Fake vs True News Classification Project

By Mohammed Faisal Khan

Why should you care about whether or not your news is real or fake?

- The spread of fake news has become easier in the digital age, as social media platforms and other online channels allow anyone to create and share content with the world. The problem with fake news is that it can be very convincing, and people may believe it to be true without questioning its authenticity. This can lead to a range of negative consequences, That is why a sophisticated method is required to identify fake news

Problem Statement

- Given - A Data Set of Fake and Real news.
- Objective - To develop a solution which detects if a given news is Fake or Real.
- Methodology used - We try to pose the problem as a text classification problem and build a deep learning model for achieving the objective.

Goal

- The Goal of This notebook is to use Machine Learning, Deep Learning and NLP to Detect Fake News

Step-1:- Importing Libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer,WordNetLemmatizer
from nltk.tokenize import word_tokenize
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Embedding, LSTM, SimpleRNN
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing import sequence
import warnings
warnings.filterwarnings('ignore')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

Step-2 : Loading and Cleaning Data

```
# 2.1 Import Data
df = pd.read_csv('news.csv',error_bad_lines=False,engine='python')
df
```

	title	text	label
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	fake
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	fake
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	fake
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	fake
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	fake
...	...	...	...
44893	'Fully committed' NATO backs new U.S. approach...	BRUSSELS (Reuters) - NATO allies on Tuesday we...	true
44894	LexisNexis withdrew two products from Chinese ...	LONDON (Reuters) - LexisNexis, a provider of l...	true
44895	Minsk cultural hub becomes haven from authorities	MINSK (Reuters) - In the shadow of disused Sov...	true
44896	Vatican upbeat on possibility of Pope Francis ...	MOSCOW (Reuters) - Vatican Secretary of State ...	true

```
# 2.2 Inspect the dataframe
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 44898 entries, 0 to 44897
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   title   44898 non-null  object
 1   text    44898 non-null  object
 2   label   44898 non-null  object
dtypes: object(3)
memory usage: 1.0+ MB
```

- After Inspecting we can see there are 44898 rows and 3 columns

```
# The df.isna()/isnull() code gives the counts of missing values
df.isna().sum()
```

```
title    0
text     0
label    0
dtype: int64
```

- We can see there is no null values in the dataset

```
df['label'].value_counts(normalize=True)
```

```
fake    0.522985
true    0.477015
Name: label, dtype: float64
```

The Percentage of True and Fake News articles:

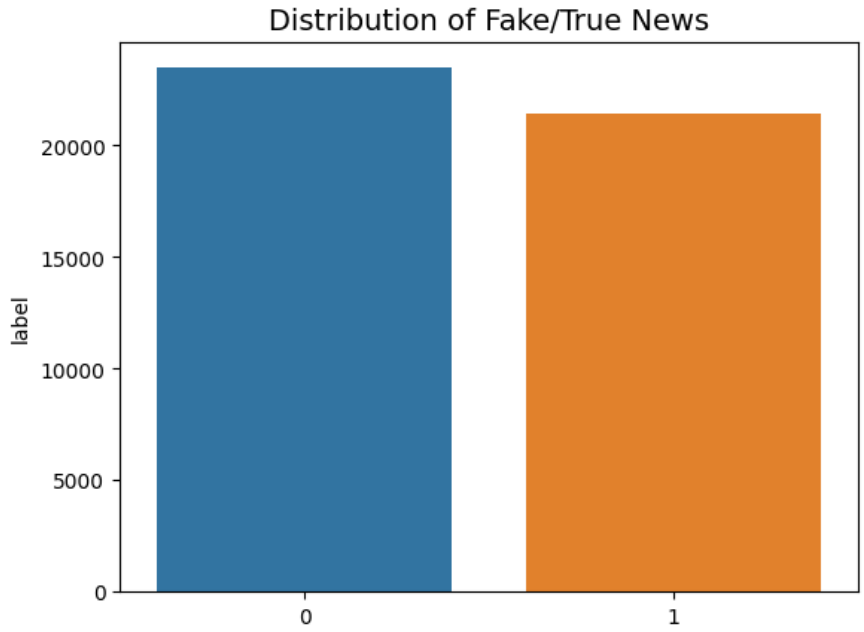
- True:- 48%
- Fake:- 52%

```
df['label'].replace({'true':1,'fake':0},inplace=True)
df.head()
```

	title	text	label
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	0
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	0

```
class_names = ['fake','true']
l_count = df['label'].value_counts()
sns.barplot(x=l_count.index, y=l_count)
plt.title('Distribution of Fake/True News',fontsize =14)
```

```
Text(0.5, 1.0, 'Distribution of Fake/True News')
```



- Data Visualization of all News Titles

```
from wordcloud import WordCloud
titles = ' '.join(title for title in df['title'])
wordcloud = WordCloud(
    background_color='white',
    max_words=300,
    width=800,
    height=400,
).generate(titles)

plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Many of available news articles are about elections and presidents of USA

```
def clean_text(msg):
  token = word_tokenize(msg.lower())
  ftoken = [i for i in token if i.isalpha()]

  stop = stopwords.words('english')
  stoken = [i for i in ftoken if i not in stop]

  lemma = WordNetLemmatizer()
  ltoken = [lemma.lemmatize(i) for i in stoken]

  return ' '.join(ltoken)
```

```
#Converting data type of 'title' column into string type
df['title']=df['title'].astype(str)
```

```
df['clean_msg'] = df['title'].apply(clean_text)
```

```
df.head()
```

	title	text	label	
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	0	donald trump sends embarrass
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	0	drunk bragging trump staffer st
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	0	sheriff david clarke becomes i
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	0	trump obsessed even oba
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	0	pope francis called donalc

Step 3:-Data Preprocessing

```
#spliting data into x and y
x = df['clean_msg']
y = df['label']
```

```
y
```

```
0        0
1        0
2        0
3        0
4        0
        ..
44893    1
44894    1
44895    1
44896    1
```

```
44897    1
Name: label, Length: 44898, dtype: int64
```

```
# Spliting the dataset into 70% and 30% for train and test respectively
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.30,random_state=1)
```

```
from sklearn.feature_extraction.text import CountVectorizer
cvec = CountVectorizer(min_df=0.02)
xtrain = cvec.fit_transform(xtrain).toarray()
xtest = cvec.transform(xtest).toarray()
```

```
x
```

```
0        donald trump sends embarrassing new year eve m...
1        drunk bragging trump staffer started russian c...
2        sheriff david clarke becomes internet joke thr...
3        trump obsessed even obama name coded website i...
4        pope francis called donald trump christmas speech
                              ...
44893         committed nato back new approach afghanistan
44894       lexisnexis withdrew two product chinese market
44895                  minsk cultural hub becomes authority
44896    vatican upbeat possibility pope francis visiti...
44897              indonesia buy billion worth russian jet
Name: clean_msg, Length: 44898, dtype: object
```

```
df.head()
```

	title	text	label	clean_msg
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	0	donald trump sends embarrassing new year eve m...
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	0	drunk bragging trump staffer started russian c...
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	0	sheriff david clarke becomes internet joke thr...
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	0	trump obsessed even obama name coded website i...
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	0	pope francis called donald trump christmas speech

```
empty = []
for indx,tl,txt,lbl,cm in df.itertuples():
  if type(cm)==str:
    if cm.isspace():
      empty.append(indx)
print(empty)
```

```
[]
```

```
xtrain
```

```
array([[0, 0, 0, ..., 0, 1, 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, 1, 0, ..., 0, 0, 0]])
```

```
xtest
```

```
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, 1, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

Model 1:- ANN

```
# Building the 'Artificial Neural Network(ANN)'
ann = Sequential()
ann.add(Dense(units=32,activation='relu'))
ann.add(Dropout(rate=0.20))
ann.add(Dense(units=24,activation='relu'))
ann.add(Dropout(rate=0.20))
ann.add(Dense(units=12,activation='relu'))
ann.add(Dropout(rate=0.20))
ann.add(Dense(units=1,activation='sigmoid'))
ann.compile(optimizer='adam',loss='binary_crossentropy')
ann.fit(xtrain,ytrain,batch_size=50,epochs=50,validation_split=0.20)
```

```
503/503 [==============================] - 1s 2ms/step - loss: 0.4369 - val_loss: 0.4311
Epoch 23/50
503/503 [==============================] - 1s 2ms/step - loss: 0.4366 - val_loss: 0.4323
Epoch 24/50
```

```
Epoch 46/50
503/503 [==============================] - 1s 2ms/step - loss: 0.4334 - val_loss: 0.4315
Epoch 47/50
503/503 [==============================] - 1s 2ms/step - loss: 0.4330 - val_loss: 0.4323
Epoch 48/50
503/503 [==============================] - 1s 3ms/step - loss: 0.4344 - val_loss: 0.4320
Epoch 49/50
503/503 [==============================] - 2s 3ms/step - loss: 0.4311 - val_loss: 0.4318
Epoch 50/50
503/503 [==============================] - 2s 4ms/step - loss: 0.4315 - val_loss: 0.4330
<keras.callbacks.History at 0x7f2337f19f60>
```

Evaluation

Let's evaluate the performance of the ANN on the test set and generate a classification report.

```
ypred = ann.predict(xtest)
ypred = ypred>0.5
```

```
421/421 [==============================] - 1s 2ms/step
```

```
from sklearn.metrics import classification_report
print(classification_report(ytest,ypred))
```

```
                  precision    recall  f1-score   support

               0       0.93      0.62      0.74      7053
               1       0.69      0.95      0.80      6417

        accuracy                           0.77     13470
       macro avg       0.81      0.78      0.77     13470
    weighted avg       0.82      0.77      0.77     13470
```

Model 2:- LogisticRegression

```
# Building The 'LogisticRegression'
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(xtrain,ytrain)
ypred = logreg.predict(xtest)
```

Evaluation

Let's evaluate the performance of the LogisticRegression on the test set and generate a classification report.

```
from sklearn.metrics import classification_report
print(classification_report(ytest,ypred))
```

```
                  precision    recall  f1-score   support

               0       0.93      0.61      0.74      7053
               1       0.69      0.95      0.80      6417

        accuracy                           0.77     13470
       macro avg       0.81      0.78      0.77     13470
    weighted avg       0.82      0.77      0.77     13470
```

Double-click (or enter) to edit

```
df.head()
```

	title	text	label	clean_msg
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	0	donald trump sends embarrassing new year eve m...
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	0	drunk bragging trump staffer started russian c...
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	0	sheriff david clarke becomes internet joke thr...
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	0	trump obsessed even obama name coded website i...
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	0	pope francis called donald trump christmas speech

```
#spliting data into x and y
x = df['clean_msg']
y = df['label']
```

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.30,random_state=1)
```

```
sentlen = []
for i in df['clean_msg']:
  sentlen.append(len(word_tokenize(i)))

df['Sentlen'] = sentlen
```

```
df.head()
```

	title	text	label	clean_msg	Sentlen
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	0	donald trump sends embarrassing new year eve m...	9
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	0	drunk bragging trump staffer started russian c...	8
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	0	sheriff david clarke becomes internet joke thr...	10
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	0	trump obsessed even obama name coded website i...	8
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	0	pope francis called donald trump christmas speech	7

```
max(sentlen)
```

```
26
```

```
min(sentlen)
```

```
1
```

```
np.quantile(sentlen,0.90)
```

```
12.0
```

```
max_len = np.quantile(sentlen,0.90)
```

```
tok = Tokenizer(char_level=False,split=' ')
tok.fit_on_texts(xtrain)
tok.index_word
```

```
 944: 'elected',
 945: 'scalia',
 946: 'fighting',
 947: 'raid',
 948: 'update',
 949: 'boost',
 950: 'access',
 951: 'highlight',
 952: 'soon',
 953: 'ask',
 954: 'ca',
 955: 'la',
 956: 'hypocrite',
 957: 'perfectly',
 958: 'bannon',
 959: 'disaster',
 960: 'bring',
 961: 'usa',
 962: 'anthem',
 963: 'camp',
 964: 'supremacist',
 965: 'joke',
 966: 'iranian',
 967: 'vet',
 968: 'dirty',
 969: 'dispute',
 970: 'finance',
 971: 'riot',
 972: 'rnc',
 973: 'troll',
 974: 'minute',
 975: 'kurd',
 976: 'anyone',
 977: 'telling',
 978: 'rep',
 979: 'nazi',
 980: 'amendment',
 981: 'voted',
 982: 'african',
 983: 'sanctuary',
 984: 'tough',
 985: 'allow',
 986: 'feel',
 987: 'concerned',
 988: 'sea',
 989: 'fall',
 990: 'avoid',
 991: 'graft',
 992: 'low',
 993: 'side',
 994: 'prove',
 995: 'address',
 996: 'god',
 997: 'parenthood',
 998: 'guilty',
 999: 'testify',
 1000: 'committed',
 ...}
```

```
vocab_len = len(tok.index_word)
vocab_len
```

```
15485
```

```
seqtrain = tok.texts_to_sequences(xtrain)
seqmattrain = sequence.pad_sequences(seqtrain,maxlen=int(max_len))
seqmattrain
```

```
array([[   0,    0,    0, ...,  320, 4905,  227],
       [   0,    0,    0, ..., 6005,   17,  291],
       [   0,    0,    0, ..., 1297, 6006,  533],
       ...,
       [   0,    0,    0, ...,   54,  738,  959],
       [   0,   87, 1007, ...,  373,   12, 2069],
       [   0,    0,    0, ...,  354,  503,   14]], dtype=int32)
```

```
seqtest = tok.texts_to_sequences(xtest)
seqmattest = sequence.pad_sequences(seqtest,maxlen=int(max_len))
seqmattest
```

```
array([[   0,    0,    0, ...,  691,  482,   13],
       [   0,    0,    0, ...,   47,    5,   10],
       [   0,    0,    0, ..., 1379, 7440,    2],
       ...,
       [   0, 8050,   65, ..., 1898,   20, 2063],
       [  37,  978, 6557, ...,  348, 6720,  159],
       [   0,    0,    0, ...,  672,    3,  323]], dtype=int32)
```

Model 3:- SimpleRNN

```
# Building The 'SimpleRNN'
rnn = Sequential()
rnn.add(Embedding(vocab_len+1,300,input_length=int(max_len),mask_zero=True))
rnn.add(SimpleRNN(units=30,activation='tanh'))
rnn.add(Dense(units=30,activation='relu'))
rnn.add(Dropout(rate=0.30))
rnn.add(Dense(units=1,activation='sigmoid'))
rnn.compile(optimizer='adam',loss='binary_crossentropy')
rnn.fit(seqmattrain,ytrain,batch_size=50,epochs=25)
```

```
Epoch 1/25
629/629 [==============================] - 60s 91ms/step - loss: 0.1842
Epoch 2/25
629/629 [==============================] - 64s 102ms/step - loss: 0.0495
Epoch 3/25
629/629 [==============================] - 50s 80ms/step - loss: 0.0165
Epoch 4/25
629/629 [==============================] - 45s 72ms/step - loss: 0.0089
Epoch 5/25
629/629 [==============================] - 44s 71ms/step - loss: 0.0078
Epoch 6/25
629/629 [==============================] - 45s 72ms/step - loss: 0.0080
Epoch 7/25
629/629 [==============================] - 46s 73ms/step - loss: 0.0050
Epoch 8/25
629/629 [==============================] - 44s 70ms/step - loss: 0.0045
Epoch 9/25
629/629 [==============================] - 42s 67ms/step - loss: 0.0036
Epoch 10/25
629/629 [==============================] - 43s 69ms/step - loss: 0.0028
Epoch 11/25
629/629 [==============================] - 43s 69ms/step - loss: 0.0027
Epoch 12/25
629/629 [==============================] - 43s 68ms/step - loss: 0.0045
Epoch 13/25
629/629 [==============================] - 45s 72ms/step - loss: 0.0021
Epoch 14/25
629/629 [==============================] - 46s 73ms/step - loss: 0.0030
Epoch 15/25
629/629 [==============================] - 46s 73ms/step - loss: 0.0032
Epoch 16/25
```

```
629/629 [==============================] - 45s 72ms/step - loss: 0.0022
Epoch 17/25
629/629 [==============================] - 46s 73ms/step - loss: 0.0012
Epoch 18/25
629/629 [==============================] - 45s 72ms/step - loss: 0.0023
Epoch 19/25
629/629 [==============================] - 45s 72ms/step - loss: 0.0014
Epoch 20/25
629/629 [==============================] - 46s 73ms/step - loss: 1.6736e-04
Epoch 21/25
629/629 [==============================] - 45s 72ms/step - loss: 3.7528e-04
Epoch 22/25
629/629 [==============================] - 46s 73ms/step - loss: 0.0029
Epoch 23/25
629/629 [==============================] - 45s 71ms/step - loss: 0.0023
Epoch 24/25
629/629 [==============================] - 46s 72ms/step - loss: 0.0012
Epoch 25/25
629/629 [==============================] - 45s 72ms/step - loss: 7.2972e-04
<keras.callbacks.History at 0x7f23360103a0>
```

Evaluation

Let's evaluate the performance of the SimpleRNN on the test set and generate a classification report.

```
ypred = rnn.predict(seqmattest)
ypred = ypred>0.5
```

```
421/421 [==============================] - 2s 3ms/step
```

```
from sklearn.metrics import classification_report
print(classification_report(ytest,ypred))
```

```
                  precision    recall  f1-score   support

               0       0.95      0.95      0.95      7053
               1       0.94      0.95      0.95      6417

        accuracy                           0.95     13470
       macro avg       0.95      0.95      0.95     13470
    weighted avg       0.95      0.95      0.95     13470
```

Model 4:- LSTM

```
# Building The 'LSTM'
rnn = Sequential()
rnn.add(Embedding(vocab_len+1,300,input_length=int(max_len),mask_zero=True))
rnn.add(LSTM(units=30,activation='tanh'))
rnn.add(Dense(units=30,activation='relu'))
rnn.add(Dropout(rate=0.30))
rnn.add(Dense(units=1,activation='sigmoid'))
rnn.compile(optimizer='adam',loss='binary_crossentropy')
rnn.fit(seqmattrain,ytrain,batch_size=50,epochs=25)
```

```
Epoch 1/25
629/629 [==============================] - 66s 97ms/step - loss: 0.1878
Epoch 2/25
629/629 [==============================] - 60s 96ms/step - loss: 0.0630
Epoch 3/25
629/629 [==============================] - 60s 96ms/step - loss: 0.0302
Epoch 4/25
629/629 [==============================] - 64s 102ms/step - loss: 0.0175
Epoch 5/25
629/629 [==============================] - 64s 101ms/step - loss: 0.0103
```

```
Epoch 6/25
629/629 [==============================] - 60s 95ms/step - loss: 0.0106
Epoch 7/25
629/629 [==============================] - 59s 93ms/step - loss: 0.0053
Epoch 8/25
629/629 [==============================] - 59s 93ms/step - loss: 0.0040
Epoch 9/25
629/629 [==============================] - 60s 95ms/step - loss: 0.0049
Epoch 10/25
629/629 [==============================] - 60s 95ms/step - loss: 0.0029
Epoch 11/25
629/629 [==============================] - 60s 95ms/step - loss: 0.0011
Epoch 12/25
629/629 [==============================] - 60s 96ms/step - loss: 0.0016
Epoch 13/25
629/629 [==============================] - 61s 97ms/step - loss: 0.0017
Epoch 14/25
629/629 [==============================] - 60s 96ms/step - loss: 0.0035
Epoch 15/25
629/629 [==============================] - 61s 97ms/step - loss: 0.0032
Epoch 16/25
629/629 [==============================] - 63s 99ms/step - loss: 0.0011
Epoch 17/25
629/629 [==============================] - 63s 100ms/step - loss: 2.8537e-04
Epoch 18/25
629/629 [==============================] - 62s 99ms/step - loss: 3.8974e-05
Epoch 19/25
629/629 [==============================] - 61s 98ms/step - loss: 2.2086e-05
Epoch 20/25
629/629 [==============================] - 61s 98ms/step - loss: 1.7384e-05
Epoch 21/25
629/629 [==============================] - 61s 97ms/step - loss: 7.2754e-06
Epoch 22/25
629/629 [==============================] - 61s 96ms/step - loss: 5.5152e-06
Epoch 23/25
629/629 [==============================] - 60s 95ms/step - loss: 5.2954e-06
Epoch 24/25
629/629 [==============================] - 60s 96ms/step - loss: 3.4516e-06
Epoch 25/25
629/629 [==============================] - 60s 96ms/step - loss: 3.2252e-06
<keras.callbacks.History at 0x7f23342cc520>
```

Evaluation

Let's evaluate the performance of the LSTM on the test set and generate a classification report.

```
ypred = rnn.predict(seqmattest)
ypred = ypred>0.5
```

```
421/421 [==============================] - 3s 5ms/step
```

```
from sklearn.metrics import classification_report
print(classification_report(ytest,ypred))
```

```
                  precision    recall  f1-score   support

               0       0.95      0.96      0.95      7053
               1       0.95      0.95      0.95      6417

        accuracy                           0.95     13470
       macro avg       0.95      0.95      0.95     13470
    weighted avg       0.95      0.95      0.95     13470
```

```
df.head()
```

	title	text	label	clean_msg	Sentlen
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	0	donald trump sends embarrassing new year eve m...	9
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	0	drunk bragging trump staffer started russian c...	8
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	0	sheriff david clarke becomes internet joke thr...	10
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	0	trump obsessed even obama name coded website i...	8
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	0	pope francis called donald trump christmas speech	7

```
x = df['clean_msg']
y = df['label']
```

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.30,random_state=1)
```

```
from sklearn.feature_extraction.text import CountVectorizer
cvec = CountVectorizer(min_df=0.02)
xtrain = cvec.fit_transform(xtrain).toarray()
xtest = cvec.transform(xtest).toarray()
```

Model 5:- Naive Bayes

```
# Building The 'Navie Bayes'
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB

def mymodel(model):
  model.fit(xtrain,ytrain)
  ypred = model.predict(xtest)

  print(classification_report(ytest,ypred))
```

Evaluation

```
gnb = GaussianNB()

mymodel(gnb)
```

```
                  precision    recall  f1-score   support

               0       0.91      0.60      0.73      7053
               1       0.68      0.94      0.79      6417

        accuracy                           0.76     13470
       macro avg       0.80      0.77      0.76     13470
    weighted avg       0.80      0.76      0.76     13470
```

```
mnb = MultinomialNB()
mymodel(mnb)
```

```
                  precision    recall  f1-score   support

               0       0.64      0.88      0.74      7053
               1       0.78      0.46      0.58      6417
```

```
              accuracy                           0.68     13470
             macro avg       0.71      0.67      0.66     13470
          weighted avg       0.71      0.68      0.67     13470
```

```
bnb = BernoulliNB()
mymodel(bnb)
```

```
                   precision    recall  f1-score   support

                0       0.91      0.60      0.73      7053
                1       0.68      0.94      0.79      6417

              accuracy                           0.76     13470
             macro avg       0.80      0.77      0.76     13470
          weighted avg       0.80      0.76      0.76     13470
```

Conclusion

- Accuracy achieved using ANN Model : 77%
- Accuracy achieved using LogisticRegression Model : 77%
- Accuracy achieved using SimpleRNN Model : 95%
- Accuracy achieved using LSTM Model : 95%
- Accuracy achieved using GaussianNB Model : 76%
- Accuracy achieved using MultinomialNB Model : 68%
- Accuracy achieved using BernoullibNB Model : 76%

✓ 0s completed at 10:52 PM