

# Implementing Matrix Factorization using SVD with NumPy

Implement matrix factorization using singular value decomposition (SVD) with numpy.

```
In [12]: import numpy as np
A = np.array([[1,2,3],[4,5,6],[7,8,9]])
U,s,Vt = np.linalg.svd(A)
Sigma = np.diag(s)
A_reconstructed = U @ Sigma @Vt

print("Original Matrix:")
print(A)
print("Reconstructed Matrix:")
print(A_reconstructed)
```

Original Matrix:

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

Reconstructed Matrix:

```
[[1. 2. 3.]
 [4. 5. 6.]
 [7. 8. 9.]]
```

## Matrix Factorization using SVD

```
In [14]: import numpy as np
A = np.array([[1,2,3],[4,5,6],[7,8,9]])
U,s,Vt = np.linalg.svd(A)
k = 2
U_k = U[:, :k]
s_k = s[:k]
Vt_k = Vt[k, :]

Sigma_k = np.diag(s_k)
A_Factorized = U_k @ Sigma_k @ Vt_k

print("Original Matrix:")
print(A)
print("Factorized Matrix:")
print(A_Factorized)
```

Original Matrix:

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
```

Factorized Matrix:

```
[[1. 2. 3.]
 [4. 5. 6.]
 [7. 8. 9.]]
```