

## Задача 1

Найти с помощью NumPy SVD для матрицы

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 5 \\ 3 & -4 & 2 \\ 1 & 6 & 5 \\ 0 & 1 & 0 \end{pmatrix}.$$

```
Ввод [1]: import numpy as np
```

```
Ввод [2]: A = np.array([[1, 2, 0],
                        [0, 0, 5],
                        [3, -4, 2],
                        [1, 6, 5],
                        [0, 1, 0]])
print(f'Матрица A:\n{A}')
```

```
Матрица A:
[[ 1  2  0]
 [ 0  0  5]
 [ 3 -4  2]
 [ 1  6  5]
 [ 0  1  0]]
```

```
Ввод [3]: U, s, W = np.linalg.svd(A)

V = W.T
```

```
Ввод [4]: print(f'Матрица U:\n{U}')
```

Матрица U:

```
Ввод [5]: D = np.zeros_like(A, dtype=float)
D[np.diag_indices(min(A.shape))] = s

print(f'Матрица D:\n{D}')
```

Матрица D:

```
[[8.82486885 0.          0.          ]
 [0.          6.14060608 0.          ]
 [0.          0.          2.53271528]
 [0.          0.          0.          ]
 [0.          0.          0.          ]]
```

```
Ввод [6]: print(f'Матрица V.T:\n{V.T}')
```

Матрица V.T:

```
[[ 0.07116451  0.71702467  0.69340553]
 [-0.36737824  0.66514082 -0.65009301]
 [-0.92734505 -0.20847855  0.31075368]]
```

```
Ввод [7]: print(np.dot(np.dot(U, D), V.T))
```

```
[[ 1.00000000e+00  2.00000000e+00 -1.72084569e-15]
 [-6.66133815e-16  6.66133815e-16  5.00000000e+00]
 [ 3.00000000e+00 -4.00000000e+00  2.00000000e+00]
 [ 1.00000000e+00  6.00000000e+00  5.00000000e+00]
 [ 5.55111512e-17  1.00000000e+00  0.00000000e+00]]
```

```
Ввод [ ]:
```

## Задача 2

Для матрицы из предыдущего задания найти:

- а) евклидову норму;
- б) норму Фробениуса.

Ввод [ ]:

**Евклидова норма это и есть норма Фробениуса.**

```
Ввод [8]: A = np.array([[1, 2, 0],
                        [0, 0, 5],
                        [3, -4, 2],
                        [1, 6, 5],
                        [0, 1, 0]])
print(f'Матрица A:\n{A}')
```

```
Матрица A:
[[ 1  2  0]
 [ 0  0  5]
 [ 3 -4  2]
 [ 1  6  5]
 [ 0  1  0]]
```

Найдем норму Фробениуса, определяемую как

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2}.$$

```
Ввод [9]: norma_A_F = (np.sum(A**2)) ** 0.5
print(f'Норма Фробениуса матрицы A:\n{norma_A_F}')
```

```
Норма Фробениуса матрицы A:
11.045361017187261
```

Нам известно сингулярное разложение матрицы, поэтому норму Фробениуса можно вычислить по формуле

$$\|A\|_F = \sqrt{\sum_{k=1}^r \mu_k^2}.$$

где  $\mu_k$  - диагональные элементы матрицы D, а  $r$  - её ранг .

```
Ввод [10]: print(f'Диагональные элементы матрицы D:\n{s}')
```

Диагональные элементы матрицы D:  
[8.82486885 6.14060608 2.53271528]

```
Ввод [11]: norma_A_F = (np.sum(s**2)) ** 0.5  
print(f'Норма Фробениуса матрицы A:\n{norma_A_F}')
```

Норма Фробениуса матрицы A:  
11.045361017187265

***Найдем норму, используя функцию `linalg.norm()` модуля `numpy`***

```
Ввод [12]: norma_A_F = np.linalg.norm(A, ord='fro')  
print(f'Норма Фробениуса матрицы A:\n{norma_A_F}')
```

Норма Фробениуса матрицы A:  
11.045361017187261