```
!unzip '/content/SMS dataset.zip'
```

```
Archive:  /content/SMS dataset.zip
  inflating: spam.csv
```

```
import numpy as np
import pandas as pd
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM,Dense
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing import sequence
from tensorflow.keras.optimizers import RMSprop
```

```
df = pd.read_csv('/content/spam.csv', delimiter=',', encoding='latin-
1')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

```
df.head()
```

```
     v1                                                 v2 Unnamed: 2
\
0   ham  Go until jurong point, crazy.. Available only ...        NaN

1   ham                      Ok lar... Joking wif u oni...        NaN

2  spam  Free entry in 2 a wkly comp to win FA Cup fina...        NaN

3   ham  U dun say so early hor... U c already then say...        NaN

4   ham  Nah I don't think he goes to usf, he lives aro...        NaN


  Unnamed: 3 Unnamed: 4
0        NaN        NaN
```

```
1        NaN        NaN
2        NaN        NaN
3        NaN        NaN
4        NaN        NaN
```

```
df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1,
inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   v1      5572 non-null   object
 1   v2      5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

```
x = df.v2
y = df.v1
encoder = LabelEncoder()
y = encoder.fit_transform(y)
y = y.reshape(-1,1)
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2)
tokenizer = Tokenizer(num_words=2000, lower=True)
tokenizer.fit_on_texts(x_train)
sequences = tokenizer.texts_to_sequences(x_train)
x_train = sequence.pad_sequences(sequences, maxlen=200)
```

```
model = Sequential()
model.add(LSTM(50,input_shape=(x_train.shape[1],1),return_sequences=Tr
ue))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
```

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

```
model.fit(x_train, y_train, batch_size=128, epochs=10,
validation_split=0.2)
```

```
Epoch 1/10
28/28 [==============================] - 35s 835ms/step - loss: 2.1547
- accuracy: 0.8603 - val_loss: 1.7465 - val_accuracy: 0.8868
Epoch 2/10
28/28 [==============================] - 22s 787ms/step - loss: 2.1547
- accuracy: 0.8603 - val_loss: 1.7465 - val_accuracy: 0.8868
```

```
Epoch 3/10
28/28 [==============================] - 21s 739ms/step - loss: 2.1547
- accuracy: 0.8603 - val_loss: 1.7465 - val_accuracy: 0.8868
Epoch 4/10
28/28 [==============================] - 21s 761ms/step - loss: 2.1547
- accuracy: 0.8603 - val_loss: 1.7465 - val_accuracy: 0.8868
Epoch 5/10
28/28 [==============================] - 22s 779ms/step - loss: 2.1547
- accuracy: 0.8603 - val_loss: 1.7465 - val_accuracy: 0.8868
Epoch 6/10
28/28 [==============================] - 22s 783ms/step - loss: 2.1547
- accuracy: 0.8603 - val_loss: 1.7465 - val_accuracy: 0.8868
Epoch 7/10
28/28 [==============================] - 22s 786ms/step - loss: 2.1547
- accuracy: 0.8603 - val_loss: 1.7465 - val_accuracy: 0.8868
Epoch 8/10
28/28 [==============================] - 21s 737ms/step - loss: 2.1547
- accuracy: 0.8603 - val_loss: 1.7465 - val_accuracy: 0.8868
Epoch 9/10
28/28 [==============================] - 21s 740ms/step - loss: 2.1547
- accuracy: 0.8603 - val_loss: 1.7465 - val_accuracy: 0.8868
Epoch 10/10
28/28 [==============================] - 22s 786ms/step - loss: 2.1547
- accuracy: 0.8603 - val_loss: 1.7465 - val_accuracy: 0.8868
```

```
<keras.callbacks.History at 0x7f14b9e294d0>
```

```
model.save('lstm.h5')
```

```
test_sequences = tokenizer.texts_to_sequences(x_test)
x_test = sequence.pad_sequences(test_sequences, maxlen=100)
acc = model.evaluate(x_test, y_test)
```

```
35/35 [==============================] - 4s 50ms/step - loss: 1.9229 -
accuracy: 0.8753
```