

```
In [1]: #!unzip dank_data-master.zip
#!pip install tensorflow_addons
#!wget http://nlp.stanford.edu/data/glove.6B.zip
#!unzip glove*.zip
```

```
In [2]: import glob
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Dense, Input, Conv2D, MaxPool2D, Activation, Dropout, Flatten, Embedding, LSTM, concatenate
from tensorflow.keras.models import Model
import tensorflow as tf
import numpy as np
import tensorflow_addons as tfa
import logging
from tensorflow.keras.preprocessing.text import Tokenizer
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.applications.vgg19 import preprocess_input
from tensorflow.keras.callbacks import LearningRateScheduler
from tensorflow.keras.callbacks import ReduceLROnPlateau
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.metrics import confusion_matrix, accuracy_score, f1_score
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [3]: training='/content/dank_data-master/data/training/*'
test='/content/dank_data-master/data/test/*'
validation='/content/dank_data-master/data/validation/*'
```

```
In [4]: training = glob.glob(training)
test = glob.glob(test)
validation = glob.glob(validation)
```

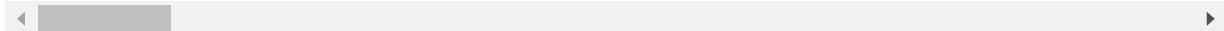
```
In [5]: final_dank=pd.read_csv('/content/dank_data-master/data/final_dank.csv')
train_labels = [fn.split('/')[-1].split('.')[0].strip() for fn in training]
validation_labels = [fn.split('/')[-1].split('.')[0].strip() for fn in validation]
test_labels = [fn.split('/')[-1].split('.')[0].strip() for fn in test]
```

```
In [6]: for labels in train_labels:
    if labels==train_labels[0]:
        train_data =final_dank[final_dank['id']==labels]
    else :
        train_data =train_data.append(final_dank[final_dank['id']==labels],sort=False)
for labels in validation_labels:
    if labels==validation_labels[0]:
        val_data =final_dank[final_dank['id']==labels]
    else :
        val_data =val_data.append(final_dank[final_dank['id']==labels],sort=False)
for labels in test_labels:
    if labels==test_labels[0]:
        test_data =final_dank[final_dank['id']==labels]
    else :
        test_data =test_data.append(final_dank[final_dank['id']==labels],sort=False)
print(train_data.shape)
print(test_data.shape)
print(val_data.shape)
train_data.head(5)
```

(3405, 68)  
(1719, 68)  
(1688, 68)

Out[6]:

|       | Unnamed:<br>0 | level_0 | index   | author               | awards | processed_words                                     | created_utc  |
|-------|---------------|---------|---------|----------------------|--------|---|--------------|
| 53606 | 96606         | 1118.0  | 32771.0 | SwiftScout4          | []     | ['dowk']  | 1.584914e+09 |
| 35469 | 61068         | 63931.0 | 63931.0 | Captain_Trisl        | []     | ['boy', 'hang', 'quarantine', 'orona', 'extra']     | 1.584168e+09 |
| 11453 | 25253         | 26477.0 | 26477.0 | Kenmoops             | []     | ['vehe', 'believ', 'lie', 'girl']                   | 1.584383e+09 |
| 53276 | 96125         | 621.0   | 32274.0 | fantastich_freidrich | []     | ['human', 'come', 'futuretim', 'travel', 'trap...'] | 1.584917e+09 |
| 65568 | 129029        | 4832.0  | 66485.0 | YashSSJB1            | []     | ['centr', 'attractionm', 'show', 'fulli', 'bui...'] | 1.584688e+09 |



```
In [7]: def file_extension(x):
    return x+".jpg"
train_data['id'] = train_data['id'].apply(file_extension)
val_data['id'] = val_data['id'].apply(file_extension)
test_data['id'] = test_data['id'].apply(file_extension)
```

```
In [8]: def numeric_to_string(x):
    if (x==1.0):
        return 'Not_dank'
    elif (x==0.0):
        return 'Dank'
train_data['dank_level_new'] = train_data['dank_level'].apply(numeric_to_string)
val_data['dank_level_new'] = val_data['dank_level'].apply(numeric_to_string)
test_data['dank_level_new'] = test_data['dank_level'].apply(numeric_to_string)
```

```
In [9]: logger = logging.getLogger()
logger.disabled = False
train_datagen = ImageDataGenerator(zoom_range=0.3, rotation_range=50,
                                   width_shift_range=0.2, height_shift_range=
0.2, shear_range=0.2,
                                   horizontal_flip=True, fill_mode='nearest')
train_generator = train_datagen.flow_from_dataframe(
    dataframe=train_data,
    directory="/content/dank_data-master/data/training/",
    x_col="id",
    y_col="dank_level_new",
    subset="training",
    batch_size=30,
    seed=42,
    class_mode="binary",
    target_size= (156,156))
predict_datagen = ImageDataGenerator(preprocessing_function=preprocess_input
)
train_prediction_generator = predict_datagen.flow_from_dataframe(
    dataframe=train_data,
    directory="/content/dank_data-master/data/training/",
    x_col="id",
    y_col="dank_level_new",
    batch_size=30,
    seed=42,
    class_mode="binary",
    shuffle=False,
    target_size= (156,156))
validation_prediction_generator = predict_datagen.flow_from_dataframe(
    dataframe=val_data,
    directory="/content/dank_data-master/data/validation/",
    x_col="id",
    y_col="dank_level_new",
    batch_size=30,
    seed=42,
    shuffle=False,
    class_mode="binary",
    target_size= (156,156))
test_prediction_generator = predict_datagen.flow_from_dataframe(
    dataframe=test_data,
    directory="/content/dank_data-master/data/test/",
    x_col="id",
    y_col="dank_level_new",
    batch_size=30,
    seed=42,
    shuffle=False,
    class_mode="binary",
    target_size= (156,156))
```

Found 3405 validated image filenames belonging to 2 classes.  
 Found 3405 validated image filenames belonging to 2 classes.  
 Found 1688 validated image filenames belonging to 2 classes.  
 Found 1719 validated image filenames belonging to 2 classes.

```
In [10]: input = Input(shape=(156,156, 3,))

#Conv Layer
Conv1 = Conv2D(filters=256,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv1')(input)
#MaxPool Layer
Pool1 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool1')(Conv1)

#Conv Layer
Conv2 = Conv2D(filters=128,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv2')(Pool1)
#MaxPool Layer
Pool2 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool2')(Conv2)

#Conv Layer
Conv3 = Conv2D(filters=64,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv3')(Pool2)
#MaxPool Layer
Pool3 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool3')(Conv3)

#Conv Layer
Conv4 = Conv2D(filters=32,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv4')(Pool3)
#MaxPool Layer
Pool4 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool4')(Conv4)

Conv5 = Conv2D(filters=16,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv5')(Pool4)
#MaxPool Layer
Pool5 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool5')(Conv5)

Conv6 = Conv2D(filters=8,kernel_size=(3,3),padding='same',data_format='channels_last',
               activation='relu',kernel_initializer=tf.keras.initializers.he_normal(seed=0),name='Conv6')(Pool5)
#MaxPool Layer
Pool6 = MaxPool2D(pool_size=(2,2),strides=(2,2),padding='same',data_format='channels_last',name='Pool6')(Conv6)

#Flatten
flatten = Flatten(data_format='channels_last',name='Flatten')(Pool6)
```

```
#FC Layer
FC1 = Dense(units=128,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=32),name='FC1')(flatten)

#FC Layer
FC2 = Dense(units=64,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=33),name='FC2')(FC1)

#output layer
Out = Dense(units=1,activation='sigmoid',kernel_initializer=tf.keras.initializers.glorot_normal(seed=3),name='Output')(FC2)

model = Model(inputs=input, outputs=Out)
model.summary()
```

Model: "model"

| Layer (type)              | Output Shape          | Param # |
|---------------------------|-----------------------|---------|
| <hr/>                     |                       |         |
| input_1 (InputLayer)      | [(None, 156, 156, 3)] | 0       |
| Conv1 (Conv2D)            | (None, 156, 156, 256) | 7168    |
| Pool1 (MaxPooling2D)      | (None, 78, 78, 256)   | 0       |
| Conv2 (Conv2D)            | (None, 78, 78, 128)   | 295040  |
| Pool2 (MaxPooling2D)      | (None, 39, 39, 128)   | 0       |
| Conv3 (Conv2D)            | (None, 39, 39, 64)    | 73792   |
| Pool3 (MaxPooling2D)      | (None, 20, 20, 64)    | 0       |
| Conv4 (Conv2D)            | (None, 20, 20, 32)    | 18464   |
| Pool4 (MaxPooling2D)      | (None, 10, 10, 32)    | 0       |
| Conv5 (Conv2D)            | (None, 10, 10, 16)    | 4624    |
| Pool5 (MaxPooling2D)      | (None, 5, 5, 16)      | 0       |
| Conv6 (Conv2D)            | (None, 5, 5, 8)       | 1160    |
| Pool6 (MaxPooling2D)      | (None, 3, 3, 8)       | 0       |
| Flatten (Flatten)         | (None, 72)            | 0       |
| FC1 (Dense)               | (None, 128)           | 9344    |
| FC2 (Dense)               | (None, 64)            | 8256    |
| Output (Dense)            | (None, 1)             | 65      |
| <hr/>                     |                       |         |
| Total params: 417,913     |                       |         |
| Trainable params: 417,913 |                       |         |
| Non-trainable params: 0   |                       |         |

---

```
In [11]: def scheduler(epoch,lr):
    if((epoch+1)%3==0):
        lr=lr*0.95
        return lr
    else:
        return lr
```

```
In [12]: filepath="model_save/weights-{epoch:02d}-{val_accuracy:.4f}.h5"
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_accuracy', mode='auto')

lrschedule = tf.keras.callbacks.LearningRateScheduler(scheduler,verbose=0.1)

#stop the training if your validation accuracy is not increased in last 2 epochs.
early_stop= EarlyStopping(monitor='val_accuracy', patience=3,verbose=1)

#If your validation accuracy at that epoch is Less than previous epoch accuracy, you have to decrease the
#Learning rate by 10%
reduce_lr = ReduceLROnPlateau(monitor='val_accuracy', factor=0.75,
                             patience=3, min_lr=0.001,verbose=1)

model.compile(
    loss='binary_crossentropy',
    optimizer=tf.keras.optimizers.RMSprop(lr=1e-5),
    metrics=[ 'accuracy',tf.keras.metrics.Precision(),tf.keras.metrics.Recall(),tf.keras.metrics.F1Score(num_classes=1)]
)
```

```
In [14]: history=model.fit_generator(train_generator,steps_per_epoch=len(train_generator),epochs=50,validation_data=test_prediction_generator,validation_steps=len(test_prediction_generator),use_multiprocessing=False,workers=12,callbacks=[lrschedule,checkpoint,reduce_lr])
```

Epoch 1/50

Epoch 00001: LearningRateScheduler reducing learning rate to 9.9999974737875  
2e-06.

114/114 [=====] - 105s 867ms/step - loss: 3.5656 - accuracy: 0.4899 - precision: 0.4816 - recall: 0.4532 - f1\_score: 0.6613 - val\_loss: 5.7788 - val\_accuracy: 0.4846 - val\_precision: 0.4741 - val\_recall: 0.6053 - val\_f1\_score: 0.6515

Epoch 2/50

Epoch 00002: LearningRateScheduler reducing learning rate to 9.9999974737875  
2e-06.

114/114 [=====] - 102s 814ms/step - loss: 2.2958 - accuracy: 0.5048 - precision: 0.4980 - recall: 0.5098 - f1\_score: 0.6600 - val\_loss: 4.9767 - val\_accuracy: 0.4753 - val\_precision: 0.4636 - val\_recall: 0.5439 - val\_f1\_score: 0.6512

Epoch 3/50

Epoch 00003: LearningRateScheduler reducing learning rate to 9.4999976000981  
3e-06.

114/114 [=====] - 101s 820ms/step - loss: 1.8914 - accuracy: 0.4975 - precision: 0.4905 - recall: 0.4902 - f1\_score: 0.6602 - val\_loss: 3.8520 - val\_accuracy: 0.4689 - val\_precision: 0.4545 - val\_recall: 0.4934 - val\_f1\_score: 0.6518

Epoch 4/50

Epoch 00004: LearningRateScheduler reducing learning rate to 9.4999957811087  
4e-06.

114/114 [=====] - 98s 810ms/step - loss: 1.6379 - accuracy: 0.4943 - precision: 0.4873 - recall: 0.4902 - f1\_score: 0.6606 - val\_loss: 3.2173 - val\_accuracy: 0.4764 - val\_precision: 0.4561 - val\_recall: 0.4308 - val\_f1\_score: 0.6518

Epoch 5/50

Epoch 00005: LearningRateScheduler reducing learning rate to 9.4999957811087  
4e-06.

114/114 [=====] - 96s 803ms/step - loss: 1.3966 - accuracy: 0.4913 - precision: 0.4846 - recall: 0.4979 - f1\_score: 0.6605 - val\_loss: 2.7347 - val\_accuracy: 0.4974 - val\_precision: 0.4798 - val\_recall: 0.4717 - val\_f1\_score: 0.6518

Epoch 6/50

Epoch 00006: LearningRateScheduler reducing learning rate to 9.02499959920533  
e-06.

114/114 [=====] - 98s 826ms/step - loss: 1.2747 - accuracy: 0.4978 - precision: 0.4908 - recall: 0.4926 - f1\_score: 0.6605 - val\_loss: 2.5493 - val\_accuracy: 0.4916 - val\_precision: 0.4736 - val\_recall: 0.4633 - val\_f1\_score: 0.6518

Epoch 7/50

Epoch 00007: LearningRateScheduler reducing learning rate to 9.02499959920533  
e-06.

114/114 [=====] - 99s 825ms/step - loss: 1.2140 - accuracy: 0.5063 - precision: 0.4994 - recall: 0.5152 - f1\_score: 0.6605 - val\_loss: 2.5724 - val\_accuracy: 0.4956 - val\_precision: 0.4795 - val\_recall: 0.5066 - val\_f1\_score: 0.6518

Epoch 8/50

Epoch 00008: LearningRateScheduler reducing learning rate to 9.02499959920533e-06.  
114/114 [=====] - 101s 835ms/step - loss: 1.1301 - accuracy: 0.5037 - precision: 0.4968 - recall: 0.5009 - f1\_score: 0.6605 - val\_loss: 2.3646 - val\_accuracy: 0.4764 - val\_precision: 0.4557 - val\_recall: 0.4272 - val\_f1\_score: 0.6518  
Epoch 9/50

Epoch 00009: LearningRateScheduler reducing learning rate to 8.573749619245064e-06.  
114/114 [=====] - 104s 873ms/step - loss: 1.0587 - accuracy: 0.5004 - precision: 0.4935 - recall: 0.4979 - f1\_score: 0.6605 - val\_loss: 2.4320 - val\_accuracy: 0.4811 - val\_precision: 0.4626 - val\_recall: 0.4537 - val\_f1\_score: 0.6518  
Epoch 10/50

Epoch 00010: LearningRateScheduler reducing learning rate to 8.573749255447183e-06.  
114/114 [=====] - 105s 881ms/step - loss: 0.9779 - accuracy: 0.5242 - precision: 0.5171 - recall: 0.5313 - f1\_score: 0.6605 - val\_loss: 1.8694 - val\_accuracy: 0.4852 - val\_precision: 0.4631 - val\_recall: 0.4079 - val\_f1\_score: 0.6518  
Epoch 11/50

Epoch 00011: LearningRateScheduler reducing learning rate to 8.573749255447183e-06.  
114/114 [=====] - 105s 873ms/step - loss: 0.9826 - accuracy: 0.5189 - precision: 0.5122 - recall: 0.5110 - f1\_score: 0.6605 - val\_loss: 1.6512 - val\_accuracy: 0.4939 - val\_precision: 0.4726 - val\_recall: 0.4043 - val\_f1\_score: 0.6518  
Epoch 12/50

Epoch 00012: LearningRateScheduler reducing learning rate to 8.145061792674824e-06.  
114/114 [=====] - 105s 871ms/step - loss: 0.9399 - accuracy: 0.5225 - precision: 0.5156 - recall: 0.5223 - f1\_score: 0.6605 - val\_loss: 1.5925 - val\_accuracy: 0.4916 - val\_precision: 0.4707 - val\_recall: 0.4152 - val\_f1\_score: 0.6518  
Epoch 13/50

Epoch 00013: LearningRateScheduler reducing learning rate to 8.145061656250618e-06.  
114/114 [=====] - 106s 875ms/step - loss: 0.9123 - accuracy: 0.5204 - precision: 0.5133 - recall: 0.5277 - f1\_score: 0.6605 - val\_loss: 1.5902 - val\_accuracy: 0.4875 - val\_precision: 0.4619 - val\_recall: 0.3646 - val\_f1\_score: 0.6518  
Epoch 14/50

Epoch 00014: LearningRateScheduler reducing learning rate to 8.145061656250618e-06.  
114/114 [=====] - 107s 880ms/step - loss: 0.8872 - accuracy: 0.5087 - precision: 0.5018 - recall: 0.5039 - f1\_score: 0.6605 - val\_loss: 1.4071 - val\_accuracy: 0.4817 - val\_precision: 0.4611 - val\_recall: 0.4284 - val\_f1\_score: 0.6518  
Epoch 15/50

Epoch 00015: LearningRateScheduler reducing learning rate to 7.73780857343808 7e-06.  
114/114 [=====] - 110s 872ms/step - loss: 0.8695 - accuracy: 0.5201 - precision: 0.5128 - recall: 0.5354 - f1\_score: 0.6605 - val\_loss: 1.4001 - val\_accuracy: 0.4910 - val\_precision: 0.4688 - val\_recall: 0.3983 - val\_f1\_score: 0.6518  
Epoch 16/50

Epoch 00016: LearningRateScheduler reducing learning rate to 7.73780811869073 7e-06.  
114/114 [=====] - 106s 880ms/step - loss: 0.8647 - accuracy: 0.5034 - precision: 0.4965 - recall: 0.5015 - f1\_score: 0.6605 - val\_loss: 1.2769 - val\_accuracy: 0.4875 - val\_precision: 0.4684 - val\_recall: 0.4465 - val\_f1\_score: 0.6518  
Epoch 17/50

Epoch 00017: LearningRateScheduler reducing learning rate to 7.73780811869073 7e-06.  
114/114 [=====] - 110s 875ms/step - loss: 0.8458 - accuracy: 0.5060 - precision: 0.4991 - recall: 0.4973 - f1\_score: 0.6605 - val\_loss: 1.2980 - val\_accuracy: 0.5015 - val\_precision: 0.4841 - val\_recall: 0.4753 - val\_f1\_score: 0.6518  
Epoch 18/50

Epoch 00018: LearningRateScheduler reducing learning rate to 7.3509177127562e -06.  
114/114 [=====] - 104s 858ms/step - loss: 0.8388 - accuracy: 0.5142 - precision: 0.5073 - recall: 0.5205 - f1\_score: 0.6605 - val\_loss: 1.2024 - val\_accuracy: 0.4939 - val\_precision: 0.4708 - val\_recall: 0.3779 - val\_f1\_score: 0.6518  
Epoch 19/50

Epoch 00019: LearningRateScheduler reducing learning rate to 7.3509177127562e -06.  
114/114 [=====] - 102s 848ms/step - loss: 0.8062 - accuracy: 0.5189 - precision: 0.5121 - recall: 0.5152 - f1\_score: 0.6605 - val\_loss: 1.2323 - val\_accuracy: 0.4933 - val\_precision: 0.4737 - val\_recall: 0.4332 - val\_f1\_score: 0.6518  
Epoch 20/50

Epoch 00020: LearningRateScheduler reducing learning rate to 7.3509177127562e -06.  
114/114 [=====] - 101s 838ms/step - loss: 0.8199 - accuracy: 0.5090 - precision: 0.5020 - recall: 0.5188 - f1\_score: 0.6605 - val\_loss: 1.1558 - val\_accuracy: 0.4962 - val\_precision: 0.4784 - val\_recall: 0.4657 - val\_f1\_score: 0.6518  
Epoch 21/50

Epoch 00021: LearningRateScheduler reducing learning rate to 6.98337182711838 9e-06.  
114/114 [=====] - 103s 849ms/step - loss: 0.7879 - accuracy: 0.5286 - precision: 0.5214 - recall: 0.5360 - f1\_score: 0.6605 - val\_loss: 1.0939 - val\_accuracy: 0.4828 - val\_precision: 0.4651 - val\_recall: 0.4645 - val\_f1\_score: 0.6518  
Epoch 22/50

Epoch 00022: LearningRateScheduler reducing learning rate to 6.98337180438102

2e-06.  
114/114 [=====] - 103s 857ms/step - loss: 0.7790 - accuracy: 0.5204 - precision: 0.5133 - recall: 0.5289 - f1\_score: 0.6605 - val\_loss: 1.0784 - val\_accuracy: 0.4916 - val\_precision: 0.4725 - val\_recall: 0.4452 - val\_f1\_score: 0.6518  
Epoch 23/50

Epoch 00023: LearningRateScheduler reducing learning rate to 6.98337180438102 2e-06.  
114/114 [=====] - 107s 878ms/step - loss: 0.7896 - accuracy: 0.5160 - precision: 0.5089 - recall: 0.5307 - f1\_score: 0.6605 - val\_loss: 1.1168 - val\_accuracy: 0.4805 - val\_precision: 0.4619 - val\_recall: 0.4525 - val\_f1\_score: 0.6518  
Epoch 24/50

Epoch 00024: LearningRateScheduler reducing learning rate to 6.63420321416197 04e-06.  
114/114 [=====] - 111s 884ms/step - loss: 0.7856 - accuracy: 0.5201 - precision: 0.5130 - recall: 0.5271 - f1\_score: 0.6605 - val\_loss: 1.0570 - val\_accuracy: 0.4823 - val\_precision: 0.4584 - val\_recall: 0.3911 - val\_f1\_score: 0.6518  
Epoch 25/50

Epoch 00025: LearningRateScheduler reducing learning rate to 6.63420314594986 85e-06.  
114/114 [=====] - 103s 859ms/step - loss: 0.7566 - accuracy: 0.5242 - precision: 0.5169 - recall: 0.5372 - f1\_score: 0.6605 - val\_loss: 1.0495 - val\_accuracy: 0.4799 - val\_precision: 0.4590 - val\_recall: 0.4248 - val\_f1\_score: 0.6518  
Epoch 26/50

Epoch 00026: LearningRateScheduler reducing learning rate to 6.63420314594986 85e-06.  
114/114 [=====] - 105s 882ms/step - loss: 0.7632 - accuracy: 0.5248 - precision: 0.5180 - recall: 0.5241 - f1\_score: 0.6605 - val\_loss: 1.0274 - val\_accuracy: 0.4817 - val\_precision: 0.4636 - val\_recall: 0.4597 - val\_f1\_score: 0.6518  
Epoch 27/50

Epoch 00027: LearningRateScheduler reducing learning rate to 6.30249298865237 4e-06.  
114/114 [=====] - 105s 875ms/step - loss: 0.7686 - accuracy: 0.5046 - precision: 0.4977 - recall: 0.5158 - f1\_score: 0.6605 - val\_loss: 1.0314 - val\_accuracy: 0.4887 - val\_precision: 0.4715 - val\_recall: 0.4777 - val\_f1\_score: 0.6518  
Epoch 28/50

Epoch 00028: LearningRateScheduler reducing learning rate to 6.30249314781394 8e-06.  
114/114 [=====] - 105s 865ms/step - loss: 0.7645 - accuracy: 0.5166 - precision: 0.5092 - recall: 0.5432 - f1\_score: 0.6605 - val\_loss: 1.0524 - val\_accuracy: 0.4881 - val\_precision: 0.4642 - val\_recall: 0.3827 - val\_f1\_score: 0.6518  
Epoch 29/50

Epoch 00029: LearningRateScheduler reducing learning rate to 6.30249314781394 8e-06.

114/114 [=====] - 109s 875ms/step - loss: 0.7933 - accuracy: 0.4963 - precision: 0.4897 - recall: 0.5092 - f1\_score: 0.6605 - val\_loss: 0.9831 - val\_accuracy: 0.4980 - val\_precision: 0.4789 - val\_recall: 0.4380 - val\_f1\_score: 0.6518  
Epoch 30/50

Epoch 00030: LearningRateScheduler reducing learning rate to 5.98736849042325e-06.

114/114 [=====] - 107s 881ms/step - loss: 0.7664 - accuracy: 0.5075 - precision: 0.5006 - recall: 0.5104 - f1\_score: 0.6605 - val\_loss: 0.9772 - val\_accuracy: 0.4916 - val\_precision: 0.4750 - val\_recall: 0.4910 - val\_f1\_score: 0.6518

Epoch 31/50

Epoch 00031: LearningRateScheduler reducing learning rate to 5.98736869505955e-06.

114/114 [=====] - 107s 883ms/step - loss: 0.7526 - accuracy: 0.5051 - precision: 0.4983 - recall: 0.5337 - f1\_score: 0.6605 - val\_loss: 1.0000 - val\_accuracy: 0.5067 - val\_precision: 0.4875 - val\_recall: 0.3995 - val\_f1\_score: 0.6518

Epoch 32/50

Epoch 00032: LearningRateScheduler reducing learning rate to 5.98736869505955e-06.

114/114 [=====] - 103s 856ms/step - loss: 0.7530 - accuracy: 0.5101 - precision: 0.5032 - recall: 0.5128 - f1\_score: 0.6605 - val\_loss: 1.0193 - val\_accuracy: 0.4933 - val\_precision: 0.4673 - val\_recall: 0.3442 - val\_f1\_score: 0.6518

Epoch 33/50

Epoch 00033: LearningRateScheduler reducing learning rate to 5.68800026030658e-06.

114/114 [=====] - 99s 817ms/step - loss: 0.7367 - accuracy: 0.5386 - precision: 0.5318 - recall: 0.5384 - f1\_score: 0.6605 - val\_loss: 0.9999 - val\_accuracy: 0.4834 - val\_precision: 0.4596 - val\_recall: 0.3899 - val\_f1\_score: 0.6518

Epoch 34/50

Epoch 00034: LearningRateScheduler reducing learning rate to 5.68800032851868e-06.

114/114 [=====] - 103s 855ms/step - loss: 0.7392 - accuracy: 0.5131 - precision: 0.5062 - recall: 0.5116 - f1\_score: 0.6605 - val\_loss: 0.9640 - val\_accuracy: 0.4718 - val\_precision: 0.4518 - val\_recall: 0.4344 - val\_f1\_score: 0.6518

Epoch 35/50

Epoch 00035: LearningRateScheduler reducing learning rate to 5.68800032851868e-06.

114/114 [=====] - 102s 848ms/step - loss: 0.7400 - accuracy: 0.5172 - precision: 0.5102 - recall: 0.5211 - f1\_score: 0.6605 - val\_loss: 0.9722 - val\_accuracy: 0.4834 - val\_precision: 0.4595 - val\_recall: 0.3887 - val\_f1\_score: 0.6518

Epoch 36/50

Epoch 00036: LearningRateScheduler reducing learning rate to 5.40360031209274e-06.

114/114 [=====] - 102s 844ms/step - loss: 0.7401 - a

```
accuracy: 0.5148 - precision: 0.5078 - recall: 0.5229 - f1_score: 0.6605 - val_loss: 0.9385 - val_accuracy: 0.4857 - val_precision: 0.4640 - val_recall: 0.4116 - val_f1_score: 0.6518
Epoch 37/50

Epoch 00037: LearningRateScheduler reducing learning rate to 5.40360042577958e-06.
114/114 [=====] - 102s 832ms/step - loss: 0.7312 - accuracy: 0.5213 - precision: 0.5144 - recall: 0.5200 - f1_score: 0.6605 - val_loss: 0.9693 - val_accuracy: 0.5044 - val_precision: 0.4858 - val_recall: 0.4320 - val_f1_score: 0.6518
Epoch 38/50

Epoch 00038: LearningRateScheduler reducing learning rate to 5.40360042577958e-06.
114/114 [=====] - 101s 830ms/step - loss: 0.7335 - accuracy: 0.5128 - precision: 0.5058 - recall: 0.5152 - f1_score: 0.6605 - val_loss: 0.9446 - val_accuracy: 0.4823 - val_precision: 0.4617 - val_recall: 0.4284 - val_f1_score: 0.6518
Epoch 39/50

Epoch 00039: LearningRateScheduler reducing learning rate to 5.13342040449060e-06.
114/114 [=====] - 102s 835ms/step - loss: 0.7462 - accuracy: 0.5104 - precision: 0.5036 - recall: 0.5027 - f1_score: 0.6605 - val_loss: 0.9310 - val_accuracy: 0.4904 - val_precision: 0.4709 - val_recall: 0.4380 - val_f1_score: 0.6518
Epoch 40/50

Epoch 00040: LearningRateScheduler reducing learning rate to 5.13342047270271e-06.
114/114 [=====] - 103s 853ms/step - loss: 0.7468 - accuracy: 0.5119 - precision: 0.5050 - recall: 0.5104 - f1_score: 0.6605 - val_loss: 0.9192 - val_accuracy: 0.4887 - val_precision: 0.4701 - val_recall: 0.4549 - val_f1_score: 0.6518
Epoch 41/50

Epoch 00041: LearningRateScheduler reducing learning rate to 5.13342047270271e-06.
114/114 [=====] - 102s 840ms/step - loss: 0.7300 - accuracy: 0.5257 - precision: 0.5183 - recall: 0.5384 - f1_score: 0.6605 - val_loss: 0.9159 - val_accuracy: 0.4916 - val_precision: 0.4706 - val_recall: 0.4140 - val_f1_score: 0.6518
Epoch 42/50

Epoch 00042: LearningRateScheduler reducing learning rate to 4.876749449067574e-06.
114/114 [=====] - 102s 851ms/step - loss: 0.7311 - accuracy: 0.5140 - precision: 0.5066 - recall: 0.5468 - f1_score: 0.6605 - val_loss: 0.8956 - val_accuracy: 0.4916 - val_precision: 0.4684 - val_recall: 0.3839 - val_f1_score: 0.6518
Epoch 43/50

Epoch 00043: LearningRateScheduler reducing learning rate to 4.8767492444312666e-06.
114/114 [=====] - 107s 849ms/step - loss: 0.7254 - accuracy: 0.5251 - precision: 0.5186 - recall: 0.5158 - f1_score: 0.6605 - val
```

```
_loss: 0.8852 - val_accuracy: 0.4898 - val_precision: 0.4670 - val_recall: 0.  
3911 - val_f1_score: 0.6518  
Epoch 44/50  
  
Epoch 00044: LearningRateScheduler reducing learning rate to 4.87674924443126  
66e-06.  
114/114 [=====] - 106s 848ms/step - loss: 0.7314 - a  
ccuracy: 0.5137 - precision: 0.5067 - recall: 0.5158 - f1_score: 0.6605 - val  
_loss: 0.8637 - val_accuracy: 0.4991 - val_precision: 0.4804 - val_recall: 0.  
4428 - val_f1_score: 0.6518  
Epoch 45/50  
  
Epoch 00045: LearningRateScheduler reducing learning rate to 4.63291178220970  
3e-06.  
114/114 [=====] - 103s 858ms/step - loss: 0.7330 - a  
ccuracy: 0.5263 - precision: 0.5185 - recall: 0.5515 - f1_score: 0.6605 - val  
_loss: 0.8635 - val_accuracy: 0.4968 - val_precision: 0.4728 - val_recall: 0.  
3562 - val_f1_score: 0.6518  
Epoch 46/50  
  
Epoch 00046: LearningRateScheduler reducing learning rate to 4.63291189589654  
1e-06.  
114/114 [=====] - 106s 865ms/step - loss: 0.7262 - a  
ccuracy: 0.5093 - precision: 0.5023 - recall: 0.5110 - f1_score: 0.6605 - val  
_loss: 0.8565 - val_accuracy: 0.4997 - val_precision: 0.4788 - val_recall: 0.  
3947 - val_f1_score: 0.6518  
Epoch 47/50  
  
Epoch 00047: LearningRateScheduler reducing learning rate to 4.63291189589654  
1e-06.  
114/114 [=====] - 105s 843ms/step - loss: 0.7301 - a  
ccuracy: 0.5087 - precision: 0.5018 - recall: 0.5063 - f1_score: 0.6605 - val  
_loss: 0.8582 - val_accuracy: 0.4898 - val_precision: 0.4660 - val_recall: 0.  
3791 - val_f1_score: 0.6518  
Epoch 48/50  
  
Epoch 00048: LearningRateScheduler reducing learning rate to 4.40126630110171  
36e-06.  
114/114 [=====] - 101s 833ms/step - loss: 0.7260 - a  
ccuracy: 0.5172 - precision: 0.5102 - recall: 0.5235 - f1_score: 0.6605 - val  
_loss: 0.8726 - val_accuracy: 0.4869 - val_precision: 0.4613 - val_recall: 0.  
3658 - val_f1_score: 0.6518  
Epoch 49/50  
  
Epoch 00049: LearningRateScheduler reducing learning rate to 4.40126632383908  
15e-06.  
114/114 [=====] - 102s 841ms/step - loss: 0.7297 - a  
ccuracy: 0.5175 - precision: 0.5101 - recall: 0.5396 - f1_score: 0.6605 - val  
_loss: 0.8645 - val_accuracy: 0.4869 - val_precision: 0.4550 - val_recall: 0.  
3105 - val_f1_score: 0.6518  
Epoch 50/50  
  
Epoch 00050: LearningRateScheduler reducing learning rate to 4.40126632383908  
15e-06.  
114/114 [=====] - 100s 832ms/step - loss: 0.7155 - a  
ccuracy: 0.5251 - precision: 0.5180 - recall: 0.5325 - f1_score: 0.6605 - val
```

```
_loss: 0.8617 - val_accuracy: 0.4956 - val_precision: 0.4689 - val_recall: 0.
3261 - val_f1_score: 0.6518
```

```
In [16]: model_checkpoint = Model(inputs=input, outputs=Out)
model_checkpoint.load_weights('/content/model_save/weights-01-0.4846.h5')
model_checkpoint.save('bestmodel_simpleconv_lstm.h5')
new_model = tf.keras.models.load_model('bestmodel_simpleconv_lstm.h5')
```

WARNING:tensorflow:No training configuration found in the save file, so the model was \*not\* compiled. Compile it manually.

```
In [17]: predict_train=new_model.predict_generator(train_prediction_generator,steps=len
(train_prediction_generator),workers=12)
predict_test=new_model.predict_generator(test_prediction_generator,steps=len(t
est_prediction_generator),workers=12)
predict_train.shape
```

Out[17]: (3405, 1)

```
In [18]: train_data_words=train_data['processed_words'].values
validation_words=val_data['processed_words'].values
test_data_words=test_data['processed_words'].values

tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_data_words)
vocab_size=len(tokenizer.word_index)
encoded_Xtrain_words = [tf.keras.preprocessing.text.one_hot(d, vocab_size,filt
ers='!"#%&()*+,-./:;<=>?@[\\]^{|}~\\t\\n') for d in train_data_words]
encoded_validation_words = [tf.keras.preprocessing.text.one_hot(d, vocab_size,
filters='!"#%&()*+,-./:;<=>?@[\\]^{|}~\\t\\n') for d in validation_words]
encoded_Xtest_words = [tf.keras.preprocessing.text.one_hot(d, vocab_size,filt
ers='!"#%&()*+,-./:;<=>?@[\\]^{|}~\\t\\n') for d in test_data_words]

padded_Xtrain_words = tf.keras.preprocessing.sequence.pad_sequences(encoded_Xt
rain_words, maxlen=20, padding='post')
padded_Xvalidation_words = tf.keras.preprocessing.sequence.pad_sequences(encode
d_validation_words, maxlen=20, padding='post')
padded_Xtest_words = tf.keras.preprocessing.sequence.pad_sequences(encoded_Xte
st_words, maxlen=20, padding='post')
```

```
In [19]: embeddings_index = dict()
f = open('/content/glove.6B.300d.txt')

for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs

f.close()
print('Loaded %s word vectors.' % len(embeddings_index))
```

Loaded 400000 word vectors.

```
In [20]: embedding_matrix = np.zeros((vocab_size+1, 300))
for word, i in tokenizer.word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector
```

```
In [21]: labelencoder = LabelEncoder()
labelencoder.fit(train_data[' subreddit'].values)
subreddit_train=labelencoder.transform(train_data[' subreddit'].values).reshape(-1,1)
subreddit_validation=labelencoder.transform(val_data[' subreddit'].values).reshape(-1,1)
subreddit_test=labelencoder.transform(test_data[' subreddit'].values).reshape(-1, 1)

print(subreddit_train.shape)
print(subreddit_test.shape)
print(subreddit_validation.shape)
```

(3405, 1)  
(1719, 1)  
(1688, 1)

```
In [22]: labelencoder = LabelEncoder()
labelencoder.fit(train_data[' is_nsfw'].values)
is_nsfw_train=labelencoder.transform(train_data[' is_nsfw'].values).reshape(-1, 1)
is_nsfw_validation=labelencoder.transform(val_data[' is_nsfw'].values).reshape(-1,1)
is_nsfw_test=labelencoder.transform(test_data[' is_nsfw'].values).reshape(-1,1)

print(is_nsfw_train.shape)
print(is_nsfw_test.shape)
print(is_nsfw_validation.shape)
```

(3405, 1)  
(1719, 1)  
(1688, 1)

```
In [23]: time_of_day_train=(train_data['time_of_day'].values).reshape(-1,1)
time_of_day_validation=(val_data['time_of_day'].values).reshape(-1,1)
time_of_day_test=(test_data['time_of_day'].values).reshape(-1,1)

print(time_of_day_train.shape)
print(time_of_day_validation.shape)
print(time_of_day_test.shape)
```

(3405, 1)  
(1688, 1)  
(1719, 1)

```
In [24]: scaler = StandardScaler()
scaler=scaler.fit(train_data['created_utc'].values.reshape(-1, 1))

created_utc_train=scaler.transform(train_data['created_utc'].values.reshape(-1, 1))
created_utc_validation=scaler.transform(val_data['created_utc'].values.reshape(-1, 1))
created_utc_test=scaler.transform(test_data['created_utc'].values.reshape(-1, 1))

print(created_utc_train.shape)
print(created_utc_test.shape)
print(created_utc_validation.shape)
```

```
(3405, 1)
(1719, 1)
(1688, 1)
```

```
In [25]: scaler = StandardScaler()
scaler=scaler.fit(train_data['subscribers'].values.reshape(-1, 1))

subscribers_train=scaler.transform(train_data['subscribers'].values.reshape(-1, 1))
subscribers_validation=scaler.transform(val_data['subscribers'].values.reshape(-1, 1))
subscribers_test=scaler.transform(test_data['subscribers'].values.reshape(-1, 1))

print(subscribers_train.shape)
print(subscribers_validation.shape)
print(subscribers_test.shape)
```

```
(3405, 1)
(1688, 1)
(1719, 1)
```

```
In [26]: #words embedding layer
words = Input(shape=(20,),name="words")
embedding=Embedding(vocab_size+1,300,weights=[embedding_matrix],input_length=20,trainable=False)(words)
lstm_layer=LSTM(300)(embedding)
flatten1 = Flatten(data_format='channels_last')(lstm_layer)

image_predicted =Input(shape=(predict_train.shape[1],),name="image_predicted")
flatten2= Flatten(data_format='channels_last')(image_predicted)

#categore_data
subreddit_train_layer =Input(shape=(subreddit_train.shape[1],),name="subreddit_train_layer")
flatten3= Flatten(data_format='channels_last')(subreddit_train_layer)
#####
is_nsfw_train_layer =Input(shape=(is_nsfw_train.shape[1],),name="is_nsfw_train_layer")
flatten4 = Flatten(data_format='channels_last')(is_nsfw_train_layer)
#####
time_of_day_train_layer =Input(shape=(time_of_day_train.shape[1],),name="time_of_day_train_layer")
flatten5 = Flatten(data_format='channels_last')(time_of_day_train_layer)

#numeric_data
created_utc_train_layer =Input(shape=(created_utc_train.shape[1],),name="created_utc_train_layer")
created_utc_dence = Dense(units=3,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=33))(created_utc_train_layer)

#numeric_data
subscribers_train_layer =Input(shape=(subscribers_train.shape[1],),name="subscribers_train_layer")
subscribers_dence = Dense(units=3,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=33))(subscribers_train_layer)

#concat layer
concatenated = concatenate([subscribers_train_layer,created_utc_dence,flatten5,flatten4,flatten3,flatten2,flatten1],axis = -1)

dense_layer1 = Dense(units=420,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=33))(concatenated)
dropout1=Dropout(0.3)(dense_layer1)

dense_layer2 = Dense(units=210,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=33))(dropout1)
dropout2=Dropout(0.3)(dense_layer2)

dense_layer3 = Dense(units=105,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=33))(dropout2)
dropout3=Dropout(0.3)(dense_layer3)

dense_layer4 = Dense(units=60,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=33))(dropout3)
dropout4=Dropout(0.3)(dense_layer4)

dense_layer5 = Dense(units=30,activation='relu',kernel_initializer=tf.keras.in
```

```
initializers.glorot_normal(seed=33))(dropout4)
dropout5=Dropout(0.3)(dense_layer5)

dense_layer6 = Dense(units=15,activation='relu',kernel_initializer=tf.keras.initializers.glorot_normal(seed=33))(dropout5)

Out = Dense(units=2,activation='softmax',kernel_initializer=tf.keras.initializers.glorot_normal(seed=3),name='Output')(dense_layer6)

model = Model(inputs=[image_predicted,words,subreddit_train_layer,is_nsfw_train_layer,time_of_day_train_layer,created_utc_train_layer,subscribers_train_layer],outputs=Out)
model.summary()
```

Model: "model\_2"

| Layer (type)                         | Output Shape    | Param # | Connected to                  |
|--------------------------------------|-----------------|---------|-------------------------------|
| words (InputLayer)                   | [None, 20]      | 0       |                               |
| embedding (Embedding)                | (None, 20, 300) | 2712000 | words[0][0]                   |
| created_utc_train_layer (InputLayer) | [None, 1]       | 0       |                               |
| time_of_day_train_layer (InputLayer) | [None, 1]       | 0       |                               |
| is_nsfw_train_layer (InputLayer)     | [None, 1]       | 0       |                               |
| subreddit_train_layer (InputLayer)   | [None, 1]       | 0       |                               |
| image_predicted (InputLayer)         | [None, 1]       | 0       |                               |
| lstm (LSTM)                          | (None, 300)     | 721200  | embedding[0]                  |
| [0]                                  |                 |         |                               |
| subscribers_train_layer (InputLayer) | [None, 1]       | 0       |                               |
| dense (Dense)                        | (None, 3)       | 6       | created_utc_train_layer[0][0] |
| train_layer[0][0]                    |                 |         |                               |
| flatten_4 (Flatten)                  | (None, 1)       | 0       | time_of_day_train_layer[0][0] |
| train_layer[0][0]                    |                 |         |                               |
| flatten_3 (Flatten)                  | (None, 1)       | 0       | is_nsfw_train_layer[0][0]     |
| train_layer[0][0]                    |                 |         |                               |
| flatten_2 (Flatten)                  | (None, 1)       | 0       | subreddit_train_layer[0][0]   |
| train_layer[0][0]                    |                 |         |                               |
| flatten_1 (Flatten)                  | (None, 1)       | 0       | image_predicted[0][0]         |
| train_layer[0][0]                    |                 |         |                               |
| flatten (Flatten)                    | (None, 300)     | 0       | lstm[0][0]                    |
|                                      |                 |         |                               |

|  |              |        |              |
|--|--------------|--------|--------------|
| concatenate (Concatenate)<br>train_layer[0][0] | (None, 308)  | 0      | subscribers_ |
| [0]  | dense[0][0]  |        | flatten_4[0] |
| [0]  | flatten_3[0] |        |              |
| [0]  | flatten_2[0] |        |              |
| [0]  | flatten_1[0] |        |              |
| [0]  | flatten[0]   |        |              |
| <hr/>  |              |        |              |
| dense_2 (Dense)<br>[0][0]                      | (None, 420)  | 129780 | concatenate  |
| <hr/>  |              |        |              |
| dropout (Dropout)<br>[0]                       | (None, 420)  | 0      | dense_2[0]   |
| <hr/>  |              |        |              |
| dense_3 (Dense)<br>[0]                         | (None, 210)  | 88410  | dropout[0]   |
| <hr/>  |              |        |              |
| dropout_1 (Dropout)<br>[0]                     | (None, 210)  | 0      | dense_3[0]   |
| <hr/>  |              |        |              |
| dense_4 (Dense)<br>[0]                         | (None, 105)  | 22155  | dropout_1[0] |
| <hr/>  |              |        |              |
| dropout_2 (Dropout)<br>[0]                     | (None, 105)  | 0      | dense_4[0]   |
| <hr/>  |              |        |              |
| dense_5 (Dense)<br>[0]                         | (None, 60)   | 6360   | dropout_2[0] |
| <hr/>  |              |        |              |
| dropout_3 (Dropout)<br>[0]                     | (None, 60)   | 0      | dense_5[0]   |
| <hr/>  |              |        |              |
| dense_6 (Dense)<br>[0]                         | (None, 30)   | 1830   | dropout_3[0] |
| <hr/>  |              |        |              |
| dropout_4 (Dropout)<br>[0]                     | (None, 30)   | 0      | dense_6[0]   |
| <hr/>  |              |        |              |
| dense_7 (Dense)<br>[0]                         | (None, 15)   | 465    | dropout_4[0] |

---

|                                 |           |    |            |
|---------------------------------|-----------|----|------------|
| Output (Dense)<br>[0]           | (None, 2) | 32 | dense_7[0] |
| =====                           |           |    |            |
| Total params: 3,682,238         |           |    |            |
| Trainable params: 970,238       |           |    |            |
| Non-trainable params: 2,712,000 |           |    |            |

---



In [27]:

```
filepath="model_save_lstm/weights-{epoch:02d}-{val_accuracy:.4f}.h5"
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_accuracy', mode='auto')

lrschedule = tf.keras.callbacks.LearningRateScheduler(scheduler, verbose=0.1)

#stop the training if your validation accuracy is not increased in last 2 epochs.
early_stop= EarlyStopping(monitor='val_accuracy', patience=2, verbose=1)

#If your validation accuracy at that epoch is Less than previous epoch accuracy, you have to decrease the learning rate by 10%
reduce_lr = ReduceLROnPlateau(monitor='val_accuracy', factor=0.9,
                               patience=0, min_lr=0.001, verbose=1)

model.compile(
    loss='categorical_crossentropy',
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999, epsilon=1e-07, amsgrad=False),
    metrics=[ 'accuracy',tf.keras.metrics.Precision(),tf.keras.metrics.Recall(),tf.keras.metrics.F1Score(num_classes=2)]
)
```

In [28]:

```
y_train =tf.keras.utils.to_categorical(train_data['dank_level'].values,2)
y_test =tf.keras.utils.to_categorical(test_data['dank_level'].values,2)
y_test.shape
```

Out[28]: (1719, 2)

```
In [30]: history=model.fit({"image_predicted":predict_train,"words":padded_Xtrain_words  
," subreddit_train_layer":subreddit_train,"is_nsfw_train_layer":is_nsfw_train,  
"time_of_day_train_layer":time_of_day_train,  
"created_utc_train_layer":created_utc_train,"subscribers_t  
rain_layer":subscribers_train},  
y_train,epochs=20,batch_size=30,  
validation_data=( {"image_predicted":predict_test,"words":pad  
ded_Xtest_words," subreddit_train_layer":subreddit_test,"is_nsfw_train_layer":i  
s_nsfw_test,"time_of_day_train_layer":time_of_day_test,  
"created_utc_train_layer":created_utc_test,"subscribers_tr  
ain_layer":subscribers_test},y_test),callbacks=[lrschedule,checkpoint,reduce_lr])
```

Epoch 1/20

Epoch 00001: LearningRateScheduler reducing learning rate to 9.9999974737875  
2e-05.

114/114 [=====] - 4s 16ms/step - loss: 0.7030 - accuracy: 0.5149 - precision\_1: 0.5149 - recall\_1: 0.5149 - f1\_score: 0.4653 - val\_loss: 0.6909 - val\_accuracy: 0.5282 - val\_precision\_1: 0.5282 - val\_recall\_1: 0.5282 - val\_f1\_score: 0.3969

Epoch 2/20

Epoch 00002: LearningRateScheduler reducing learning rate to 9.9999974737875  
2e-05.

114/114 [=====] - 1s 10ms/step - loss: 0.6947 - accuracy: 0.5136 - precision\_1: 0.5136 - recall\_1: 0.5136 - f1\_score: 0.5098 - val\_loss: 0.6903 - val\_accuracy: 0.5294 - val\_precision\_1: 0.5294 - val\_recall\_1: 0.5294 - val\_f1\_score: 0.3968

Epoch 3/20

Epoch 00003: LearningRateScheduler reducing learning rate to 9.4999976000981  
4e-05.

114/114 [=====] - 1s 10ms/step - loss: 0.6912 - accuracy: 0.5077 - precision\_1: 0.5077 - recall\_1: 0.5077 - f1\_score: 0.5058 - val\_loss: 0.6897 - val\_accuracy: 0.5480 - val\_precision\_1: 0.5480 - val\_recall\_1: 0.5480 - val\_f1\_score: 0.5298

Epoch 4/20

Epoch 00004: LearningRateScheduler reducing learning rate to 9.4999957811087  
4e-05.

114/114 [=====] - 1s 10ms/step - loss: 0.6899 - accuracy: 0.5432 - precision\_1: 0.5432 - recall\_1: 0.5432 - f1\_score: 0.5417 - val\_loss: 0.6893 - val\_accuracy: 0.5462 - val\_precision\_1: 0.5462 - val\_recall\_1: 0.5462 - val\_f1\_score: 0.5108

Epoch 5/20

Epoch 00005: LearningRateScheduler reducing learning rate to 9.4999957811087  
4e-05.

114/114 [=====] - 1s 11ms/step - loss: 0.6962 - accuracy: 0.5051 - precision\_1: 0.5051 - recall\_1: 0.5051 - f1\_score: 0.5015 - val\_loss: 0.6895 - val\_accuracy: 0.5398 - val\_precision\_1: 0.5398 - val\_recall\_1: 0.5398 - val\_f1\_score: 0.5344

Epoch 6/20

Epoch 00006: LearningRateScheduler reducing learning rate to 9.02499959920533  
e-05.

114/114 [=====] - 1s 11ms/step - loss: 0.6908 - accuracy: 0.5111 - precision\_1: 0.5111 - recall\_1: 0.5111 - f1\_score: 0.5071 - val\_loss: 0.6889 - val\_accuracy: 0.5369 - val\_precision\_1: 0.5369 - val\_recall\_1: 0.5369 - val\_f1\_score: 0.5303

Epoch 7/20

Epoch 00007: LearningRateScheduler reducing learning rate to 9.02499959920533  
e-05.

114/114 [=====] - 1s 11ms/step - loss: 0.6912 - accuracy: 0.5186 - precision\_1: 0.5186 - recall\_1: 0.5186 - f1\_score: 0.5141 - val\_loss: 0.6893 - val\_accuracy: 0.5288 - val\_precision\_1: 0.5288 - val\_recall\_1: 0.5288 - val\_f1\_score: 0.5286

Epoch 8/20

Epoch 00008: LearningRateScheduler reducing learning rate to 9.02499959920533e-05.  
114/114 [=====] - 1s 10ms/step - loss: 0.6913 - accuracy: 0.5354 - precision\_1: 0.5354 - recall\_1: 0.5354 - f1\_score: 0.5352 - val\_loss: 0.6891 - val\_accuracy: 0.5300 - val\_precision\_1: 0.5300 - val\_recall\_1: 0.5300 - val\_f1\_score: 0.5223  
Epoch 9/20

Epoch 00009: LearningRateScheduler reducing learning rate to 8.573749619245064e-05.  
114/114 [=====] - 1s 10ms/step - loss: 0.6903 - accuracy: 0.5296 - precision\_1: 0.5296 - recall\_1: 0.5296 - f1\_score: 0.5238 - val\_loss: 0.6892 - val\_accuracy: 0.5311 - val\_precision\_1: 0.5311 - val\_recall\_1: 0.5311 - val\_f1\_score: 0.5228  
Epoch 10/20

Epoch 00010: LearningRateScheduler reducing learning rate to 8.573749801144004e-05.  
114/114 [=====] - 1s 11ms/step - loss: 0.6909 - accuracy: 0.5269 - precision\_1: 0.5269 - recall\_1: 0.5269 - f1\_score: 0.5256 - val\_loss: 0.6889 - val\_accuracy: 0.5352 - val\_precision\_1: 0.5352 - val\_recall\_1: 0.5352 - val\_f1\_score: 0.5287  
Epoch 11/20

Epoch 00011: LearningRateScheduler reducing learning rate to 8.573749801144004e-05.  
114/114 [=====] - 1s 11ms/step - loss: 0.6880 - accuracy: 0.5296 - precision\_1: 0.5296 - recall\_1: 0.5296 - f1\_score: 0.5275 - val\_loss: 0.6888 - val\_accuracy: 0.5358 - val\_precision\_1: 0.5358 - val\_recall\_1: 0.5358 - val\_f1\_score: 0.5353  
Epoch 12/20

Epoch 00012: LearningRateScheduler reducing learning rate to 8.145062311086804e-05.  
114/114 [=====] - 1s 10ms/step - loss: 0.6869 - accuracy: 0.5553 - precision\_1: 0.5553 - recall\_1: 0.5553 - f1\_score: 0.5509 - val\_loss: 0.6882 - val\_accuracy: 0.5398 - val\_precision\_1: 0.5398 - val\_recall\_1: 0.5398 - val\_f1\_score: 0.5362  
Epoch 13/20

Epoch 00013: LearningRateScheduler reducing learning rate to 8.145062020048499e-05.  
114/114 [=====] - 1s 11ms/step - loss: 0.6898 - accuracy: 0.5427 - precision\_1: 0.5427 - recall\_1: 0.5427 - f1\_score: 0.5412 - val\_loss: 0.6881 - val\_accuracy: 0.5358 - val\_precision\_1: 0.5358 - val\_recall\_1: 0.5358 - val\_f1\_score: 0.5358  
Epoch 14/20

Epoch 00014: LearningRateScheduler reducing learning rate to 8.145062020048499e-05.  
114/114 [=====] - 1s 11ms/step - loss: 0.6894 - accuracy: 0.5520 - precision\_1: 0.5520 - recall\_1: 0.5520 - f1\_score: 0.5515 - val\_loss: 0.6877 - val\_accuracy: 0.5404 - val\_precision\_1: 0.5404 - val\_recall\_1: 0.5404 - val\_f1\_score: 0.5370  
Epoch 15/20

```

Epoch 00015: LearningRateScheduler reducing learning rate to 7.73780891904607
4e-05.
114/114 [=====] - 1s 10ms/step - loss: 0.6849 - accuracy: 0.5623 - precision_1: 0.5623 - recall_1: 0.5623 - f1_score: 0.5536 - val_loss: 0.6874 - val_accuracy: 0.5398 - val_precision_1: 0.5398 - val_recall_1: 0.5398 - val_f1_score: 0.5386
Epoch 16/20

Epoch 00016: LearningRateScheduler reducing learning rate to 7.73780921008437
9e-05.
114/114 [=====] - 1s 10ms/step - loss: 0.6915 - accuracy: 0.5406 - precision_1: 0.5406 - recall_1: 0.5406 - f1_score: 0.5398 - val_loss: 0.6874 - val_accuracy: 0.5398 - val_precision_1: 0.5398 - val_recall_1: 0.5398 - val_f1_score: 0.5364
Epoch 17/20

Epoch 00017: LearningRateScheduler reducing learning rate to 7.73780921008437
9e-05.
114/114 [=====] - 1s 10ms/step - loss: 0.6902 - accuracy: 0.5266 - precision_1: 0.5266 - recall_1: 0.5266 - f1_score: 0.5187 - val_loss: 0.6872 - val_accuracy: 0.5416 - val_precision_1: 0.5416 - val_recall_1: 0.5416 - val_f1_score: 0.5401
Epoch 18/20

Epoch 00018: LearningRateScheduler reducing learning rate to 7.35091874958015
9e-05.
114/114 [=====] - 1s 11ms/step - loss: 0.6876 - accuracy: 0.5473 - precision_1: 0.5473 - recall_1: 0.5473 - f1_score: 0.5444 - val_loss: 0.6869 - val_accuracy: 0.5404 - val_precision_1: 0.5404 - val_recall_1: 0.5404 - val_f1_score: 0.5370
Epoch 19/20

Epoch 00019: LearningRateScheduler reducing learning rate to 7.35091889509931
2e-05.
114/114 [=====] - 1s 10ms/step - loss: 0.6839 - accuracy: 0.5660 - precision_1: 0.5660 - recall_1: 0.5660 - f1_score: 0.5654 - val_loss: 0.6869 - val_accuracy: 0.5422 - val_precision_1: 0.5422 - val_recall_1: 0.5422 - val_f1_score: 0.5380
Epoch 20/20

Epoch 00020: LearningRateScheduler reducing learning rate to 7.35091889509931
2e-05.
114/114 [=====] - 1s 11ms/step - loss: 0.6866 - accuracy: 0.5618 - precision_1: 0.5618 - recall_1: 0.5618 - f1_score: 0.5565 - val_loss: 0.6873 - val_accuracy: 0.5422 - val_precision_1: 0.5422 - val_recall_1: 0.5422 - val_f1_score: 0.5386

```

In [33]:

```

model_checkpoint = Model(inputs=[image_predicted,words,subreddit_train_layer,i
s_nsfw_train_layer,time_of_day_train_layer,created_utc_train_layer,subscribers
_train_layer],outputs=Out)
model_checkpoint.load_weights('/content/model_save_lstm/weights-03-0.5480.h5')
model_checkpoint.save('bestmodel_simpleconv_lstm_1.h5')
new_model = tf.keras.models.load_model('bestmodel_simpleconv_lstm_1.h5')

```

WARNING:tensorflow:No training configuration found in the save file, so the model was \*not\* compiled. Compile it manually.

```
In [35]: test_prediction=new_model.predict([predict_test,padded_Xtest_words,subreddit_test,
                                         is_nsfw_test,time_of_day_test,
                                         created_utc_test,subscribers_test])
test_prediction=np.argmax(test_prediction,axis=-1)
print(test_prediction.shape)
y_test =tf.keras.utils.to_categorical(test_data[ 'dank_level'].values,2)
y_test=np.argmax(y_test,axis=-1)
y_test.shape
```

(1719,)

Out[35]: (1719,)

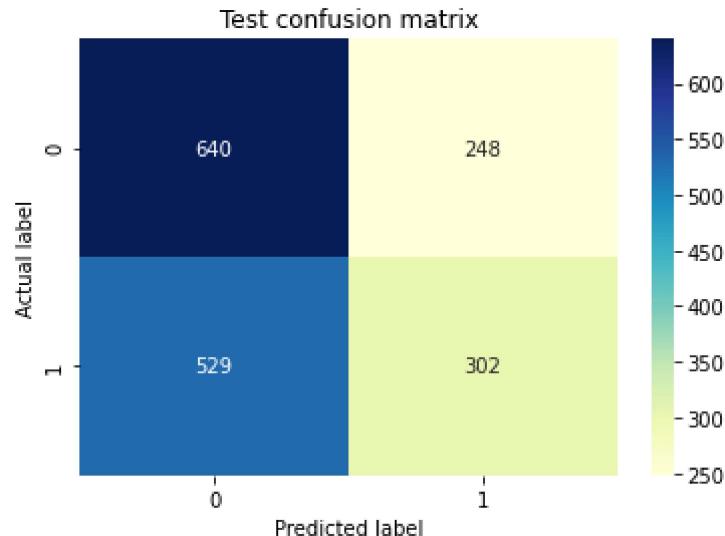
```
In [36]: accuracy=accuracy_score(y_test,test_prediction)
print("Test accuracy_score",accuracy)
f1_test_score=f1_score(y_test,test_prediction)
print("Test F1_score",f1_test_score)
print("Test confusion matrix")
cnf_matrix2=confusion_matrix(y_test,test_prediction)
p = sns.heatmap(pd.DataFrame(cnf_matrix2), annot=True, cmap="YlGnBu" ,fmt='g')
plt.title('Test confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

Test accuracy\_score 0.5479930191972077

Test F1\_score 0.43736422881969583

Test confusion matrix

Out[36]: Text(0.5, 15.0, 'Predicted label')



```
In [37]: file = '/content/model_1.png'
tf.keras.utils.plot_model(model,to_file=file, show_shapes=True)
```

Out[37]:

