Introduction to Linear Algebra

Week 1 Assignment Submission

1a(i)

v = [2, 4, 6, 8, 10] and u = [1, 3, 5, 7, 9]

component-wise division v/u:

= [2/1, 4/3, 6/5, 8/7, 10/9]

Answer = [2, 1.33, 1.2, 1.14, 1.11]

1a(ii)

3v − 2u + 1/2v:

= 3[2, 4, 6, 8, 10] - 2[1, 3, 5, 7, 9] + ½ [2, 4, 6, 8, 10]

= [6, 12, 18, 24, 30] - [2, 6, 10, 14, 18] + [1, 2, 3, 4, 5]
= [4, 6, 8, 10, 12] + [1, 2, 3, 4, 5]
Answer = [5, 8, 11, 14, 17]

1b.
Euclidean norm (magnitude) of the vector v = [8, 0, 5, 2, 1]

$$\sqrt{(8^2)+(0^2)+(5^2)+(2^2)+(1^2)}$$

=

$$\sqrt{(64)+(0)+(25)+(4)+(1)}$$

=

$$\sqrt{94}$$

Answer = $\mathbf{9.695}$

2.

Compute the determinant of the matrix A = $\begin{pmatrix} 2 & 3 & 7 \\ -4 & 0 & 6 \\ 1 & 5 & 0 \end{pmatrix}$

Det A = $\begin{pmatrix} 2 & \cancel{3} & \cancel{7} \\ \cancel{-4} & 0 & 6 \\ \cancel{1} & 5 & 0 \end{pmatrix} = 2\begin{pmatrix} 0 & 6 \\ 5 & 0 \end{pmatrix}$ = 2(0 – 30) = -60

= $\begin{pmatrix} \cancel{2} & 3 & \cancel{7} \\ -4 & \cancel{0} & 6 \\ 1 & \cancel{5} & 0 \end{pmatrix} = 3\begin{pmatrix} -4 & 6 \\ 1 & 0 \end{pmatrix}$ = 3(0 - 6) = -18

= $\begin{pmatrix} \cancel{2} & \cancel{3} & 7 \\ -4 & 0 & \cancel{6} \\ 1 & 5 & \cancel{0} \end{pmatrix} = 7\begin{pmatrix} -4 & 0 \\ 1 & 5 \end{pmatrix}$ = 7(-20 – 0) = -140

Det A = (-60) – (-18) + (-140)

Answer = Det A = -182

3a.

Compute the eigenvalues and eigenvectors of vector B = $\begin{pmatrix} 3 & 2 & 4 \\ 2 & 0 & 2 \\ 4 & 2 & 3 \end{pmatrix}$

Det (B – λI_3) = 0

$$= \det\left(\begin{pmatrix} 3 & 2 & 4 \\ 2 & 0 & 2 \\ 4 & 2 & 3 \end{pmatrix} - \lambda\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\right) = 0$$

$$= \det\left(\begin{pmatrix} 3 & 2 & 4 \\ 2 & 0 & 2 \\ 4 & 2 & 3 \end{pmatrix} - \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{pmatrix}\right)$$

$$= \det\begin{pmatrix} 3-\lambda & 2 & 4 \\ 2 & 0-\lambda & 2 \\ 4 & 2 & 3-\lambda \end{pmatrix} = 0$$

$$= 3 - \lambda \begin{pmatrix} 0-\lambda & 2 \\ 2 & 3-\lambda \end{pmatrix} - 2\begin{pmatrix} 2 & 2 \\ 4 & 3-\lambda \end{pmatrix} + 4\begin{pmatrix} 2 & 0-\lambda \\ 4 & 2 \end{pmatrix} = 0$$

= (3 – λ) (-λ(3-λ) – 4) – 2(2(3-λ) – 8) + 4(4 – (-4λ) = 0

= (3 – λ) (-3λ + λ^2 – 4) – 2(6 – 2λ – 8) + 4(4 + 4λ) = 0

= (3 – λ) (λ^2 – 3λ – 4) – 2 (-2λ – 2) + (16 + 16λ) = 0

= (3$\lambda^{2\,-}$ 9λ – 12 – λ^3 + 3λ^2 + 4λ) + 4λ + 4 + 16 + 16λ = 0

= (-λ^3 + 6λ^2 – 5λ – 12) + 20λ + 20 = 0

= (-λ^3 + 6λ^2 + 15λ + 8) = 0

= −(λ + 1) (λ^2 − 7λ − 8) = 0
= −(λ+1) (λ+1) (λ−8) = 0

Answer: Eigenvalues =

1. **$\lambda_1 = -1$**
2. **$\lambda_2 = 8$**
3. **$\lambda_3 = -1$**

To calculate Eigenvectors,

Let V = $\begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{c} \end{pmatrix}$

For λ=-1, V is an eigenvector of B if

BV = V

$= \begin{pmatrix} 3-(-1) & 2 & 4 \\ 2 & 0-(-1) & 2 \\ 4 & 2 & 3-(-1) \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$

Using gaussian elimination:

$= \begin{array}{c} \begin{pmatrix} 4 & 2 & 4 \\ 2 & 1 & 2 \\ 4 & 2 & 4 \end{pmatrix} \end{array} \begin{array}{c} o \\ p \\ q \end{array}$

$o/4 \rightarrow o = \begin{pmatrix} 1 & 1/2 & 1 \\ 2 & 1 & 2 \\ 4 & 2 & 4 \end{pmatrix}$

$p-2o \rightarrow p = \begin{pmatrix} 1 & 1/2 & 1 \\ 0 & 0 & 0 \\ 4 & 2 & 4 \end{pmatrix}$

$q-4o \rightarrow q = \begin{pmatrix} 1 & 1/2 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

$= a + \frac{1}{2}b + c = 0$
$a = -\frac{1}{2}b - c$
$b = b$
$c = c$

Taking a=-1 leads to $V_{\lambda=-1} = \begin{pmatrix} -\frac{1}{2}b - c \\ b \\ c \end{pmatrix}$

$= \begin{pmatrix} -\frac{1}{2}b \\ b \\ 0 \end{pmatrix} + \begin{pmatrix} -c \\ 0 \\ c \end{pmatrix}$

$= b\begin{pmatrix} -\frac{1}{2} \\ 1 \\ 0 \end{pmatrix} + c\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$

=Let b = 1 and c = 0

$$\mathbf{V_1} = \begin{pmatrix} -\frac{1}{2} \\ 1 \\ 0 \end{pmatrix}$$

=Let b = 0 and c = 1

$$\mathbf{V_2} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

For λ=8, V is a eigenvector of B if

BV = V

$$= \begin{pmatrix} 3-(8) & 2 & 4 \\ 2 & 0-(8) & 2 \\ 4 & 2 & 3-(8) \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

Using gaussian elimination:

$$= \begin{array}{cc} \begin{pmatrix} -5 & 2 & 4 \\ 2 & -8 & 2 \\ 4 & 2 & -5 \end{pmatrix} & \begin{matrix} o \\ p \\ q \end{matrix} \end{array}$$

$$o/-5 \rightarrow o = \begin{pmatrix} 1 & \frac{2}{-5} & \frac{4}{-5} \\ 2 & -8 & 2 \\ 4 & 2 & -5 \end{pmatrix}$$

$$p-2o \rightarrow p = \begin{pmatrix} 1 & \frac{2}{-5} & \frac{4}{-5} \\ 0 & -\frac{36}{5} & \frac{18}{5} \\ 4 & 2 & -5 \end{pmatrix}$$

$$q-4o \rightarrow q = \begin{pmatrix} 1 & -\frac{2}{5} & -\frac{4}{5} \\ 0 & -\frac{36}{5} & \frac{18}{5} \\ 0 & \frac{18}{5} & -\frac{9}{5} \end{pmatrix}$$

$$p \,/ -\frac{36}{5} \rightarrow p = \begin{pmatrix} 1 & -\frac{2}{5} & -\frac{4}{5} \\ 0 & 1 & -\frac{1}{2} \\ 0 & \frac{18}{5} & -\frac{9}{5} \end{pmatrix}$$

$$q - \frac{18}{5}\, p \rightarrow q = \begin{pmatrix} 1 & -\frac{2}{5} & -\frac{4}{5} \\ 0 & 1 & -\frac{1}{2} \\ 0 & 0 & 0 \end{pmatrix}$$

$$o - \left(-\frac{2}{5}\right) p \rightarrow o = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -\frac{1}{2} \\ 0 & 0 & 0 \end{pmatrix}$$

a – c = 0
b – ½c = 0

Taking a=-1 leads to $V_{\lambda=8} = \begin{pmatrix} c \\ \frac{1}{2}c \\ c \end{pmatrix}$

$=c \begin{pmatrix} 1 \\ \frac{1}{2} \\ 1 \end{pmatrix}$

=Let c = 1

$$= \begin{pmatrix} \mathbf{1} \\ \mathbf{\frac{1}{2}} \\ \mathbf{1} \end{pmatrix}$$

3(b)

Verify whether the following vectors are eigenvectors of the matrix D = $\begin{pmatrix} 3 & 1 & 2 \\ 1 & 3 & 1 \\ 2 & 1 & 3 \end{pmatrix}$

v1 = [1, 1, 1] and v2 = [−1, 1, −1].

For v1 and v2 to be eigenvectors of D:

DV = λV

D(v1) = $\begin{pmatrix} 3 & 1 & 2 \\ 1 & 3 & 1 \\ 2 & 1 & 3 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 6 \end{pmatrix}$

$\begin{pmatrix} \mathbf{6} \\ \mathbf{5} \\ \mathbf{6} \end{pmatrix}$ **is not a scalar multiple of** $\begin{pmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{pmatrix}$**, therefore v1 is not an eigenvector of D.**

D(v2) = $\begin{pmatrix} 3 & 1 & 2 \\ 1 & 3 & 1 \\ 2 & 1 & 3 \end{pmatrix}\begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -4 \\ 1 \\ -4 \end{pmatrix}$

$\begin{pmatrix} \mathbf{-4} \\ \mathbf{1} \\ \mathbf{-4} \end{pmatrix}$ **is not a scalar multiple of** $\begin{pmatrix} \mathbf{-1} \\ \mathbf{1} \\ \mathbf{-1} \end{pmatrix}$**, therefore v1 is not an eigenvector of D.**

4. (a)

Given a dataset with three features X = {(4, 2, 3),(3, 13, 4),(9, 4, 5),(0, 5, 7)}, perform PCA to reduce the dimensionality of the dataset to two dimensions.

X = $\begin{pmatrix} 4 & 2 & 3 \\ 3 & 13 & 4 \\ 9 & 4 & 5 \\ 0 & 5 & 7 \end{pmatrix}$

Step 1 - Standardization

z = value − mean standard deviation

Mean (x) = $\frac{4+3+9+0}{4} = \frac{16}{4} = 4$

Mean (y) = $\frac{2+13+4+5}{4} = \frac{24}{4} = 6$

Mean (z) = $\frac{3+4+5+7}{4} = \frac{19}{4} = 4.75$

calculate the standard deviation of each feature:

$\sigma_x = \sqrt{\frac{(4-4)^2+(3-4)^2+(9-4)^2+(0-4)^2}{4}} = \sqrt{\frac{0+(-1)^2+(5)^2+(-4)^2}{4}} = \sqrt{\frac{52}{4}} = \sqrt{10.5} = 3.24$

$\sigma_y = \sqrt{\frac{(2-6)^2+(13-6)^2+(4-6)^2+(5-6)^2}{4}} = \sqrt{\frac{(-4)^2+(7)^2+(-2)^2+(-1)^2}{4}} = \sqrt{\frac{70}{4}} = \sqrt{17.5} = 4.18$

$\sigma_z = \sqrt{\frac{(3-4.75)^2+(4-4.75)^2+(5-4.75)^2+(7-4.75)^2}{4}} = \sqrt{\frac{(-1.75)^2+(-0.75)^2+(0.25)^2+(2.25)^2}{4}} = \sqrt{\frac{8.75}{4}} = \sqrt{2.1875}$ $= 1.48$

Standardize the data:

$z_x = \frac{x-\mu}{\sigma_x}$

$z_y = \frac{y-\mu}{\sigma_y}$

$z_z = \frac{z-\mu}{\sigma_z}$

$$= \begin{pmatrix} \frac{4-4}{3.24} & \frac{2-6}{4.18} & \frac{3-4.75}{1.48} \\ \frac{3-4}{3.24} & \frac{13-6}{4.18} & \frac{4-4.75}{1.48} \\ \frac{9-4}{3.24} & \frac{4-6}{4.18} & \frac{5-4.75}{1.48} \\ \frac{0-4}{3.24} & \frac{5-6}{4.18} & \frac{7-4.75}{1.48} \end{pmatrix} = \begin{pmatrix} 0 & -0.96 & -1.18 \\ -0.31 & 1.67 & -0.51 \\ 1.54 & -0.48 & 0.17 \\ -1.23 & -0.24 & 1.52 \end{pmatrix}$$

Step 2 - Covariance Matrix Computation
Calculate the covariance matrix Σ:

$$\boldsymbol{\Sigma} = \frac{\boldsymbol{1}}{\boldsymbol{n-1}} (X^{T} \cdot X)$$

$$= \frac{\boldsymbol{1}}{\boldsymbol{3}} \begin{pmatrix} 0 & -0.31 & 1.54 & -1.23 \\ -0.96 & 1.67 & -0.48 & -0.24 \\ -1.18 & -0.51 & 0.17 & 1.52 \end{pmatrix} \begin{pmatrix} 0 & -0.96 & -1.18 \\ -0.31 & 1.67 & -0.51 \\ 1.54 & -0.48 & 0.17 \\ -1.23 & -0.24 & 1.52 \end{pmatrix}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & -0.24 & -0.36 \\ -0.24 & 1 & -0.04 \\ -0.37 & -0.04 & 1 \end{pmatrix}$$

Step 3 - Compute Eigenvectors and Eigenvalues
Compute the eigenvalues and eigenvectors of the covariance matrix Σ:

Eigenvalues: [0.54, 1.42, 1.04]

Eigenvectors: $\begin{pmatrix} -0.69 & -0.72 & 0.037 \\ -0.42 & 0.36 & -0.84 \\ -0.59 & 0.59 & 0.55 \end{pmatrix}$

Step 4 - Create Feature Vector
Rank eigenvectors by eigenvalues to obtain principal components:

Sorted eigenvalues: [1.42, 1.04, 0.54]

Top 2 eigenvectors: $\begin{pmatrix} -0.72 & 0.037 \\ 0.36 & -0.84 \\ 0.59 & 0.55 \end{pmatrix}$

Step 5: Recast the Data Along the Principal Components Axes

Xreduced = (Xstandardized)(Top 2 Eigenvectors)

$$= \begin{pmatrix} 0 & -0.96 & -1.18 \\ -0.31 & 1.67 & -0.51 \\ 1.54 & -0.48 & 0.17 \\ -1.23 & -0.24 & 1.52 \end{pmatrix} \begin{pmatrix} -0.72 & 0.037 \\ 0.36 & -0.84 \\ 0.59 & 0.55 \end{pmatrix}$$

$$\text{Xreduced} = \begin{pmatrix} -1.04 & 0.15 \\ 0.51 & -1.69 \\ -1.18 & 0.54 \\ 1.71 & 0.98 \end{pmatrix}$$

4(b)
Compute the percentage of variance explained by each principal component in the PCA transformation.

Total Variance = 0.54 + 1.42 + 1.04 = 3.0

= percentage of variance explained by each principal component = eigenvalue/total variance

$= \frac{0.54}{3.0} . 100$ **= 18%**

$= \frac{1.42}{3.0} . 100$ **= 47.3%**

$= \frac{1.04}{3.0} . 100$ **= 34.7%**

```
#import libraries
import numpy as np
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

Question 1:

Component-wise division v/u

```
#use numpy to create vector v and u
v = np.array([2,4,6,8,10])
u = np.array([1,3,5,7,9])
```

```
#divide vector v by vector u
v/u
```

```
array([2.        , 1.33333333, 1.2       , 1.14285714, 1.11111111])
```

Question 1(ii):

Compute 3v − 2u + 0.5v

```
#compute the second operation
(3*v) - (2*u) + (0.5*v)
```

```
array([ 5.,  8., 11., 14., 17.])
```

Question 1b:

Compute the Euclidean norm (magnitude) of the vector v = [8, 0, 5, 2, 1]

```
#use numpy to create vector v
v = np.array([8, 0, 5, 2, 1])
```

```
# Compute the Euclidean magnitude
euclidean_norm = np.linalg.norm(v)
euclidean_norm
```

```
9.695359714832659
```

Question 2:

Compute the determinant of a matrix A

```
#use numpy to create matrix A

A = np.array([[2, 3, 7],
              [-4, 0, 6],
              [1, 5, 0]])
```

```
# Compute the determinant
determinant = np.linalg.det(A)
determinant
```

```
-181.99999999999997
```

Question 3:

Compute the eigenvalues and eigenvectors of matrix B.

```
#use numpy to create matrix B
B = np.array([[3, 2, 4],
              [2, 0, 2],
              [4, 2, 3]])
```

```
# Compute the eigenvalues and eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(B)

# Output the results
print("Eigenvalues of the matrix B are: \n", eigenvalues)
print("\nEigenvectors of the matrix B are:\n", eigenvectors)
```

```
Eigenvalues of the matrix B are: 
 [-1.  8. -1.]

Eigenvectors of the matrix B are:
 [[-0.74535599  0.66666667 -0.21539222]
 [ 0.2981424   0.33333333 -0.77010996]
 [ 0.59628479  0.66666667  0.6004472 ]]
```

Question 3:

Verify whether the following vectors are eigenvectors of the matrix D

```
#use numpy to create matrix D
D = np.array([[3, 1, 2],
              [1, 3, 1],
              [2, 1, 3]])

#compute vector v1 and v2
v1 = np.array([1, 1, 1])
v2 = np.array([-1, 1, -1])
```

```
# Compute D * v1 and D * v2
Dv1 = np.dot(D, v1)
Dv2 = np.dot(D, v2)

print("D * v1 =", Dv1)
print("D * v2 =", Dv2)
```

```
D * v1 = [6 5 6]
D * v2 = [-4  1 -4]
```

```
# Check if D * v1 is a scalar multiple of v1
if np.allclose(Dv1, v1 * (Dv1[0] / v1[0])):
    print("v1 is an eigenvector of D with eigenvalue", Dv1[0] / v1[0])
else:
    print("v1 is not an eigenvector of D")

# Check if D * v2 is a scalar multiple of v2
if np.allclose(Dv2, v2 * (Dv2[0] / v2[0])):
    print("v2 is an eigenvector of D with eigenvalue", Dv2[0] / v2[0])
else:
    print("v2 is not an eigenvector of D")
```

```
v1 is not an eigenvector of D
v2 is not an eigenvector of D
```

Question 4:

Perform PCA on a dataset to reduce the dimensionality of the dataset to two dimensions.

```
#use numpy to create the dataset
X = np.array([[4, 2, 3],
              [3, 13, 4],
              [9, 4, 5],
              [0, 5, 7]])
```

Step 1 - Standardization

```
# Step 1: Standardize the dataset
scaler = StandardScaler(with_std=False)
X_standardized = scaler.fit_transform(X)

# Calculate the standard deviations for scaling
std_devs = np.std(X_standardized, axis=0)
X_standardized = X_standardized / std_devs
```

Step 2 - Covariance Matrix Computation

```
# Step 2: Compute the covariance matrix
n = X_standardized.shape[0]
cov_matrix = np.dot(X_standardized.T, X_standardized) / n
```

Step 3 - Compute Eigenvectors and Eigenvalues

```
# Step 3: Compute eigenvalues and eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(cov_matrix)
print("\nEigenvalues:\n", eigenvalues)
print("\nEigenvectors:\n", eigenvectors)
```

```
Eigenvalues:
 [0.54397045 1.41900369 1.03702586]

Eigenvectors:
 [[-0.69154646 -0.72136936  0.03727924]
 [-0.41590627  0.35545374 -0.83706309]
 [-0.59058062  0.59437268  0.54583482]]
```

Step 4 - Create Feature Vector

```
# Step 4: Sort the eigenvalues and corresponding eigenvectors
sorted_indices = np.argsort(eigenvalues)[::-1]
sorted_eigenvalues = eigenvalues[sorted_indices]
sorted_eigenvectors = eigenvectors[:, sorted_indices]

print("\nSorted eigenvalues:\n", sorted_eigenvalues)
```

```
Sorted eigenvalues:
 [1.41900369 1.03702586 0.54397045]

Top 2 eigenvectors:
 [[-0.72136936  0.03727924]
 [ 0.35545374 -0.83706309]
 [ 0.59437268  0.54583482]]
```

```
# Choose the top 2 eigenvectors
top_2_eigenvectors = sorted_eigenvectors[:, :2]

print("\nTop 2 eigenvectors:\n", top_2_eigenvectors)
```

```
Top 2 eigenvectors:
 [[-0.72136936  0.03727924]
 [ 0.35545374 -0.83706309]
 [ 0.59437268  0.54583482]]
```

Step 5: Recast the Data Along the Principal Components Axes

```
# Step 6: Transform the original dataset
X_reduced = np.dot(X_standardized, top_2_eigenvectors)

print("\nReduced dataset:\n", X_reduced)
```

```
Reduced dataset:
 [[-1.04315003  0.15454493]
 [ 0.51600533 -1.68896784]
 [-1.18256916  0.54997873]
 [ 1.70971385  0.98444418]]
```

Question 4b

Compute the percentage of variance explained by each principal component

```
explained_variance = sorted_eigenvalues / np.sum(sorted_eigenvalues) * 100

# Output the results
print("Standardized dataset:\n", X_standardized)
print("\nCovariance matrix:\n", cov_matrix)



print("\nPercentage of variance explained by each principal component:\n", explained_variance)
```

```
Standardized dataset:
 [[ 0.         -0.95618289 -1.18321596]
 [-0.3086067   1.67332005 -0.50709255]
 [ 1.5430335  -0.47809144  0.16903085]
 [-1.2344268  -0.23904572  1.52127766]]

Covariance matrix:
 [[ 1.         -0.23975611 -0.36514837]
 [-0.23975611  1.         -0.0404061 ]
 [-0.36514837 -0.0404061   1.        ]]

Percentage of variance explained by each principal component:
 [47.30012302 34.56752871 18.13234827]
```