SEAS 8510 - Analytical MEthods for ML

Homework 3

Due Date: April 13, 2024 (9:00am EST)

Your homework 2 should be submitted as a pdf containing your responses, your code, and your printed results where applicable.

All of the imports

In [ ]:
```
import pandas as pd
import numpy as np
from numpy.linalg import matrix_rank
import datetime as dt
import matplotlib.pyplot as plt
from sympy import *
%matplotlib inline
```

Functions

Reduce a matrix to row echelon form

In [ ]:
```
def row_echelon(A):
    """ Return Row Echelon Form of matrix A """

    # if matrix A has no columns or rows,
    # it is already in REF, so we return itself
    r, c = A.shape
    if r == 0 or c == 0:
        return A

    # we search for non-zero element in the first column
    for i in range(len(A)):
        if A[i,0] != 0:
            break
    else:
        # if all elements in the first column is zero,
        # we perform REF on matrix from second column
        B = row_echelon(A[:,1:])
        # and then add the first zero-column back
        return np.hstack([A[:,:1], B])

    # if non-zero element happens not in the first row,
    # we switch rows
    if i > 0:
        ith_row = A[i].copy()
        A[i] = A[0]
        A[0] = ith_row
    # we divide first row by first element in it
    A[0] = A[0] / A[0,0]
    # we subtract all subsequent rows with first row (it has 1 now as first element
    # multiplied by the corresponding element in the first column
    A[1:] -= A[0] * A[1:,0:1]

    # we perform REF on matrix from second row, from second column
    B = row_echelon(A[1:,1:])

    # we add first row and first (zero) column, and return
    return np.vstack([A[:1], np.hstack([A[1:,:1], B]) ])
```

Reduced Row Echelon Form

In [ ]:
```
def reduce_to_row_echelon_form(A):
  """
  This function reduces a matrix to reduced row echelon form using Gaussian Elimina

  Args:
      A: A square matrix represented as a nested list.

  Returns:
      A new matrix in reduced row echelon form (nested list), or None if the matrix
  """
  # Error handling for non-square matrices
  if not all(len(row) == len(A) for row in A):
    raise ValueError("Input matrix must be square.")

  # Copy the matrix to avoid modifying the original
  rows = len(A)
  cols = len(A[0])
  B = [[A[i][j] for j in range(cols)] for i in range(rows)]

  # Gaussian Elimination
  for i in range(rows):
    # Find a pivot row with a non-zero leading variable (or swap rows if necessary)
    pivot_row = i
    while B[pivot_row][i] == 0 and pivot_row < rows - 1:
      pivot_row += 1
    if pivot_row == rows - 1 and B[pivot_row][i] == 0:
      return None  # Matrix is singular

    # Swap pivot row with the current row if necessary
    if pivot_row != i:
      B[i], B[pivot_row] = B[pivot_row], B[i]

    # Eliminate elements below the leading variable in the current row
    for j in range(i + 1, rows):
      if B[j][i] != 0:
        factor = B[j][i] / B[i][i]
        for k in range(cols):
          B[j][k] -= factor * B[i][k]
  # Check for leading 1s in the diagonal (reduced row echelon form)
  for i in range(rows):
    if B[i][i] != 1:
      # Normalize the leading variable (avoiding division by zero)
      if B[i][i] != 0:
        factor = 1 / B[i][i]
        for k in range(cols):
          B[i][k] *= factor

  return B
```

Calculate Eigenvalues and Eigenvectors

In [ ]:
```
def calculate_eigenvalues_eigenvectors(A):
  """
  This function calculates eigenvalues and eigenvectors of a square matrix A.

  Args:
      A: A square matrix represented as a nested list.

  Returns:
      A tuple containing a list of eigenvalues and a list of corresponding eigenvec
  """

  # Error handling for non-square matrices
  if not all(len(row) == len(A) for row in A):
    raise ValueError("Input matrix must be square.")

  n = len(A)  # Matrix dimension

  # Initialize variables
  eigenvalues = []
  eigenvectors = []

  # Iterate for each eigenvalue
  for i in range(n):
    # Choose a pivot element (can be any element, but diagonal is often preferred)
    pivot = A[i][i]

    # Check if the matrix is 1x1 (trivial case)
    if n == 1:
      eigenvalues.append(pivot)
      eigenvectors.append([1])
      continue

    # Check for degeneracy (all elements in the column below pivot are zero)
    degenerate = True
    for j in range(i + 1, n):
      if A[j][i] != 0:
        degenerate = False
        break

    # Degenerate case: replicate the existing eigenvector
    if degenerate:
      if not eigenvalues:
        raise ValueError("Matrix is singular (no eigenvalues)")
      eigenvalues.append(eigenvalues[0])
      eigenvector_copy = eigenvectors[0].copy()
      eigenvectors.append(eigenvector_copy)
      continue

    # Reduce matrix to 2x2 sub-matrix around the pivot
    sub_matrix = [[A[i][i], A[i][i + 1]], [A[i + 1][i], A[i + 1][i + 1]]]

    # Solve the characteristic equation of the 2x2 sub-matrix
    det_sub = sub_matrix[0][0] * sub_matrix[1][1] - sub_matrix[0][1] * sub_matrix[1
    trace_sub = sub_matrix[0][0] + sub_matrix[1][1]
    lambda1 = (trace_sub + np.sqrt(trace_sub**2 - 4 * det_sub)) / 2
    lambda2 = (trace_sub - np.sqrt(trace_sub**2 - 4 * det_sub)) / 2

    # Store the eigenvalues
    eigenvalues.append(lambda1)
    if lambda1 != lambda2:  # Non-degenerate case, add another eigenvalue
      eigenvalues.append(lambda2)

    # Back-substitution to solve for eigenvectors
    for lambda_val in (lambda1, lambda2):  # Solve for each eigenvalue
      # Create a temporary vector for the eigenvector
      eigenvector = [0] * n
      eigenvector[i] = 1  # Set the pivot element to 1

      # Solve for other elements using back-substitution
      for j in range(i + 1, n):
        eigenvector[j] = (A[j][i] - lambda_val * eigenvector[i]) / (A[j][j] - lambd
      for j in range(i - 1, -1, -1):
        sum_term = 0
        for k in range(j + 1, n):
          sum_term += A[j][k] * eigenvector[k]
        eigenvector[j] = (A[j][i] - lambda_val * eigenvector[i] - sum_term) / (A[j]

      # Normalize the eigenvector
      norm = np.linalg.norm(eigenvector)  # You can use np.sqrt(sum(x^2 for x in ei
      if norm != 0:
        eigenvector = [x / norm for x in eigenvector]

      # Add the eigenvector to the result list
      eigenvectors.append(eigenvector)

  return eigenvalues, eigenvectors
```

2.10

Are the following sets of vectors linearly independent?

a.

$$x_1 = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix},\ x_2 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix},\ x_3 = \begin{bmatrix} 3 \\ -3 \\ 8 \end{bmatrix}$$

To find this out, we merge them into a matrix and convert it to row echelon form. If every column is a pivot column then the vectors are linearly independent.

```
In [ ]: A = np.array([[2, 1, 3],
                      [-1, 1, -3],
                      [3, -2, 8]])
        print("The vectors combined as A are:")
        print(A)
        print("A in row echelon format:")
        print(row_echelon(A))
        print(f"Rank of A is {matrix_rank(A)}")
```

```
The vectors combined as A are:
[[ 2  1  3]
 [-1  1 -3]
 [ 3 -2  8]]
A in row echelon format:
[[ 1  0  1]
 [ 0  1 -2]
 [ 0  0  1]]
Rank of A is 3
```

This result des not match my manual effort. So, I am thinking that my program is not correct and I will use the manual approach.

Manual steps.

$$\begin{bmatrix} 2 & 1 & 3 \\ -1 & 1 & -3 \\ 3 & -2 & 8 \end{bmatrix} \text{Add row 2 to row 1} \begin{bmatrix} 1 & 2 & 0 \\ -1 & 1 & -3 \\ 3 & -2 & 8 \end{bmatrix} \text{Add row 1 to row 2}$$
$$\begin{bmatrix} 1 & 2 & 0 \\ 0 & 3 & -3 \\ 3 & -2 & 8 \end{bmatrix} \text{Subtract three times row 1 from row 3} \begin{bmatrix} 1 & 2 & 0 \\ 0 & 3 & -3 \\ 0 & -8 & 8 \end{bmatrix} \text{Divide row 2 by 3}$$
$$\begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & -1 \\ 0 & -8 & 8 \end{bmatrix} \text{Add eight times row 2 to row 3} \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

Columns one and two are pivot columns but column three is not a pivot column. Therefore, these vectors are linearly dependant.

b.

$$x_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \; x_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \; x_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

To find this out, we merge them into a matrix and convert it to row echelon form. If all columns are pivot columns then the vectors are linearly independent. Otherwise, they are linearly dependent.

```
In [ ]: A = np.array([[1, 1, 1],
                      [2, 1, 0],
                      [1, 0, 0],
                      [0, 1, 1],
                      [0, 1, 1]])
        print("The vectors combined as A are:")
        print(A)
        print("A in row echelon format:")
        print(row_echelon(A))
        print(f"Rank of A is {matrix_rank(A)}")
```

```
The vectors combined as A are:
[[1 1 1]
 [2 1 0]
 [1 0 0]
 [0 1 1]
 [0 1 1]]
A in row echelon format:
[[1 1 1]
 [0 1 2]
 [0 0 1]
 [0 0 0]
 [0 0 0]]
Rank of A is 3
```

Manual steps.

$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ Subtract two times row one from row two. $\begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ Subtract row one from row three. $\begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & -1 & -1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ Add two times row three to row two $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & -1 & -1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ Add row two to row three $\begin{bmatrix}1&1&1\\0&1&0\\0&0&-1\\0&1&1\\0&1&1\end{bmatrix}$ Subtract row two from row four $\begin{bmatrix}1&1&1\\0&1&0\\0&0&-1\\0&0&1\\0&1&1\end{bmatrix}$ Subtract row two from row five $\begin{bmatrix}1&1&1\\0&1&0\\0&0&-1\\0&0&1\\0&0&1\end{bmatrix}$ Add row three to row four $\begin{bmatrix}1&1&1\\0&1&0\\0&0&-1\\0&0&0\\0&0&1\end{bmatrix}$ Add row three to row five $\begin{bmatrix}1&1&1\\0&1&0\\0&0&-1\\0&0&0\\0&0&0\end{bmatrix}$ Multiply row three by -1 $\begin{bmatrix}1&1&1\\0&1&0\\0&0&1\\0&0&0\\0&0&0\end{bmatrix}$ Each column is a pivot column therefore the vectors are independent.

2.11 Write

$y = \begin{bmatrix}1\\-2\\5\end{bmatrix}$ as a linear combination of $x_1 = \begin{bmatrix}1\\1\\1\end{bmatrix}, \ x_2 = \begin{bmatrix}1\\2\\2\end{bmatrix}, \ x_3 = \begin{bmatrix}2\\-1\\1\end{bmatrix}$

This means that we need to find scalers $m, \ n, \ o$ such that $y = mx_1 + nx_2 + ox_3$

Manual steps.

$\begin{bmatrix}1&1&2&1\\1&2&-1&-2\\1&2&1&5\end{bmatrix}$ Subtract row 1 from row 2 $\begin{bmatrix}1&1&2&1\\0&1&-3&-3\\1&2&1&5\end{bmatrix}$ Subtract row 1 from row 3 $\begin{bmatrix}1&1&2&1\\0&1&-3&-3\\0&1&-1&4\end{bmatrix}$ Subtract row 2 from row 1 $\begin{bmatrix}1&0&5&4\\0&1&-3&-3\\0&1&-1&4\end{bmatrix}$ Subtract row 2 from row 3 $\begin{bmatrix}1&0&5&4\\0&1&-3&-3\\0&0&2&7\end{bmatrix}$ Divide row 2 by 2 $\begin{bmatrix}1&0&5&4\\0&1&-3&-3\\0&0&1&\frac{7}{2}\end{bmatrix}$ Subtract 5 times the row 3 from row 1 $\begin{bmatrix}1&0&0&-\frac{27}{2}\\0&1&-3&-3\\0&0&1&\frac{7}{2}\end{bmatrix}$ Add 3 times row 3 to row 2 $\begin{bmatrix}1&0&0&-\frac{27}{2}\\0&1&0&\frac{15}{2}\\0&0&1&\frac{7}{2}\end{bmatrix}$ Therefore $m = -\frac{27}{2}, \ n = \frac{15}{2}, \ o = \frac{7}{2}$

4.3 Compute the eigenspaces of

a.

$A := \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$

First find all the eigenvalues by solving:

$det(A - \lambda In) = 0$

$det\left(\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = 0$

$det\left(\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right) = 0$

$det\left(\begin{bmatrix} 1-\lambda & 0 \\ 1 & 1-\lambda \end{bmatrix} \right) = 0$

$((1-\lambda)(1-\lambda)) - (1 \times 0) = 0$

Therefore 1 is the only eignevalue.

Now substitute the eigenvalues into $(A - \lambda In)$ to get the eigenvectors.

$\begin{bmatrix} 1-\lambda & 0 \\ 1 & 1-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$

For $\lambda = 1$

$\begin{bmatrix} 1-1 & 0 \\ 1 & 1-1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$

$\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$

For $\lambda_1 = 1$ the $E_1 = span \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

b.

$B := \begin{bmatrix} -2 & 2 \\ 2 & 1 \end{bmatrix}$

First find all the eigenvalues by solving:

$$det(A - \lambda In) = 0$$

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

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

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

$$((-2-\lambda)(1-\lambda)) - (2 \times 2) = 0$$

$$\lambda^2 + \lambda - 6 = 0$$

$$(\lambda + 3)(\lambda - 2) = 0$$

Therefore $\lambda_1 = -3$ and $\lambda_2 = 2$ are the eigenvalues.

Now substitute the eigenvalues into $(A - \lambda In)$ to get the eigenvectors.

$$\begin{bmatrix} -2-\lambda & 2 \\ 2 & 1-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

For $\lambda_1 = -3$

$$\begin{bmatrix} -2-(-3) & 2 \\ 2 & 1-(-3) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

For $\lambda_1 = -3$ the $E_{-3} = span \begin{bmatrix} -2 \\ 1 \end{bmatrix}$

For $\lambda_2 = 2$

$$\begin{bmatrix} -2-2 & 2 \\ 2 & 1-2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} -4 & 2 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

For $\lambda_2 = 2$ the $E_2 = span \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

4.4 Compute all eigenspaces of

$$A = \begin{bmatrix} 0 & -1 & 1 & 1 \\ -1 & 1 & -2 & 3 \\ 2 & -1 & 0 & 0 \end{bmatrix}$$

First find all the eigenvalues by solving:

$$det(A - \lambda In) = 0$$

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

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

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

```
In [ ]: # per your note, I decided to do this in Python
        A = np.array([[0,-1,1,1],
                      [-1,1,-2,3],
                      [2,-1,0,0],
                      [1,-1,1,0]])

        # Calculate the eigenvalues and eigenvectors
        eigvals, eigvecs = np.linalg.eig(A)

        # Print them out
        print(f'Eigenvalues = {eigvals}')
        print(f'Eigenvectors: \n{eigvecs}')
```

```
Eigenvalues = [ 2.          1.          -1.00000002 -0.99999998]
Eigenvectors:
[[ 5.77350269e-01  5.00000000e-01  1.58709825e-08  1.58709821e-08]
 [-1.44897623e-16  5.00000000e-01 -7.07106773e-01  7.07106789e-01]
 [ 5.77350269e-01  5.00000000e-01 -7.07106789e-01  7.07106773e-01]
 [ 5.77350269e-01  5.00000000e-01  3.20493788e-17 -6.19518495e-17]]
```

4.5 Diagonalizability of a matrices unrelated to its invertibility. Determine for each of the following four matrices whether they are diagonizable and/or invertible.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

This is a 2x2 identity matrix. Any identity matrix is invertible, it is its own inverse. It already meets the definition of a diagonizable matrix and so it is diagonizable.

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Test for invertibility by calculating the Determinant. $1 \times 0 - 0 \times 0 = 0 - 0 = 0$ therefore this matrix is not invertible.

Test for diagonizable by seeing if the eigenvectors have the same dimensionality as the original matrix.

First find all the eigenvalues by solving:

$$det(A - \lambda In) = 0$$

$$det\left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = 0$$

$$det\left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right) = 0$$

$$det\left(\begin{bmatrix} 1-\lambda & 0 \\ 0 & 0-\lambda \end{bmatrix} \right) = 0$$

$$((1-\lambda)(0-\lambda)) - (0 \times 0) = 0$$

$$\lambda^2 - \lambda - 0 = 0$$

$$(\lambda)(\lambda - 1) = 0$$

Therefore $\lambda_1 = 0$ and $\lambda_2 = 1$ are the eigenvalues.

$$\begin{bmatrix} 1-\lambda & 0 \\ 0 & 0-\lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

For $\lambda_1 = 0$

$$\begin{bmatrix} 1-0 & 0 \\ 0 & 0-0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

For $\lambda_1 = 0$ the $E_0 = span \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

For $\lambda_2 = 1$

$$\begin{bmatrix} 1-1 & 0 \\ 0 & 0-1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

For $\lambda_2 = 1$ the $E_1 = span \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Since the eigenvectors have the same dimensionality as the original matrix, this matrix is diangonalizable.

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Test for invertibility by calculating the Determinant. $1 \times 1 - 0 \times 1 = 1 - 0 = 1$ the determinant is non zero, therefore this matrix is invertible.

Test for diagonizable by seeing if the eigenvectors have the same dimensionality as the original matrix.

First find all the eigenvalues by solving:

$$det(A - \lambda In) = 0$$

$$det\left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = 0$$

$$det\left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right) = 0$$

$$det\left(\begin{bmatrix} 1-\lambda & 1 \\ 0 & 1-\lambda \end{bmatrix} \right) = 0$$

$$((1-\lambda)(1-\lambda)) - (0 \times 1) = 0$$

$$\lambda^2 - 2\lambda - 1 - 0 = 0$$

$$(\lambda - 1)(\lambda - 1) = 0$$

Therefore $\lambda_1 = 1$ with a multiplicity of 2 is the eigenvalue.

Since there is only one eigenvalue the eigenvectors cannot have the same dimensionality as the original matrix, this matrix is not diangonalizable.

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Test for invertibility by calculating the Determinant. $0 \times 0 - 0 \times 1 = 0 - 0 = 0$ therefore this matrix is not invertible.

Test for diagonizable by seeing if the eigenvectors have the same dimensionality as the original matrix.

First find all the eigenvalues by solving:

$$det(A - \lambda In) = 0$$

$$det\left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 0$$

$$det\left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right) = 0$$

$$det\left(\begin{bmatrix} 0 - \lambda & 1 \\ 0 & 0 - \lambda \end{bmatrix}\right) = 0$$

$$((0 - \lambda)(0 - \lambda)) - (0 \times 1) = 0$$

$$\lambda^2 - 0\lambda - 0 = 0$$

$$(\lambda)(\lambda) = 0$$

Therefore $\lambda_1 = 0$ with a multiplicity of 1 is the only eigenvalue.

Since there is only one eigenvalue the eigenvectors cannot have the same dimensionality as the original matrix, this matrix is not diangonalizable.

4.6 Compute the eigenspaces of the following transformation matrices. Are they diagonalizable?

a. For

$$A = \begin{bmatrix} 2 & 3 & 0 \\ 1 & 4 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

First find all the eigenvalues by solving:

$$det(A - \lambda In) = 0$$

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

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

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

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

$$(2-\lambda)\;((4-\lambda)(1-\lambda)-(0\times 3))\;-3\;((1\times 1-\lambda)-(0\times 3))=0$$

$$\lambda_1=5$$

$$\lambda_2=1$$

For $\lambda_1=5$ the $E_5=span\begin{bmatrix}1\\1\\0\end{bmatrix}$

For $\lambda_2=1$ the $E_1=span\begin{bmatrix}-3\\1\\0\end{bmatrix}$

Because there are only two Eigenvectors, they do not have the same dimensionality as the original matrix and this matrix is not diagonizable.

b. For

$$B=\begin{bmatrix}1&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$$

First find all the eigenvalues by solving:

$$det(A-\lambda In)=0$$

$$det\left(\begin{bmatrix}1&1&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}-\lambda\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\right)=0$$

$$det\left(\begin{bmatrix}1 & 1 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix} - \begin{bmatrix}\lambda & 0 & 0 & 0\\0 & \lambda & 0 & 0\\0 & 0 & \lambda & 0\\0 & 0 & 0 & \lambda\end{bmatrix}\right) = 0$$

Remainder TBD

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

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

$$(2-\lambda)\left((4-\lambda)(1-\lambda)-(0\times 3)\right) -3\left((1\times 1-\lambda)-(0\times 3)\right) = 0$$

$\lambda_1 = 5$

$\lambda_2 = 1$

For $\lambda_1 = 5$ the $E_5 = span\begin{bmatrix}1\\1\\0\end{bmatrix}$

For $\lambda_2 = 1$ the $E_1 = span\begin{bmatrix}-3\\1\\0\end{bmatrix}$

Because there are only two Eigenvectors, they do not have the same dimensionality as the original matrix and this matrix is not diagonizable.