To understand machine learning in depth, let us understand a few topics related to linear algebra. You do not need to necessarily understand linear algebra to get started with machine learning. However, it will be useful for you when you want to understand the ML algorithms in depth.

In Linear Algebra, data is represented in a linear format, in turn using matrices and vectors.

Matrix and Vectors:

Matrices (plural of matrix) are used throughout machine learning algorithms, specifically looking at input variables i.e. the variables we try to understand in machine learning.

A matrix is a two-dimensional array that m rows and n columns. A vector that has only one row or one column is called a vector.

e.g. X below is a matrix while Y is a vector, also c/a column vector.

$$X = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \qquad Y = \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix}$$

Multiplying a matrix with a vector:

To multiply a row vector with a column vector, the row vector must have as many columns as the column vector has rows.

Let us look at an example of multiplying a matrix with a vector, where A is a m*n matrix and X is a n*1 column vector.

e.g. $X = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$ **and A =** $\begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{4} & \mathbf{5} & \mathbf{6} \\ 7 & 8 & 9 \end{pmatrix}$

If we need to find AX, by definition, the number of columns in A is equal to number of rows in X.

Hence, $AX = \begin{pmatrix} 13 \\ 31 \\ 49 \end{pmatrix}$

*We first multiply row 1 of the matrix with col of the vector i.e., (1*2 + 2*1 + 3*3) = 13*

Multiplying a matrix with another matrix:

Multiplication of a matrix with another matrix is called as a dot product. To be able to multiply two matrices, the number of columns in the first matrix should be equal to the number of rows in the second matrix.

e.g. $$\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix} \quad \mathbf{X} \quad \begin{pmatrix}7 & 10\\ 8 & 11\\ 9 & 12\end{pmatrix} \quad = \quad \begin{pmatrix}50 & 68\\ 122 & 167\end{pmatrix}$$

The resultant matrix would have number of rows same as the rows of the first matrix and number of columns same as the columns of the second matrix.

*(first no. would be: (1*7) + (2*8) + (3*9) = 50, the fourth would be (4*10) + (5*11) + (6*12) = 167)*

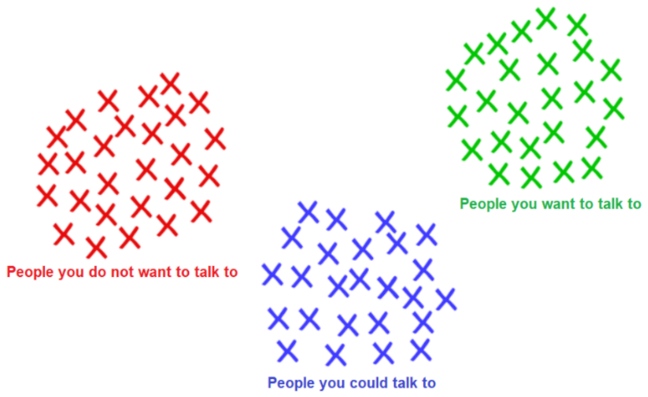
Let us now try to understand **un-supervised learning** using a simple example.

A friend throws a birthday party at a resort and invites you to be a part of it. Once you go there, you realize that there are all unfamiliar faces i.e., you do not know anyone at the party. You decide to categorize people into three buckets:

- People you want to talk to.
- People you could talk to.
- People you do not want to talk to.

Since you have no prior knowledge / information about these people, you would do the categorization based on different features such as their age group, gender, dressing sense, smartness, area of interest (e.g. are they playing games, chatting in a group, singing and dancing, enjoying drinks etc.).

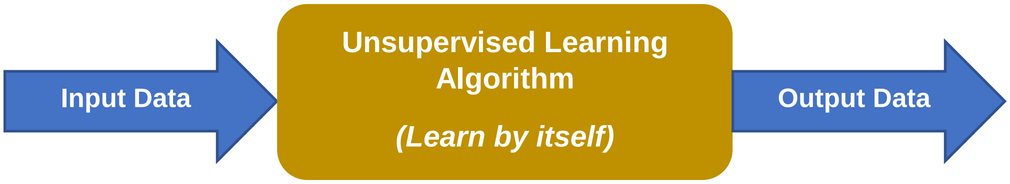
This is **un-supervised learning.** You did not have any prior information about the people, and you classified them on the go.



Machine learning (data science) techniques can be divided into three categories:

- **Supervised Learning:** Algorithms have prior data (also c/a labeled data) to learn from
- **Unsupervised Learning:** Algorithms have no prior data to learn from i.e., they learn from unlabeled data by finding patterns and grouping them accordingly.
- **Reinforcement Learning:** Algorithms learn patterns or behaviors in a re-iterative manner by correcting themselves over and over again.

In this module, our focus will be on unsupervised learning.



Unsupervised Learning is a category of techniques that trains machines (computers) to use a set of unlabeled / unseen data and learn by itself. The machines are provided with a large volume of data and they are expected to identify hidden patterns. Since there is no prior information available, there is no defined outcome. The machines only need to determine anything that they find interesting in the given data.

Methods under unsupervised learning:

Unsupervised learning methods can further be divided into two categories:

- Clustering
- Dimension-reduction

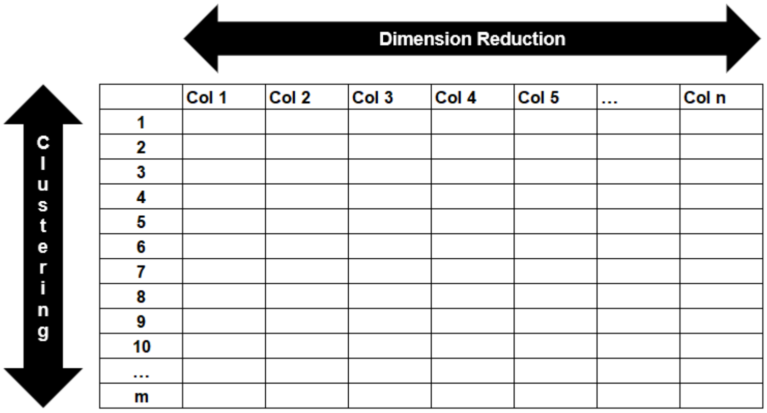
1. **Clustering** is a technique to create groups of data points in a way that:

- Points in the same cluster are as similar to each other as possible.
- Points in different clusters are as dissimilar to each other as possible.

For example, the creating groups of people at a party as we saw above.

Another example of clustering is what we often see on the e-commerce websites on a day-to-day basis. If you go to the women's sections, you will see there are clusters of products like apparels, jewelry, footwear, cosmetics etc. Further, each of these clusters is divided into sub-clusters. For instance, apparels would be divided into office wear, casual wear, ethnic, nightwear etc.

2. Dimension reduction is a technique to reduce the dimensions (also c/a features or columns) of data by choosing vectors that we also call as principal components. It is used when the number of dimensions in a given dataset are too high. The idea is to reduce the size of the data while still preserving as much originality as possible.



	Col 1	Col 2	Col 3	Col 4	Col 5	...	Col n
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
...							
m							

Using distances in Machine Learning:

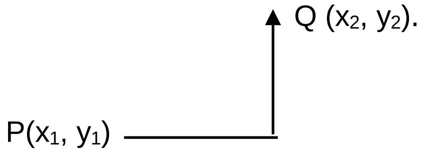
In clustering, grouping of observations into groups requires some methods for computing the distance or similarity / dissimilarity between each pair of observations. This is done using distance matrix and there are many methods to calculate this distance.

Euclidean distance: Euclidean distance is calculated as the square root of the sum of the squared differences b/w two vectors.

Say there are two points P (x_1, y_1) and Q (x_2, y_2). The euclidean distance b/w these two points would be calculated as:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Manhattan distance: Also called as city block distance, it calculates the distance b/w two points like on a uniform gird



Scaling:

Input variables in a dataset may have different units e.g. kilometers, hours, kilograms etc. i.e. different scales. This difference in scales increases the difficultly in modeling, in turn resulting in an unstable model. Unless you scale the data, the importance of 1 km would be same as 1 kg, would be same as 1 hr, would be same as 1 cm etc.

In other words, while 1000gms and 1kg mean the same thing, a quantity of 1000 ML is at a different scale. Unless both the weight (i.e. gms) and the volume (i.e. ml) are in the same scale, the ML algorithms might give one of them a slightly more weightage than the other. That in turn, puts down the effect of the variable with the lower scale.

Thus, scaling the variables is an important step in machine learning models. By scaling the variables, we can compare the variables on an equal level.

Normalization is one of the ways of scaling data so that all the values lie b/w 0 and 1. A value can be normalized as following:

$$y = (x - \text{min}) / (\text{max} - \text{min})$$

where,

- y: normalized variable
- x: variable of interest
- min: minimum value of the variable x
- max: maximum value of the variable y

Standardization is another way of scaling data so that the mean of observations is 0 and standard deviation is 1. A value can be standardized as follows:

$$y = (x - \text{mean}) / \text{std\_dev}$$

where,

- y: standardized variable
- x: variable of interest
- mean: average of the variable x
- std_dev: standard deviation of the variable x

Variance v/s Covariance:

Variance and covariance are mathematical terms that help us understand spread of the data. While variance calculates how far is an observation from the mean, covariance calculates the how two variables are related to each other. Larger the variance, the more spread is the data.

Variance:

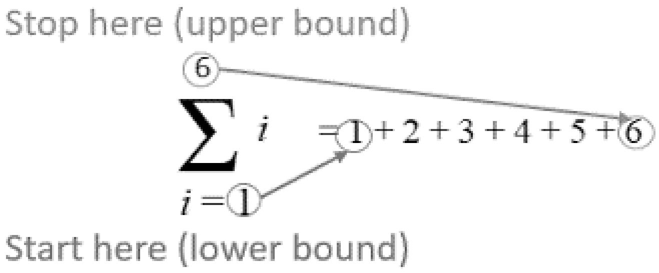
$$s^2 = \frac{\Sigma\,(x - \bar{x}\,)^2}{n - 1}$$

Covariance:

$$Cov_{xy} = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{(n-1)}$$

where,

- x: variable of interest
- x̄: Mean of the variable x
- n: Sample size i.e. the number of observations
- Σ: Summation



In case of co-variance, since we use two variables, we can understand them the same way as for variance.

While **probability** refers to the likelihood of an event occurring, **probability distribution** looks at each possible outcome of a random variable and their corresponding probabilities.

For instance, if a factory has three machines, the probability of machines failing will be measured in terms of probability distribution i.e. P(0 failing), P(1 failing), P(2 failing) and P(3 failing).

Probability is simply the likelihood of something happening.

$$\mathit{Probability}(A) = \frac{\#\ \mathit{ways\ A\ can\ happen}}{\mathit{Total\ number\ of\ outcomes}}$$

For instance, probability of getting heads on flipping a coin would be simply 1/ 2 = 0.5

- Remember, probability of an event can only lie between 0 and 1.
- Also, the sum of probability of all events in a random space will be equal to 1

i.e., P(head) + P(tail) = 1

Finally, when we use these graphs to represent probability distributions of a random variable, they become random graphs. In other words, the intersection of graphs and probability distribution are random graphs.

Let us explore more in the upcoming videos. Happy Learning!