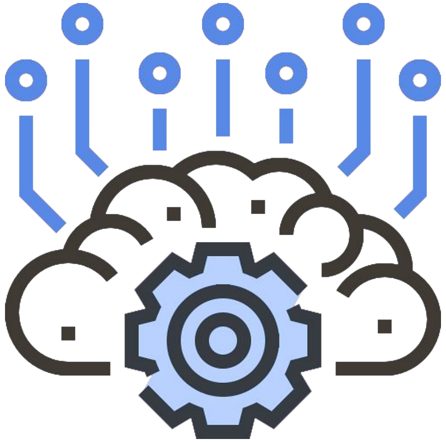
Module 2 Introduction to Machine Learning



What is machine learning?

> “**Learning is any process by which a system improves performance from experience.”**
>
> \- Herbert Simon

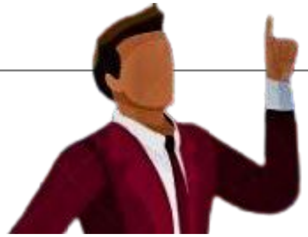
Definition by Tom Mitchell (1998):

Machine Learning is the study of algorithms that

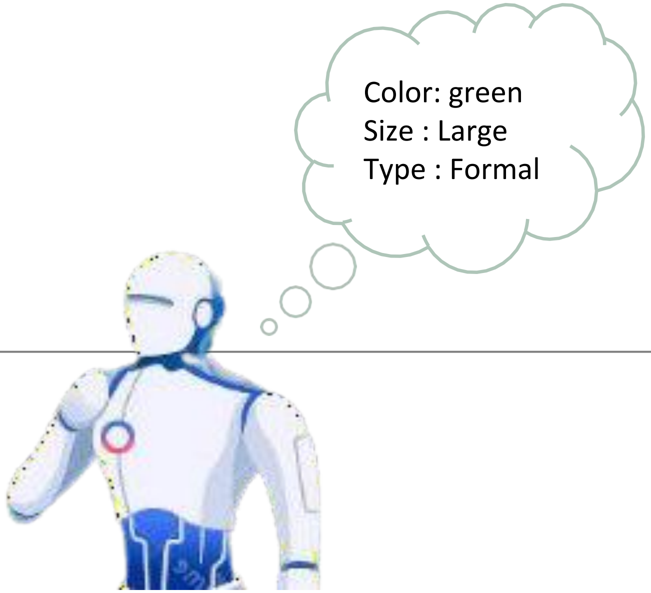
- improve their performance **P**
- at some task **T**
- with experience **E**.

A well-defined learning task is given by **<P,T,E>**

This is a shirt we used to wear.

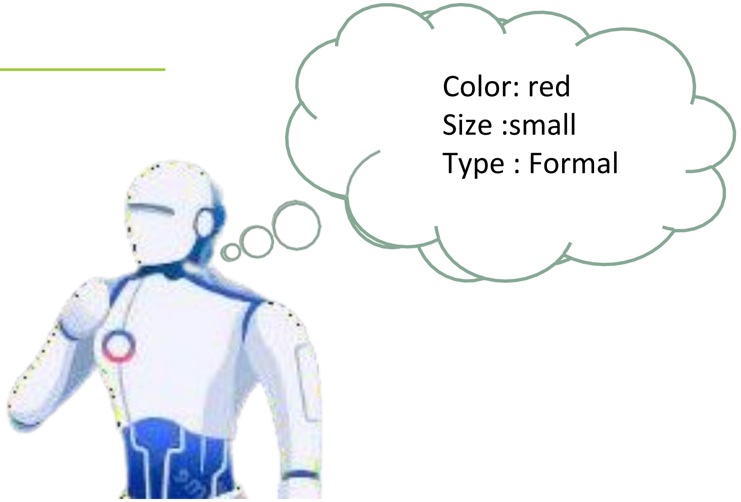


ok



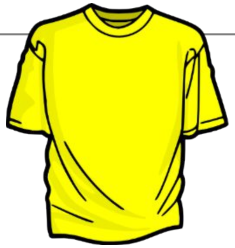
Is this a shirt ?

ok

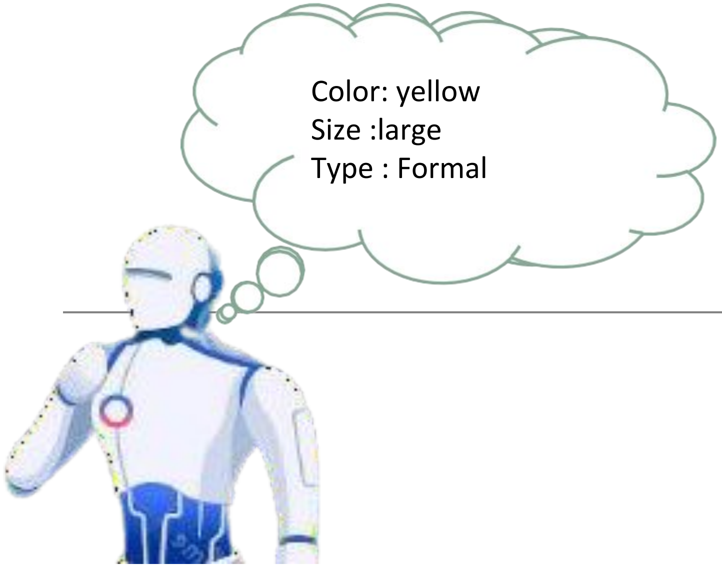


This is also a shirt

Is this a shirt ?



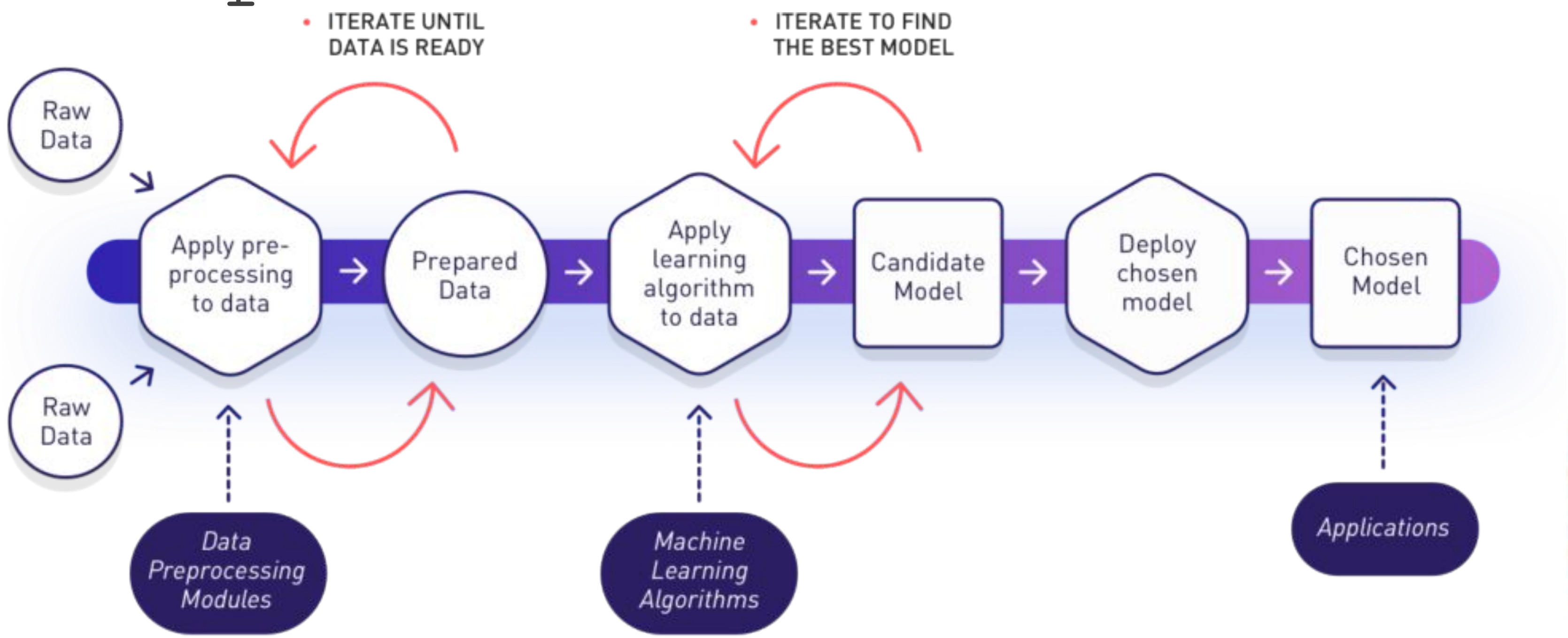
ok



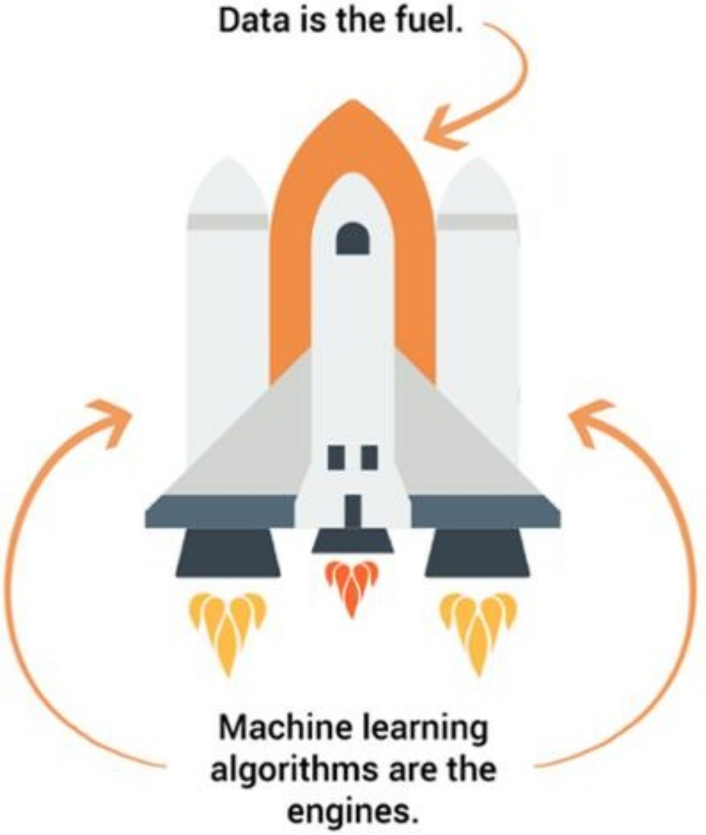
This is also a shirt

Yes, these are all shirts

Machine Learning Pipeline



• ITERATE UNTIL DATA IS READY
• ITERATE TO FIND THE BEST MODEL
Raw Data
Raw Data
Apply pre-processing to data
Prepared Data
Apply learning algorithm to data
Candidate Model
Deploy chosen model
Chosen Model
Data Preprocessing Modules
Machine Learning Algorithms
Applications



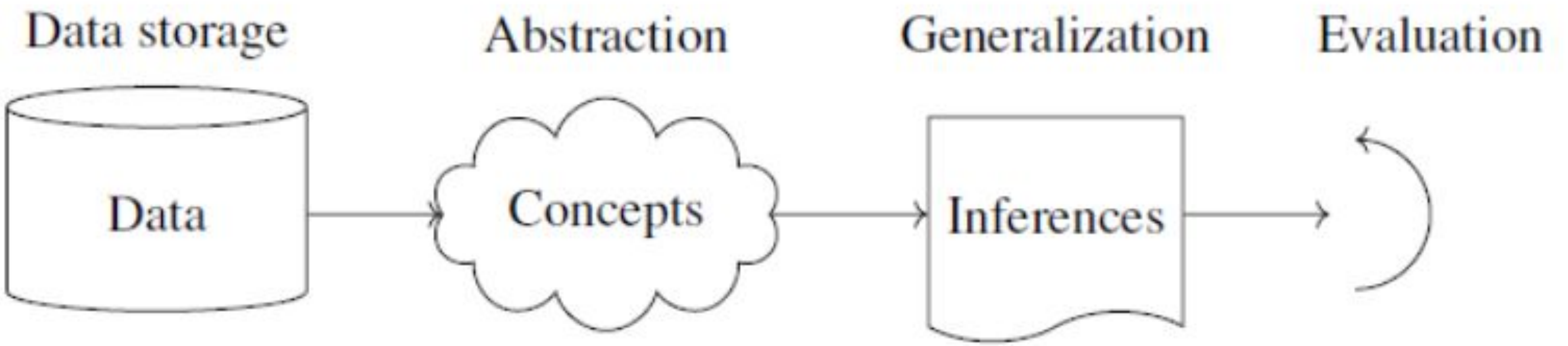
Learning

- **Definition of learning**
- Definition A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks T, as measured by P, improves with experience E
- **.Example**: Handwriting recognition learning problem
- **Task T:** Recognising and classifying handwritten words within images
- **Performance P**: Percent of words correctly classified
- **Training experience E**: A dataset of handwritten words with given classifications

Components of Learning

- **Basic components of learning process**
- The learning process, whether by a human or a machine, can be divided into four components, namely, data storage, abstraction, generalization and evaluation.
- Figure illustrates the various components and the steps involved in the learning process

Components of Learning



Data Storage

- Facilities for storing and retrieving huge amounts of data are an important component of the learning process. Humans and computers alike utilize data storage as a foundation for advanced reasoning.
- In a human being, the data is stored in the brain and data is retrieved using electrochemical signals.
- Computers use hard disk drives, flash memory, random access memory and similar devices to store data and use cables and other technology to retrieve data.

Abstraction

- The second component of the learning process is known as abstraction.
- Abstraction is the process of extracting knowledge about stored data. This involves creating general concepts about the data as a whole. The creation of knowledge involves application of known models and creation of new models.
- The process of fitting a model to a dataset is known as training. When the model has been trained, the data is transformed into an abstract form that summarizes the original information

3. Generalization

- The third component of the learning process is known as generalisation. The term generalization describes the process of turning the knowledge about stored data into a form that can be utilized for future action.
- These actions are to be carried out on tasks that are similar, but not identical, to those what have been seen before. In generalization, the goal is to discover those properties of the data that will be most relevant to future tasks.

4. Evaluation

- Evaluation is the last component of the learning process.
- It is the process of giving feedback to the user to measure the utility of the learned knowledge. This feedback is then utilised to effect improvements in the whole learning process

When to machine learning?

ML is used when:

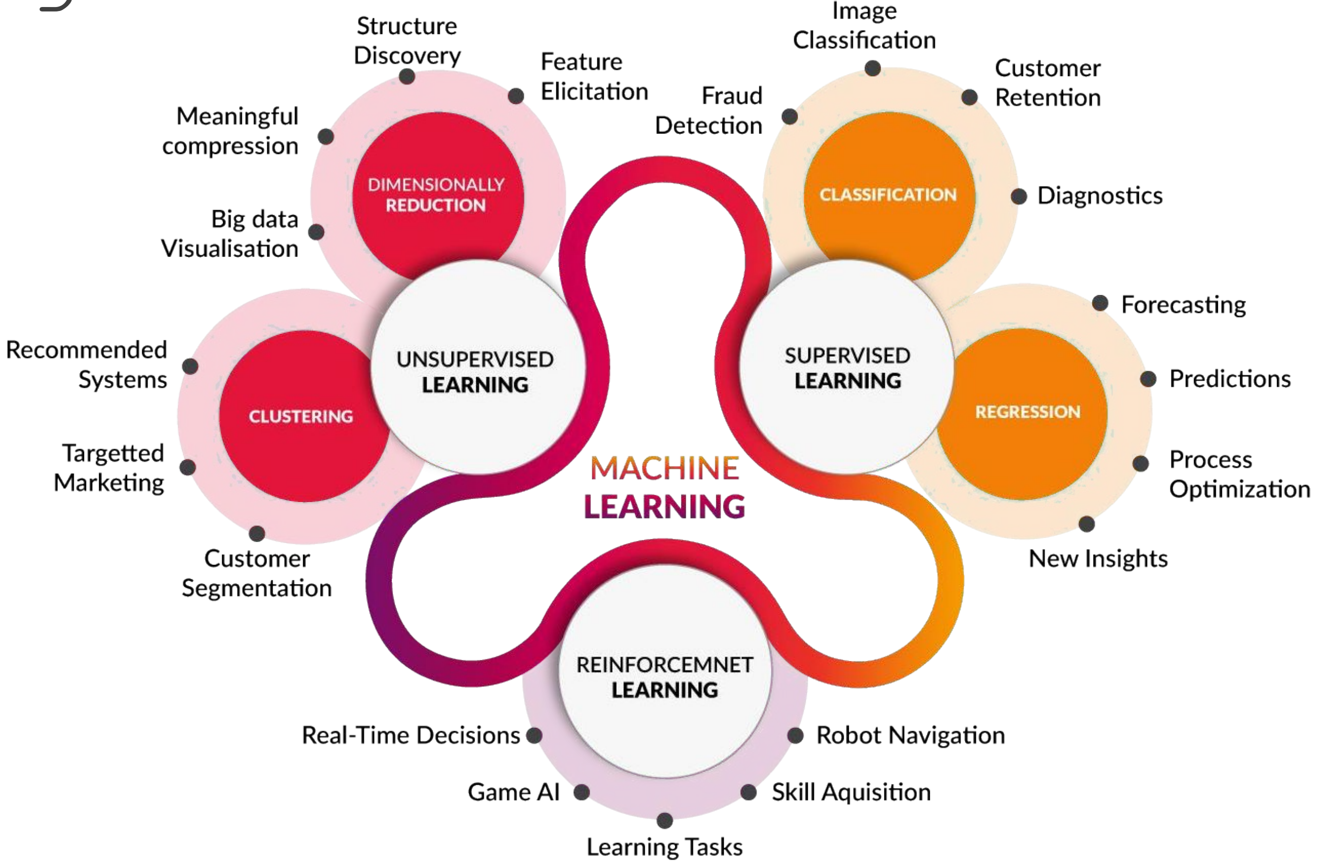
- Human expertise does not exist (navigating on Mars)
- Humans can’t explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)

Learning isn’t always useful - There is no need to “learn” to calculate payroll

Abstraction and Generalization -summary

- **Abstraction**
- Organizing and making sense of the immense amount of data /knowledge we have
- **Generalization**
- The ability of an algorithm to perform accurately on new ,unseen examples after having trained on a learning dataset

Types of Learning



Supervised Learning

Input : Labeled Data

X (*features*)	Y (*labels*)
$x_{11}, x_{12}, x_{13},$ ....x_{1n}	y_1
. . .	. . .
$x_{k1}, xk_2, x_{k3},$ ....x_{kn}	y_k

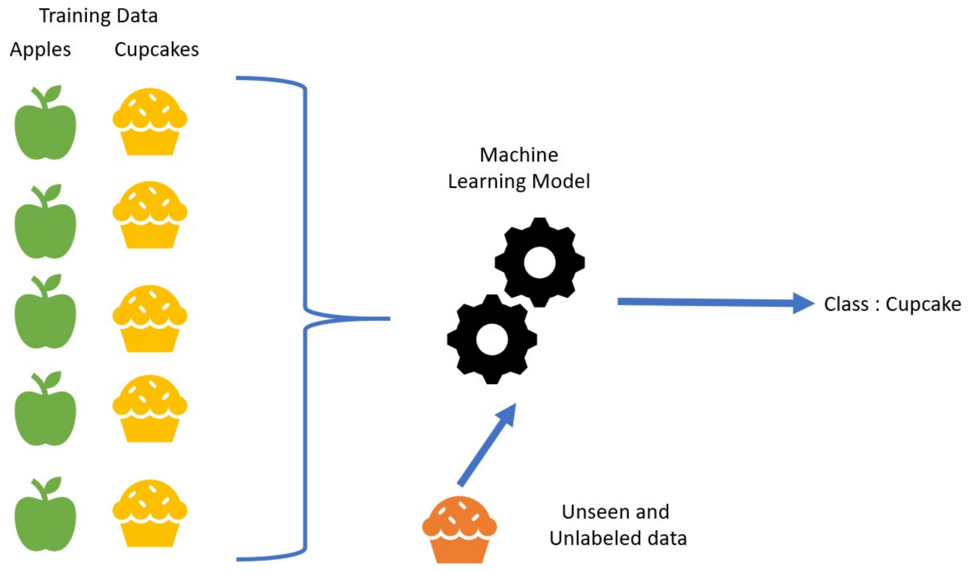
Goal : Construct a predictor $f : X \to Y$

to minimize the error between $\overset{□}{Y}$, Y

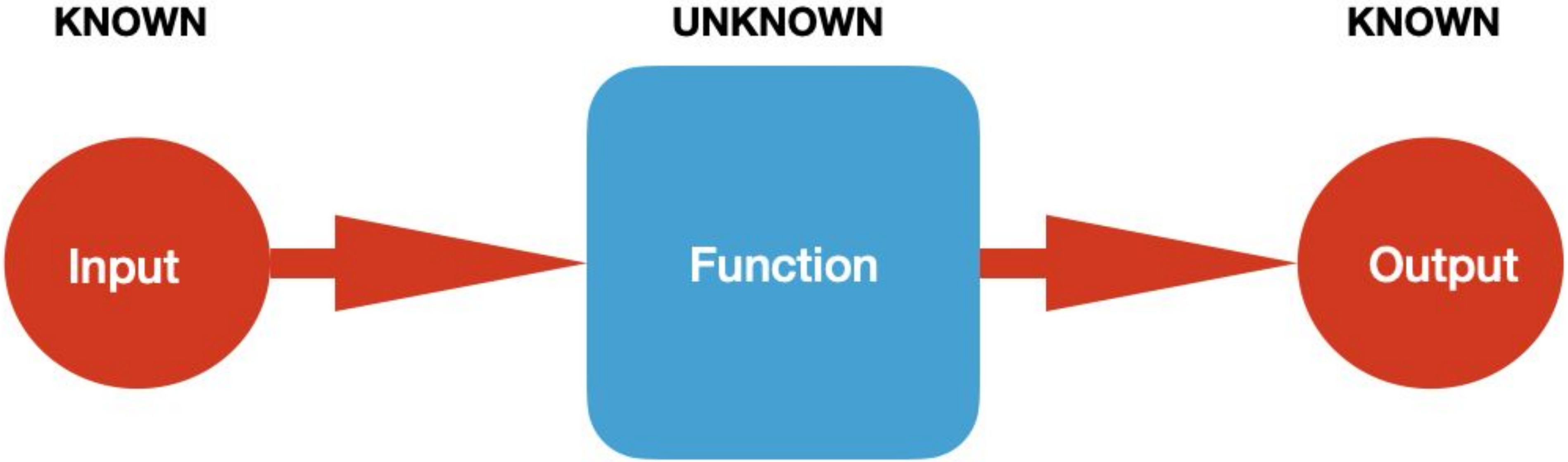
( )

where, $\overset{□}{Y} = f(X)$

Use : using predictor to predict ?ஒ◌ʃ = f ?ஒ◌ʃ



Supervised learning



KNOWN

UNKNOWN

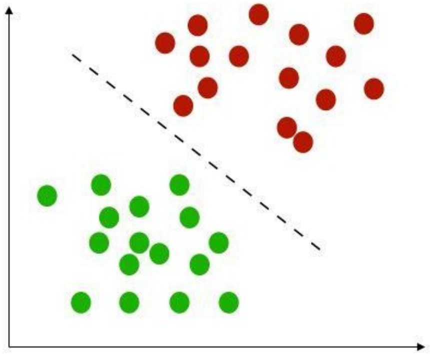
KNOWN

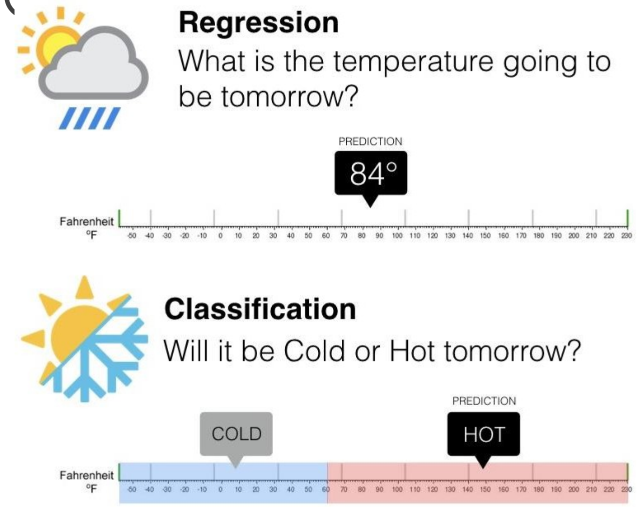
Input

Function

Output

Supervised Learning



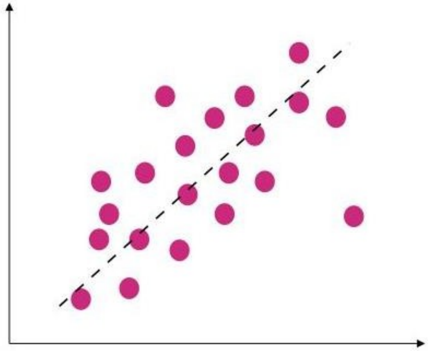


Regression

What is the temperature going to be tomorrow?

Classification

Will it be Cold or Hot tomorrow?



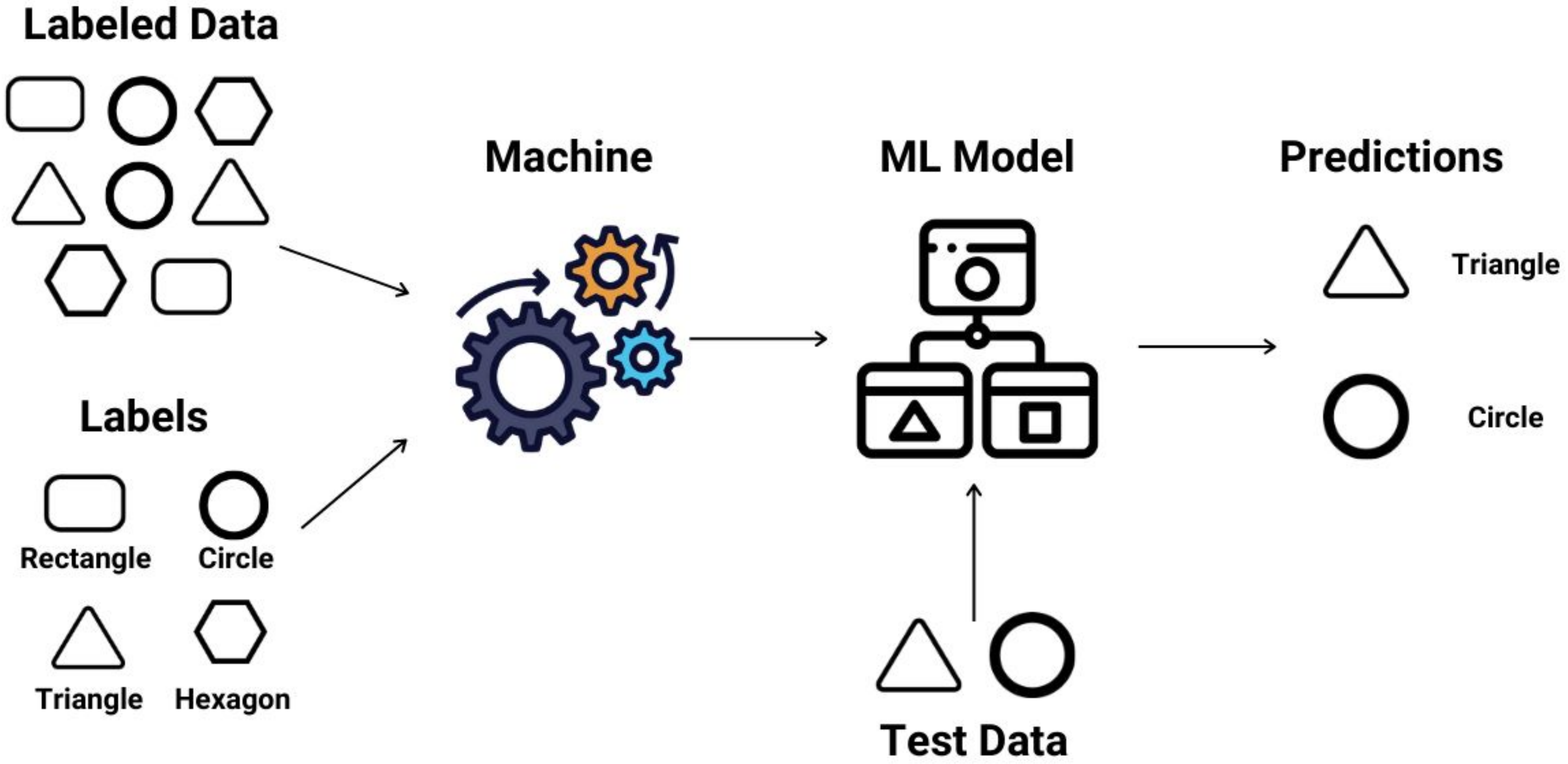
Classification

- features and discrete labels
- maps an input to discrete label(class)
- Eg: spam or not, type of cancer

Regression

- features and continues real values
- Predict a real value for an input
- Eg: gold price, temperature

Supervised Learning



Labeled Data

Labels

Rectangle

Circle

Triangle

Hexagon

Machine

ML Model

Test Data

Predictions

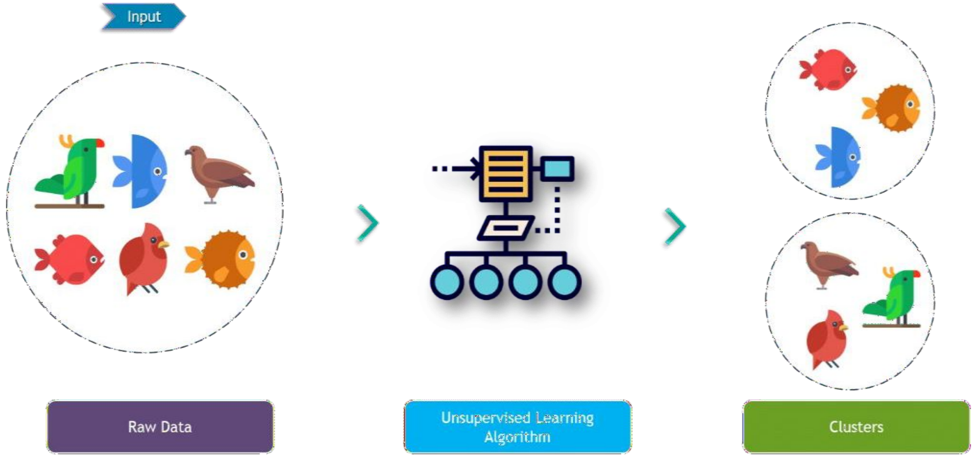
Triangle

Circle

Unsupervised Learning

Input : Unlabeled Data

X (features)
$x_{11}, x_{12}, x_{13}, \ldots\ldots\ldots\ldots x_{1n}$
$\vdots$
$x_{k1}, xk_2, x_{k3}, \ldots\ldots\ldots\ldots x_{kn}$


Input

Raw Data

Unsupervised Learning Algorithm

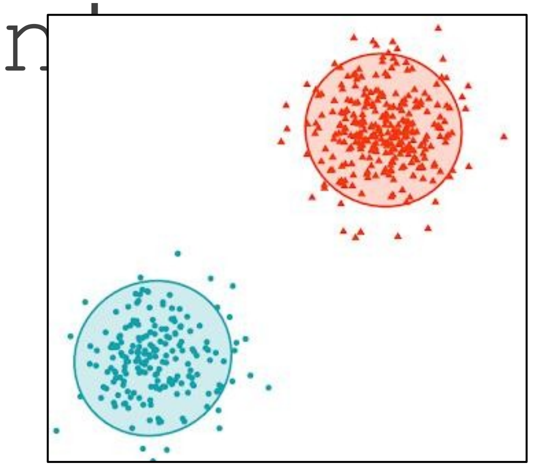
Clusters

Goal : Construct a analyzer to find the hidden relationship between inputs

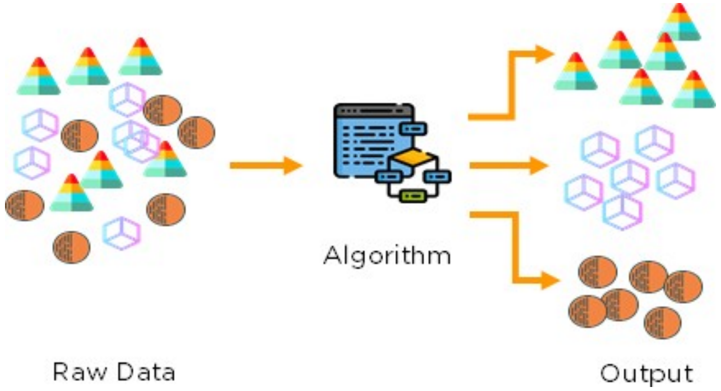
$x_1, x_2, x_3, \ldots , x_k$

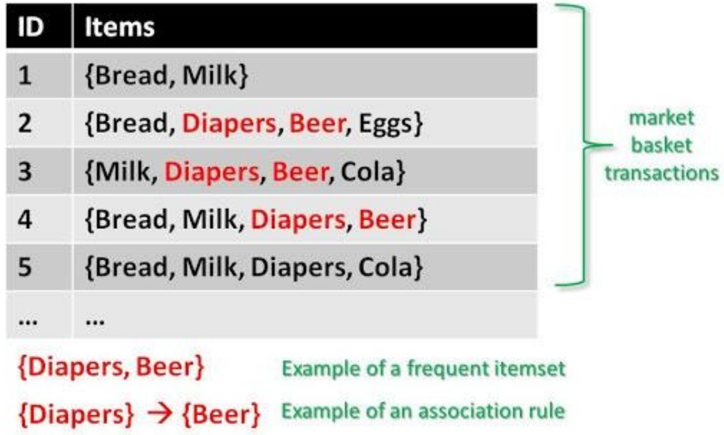
Use : Group or associate inputs according to their similarity

Unsupervised Learn



Clustering





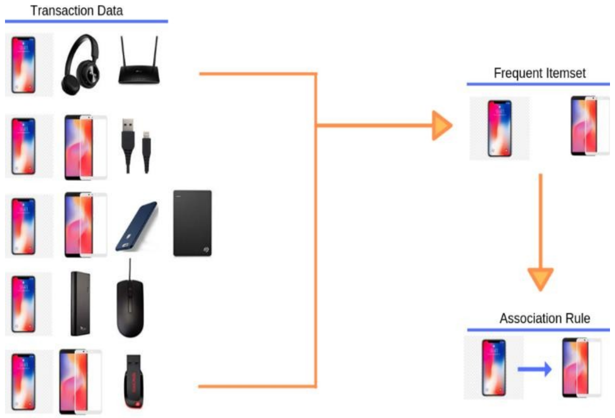
ID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}
…	…

market basket transactions

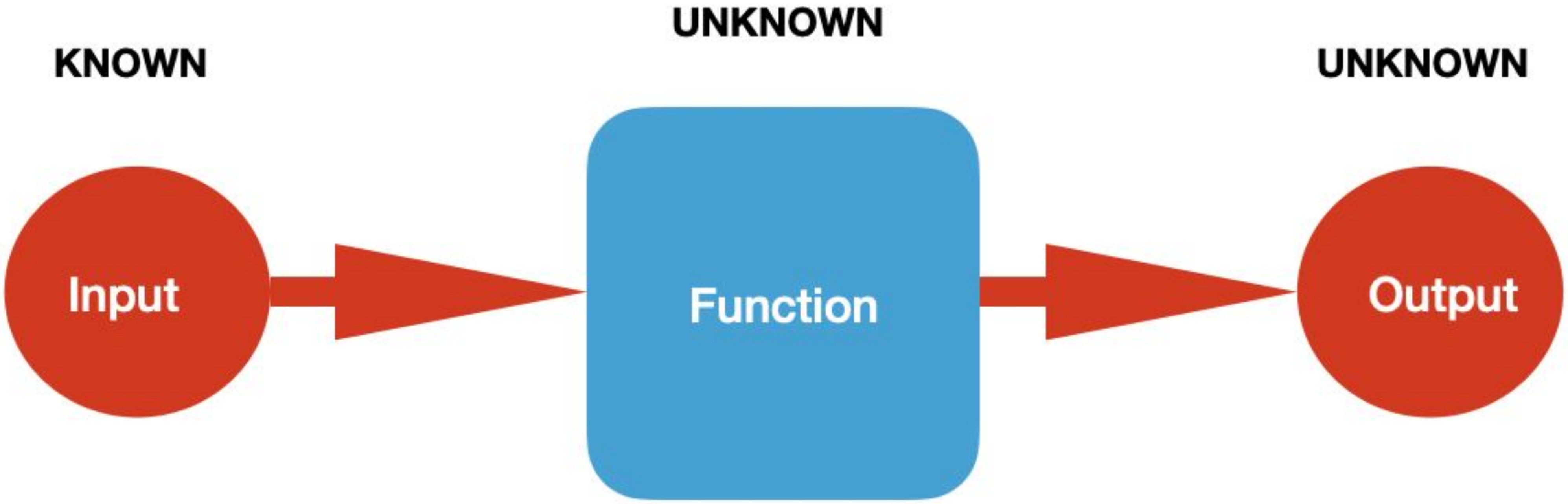
{Diapers, Beer} Example of a frequent itemset

{Diapers} → {Beer} Example of an association rule

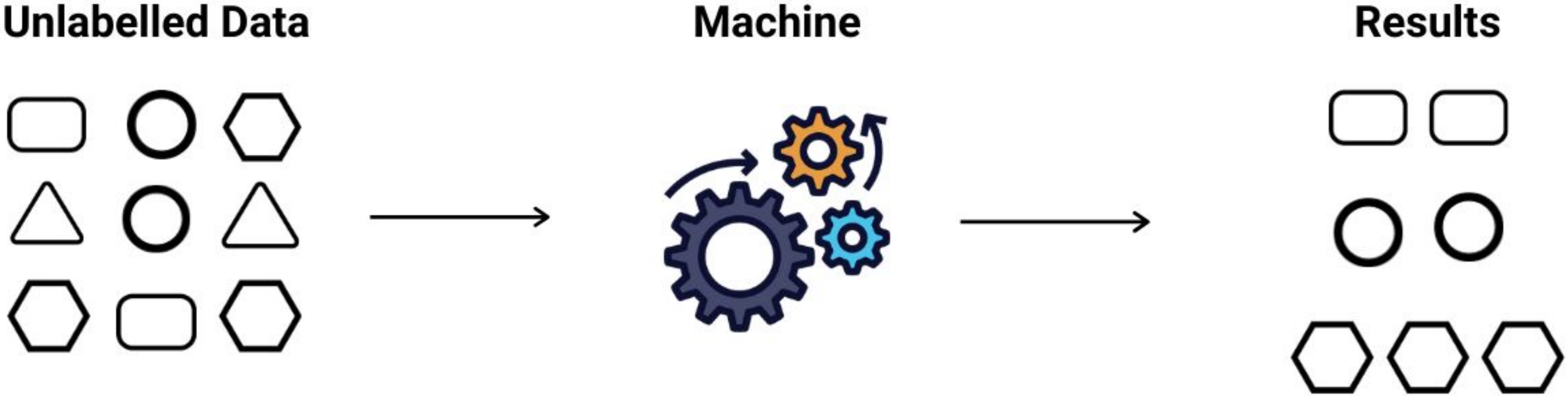
Itemset and rule mining



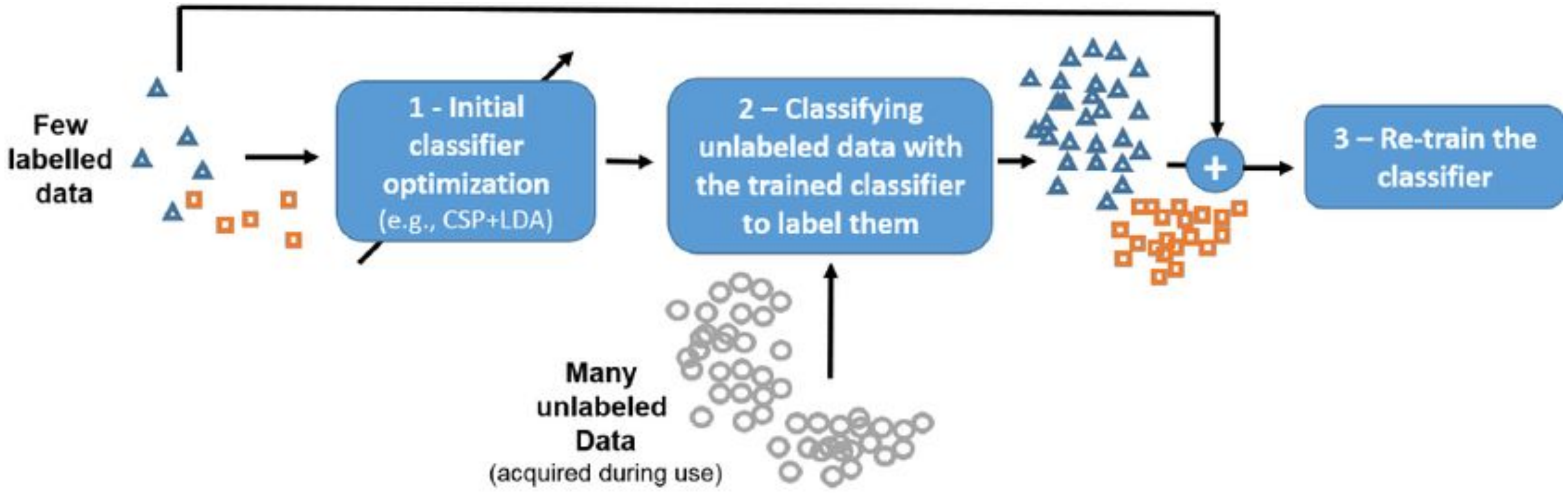
Unsupervised learning



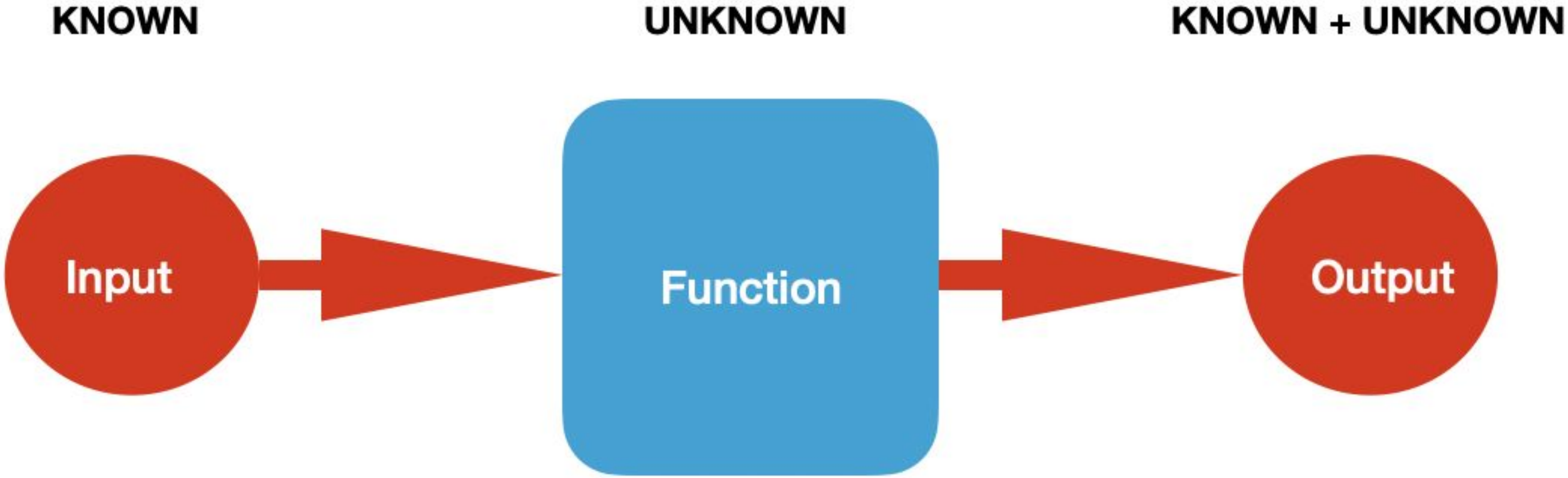
Unsupervised Learning



Semi Supervised Learning

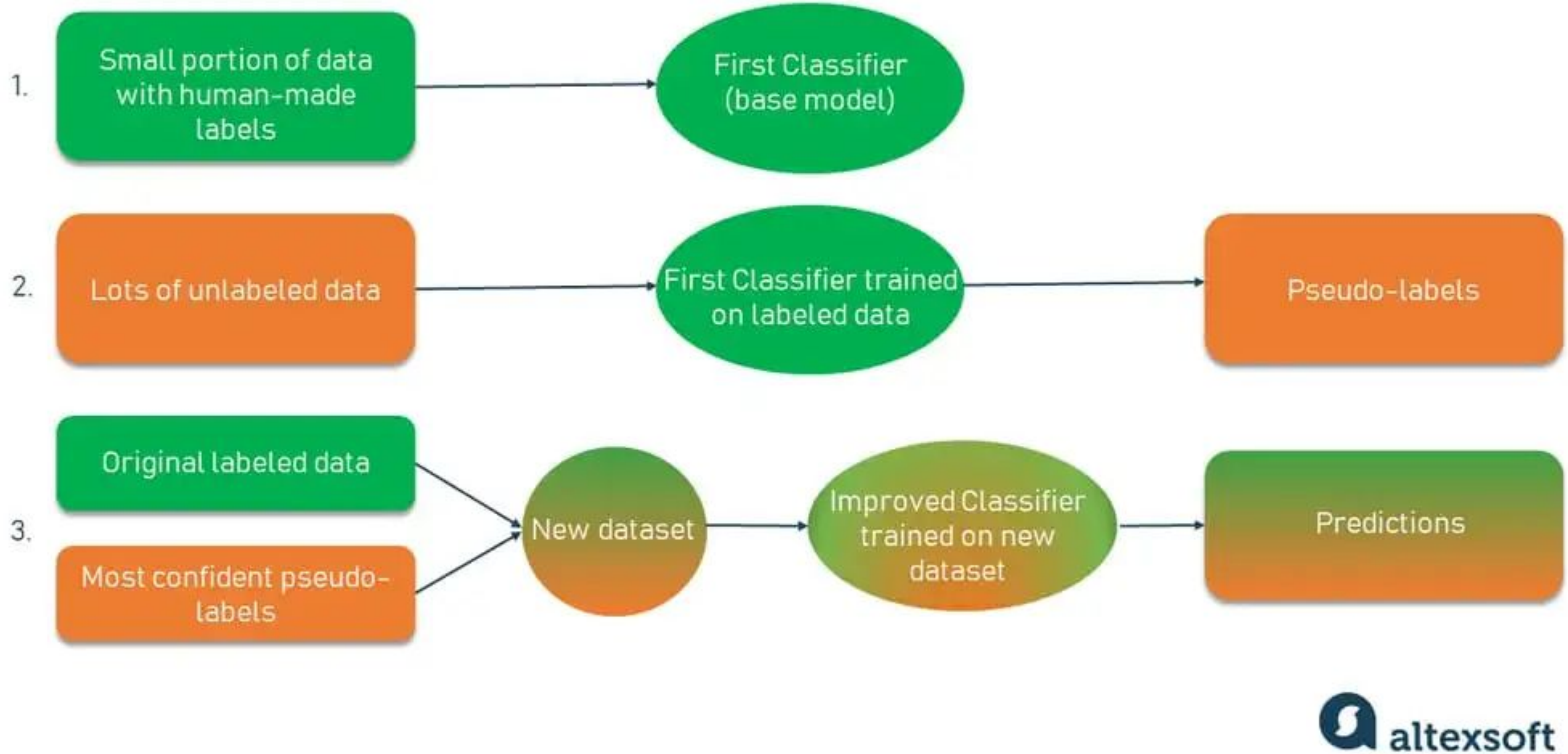

Few labelled data
1 - Initial classifier optimization (e.g., CSP+LDA)
2 – Classifying unlabeled data with the trained classifier to label them
+
3 – Re-train the classifier
Many unlabeled Data (acquired during use)

Semi supervised learning



It is a method that uses a small amount of labeled data and a large amount of unlabeled data to train a model. The goal of semi-supervised learning is to learn a function that can accurately predict the output variable based on the input variables, similar to supervised learning. However, unlike supervised learning, the algorithm is trained on a dataset that contains both labeled and unlabeled dataa.

SEMI-SUPERVISED SELF-TRAINING METHOD

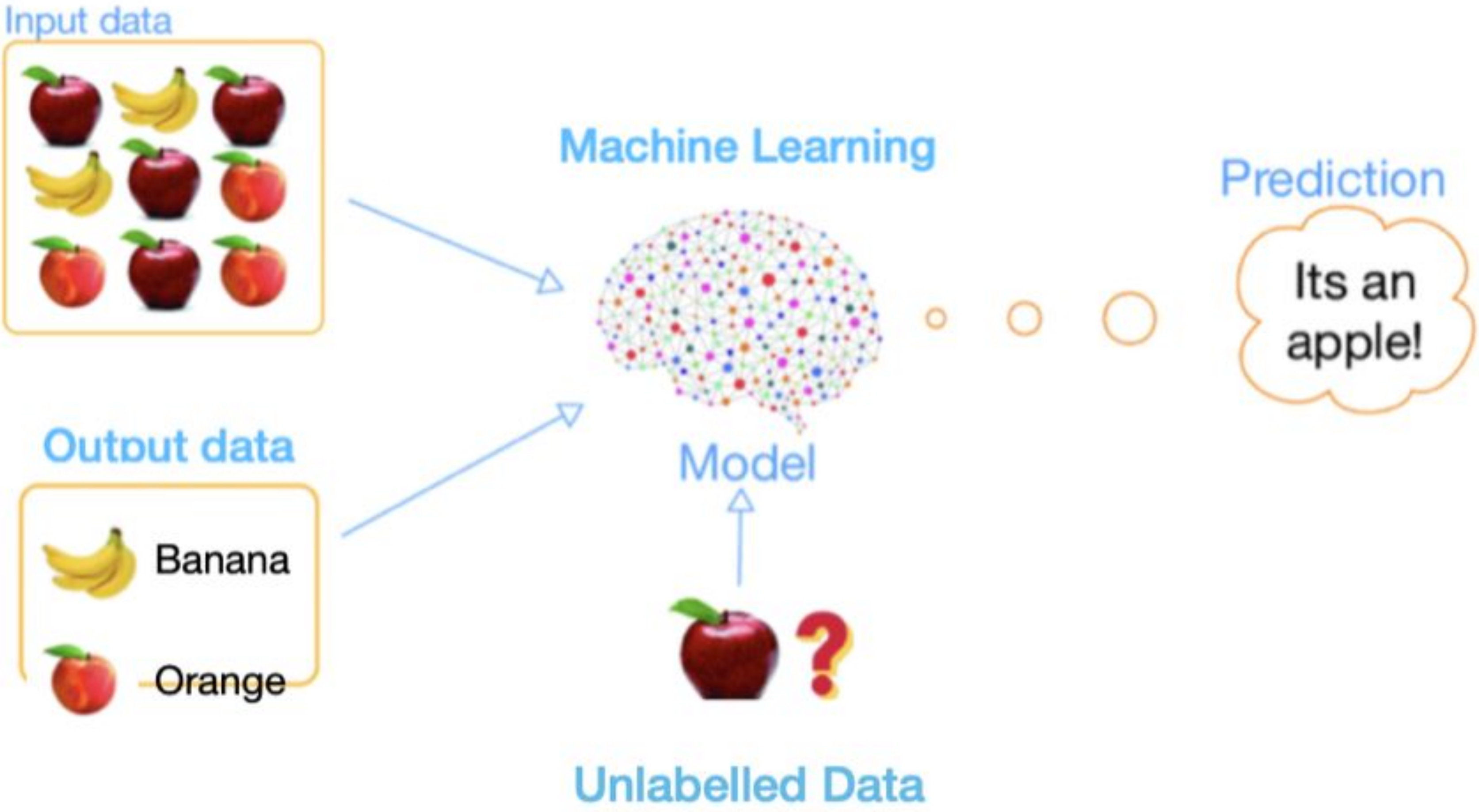


1. Small portion of data with human-made labels → First Classifier (base model)

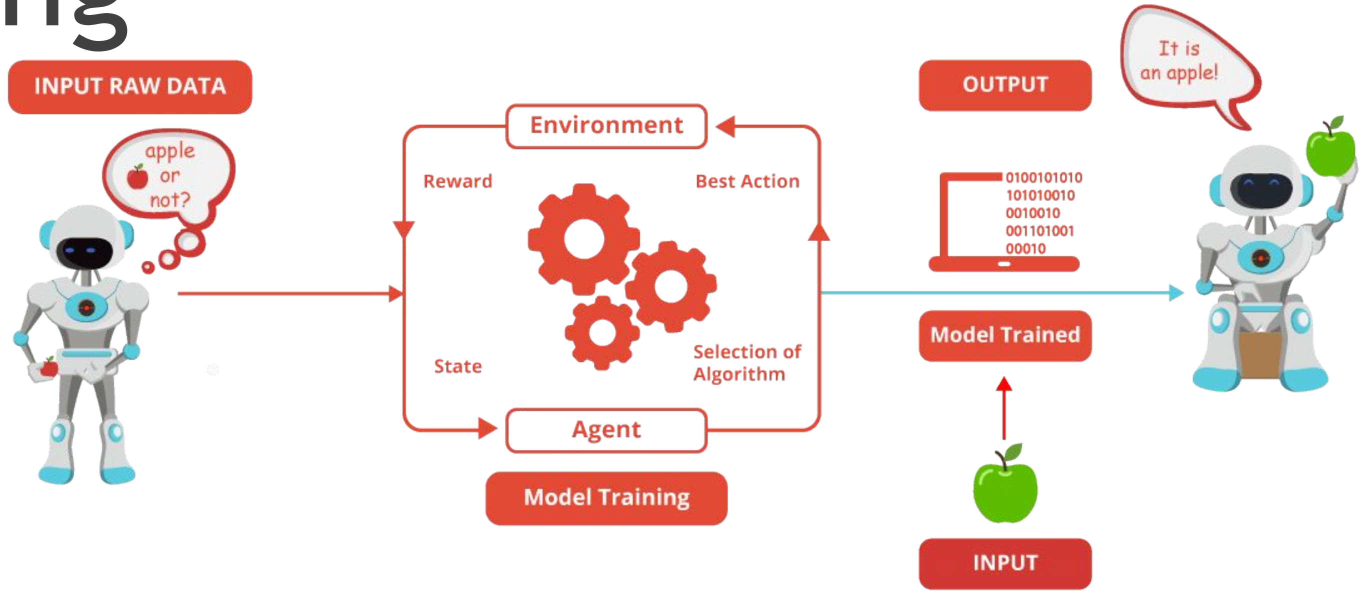
2. Lots of unlabeled data → First Classifier trained on labeled data → Pseudo-labels

3. Original labeled data + Most confident pseudo-labels → New dataset → Improved Classifier trained on new dataset → Predictions

altexsoft


Input data
Machine Learning
Prediction
Its an apple!
Output data
Model
Banana
Orange
?
Unlabelled Data

Reinforcement Learning



- Use software agents
- Based on rewards
- Objective is to maximize rewards for better learning

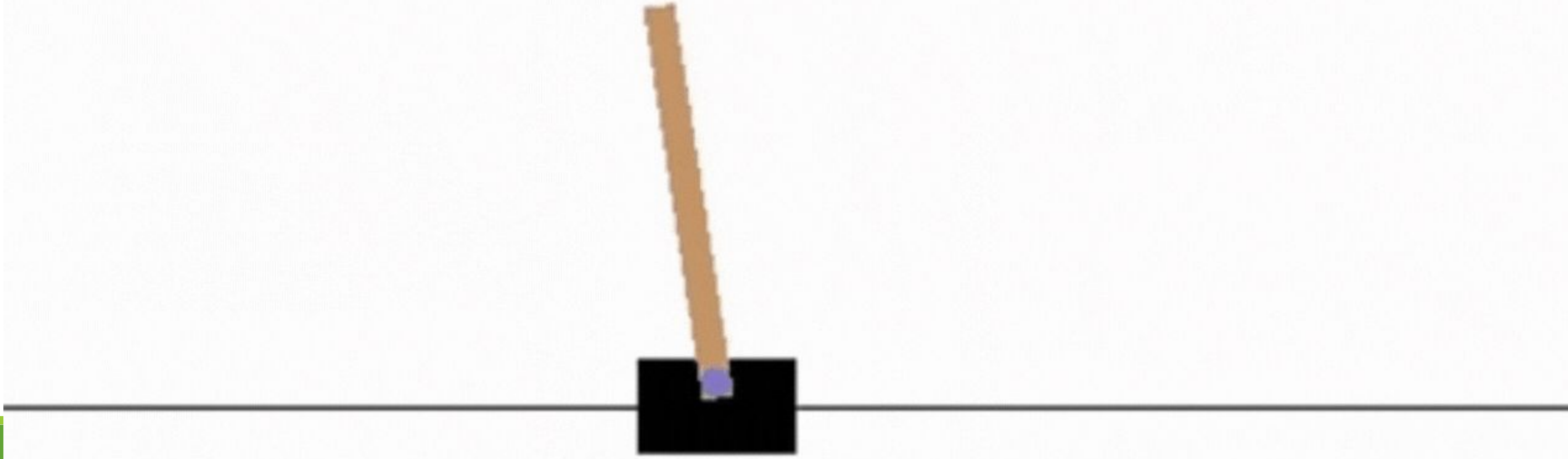
Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation.

Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

Balancing the
stick on our hands

Cart-pole Problem

Reinforcement Learning Agent is trying to balance the wooden pole on the small cart



The kNN algorithm

- The nearest neighbours approach to classification is utilized by the kNN algorithm.
-

KNN Classifier
Input value
Predicted Output

Lazy learning and eager learning

- **Lazy learner:**
- Just store Data set without learning from it
- Start classifying data when it receive Test data
- So it takes less time learning and more time classifying data

Why Knn is a lazy learner

- KNN algorithm is the Classification algorithm. ... K-NN is a lazy learner because it doesn't learn a discriminative function from the training data but memorizes the training dataset instead.
- There is no training time in K-NN.
- The prediction step in K-NN is expensive.

- **Eager learner:**
- When it receive data set it starts classifying (learning)
- Then it does not wait for test data to learn
- So it takes long time learning and less time classifying data

- Strengths

- Simple and effective
- Makes no assumptions about the underlying data distribution
- Fast training phase

Weaknesses -KNN

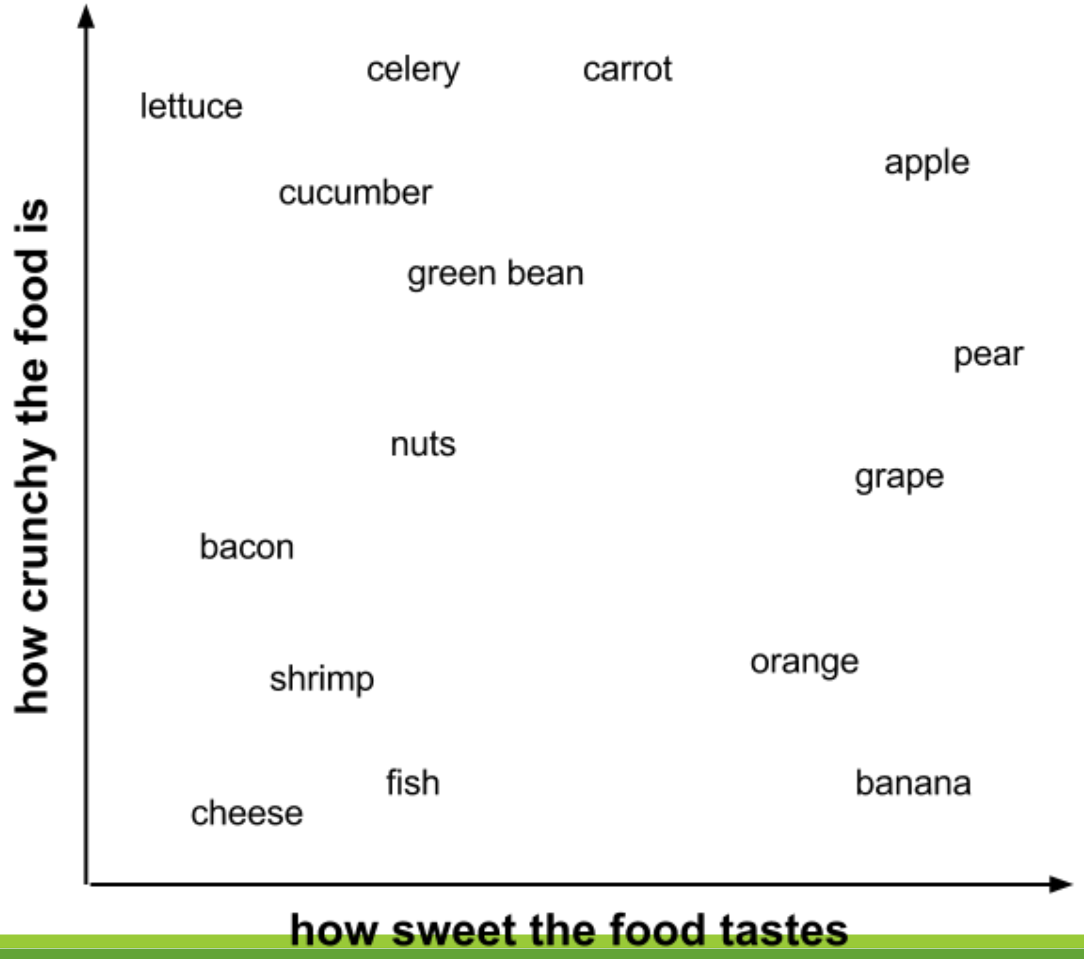
- Does not produce a model, which limits the ability to find novel insights in relationships among features
- Slow classification phase
- Requires a large amount of memory
- Nominal features and missing data require additional processing

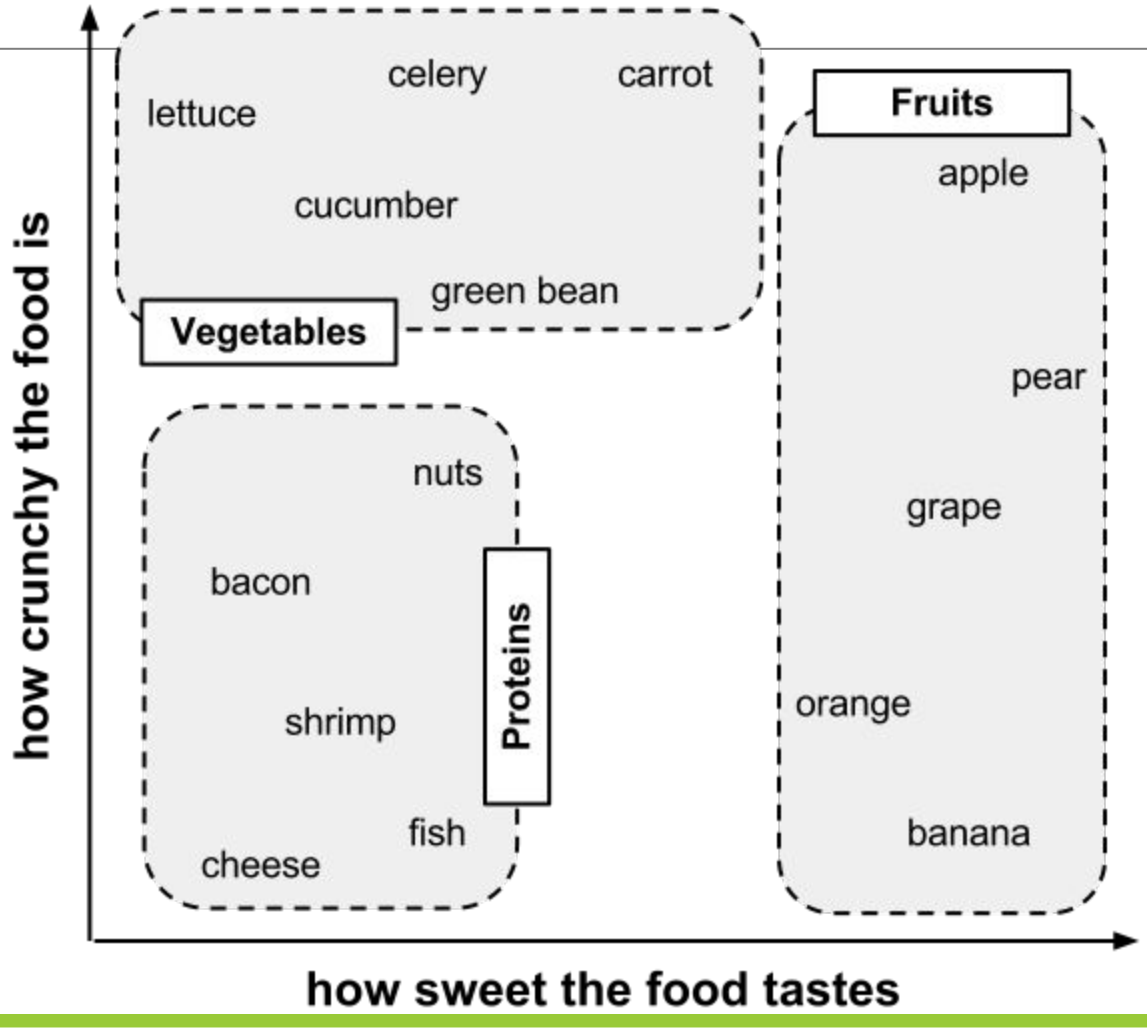
- The kNN algorithm begins with a **training dataset made up of examples that are classified into several categories, as labelled by a nominal variable.**
- Assume that we have a **test dataset containing unlabelled examples** that otherwise have the same features as the training data. For each record in the test dataset, **kNN identifies k records in the training data that are the "nearest" in similarity, where k is an integer specified in advance.**
- The unlabelled test instance is assigned the class of the majority of the k nearest neighbours

- Suppose that prior to eating the mystery meal we created a taste dataset in which we recorded our impressions of a number of ingredients we tasted previously. To keep things simple, we recorded only two features of each ingredient.
- The first is a measure from 1 to 10 of how crunchy the ingredient is, and the second is a 1 to 10 score of how sweet the ingredient tastes. We then labelled each ingredient as one of three types of food**: fruits, vegetables, or proteins.** The first few rows of such a dataset might be structured as follows:

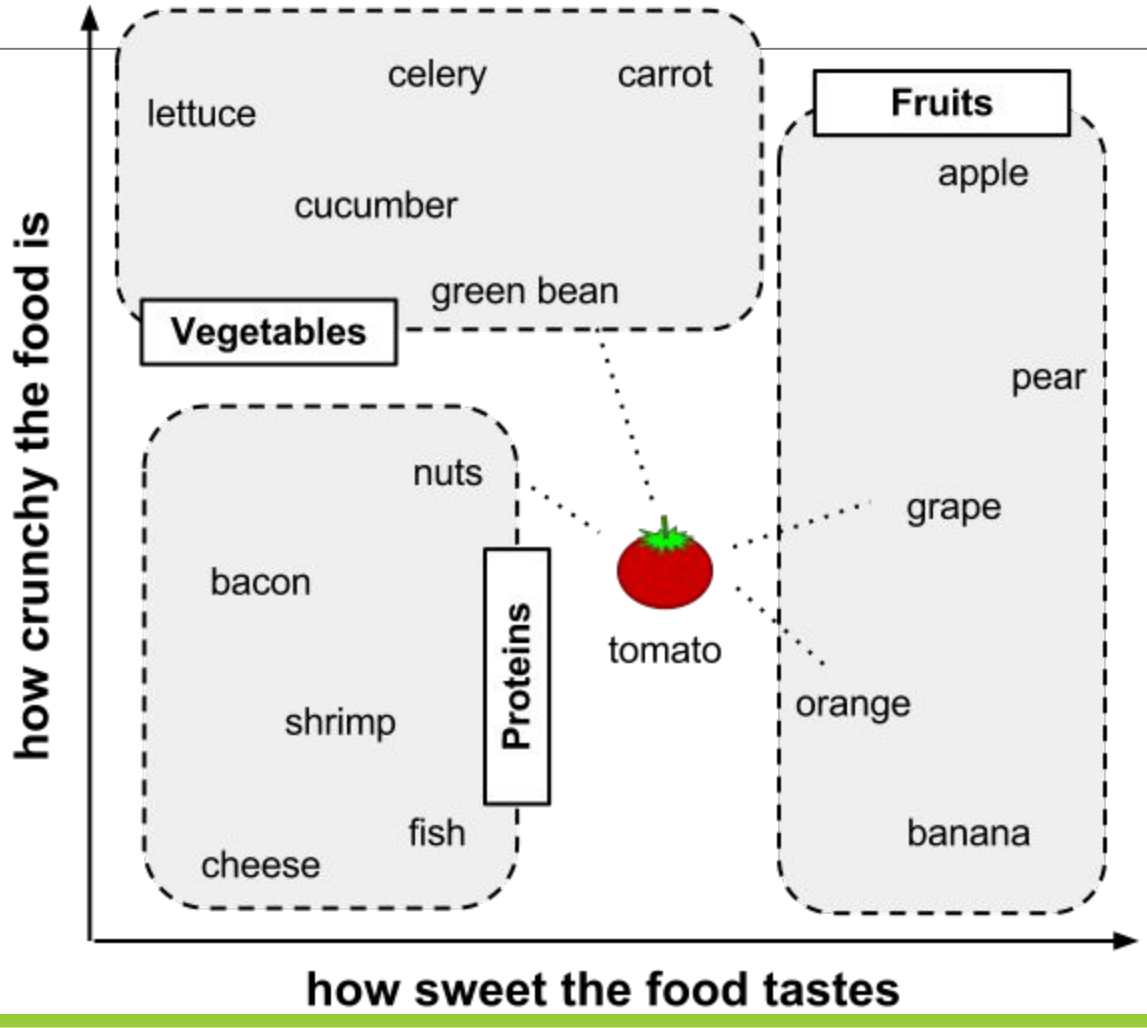
ingredient	sweetness	crunchiness	food type
Apple	10	9	fruit
Bacon	1	4	protein
banana	10	1	fruit
Carrot	7	10	vegetable
celery	3	10	vegetable
Cheese	1	1	protein

- The kNN algorithm treats the features as coordinates in a multidimensional feature space.
-




celery
carrot
lettuce
cucumber
green bean
Vegetables
Fruits
apple
pear
grape
orange
banana
nuts
bacon
shrimp
fish
cheese
Proteins
how crunchy the food is
how sweet the food tastes

- **is a tomato a fruit or a vegetable?** We can use a nearest neighbour approach to determine which class is a better fit as shown in the following figure:
-



how crunchy the food is
how sweet the food tastes
Vegetables
lettuce
celery
carrot
cucumber
green bean
Fruits
apple
pear
grape
orange
banana
Proteins
nuts
bacon
shrimp
fish
cheese
tomato

- Calculating distance
- **Locating the tomato's nearest neighbours requires a distance function**, or a formula that measures the similarity between two instances.
- There are many different ways to calculate distance. Traditionally, the kNN algorithm uses Euclidean distance, which is the distance one would measure

- ~~Euclidean distance is specified by the following formula~~, where p and q are the examples to be compared, each having n features. The term p 1 refers to the value of the first feature of example p, while q 1 refers to the value of the first feature of example q:
- dist (p , q) =√ (p 1 − q 1)^2 + (p 2 − q 2)^2 + ... + (p n − q n)^2

- The distance formula involves comparing the values of each feature. For example, to calculate the distance between the tomato (sweetness = 6, crunchiness = 4), and the green bean (sweetness = 3, crunchiness = 7),
- we can use the formula as follows:
- dist (tomato , green bean) = $\surd(6 - 3)^2 + (4 - 7)^2 = 4.2$

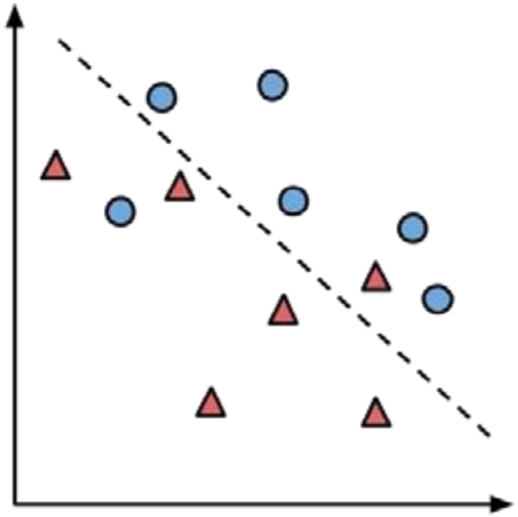
- we can calculate the distance between the tomato and several of its closest neighbors as follows:

ingredient	sweetness	crunchiness	food type	distance to the tomato
Grape	8	5	fruit	sqrt((6 - 8)^2 + (4 - 5)^2) = 2.2
green bean	3	7	vegetable	sqrt((6 - 3)^2 + (4 - 7)^2) = 4.2
nuts	3	6	protein	sqrt((6 - 3)^2 + (4 - 6)^2) = 3.6
Orange	7	3	fruit	sqrt((6 - 7)^2 + (4 - 3)^2) = 1.4

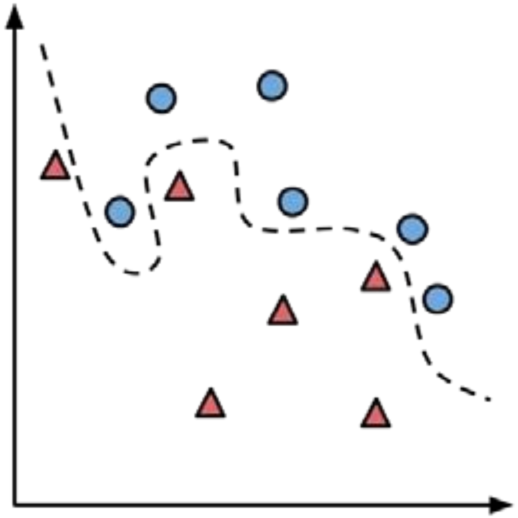
- To classify the tomato as a vegetable, protein, or fruit, we'll begin by assigning the tomato, the food type of its single nearest neighbour
- This is called 1NN classification because k = 1. The orange is the nearest neighbour to the tomato, with a distance of 1.4. As orange is a fruit, the 1NN algorithm would classify tomato as a fruit.

- Deciding **how many neighbours to use for kNN determines how well the model will generalize to future data.**
- The balance between **over-fitting and under fitting the training data is a problem known as the bias-variance trade off.**

Selection of k



Larger k



Smaller k

- Typically, k is set somewhere between 3 and 10. One common practice is to set k equal to the square root of the number of training examples.
- In the food classifier we developed previously, we might set k = 4, because there were 15 example ingredients in the training data and the square root of 15 is 3.87.

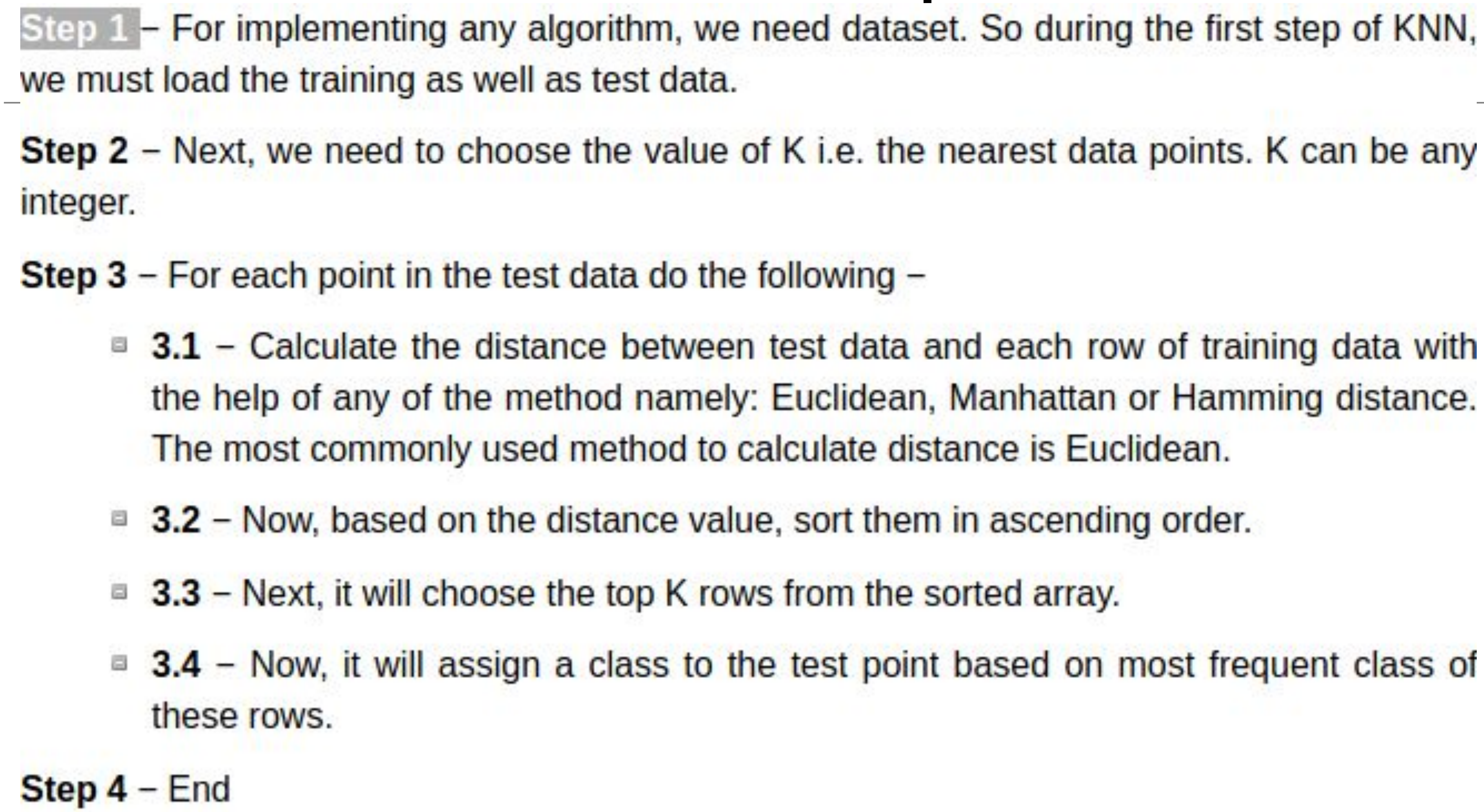
Preparing data for use with kNN

- ~~Features are typically transformed to a standard range prior to applying the kNN algorithm.~~
- The traditional method of rescaling features for kNN is **min-max normalization.**
- This process transforms a feature such that all of its values fall in a range between 0
- and 1. The formula for normalizing a feature is as follows.
- Essentially, the formula subtracts the minimum of feature X from each value and divides by the range of X:
- $X \text{ new} = X - \min (X) / \max (X) - \min (X)$

- The Euclidean distance formula is not defined for nominal data. Therefore, to calculate the distance between nominal features, we need to convert them into a numeric format.
- A typical solution utilizes dummy coding, where a value of 1 indicates one category, and 0 indicates the other.

- Another common transformation is called **z-score standardization**. The following formula subtracts the mean value of feature X and divides by the standard deviation of X:
- X new = X − μ/ σ = X − Mean (X)/StdDev (X)

kNN -steps



Step 1 – For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

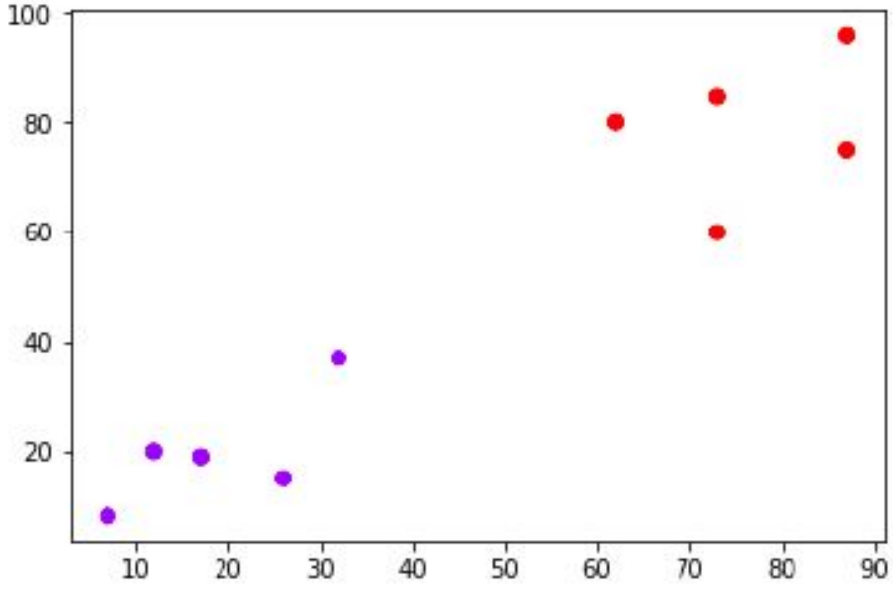
Step 2 – Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

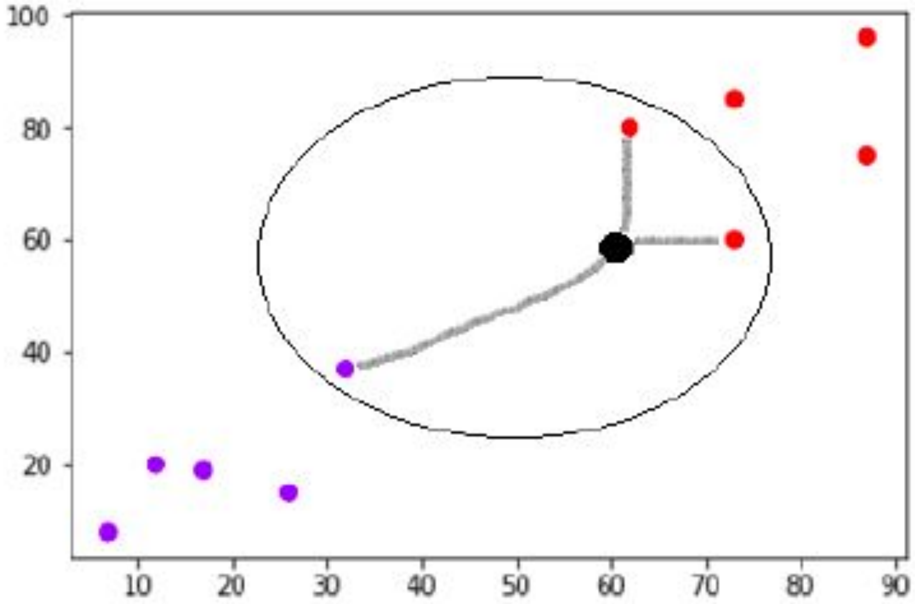
Step 3 – For each point in the test data do the following –

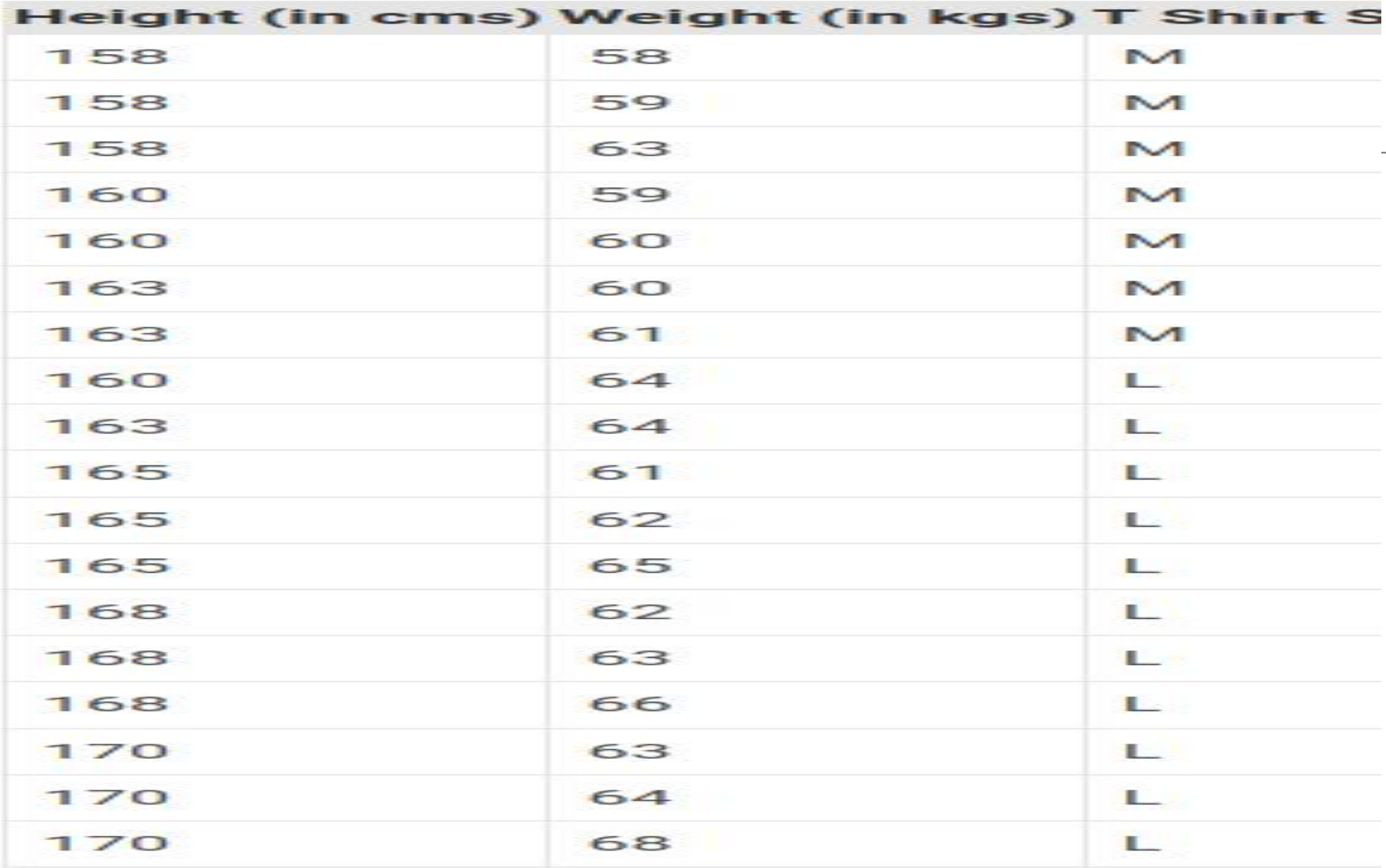
- **3.1** – Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
- **3.2** – Now, based on the distance value, sort them in ascending order.
- **3.3** – Next, it will choose the top K rows from the sorted array.
- **3.4** – Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 – End

KNN example


100
80
60
40
20
10
20
30
40
50
60
70
80
90


100
80
60
40
20
10
20
30
40
50
60
70
80
90



Height (in cms)	Weight (in kgs)	T Shirt S
158	58	M
158	59	M
158	63	M
160	59	M
160	60	M
163	60	M
163	61	M
160	64	L
163	64	L
165	61	L
165	62	L
165	65	L
168	62	L
168	63	L
168	66	L
170	63	L
170	64	L
170	68	L

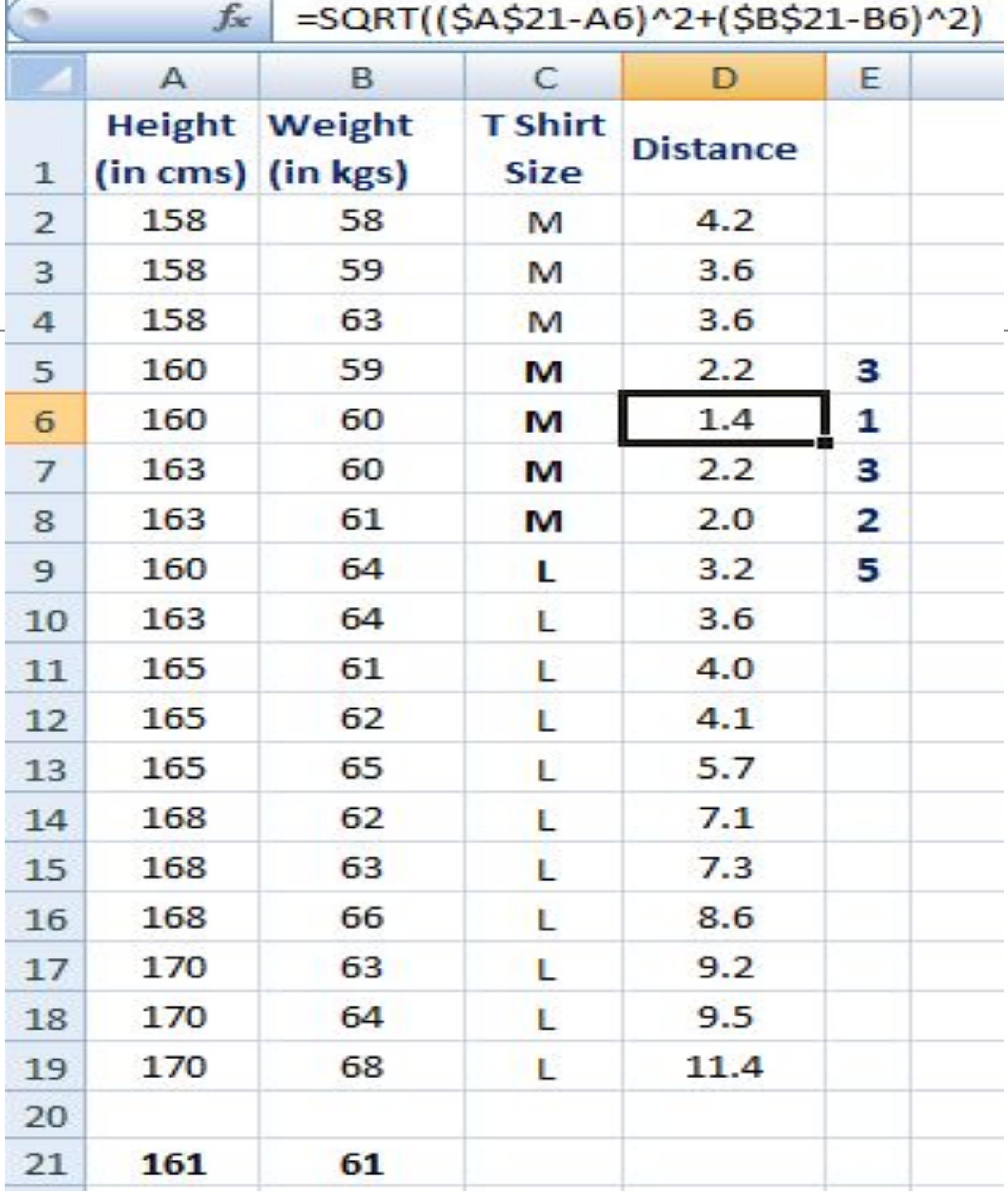
Step 1

- **Calculate Similarity based on distance function**
- The idea to use distance measure is to find the distance (similarity) between new sample and training cases and then finds the k-closest customers to new customer in terms of height and weight.
- New customer named 'Monica' has height 161cm and weight 61kg.

- Euclidean distance between first observation and new observation (monica) is as follows -
- =SQRT((161-158)^2+(61-58)^2)
- we will **calculate distance of all the training cases with new case and calculates the rank in terms of distance.**
- The smallest distance value will be ranked 1 and considered as nearest neighbour

Step 2

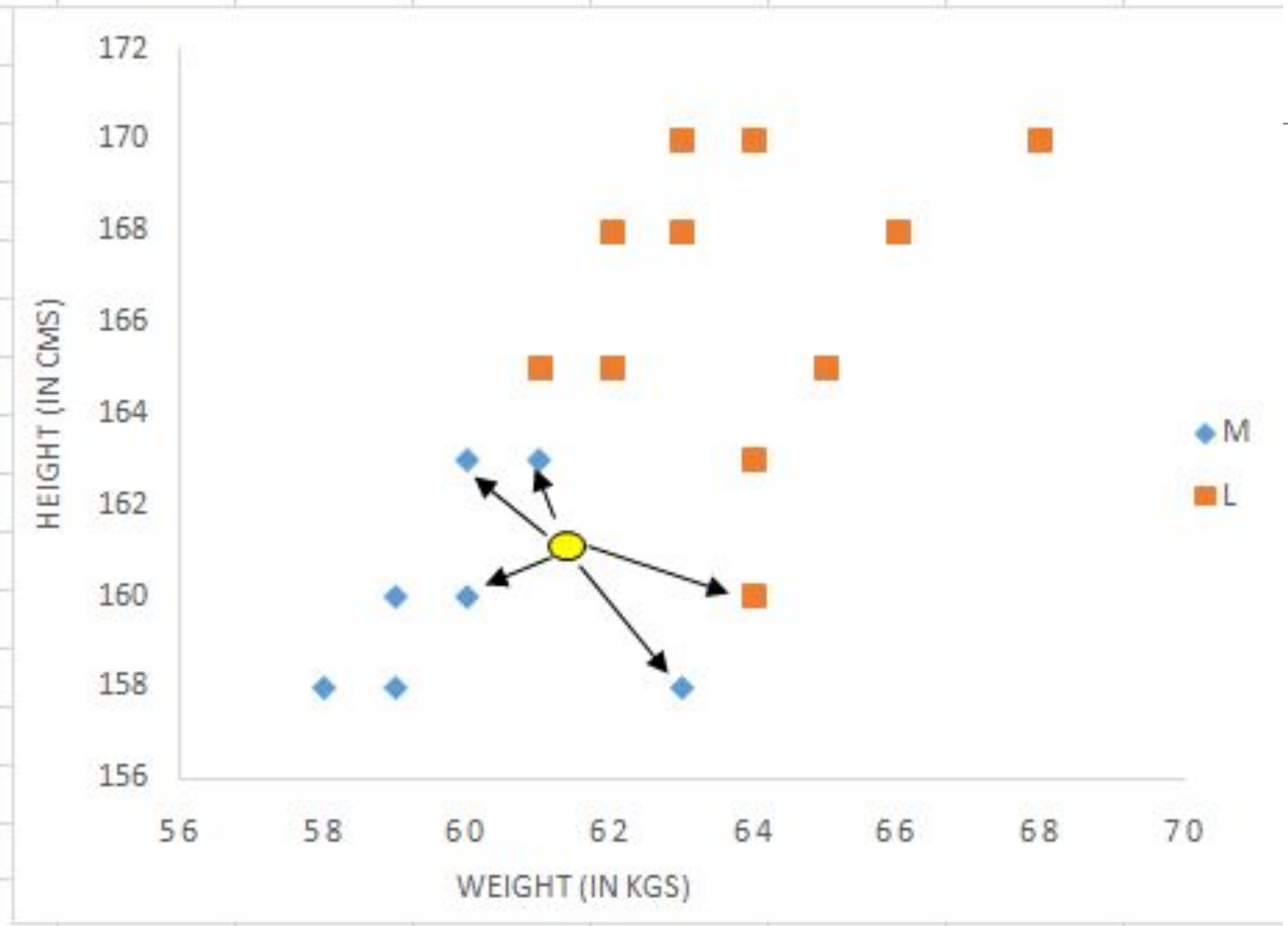
- **Find K-Nearest Neighbours**
- Let k be 5. Then the algorithm searches for the 5 customers closest to Monica, i.e. most similar to Monica in terms of attributes, and see what categories those 5 customers were in.
- If 4 of them had ‘Medium T shirt sizes’ and 1 had ‘Large T shirt size’ then your best guess for Monica is ‘Medium T shirt Find K-Nearest Neighbours

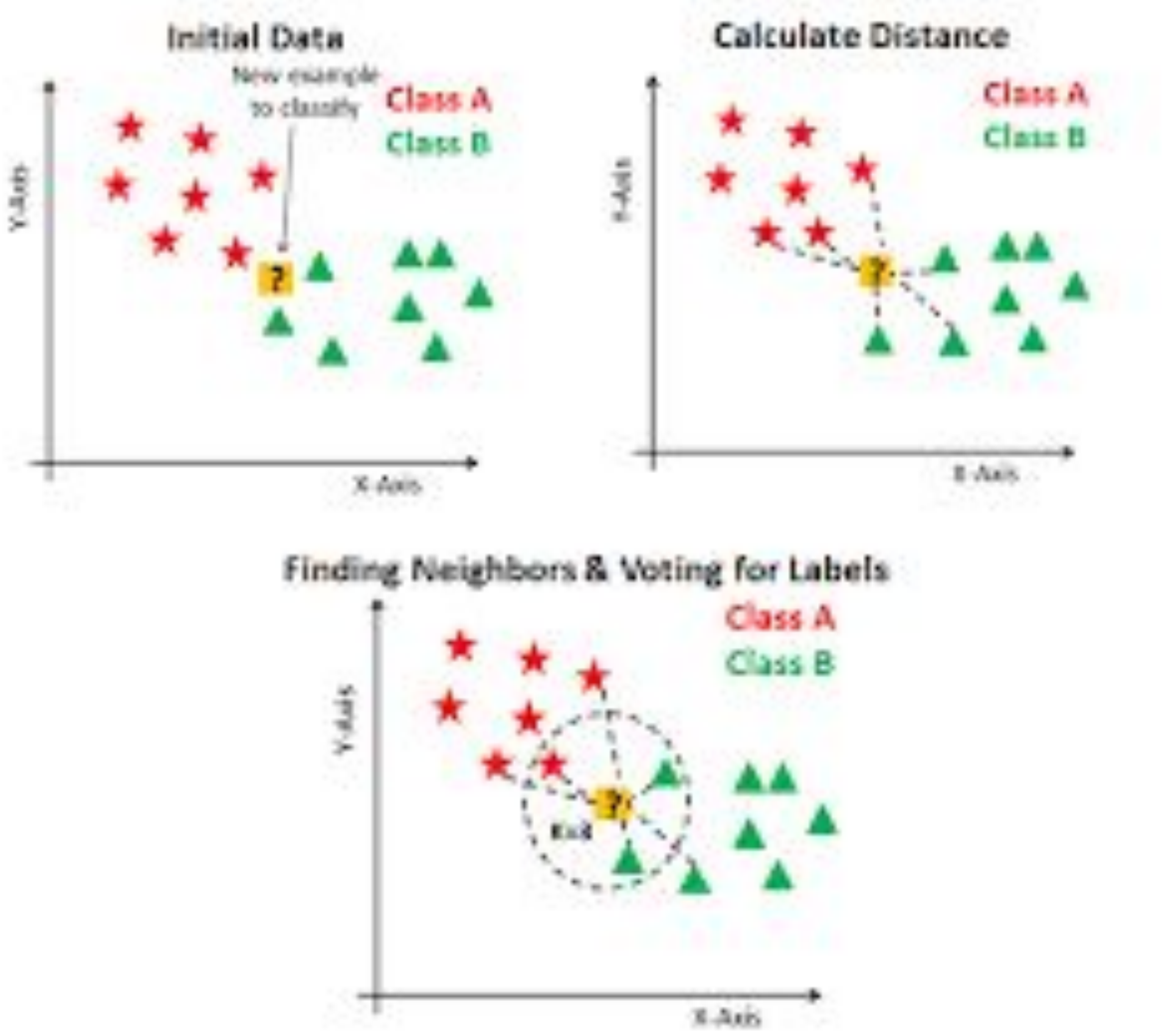


```
=SQRT(($A$21-A6)^2+($B$21-B6)^2)
```

	A	B	C	D	E
1	Height (in cms)	Weight (in kgs)	T Shirt Size	Distance	
2	158	58	M	4.2	
3	158	59	M	3.6	
4	158	63	M	3.6	
5	160	59	**M**	2.2	**3**
6	160	60	**M**	1.4	**1**
7	163	60	**M**	2.2	**3**
8	163	61	**M**	2.0	**2**
9	160	64	**L**	3.2	**5**
10	163	64	L	3.6	
11	165	61	L	4.0	
12	165	62	L	4.1	
13	165	65	L	5.7	
14	168	62	L	7.1	
15	168	63	L	7.3	
16	168	66	L	8.6	
17	170	63	L	9.2	
18	170	64	L	9.5	
19	170	68	L	11.4	
20					
21	**161**	**61**			

- In the graph below, binary dependent variable (T-shirt size) is displayed in blue and orange color. 'Medium T-shirt size' is in blue color and 'Large T-shirt size' in orange color.
- New customer information is exhibited in yellow circle.
- Four blue highlighted data points and one orange highlighted data point are close to yellow circle. so the prediction for the new case is blue highlighted data point which is Medium T-shirt size.


HEIGHT (IN CMS)
172
170
168
166
164
162
160
158
156
WEIGHT (IN KGS)
56
58
60
62
64
66
68
70
◆ M
■ L



Initial Data

Calculate Distance

Finding Neighbors & Voting for Labels