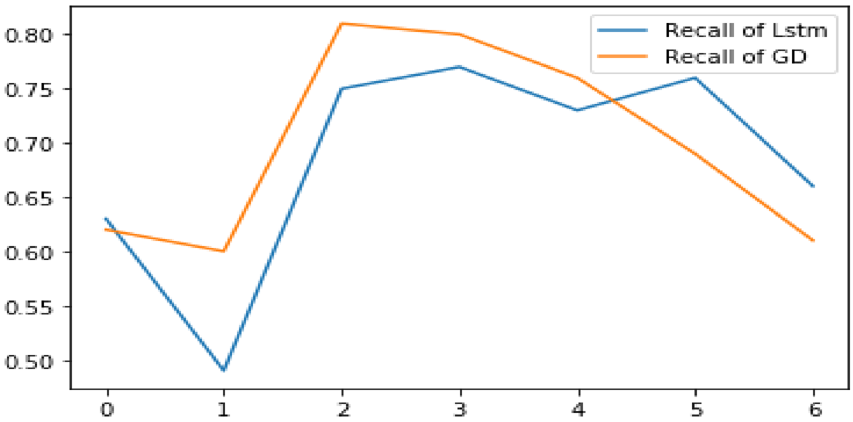
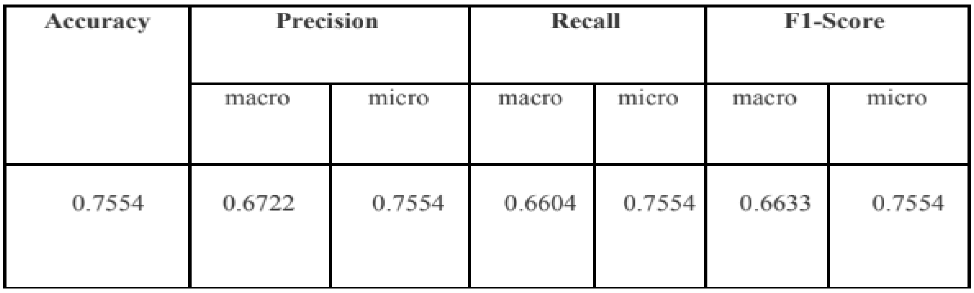
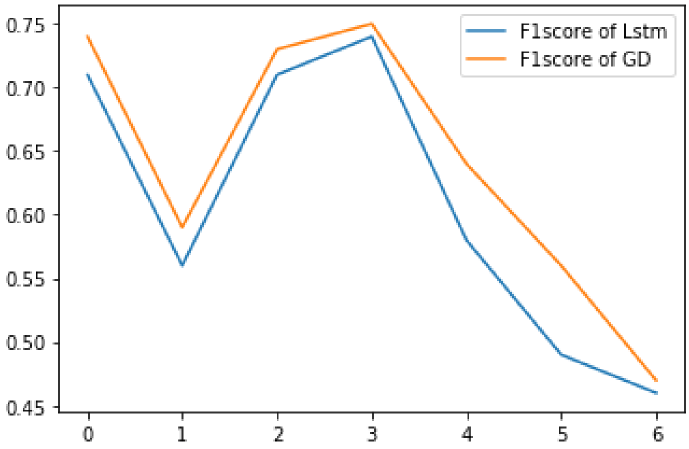
RESULTS OF THE PROJECT (SCREEN SHOTS)

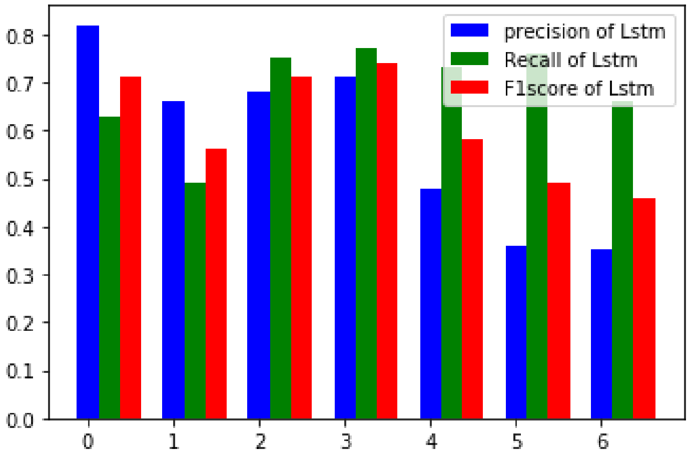
LSTM RESULTS



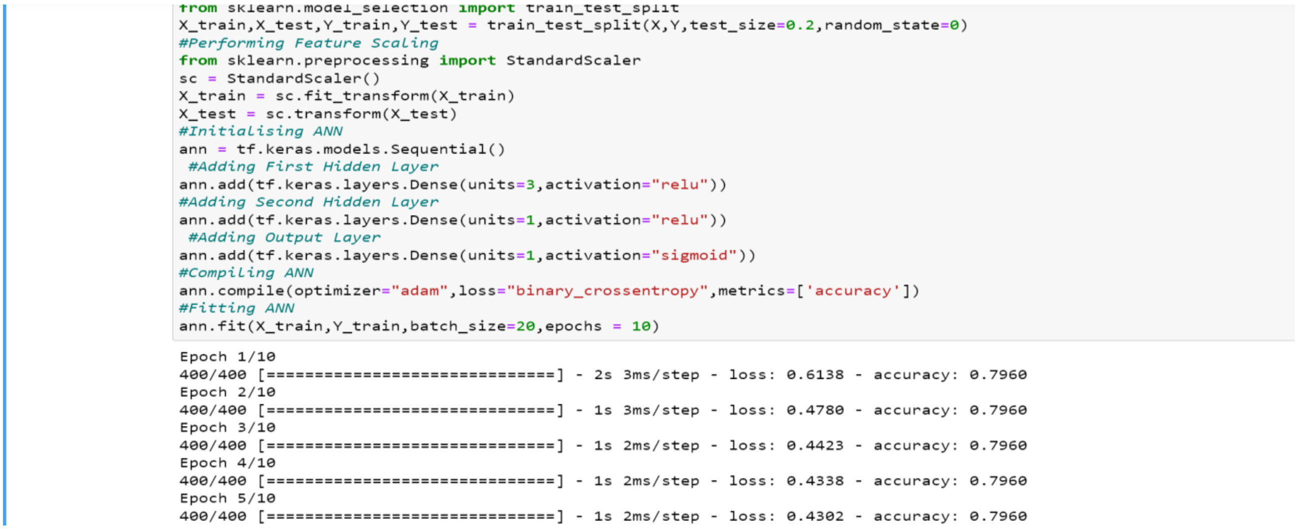


Accuracy	Precision		Recall		F1-Score	
	macro	micro	macro	micro	macro	micro
0.7554	0.6722	0.7554	0.6604	0.7554	0.6633	0.7554





ANN RESULTS



```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=0)
#Performing Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
#Initialising ANN
ann = tf.keras.models.Sequential()
 #Adding First Hidden Layer
ann.add(tf.keras.layers.Dense(units=3,activation="relu"))
#Adding Second Hidden Layer
ann.add(tf.keras.layers.Dense(units=1,activation="relu"))
 #Adding Output Layer
ann.add(tf.keras.layers.Dense(units=1,activation="sigmoid"))
#Compiling ANN
ann.compile(optimizer="adam",loss="binary_crossentropy",metrics=['accuracy'])
#Fitting ANN
ann.fit(X_train,Y_train,batch_size=20,epochs = 10)
```

```
Epoch 1/10
400/400 [==============================] - 2s 3ms/step - loss: 0.6138 - accuracy: 0.7960
Epoch 2/10
400/400 [==============================] - 1s 3ms/step - loss: 0.4780 - accuracy: 0.7960
Epoch 3/10
400/400 [==============================] - 1s 2ms/step - loss: 0.4423 - accuracy: 0.7960
Epoch 4/10
400/400 [==============================] - 1s 2ms/step - loss: 0.4338 - accuracy: 0.7960
Epoch 5/10
400/400 [==============================] - 1s 2ms/step - loss: 0.4302 - accuracy: 0.7960
```