**Experiment: -8**

**AIM:** Implementing LSTM

Code:

import numpy as np

import tensorflow as tf

from tensorflow.keras.datasets import imdb

from tensorflow.keras.preprocessing import sequence

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout

# Set parameters

max\_features = 20000 # Number of words to consider as features

maxlen = 80 # Cut texts after this number of words (number of words in each review)

batch\_size = 32

# Load the IMDB dataset

(x\_train, y\_train), (x\_test, y\_test) = imdb.load\_data(num\_words=max\_features)

# Pad sequences to ensure they have the same length

x\_train = sequence.pad\_sequences(x\_train, maxlen=maxlen)

x\_test = sequence.pad\_sequences(x\_test, maxlen=maxlen)

# Build the LSTM model

model = Sequential()

model.add(Embedding(max\_features, 128, input\_length=maxlen))

model.add(LSTM(128, dropout=0.2, recurrent\_dropout=0.2))

model.add(Dense(1, activation='sigmoid'))

# Compile the model

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

# Train the model

model.fit(x\_train, y\_train, batch\_size=batch\_size, epochs=15, validation\_data=(x\_test, y\_test))

# Evaluate the model

score, acc = model.evaluate(x\_test, y\_test, batch\_size=batch\_size)

print(f'Test score: {score}')

print(f'Test accuracy: {acc}')