In [ ]:

```
# importing necessary packages
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import pandas as pd
```

In [ ]:

```
# Loading the dataset
df = pd.read_csv('noisy_mnist.csv',header=None)
df.head()
```

Out[33]:

	0	1	2	3	4	5	6	7	
0	0.264608	0.060024	0.146811	0.336136	0.280311	-0.139760	0.233830	0.371765	0.6690
1	-0.108767	-0.202736	-0.170056	0.332693	0.186255	0.280204	0.253478	0.363563	0.0396
2	-0.199832	-0.295294	-0.099008	0.026376	0.074813	0.157204	0.042648	0.261847	-0.0179
3	0.277441	-0.031975	-0.070889	0.047647	0.075778	0.556461	0.741170	0.604523	0.7705
4	0.256846	-0.118795	-0.155106	-0.134832	0.297439	0.241704	0.469472	0.815539	0.7633

5 rows × 256 columns

In [ ]:

```
df.shape
```

Out[34]:

```
(4055, 256)
```

In [ ]:

```
# Function to plot the images
def plot_mnist_images(X):
  '''
  X: MNIST image
  '''
  fig,ax = plt.subplots(10,10,figsize =(5,5))
  ax=ax.flatten()
  for img,a in zip(X,ax):
    a.imshow(img.reshape(16,16),cmap='Greys')
    a.axis('off')
```

In [ ]:

```
# dataframe to numpy array
X = np.array(df)
# plot_mnist_images(X)
X.shape
```

Out[43]:

```
(4055, 256)
```

In [ ]:

```
# train test split
X_train,X_test = train_test_split(X,train_size=0.6)
```

In [ ]:

```
# svd
U,S,Vt = np.linalg.svd(X_train)
```

In [ ]:

```
# Function to get the principal components
def get_principal_comps(X,Vt,n):
  '''
  X: data matrix
  Vt: right singular matrix
  '''
  PC = X@(Vt.T)
  return PC[:,:n]
```

In [ ]:

```
# Function to reconstruct the images
def reconstruct_data_mat(PC,Vt,n):
  '''
  PC: principal components
  Vt: Right singular matrix
  '''
  X_ = PC@(Vt[:n,:])
  return X_
```

In [ ]:

```
# reconstructing the images
n=256
PC_image = get_principal_comps(X_train,Vt,n)
recons_image = reconstruct_data_mat(PC_image,Vt,n)
```

In [ ]:

```
# PLotting the actual image and reconstructed image
plot_mnist_images(X_train)
plot_mnist_images(recons_image)
```

In [ ]: