**Implement RNN for sentiment analysis on movie reviews.**

import numpy as np

import tensorflow as tf

from tensorflow.keras.datasets import imdb

from tensorflow.keras.preprocessing.sequence import pad\_sequences

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Embedding, SimpleRNN, Dense

import matplotlib.pyplot as plt

# Step 1: Load and Preprocess Data

vocab\_size = 10000 # Number of words to consider as features

max\_length = 200 # Maximum length of a review

(train\_data, train\_labels), (test\_data, test\_labels) = imdb.load\_data(num\_words=vocab\_size)

# Pad sequences to ensure equal length

train\_data = pad\_sequences(train\_data, maxlen=max\_length, padding='post', truncating='post')

test\_data = pad\_sequences(test\_data, maxlen=max\_length, padding='post', truncating='post')

# Step 2: Build RNN Model

model = Sequential([

Embedding(input\_dim=vocab\_size, output\_dim=32, input\_length=max\_length),

SimpleRNN(32, return\_sequences=False),

Dense(1, activation='sigmoid')

])

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

# Step 3: Train Model

history = model.fit(train\_data, train\_labels, epochs=5, batch\_size=64, validation\_data=(test\_data, test\_labels))

# Step 4: Evaluate Model

loss, accuracy = model.evaluate(test\_data, test\_labels)

print(f'Test Accuracy: {accuracy:.4f}')

# Step 5: Plot Accuracy and Loss

plt.figure(figsize=(12, 4))

plt.subplot(1, 2, 1)

plt.plot(history.history['accuracy'], label='Train Accuracy')

plt.plot(history.history['val\_accuracy'], label='Validation Accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

plt.subplot(1, 2, 2)

plt.plot(history.history['loss'], label='Train Loss')

plt.plot(history.history['val\_loss'], label='Validation Loss')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.show()