Assignment-4

Emerging Methods For Early Detection Of Forest Fires

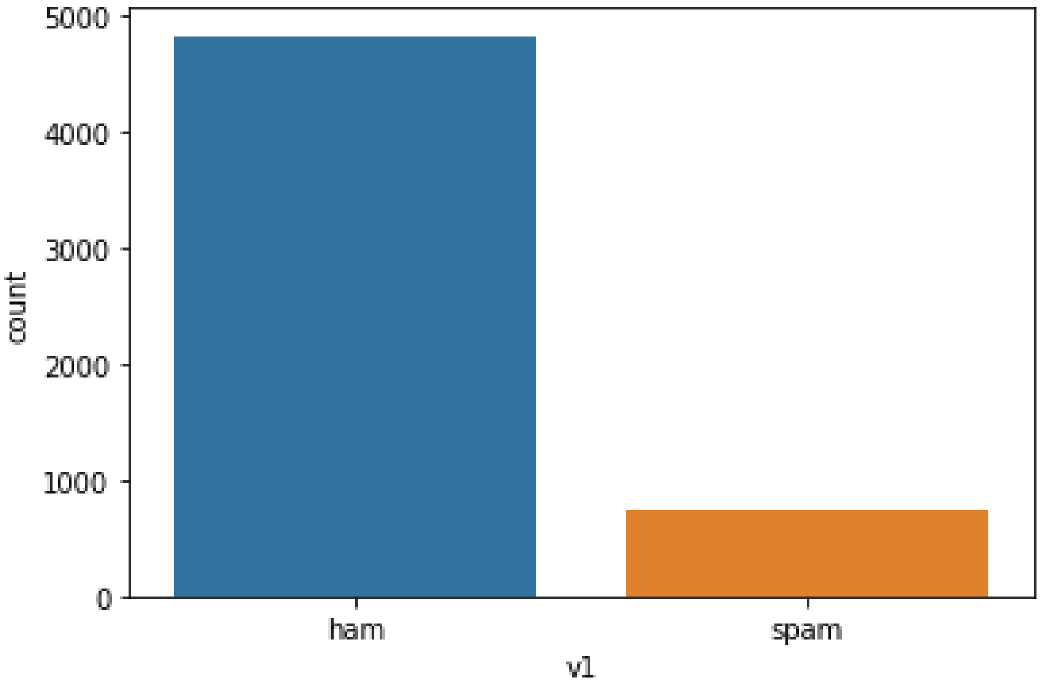
DATE	27 october 2022
TEAM MEMBERS	*Sushilnandas(Leader) – Santhiya – Amala Ajin – Sherlinfrancis – Karthick*
TEAM ID	PNT2022TMID52514
MAXIMUM MARKS	2 Marks

```
importpandasaspd import numpy as np
importmatplotlib.pyplotasplt import seaborn as
sns
fromsklearn.model_selection
importtrain_test_splitfromsklearn.preprocessing
import LabelEncoderfrom tensorflow.keras.models
import Model
fromtensorflow.keras.layers
importLSTM,Activation,Dense,Dropout,Input,Embedd
ing from tensorflow.keras.optimizers
importRMSpropfromtensorflow.keras.preprocessing.
text import Tokenizerfrom
tensorflow.keras.preprocessing import
sequencefrom tensorflow.keras.utils
importto_categoricalfromtensorflow.keras.callbac
ks import EarlyStopping
%matplotlibinlineimport csv
withopen('/spam.csv','r')ascsvfile:reader
= csv.reader(csvfile)df=
pd.read_csv(r'/spam.csv',encoding='latin-1')
df.head()
v1  v2Unnamed:2\0
hamGountiljurongpoint,crazy..Availableonly...
    NaN
ham Oklar...Jokingwifuoni...  NaN
```

```
spamFreeentryin2awklycomptowinFACupfina...  NaN
hamU dun say so early hor... U c already then
say...  NaN 4 hamNah I don't think he goes to
usf, he lives aro...  NaN


Unnamed:3Unnamed:4
NaN NaN
NaN NaN
NaN NaN
NaNNaN4NaNNaNdf.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):
#ColumnNon-NullCountDtype
 -----  -----------  ------------------------  ---------

0   v1  5572non-null   object
1   v2  5572 non-null  object dtypes:
object(2)memory usage:
87.2+KBsns.countplot(df.v1)
/usr/local/lib/python3.7/dist-
packages/seaborn/_decorators.py:43:
FutureWarning:Passthefollowingvariableasakeyword
arg:x.Fromversion 0.12, the only valid
positional argument will be `data`, and passing
other
argumentswithoutanexplicitkeywordwillresultinane
rroror misinterpretation.FutureWarning
<matplotlib.axes._subplots.AxesSubplotat0x7f5197
dac250>
```



```
X=df.v2Y= df.v1
le=LabelEncoder()Y= le.fit_transform(Y)
Y=Y.reshape(-1,1)
X_train,X_test,Y_train,Y_test=train_test_split(X
,Y,test_size=0.20) max_words= 1000max_len
= 150
tok=Tokenizer(num_words=max_words)tok.fit_on_tex
ts(X_train)
sequences=tok.texts_to_sequences(X_train)sequenc
es_matrix=
sequence.pad_sequences(sequences,maxlen=max_len)
defRNN():
inputs=Input(name='inputs',shape=[max_len])
layer=Embedding(max_words,50,input_length=max_le
n)(inputs) layer = LSTM(128)(layer)  layer=
Dense(256,name='FC1')(layer)         layer =
Activation('relu')(layer)     layer =
Dropout(0.5)(layer)
layer = Dense(1,name='out_layer')(layer)layer=
Activation('tanh')(layer) model =
Model(inputs=inputs,outputs=layer)   return
model
model = RNN()model.summary()
model.compile(loss='binary_crossentropy',optimiz
er=RMSprop(),metrics=['accura cy','mse','mae'])
Model:"model"
```

```
Layer(type)    OutputShapeParam#
================================================
=================
inputs(InputLayer)[(None,150)]   0
embedding(Embedding)   (None,150,50)  50000
lstm(LSTM) (None,128) 91648
FC1(Dense) (None,256) 33024
activation(Activation)    (None,256) 0
dropout(Dropout)   (None,256) 0
out_layer(Dense)   (None,1)   257
activation_1(Activation)  (None,1)   0


================================================
=================
Totalparams:174,929
Trainableparams:174,929
Non-trainableparams:0
```

```
model.fit(sequences_matrix,Y_train,batch_size=12
8,epochs=10,


validation_split=0.2,callbacks=[EarlyStopping(mo
nitor='val_loss',min_delta=0. 0001)])
Epoch1/10
28/28[==============================]-
17s486ms/step-loss:0.2960-
accuracy: 0.8819 - mse: 0.0821 - mae: 0.1563 -
val_loss: 0.1341 - val_accuracy: 0.9675 -
```

```
val_mse: 0.0344 - val_mae: 0.1237Epoch 2/10
28/28[==============================] -
13s462ms/step-loss:0.1149-
accuracy:0.9764-mse:0.0381-mae:0.1538-
val_loss:0.1321-
val_accuracy:0.9798-val_mse:0.0437-
val_mae:0.1695
```

```
<keras.callbacks.Historyat0x7f5193192590>
test_sequences=tok.texts_to_sequences(X_test)tes
t_sequences_matrix=
sequence.pad_sequences(test_sequences,maxlen=max
_len)accr=
model.evaluate(test_sequences_matrix,Y_test)
35/35[==============================]-
3s78ms/step-loss:0.1590-
accuracy:0.9812-mse:0.0451-mae:0.1733
```

```
print('Testset\nLoss:{:0.3f}\nAccuracy:
{:0.3f}'.format(accr[0],accr[1])) Test set
```

```
Loss: 0.159Accuracy: 0.981
model.save("./assign4model.h5")
fromtensorflow.keras.modelsimportload_modelm2
= load_model("./assign4model.h5")
m2.evaluate(test_sequences_matrix,Y_test)
35/35[==============================]-
3s68ms/step-loss:0.1590-
accuracy:0.9812-mse:0.0451-mae:0.1733

[0.1589982509613037,
0.9811659455299377,
0.04506031796336174,
0.1733826422691345]
```