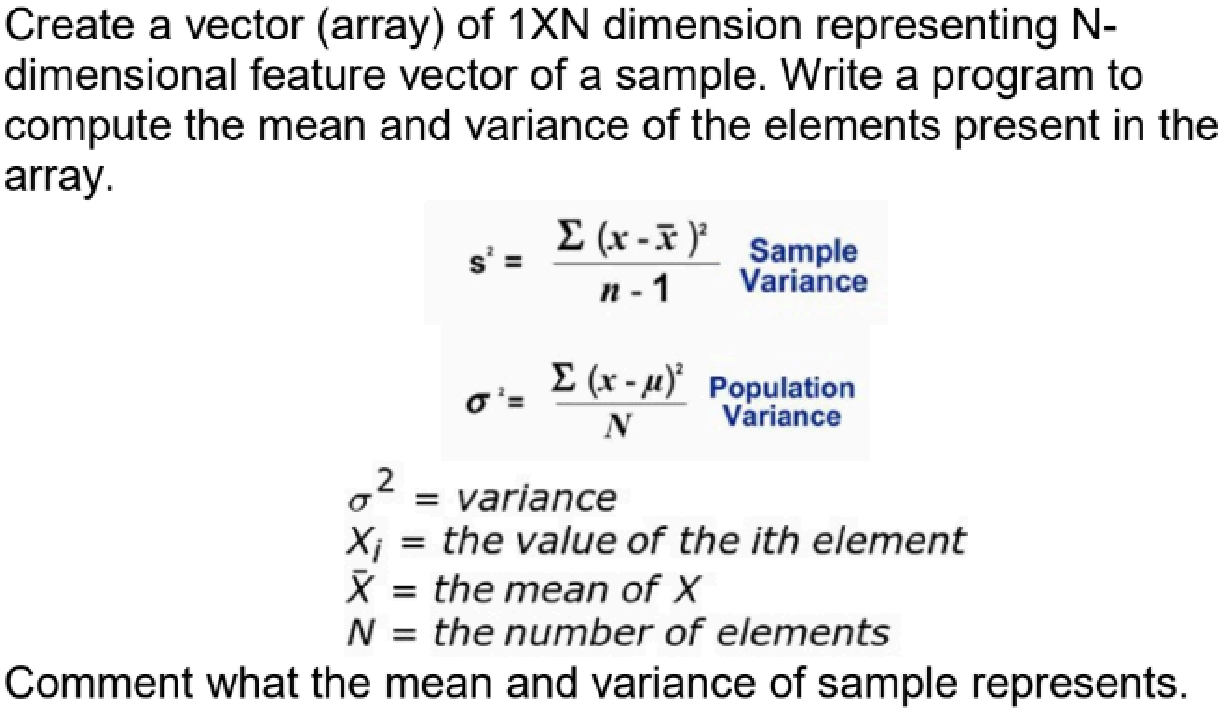
Machine Learning Lab-03 Assignment

Perumalla Dharan
AP21110010201
CSE-D

1.



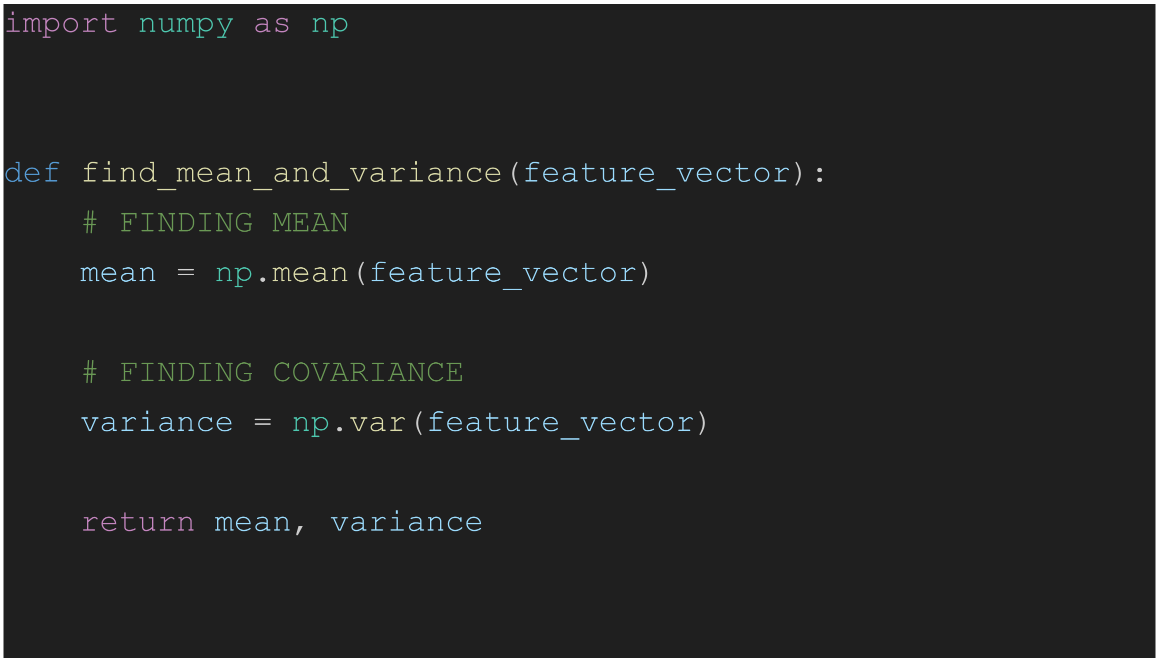
Create a vector (array) of 1XN dimension representing N-dimensional feature vector of a sample. Write a program to compute the mean and variance of the elements present in the array.

$$s^2 = \frac{\Sigma (x - \bar{x})^2}{n - 1}$$ Sample Variance

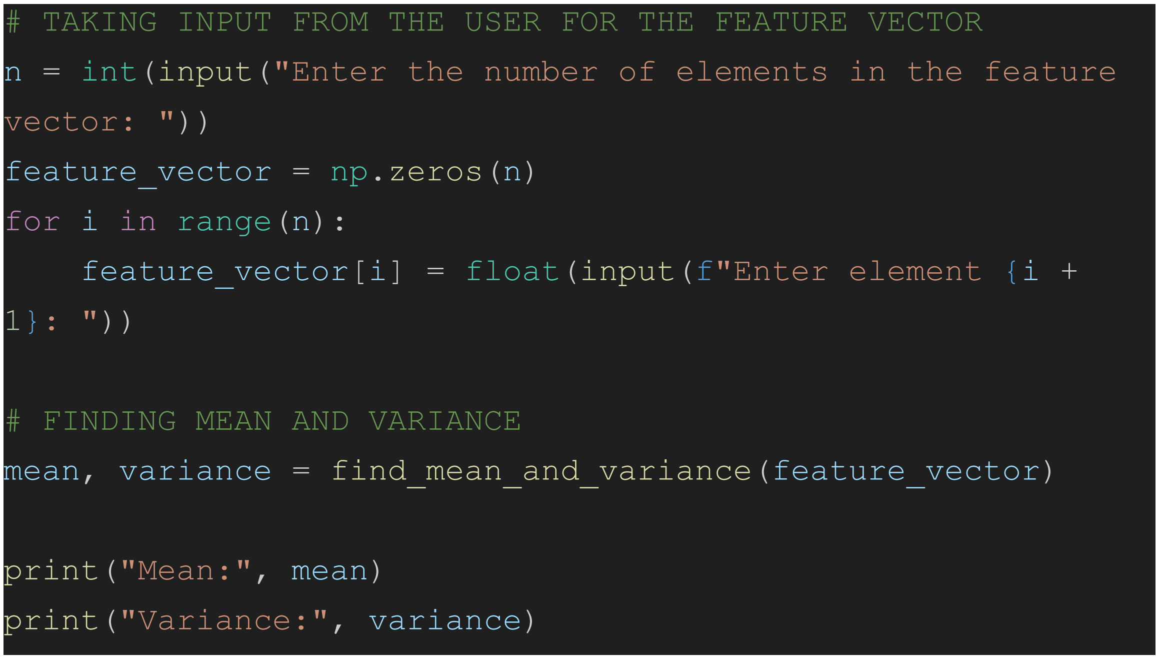
$$\sigma^2 = \frac{\Sigma (x - \mu)^2}{N}$$ Population Variance

σ^2 = *variance*
X_i = *the value of the ith element*
$\bar{X}$ = *the mean of X*
N = *the number of elements*

Comment what the mean and variance of sample represents.

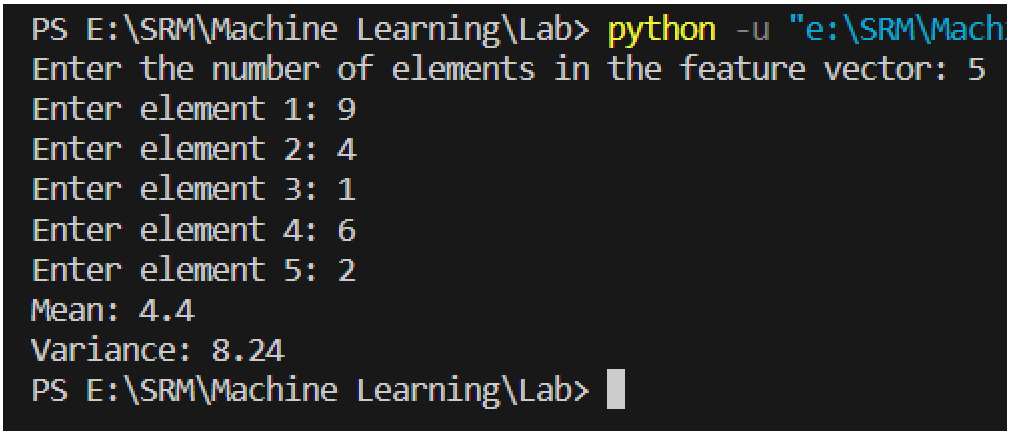


```
import numpy as np


def find_mean_and_variance(feature_vector):
    # FINDING MEAN
    mean = np.mean(feature_vector)

    # FINDING COVARIANCE
    variance = np.var(feature_vector)

    return mean, variance
```



```python
# TAKING INPUT FROM THE USER FOR THE FEATURE VECTOR
n = int(input("Enter the number of elements in the feature vector: "))
feature_vector = np.zeros(n)
for i in range(n):
    feature_vector[i] = float(input(f"Enter element {i + 1}: "))

# FINDING MEAN AND VARIANCE
mean, variance = find_mean_and_variance(feature_vector)

print("Mean:", mean)
print("Variance:", variance)
```

Output-



```
PS E:\SRM\Machine Learning\Lab> python -u "e:\SRM\Mach
Enter the number of elements in the feature vector: 5
Enter element 1: 9
Enter element 2: 4
Enter element 3: 1
Enter element 4: 6
Enter element 5: 2
Mean: 4.4
Variance: 8.24
PS E:\SRM\Machine Learning\Lab>
```

2.

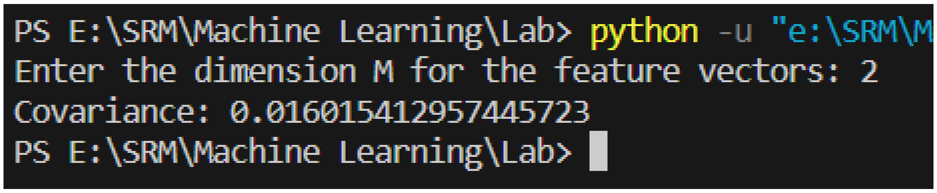
Create two vectors each of dimension 1XM each representing N-dimensional feature vector of a sample. Write a program to compute the Covariance between them.

$$COV(x,y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{n-1}$$

Comment what Covariance between two vectors represents.

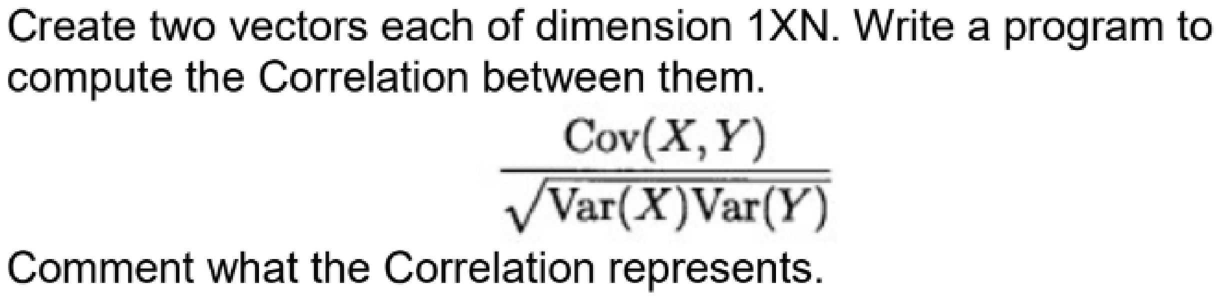
```
import numpy as np


def find_covariance(vector1, vector2):
    # FINDING COVARIANCE
    covariance = np.cov(vector1, vector2)[0][1]

    return covariance


M = int(input("Enter the dimension M for the feature vectors: "))
vector1 = np.random.rand(M)
vector2 = np.random.rand(M)

# FINDING COVARIANCE
covariance = find_covariance(vector1, vector2)

print("Covariance:", covariance)
```

Output-



```
PS E:\SRM\Machine Learning\Lab> python -u "e:\SRM\M
Enter the dimension M for the feature vectors: 2
Covariance: 0.016015412957445723
PS E:\SRM\Machine Learning\Lab> 
```

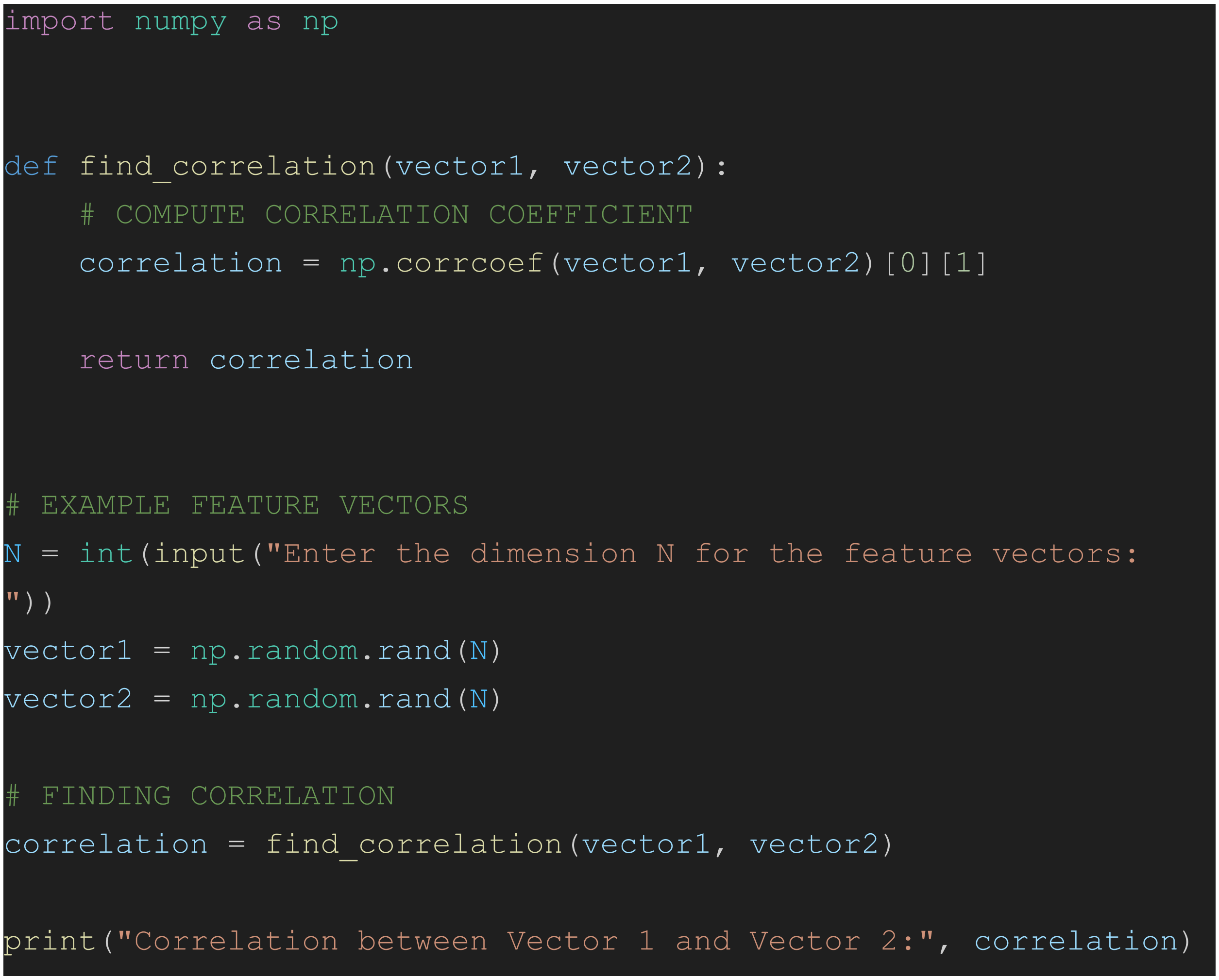
3.



Create two vectors each of dimension 1XN. Write a program to compute the Correlation between them.

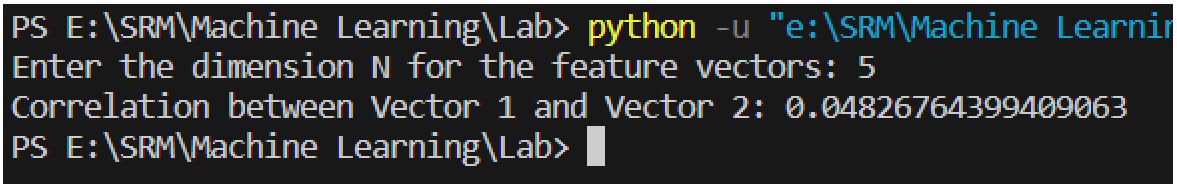
$$\frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}$$

Comment what the Correlation represents.



```python
import numpy as np


def find_correlation(vector1, vector2):
    # COMPUTE CORRELATION COEFFICIENT
    correlation = np.corrcoef(vector1, vector2)[0][1]

    return correlation


# EXAMPLE FEATURE VECTORS
N = int(input("Enter the dimension N for the feature vectors: "))
vector1 = np.random.rand(N)
vector2 = np.random.rand(N)

# FINDING CORRELATION
correlation = find_correlation(vector1, vector2)

print("Correlation between Vector 1 and Vector 2:", correlation)
```

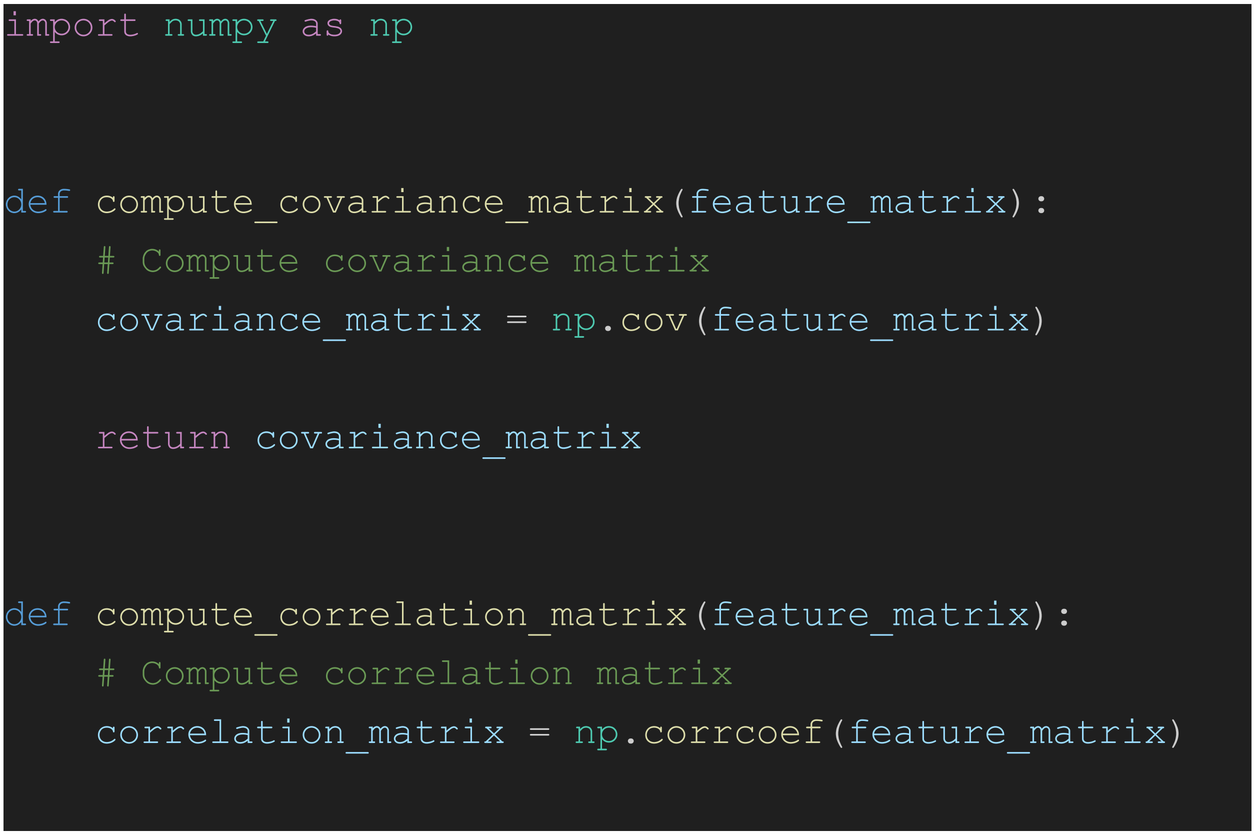
Output-



```
PS E:\SRM\Machine Learning\Lab> python -u "e:\SRM\Machine Learni
Enter the dimension N for the feature vectors: 5
Correlation between Vector 1 and Vector 2: 0.04826764399409063
PS E:\SRM\Machine Learning\Lab>
```

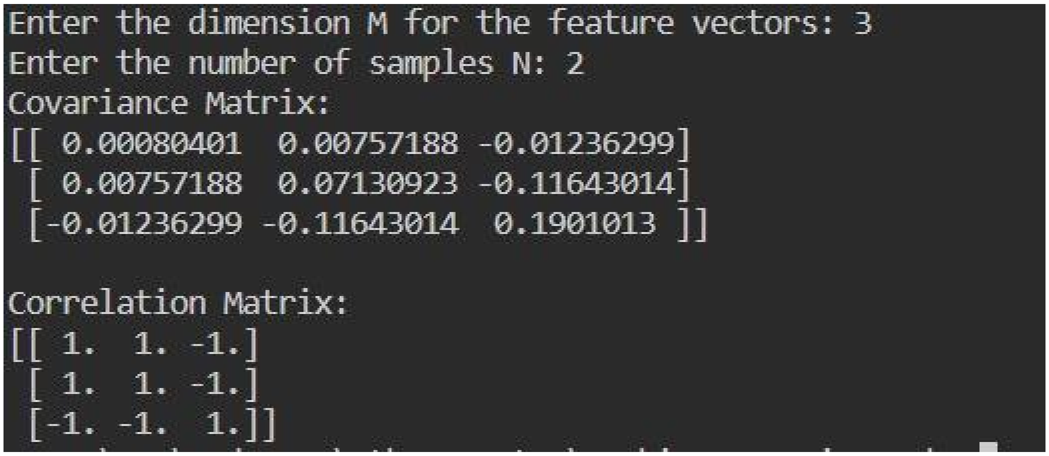
4.

Create a Matrix of MXN dimension representing the M-dimensional feature vector for N number of samples i. e (i,j)th entry of the matrix represents the ith feature of jth sample. Write a program to compute the covariance matrix and correlation matrix. Comment on takeaways from these matrixes.



```python
import numpy as np


def compute_covariance_matrix(feature_matrix):
    # Compute covariance matrix
    covariance_matrix = np.cov(feature_matrix)

    return covariance_matrix


def compute_correlation_matrix(feature_matrix):
    # Compute correlation matrix
    correlation_matrix = np.corrcoef(feature_matrix)
    return correlation_matrix


# Example feature matrix dimensions
M = int(input("Enter the dimension M for the feature
vectors: "))
N = int(input("Enter the number of samples N: "))

# Generate a random feature matrix of dimension MxN
feature_matrix = np.random.rand(M, N)

# Compute covariance matrix
covariance_matrix =
compute_covariance_matrix(feature_matrix)

# Compute correlation matrix
correlation_matrix =
compute_correlation_matrix(feature_matrix)

print("Covariance Matrix:")
print(covariance_matrix)

print("\nCorrelation Matrix:")
print(correlation_matrix)
```

Output-



```
Enter the dimension M for the feature vectors: 3
Enter the number of samples N: 2
Covariance Matrix:
[[ 0.00080401  0.00757188 -0.01236299]
 [ 0.00757188  0.07130923 -0.11643014]
 [-0.01236299 -0.11643014  0.1901013 ]]

Correlation Matrix:
[[ 1.  1. -1.]
 [ 1.  1. -1.]
 [-1. -1.  1.]]
```