Q3. Singular Value Decomposition (SVD): ¶

Perform Singular Value Decomposition on the matrix A obtained in Question 2. Separate and print matrices U, Σ, and $V\dot{\circ}$. Verify that A equals the product of U, Σ, and $V\dot{\circ}$. Additionally, find the rank 2 and rank 3 approximations of matrix A.

In [5]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [6]:
```
A = np.random.randint(0, 10, size=(5, 5))
print("Original matrix A:")
print(A)
```

```
Original matrix A:
[[1 7 4 5 5]
 [0 7 4 9 7]
 [0 0 8 4 8]
 [5 8 2 1 0]
 [5 4 3 6 9]]
```

In [7]:
```
#Q 3
# Perform Singular Value Decomposition (SVD)
U, Sigma, V_transpose = np.linalg.svd(A)
Sigma_mat = np.zeros_like(A, dtype=float)
Sigma_mat[:min(A.shape[0], A.shape[1]), :min(A.shape[0], A.shape[1])] = np.c
print("Matrix U:")
print(U)
print("\nMatrix Sigma:")
print(Sigma_mat)
print("\nMatrix V:")
print(V_transpose.T)
```

```
Matrix U:
[[-0.43668603  0.19289904  0.25467238 -0.23815702 -0.80654726]
 [-0.56380788  0.02863436  0.65637188  0.18776429  0.46391946]
 [-0.41201734 -0.63414725 -0.27164311 -0.57461949  0.15531079]
 [-0.24475937  0.74698697 -0.35326747 -0.39628656  0.31664235]
 [-0.5116254  -0.04286924 -0.55292861  0.6486873  -0.09938022]]

Matrix Sigma:
[[23.91825206  0.          0.          0.          0.        ]
 [ 0.          9.74520972  0.          0.          0.        ]
 [ 0.          0.          5.32873449  0.          0.        ]
 [ 0.          0.          0.          4.46507646  0.        ]
 [ 0.          0.          0.          0.          0.784721  ]]

Matrix V:
[[-0.17637618  0.38105775 -0.80249974  0.22930104  0.35651317]
 [-0.46023572  0.75474543  0.25136467 -0.20790066 -0.33486643]
 [-0.38976549 -0.28954515 -0.16782382 -0.81634393  0.26393981]
 [-0.5109187  -0.08461727  0.45476037  0.37993813  0.61696193]
 [-0.58661697 -0.44063346 -0.24049561  0.30565987 -0.55718632]]
```

In [8]:
```
# Now we show that A equals the product of U, Sigma, and V
A_reconstructed = np.dot(U, np.dot(Sigma_mat, V_transpose))
print("\nVerification that A equals the product of U, Sigma, and V:")
print(np.allclose(A, A_reconstructed))
```

```
Verification that A equals the product of U, Sigma, and V:
True
```

In [ ]: