```
data = pd.read_csv('/content/sentimentdataset.csv')
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Embedding, SpatialDropout1D, Concatenate, Input
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences


sentiment_encoder = LabelEncoder()
data['Sentiment'] = sentiment_encoder.fit_transform(data['Sentiment'])


data['Hour_sin'] = np.sin(2 * np.pi * data['Hour'] / 24)
data['Hour_cos'] = np.cos(2 * np.pi * data['Hour'] / 24)


X = data[['Text', 'Sentiment', 'Hour_sin', 'Hour_cos']]
y = data['Sentiment']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


tokenizer = Tokenizer(num_words=1000)
tokenizer.fit_on_texts(X_train['Text'])
X_text_train = tokenizer.texts_to_sequences(X_train['Text'])
X_text_test = tokenizer.texts_to_sequences(X_test['Text'])
X_text_train = pad_sequences(X_text_train, maxlen=100)
X_text_test = pad_sequences(X_text_test, maxlen=100)


X_sentiment_train = X_train['Sentiment'].values.reshape(-1, 1)
X_sentiment_test = X_test['Sentiment'].values.reshape(-1, 1)


X_temporal_train = X_train[['Hour_sin', 'Hour_cos']].values
X_temporal_test = X_test[['Hour_sin', 'Hour_cos']].values

num_classes = len(np.unique(y))


text_input = Input(shape=(X_text_train.shape[1],))
embedded_text = Embedding(input_dim=len(tokenizer.word_index) + 1, output_dim=64, input_length=X_text_train.shape[1])(text_input)
text_lstm = LSTM(64)(embedded_text)

sentiment_input = Input(shape=(1,))
sentiment_dense = Dense(32, activation='relu')(sentiment_input)

temporal_input = Input(shape=(2,))
temporal_dense = Dense(32, activation='relu')(temporal_input)

concatenated = Concatenate()([text_lstm, sentiment_dense, temporal_dense])
output = Dense(num_classes, activation='softmax')(concatenated)

model = Model(inputs=[text_input, sentiment_input, temporal_input], outputs=output)


model.compile(optimizer=Adam(), loss='sparse_categorical_crossentropy', metrics=['accuracy'])


model.fit([X_text_train, X_sentiment_train, X_temporal_train], y_train, epochs=20, batch_size=32, validation_split=0.2)
```

```
Epoch 1/20
15/15 [==============================] - 7s 102ms/step - loss: 20.5230 - accuracy: 0.0171 - val_loss: 18.0568 - val_accuracy: 0.0684
Epoch 2/20
15/15 [==============================] - 1s 58ms/step - loss: 12.0976 - accuracy: 0.0577 - val_loss: 15.3358 - val_accuracy: 0.0000e
Epoch 3/20
15/15 [==============================] - 1s 61ms/step - loss: 8.4582 - accuracy: 0.0321 - val_loss: 13.7273 - val_accuracy: 0.0684
Epoch 4/20
15/15 [==============================] - 1s 59ms/step - loss: 6.6876 - accuracy: 0.0983 - val_loss: 13.1608 - val_accuracy: 0.0769
Epoch 5/20
15/15 [==============================] - 1s 59ms/step - loss: 5.8536 - accuracy: 0.1090 - val_loss: 12.9577 - val_accuracy: 0.0000e+
Epoch 6/20
15/15 [==============================] - 1s 59ms/step - loss: 5.3489 - accuracy: 0.1239 - val_loss: 12.6251 - val_accuracy: 0.1197
Epoch 7/20
```

```
15/15 [==============================] - 1s 60ms/step - loss: 5.0130 - accuracy: 0.1667 - val_loss: 12.4767 - val_accuracy: 0.0598
Epoch 8/20
15/15 [==============================] - 1s 65ms/step - loss: 4.7635 - accuracy: 0.1581 - val_loss: 12.4682 - val_accuracy: 0.1368
Epoch 9/20
15/15 [==============================] - 1s 94ms/step - loss: 4.6151 - accuracy: 0.1368 - val_loss: 12.5037 - val_accuracy: 0.1368
Epoch 10/20
15/15 [==============================] - 1s 82ms/step - loss: 4.4944 - accuracy: 0.1026 - val_loss: 12.4764 - val_accuracy: 0.0598
Epoch 11/20
15/15 [==============================] - 1s 59ms/step - loss: 4.3277 - accuracy: 0.1282 - val_loss: 12.5016 - val_accuracy: 0.0940
Epoch 12/20
15/15 [==============================] - 1s 60ms/step - loss: 4.2473 - accuracy: 0.1517 - val_loss: 12.4484 - val_accuracy: 0.1538
Epoch 13/20
15/15 [==============================] - 1s 61ms/step - loss: 4.1014 - accuracy: 0.1923 - val_loss: 12.4368 - val_accuracy: 0.1538
Epoch 14/20
15/15 [==============================] - 1s 60ms/step - loss: 3.9948 - accuracy: 0.1987 - val_loss: 12.3464 - val_accuracy: 0.1453
Epoch 15/20
15/15 [==============================] - 1s 65ms/step - loss: 3.8466 - accuracy: 0.1880 - val_loss: 12.3274 - val_accuracy: 0.1453
Epoch 16/20
15/15 [==============================] - 1s 59ms/step - loss: 3.7847 - accuracy: 0.2051 - val_loss: 12.3476 - val_accuracy: 0.0513
Epoch 17/20
15/15 [==============================] - 1s 59ms/step - loss: 3.7212 - accuracy: 0.1624 - val_loss: 12.3070 - val_accuracy: 0.1453
Epoch 18/20
15/15 [==============================] - 1s 60ms/step - loss: 3.6163 - accuracy: 0.2073 - val_loss: 12.3056 - val_accuracy: 0.1453
Epoch 19/20
15/15 [==============================] - 1s 60ms/step - loss: 3.5043 - accuracy: 0.2222 - val_loss: 12.3884 - val_accuracy: 0.0855
Epoch 20/20
15/15 [==============================] - 1s 61ms/step - loss: 3.4189 - accuracy: 0.2350 - val_loss: 12.3696 - val_accuracy: 0.1197
<keras.src.callbacks.History at 0x7cbf20edb070>
```

```
y_pred = model.predict([X_text_test, X_sentiment_test, X_temporal_test])
predicted_labels = np.argmax(y_pred, axis=1)


output_df = pd.DataFrame({'Text': X_test['Text'], 'Predicted_Mood': predicted_labels})


output_df.to_csv('sulakpredicted_labels.csv', index=False)
```

```
5/5 [==============================] - 0s 17ms/step
```

```
y_pred = model.predict([X_text_test, X_sentiment_test, X_temporal_test])
predicted_labels = np.argmax(y_pred, axis=1)


output_df = pd.DataFrame({'Text': X_test['Text'], 'Predicted_Mood': predicted_labels})


output_df.to_csv('sulpredicted_labels.csv', index=False)
```

```
5/5 [==============================] - 1s 14ms/step
```

```
pred1 = pd.read_csv('/content/sulpredicted_labels.csv')
```

	Text	Predicted_Mood
0	Exploring the world of digital art. It's never...	110
1	Feeling inspired after attending a workshop. ...	214
2	Eyes wide open in the night, fearful shadows ...	152
3	A soul weathered by the storm of heartbreak, s...	172
4	Attended a wine tasting event, savoring the ri...	172

```
moodtest_data = pd.read_csv("/content/moodtest.csv")
```

```
actual_values = moodtest_data['Sentiment']
predicted_values = pred['Predicted_labels']
```

```
report = classification_report(actual_values, predicted_values)

print("Classification Report:")
print(report)
```

```
Classification Report:
              precision    recall  f1-score   support
```

```
   Excitement       0.00      0.00      0.00         0
          Bad       1.00      1.00      1.00         6
  Contentment       1.00      1.00      1.00         8
    Curiosity       1.00      1.00      1.00         4
  Embarrassed       1.00      1.00      1.00         8
    Excitement      0.00      0.00      0.00         0
   Excitement       0.95      1.00      0.98        20
    Gratitude       1.00      1.00      1.00         5
        Happy       0.92      0.79      0.85        14
         Hate       1.00      1.00      1.00         6
          Joy       0.94      0.97      0.95        30
         Joy        0.00      0.00      0.00         0
  Mischievous       1.00      1.00      1.00         2
      Neutral       1.00      0.79      0.88        14
     Positive       1.00      1.00      1.00         1
       Relief       1.00      1.00      1.00         1
          Sad       1.00      1.00      1.00         9

     accuracy                           0.95       128
    macro avg       0.81      0.80      0.80       128
 weighted avg       0.97      0.95      0.95       128

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill
  _warn_prf(average, modifier, msg_start, len(result))
```