Ex.No.7

BUILD AUTOENCODERS WITH KERAS/TENSORFLOW

AIM:

To build autoencoders with Keras/TensorFlow

PROCEDURE:

1. Download and load the dataset.
2. Perform analysis and preprocessing of the dataset.
3. Build autoencoders using Keras/TensorFlow.
4. Compile and fit the model.
5. Perform prediction with the test dataset.
6. Calculate performance metrics.

PROGRAM:

```
import numpy as np
from keras.layers import Input, Dense
from keras.models import Model

input_dim = 784

encoding_dim = 32

input_img = Input(shape=(input_dim,))

encoded = Dense(128, activation='relu')(input_img)
```

```
encoded = Dense(64, activation='relu')(encoded)
encoded = Dense(encoding_dim, activation='relu')(encoded)

decoded = Dense(64, activation='relu')(encoded)
decoded = Dense(128, activation='relu')(decoded)
decoded = Dense(input_dim, activation='sigmoid')(decoded)

autoencoder = Model(input_img, decoded)
encoder = Model(input_img, encoded)

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

x_train = np.random.random((1000, input_dim))
x_test = np.random.random((200, input_dim))

autoencoder.fit(x_train, x_train,
                epochs=50,
                batch_size=256,
                shuffle=True,
                validation_data=(x_test, x_test))

encoded_imgs = encoder.predict(x_test)
decoded_imgs = autoencoder.predict(x_test)

print("Shape of encoded images:", encoded_imgs.shape)
print("Shape of decoded images:", decoded_imgs.shape)
```

OUTPUT:

```
Epoch 48/50
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 24ms/step - loss: 0.6862 - val_loss: 0.6923
Epoch 49/50
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step - loss: 0.6860 - val_loss: 0.6923
Epoch 50/50
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - loss: 0.6858 - val_loss: 0.6923
```

```
Shape of encoded images: (200, 32)
Shape of decoded images: (200, 784)
```

RESULT:

Thus, an autoencoder using Keras/TensorFlow was successfully implemented.