**ASSESMENT-7**

**PROBLEM STATEMENT-** Applying the Autoencoder algorithms for encoding the real-world data.

**SOURCE CODE:**

* Open the folder in the Spyder IDE
* Take any .csv(comma separated value) that should be real world data. Now in this program I have chosen the **“smartphones.csv”** .
* The Command to install

**“pip install tensorflow numpy”**

**File Name: program7.py**

import numpy as np

import tensorflow as tf

from tensorflow.keras import layers, models

# Load your real-world data (replace this with your data loading code)

def load\_data():

    real\_data = np.genfromtxt('smartphones.csv', delimiter=',', skip\_header=1)

    return real\_data

# Define autoencoder architecture

def build\_autoencoder(input\_dim, encoding\_dim):

    autoencoder = models.Sequential([

        layers.Dense(64, activation='relu', input\_shape=(input\_dim,)),

        layers.Dense(encoding\_dim, activation='relu'),

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

        layers.Dense(input\_dim, activation='sigmoid')

    ])

    autoencoder.compile(optimizer='adam', loss='mse')

    return autoencoder

# Train the autoencoder

def train\_autoencoder(autoencoder, real\_data, epochs=50, batch\_size=32):

    autoencoder.fit(real\_data, real\_data, epochs=epochs, batch\_size=batch\_size)

# Encode the real-world data

def encode\_data(autoencoder, real\_data):

    encoded\_data = autoencoder.predict(real\_data)

    return encoded\_data

# Main function

def main():

    # Load real-world data

    real\_data = load\_data()

    # Define autoencoder parameters

    input\_dim = real\_data.shape[1]  # Number of features

    encoding\_dim = 5  # Dimensionality of the latent space

    # Build and train autoencoder

    autoencoder = build\_autoencoder(input\_dim, encoding\_dim)

    train\_autoencoder(autoencoder, real\_data)

    # Encode the real-world data

    encoded\_data = encode\_data(autoencoder, real\_data)

    # Print the shape of the encoded data

    print("Shape of encoded data:", encoded\_data.shape)

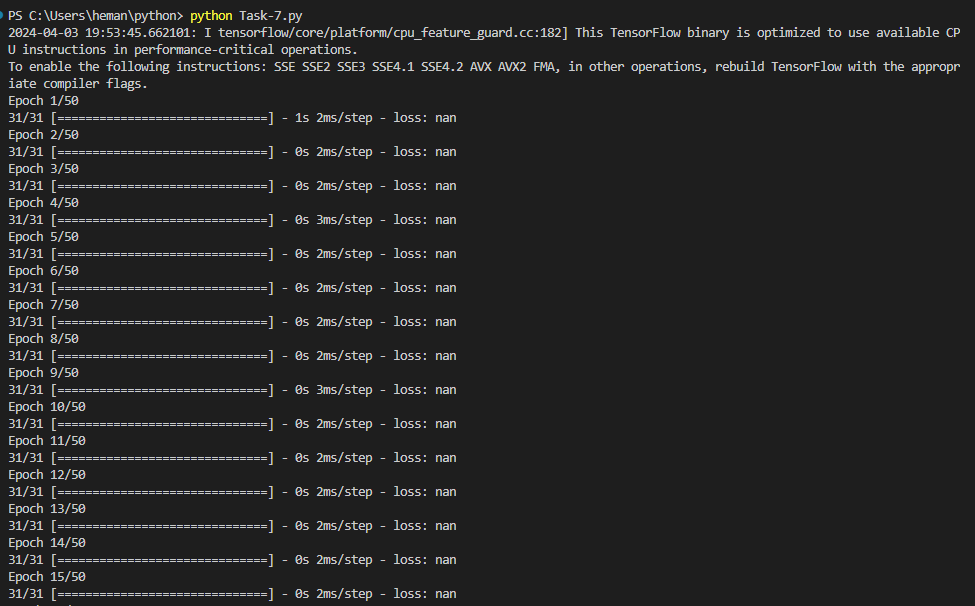
if \_\_name\_\_ == "\_\_main\_\_":

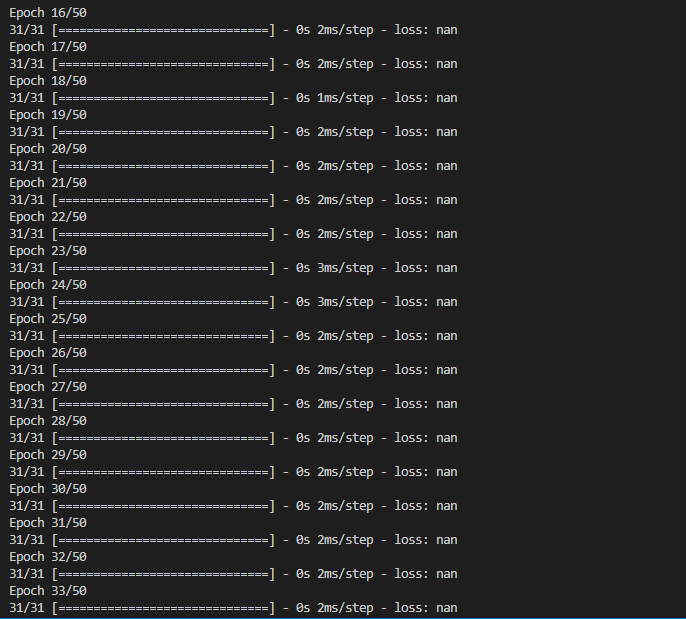
    main()

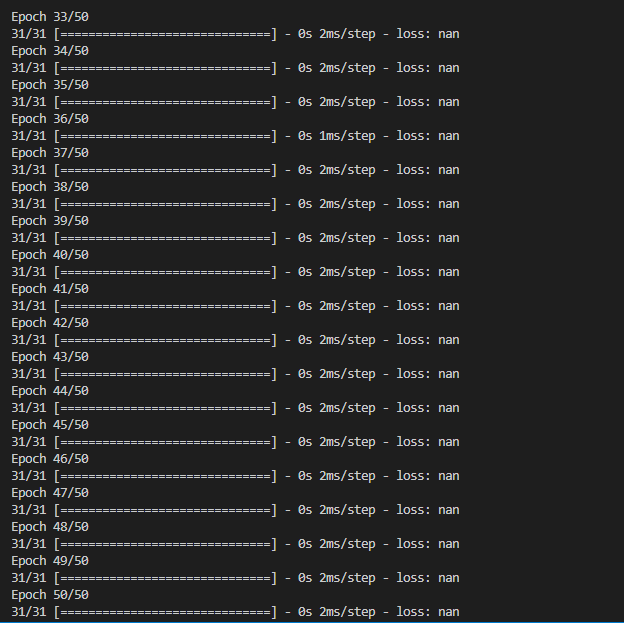
**Explanation:**

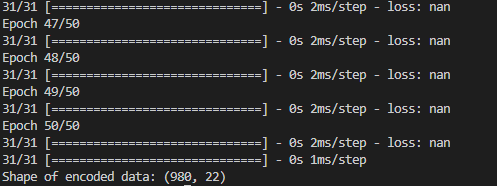
* The **load\_data()** function is responsible for loading your real-world data. You should replace this function with your actual data loading code.
* The **build\_autoencoder()** function defines the architecture of the autoencoder. It returns a compiled autoencoder model.
* The **train\_autoencoder()** function trains the autoencoder on the real-world data.
* The **encode\_data()** function uses the trained autoencoder to encode the real-world data into a lower-dimensional latent space.
* In the **main()** function, we load the real-world data, define the autoencosder parameters, build and train the autoencoder, and encode the real-world data.
* We print the shape of the encoded data to verify the encoding process.

**OUTPUT:**

****

****

****

****