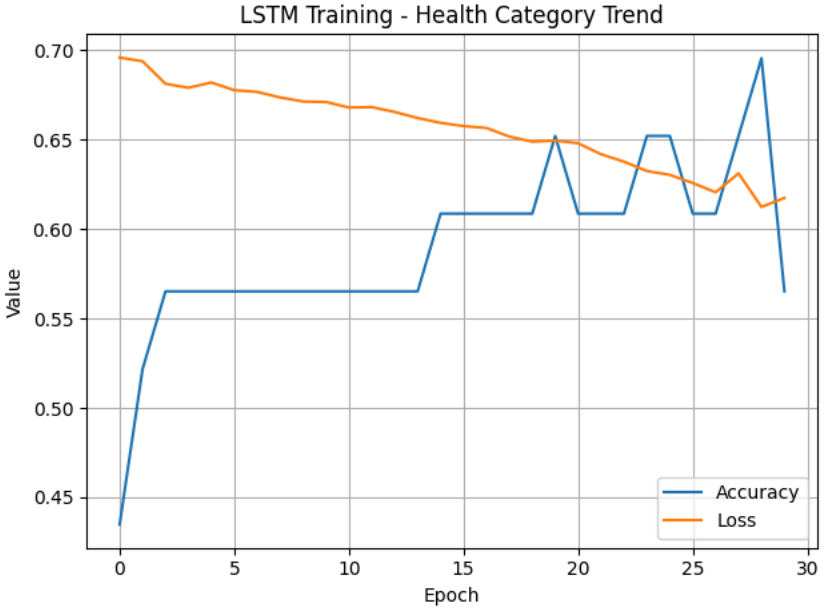
```
!pip install tensorflow matplotlib numpy

import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense

# Simulate 30 days of food category logs
np.random.seed(42)
days = 30
X_data = []

for _ in range(days):
    healthy = np.random.randint(1, 4)
    unhealthy = np.random.randint(0, 4)
    once = np.random.randint(0, 3)
    X_data.append([healthy, unhealthy, once])

X_data = np.array(X_data)

# Labels: 1 if unhealthy > healthy that day
y_data = np.array([1 if row[1] > row[0] else 0 for row in X_data])

# Create 7-day sequences
time_steps = 7
X_seq, y_seq = [], []

for i in range(len(X_data) - time_steps):
    X_seq.append(X_data[i:i+time_steps])
    y_seq.append(y_data[i+time_steps])

X_seq = np.array(X_seq)
y_seq = np.array(y_seq)

# Build the LSTM
model = Sequential([
    LSTM(32, input_shape=(time_steps, 3)),
    Dense(1, activation='sigmoid')
])
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train
history = model.fit(X_seq, y_seq, epochs=30, batch_size=4, verbose=0)

# Plot
plt.plot(history.history['accuracy'], label='Accuracy')
plt.plot(history.history['loss'], label='Loss')
plt.title("LSTM Training - Health Category Trend")
plt.xlabel("Epoch")
plt.ylabel("Value")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.11/dist-packages (2.18.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (2.0.2)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (25.2.1
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/lib/python3.11/dist-packages (from tens
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from tensorflow) (24.2)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0dev,>=3.20.3 in /usr/
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from tensorflow) (75.2.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.0.1)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (4.
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (1.71.0)
Requirement already satisfied: tensorboard<2.19,>=2.18 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (2.1
Requirement already satisfied: keras>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.8.0)
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (3.13.0)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from tensorflow) (0.4
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.11/dist-packages (from ten
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (2.8.2)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0->te
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages (from keras>=3.5.0->tensorflow) (0.14.1
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->tensor
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0->
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18-
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from te
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18-
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1->tenso
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5.0
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich->keras>=3.5
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich->
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:200: UserWarning: Do not pass an `input_shape`/`inpu
  super().__init__(**kwargs)
```

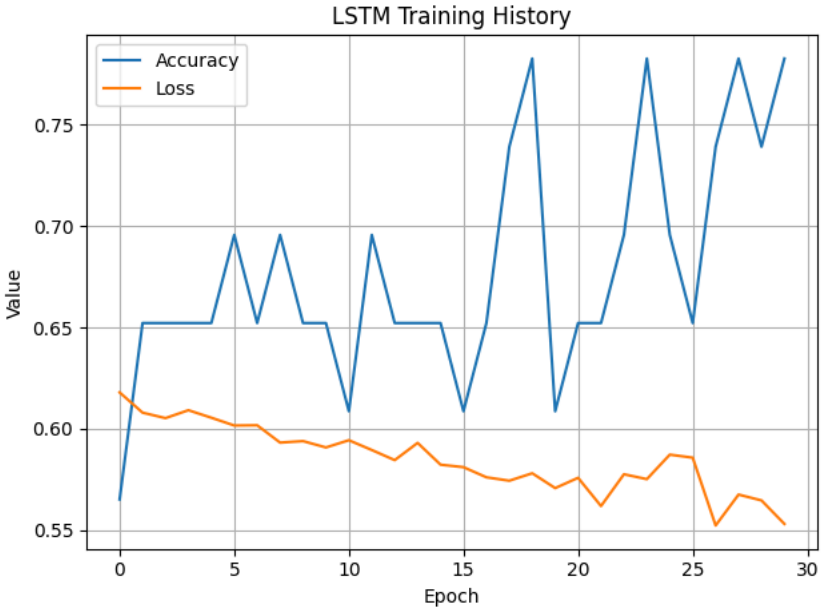



```
model.save("habit_lstm.keras")
```

```
import numpy as np
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
# Train the model and capture the training history
history_obj = model.fit(X_seq, y_seq, epochs=30, batch_size=4, verbose=0)

# Use the returned history object
history = history_obj.history
```

```
history = history_obj.history

# Prediction
y_pred_probs = model.predict(X_seq)
y_pred = (y_pred_probs > 0.5).astype("int32").flatten()

# Classification report
print("=== Classification Report ===")
print(classification_report(y_seq, y_pred, target_names=["Balanced", "Unhealthy Trend"]))

print("\n=== Confusion Matrix ===")
print(confusion_matrix(y_seq, y_pred))

# Plot
plt.plot(history['accuracy'], label='Accuracy')
plt.plot(history['loss'], label='Loss')
plt.title("LSTM Training History")
plt.xlabel("Epoch")
plt.ylabel("Value")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
=== Classification Report ===
                precision    recall  f1-score   support

      Balanced       0.79      0.85      0.81        13
Unhealthy Trend       0.78      0.70      0.74        10

      accuracy                           0.78        23
     macro avg       0.78      0.77      0.78        23
  weighted avg       0.78      0.78      0.78        23


=== Confusion Matrix ===
[[11  2]
 [ 3  7]]
```



Phase 2

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout

#Generate synthetic CNN label data (0=Healthy, 1=Occasional, 2=Unhealthy)
np.random.seed(42)
```

```
total_days = 100
cnn_labels = np.random.choice([0, 1, 2], size=total_days, p=[0.4, 0.3, 0.3])  # Simulated CNN outputs

#Build 7-day sequences
sequence_length = 7
X, y = [], []

def label_trend(seq):
    unhealthy = sum(1 for i in seq if i == 2)
    occasional = sum(1 for i in seq if i == 1)
    # Strict rule: 3+ 🍔 or 5+ (🍔 + 🟡) = ⚠️
    return 1 if (unhealthy >= 3 or (unhealthy + occasional) >= 5) else 0

for i in range(len(cnn_labels) - sequence_length):
    seq = cnn_labels[i:i + sequence_length]
    label = label_trend(seq)
    one_hot_seq = tf.keras.utils.to_categorical(seq, num_classes=3)
    X.append(one_hot_seq)
    y.append(label)

X = np.array(X)
y = np.array(y)

print(f" Created {len(X)} sequences. Shape: {X.shape}")
```

```
 Created 93 sequences. Shape: (93, 7, 3)
```

train

```
#Build the improved LSTM model
model = Sequential([
    LSTM(64, return_sequences=True, input_shape=(sequence_length, 3)),
    LSTM(32),
    Dropout(0.3),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')  # Binary output
])

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

#Train
history = model.fit(X, y, epochs=40, batch_size=8, validation_split=0.2, verbose=0)

#Evaluation
y_pred_probs = model.predict(X)
y_pred = (y_pred_probs > 0.5).astype(int).flatten()


# Save the model
model.save("habit_lstm_strict.keras")
print("Phase 2 LSTM model saved as 'habit_lstm_strict.keras'")
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/rnn/rnn.py:200: UserWarning: Do not pass an `input_shape`/`inpu
  super().__init__(**kwargs)
3/3 ━━━━━━━━━━━━━━━━━━━━ 1s 261ms/step
Phase 2 LSTM model saved as 'habit_lstm_strict.keras'
```

```
!cp habit_lstm_strict.keras /content/drive/MyDrive/
print("Model copied to Drive: MyDrive/habit_lstm_strict.keras")
```

```
Model copied to Drive: MyDrive/habit_lstm_strict.keras
```

```
print("\n=== Classification Report ===")
print(classification_report(y, y_pred, target_names=["Balanced", "Unhealthy Trend"]))
print("\n=== Confusion Matrix ===")
print(confusion_matrix(y, y_pred))

# 6. Plot
plt.plot(history.history['accuracy'], label='Accuracy')
plt.plot(history.history['val_accuracy'], label='Val Accuracy')
plt.plot(history.history['loss'], label='Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title("Phase 2 LSTM Training History")
plt.xlabel("Epoch")
plt.ylabel("Value")
plt.legend()
```

```
plt.grid(True)
plt.tight_layout()
plt.show()
```

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

       Balanced       1.00      0.98      0.99        53
Unhealthy Trend       0.98      1.00      0.99        40

       accuracy                           0.99        93
      macro avg       0.99      0.99      0.99        93
   weighted avg       0.99      0.99      0.99        93


=== Confusion Matrix ===
[[52  1]
 [ 0 40]]
```