**ASSESMENT-5**

**PROBLEM STATEMENT-** Applying the Deep Learning Models in the field of Natural Language Processing.

**SOURCE CODE:**

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

**“pip install tensorflow”**

**File Name: program5.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 deep learning model for sentiment analysis

model = models.Sequential([

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

    layers.GlobalAveragePooling1D(),

    layers.Dense(64, activation='relu'),

    layers.Dropout(0.5),

    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=10, batch\_size=32, validation\_split=0.2)

# Evaluate the model

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

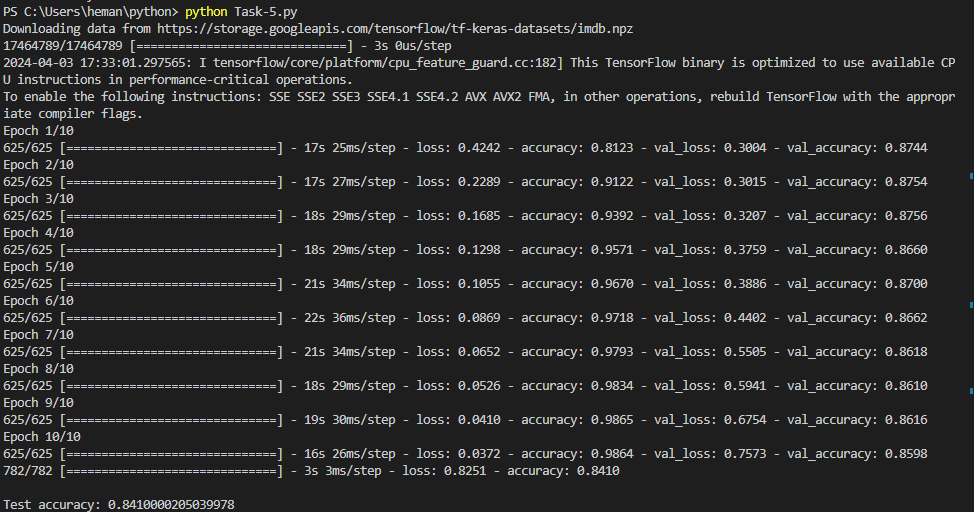
print("\nTest accuracy:", test\_acc)

**Explanation:**

Here's a breakdown of the code:

1. **Load and Preprocess Data:**
   * It loads the IMDB movie review dataset using TensorFlow's datasets module.
   * Preprocesses the reviews by padding sequences to a fixed length (maxlen) and limiting the vocabulary size (max\_features).
2. **Define Deep Learning Model:**
   * It defines a deep learning model for sentiment analysis using the Sequential API in Keras.
   * The architecture consists of an embedding layer, a global average pooling layer, a dense layer with ReLU activation, a dropout layer for regularization, and a dense layer with sigmoid activation for binary classification.
3. **Compile the Model:**
   * It compiles the model using the Adam optimizer and binary cross-entropy loss function.
4. **Train the Model:**
   * It trains the model on the training data for 10 epochs with a batch size of 32 and 20% validation split.
5. **Evaluate the Model:**
   * It evaluates the trained model on the test data and prints the test accuracy.

**OUTPUT:**

****