Name	Ronald Jim Roy S
Register number	210519205041
Team size	4
Team ID	PNT2022TMID25098

Import libraries

```
import pandas as pd
import numpy as np
```

Read dataset

```
!unzip '/content/archive.zip'
```

```
Archive: /content/archive.zip
  inflating: spam.csv
```

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

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

Data preprocessing

```
df.isnull().sum()
```

```
v1            0
v2            0
Unnamed: 2    5522
Unnamed: 3    5560
Unnamed: 4    5566
dtype: int64
```

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

```
df.isnull().sum()
```

```
v1    0
v2    0
dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
df['v1'] = le.fit_transform(df['v1'])
df['v2'] = le.fit_transform(df['v2'])
```

```
df.head()
```

	v1	v2
0	0	1079
1	0	3101
2	1	1000

```
X = df.v2
y = df.v1
le = LabelEn oder()
y = le.fit_t ansform(y)
y = y.reshap (-1,1)
```

```
from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest = train_test_split(X,y,test_size=0.3,random_state=0)
```

```
xtrain.shape, xtest.shape
```

```
((3900,), (1672,))
```

Create model and add LSTM layers

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
```

```
model = Sequential()
model.add(LSTM(50, input_shape=(60, 1),return_sequences=True))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50,return_sequences=True))
model.add(Dense(1))
```

compile the model

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

Fit the model

```
model.fit(xtrain,ytrain,batch_size=30,epochs=10)
```

```
Epoch 1/10
WARNING:tensorflow:Model was constructed with shape (None, 60, 1) for input KerasTensor(type_spec=TensorSpec(shape=(None, 60, 1),
WARNING:tensorflow:Model was constructed with shape (None, 60, 1) for input KerasTensor(type_spec=TensorSpec(shape=(None, 60, 1),
130/130 [==============================] - 8s 6ms/step - loss: 0.1151 - accuracy: 0.8695
Epoch 2/10
130/130 [==============================] - 1s 6ms/step - loss: 0.1136 - accuracy: 0.8695
Epoch 3/10
130/130 [==============================] - 1s 6ms/step - loss: 0.1131 - accuracy: 0.8695
Epoch 4/10
130/130 [==============================] - 1s 6ms/step - loss: 0.1131 - accuracy: 0.8695
Epoch 5/10
130/130 [==============================] - 1s 6ms/step - loss: 0.1128 - accuracy: 0.8690
Epoch 6/10
130/130 [==============================] - 1s 6ms/step - loss: 0.1125 - accuracy: 0.8692
Epoch 7/10
130/130 [==============================] - 1s 6ms/step - loss: 0.1126 - accuracy: 0.8690
Epoch 8/10
130/130 [==============================] - 1s 7ms/step - loss: 0.1129 - accuracy: 0.8692
Epoch 9/10
130/130 [==============================] - 1s 6ms/step - loss: 0.1129 - accuracy: 0.8692
Epoch 10/10
130/130 [==============================] - 1s 7ms/step - loss: 0.1129 - accuracy: 0.8695
<keras.callbacks.History at 0x7f68c9823950>
```

Save the model

```
model.save('sms apam.h5')
```

Test the model

```
ypred = model.predict(xtest)
```

```
WARNING:tensorflow:Model was constructed with shape (None, 60, 1) for input KerasTensor(type_spec=TensorSpec(shape=(None, 60, 1),
53/53 [==============================] - 2s 3ms/step
```

```
ypred.flatten()
```

```
array([0.17988086, 0.1437762 , 0.1437762 , ..., 0.1437762 , 0.1437762 ,
       0.1437762 ], dtype=float32)
```

```
pd.DataFrame({'Actual value':ytest.flatten(),
```

```
            'Predicted  value':np.round(ypred.flatten()).astype('int')}).head(20)
```

	Actual value	Predicted value
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	1	0
10	0	0
11	0	0
12	0	0
13	1	0
14	0	0
15	0	0
16	1	0
17	0	0
18	0	0
19	0	0