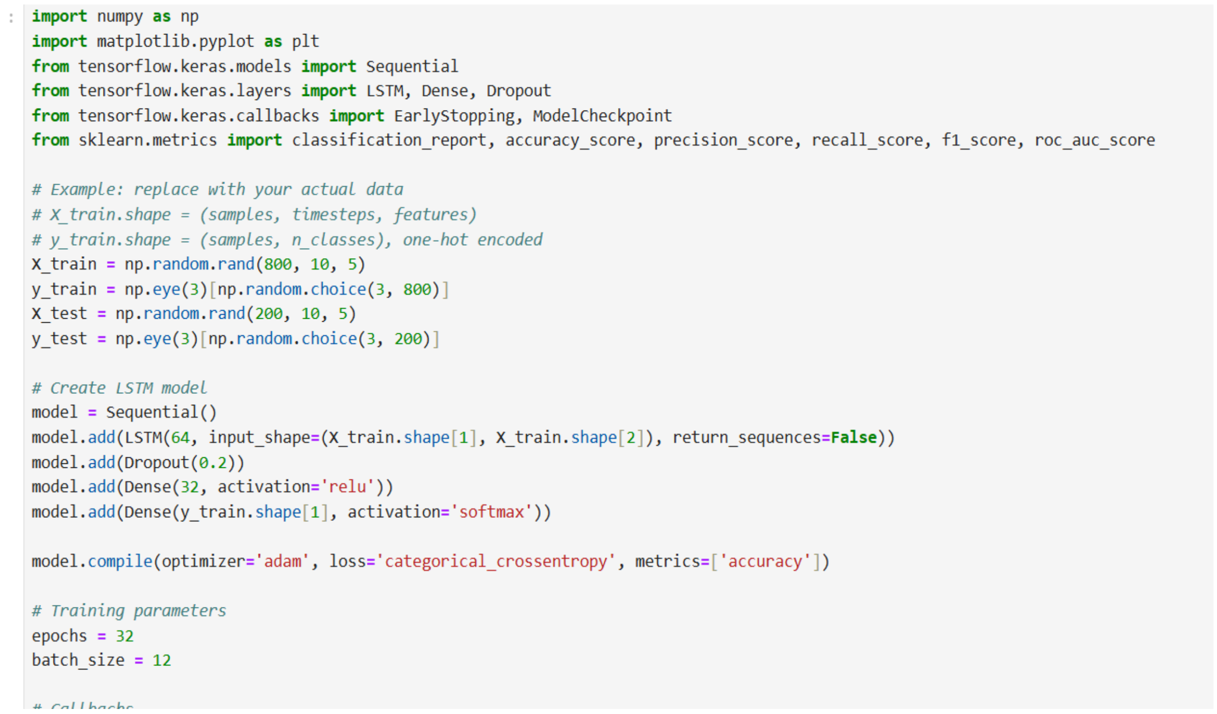
NAME : SRAVAN KUMAR REDDY

SID : 2465382

ASSIGNMENT -11

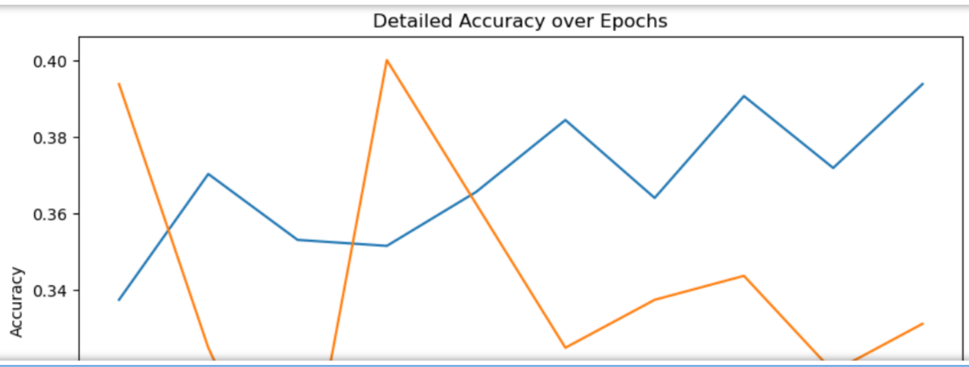


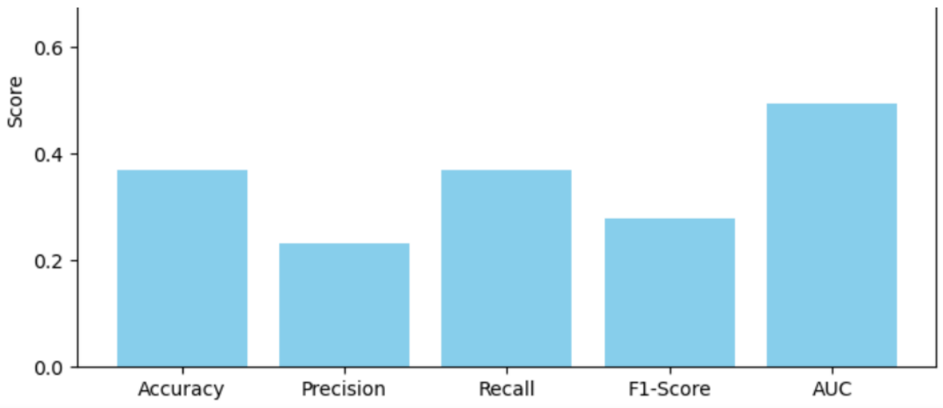
```python
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from sklearn.metrics import classification_report, accuracy_score, precision_score, recall_score, f1_score, roc_auc_score

# Example: replace with your actual data
# X_train.shape = (samples, timesteps, features)
# y_train.shape = (samples, n_classes), one-hot encoded
X_train = np.random.rand(800, 10, 5)
y_train = np.eye(3)[np.random.choice(3, 800)]
X_test = np.random.rand(200, 10, 5)
y_test = np.eye(3)[np.random.choice(3, 200)]

# Create LSTM model
model = Sequential()
model.add(LSTM(64, input_shape=(X_train.shape[1], X_train.shape[2]), return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(Dense(y_train.shape[1], activation='softmax'))

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

# Training parameters
epochs = 32
batch_size = 12
```

```python
print("Accuracy:", acc)
print("Precision:", prec)
print("Recall:", rec)
print("F1-Score:", f1)
print("AUC:", auc)

# Plot detailed accuracy
plt.figure(figsize=(10,6))
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Detailed Accuracy over Epochs')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

# Add metrics to ML Models performance histogram
metrics = [acc, prec, rec, f1, auc]
metrics_labels = ['Accuracy', 'Precision', 'Recall', 'F1-Score', 'AUC']

plt.figure(figsize=(8,5))
plt.bar(metrics_labels, metrics, color='skyblue')
plt.title("ML Models Performance (LSTM)")
plt.ylim(0, 1)
plt.ylabel("Score")
plt.show()
```

```
Epoch 1/32
49/54 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.3123 - loss: 1.1081
Epoch 1: val_accuracy improved from None to 0.35625, saving model to best_lstm_model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is cor
recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.kera
54/54 ━━━━━━━━━━━━━━━━━━━━ 5s 21ms/step - accuracy: 0.3391 - loss: 1.1023 - val_accuracy: 0.3562 - val_loss: 1.1028
Epoch 2/32
45/54 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.3903 - loss: 1.0902
Epoch 2: val_accuracy improved from 0.35625 to 0.36875, saving model to best_lstm_model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is cor
recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.kera
54/54 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - accuracy: 0.3734 - loss: 1.0968 - val_accuracy: 0.3688 - val_loss: 1.0996
Epoch 3/32
46/54 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.3455 - loss: 1.0994
Epoch 3: val_accuracy did not improve from 0.36875
54/54 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.3516 - loss: 1.1010 - val_accuracy: 0.3562 - val_loss: 1.0980
Epoch 4/32
48/54 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.3210 - loss: 1.1066
Epoch 4: val_accuracy did not improve from 0.36875
54/54 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - accuracy: 0.3781 - loss: 1.0964 - val_accuracy: 0.3562 - val_loss: 1.0998
Epoch 5/32
47/54 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.3782 - loss: 1.0928
Epoch 5: val_accuracy did not improve from 0.36875
54/54 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.3688 - loss: 1.0957 - val_accuracy: 0.3625 - val_loss: 1.0972
Epoch 6/32
45/54 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.3789 - loss: 1.0933
Epoch 6: val_accuracy did not improve from 0.36875
54/54 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.3547 - loss: 1.0940 - val_accuracy: 0.3625 - val_loss: 1.0996
Epoch 7/32
53/54 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.3618 - loss: 1.0928
Epoch 7: val_accuracy did not improve from 0.36875
```

```
53/54 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.3618 - loss: 1.0928
Epoch 7: val_accuracy did not improve from 0.36875
54/54 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - accuracy: 0.3594 - loss: 1.0943 - val_accuracy: 0.3438 - val_loss: 1.1119
Epoch 8/32
53/54 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.3427 - loss: 1.0968
Epoch 8: val_accuracy improved from 0.36875 to 0.38125, saving model to best_lstm_model.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is co
recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.ker
54/54 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.3547 - loss: 1.0957 - val_accuracy: 0.3812 - val_loss: 1.0999
Epoch 9/32
43/54 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.4073 - loss: 1.0862
Epoch 9: val_accuracy did not improve from 0.38125
54/54 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.3734 - loss: 1.0908 - val_accuracy: 0.3562 - val_loss: 1.1095
Epoch 10/32
52/54 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.3877 - loss: 1.0948
Epoch 10: val_accuracy did not improve from 0.38125
54/54 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.3750 - loss: 1.0940 - val_accuracy: 0.3562 - val_loss: 1.1019
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 47ms/step
C:\Users\B sravan kumar reddy\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:1565: UndefinedMetricWarning: Precision
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
Accuracy: 0.395
Precision: 0.156025
Recall: 0.395
F1-Score: 0.22369175627240143
AUC: 0.505693952870721
```


Detailed Accuracy over Epochs
Accuracy


Score
Accuracy
Precision
Recall
F1-Score
AUC