# KNN Algorithm

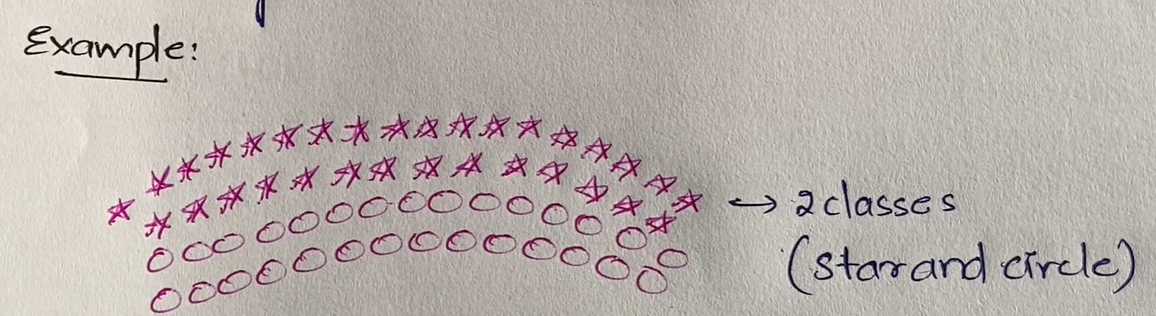
* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
* It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

# 

# 

**Radial Basis Function**

* Used in ANN
* Has only one hidden node
* In multilayer network we have multiple hidden nodes, but here only one hidden node is there



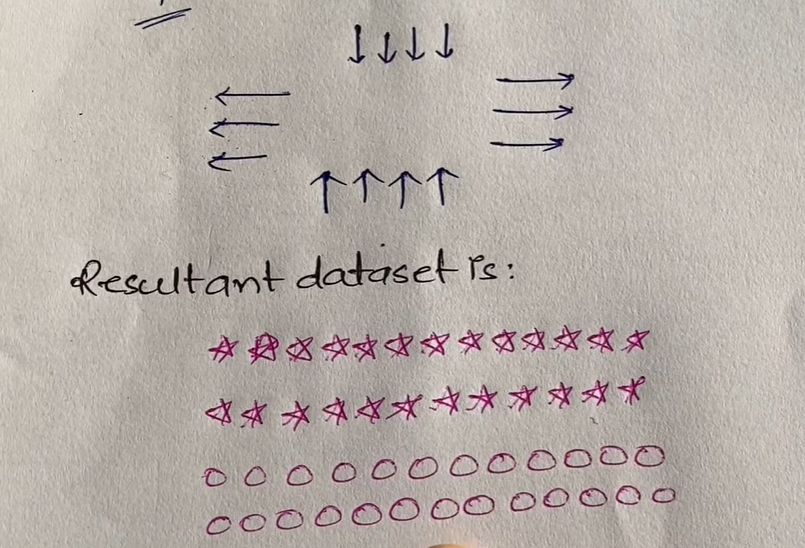
* There are 2 classes star & circle
* Here we can’t separate both these classes using a single line
* Data is not linearly separable
* So we need to convert data from non-linear to linearly separable data.
* We have 2 steps

1. Increase the dimensionality

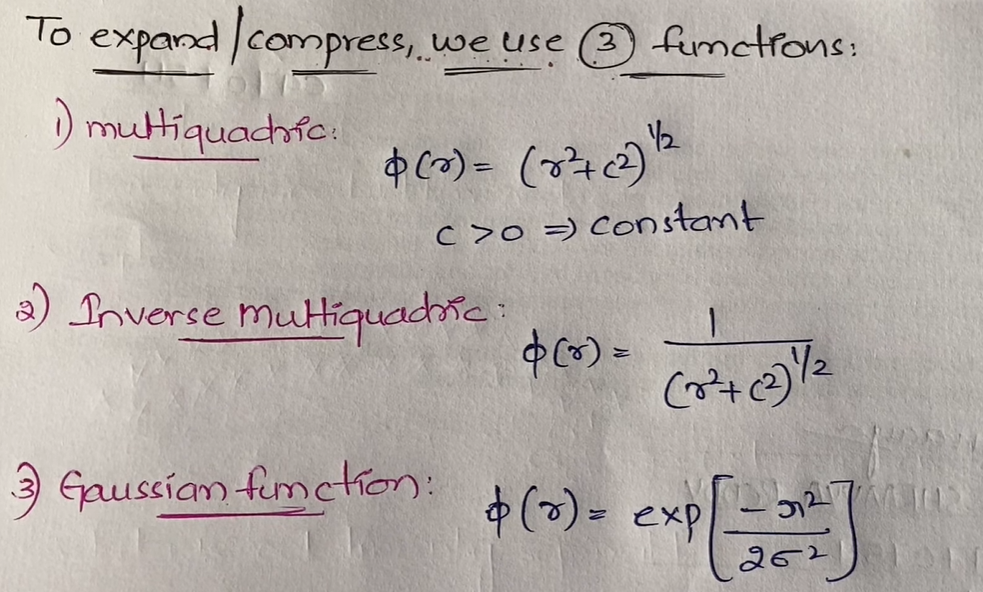
* Example to convert from 2D to 3D
* This step is only used if required i.e, not mandatory

1. Expanding the dimension (Horizontal)

Compress the direction (Vertical)



* We should not directly expand and compress the data. We need to follow certain standard.
* Consider a centre randomly
* Draw concentric circles i.e, from same centre multiple circles are drawn
* All of them have different radius

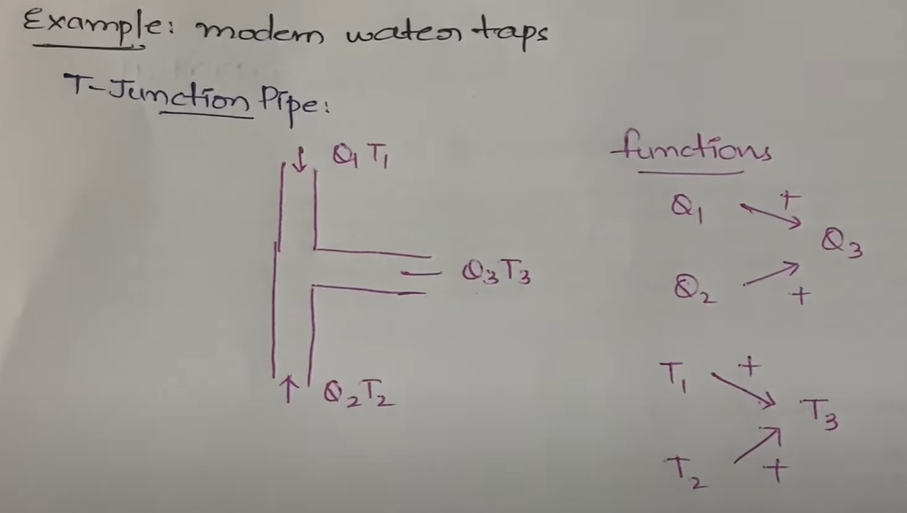


**Case-based Reasoning (CBR)**

* Case-based reasoning is an instance-based learning method
* CBR follows following properties

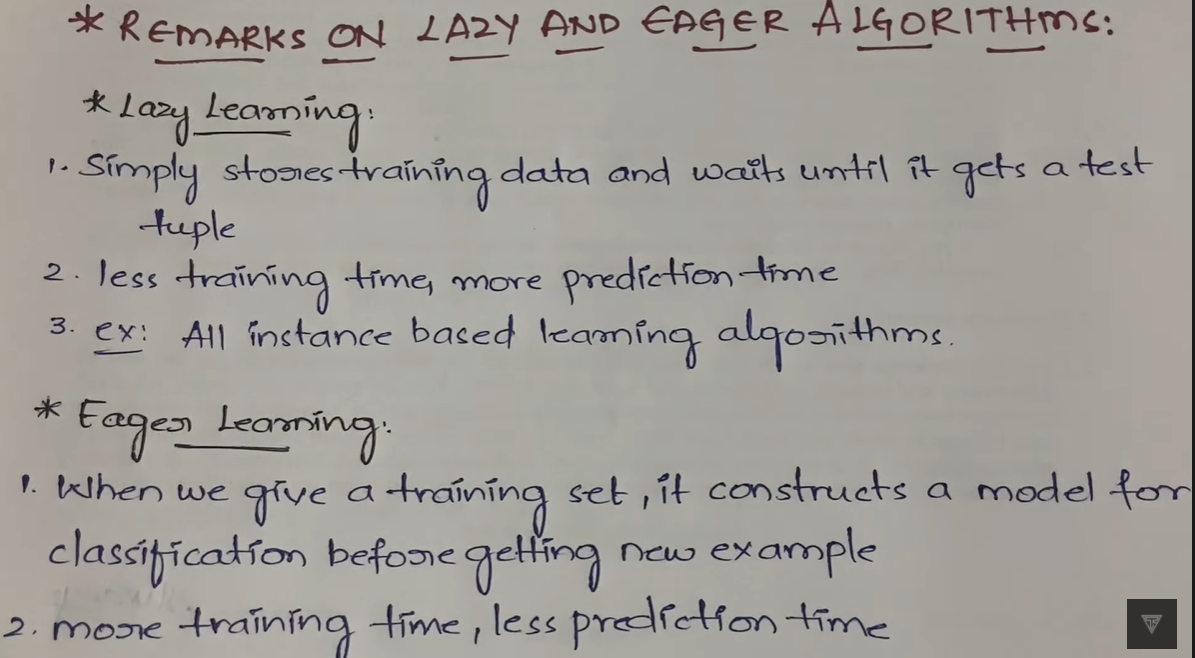
1. They are lazy learners
2. Classification is different for each instance

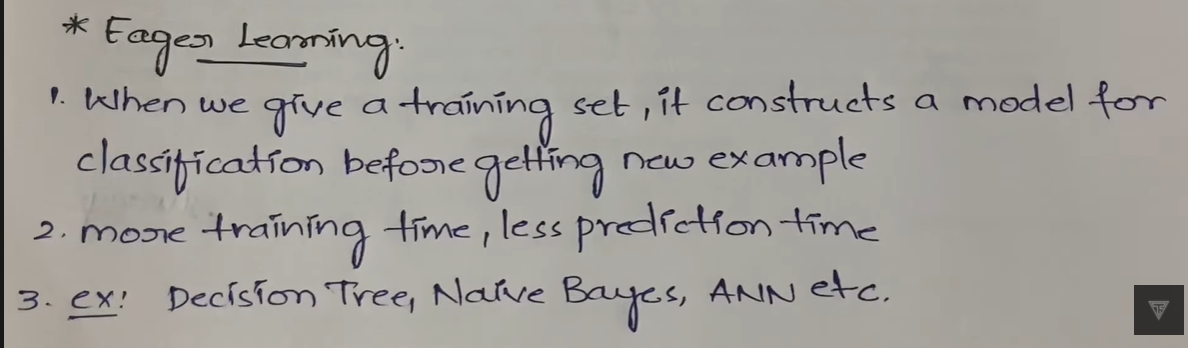
* In CBR, everything is considered as case (like switch-case different situations) and based on previous case we propose a solution
  + Instances are represented as symbols (not values)
* CBR has 3 components:
  + Similarity functions
  + Approximation/Adjustment of instances
  + Symbolic representation of instances
* For modelling a CBR, we use a CADET (Case Based Design Tool) system.
* CADET has 75 predefined libraries.



Q – Water flow, T – Temperature

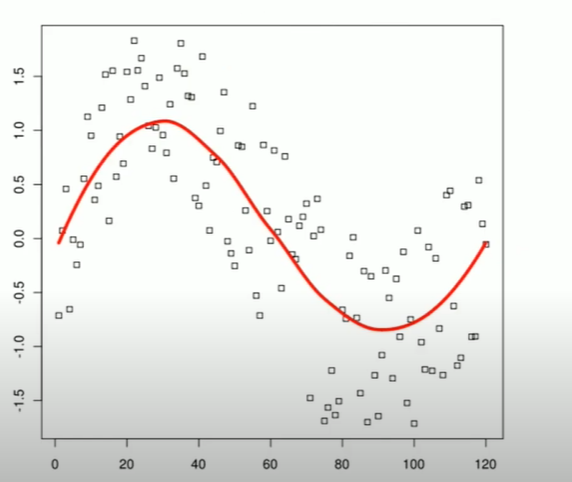
* Here we use these functions in order to get a particular water flow, and temperature of the water.
* These functions are used in order to get the water cool, medium, hot etc.

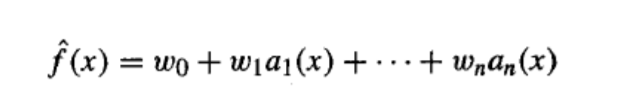
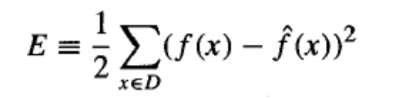




**Locally Weighted Regression**

* Locally Weighted Regression is instance-based learning algorithm.
* The phase “Locally Weighted Regression” is called
  + Local, because the function is approximated based only on data near the query point.
  + Weighted, because the contribution of each training example is weighted by its distance from query point.
  + Regression, because it’s widely used in statistical problems only for real-valued functions but not discrete-valued.



* We need to find a curve/line that fits the given data.
* the target function f is approximated near xq using a linear function of the form:
  + 
  + ai(x) denotes the value of the ith attribute of the instance x
* In contrast, local learning systems will divide the global learning problem into multiple smaller/simpler learning problems and this is usually achieved by dividing the cost function into multiple independent local cost functions.
* The disadvantage of global methods is that sometimes no parameter values can provide a sufficiently good approximation.
* Here we know the ai(x) values but we don’t know the weights of the function so we assign random weight values using gradient descent method.
* Then we will try to fine tune these values in such a way that it best fits the training examples.
* Gradient descent of ANN:
  + 
  + f(x) is known to us and f̂(x) is calculated o/p that will change.