**Practical No:2b**

import numpy as np

import pandas as pd

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

from sklearn.preprocessing import LabelEncoder

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.optimizers import Adam

# Load the dataset

data = pd.read\_csv("C:/Users/Administrator/Documents/BE/DL Pratical/letter-recognition (1).csv")

# Check the column names

print("Column Names:", data.columns)

# Preprocess the data

X = data.drop(*columns*=['T'])  # Features

y = data['T']  # Target

# Encode target labels

label\_encoder = LabelEncoder()

y\_encoded = label\_encoder.fit\_transform(y)

# Split the data into training and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_encoded, *test\_size*=0.2, *random\_state*=42)

# Build the model

model = Sequential([

    Dense(128, *activation*='relu', *input\_shape*=(X\_train.shape[1],)),

    Dense(64, *activation*='relu'),

    Dense(len(label\_encoder.classes\_), *activation*='softmax')  # Output layer with number of classes

])

# Compile the model

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

# Train the model

model.fit(X\_train, y\_train, *epochs*=20, *batch\_size*=32, *validation\_split*=0.2)

# Evaluate the model

test\_loss, test\_accuracy = model.evaluate(X\_test, y\_test)

print("Test Accuracy:", test\_accuracy)

# Make predictions

predictions = model.predict(X\_test)

# Decode the predicted labels

predicted\_labels = label\_encoder.inverse\_transform(np.argmax(predictions, *axis*=1))

# Decode the actual labels

actual\_labels = label\_encoder.inverse\_transform(y\_test)

# Print some predicted and actual results

print("Some Predicted and Actual Results:")

for i in range(10):

    print("Predicted:", predicted\_labels[i], "Actual:", actual\_labels[i])