2. BUILDING A END TO END SPEECH RECOGINITION PIPE LINE SIGNAL PROCESSING ACOUSTIC MODELING &PERFORMANCE EVALUATION.

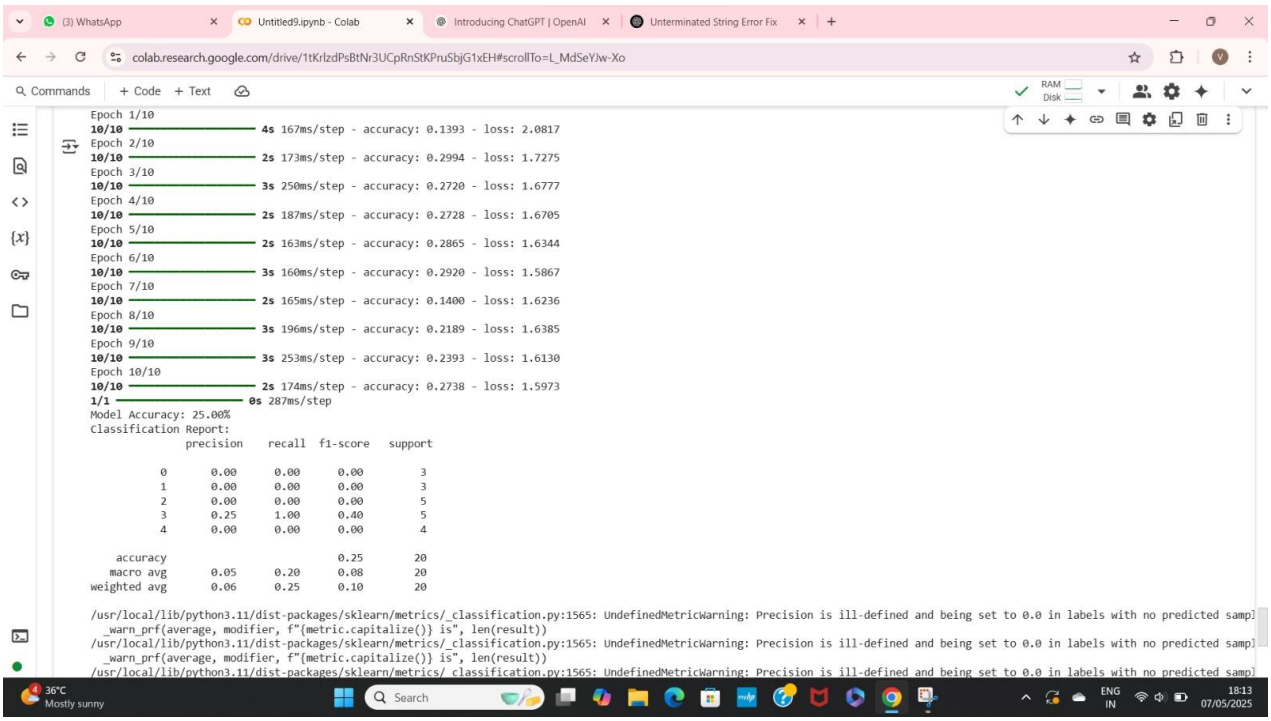
Program:

```
import  numpy  as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Input
from sklearn.metrics import accuracy_score, classification_report
Step 1: Signal Processing (Simulated)
def preprocess_signal(signal, sample_rate=16000):
    """
    Simulate signal processing: Normalize and extract MFCC-like features.
    """
    # Normalize signal
    signal = signal / np.max(np.abs(signal))
    # Simulate feature extraction
    features = np.log1p(np.abs(np.fft.fft(signal))[:len(signal) // 2])
    return features
Step 2: Acoustic Modeling
def build_acoustic_model(input_dim, output_dim):
    """
    Build a simple LSTM-based acoustic model.
    """
    model = Sequential([
```

```
        Input(shape=(None, input_dim)),
        LSTM(128, return_sequences=False),
        Dense(output_dim, activation='softmax')
    ])
    model.compile(optimizer='adam',
loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    return model
# Step 3: Performance Evaluation
def evaluate_model(model, X_test, y_test):
    """
Evaluate the model on the test set.
    """
    predictions = model.predict(X_test)
    predicted_labels = np.argmax(predictions, axis=1)
    accuracy = accuracy_score(y_test, predicted_labels)
    report = classification_report(y_test, predicted_labels)
    return accuracy, report


# Simulated Pipeline
if __name__ == "__main__":
    # Simulate raw audio signal (sine wave + noise)
    duration = 1.0  # seconds
    sample_rate = 16000
    time = np.linspace(0, duration, int(sample_rate * duration),
endpoint=False)
```

```
    signal = 0.5 * np.sin(2 * np.pi * 440 * time) + 0.1 *
np.random.randn(len(time))


    # Step 1: Signal Processing
    features = preprocess_signal(signal)


    # Simulate Dataset
    num_samples = 100
    input_dim = features.shape[0]
    X = np.random.rand(num_samples, 10, input_dim)  # Simulated
features
    y = np.random.randint(0, 5, size=(num_samples,))  # Simulated
labels (5 classes)
      # Split into train and test sets
    split_idx = int(0.8 * num_samples)
    X_train, X_test = X[:split_idx], X[split_idx:]
    y_train, y_test = y[:split_idx], y[split_idx:]
     # Step 2: Acoustic Modeling
    model = build_acoustic_model(input_dim, output_dim=5)
    model.fit(X_train, y_train, epochs=10, batch_size=8, verbose=1)
     # Step 3: Performance Evaluation
    accuracy, report = evaluate_model(model, X_test, y_test)
    print(f"Model Accuracy: {accuracy * 100:.2f}%")
    print("Classification Report:\n", report)
```

output:


(3) WhatsApp
Untitled9.ipynb - Colab
Introducing ChatGPT | OpenAI
Unterminated String Error Fix
colab.research.google.com/drive/1tKrlzdPsBtNr3UCpRnStKPruSbjG1xEH#scrollTo=L_MdSeYJw-Xo
Commands
+ Code
+ Text
RAM
Disk
Epoch 1/10
10/10 4s 167ms/step - accuracy: 0.1393 - loss: 2.0817
Epoch 2/10
10/10 2s 173ms/step - accuracy: 0.2994 - loss: 1.7275
Epoch 3/10
10/10 3s 250ms/step - accuracy: 0.2720 - loss: 1.6777
Epoch 4/10
10/10 2s 187ms/step - accuracy: 0.2728 - loss: 1.6705
Epoch 5/10
10/10 2s 163ms/step - accuracy: 0.2865 - loss: 1.6344
Epoch 6/10
10/10 3s 160ms/step - accuracy: 0.2920 - loss: 1.5867
Epoch 7/10
10/10 2s 165ms/step - accuracy: 0.1400 - loss: 1.6236
Epoch 8/10
10/10 3s 196ms/step - accuracy: 0.2189 - loss: 1.6385
Epoch 9/10
10/10 3s 253ms/step - accuracy: 0.2393 - loss: 1.6130
Epoch 10/10
10/10 2s 174ms/step - accuracy: 0.2738 - loss: 1.5973
1/1 0s 287ms/step
Model Accuracy: 25.00%
Classification Report:
precision recall f1-score support
0 0.00 0.00 0.00 3
1 0.00 0.00 0.00 3
2 0.00 0.00 0.00 5
3 0.25 1.00 0.40 5
4 0.00 0.00 0.00 4
accuracy 0.25 20
macro avg 0.05 0.20 0.08 20
weighted avg 0.06 0.25 0.10 20
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being 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))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being 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))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use zero_division parameter to control this behavior.
36°C
Mostly sunny
Search
ENG
IN
18:13
07/05/2025