

## Assignment -4

### SMS SPAM Classification

Assignment Date	08 NOVEMBER 2022
Team id	PNT2022TMID46647
Maximum Marks	2 Marks

#### Question-1:

Download the dataset

#### Question-2:

Import required library

**Solution**

```
import nltk
import pandas as pd
import re
from nltk.corpus import stopwords from nltk.stem.porter
import PorterStemmer from sklearn.feature_extraction.text
import CountVectorizer from sklearn.model_selection import
train_test_split from tensorflow.keras.models import
Sequential from tensorflow.keras.layers import Dense
```



```
Import necessary libraries

[ ] import nltk
import pandas as pd
import re

[ ] from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer

[ ] from sklearn.feature_extraction.text import CountVectorizer

[ ] from sklearn.model_selection import train_test_split

[ ] from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

#### Question-3:

Read dataset and do pre-processing

**Solution**

```
data=pd.read_csv('/content/drive/MyDrive/assignment 4/spam.csv',encoding='latin')
nltk.download('stopwords') ps=PorterStemmer() input=[] for i in range(0,5572):
review=data['v2'][i] review=re.sub('[^a-zA-Z]', ' ',review)
review=review.lower() review=review.split() review=[ps.stem(word) for word in review
if not word in set(stopwords.words('english'))] review=' '.join(review) input.append(review)
cv=CountVectorizer(max_features=7000) x=cv.fit_transform(input).toarray()
y=data['v1'].values x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.2)
```

```
Read dataset

[ ] data=pd.read_csv('/content/drive/MyDrive/assignment 4/spam.csv',encoding='latin')

Preprocessing

[ ] nltk.download('stopwords')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
True

[ ] ps=PorterStemmer()
input=[]

[ ] for i in range(0,5572):
review=data['v2'][i]
review=re.sub('[^a-zA-Z]', ' ',review)
review=review.lower()
review=review.split()
review=[ps.stem(word) for word in review if not word in set(stopwords.words('english'))]
review=' '.join(review)
input.append(review)

0s completed at 8:54 PM

[ ] cv=CountVectorizer(max_features=7000)

[ ] x=cv.fit_transform(input).toarray()
x
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])

[ ] y=data['v1'].values
y
array(['ham', 'ham', 'spam', ..., 'ham', 'ham', 'ham'], dtype=object)

x.shape
(5572, 6221)

[ ] x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.2)
```

#### Question-4:

Create Model

**Solution** model=Sequential()

#### Question-5:

Add Layers (LSTM, Dense-(Hidden Layers), Output) **Solution**

model.add(Dense(units=6221,activation='relu'))

model.add(Dense(units=7000,activation='relu'))

model.add(Dense(units=1,activation='sigmoid'))

```
+ Code + Text
Create model
[ ] model=Sequential()
Add layers
Input layer
[ ] model.add(Dense(units=6221,activation='relu'))
Hidden layer
[ ] model.add(Dense(units=7000,activation='relu'))
Output layer
[ ] model.add(Dense(units=1,activation='sigmoid'))
```

### Question-6:

Compile The Model [Solution](#)

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

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

### Question-7:

Fit The Model [Solution](#)

```
model.fit(x_train,y_train,epochs=5)
```

```
model.fit(x_train,y_train,epochs=5)
```

### Question-7:

Save The Model

[Solution](#)

```
model.save("Flowers.h5")
```

```
Fit the model
[ ] model.save('spam.h5')
```