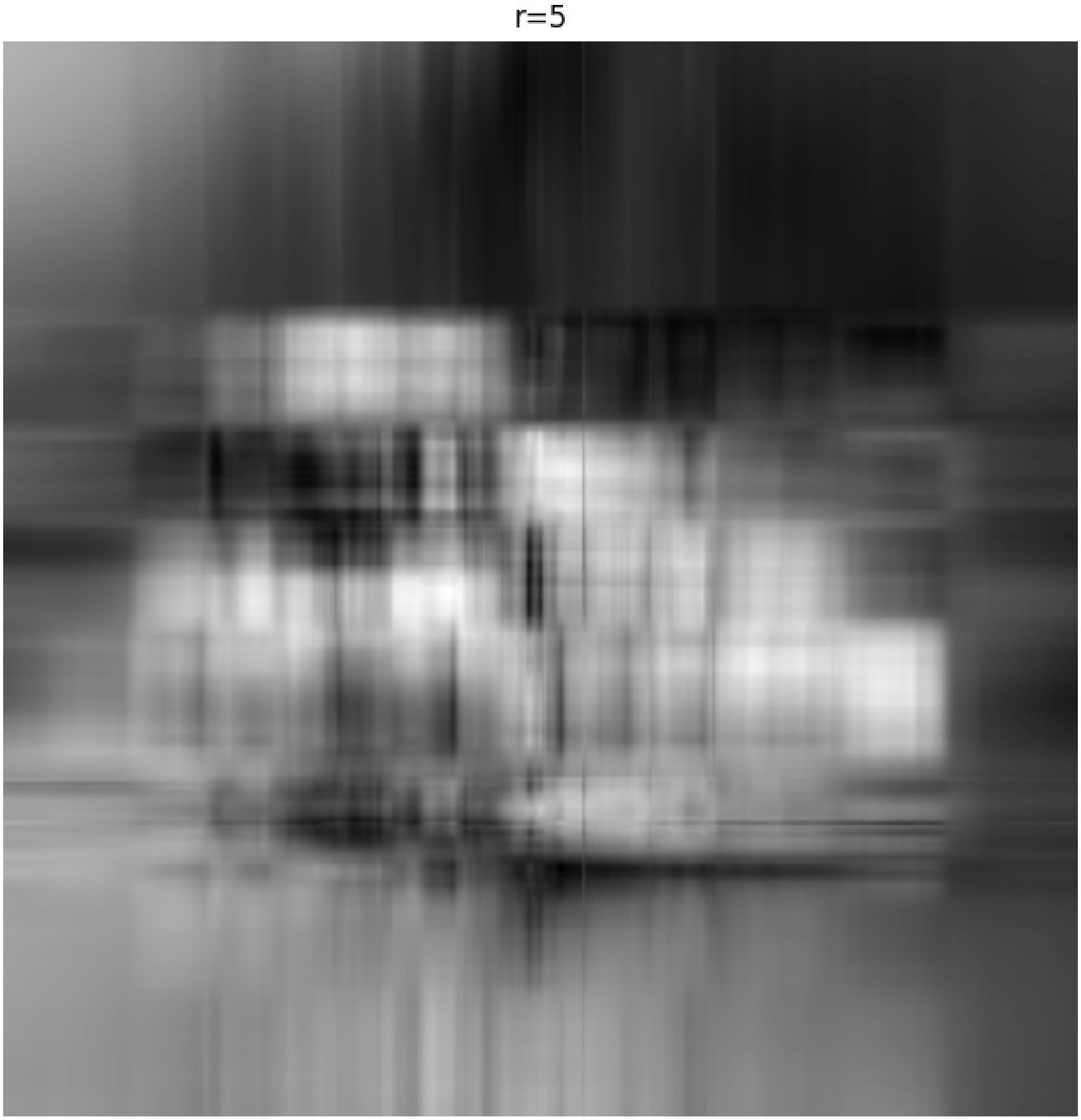
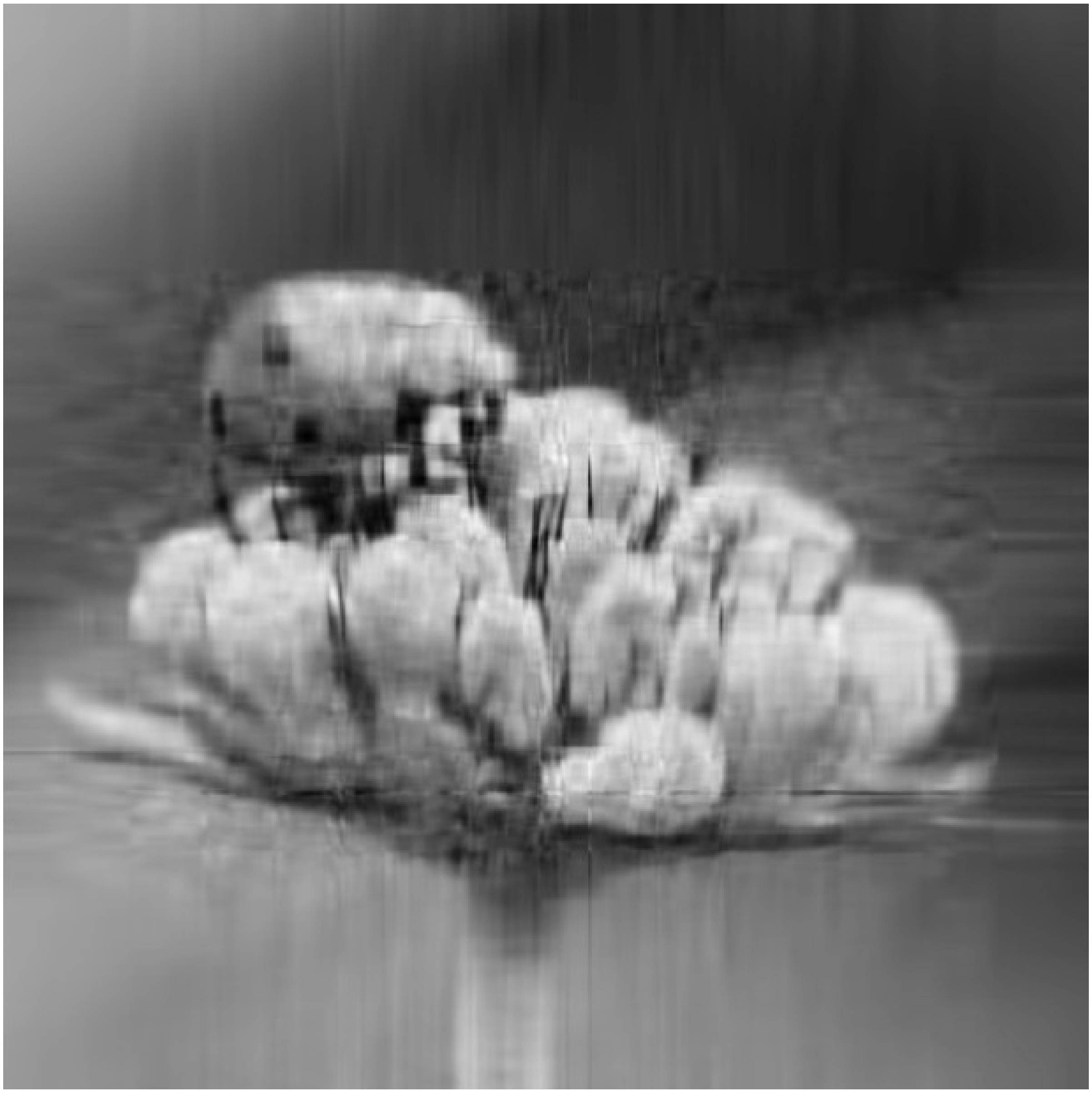
```python
from matplotlib.image import imread
import matplotlib.pyplot as plt
import numpy as np
import os
plt.rcParams['figure.figsize'] = [16,8]

A = imread('13.webp')
X = np.mean(A,-1) # convert RGB to grayscale

#img = plt.imshow(256-X)
img = plt.imshow(X)
img.set_cmap('gray')
plt.axis('off')
plt.show()
```

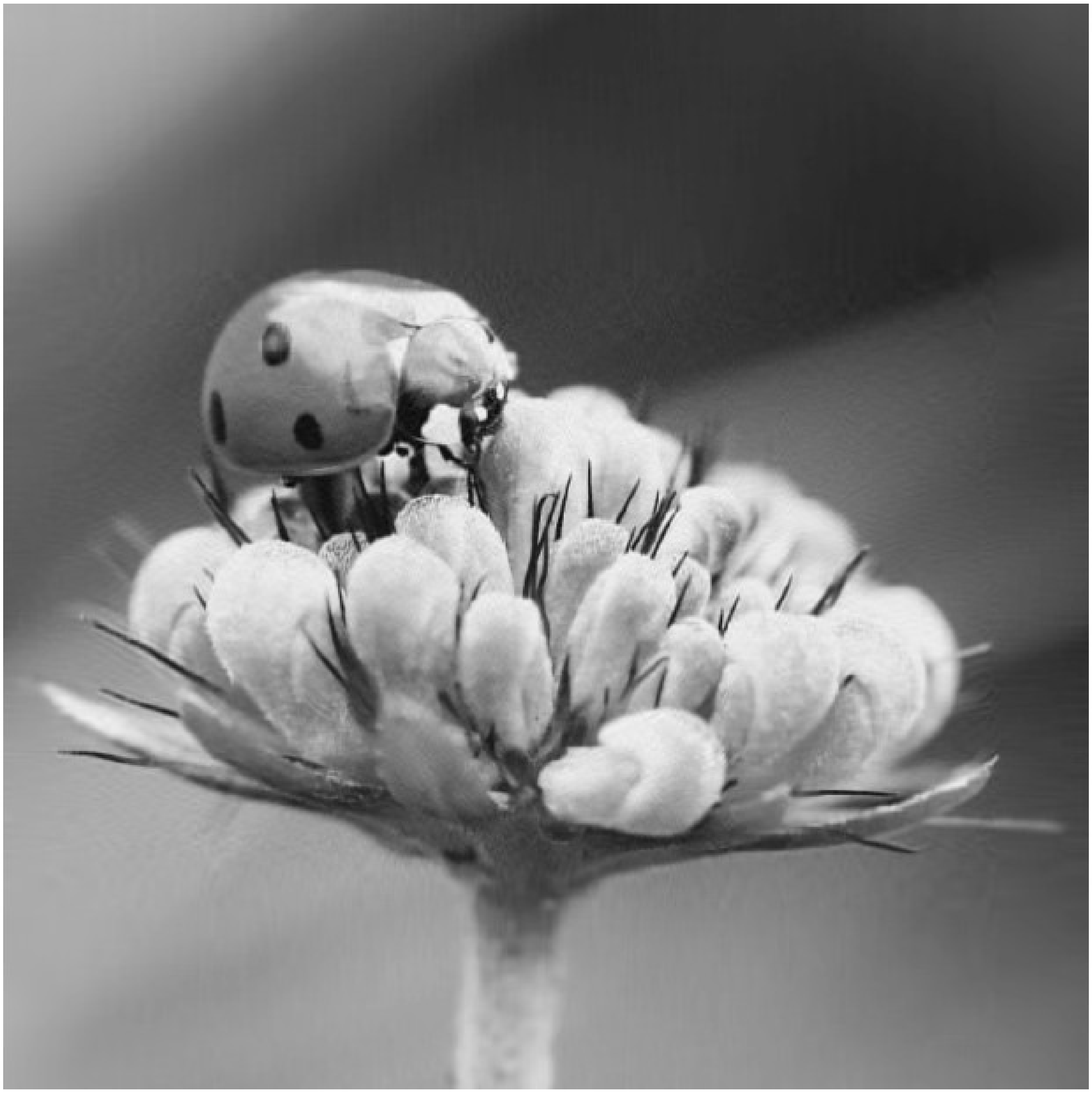
```python
U, S, VT = np.linalg.svd(X,full_matrices=False)
print(S.shape)
S = np.diag(S)

j=0
for r in (5,20,100,650):
    # Construct approximate image
    Xapprox = U[:,:r]@S[0:r,:r]@VT[:r,:]
    plt.figure(j+1)
    j += 1
    #img = plt.imshow(256-Xapprox)
    img = plt.imshow(Xapprox)
    img.set_cmap('gray')
    plt.axis('off')
    plt.title('r='+str(r))
    plt.show()
```

```
(800,)
```

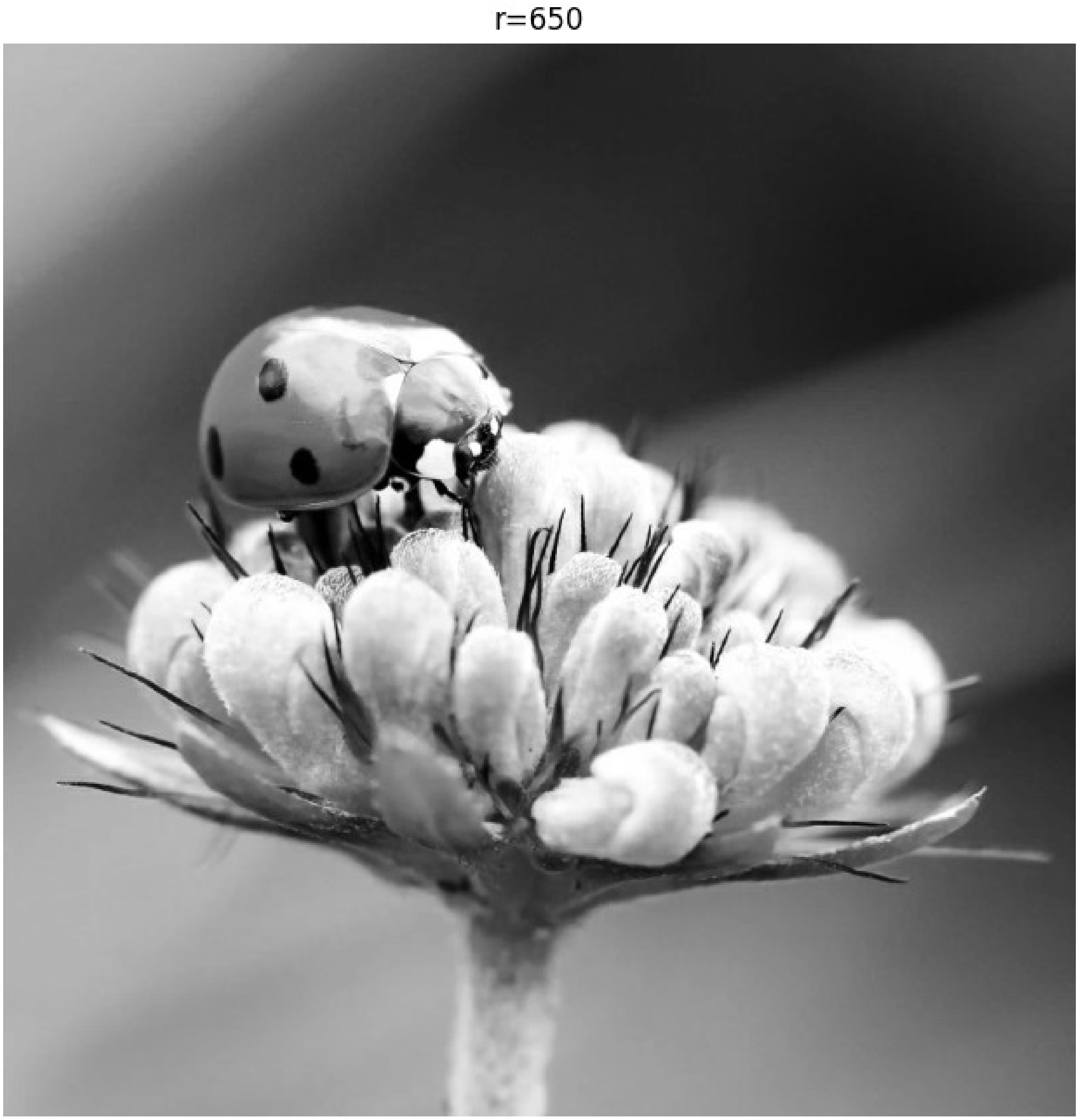




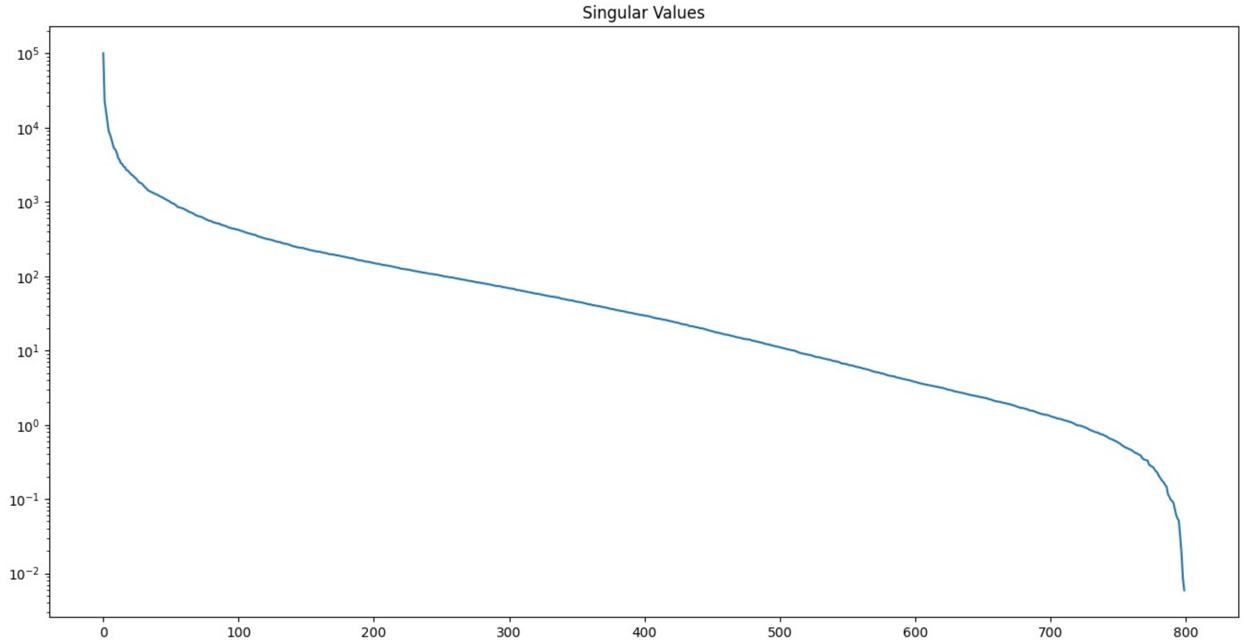
r=20

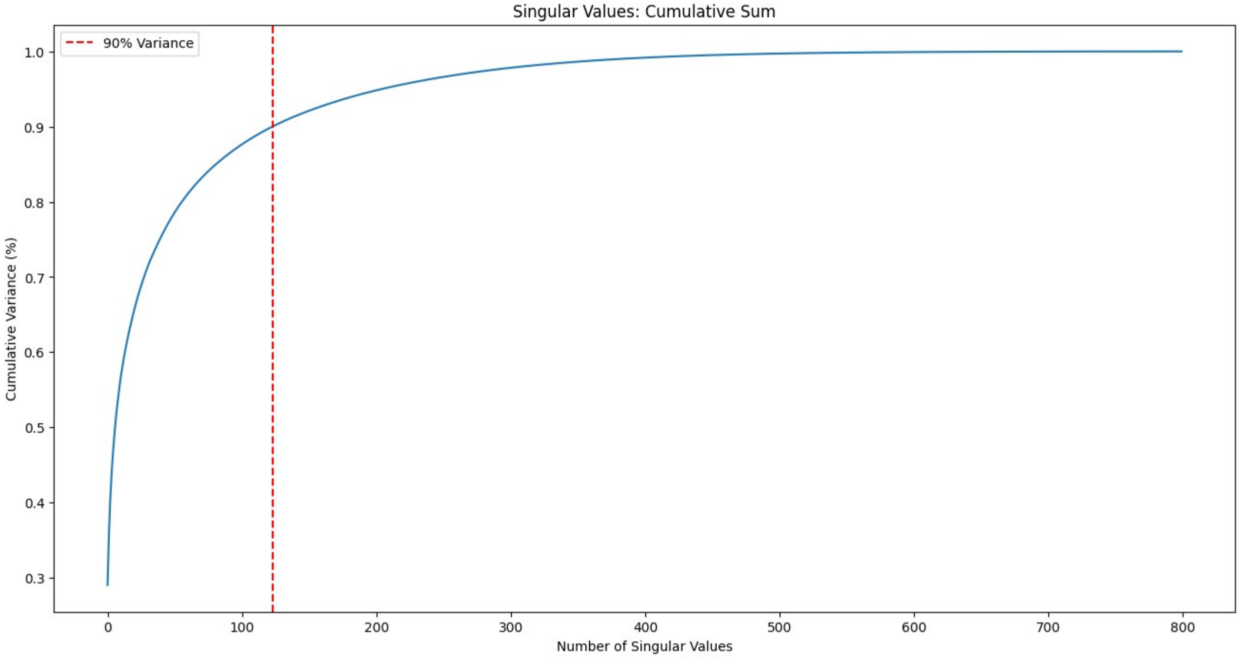


r=100



```
total_variance = np.sum(np.diag(S))
cumulative_variance = np.cumsum(np.diag(S))
percentage_variance = cumulative_variance / total_variance

index_90_percent = np.argmax(percentage_variance >= 0.9)

plt.figure(1)
plt.semilogy(np.diag(S))
plt.title('Singular Values')
plt.show()
plt.figure(2)
plt.plot(percentage_variance)
plt.title('Singular Values: Cumulative Sum')
plt.xlabel('Number of Singular Values')
plt.ylabel('Cumulative Variance (%)')

plt.axvline(x=index_90_percent, color='r', linestyle='--', label='90%
Variance')
plt.legend()
plt.show()

print("Liczba wartości singularnych dla zachowania 90% informacji:",
index_90_percent + 1)
```





```
Liczba wartości singularnych dla zachowania 90% informacji: 124
```