to not hinder cyb...	politicsNews	1
...	...	...	...
10596	'Fully committed' NATO backs new U.S. approach...	worldnews	1
10597	LexisNexis withdrew two products from Chinese ...	worldnews	1
10598	Minsk cultural hub becomes haven from authorities	worldnews	1
10599	Vatican upbeat on possibility of Pope Francis ...	worldnews	1
10600	Indonesia to buy $1.14 billion worth of Russia...	worldnews	1

10601 rows × 3 columns

```
false_df
```

	title	subject	class
0	The Leaker-In-Chief Just Tweeted Classified I...	News	0
1	Senator Who Has Accomplished Nothing Is Very ...	News	0
2	Breaking: MAJOR Action Taking Place At Trump ...	News	0
3	Three Days Before Trump Calls For More Police...	News	0
4	Official GOP Twitter Account Went Fully Delus...	News	0
...	...	...	...
18301	McPain: John McCain Furious That Iran Treated ...	MiddleEast	0
18302	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	MiddleEast	0
18303	Sunnistan: US and Allied ‘Safe Zone’ Plan to T...	MiddleEast	0
18304	How to Blow $700 Million: Al Jazeera America F...	MiddleEast	0
18305	10 U.S. Navy Sailors Held by Iranian Military ...	MiddleEast	0

18306 rows × 3 columns

```
news_df = pd.concat([true_df,false_df])
```

```
news_df
```

	title	subject	class
0	U.S. bill to regulate internet ads gains bipar...	politicsNews	1
1	Republican Senator Hatch rejects bipartisan he...	politicsNews	1
2	Trump to announce Fed chair decision in 'comin...	politicsNews	1
3	Trump does not support Alexander-Murray health...	politicsNews	1
4	Pentagon chief asks Congress to not hinder cyb...	politicsNews	1
...	...	...	...
18301	McPain: John McCain Furious That Iran Treated ...	MiddleEast	0
18302	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	MiddleEast	0
18303	Sunnistan: US and Allied 'Safe Zone' Plan to T...	MiddleEast	0
18304	How to Blow $700 Million: Al Jazeera America F...	MiddleEast	0
18305	10 U.S. Navy Sailors Held by Iranian Military ...	MiddleEast	0

28907 rows × 3 columns

```
news_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 28907 entries, 0 to 18305
Data columns (total 3 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   title    28907 non-null  object
 1   subject  28907 non-null  object
 2   class    28907 non-null  int64 
dtypes: int64(1), object(2)
memory usage: 903.3+ KB
```

```
news_df.isnull().sum()
```

```
title      0
subject    0
class      0
dtype: int64
```

```
def transform_text(text):
    text = text.lower()
    text = nltk.word_tokenize(text)

    y = []
    for i in text:
        if i.isalnum():
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        if i not in stopwords.words('english') and i not in string.punctuation:
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        y.append(ps.stem(word=i))


    return " ".join(y)
```

```
news_df['subject'].value_counts()
```

```
politicsNews       9208
News               8084
politics           4504
LeftNews           2910
Government News    1570
worldnews          1393
MiddleEast          711
```

```
USNews              527
Name: subject, dtype: int64
```

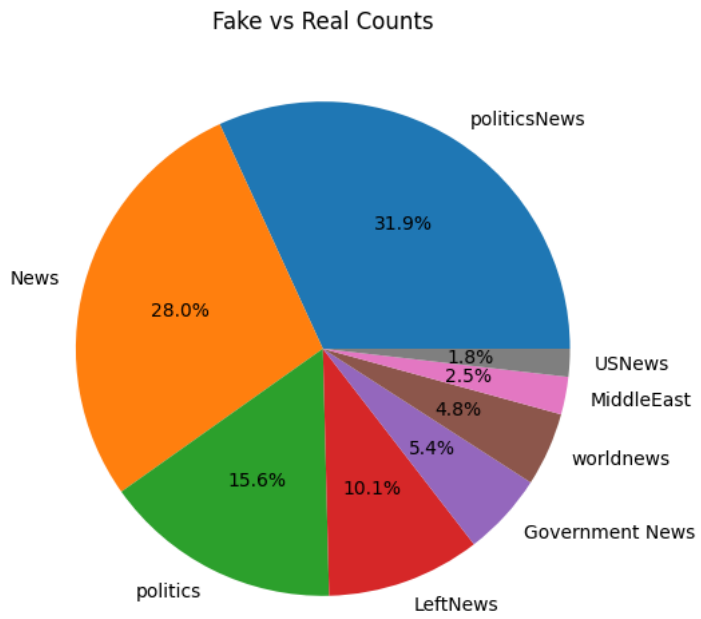
```
import matplotlib.pyplot as plt

fake_true_counts = news_df['subject'].value_counts().rename_axis('subject').to_frame('counts').reset_index()
fake_true_counts['subject'] = fake_true_counts['subject'].replace({1: 'Fake', 0: 'True'})

# Set the figure size
plt.figure(figsize=(8, 6))

# Create a pie plot of the counts
ax = plt.pie(fake_true_counts['counts'], labels=fake_true_counts['subject'], autopct='%1.1f%%')
plt.title('Fake vs Real Counts')

# Show the plot
plt.show()
```




```
news_df
```

	title	subject	class
0	U.S. bill to regulate internet ads gains bipar...	politicsNews	1
1	Republican Senator Hatch rejects bipartisan he...	politicsNews	1
2	Trump to announce Fed chair decision in 'comin...	politicsNews	1
3	Trump does not support Alexander-Murray health...	politicsNews	1
4	Pentagon chief asks Congress to not hinder cyb...	politicsNews	1
...	...	...	...
18301	McPain: John McCain Furious That Iran Treated ...	MiddleEast	0
18302	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	MiddleEast	0
18303	Sunnistan: US and Allied ‘Safe Zone’ Plan to T...	MiddleEast	0
18304	How to Blow $700 Million: Al Jazeera America F...	MiddleEast	0
18305	10 U.S. Navy Sailors Held by Iranian Military ...	MiddleEast	0

28907 rows × 3 columns

```
news_df['title_lower']=news_df['title'].apply(transform_text)
news_df['subject_lower']=news_df['subject'].apply(transform_text)
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
#creating vectorizer
vectorizer = TfidfVectorizer(ngram_range=(1,2),max_features=2000)
```

```
X = news_df[['title',"subject"]].values
y = news_df['class'].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42)


!pip install --upgrade tensorflow
!pip install --upgrade keras
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-packages
Collecting tensorflow
  Downloading tensorflow-2.13.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 524.1/524.1 MB 3.1 MB/s eta 0:00:00
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: flatbuffers>=23.1.21 in /usr/local/lib/python3.10/dis
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.10/dist
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-package
Collecting keras<2.14,>=2.13.1 (from tensorflow)
  Downloading keras-2.13.1-py3-none-any.whl (1.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 48.1 MB/s eta 0:00:00
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: numpy<=1.24.3,>=1.22 in /usr/local/lib/python3.10/dis
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-package
Collecting tensorboard<2.14,>=2.13 (from tensorflow)
  Downloading tensorboard-2.13.0-py3-none-any.whl (5.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.6/5.6 MB 34.6 MB/s eta 0:00:00
Collecting tensorflow-estimator<2.14,>=2.13.0 (from tensorflow)
  Downloading tensorflow_estimator-2.13.0-py2.py3-none-any.whl (440 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 440.8/440.8 kB 42.8 MB/s eta 0:00:00
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-pa
Collecting typing-extensions<4.6.0,>=3.6.6 (from tensorflow)
  Downloading typing_extensions-4.5.0-py3-none-any.whl (27 kB)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/li
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/di
Requirement already satisfied: google-auth-oauthlib<1.1,>=0.5 in /usr/local/lib/pyth
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/l
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/d
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/di
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/di
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.1
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in /usr/local/lib/python3.10/dis
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-pac
Installing collected packages: typing-extensions, tensorflow-estimator, keras, tenso
  Attempting uninstall: typing-extensions
    Found existing installation: typing_extensions 4.7.1
    Uninstalling typing_extensions-4.7.1:
      Successfully uninstalled typing_extensions-4.7.1
  Attempting uninstall: tensorflow-estimator
    Found existing installation: tensorflow-estimator 2.12.0
    Uninstalling tensorflow-estimator-2.12.0:
      Successfully uninstalled tensorflow-estimator-2.12.0
  Attempting uninstall: keras
    Found existing installation: keras 2.12.0
    Uninstalling keras-2.12.0:
      Successfully uninstalled keras-2.12.0
  Attempting uninstall: tensorboard
    Found existing installation: tensorboard 2.12.3
    Uninstalling tensorboard-2.12.3:
      Successfully uninstalled tensorboard-2.12.3
  Attempting uninstall: tensorflow
    Found existing installation: tensorflow 2.12.0
    Uninstalling tensorflow-2.12.0:
      Successfully uninstalled tensorflow-2.12.0
Successfully installed keras-2.13.1 tensorboard-2.13.0 tensorflow-2.13.0 tensorflow-
WARNING: The following packages were previously imported in this runtime:
  [keras,tensorboard,tensorflow]
You must restart the runtime in order to use newly installed versions.
```

RESTART RUNTIME

```
Requirement already satisfied: keras in /usr/local/lib/python3.10/dist-packages (2.1
```

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split

# Assuming X_train and X_test are NumPy arrays of strings
X_train_list = X_train.astype(str).tolist()
X_test_list = X_test.astype(str).tolist()

# Create a tokenizer and fit on the texts
t = Tokenizer()
t.fit_on_texts(X_train_list)

# Convert the text sequences to numerical sequences
encoded_train = t.texts_to_sequences(X_train_list)
encoded_test = t.texts_to_sequences(X_test_list)

# Pad the sequences to make them of equal length
max_sequence_length = max(len(seq) for seq in encoded_train)
encoded_train_padded = pad_sequences(encoded_train, maxlen=max_sequence_length, padding='post')
encoded_test_padded = pad_sequences(encoded_test, maxlen=max_sequence_length, padding='post')


max_length = 8
padded_train = pad_sequences(encoded_train, maxlen=max_length, padding='post')
padded_test = pad_sequences(encoded_test, maxlen=max_length, padding='post')
print(padded_train)
```

```
[[ 1905     2     0 ...     0     0     0]
 [ 1906     4     0 ...     0     0     0]
 [ 1907     1     0 ...     0     0     0]
 ...
 [21086     1     0 ...     0     0     0]
 [21087     2     0 ...     0     0     0]
 [21088     5     0 ...     0     0     0]]
```

```
### Will calculate vocabulary size , no of unique tokens(words/sentences) I have in my data
vocab_size = len(t.word_index) + 1
## Will be used as input_dim for Embeddings


# define the model
model = Sequential()
model.add(Embedding(vocab_size, 24, input_length=max_length))

## Why embedding ? To add contextual vectorizartion to inputs i.e each input will be represented as a dim vector of size
### input_dim :- Unique tokens in vocabulary
### output dim :- 24 (Each word will be represented in 24 *1 vector)
model.add(SimpleRNN(24, return_sequences=False))
## As each input is converted into embedding of 24 tokens , input for simple RNN is 24)
model.add(Dense(1, activation='sigmoid'))

# compile the model
model.compile(optimizer='rmsprop', loss='binary_crossentropy', metrics=['accuracy'])

# summarize the model
print(model.summary())
```

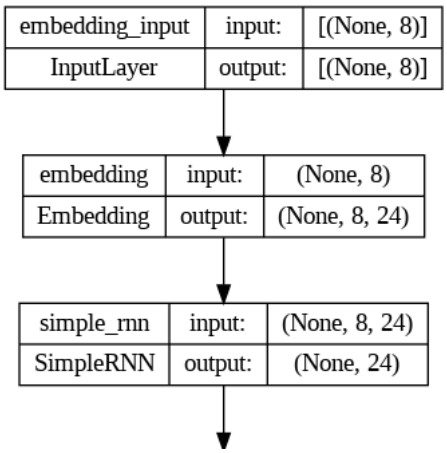
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 8, 24)             506136

 simple_rnn (SimpleRNN)      (None, 24)                1176

 dense (Dense)               (None, 1)                 25

=================================================================
Total params: 507,337
Trainable params: 507,337
Non-trainable params: 0
_________________________________________________________________
None
```

```
from keras.utils import plot_model
plot_model(model, to_file='model_1.png', show_shapes=True, show_layer_names=True)
```




```
early_stop = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=10)

## Early stop will help us prevent early stopping , it will monitor Validation loss with patience =10
## Interpretation :-
# monitor: Specifies the quantity to monitor during training. In this case, 'val_loss' is used, indicating that the validation loss will
# mode: Specifies the direction of improvement to consider as an improvement. The 'min' mode is used here, meaning that the validation l
# verbose: Controls the verbosity mode. Setting it to 1 means that a message will be printed to the console when the callback is trigger
# patience: Specifies the number of epochs to wait for improvement in the monitored quantity before stopping the training. In this case,

# fit the model
model.fit(x=padded_train,
          y=y_train,
          epochs=100,
          validation_data=(padded_test, y_test), verbose=1,
          callbacks=[early_stop]
          )
```

```
Epoch 99/100
723/723 [==============================] - 33s 46ms/step - loss: 1.9288e-07 - accuracy: 1.0000 - val_loss: 0.4153 - val_accuracy
Epoch 100/100
723/723 [==============================] - 35s 48ms/step - loss: 1.9092e-07 - accuracy: 1.0000 - val_loss: 0.4151 - val_accuracy
<keras.callbacks.History at 0x7c0b743dbee0>
```

```
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

def c_report(y_true, y_pred):
   print("Classification Report")
   print(classification_report(y_true, y_pred))
   acc_sc = accuracy_score(y_true, y_pred)
   print("Accuracy : "+ str(acc_sc))
   return acc_sc

def plot_confusion_matrix(y_true, y_pred):
   mtx = confusion_matrix(y_true, y_pred)
   sns.heatmap(mtx, annot=True, fmt='d', linewidths=.5,
               cmap="Blues", cbar=False)
   plt.ylabel('True label')
   plt.xlabel('Predicted label')
```

```
preds = (model.predict(padded_test) > 0.5).astype("int32")
```

```
181/181 [==============================] - 3s 14ms/step
```

```
c_report(y_test, preds)
```

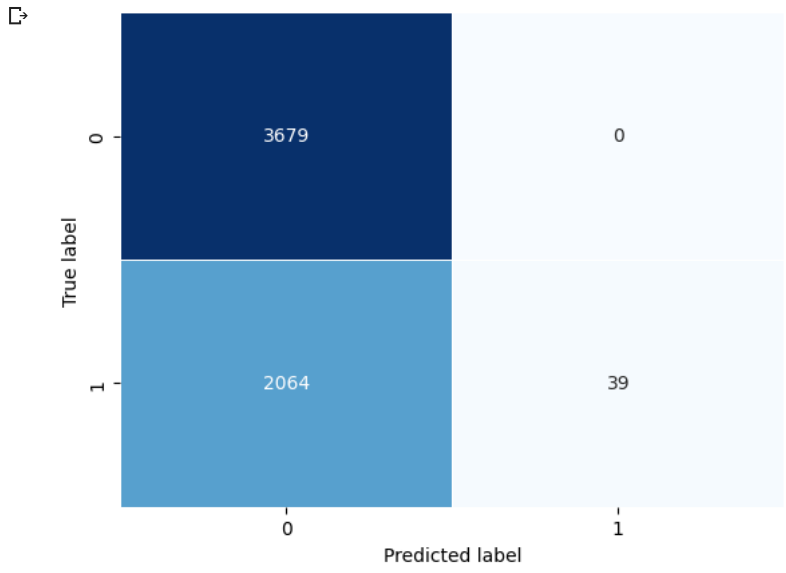
```
Classification Report
              precision    recall  f1-score   support

           0       0.64      1.00      0.78      3679
           1       1.00      0.02      0.04      2103

    accuracy                           0.64      5782
   macro avg       0.82      0.51      0.41      5782
weighted avg       0.77      0.64      0.51      5782

Accuracy : 0.6430300933932895
0.6430300933932895
```

```
import seaborn as sns
import matplotlib.pyplot as plt
plot_confusion_matrix(y_test, preds)
```




```
from tensorflow.keras.preprocessing.text import Tokenizer

sms = ["Fans line up for Taylor Swift merchandise on the second SoFi tour day"]
sms_proc = t.texts_to_sequences(sms)
sms_proc = pad_sequences(sms_proc, maxlen=max_length, padding='post')
pred = (model.predict(sms_proc)>0.5).astype("int32").item()
print(pred)
```

```
1/1 [==============================] - 0s 28ms/step
0
```