Machine Learning Diploma

Session 1 : Linear Algebra

Agenda:

1	Introduction to Linear Algebra
2	Vectors
3	Matrices
4	Tensors
5	Linear Algebra in Practice (Numpy)
6	Applications of Linear Algebra

1. Introduction To Linear Algebra

What is Linear Algebra?

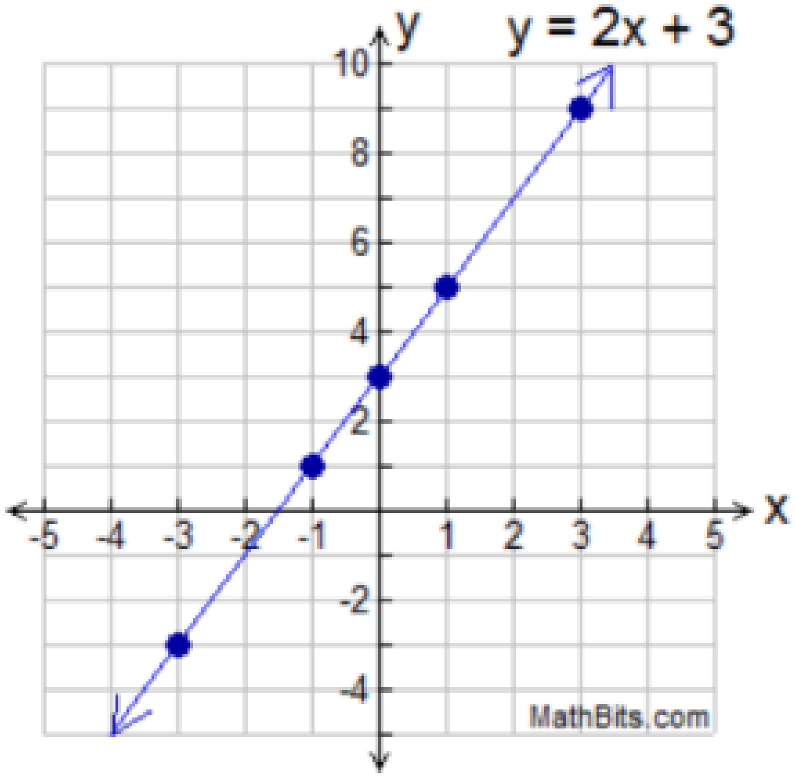
- Linear Algebra is a about representing the mathematical **linear equations** in a form that makes calculations much faster and efficient.
- So, let’s first talk about linear equations.

Linear Equations:

- A linear equation is a series of mathematical operations over input terms to calculate a value of output term.
- For example, expression $y = 4*x_1 + 7*x_2 + 2*x_3 + 5$, is called a linear equation.
- x_1, x_2 , & x_3 are called input terms or independant variables, while y is called dependant variable.
- 4, 7, & 2 are called parameters or weights, while 5 is called the bias.

Linear Equation Example:

x	$y = 2x + 3$	y
-3	2(-3) + 3	-3
-1	2(-1) + 3	1
0	2(0) + 3	3
1	2(1) + 3	5
3	2(3) + 3	9



Why do we call it a Linear Equation?

- The reason why we call it a linear equation is that; every term in the equation has a degree of one, which means that every term must have a power of 1.

- For example:
  - x_1^2 is not linear, because the term x_1 has power of 2.
  - x_1*x_2 is not linear, because the term (x_1*x_2) has power of 2.
  - x_1 is linear, because the term x_1 has power of 1.

Linear Equations:

- So, we can define the linear equation as a series of terms added together, where each term is a single variable multiplied by a constant number called parameter or coefficient.

- One term must consist of one variable, which means this variable cannot be multiplies by another variable; neither itself nor another variable.

Task 1: Which of the following is linear?

Equations
$y = 8x - 9$
$y = x^2 - 7$
$\sqrt{y} + x = 6$
$y + 3x - 1 = 0$
$y^2 - x = 9$

Task 1: Solution:

Equations	Linear or Non-Linear
$y = 8x - 9$	Linear
$y = x^2 - 7$	Non-Linear
$\sqrt{y} + x = 6$	Non-Linear
$y + 3x - 1 = 0$	Linear
$y^2 - x = 9$	Non-Linear

2. Vectors

What Are Vectors?

- A vector is way to represent many values in shorter terms.
- For example, $y = 4*x_1 + 7*x_2 + 2*x_3 + 5$, is an equation that has three variables x_1, x_2 , & x_3 , we can represent these 3 variable with a single vector x, where $x = [x_1, x_2, x_3]$.
- We call x a vector which made it easier for us to represent the long mathematical terms, in a shorter term.

Vectors In real life:

- Vectors can be used to represent objects in real life.

- For example, if we have a human(object) with height=180cm, weight=90kg, & age=40y, we can represent this object(human) as a single vector v, where v=[180, 90, 40].

Vector Operations:

- Many operations can be carried out over the vectors to form new vectors.
- Below are the fundamental mathematical operations of vectors:
  - Vector addition.
  - Vector subtraction.
  - Vector elementwise Multiplication.
  - Vector/Scaler Multiplication.
  - Dot Product Multiplication.

Vector Operations:

1. Vector Addition:
   - Adding two vectors together to get a new vector as a result.
   - Example:
     - $v_1 = [1, 2]$
     - $v_2 = [2, 1]$
     - $v_3 = v_1 + v_2$
     - $v_3 = [1, 2] + [2, 1] = [1 + 2, 2 + 1] = [3, 3]$.

Vector Operations:

2. Vector Subtraction:
   - Subtract a vector from another to get a new vector as a result.
   - Example:
     - $v_1 = [4, 6]$
     - $v_2 = [5, 3]$
     - $v_3 = v_1 - v_2$
     - $v_3 = [4, 6] - [5, 3] = [4 - 5, 6 - 3] = [-1, 3]$.

Vector Operations:

3. Vector/Scaler Multiplication:
   - Multiply each element in a vector by a scaler to get a new vector as a result.
   - Example:
     - $v_1 = [4, 6]$
     - $s = 2$
     - $v_2 = s * v_1$
     - $v_2 = 2 * [4, 6] = [2 * 4, 2 * 6] = [8, 12]$.

Vector Operations:

4. Vector elementwise Multiplication :
   - Multiply each element in a vector by its corresponding element in the other vector to get a new vector as a result.
   - Example:
     - $v_1 = [4, 6]$ $v_2 = [5, 3]$
     - $v_3 = v_1 * v_2$
     - $v_3 = [4, 6] * [5, 3] = [4 * 5, 6 * 3] = [20, 18]$.

Vector Operations:

5. Dot Product Multiplication :
   - Multiply each element in a vector by its corresponding element in the other vector, then sum all values to get a scalar as a result.
   - Example:
     - $v_1 = [1,2,3]$ $v_2 = [3,4,5]$
     - $v_3 = v_1 . v_2$
     - $v_3 = [1,2,3] . [3,4,5] = 1*3 + 2*4 + 3*5 = 27.$

Vector Operations:

- We will apply these operations in practice later in this chapter, using a library called Numpy.

Why are Vector Operations useful?

- The reason why vector operations are useful is that we can use vector operations to represent a linear equation in simple terms.
- For example, $y = 4*x_1 + 7*x_2 + 2*x_3 + 5$, is a linear equation represented by 4 terms, but using vectors we can represent it in shorter terms, as below.

Why are Vector Operations useful?

- Where, input variables $x_1, x_2, \& x_3$ can be represented by vector x, where $x = [x_1, x_2, x_3]$.
- While; parameters 4, 7, & 2 can be represented by vector a, where a=[4, 7, 2].
- Finally; 5 is a scaler which can be represented by b, where b is the bias.

Why are Vector Operations useful?

- Having all of that in mind, we can represent the linear equation to be $y = a \, . \, x + b$.
- As you see, we applied dot-product-multiplication between a & x, then added the result to b.
- Now the linear equation is represented by only two terms instead of 4, which makes things easier while applying more complex mathematical calculations.

3. Matrices

What are Matrices?

- A matrix is a vector of vectors.
- For example:
  - v1 = [4, 5, 3]
  - v2 = [1, 6, 2]
  - M = [v1, v2] = [[4, 5, 3]
    [1, 6, 2]]

What are Matrices?

- Each row in the matrix is a vector

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \\ 7 & 2 & 3 \end{bmatrix}$$

- Each column in the matrix is also a vector

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \\ 7 & 2 & 3 \end{bmatrix}$$

Matrix Size

- Matrix size is defined by its number of rows & columns.

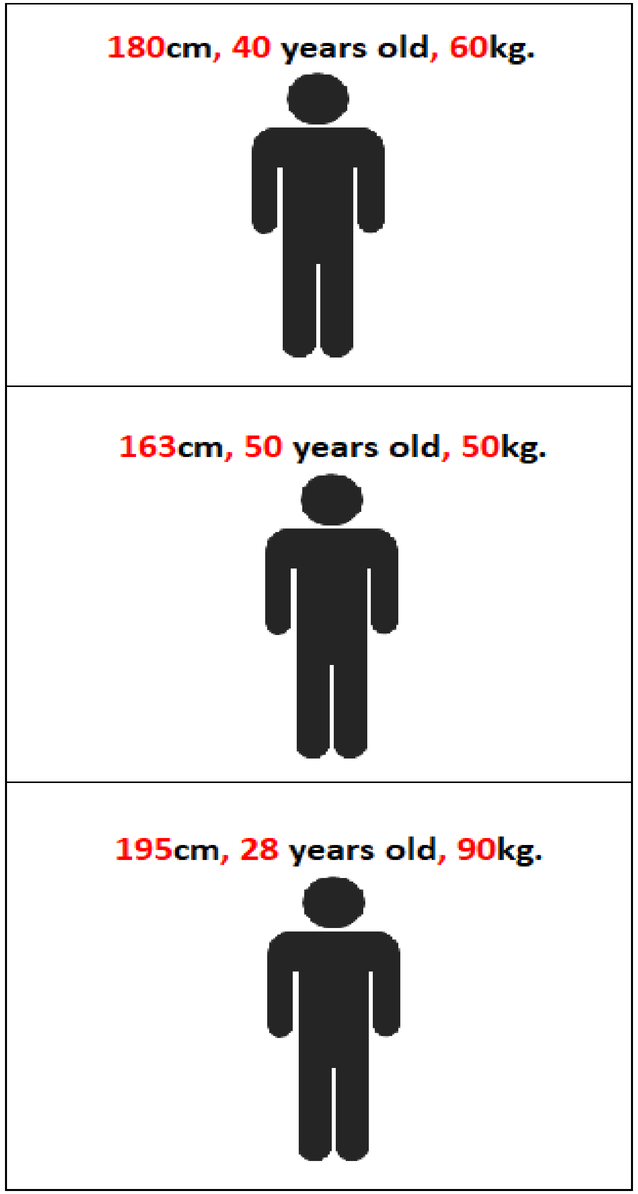
- Size = N X M
  - N is rows number.
  - M is columns number.

$\begin{bmatrix}1 & 2 & 3\\ 2 & 5 & 6\\ 7 & 2 & 3\end{bmatrix}$ 3X3 Matrix	$\begin{bmatrix}1 & 3\\ 2 & 6\\ 7 & 3\end{bmatrix}$ 3X2 Matrix	$\begin{bmatrix}1 & 2 & 3\\ 7 & 2 & 3\end{bmatrix}$ 2X3 Matrix
$\begin{bmatrix}1\\ 6\\ 7\end{bmatrix}$ 3X1 Matrix	$\begin{bmatrix}1 & 7 & 3\end{bmatrix}$ 1X3 Matrix	$\begin{bmatrix}1 & 3\\ 7 & 3\end{bmatrix}$ 2X2 Matrix

Why Matrices?

- Suppose we have three persons(3 objects).
- We can represent these 3 objects using:

3 Vectors		1 Matrix
- Person1 = [180, 40, 60] - Person2 = [163, 50, 50] - Person3 = [195, 28, 90]	OR	- Persons = [[180, 40, 60] [163, 50, 50] [195, 28, 90]]



Why Matrices?

- As you saw from the previous example, vectors are not good enough when it comes to representing large number of objects.
- Later in machine learning, we will have huge number of objects, thousands & sometimes millions, and we will represent them as a single matrix, instead of thousands of vectors.

Matrix Operations:

- Below are the fundamental mathematical operations that can be carried over matrices:
  - Matrix Addition.
  - Matrix Subtraction.
  - Matrix-Scalar Multiplication.
  - Matrix-Elementwise Multiplication.
  - Matrix Dot-product Multiplication.

Matrix Operations:

1. Matrix Addition:

- Add every element in the 1^{st} matrix to its corresponding element in the 2^{nd} matrix.
- Example:

$$\begin{bmatrix} 8 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix} + \begin{bmatrix} 4 & 6 & 6 \\ 4 & 2 & 6 \end{bmatrix} = \begin{bmatrix} 12 & 9 & 8 \\ 5 & 4 & 7 \end{bmatrix}$$

Matrix Operations:

2. Matrix Subtraction:

- Subtract every element in the 1st matrix from its corresponding element in the 2nd matrix.
- Example:

$$\begin{bmatrix} 8 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix} - \begin{bmatrix} 4 & 6 & 6 \\ 4 & 2 & 6 \end{bmatrix} = \begin{bmatrix} 4 & -3 & -4 \\ -3 & 0 & -5 \end{bmatrix}$$

Matrix Operations:

3. Matrix-Scalar Multiplication:

- Multiply all elements in the matrix by a scalar.
- Example:

$$\begin{bmatrix} 8 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix} * 2 = \begin{bmatrix} 16 & 6 & 4 \\ 2 & 4 & 2 \end{bmatrix}$$

Matrix Operations:

4. Matrix-Elementwise Multiplication:

- Multiply each element in the 1st matrix by its corresponding element in the 2nd matrix.

- Example:

$$\begin{bmatrix} 8 & \boxed{3} & 2 \\ \boxed{1} & 2 & 1 \end{bmatrix} * \begin{bmatrix} .5 & \boxed{2} & 1 \\ \boxed{3} & .5 & 4 \end{bmatrix} = \begin{bmatrix} 4 & \boxed{6} & 2 \\ \boxed{3} & 1 & 4 \end{bmatrix}$$

Matrix Operations:

5. Matrix Dot-product Multiplication:

- Apply Dot-product multiplication between each row in the 1^{st} matrix & each column in the 2^{nd} matrix.
- Number of columns in the 1^{st} matrix must equal Number of rows in the 2^{nd} matrix.

Matrix Operations:

5. Matrix Dot-product Multiplication:

- Example:

$$\begin{bmatrix}1 & 2 & 3\\ 7 & 2 & 3\end{bmatrix} \cdot \begin{bmatrix}1 & 2 & 3\\ 2 & 5 & 6\\ 7 & 2 & 3\end{bmatrix} = \begin{bmatrix}26 & 18 & 24\\ 32 & 30 & 42\end{bmatrix}$$

$$\underbrace{2 \text{ X } \boxed{3 \qquad\qquad 3} \text{ X } 3}_{\text{New matrix = 2 X 3}} \qquad\qquad 2 \text{ X } 3$$

Why are Matrix Operations Useful?

- Matrix Operations are useful to represent a System of linear equations.
- System of linear equations is simply a set of linear equations, for example:
  - $y_1 = 4*x_1 + 7*x_2 + 2*x_3 + 5$
  - $y_2 = 3*x_1 + 9*x_2 + 4*x_3 + 2$
  - $y_3 = 8*x_1 + 2*x_2 + 3*x_3 + 1$

Why are Matrix Operations Useful?

- Parameters of all equations in the previous example can be represented with a single 3X3 matrix **A**.

$$A = \begin{bmatrix} 4 & 7 & 2 \\ 3 & 9 & 4 \\ 8 & 2 & 3 \end{bmatrix}$$

- The 1^{st} row in matrix **A** represents parameters of the 1^{st} linear equation in the system, while the 2^{nd} row represents the 2^{nd} equation, and so on.

- The input variable can also be represented with a 3X1 matrix **X**.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

- Biases of all equations can be represented with one vector **b**.

$$b = \begin{bmatrix} 5 \\ 2 \\ 1 \end{bmatrix}$$

Why are Matrix Operations Useful?

- We can represent the whole system using simple terms, which makes things easier to represent & faster to calculate.

$$
A \,.\, X + b = \underbrace{\overset{A}{\begin{bmatrix} 4 & 7 & 2 \\ 3 & 9 & 4 \\ 8 & 2 & 3 \end{bmatrix}}}_{\substack{3 \times 3 \\ \text{Matrix}}} \cdot \underbrace{\overset{X}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}}_{\substack{3 \times 1 \\ \text{Matrix}}} + \underbrace{\overset{b}{\begin{bmatrix} 5 \\ 2 \\ 1 \end{bmatrix}}}_{\text{vector}}
$$

Matrix Types?

Square Matrix	Rectangular Matrix	Symmetric Matrix
➢ Is a matrix where the number of rows N is equivalent to the number of columns M. $$\begin{bmatrix} 1 & 4 & 3 \\ 3 & 9 & 1 \\ 8 & 2 & 7 \end{bmatrix}$$	➢ Is a matrix where the number of rows N and columns M are not equal. $$\begin{bmatrix} 5 & 4 & 5 & 5 & 2 \\ 8 & 2 & 9 & 1 & 0 \\ 1 & 0 & 1 & 6 & 4 \end{bmatrix}$$	➢ Is a square matrix where the top-right triangle is the same as the bottom-left triangle. $$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 2 & 3 \\ 4 & 3 & 2 & 1 & 2 \\ 5 & 4 & 3 & 2 & 1 \end{bmatrix}$$

Matrix Types?

Triangular Matrix	Diagonal Matrix
➢ Is a square matrix that has non-zero values in the upper-right triangle or the lower-left triangle of the matrix, while the remaining elements are filled with zero values. Upper Triangular $\begin{bmatrix} 1 & 2 & 3 \\ 0 & 2 & 3 \\ 0 & 0 & 3 \end{bmatrix}$ Lower Triangular $\begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 2 & 3 \end{bmatrix}$	➢ Is a matrix where values outside the main diagonal have a zero value, while only the main diagonal has non-zero values. ➢ Doesn’t have to be square matrix $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$ $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Matrix Types?

Identity Matrix

- Is a square matrix where all values along the main diagonal have a value of 1, while all other values are zero.

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

4. Tensors

What are Tensors?

- A tensor is a general concept in linear algebra of scalars, vectors, and matrices.
- For example:
  - A scalar: is a 0-dimensional Tensor.
  - A vector: is a 1-dimensional Tensor.
  - A matrix: is a 2-dimensional Tensor.
  - An Image: is a 3-dimensional Tensor.
  - A video : is a 4-dimensional Tensor

5. Linear Algebra in Practice(Numpy)

How to apply linear algebra in practice?

- So far, we have talked about linear algebra concepts; such as, vectors, matrices, and operations over them.
- To apply these concepts, we will use a library called Numpy.
- In this session, we will introduce Numpy and study it in detail later, in the next session.

What is Numpy?

- Numpy is a library that allows us to create vectors, and matrices by creating Numpy arrays; which is the main data type in Numpy.

- To use Numpy, first we need to import it:

```
import numpy as np
```

Vectors using Numpy?

- This is how you create a Numpy vector:

```python
v1 = np.array([10, 6, 7])
v2 = np.array([3, 5, 12])
print(v1)
print(v2)
```

```
[10  6  7]
[ 3  5 12]
```

Vectors using Numpy?

➢ Vector operations:

<table>
<tr>
<td>

Vector addition

```
1  v1 + v2
```

```
array([13, 11, 19])
```

</td>
<td>

Vector Subtraction

```
1  v1 - v2
```

```
array([ 7,  1, -5])
```

</td>
<td>

Vector elementwise Multiplication

```
1  v1 * v2
```

```
array([30, 30, 84])
```

</td>
</tr>
<tr>
<td></td>
<td>

Vector/Scaler Multiplication

```
1  v1 * 5
```

```
array([50, 30, 35])
```

</td>
<td>

Dot Product Multiplication

```
1  v1.dot(v2)
```

```
144
```

</td>
</tr>
</table>

Matrices using Numpy?

➢ This is how you create a Numpy matrix:

```
M1 = np.array([[1, 2, 3],
               [4, 5, 6],
               [7, 8, 9]])

M2 = np.array([[0, 2, 3],
               [8, 2, 1],
               [3, 8, 5]])
print(M1)
print("============")
print(M2)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
============
[[0 2 3]
 [8 2 1]
 [3 8 5]]
```

Matrices using Numpy?

- Matrix operations:

Matrix Addition

```
1  M1 + M2
```

```
array([[ 1,  4,  6],
       [12,  7,  7],
       [10, 16, 14]])
```

Matrix Subtraction

```
1  M1 - M2
```

```
array([[ 1,  0,  0],
       [-4,  3,  5],
       [ 4,  0,  4]])
```

Matrix-Scalar Multiplication

```
1  M1 * 5
```

```
array([[ 5, 10, 15],
       [20, 25, 30],
       [35, 40, 45]])
```

Matrix-Elementwise Multiplication

```
1  M1 * M2
```

```
array([[ 0,  4,  9],
       [32, 10,  6],
       [21, 64, 45]])
```

Matrix Dot-product Multiplication

```
1  M1.dot(M2)
```

```
array([[ 25,  30,  20],
       [ 58,  66,  47],
       [ 91, 102,  74]])
```

6. Applications of Linear Algebra

1. Datasets:

- Dataset is a matrix, that carries data or information about set of objects.

- Each row in the Dataset is called sample, record, or example.
- Each column in the Dataset is called attribute or feature.

Dataset Example:

- This dataset is called houses dataset.
- Each row is a vector, that represents information about one sample, or one object (one house).
- Each column is a vector, that represents one feature, for example, the Length of all houses is a feature, while the city of all houses is another feature, and so on.

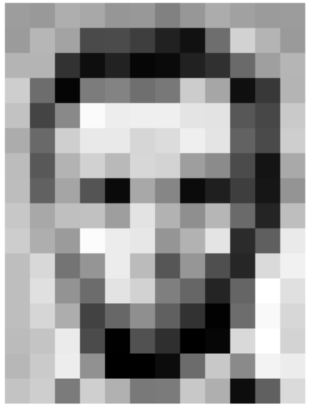
Length	Width	City	Price
20	10	Cairo	5000000
15	15	Alex	4000000
30	20	Aswan	1500000
10	50	Alex	8000000
5	15	Giza	800000
12	10	Cairo	1000000
5	30	Luxor	500000
7	20	Aswan	700000
20	40	Alex	9000000
8	20	Cairo	900000
6	14	Giza	6000000

2. Images:

- An Image is a matrix of numbers, where each cell in this matrix represents a degree of the color.
- Each cell in the image matrix is called a pixel, and it carries a number which is translated into a color displayed on the computer screen.

Image Matrix Examples:

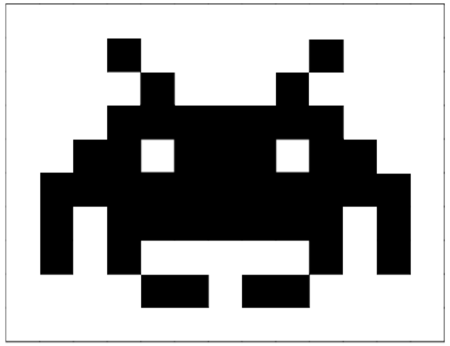
Example 1



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Example 2



1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	0	1	1	1
1	1	1	1	0	1	1	1	0	1	1	1	1
1	1	1	0	0	0	0	0	0	0	1	1	1
1	1	0	0	1	0	0	0	1	0	0	1	1
1	0	0	0	0	0	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	0	0	1	0	1
1	0	1	0	1	1	1	1	1	0	1	0	1
1	1	1	1	0	0	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1

1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	0	1	1	1
1	1	1	1	0	1	1	1	0	1	1	1	1
1	1	1	0	0	0	0	0	0	0	1	1	1
1	1	0	0	1	0	0	0	1	0	0	1	1
1	0	0	0	0	0	0	0	0	0	0	0	1
1	0	1	0	0	0	0	0	0	0	1	0	1
1	0	1	0	1	1	1	1	1	0	1	0	1
1	1	1	1	0	0	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1

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

3. Data processing:

- Data processing is applying operations over the dataset, to make it clean and consistent.
- We will talk more about data processing in detail in coming sessions.

3. Machine Learning:

- Linear algebra is important for Machine Learning , because machine learning is about learning from data.

- Datasets are represented as matrices, so we will need to use linear algebra to represent data and apply operations over it.

Thank You