**ASSESMENT-6**

**PROBLEM STATEMENT-** Train a sentiment analysis model on IMDB dataset, use RNN layers with LSTM/GRU notes

**SOURCE CODE:**

* Open the folder in the Spyder IDE
* The Command to install

**“pip install tensorflow”**

**File Name: program6.py**

import tensorflow as tf

from tensorflow.keras import layers, models, datasets, preprocessing

# Load and preprocess IMDB movie review dataset

max\_features = 10000

maxlen = 200

embedding\_dim = 128

(train\_data, train\_labels), (test\_data, test\_labels) = datasets.imdb.load\_data(num\_words=max\_features)

train\_data = preprocessing.sequence.pad\_sequences(train\_data, maxlen=maxlen)

test\_data = preprocessing.sequence.pad\_sequences(test\_data, maxlen=maxlen)

# Define RNN model with LSTM layers

model = models.Sequential([

    layers.Embedding(max\_features, embedding\_dim, input\_length=maxlen),

    layers.LSTM(64, dropout=0.2, recurrent\_dropout=0.2),  # LSTM layer with dropout

    layers.Dense(1, activation='sigmoid')

])

# Define RNN model with GRU layers

# model = models.Sequential([

#     layers.Embedding(max\_features, embedding\_dim, input\_length=maxlen),

#     layers.GRU(64, dropout=0.2, recurrent\_dropout=0.2),  # GRU layer with dropout

#     layers.Dense(1, activation='sigmoid')

# ])

# Compile the model

model.compile(optimizer='adam',

              loss='binary\_crossentropy',

              metrics=['accuracy'])

# Train the model

model.fit(train\_data, train\_labels, epochs=5, batch\_size=32, validation\_split=0.2)

# Evaluate the model

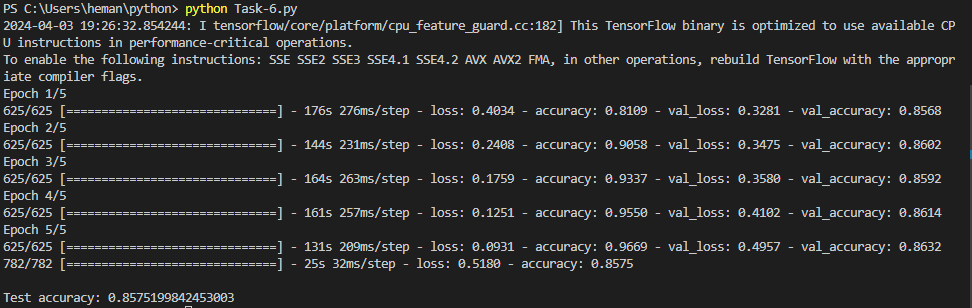
test\_loss, test\_acc = model.evaluate(test\_data, test\_labels)

print("\nTest accuracy:", test\_acc)

**Explanation:**

* We load and preprocess the IMDB movie review dataset, limiting the vocabulary size and padding sequences to a fixed length.
* We define an RNN model with either LSTM or GRU layers, followed by a dense layer with sigmoid activation for binary classification.
* We compile the model using the Adam optimizer and binary cross-entropy loss function.
* We train the model on the training data for 5 epochs with a batch size of 32 and a validation split of 20%.
* Finally, we evaluate the trained model on the test data and print the test accuracy.

**OUTPUT:**

****