GENERATIVE AI

ASSIGNMENT 1

Done by:
Siddhant Srinivas 2021A7PS0050H
Nikhil Dhanaraj 2021A7PS0427H
PS Varshith 2021A8PS1660H

PROBLEM STATEMENT:

Given an input image, project the image into a latent vector space of given dimensions and reconstruct the image from that space, minimizing the loss of information in the process.

DATASET:

The dataset from which the images are obtained is the ‘MNIST’ dataset, which consists of images of handwritten digits from 0 to 9. There are 70,000 images, 60,000 of which comprise the training set. The remaining 10,000 images comprise the testing set.

MODELS IMPLEMENTED:

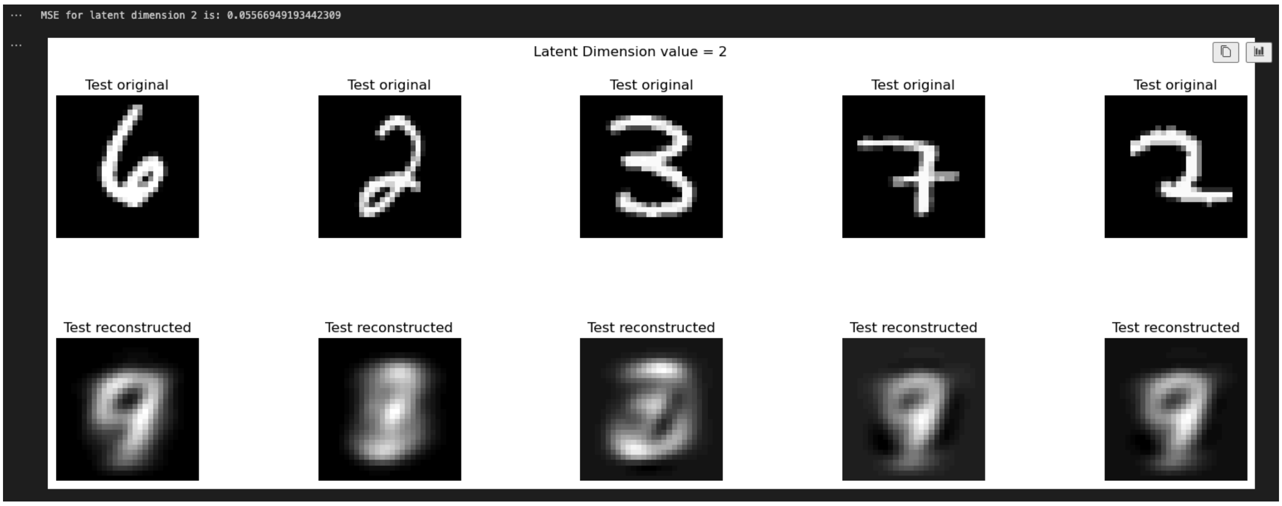
The image reconstruction task was performed using 3 models: PCA, Probabilistic PCA and Variational Autoencoders.

LIBRARIES USED:

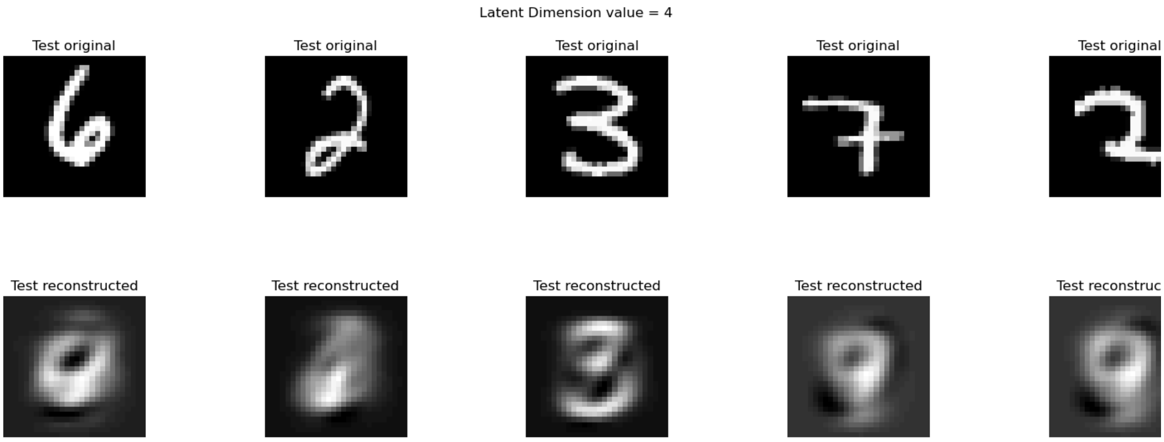
To load the MNIST dataset, ‘keras’ was used. Sklearn was used in the implementation of PCA. Keras was also used in VAEs for modules such as Input, Dense, Lambda, Model and Backend. Other libraries used include standard libraries such as numpy, matplotlib and random.

Part A: Principal Component Analysis

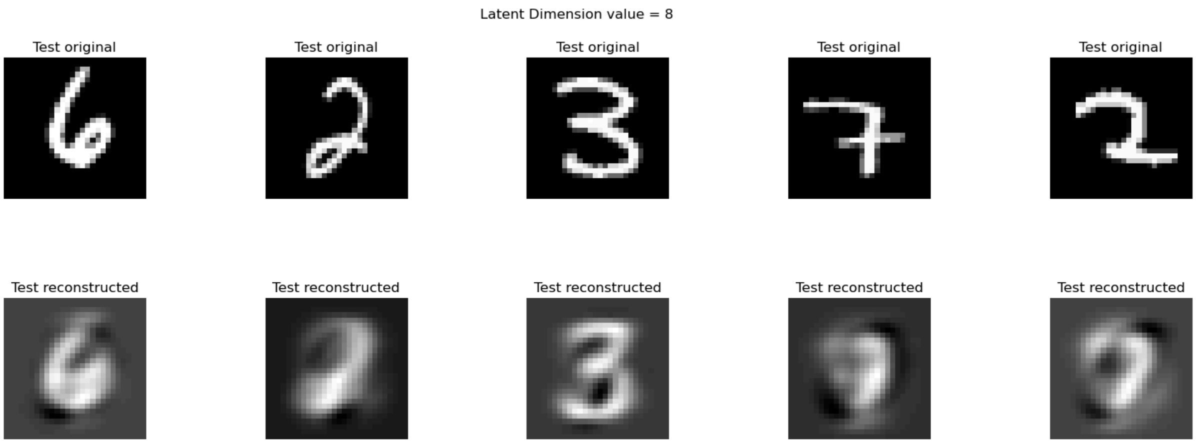
- Latent dimension 2:



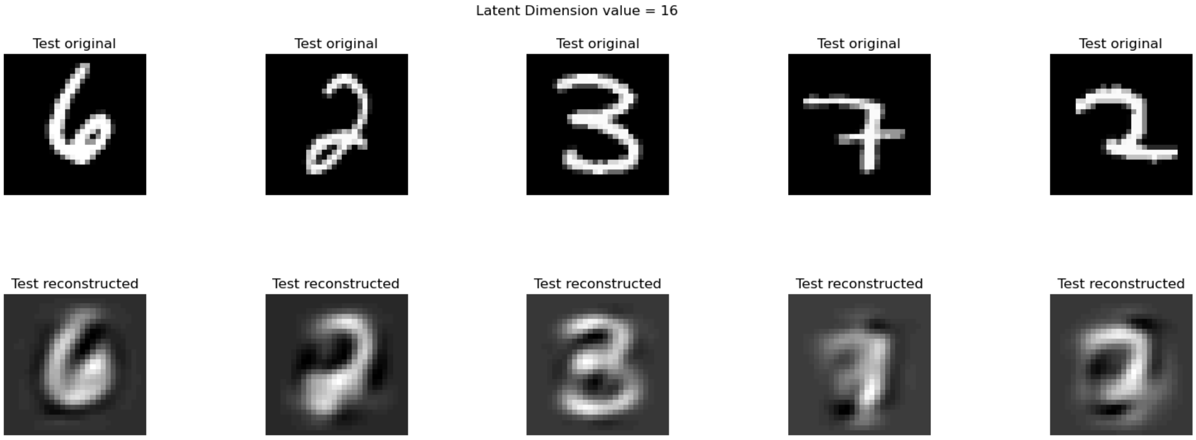
- Latent dimension 4:



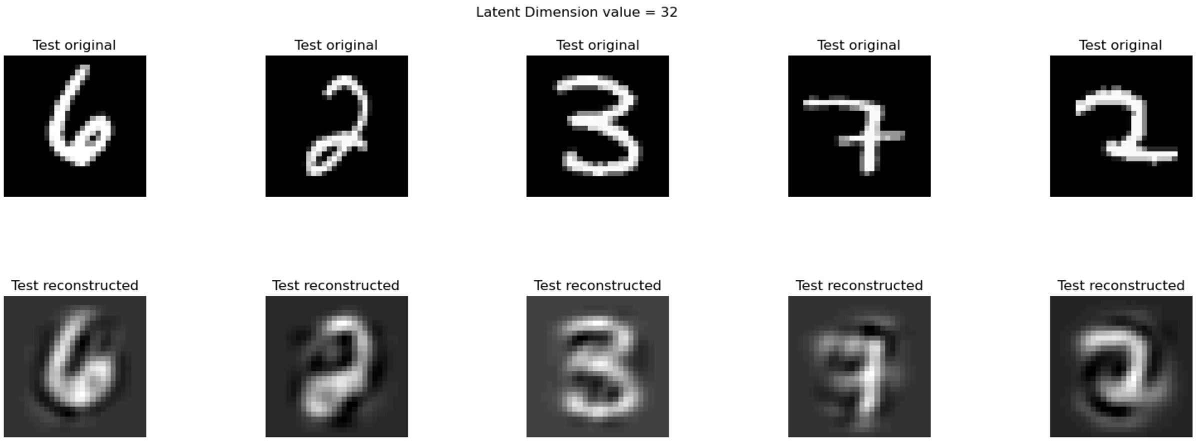
- Latent dimension 8:



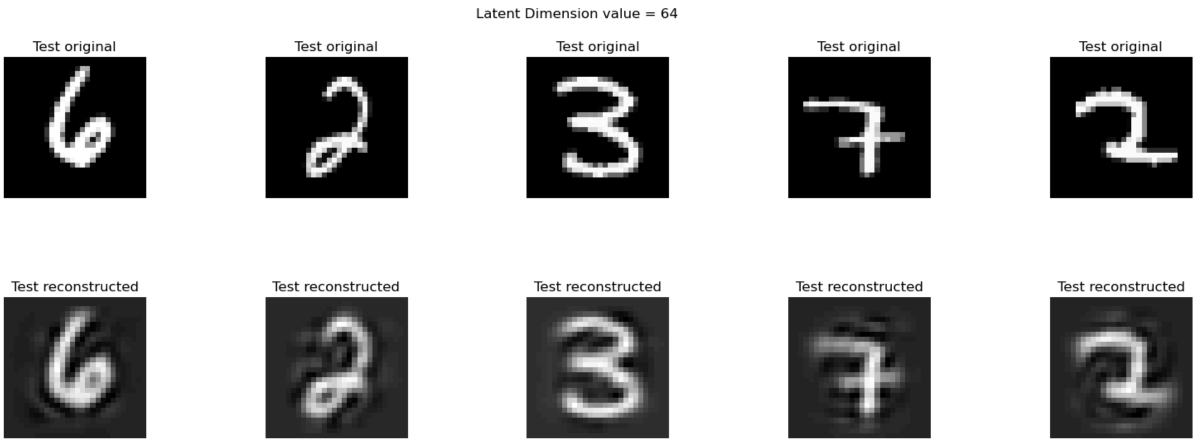
- Latent dimension 16:



- Latent dimension 32:



- Latent dimension 64:



Mean Squared Error:

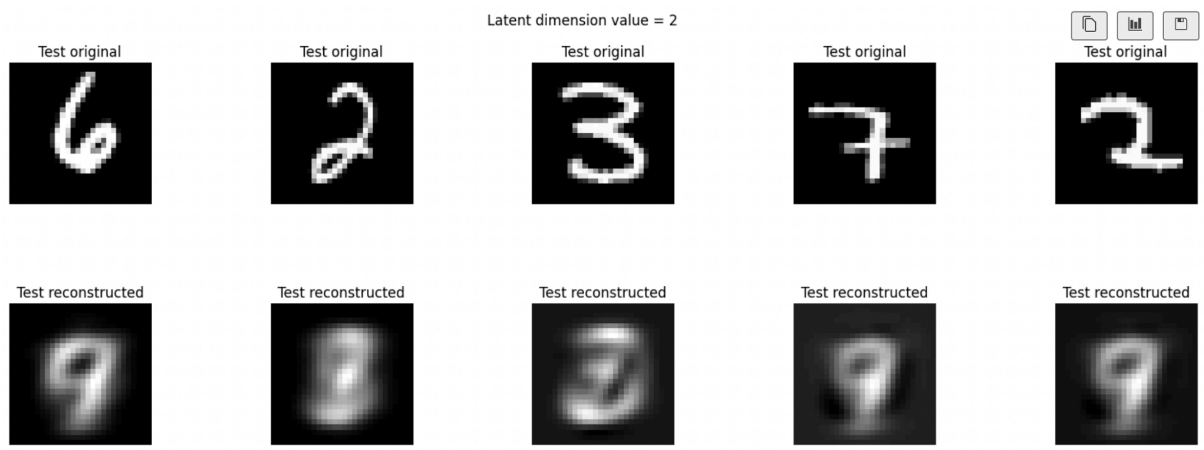
Latent variable	*Mean Squared Error*
2	0.05566949193442309
4	0.047903465065584565
8	0.03744093391903097
16	0.02686019078827122
32	0.016829987691859553
64	0.00904777044675469

Conclusions:

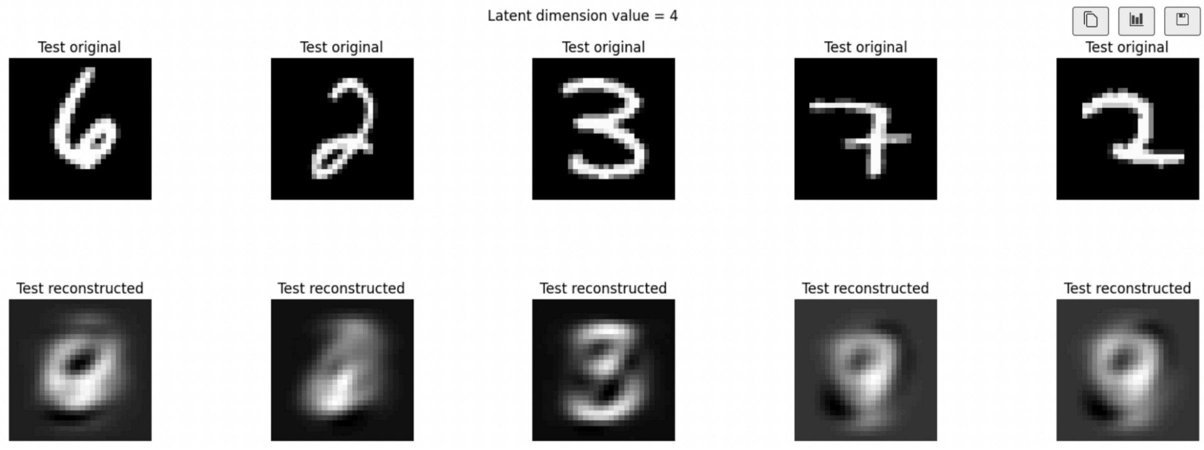
Clearly, MSE reduces when we increase the latent variable, with the highest error being for latent variable 2 and the lowest error being for the latent variable 64. This can be attributed to the fact that an increase in latent dimension leads to a greater number of principal components, and thus a capture of more information about patterns present in the input images. Additionally, you account for more variance in the data, leading to more accurate reconstruction.

Part B: Probabilistic Principal Component Analysis

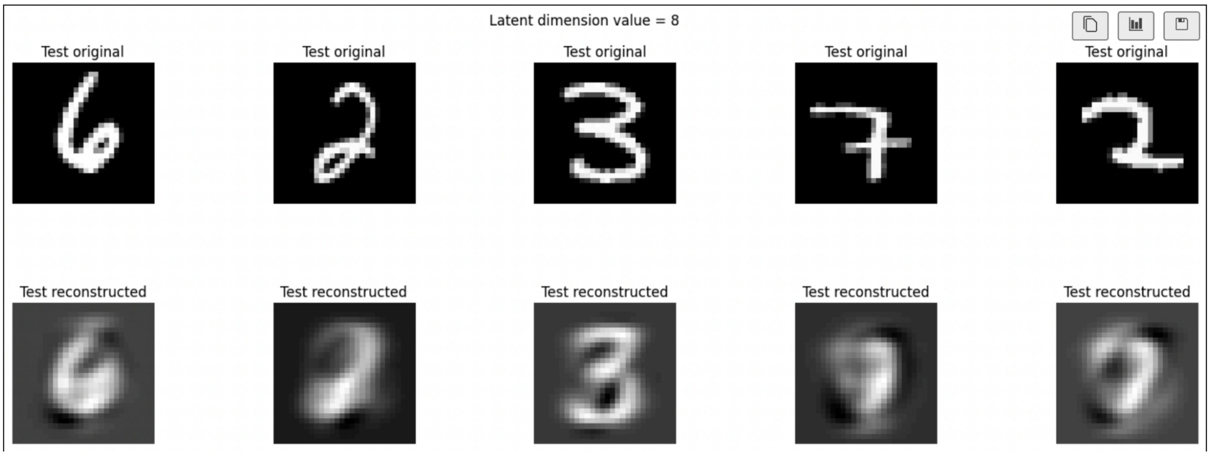
- Latent dimension 2:



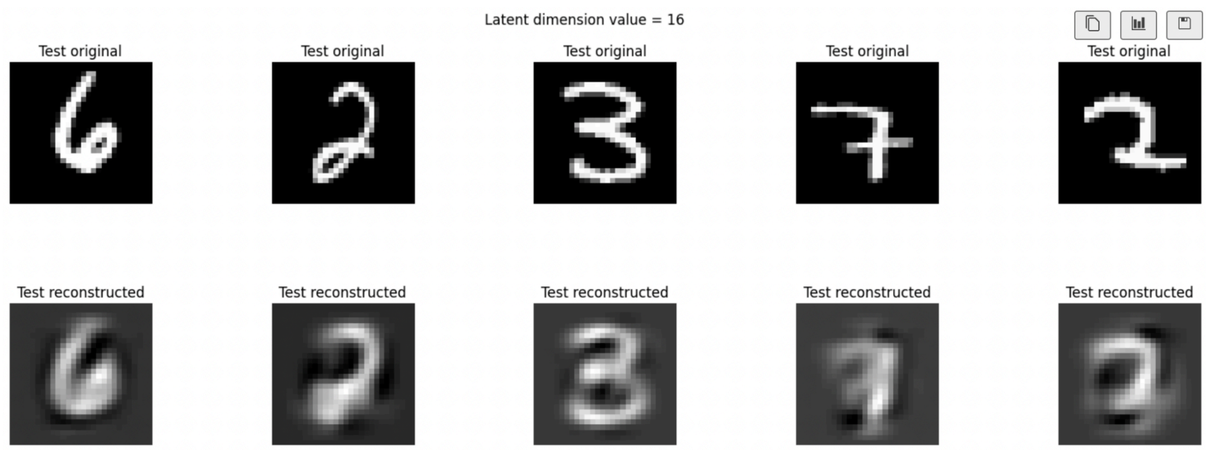
- Latent dimension 4:



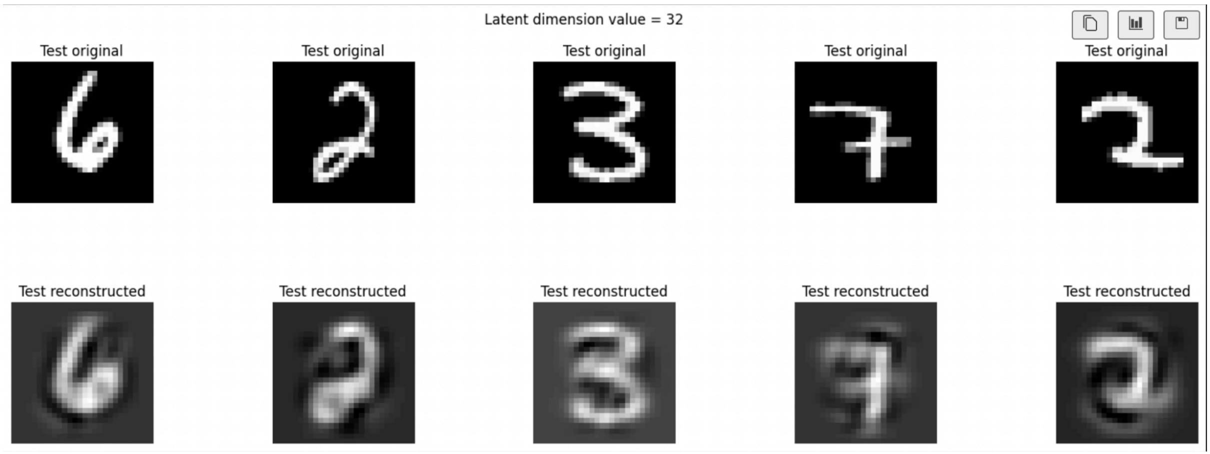
- Latent dimension 8:



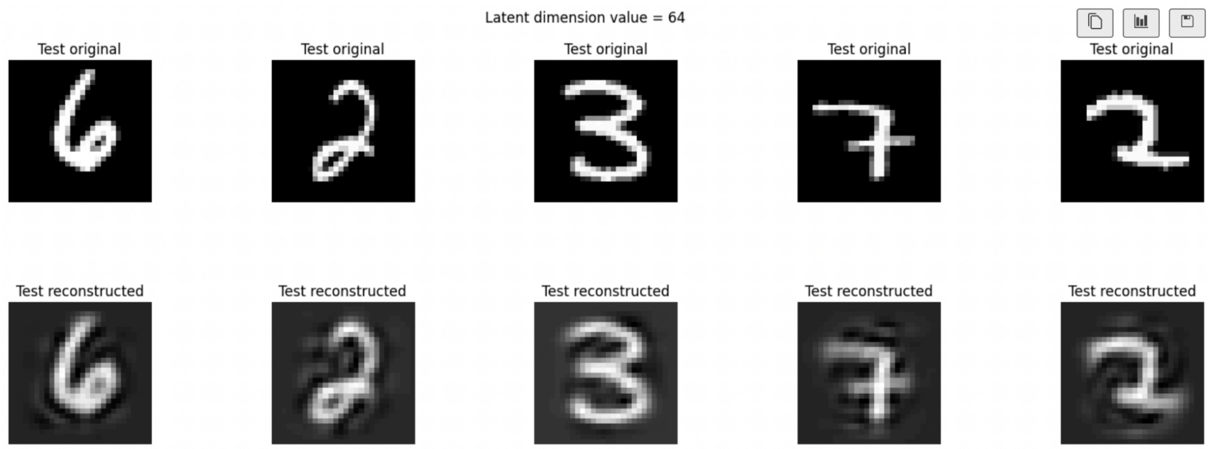
- Latent dimension 16:



- Latent dimension 32:



- Latent dimension 64:



Mean Squared Error:

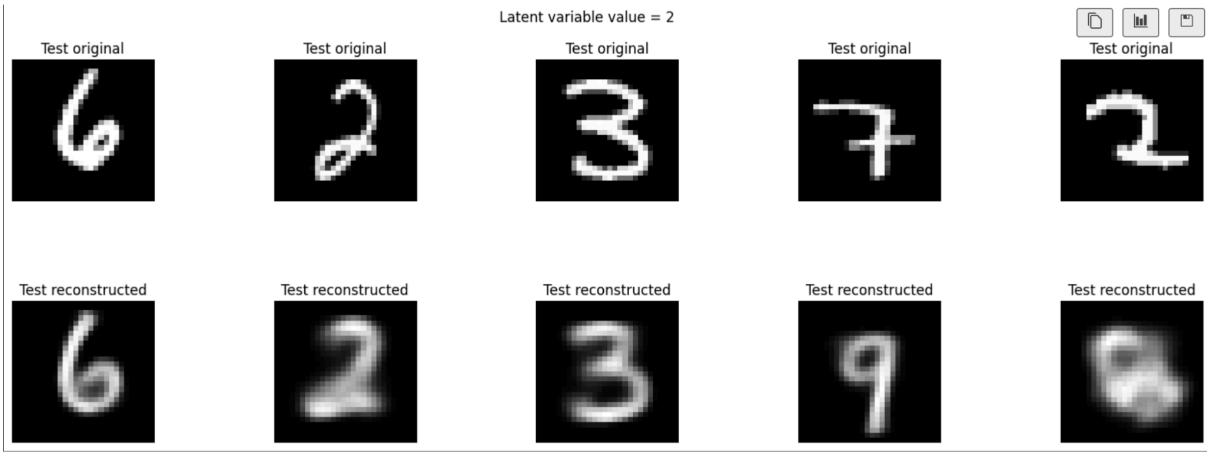
Latent Variable Dimension	*Mean Squared Error*
2	0.055616065817805635
4	0.04785867676369214
8	0.037419426603041844
16	0.02685230807925931
32	0.01682450735610923
64	0.009045238607576146

Conclusions:

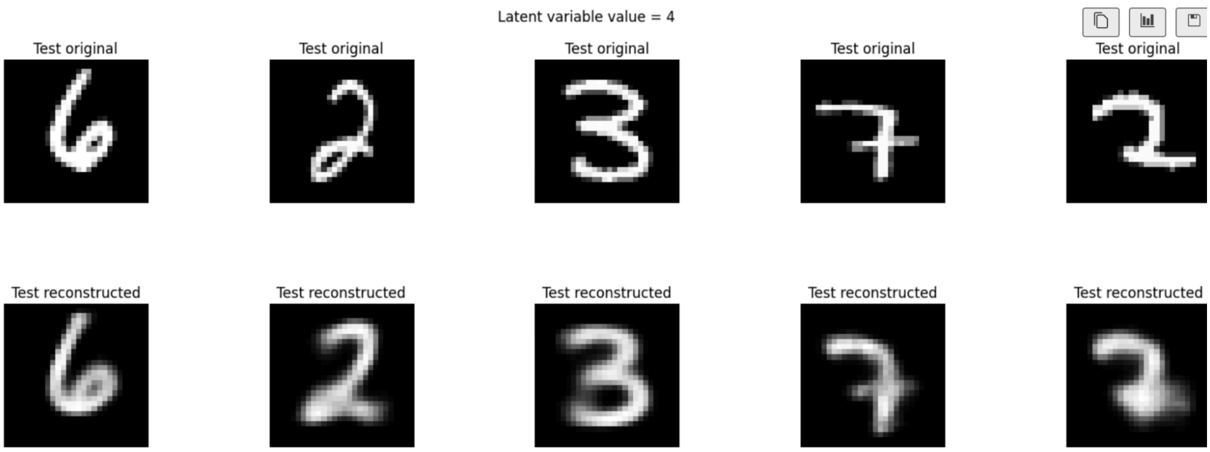
PPCA follows the same trend as PCA, of decreasing MSE with increasing latent dimension. You can notice that PPCA has a very similar MSE to PCA, with PPCA slightly outperforming PCA.

Part C: Variational Autoencoders

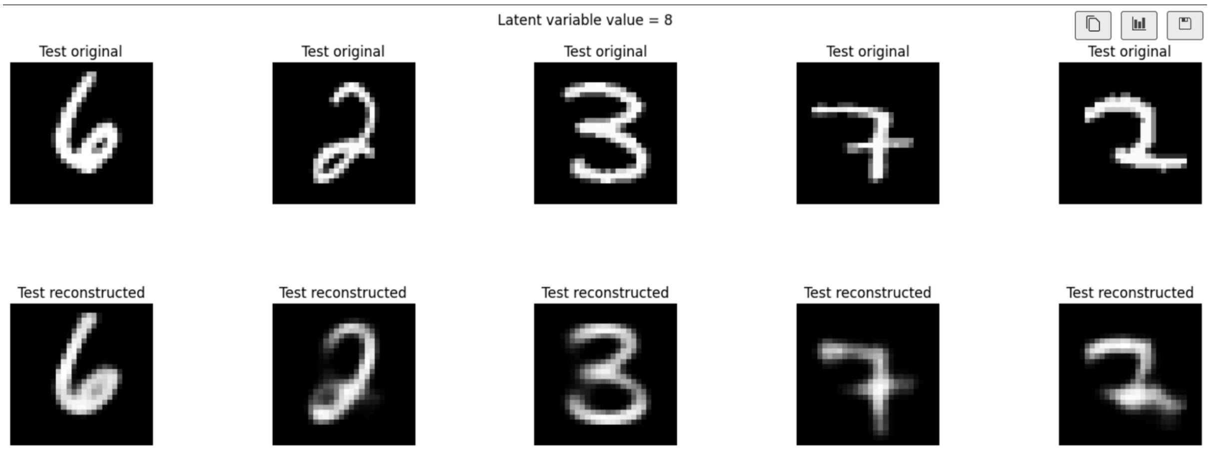
- Latent dimension 2:



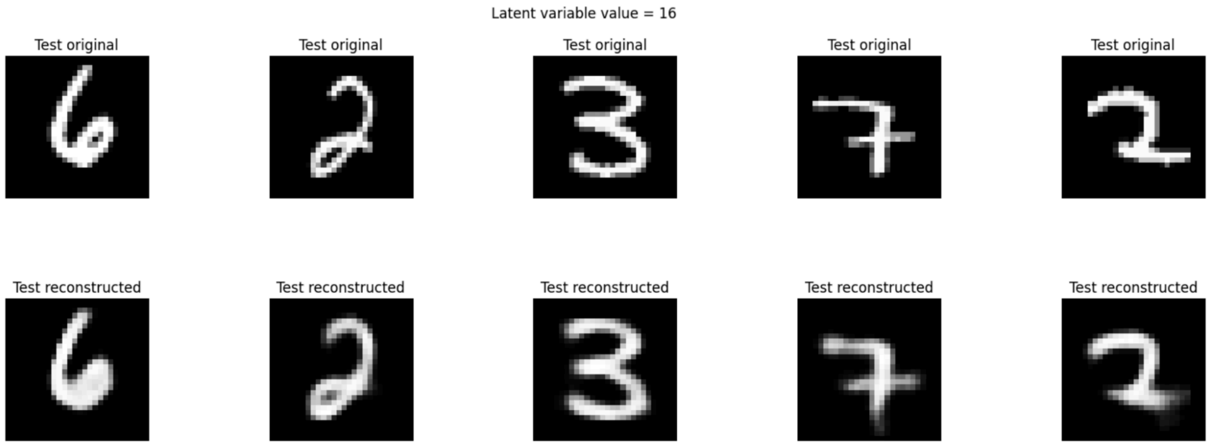
- Latent dimension 4:



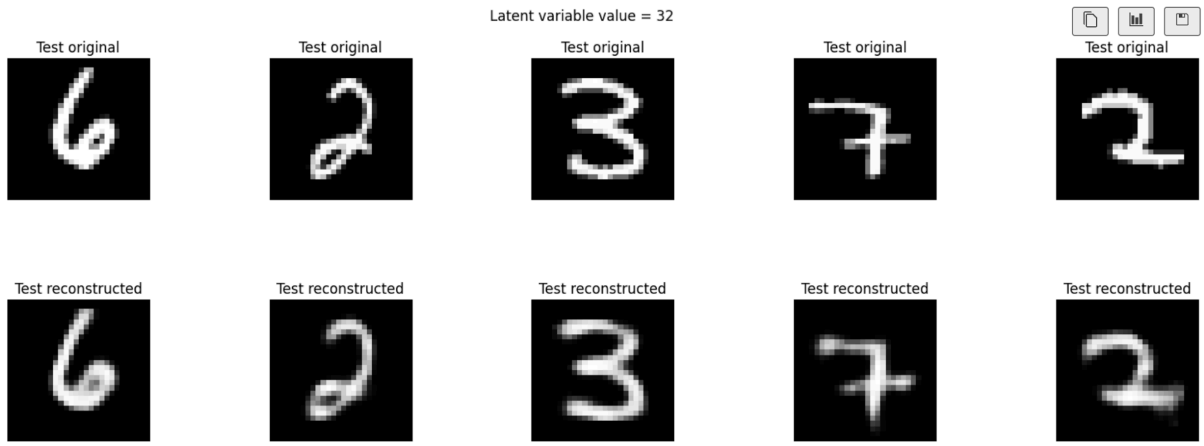
- Latent dimension 8:



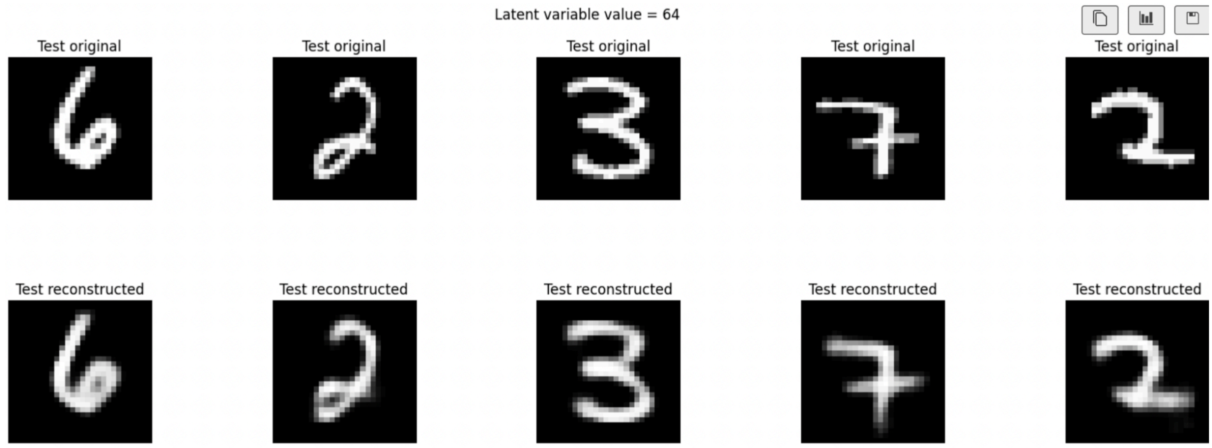
- Latent dimension 16:



- Latent dimension 32:



- Latent dimension 64:



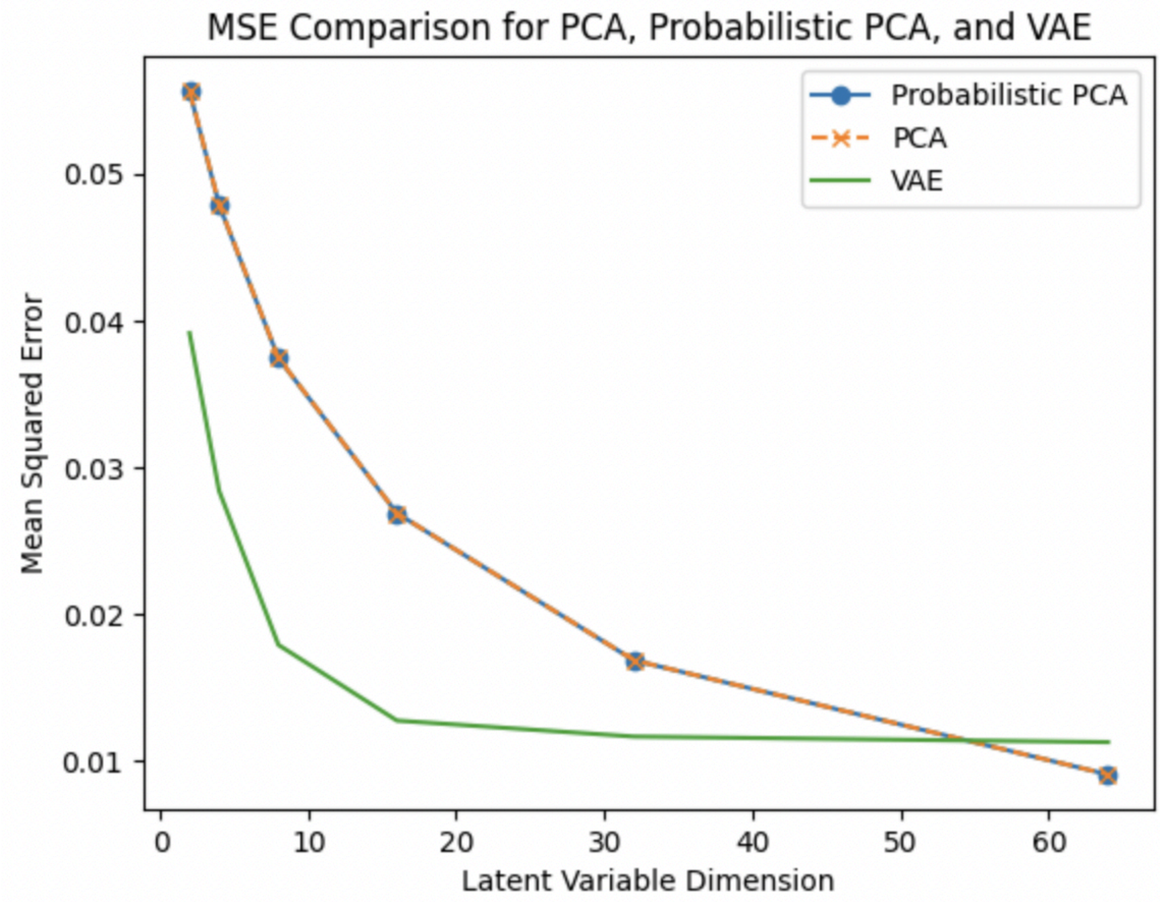
Mean Squared Error:

Latent Variable Dimension	***Mean Squared Error***
2	0.038879427531066546
4	0.028420273851421766
8	0.018193082938523827
16	0.013163022279376933
32	0.011442947036760668
64	0.011945405794136223

Conclusions:

The MSE decreases but then flattens at dimensions 16 with no real difference between dimensions 16,32 and 64. A possible reason for the slight increase in MSE from 32 to 64 is saturation. 32 might be very close to the optimal latent dimension value for the given model, which is why an increase in latent dimension doesn’t really improve performance.

Part D: Comparative study:


MSE Comparison for PCA, Probabilistic PCA, and VAE
Probabilistic PCA
PCA
VAE
Mean Squared Error
0.05
0.04
0.03
0.02
0.01
0
10
20
30
40
50
60
Latent Variable Dimension

x`