**CLASSIFICATION OF TEXT DATA USING RNN AND CNN**

**SOURCE CODE**

import numpy as np

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Embedding, LSTM, Conv1D, GlobalMaxPooling1D, Dense

from tensorflow.keras.preprocessing.text import Tokenizer

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

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

# Sample text data and labels (replace with your dataset)

texts = ['Text sample 1', 'Text sample 2', 'Text sample 3', ...]

labels = [0, 1, 0, ...] # Example labels (0 and 1)

# Tokenize and pad sequences

tokenizer = Tokenizer(num\_words=10000)

tokenizer.fit\_on\_texts(texts)

sequences = tokenizer.texts\_to\_sequences(texts)

max\_len = max(len(seq) for seq in sequences)

padded\_sequences = pad\_sequences(sequences, maxlen=max\_len)

# Split data into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(padded\_sequences, labels, test\_size=0.2, random\_state=42)

# RNN model

rnn\_model = Sequential([

Embedding(10000, 128, input\_length=max\_len),

LSTM(128),

Dense(1, activation='sigmoid')

])

# CNN model

cnn\_model = Sequential([

Embedding(10000, 128, input\_length=max\_len),

Conv1D(128, 5, activation='relu'),

GlobalMaxPooling1D(),

Dense(1, activation='sigmoid')

])

# Compile and train models

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

rnn\_model.fit(X\_train, y\_train, epochs=10, batch\_size=32, validation\_split=0.2)

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

cnn\_model.fit(X\_train, y\_train, epochs=10, batch\_size=32, validation\_split=0.2)

# Evaluate models

rnn\_scores = rnn\_model.evaluate(X\_test, y\_test)

cnn\_scores = cnn\_model.evaluate(X\_test, y\_test)

print("RNN Accuracy:", rnn\_scores[1])

print("CNN Accuracy:", cnn\_scores[1])