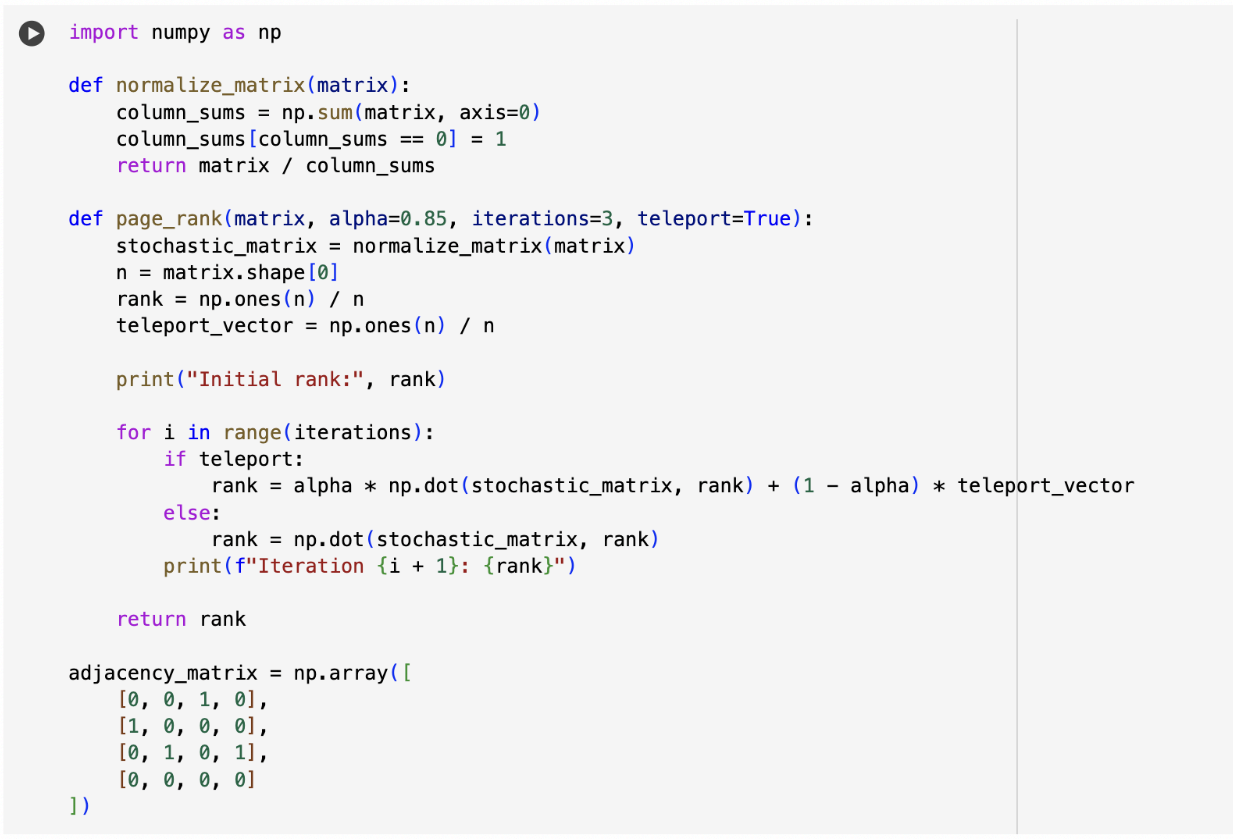
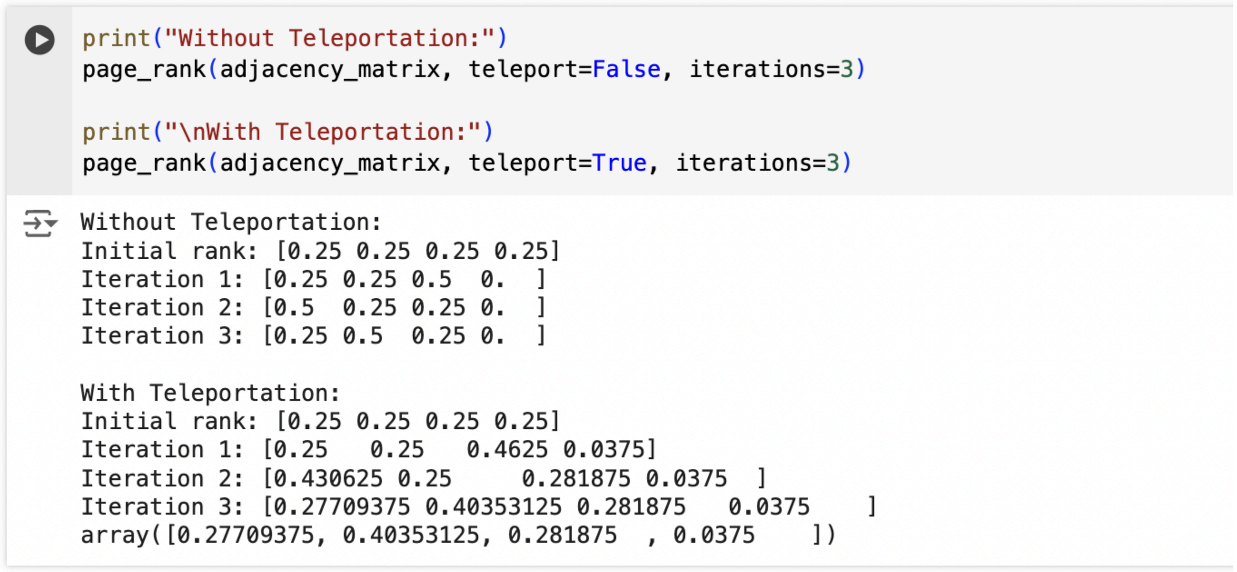
Bhuvi Ghosh
60009210191

ML-IV Experiment-5



```python
import numpy as np

def normalize_matrix(matrix):
    column_sums = np.sum(matrix, axis=0)
    column_sums[column_sums == 0] = 1
    return matrix / column_sums

def page_rank(matrix, alpha=0.85, iterations=3, teleport=True):
    stochastic_matrix = normalize_matrix(matrix)
    n = matrix.shape[0]
    rank = np.ones(n) / n
    teleport_vector = np.ones(n) / n

    print("Initial rank:", rank)

    for i in range(iterations):
        if teleport:
            rank = alpha * np.dot(stochastic_matrix, rank) + (1 - alpha) * teleport_vector
        else:
            rank = np.dot(stochastic_matrix, rank)
        print(f"Iteration {i + 1}: {rank}")

    return rank

adjacency_matrix = np.array([
    [0, 0, 1, 0],
    [1, 0, 0, 0],
    [0, 1, 0, 1],
    [0, 0, 0, 0]
])
```



```python
print("Without Teleportation:")
page_rank(adjacency_matrix, teleport=False, iterations=3)

print("\nWith Teleportation:")
page_rank(adjacency_matrix, teleport=True, iterations=3)
```

```
Without Teleportation:
Initial rank: [0.25 0.25 0.25 0.25]
Iteration 1: [0.25 0.25 0.5  0.  ]
Iteration 2: [0.5  0.25 0.25 0.  ]
Iteration 3: [0.25 0.5  0.25 0.  ]

With Teleportation:
Initial rank: [0.25 0.25 0.25 0.25]
Iteration 1: [0.25   0.25   0.4625 0.0375]
Iteration 2: [0.430625 0.25     0.281875 0.0375  ]
Iteration 3: [0.27709375 0.40353125 0.281875   0.0375    ]
array([0.27709375, 0.40353125, 0.281875  , 0.0375    ])
```