Muhammad Tahir K214503

DS LAB

creating a matrix of 6 * 8

```
from numpy import array
from numpy.linalg import eig
A = array([[5, 4, 11, 10, 5, 4,6 ,9], [4, 5, 10, 11, 5, 3, 2, 7], [11, 10, 25, 24, 2, 3, 5, 8], [10, 11, 24, 25, 4, 5, 11, 4], [2, 10, 14
print(A)
```

```
[[ 5  4 11 10  5  4  6  9]
 [ 4  5 10 11  5  3  2  7]
 [11 10 25 24  2  3  5  8]
 [10 11 24 25  4  5 11  4]
 [ 2 10 14 24 12  3  6 11]
 [19 16 14 25 12  5  7  9]]
```

```
AT=A.transpose()
print(AT)
```

```
[[ 5  4 11 10  2 19]
 [ 4  5 10 11 10 16]
 [11 10 25 24 14 14]
 [10 11 24 25 24 25]
 [ 5  5  2  4 12 12]
 [ 4  3  3  5  3  5]
 [ 6  2  5 11  6  7]
 [ 9  7  8  4 11  9]]
```

```
import numpy as np
AAT=np.matmul(A, AT)
print(AAT)
```

```
[[ 420  372  734  750  651  766]
 [ 372  349  693  695  620  723]
 [ 734  693 1524 1530 1199 1465]
 [ 750  695 1530 1600 1239 1513]
 [ 651  620 1199 1239 1186 1294]
 [ 766  723 1465 1513 1294 1737]]
```

```
from numpy import linalg as LA
w, v=LA.eig(AAT) # drop one column
print(v)
print(w)
```

```
[[-0.24428945  0.04321798 -0.21027774  0.30673538 -0.69537734 -0.5626739 ]
 [-0.22894429  0.07270125 -0.1968392  -0.865957   -0.36105379  0.15268268]
 [-0.47728064 -0.51489321  0.07192584  0.27725997 -0.25073887  0.60180709]
 [-0.49015046 -0.47826428  0.10224206 -0.21211664  0.44456715 -0.52718967]
 [-0.41047103  0.36426077 -0.72276975  0.17999335  0.35391549  0.13703188]
 [-0.50137454  0.60523474  0.61564171  0.04204421  0.01801151  0.03475092]]
[6.37348539e+03 2.15423451e+02 1.47458664e+02 1.61613168e+00
 4.71566381e+01 3.08597267e+01]
```

```
w.sort(axis=-1) # Sort the Eigen Values in Asending Order
print(w)
w[1:] # Top 3 elements in a Sorted array
```

```
[1.61613168e+00 3.08597267e+01 4.71566381e+01 1.47458664e+02
 2.15423451e+02 6.37348539e+03]
array([  30.85972674,   47.15663814,  147.45866449,  215.42345055,
       6373.48538839])
```

```
# Method 1: v[:,0:3] # First Three Vectors

# Method 2" Delete column at index 3, as its EigenValue is Very Small

FinalSelectedVectors = np.delete(v, 3, axis=1)
print('Modified Eigen Vectors by removing columns at index 3')
print(FinalSelectedVectors)
```

```
Modified Eigen Vectors by removing columns at index 3
[[-0.24428945  0.04321798 -0.21027774 -0.69537734 -0.5626739 ]
 [-0.22894429  0.07270125 -0.1968392  -0.36105379  0.15268268]
 [-0.47728064 -0.51489321  0.07192584 -0.25073887  0.60180709]
 [-0.49015046 -0.47826428  0.10224206  0.44456715 -0.52718967]
 [-0.41047103  0.36426077 -0.72276975  0.35391549  0.13703188]
 [-0.50137454  0.60523474  0.61564171  0.01801151  0.03475092]]
```

```
# Append two Zero Rows at the end of the Selected Eigen Vectors
AppendedVectors = np.insert(FinalSelectedVectors,[4], [[0],[0], ], axis = 0)
print(AppendedVectors)
```

```
[[-0.24428945  0.04321798 -0.21027774 -0.69537734 -0.5626739 ]
 [-0.22894429  0.07270125 -0.1968392  -0.36105379  0.15268268]
 [-0.47728064 -0.51489321  0.07192584 -0.25073887  0.60180709]
 [-0.49015046 -0.47826428  0.10224206  0.44456715 -0.52718967]
 [ 0.          0.          0.          0.          0.        ]
 [ 0.          0.          0.          0.          0.        ]
 [-0.41047103  0.36426077 -0.72276975  0.35391549  0.13703188]
 [-0.50137454  0.60523474  0.61564171  0.01801151  0.03475092]]
```

```
PCA=np.matmul(A, AppendedVectors)

print(PCA)
```

```
[[-19.26401308  -2.30689587   1.17901629  -0.94796135   0.28029217]
 [-16.61690455  -4.90829623   2.88256662  -1.37001675  -0.7509774 ]
 [-34.73560535 -16.48508079   1.28179331  -4.94487908  -1.30679402]
 [-35.19045801 -16.65434295  -5.47363729  -1.86380261  -1.0372466 ]
 [-29.2015079   -9.03025236   3.50726412   4.47959448  -2.62132231]
 [-34.62596665  -9.18381218  -3.10030383  -8.74568333 -11.73034214]]
```

```
U, s, V = np.linalg.svd(A, full_matrices=True) # , compute_uv=True
```

```
U.shape, V.shape, s.shape
```

```
((6, 6), (8, 8), (6,))
```

```
print(U)
print(V)
print(s)
```

```
[[ 0.24428945  0.04321798 -0.21027774 -0.69537734 -0.5626739   0.30673538]
 [ 0.22894429  0.07270125 -0.1968392  -0.36105379  0.15268268 -0.865957  ]
 [ 0.47728064 -0.51489321  0.07192584 -0.25073887  0.60180709  0.27725997]
 [ 0.49015046 -0.47826428  0.10224206  0.44456715 -0.52718967 -0.21211664]
 [ 0.41047103  0.36426077 -0.72276975  0.35391549  0.13703188  0.17999335]
 [ 0.50137454  0.60523474  0.61564171  0.01801151  0.03475092  0.04204421]]
[[ 0.28353637  0.30579715  0.51905246  0.63951919  0.20321487  0.11630205
   0.19633376  0.23307849]
 [ 0.1559147   0.23525444 -0.65238964  0.05389111  0.6316347   0.03910326
  -0.06870843  0.29430793]
 [ 0.84215572  0.21750213 -0.12593691 -0.15985565 -0.22797026  0.01690309
  -0.01632013 -0.38669126]
 [-0.317968    0.23640518 -0.2405059   0.45366376  0.06666005 -0.18089924
   0.1444206  -0.72203161]
 [ 0.01434233  0.11845506  0.02430149  0.26532793 -0.1608789  -0.36692496
  -0.86321313  0.09550065]
 [ 0.1237885  -0.15015359 -0.26450305  0.20774127 -0.33478606 -0.66825346
   0.42145563  0.33577189]
 [-0.1134532   0.54981136  0.33037936 -0.49289823  0.25757926 -0.50663825
   0.09652725 -0.01974188]
 [-0.23783797  0.63959255 -0.22708185 -0.01556628 -0.54682403  0.33772116
   0.05940782  0.25679682]]
[79.83411168 14.67731074 12.14325593  6.86706911  5.55515317  1.27127168]
```