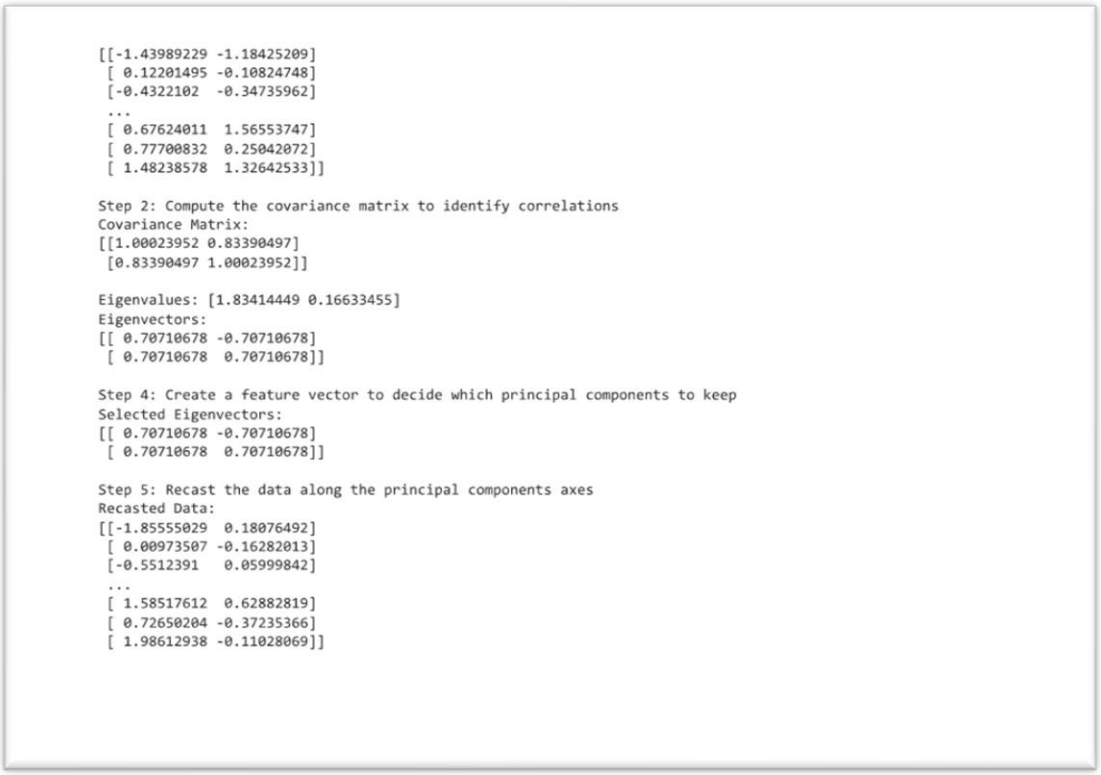
```
import pandas as pd
import numpy as np


def standardize_data(X):
    mean_X = np.mean(X, axis=0)
    std_X = np.std(X, axis=0)
    standardized_X = (X - mean_X) / std_X
    return standardized_X


def compute_covariance_matrix(X):
    covariance_matrix = np.cov(X, rowvar=False)
    return covariance_matrix


def compute_eigen(covariance_matrix):
    eigenvalues, eigenvectors = np.linalg.eig(covariance_matrix)
    return eigenvalues, eigenvectors


def feature_vector(eigenvalues, eigenvectors, variance_threshold=0.95):
    cumulative_variance = np.cumsum(eigenvalues) / np.sum(eigenvalues)
    num_components = np.argmax(cumulative_variance >= variance_threshold) + 1
    selected_eigenvectors = eigenvectors[:, :num_components]
    return selected_eigenvectors


def recast_data(X, selected_eigenvectors):
    recasted_data = np.dot(X, selected_eigenvectors)
```

```
    return recasted_data


data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/abalone.csv')


selected_data = data.iloc[:, [2, 3]].values


print("Step 1: Standardize the range of continuous initial variables")
standardized_data = standardize_data(selected_data)
print("Standardized Data:")
print(standardized_data)


print("\nStep 2: Compute the covariance matrix to identify correlations")
cov_matrix = compute_covariance_matrix(standardized_data)
print("Covariance Matrix:")
print(cov_matrix)


eigenvalues, eigenvectors = compute_eigen(cov_matrix)

print("\nEigenvalues:", eigenvalues)
print("Eigenvectors:")
print(eigenvectors)


print("\nStep 4: Create a feature vector to decide which principal components to keep")
selected_eigenvectors = feature_vector(eigenvalues, eigenvectors)
print("Selected Eigenvectors:")
```

```
print(selected_eigenvectors)


print("\nStep 5: Recast the data along the principal components axes")
recasted_data = recast_data(standardized_data, selected_eigenvectors)
print("Recasted Data:")
print(recasted_data)
```

OUTPUT:



```
[[-1.43989229 -1.18425209]
 [ 0.12201495 -0.10824748]
 [-0.4322102  -0.34735962]
 ...
 [ 0.67624011  1.56553747]
 [ 0.77700832  0.25042072]
 [ 1.48238578  1.32642533]]

Step 2: Compute the covariance matrix to identify correlations
Covariance Matrix:
[[1.00023952 0.83390497]
 [0.83390497 1.00023952]]

Eigenvalues: [1.83414449 0.16633455]
Eigenvectors:
[[ 0.70710678 -0.70710678]
 [ 0.70710678  0.70710678]]

Step 4: Create a feature vector to decide which principal components to keep
Selected Eigenvectors:
[[ 0.70710678 -0.70710678]
 [ 0.70710678  0.70710678]]

Step 5: Recast the data along the principal components axes
Recasted Data:
[[-1.85555029  0.18076492]
 [ 0.00973507 -0.16282013]
 [-0.5512391   0.05999842]
 ...
 [ 1.58517612  0.62882819]
 [ 0.72650204 -0.37235366]
 [ 1.98612938 -0.11028069]]
```