# 12.680

EE25BTECH11043 - Nishid Khandagre

October 10, 2025

# Question

The rank of matrix is ?

$$\begin{pmatrix} 1 & 2 & 2 & 3 \\ 3 & 4 & 2 & 5 \\ 5 & 6 & 2 & 7 \\ 7 & 8 & 2 & 9 \end{pmatrix}$$

# Theoretical Solution

Let the given matrix be $\mathbf{A}$:

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 2 & 3 \\ 3 & 4 & 2 & 5 \\ 5 & 6 & 2 & 7 \\ 7 & 8 & 2 & 9 \end{pmatrix} \tag{1}$$

$$R_2 \to R_2 - 3R_1 \ , \ R_3 \to R_3 - 5R_1 \ , \ R_4 \to R_4 - 7R_1$$

$$\begin{pmatrix} 1 & 2 & 2 & 3 \\ 0 & -2 & -4 & -4 \\ 0 & -4 & -8 & -8 \\ 0 & -6 & -12 & -12 \end{pmatrix} \tag{2}$$

# Theoretical Solution

$R_2 \to -\frac{1}{2}R_2$

$$\begin{pmatrix} 1 & 2 & 2 & 3 \\ 0 & 1 & 2 & 2 \\ 0 & -4 & -8 & -8 \\ 0 & -6 & -12 & -12 \end{pmatrix} \tag{3}$$

$R_3 \to R_3 + 4R_2$ , $R_4 \to R_4 + 6R_2$ , $R_1 \to R_1 - 2R_2$

$$\begin{pmatrix} 1 & 0 & -2 & -1 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4}$$

The number of non-zero rows (pivot rows) in the row-echelon form is 2. Therefore, the rank of the matrix $A$ is 2.

# C Code

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

// Function to swap two rows in a matrix
void swapRows(int *matrix, int r1, int r2, int cols) {
    for (int j = 0; j < cols; j++) {
        int temp = *(matrix + r1 * cols + j);
        *(matrix + r1 * cols + j) = *(matrix + r2 * cols + j);
        *(matrix + r2 * cols + j) = temp;
    }
}
```

# C Code

```c
// Function to calculate the rank of a matrix
int calculate_rank(int *matrix, int rows, int cols) {
    int rank = 0;
    int lead = 0; // Current column to process

    for (int r = 0; r < rows && lead < cols; r++) {
        int i = r;
//Find a row with a non-zero element in the current column 'lead'
    while (i < rows && *(matrix + i * cols + lead) == 0) {
        i++;
    }
    if (i == rows) {
        // No pivot found in this column, move to the next column
        lead++;
        r--; // Re-process the current row with the new lead
            column
        continue;
        }
```

# C Code

```
        // Swap the current row with the pivot row
        swapRows(matrix, r, i, cols);
        // Eliminate other rows
        for (i = 0; i < rows; i++) {
            if (i != r) {
                int factor = *(matrix + i * cols + lead);
                int pivot_val = *(matrix + r * cols + lead);

                if (pivot_val == 0) continue;
                for (int j = lead; j < cols; j++) {
                    *(matrix + i * cols + j) = (pivot_val * *(
                        matrix + i * cols + j)) - (factor * *(
                        matrix + r * cols + j));
                }
            }
        }
        lead++;
    }
```

# C Code

```
    // Count non-zero rows (each non-zero row indicates a pivot)
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (*(matrix + i * cols + j) != 0) {
                rank++;
                break; // Found a non-zero element in this row,
                    move to the next row
            }
        }
    }
    return rank;
}
```

# Python Code (using C shared library)

```
import ctypes
import numpy as np

# Load the shared library
lib_code = ctypes.CDLL(./code25.so)

# Define the argument types and return type for the C function
lib_code.calculate_rank.argtypes = [
    ctypes.POINTER(ctypes.c_int), # matrix_ptr (flattened 2D
        array)
    ctypes.c_int, # rows
    ctypes.c_int # cols
]
lib_code.calculate_rank.restype = ctypes.c_int
```

# Python Code (using C shared library)

```
# The matrix from the image
matrix_data = [
    [1, 2, 2, 3],
    [3, 4, 2, 5],
    [5, 6, 2, 7],
    [7, 8, 2, 9]
]

rows = len(matrix_data)
cols = len(matrix_data[0])

# Flatten the matrix into a 1D list for C compatibility
flattened_matrix = [item for sublist in matrix_data for item in
    sublist]
```

# Python Code (using C shared library)

```
# Convert the flattened list to a C-compatible array
C_int_array = ctypes.c_int * (rows * cols)
c_matrix = C_int_array(*flattened_matrix)

# Call the C function to calculate the rank
rank = lib_code.calculate_rank(c_matrix, rows, cols)

print(fThe rank of the matrix is: {rank})
```

# Pure Python Code

```python
import numpy as np # <-- This line imports the numpy library

# Define the matrix from the image
matrix = np.array([
    [1, 2, 2, 3],
    [3, 4, 2, 5],
    [5, 6, 2, 7],
    [7, 8, 2, 9]
])

# Calculate the rank of the matrix using numpy's linear algebra
    module
rank = np.linalg.matrix_rank(matrix)

print(fThe given matrix is:\n{matrix})
print(fThe rank of the matrix is: {rank})
```