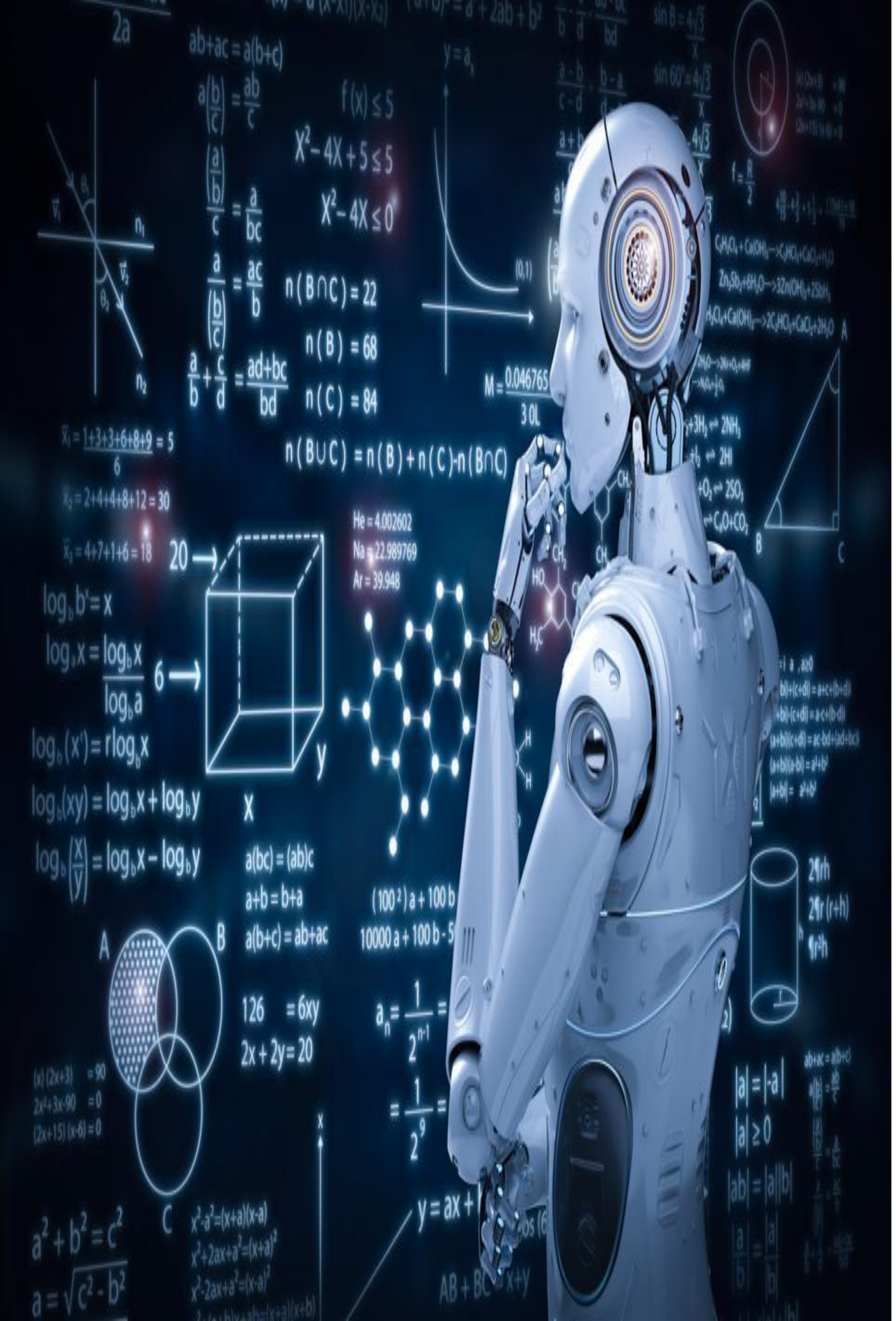


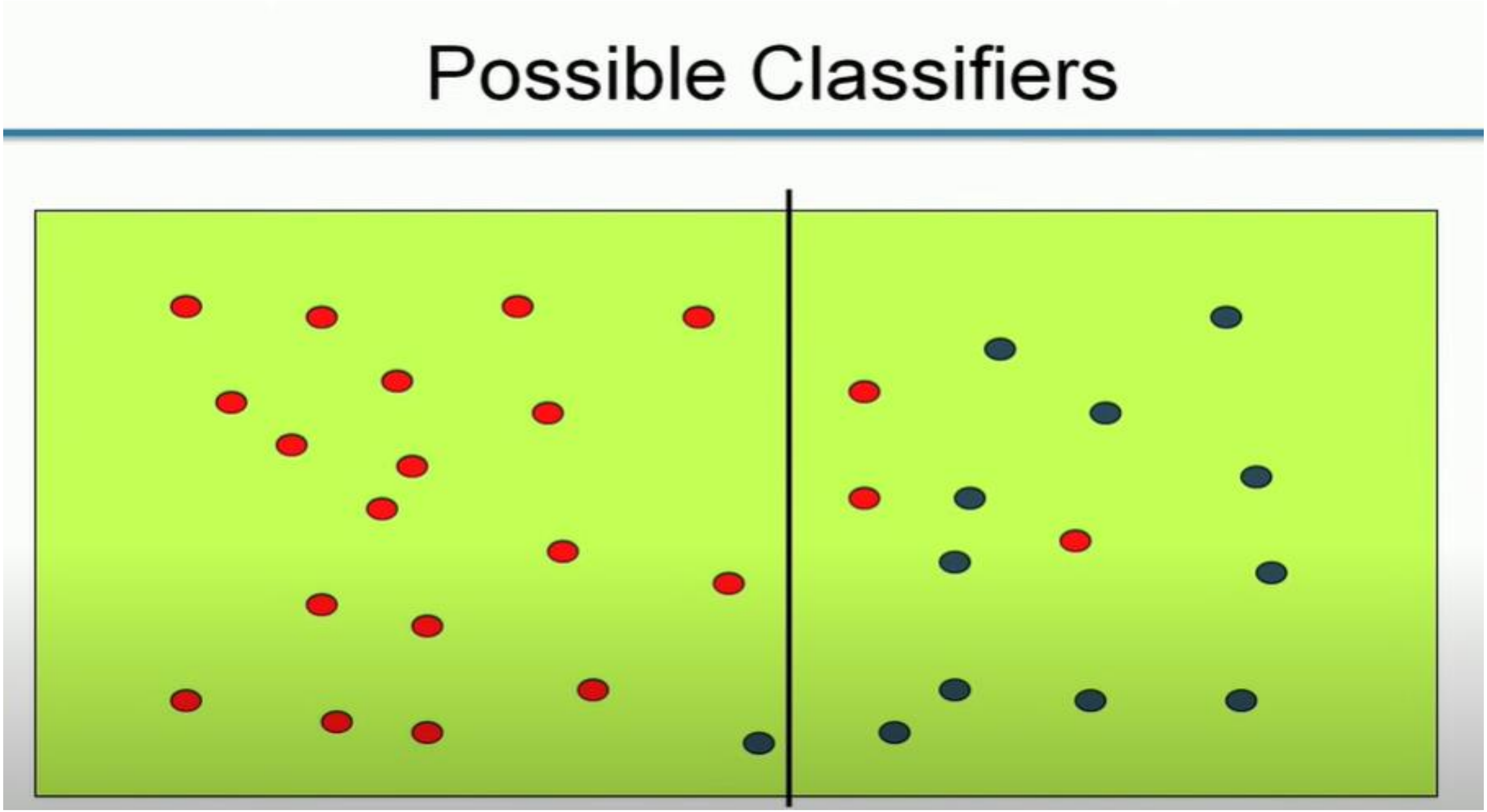
Practical Machine Learning

Day 9: Mar22 DBDA

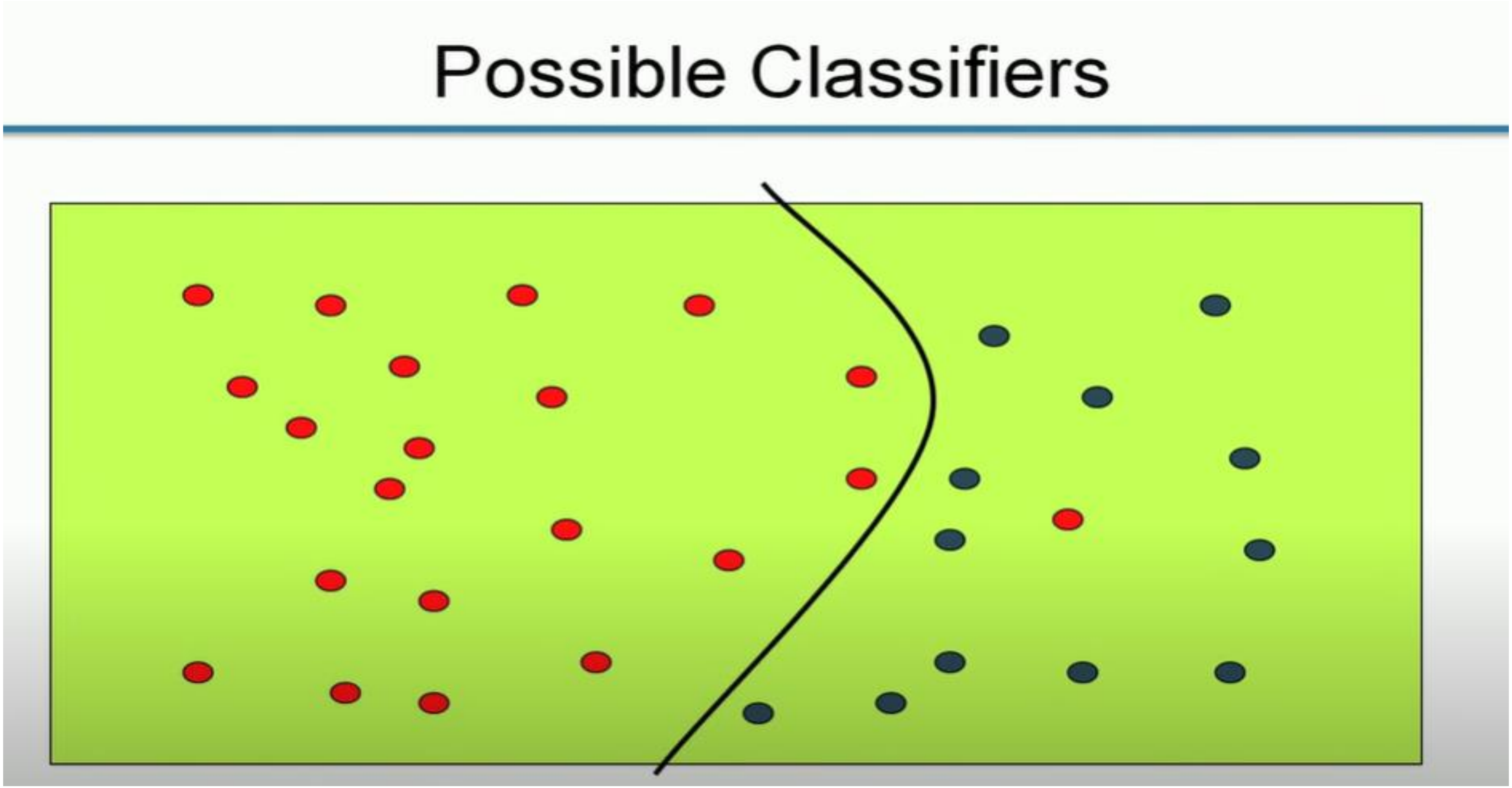
Kiran Waghmare

Agenda

- Classification Algorithm
- kNN
- Naïve Bayes

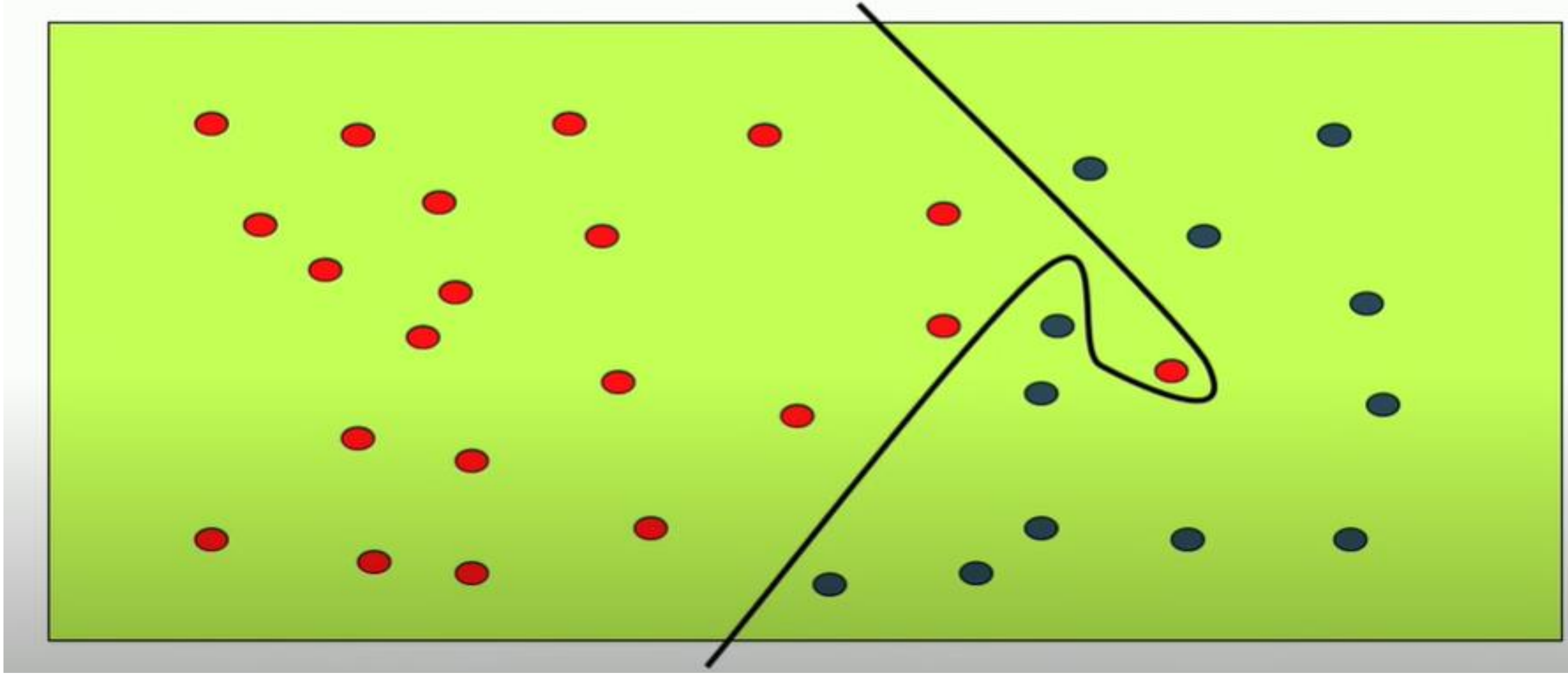


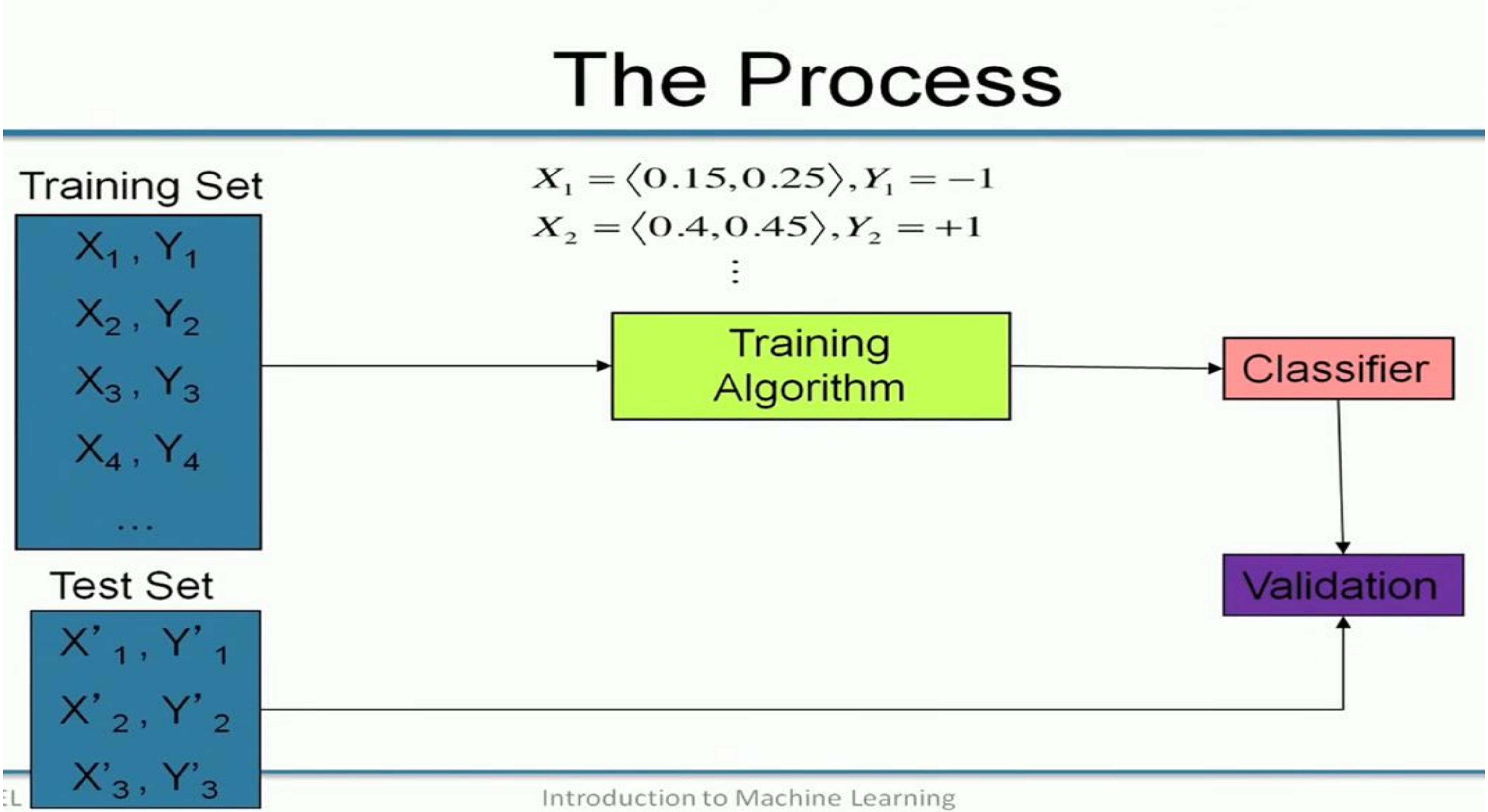
Possible Classifiers



Possible Classifiers

Possible Classifiers





The Process

Training Set

X_1, Y_1

X_2, Y_2

X_3, Y_3

X_4, Y_4

...

$$X_1 = \langle 0.15, 0.25 \rangle, Y_1 = -1$$

$$X_2 = \langle 0.4, 0.45 \rangle, Y_2 = +1$$

$$\vdots$$

Training Algorithm

Classifier

Validation

Test Set

X'_1, Y'_1

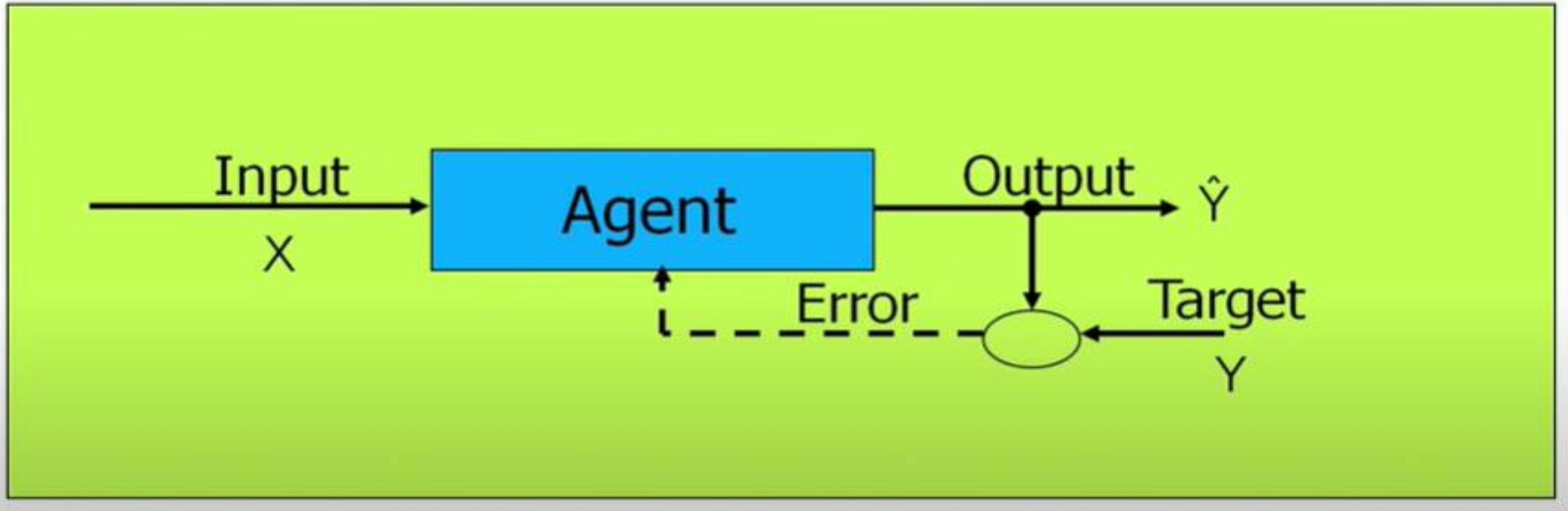
X'_2, Y'_2

X'_3, Y'_3

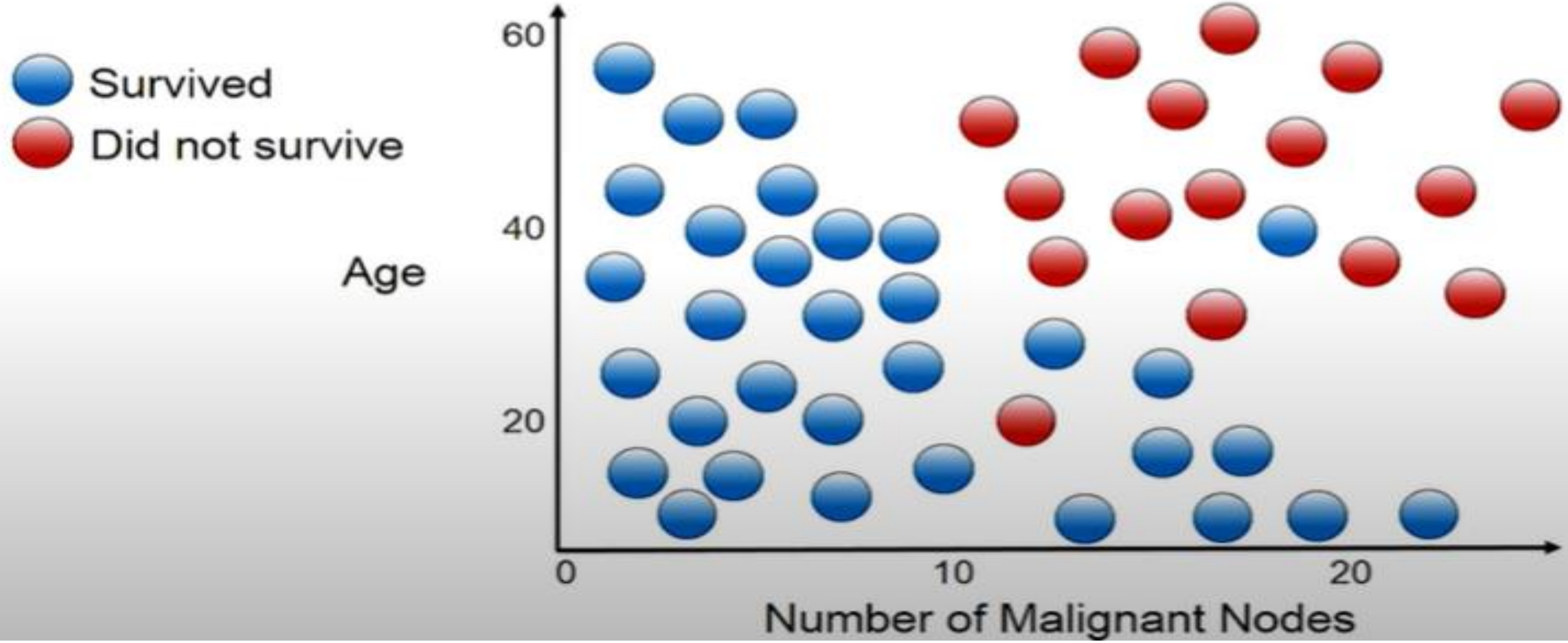
EL

Introduction to Machine Learning

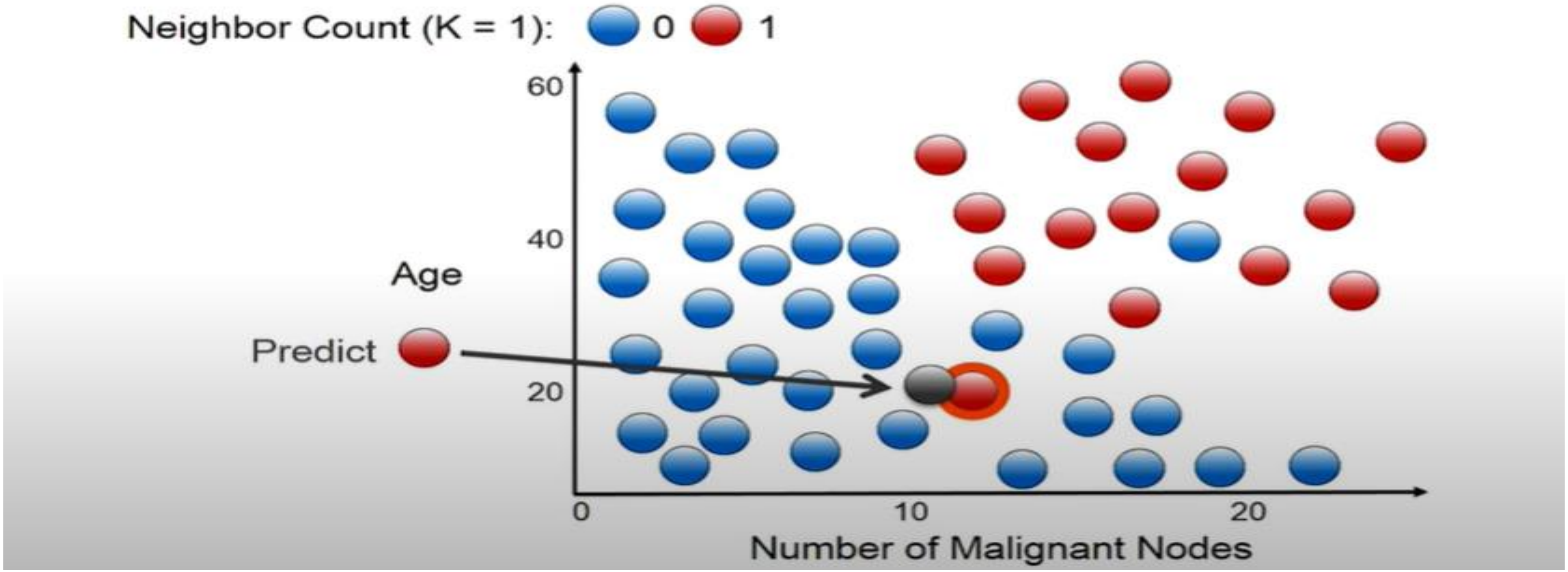
Training



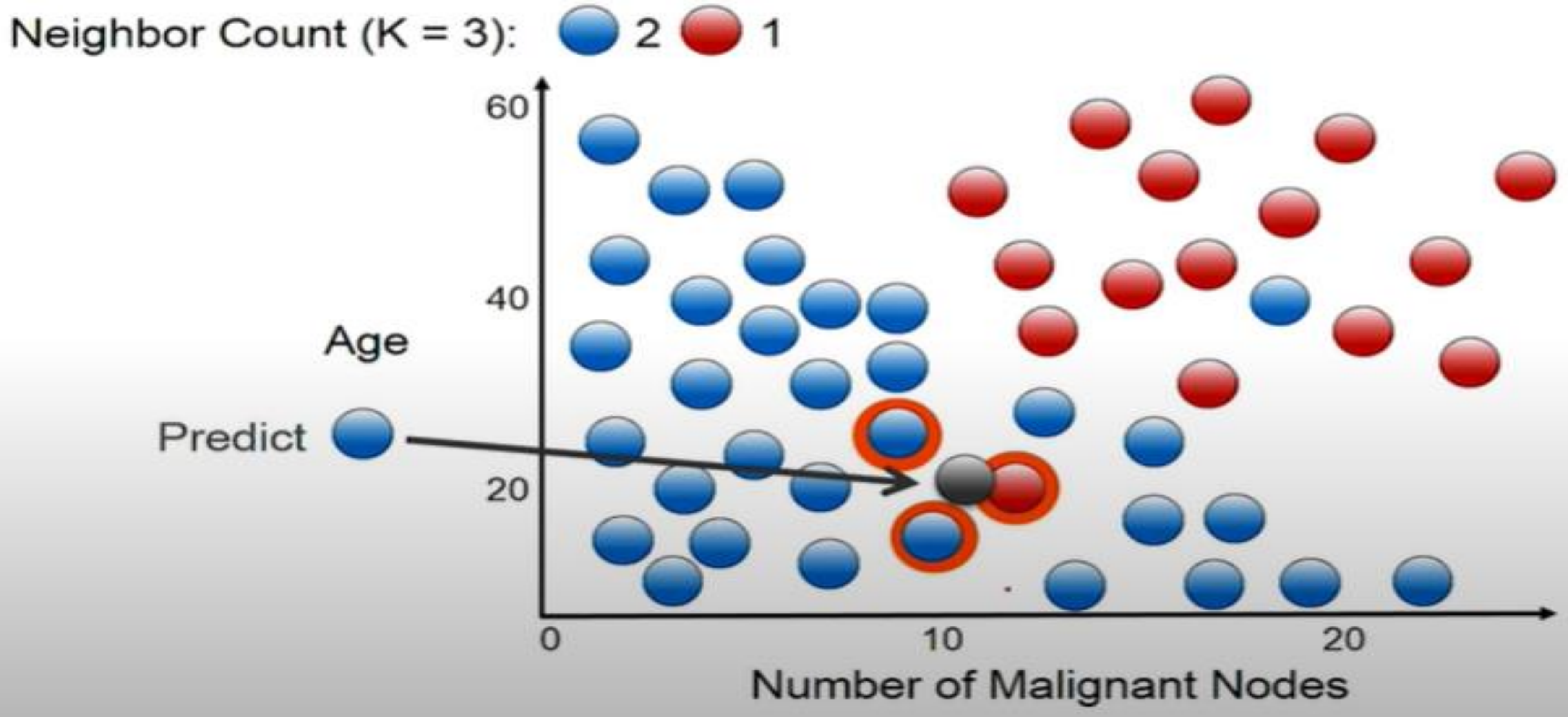
K-Nearest Neighbour- Classification



.. Classification



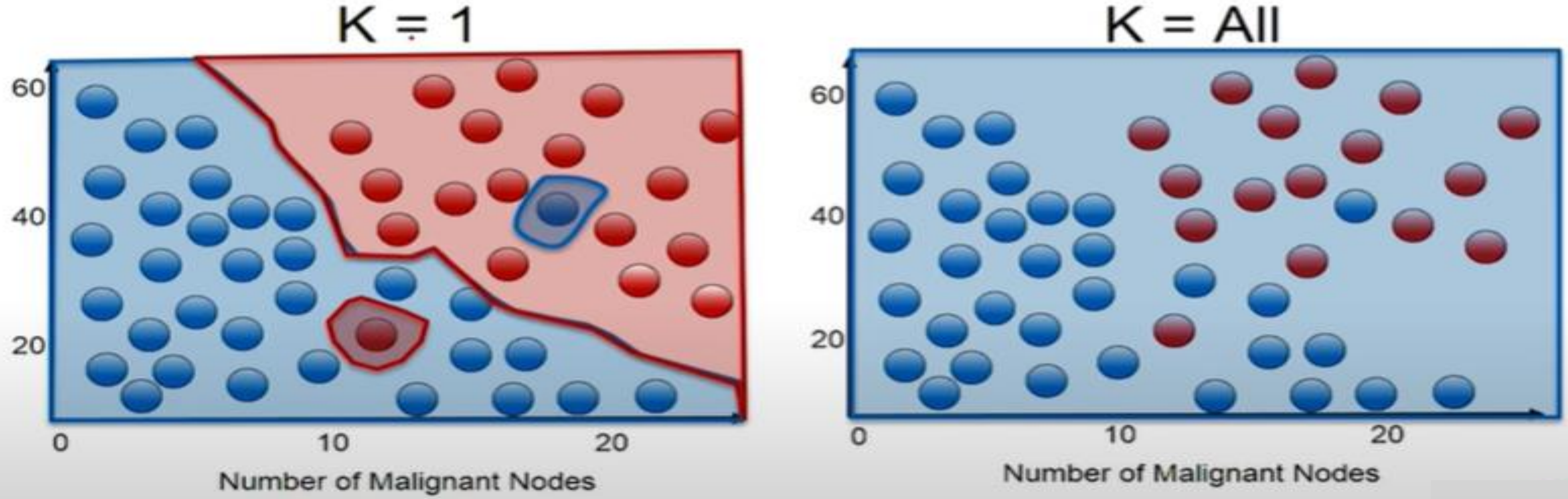
...Classification



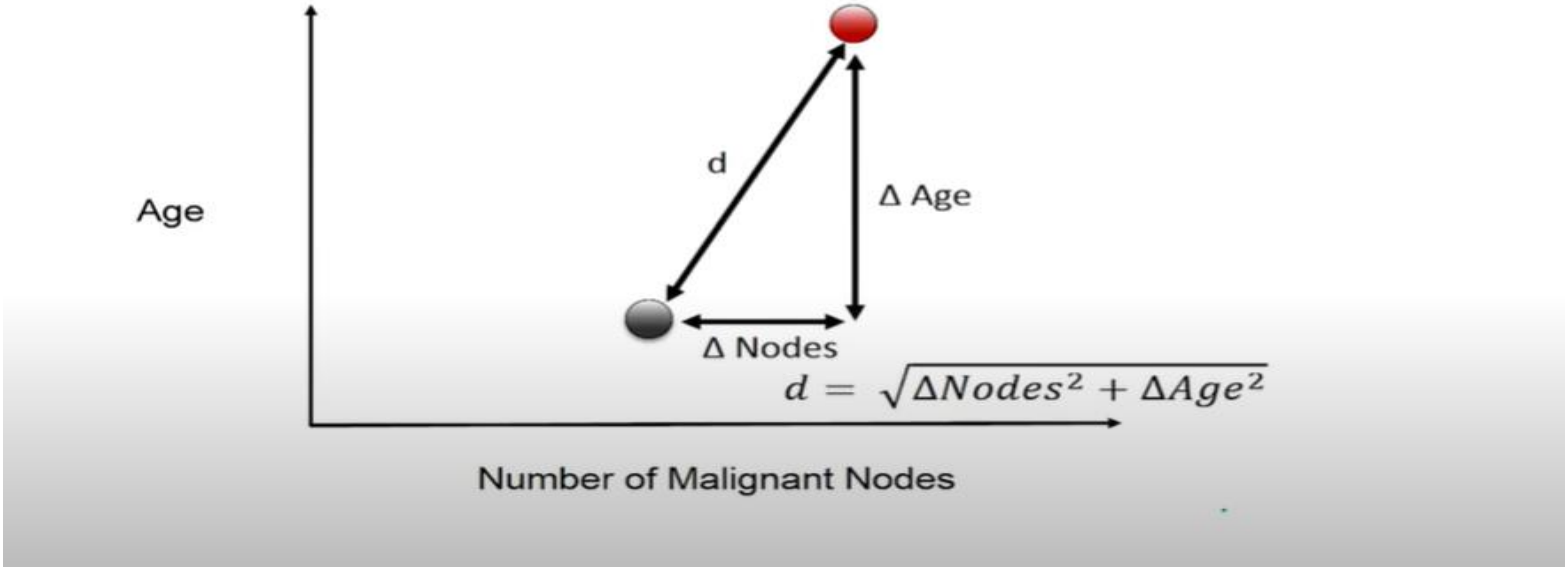
KNN parameters

- K – nearest neighbours
- Distance metric

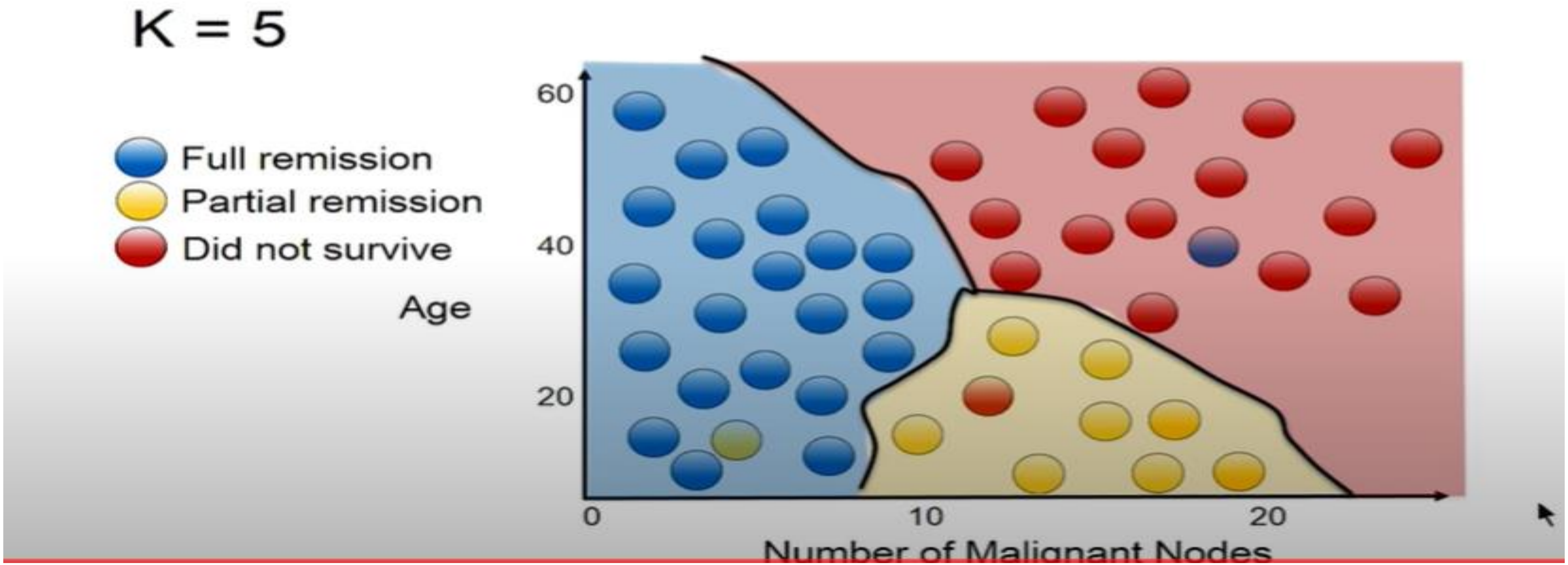
Choosing K

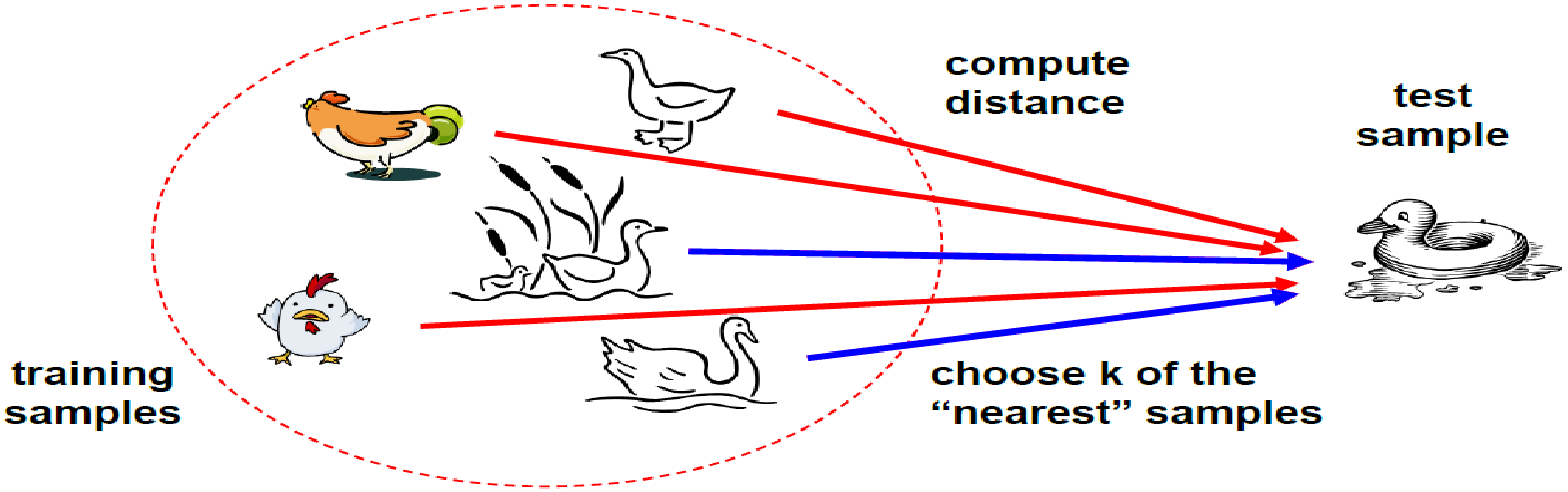

K = 1
60
40
20
0
10
20
Number of Malignant Nodes
K = All
60
40
20
0
10
20
Number of Malignant Nodes

Distance Metric- Euclidean Distance


Age
d
Δ Age
Δ Nodes
$d = \sqrt{\Delta Nodes^2 + \Delta Age^2}$
Number of Malignant Nodes

Multiple Classes



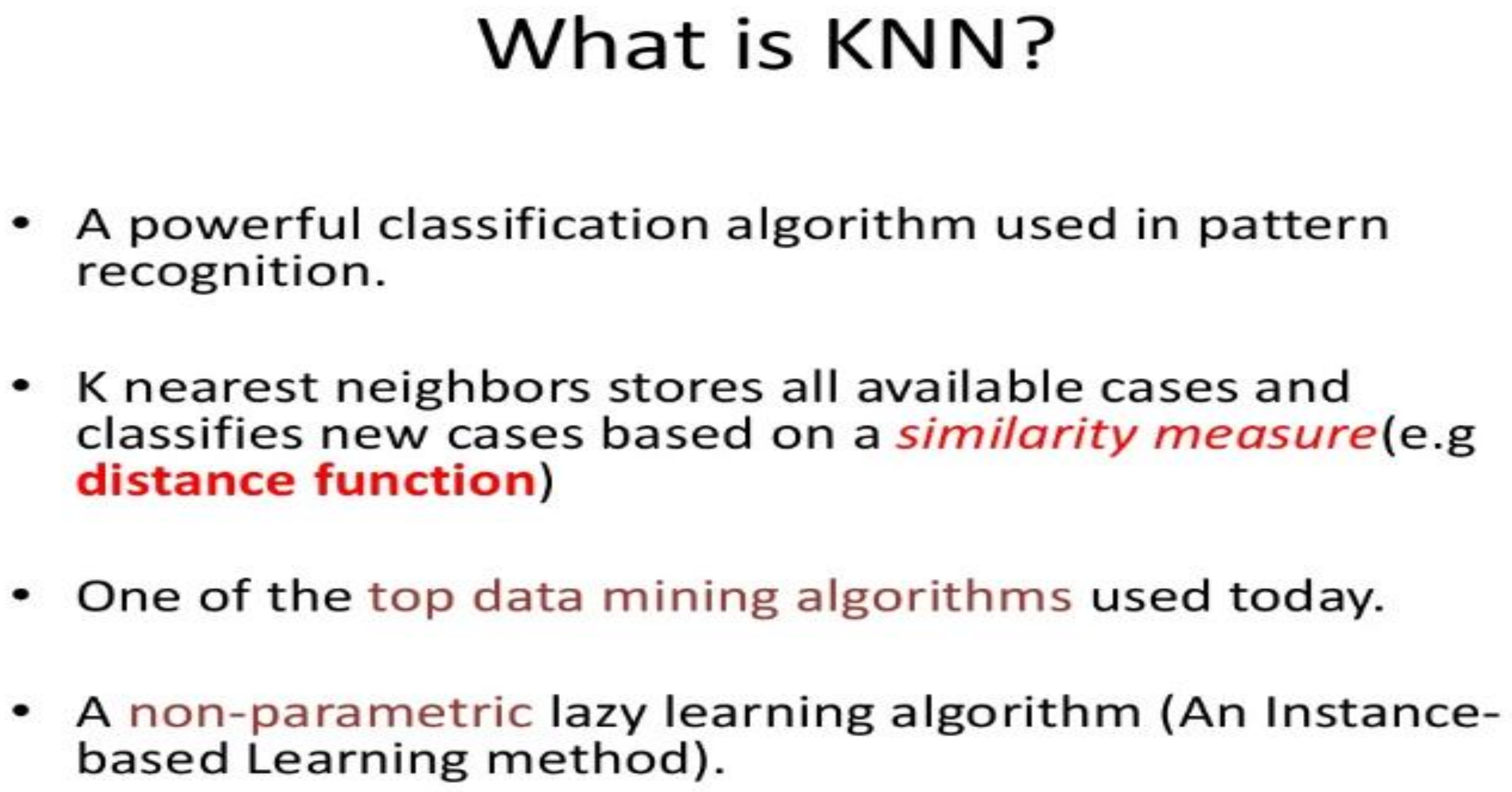


compute distance

test sample

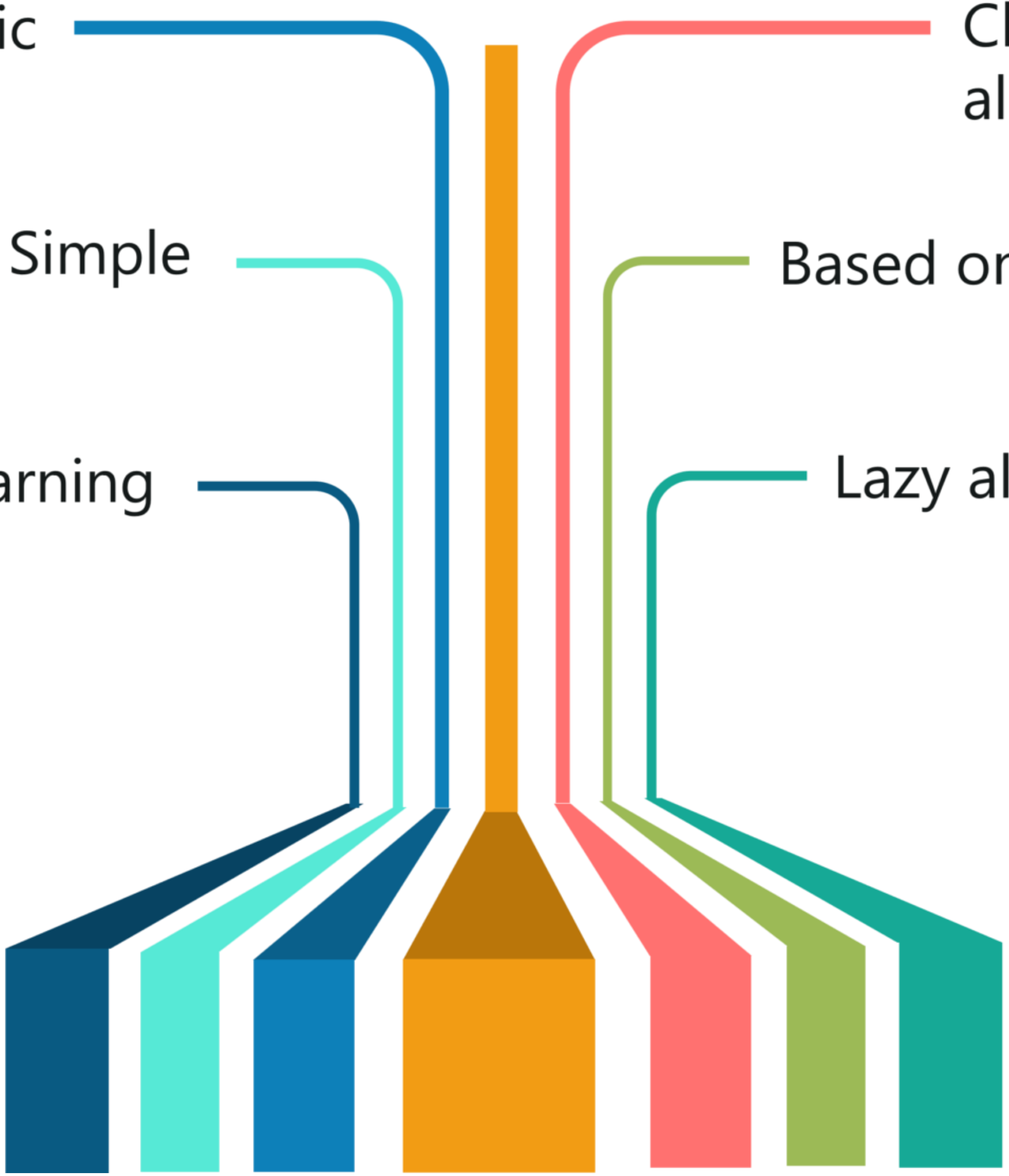
training samples

choose k of the “nearest” samples



What is KNN?

- A powerful classification algorithm used in pattern recognition.
- K nearest neighbors stores all available cases and classifies new cases based on a *similarity measure*(e.g **distance function**)
- One of the top data mining algorithms used today.
- A non-parametric lazy learning algorithm (An Instance-based Learning method).



Non-parametric

Simple

Supervised Learning algorithm

Classification & regression algorithm

Based on feature similarity

Lazy algorithm

Nearest neighbor classification

- *k*-Nearest neighbor classifier is a lazy learner.
  - Does not build model explicitly.
  - Unlike eager learners such as decision tree induction and rule-based systems.
  - Classifying unknown samples is relatively expensive.
- *k*-Nearest neighbor classifier is a local model, vs. global models of linear classifiers.
- *k*-Nearest neighbor classifier is a non-parametric model, vs. parametric models of linear classifiers.

Lazy learners

•‘**Lazy**’: Do not create a model of the training instances in advance

•When an instance arrives for testing, runs the algorithm to get the class prediction

•**Example, K** – nearest neighbor classifier

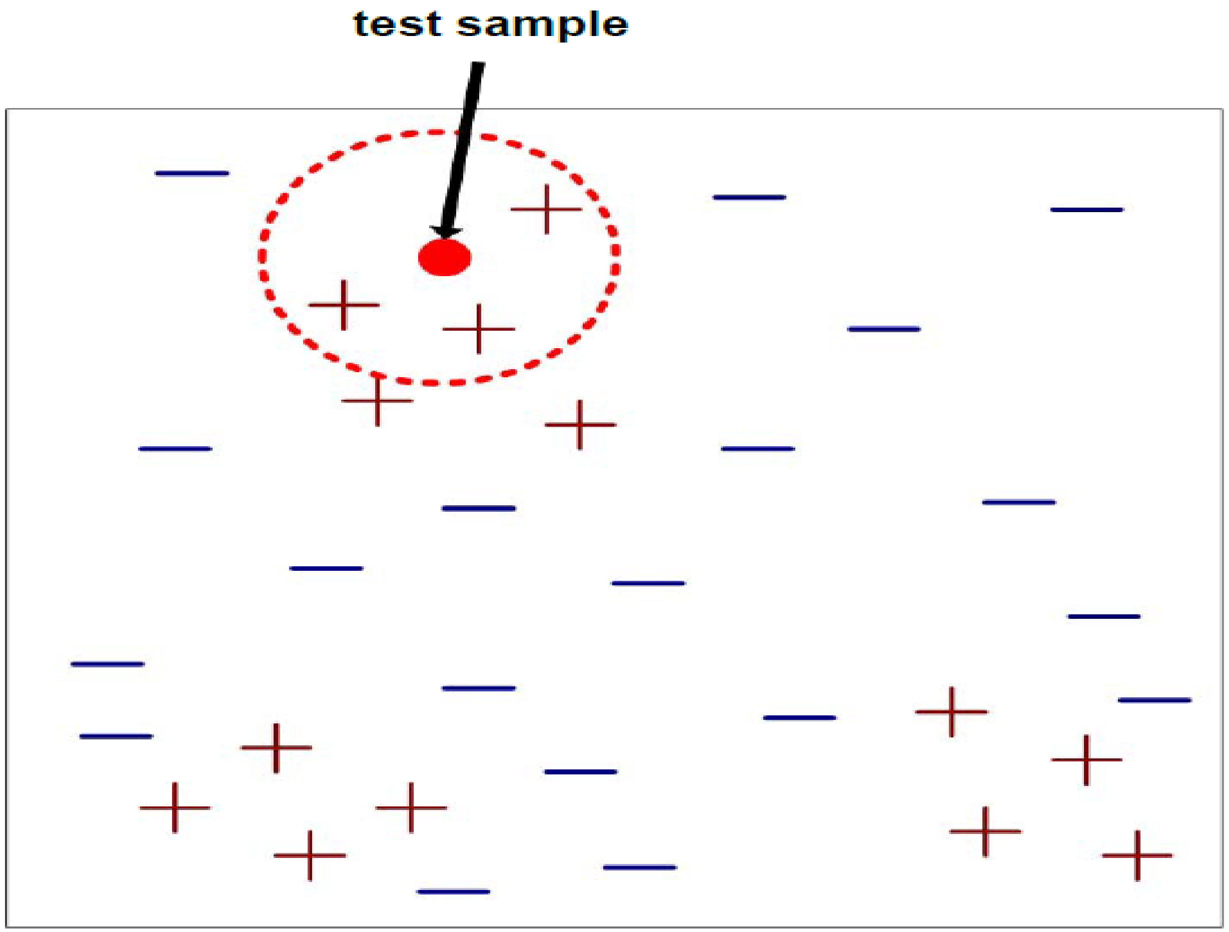
(K – NN classifier)

“One is known by the company one keeps”

Simple Analogy..

- Tell me about your friends(*who your neighbors are*) and *I will tell you who you are*.

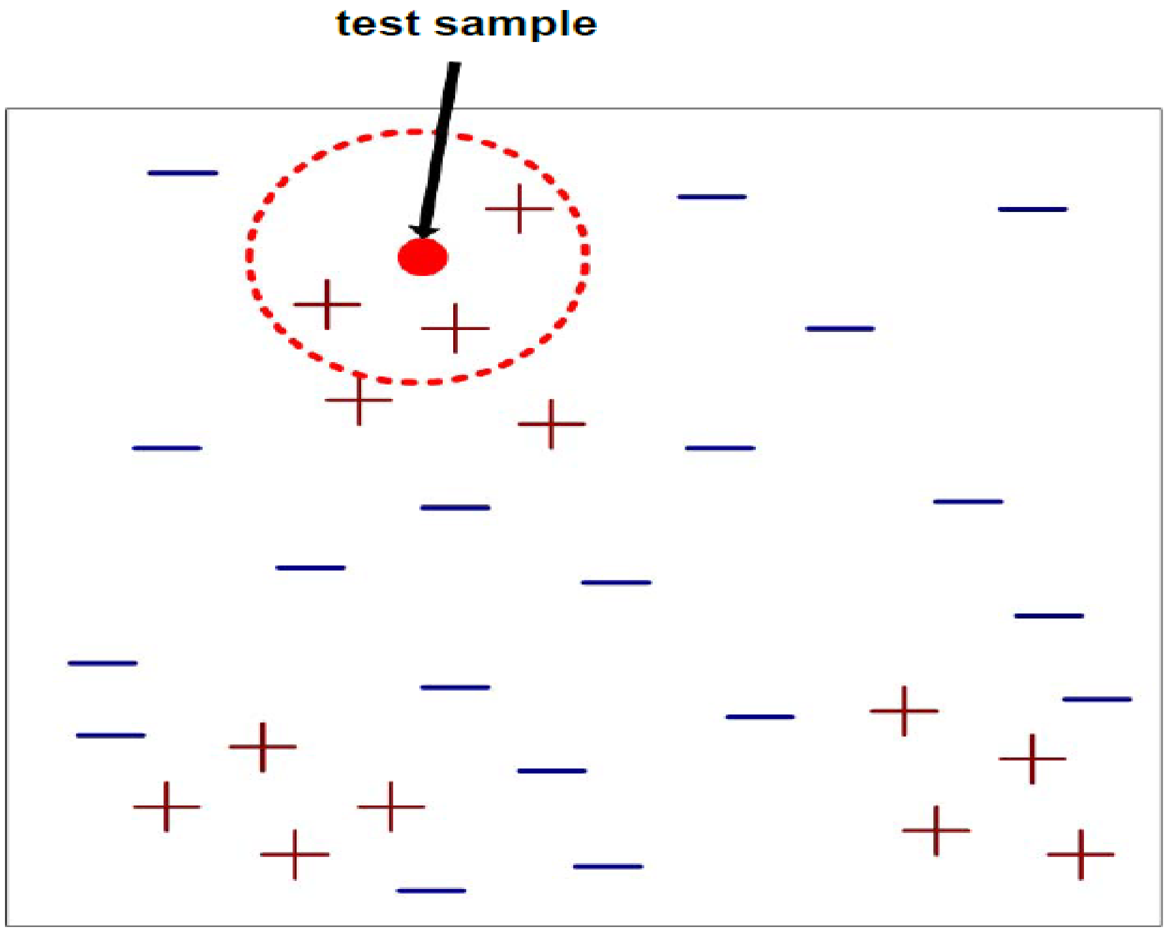
Nearest Neighbor Classifiers



Requires three inputs:

1. The set of stored samples
2. Distance metric to compute distance between samples
3. The value of *k*, the number of nearest neighbors to retrieve

Nearest Neighbor Classifiers

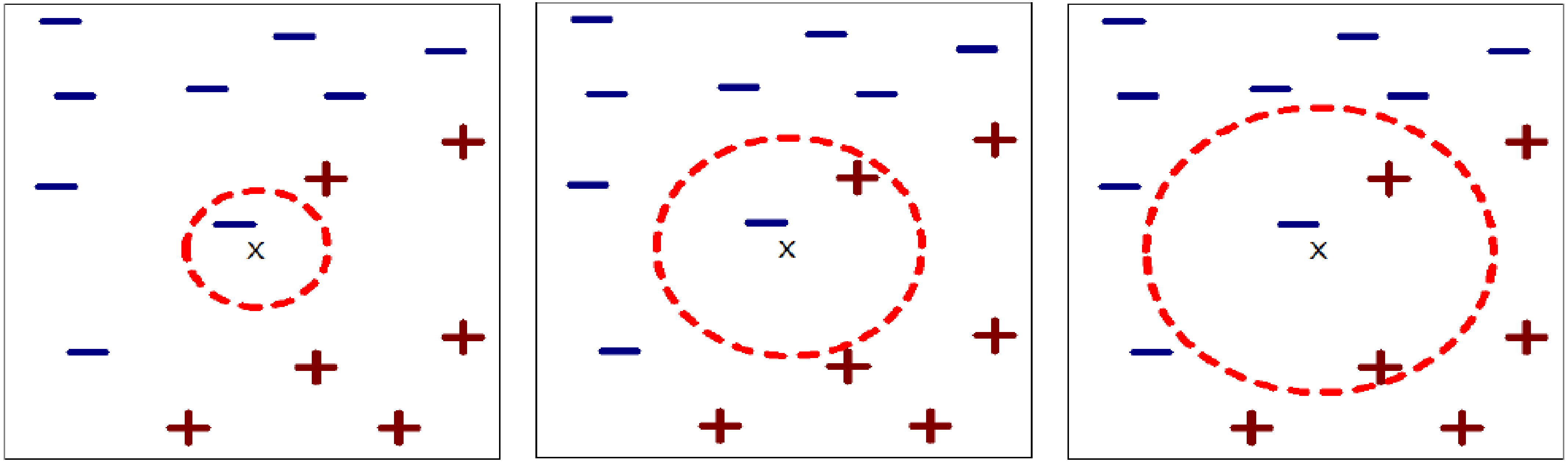


To classify test sample:

1. Compute distances to samples in training set
2. Identify *k* nearest neighbors
3. Use class labels of nearest neighbors to determine class label of test sample (e.g. by taking majority vote)

Definition of Nearest Neighbors

k-nearest neighbors of test sample x are training samples that have the *k* smallest distances to x



1-nearest neighbor 2-nearest neighbor 3-nearest neighbor

Distances for nearest neighbors

- Options for computing distance between two samples:
  - Euclidean distance

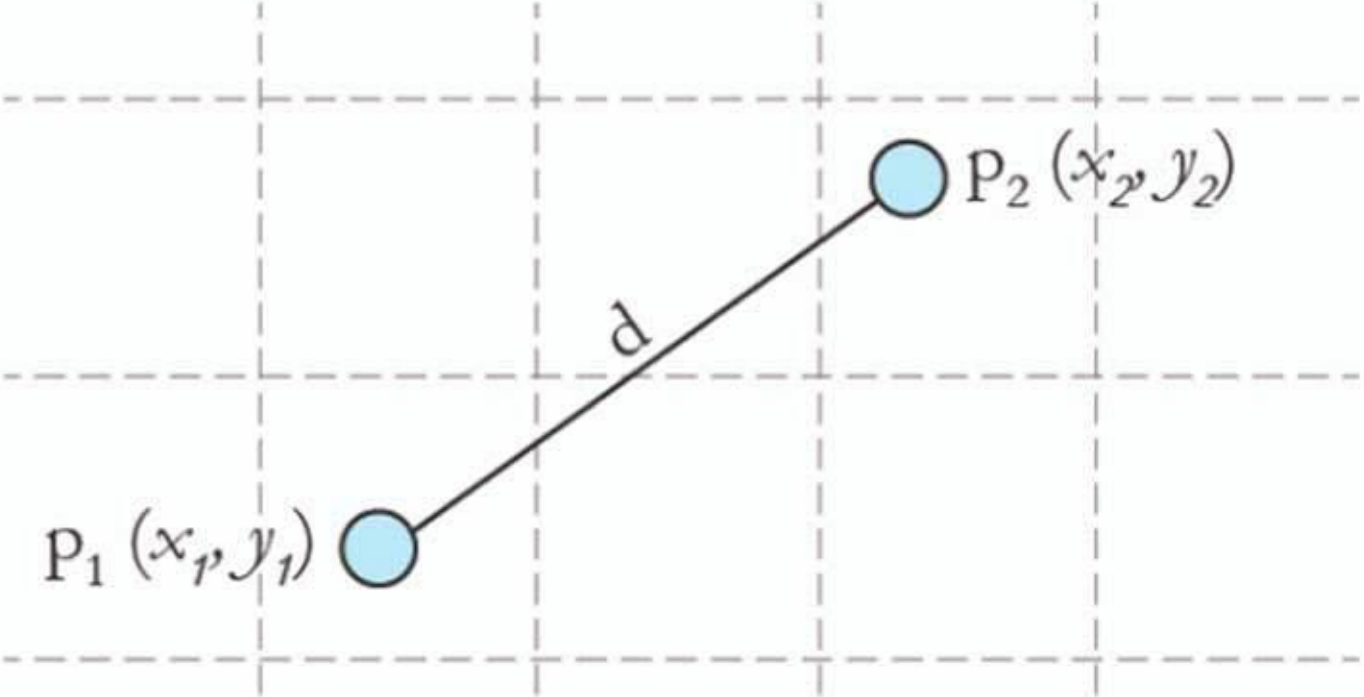
$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

  - Cosine similarity

$$d(\mathbf{x},\mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$$

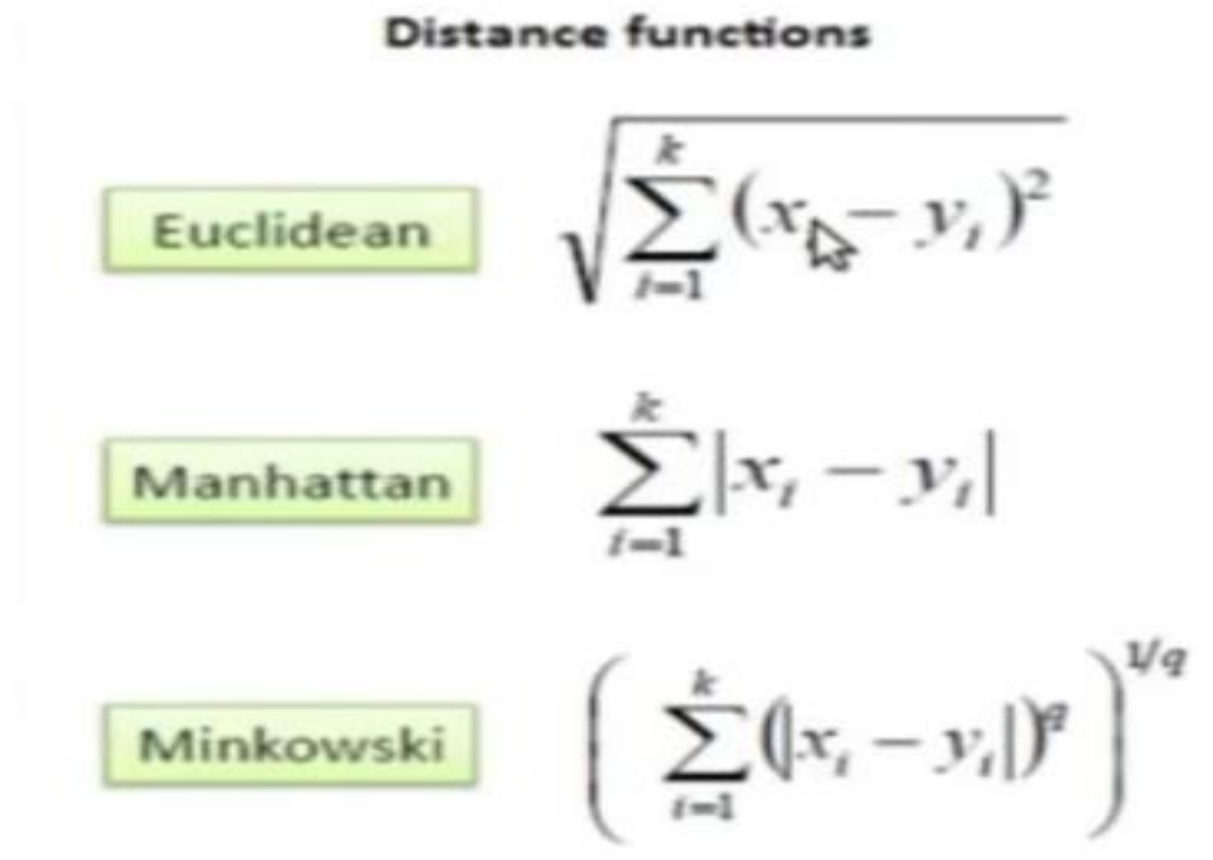
  - Hamming distance
  - String edit distance
  - Kernel distance
  - Many others

What is **Euclidean distance?**



$$\text{Euclidean distance (d)} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Distance measure for Continuous Variables



Distance functions

Euclidean $$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan $$\sum_{i=1}^{k}|x_i - y_i|$$

Minkowski $$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$

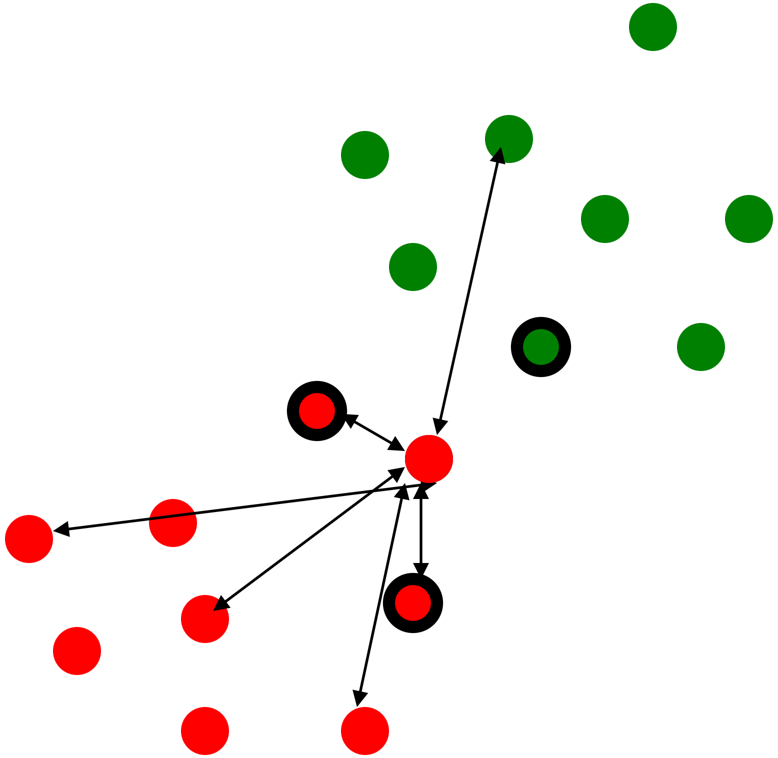
K-NN classifier schematic

For a test instance,

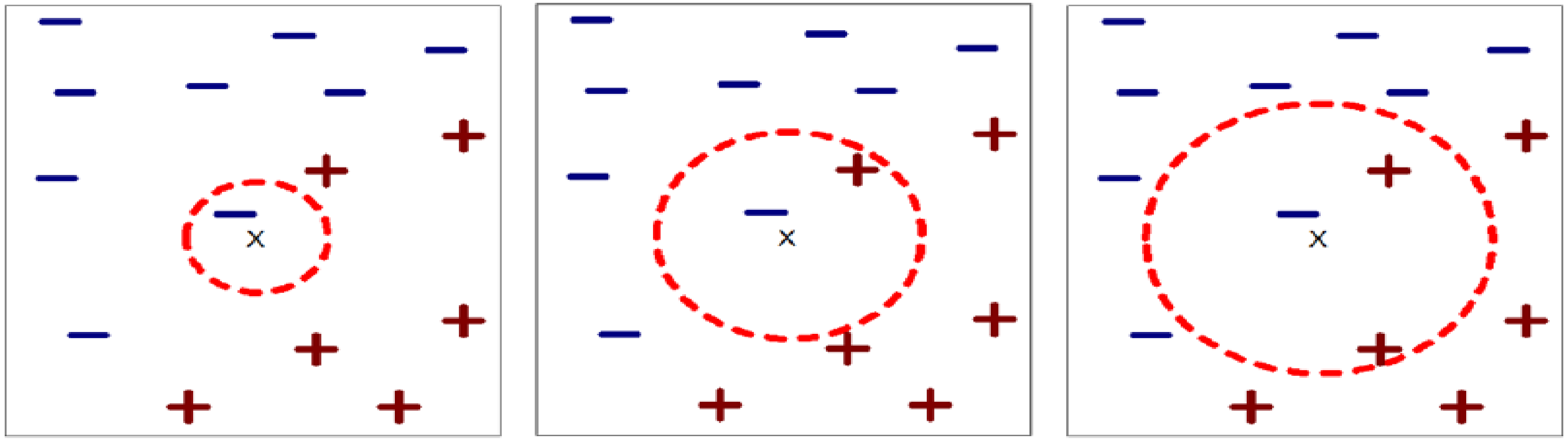
1) Calculate distances from training pts.
2) Find K-nearest neighbours (say, K = 3)
3) Assign class label based on majority

$$dist(\mathbf{X_1}, \mathbf{X_2}) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}.$$

$$v' = \frac{v - min_A}{max_A - min_A},$$



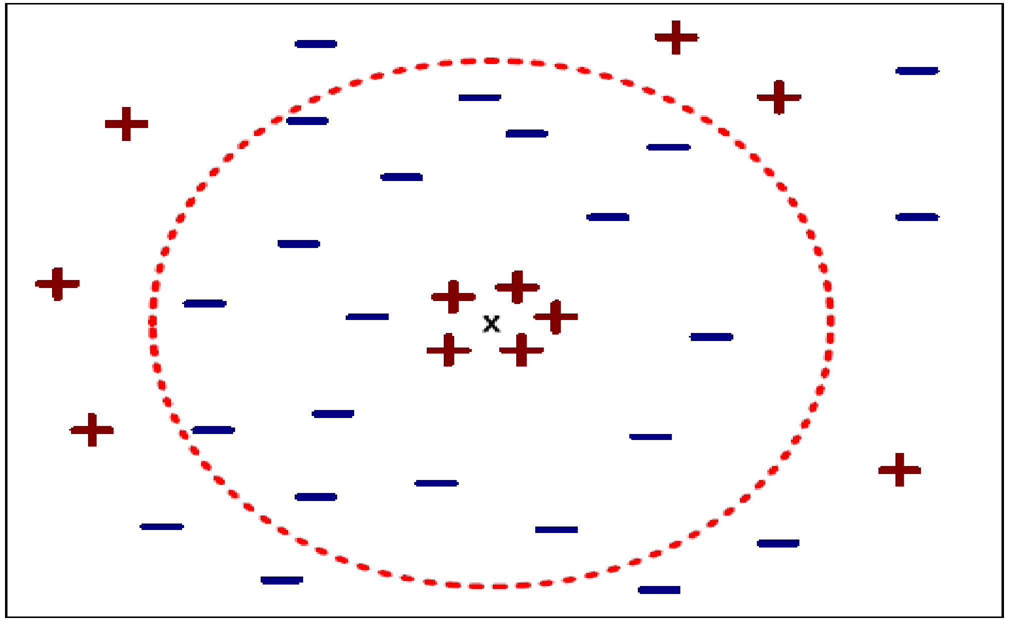
Predicting class from nearest neighbors



nearest neighbors	1	2	3
majority vote	–	?	+
distance-weighted vote	–	–	– or +

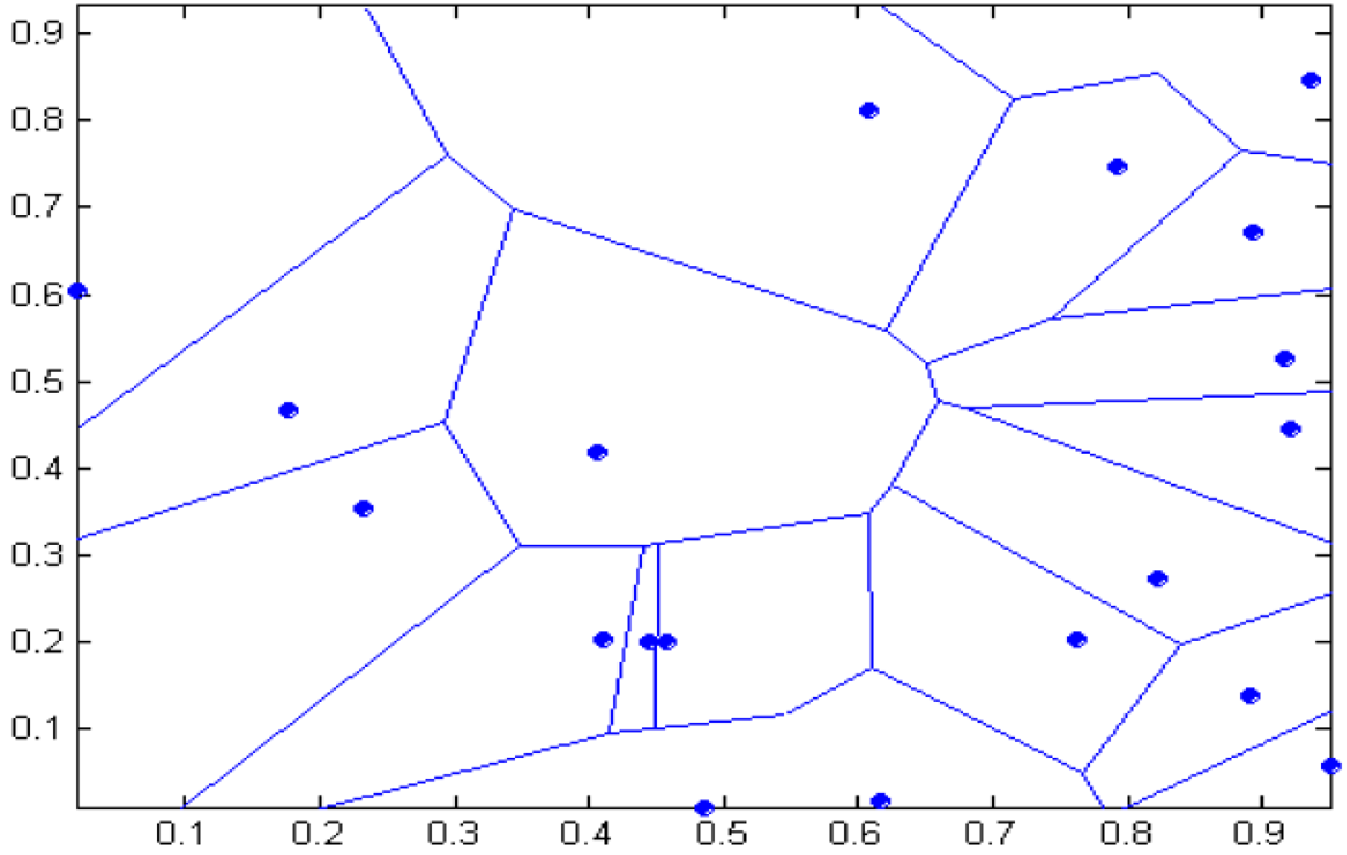
Predicting class from nearest neighbors

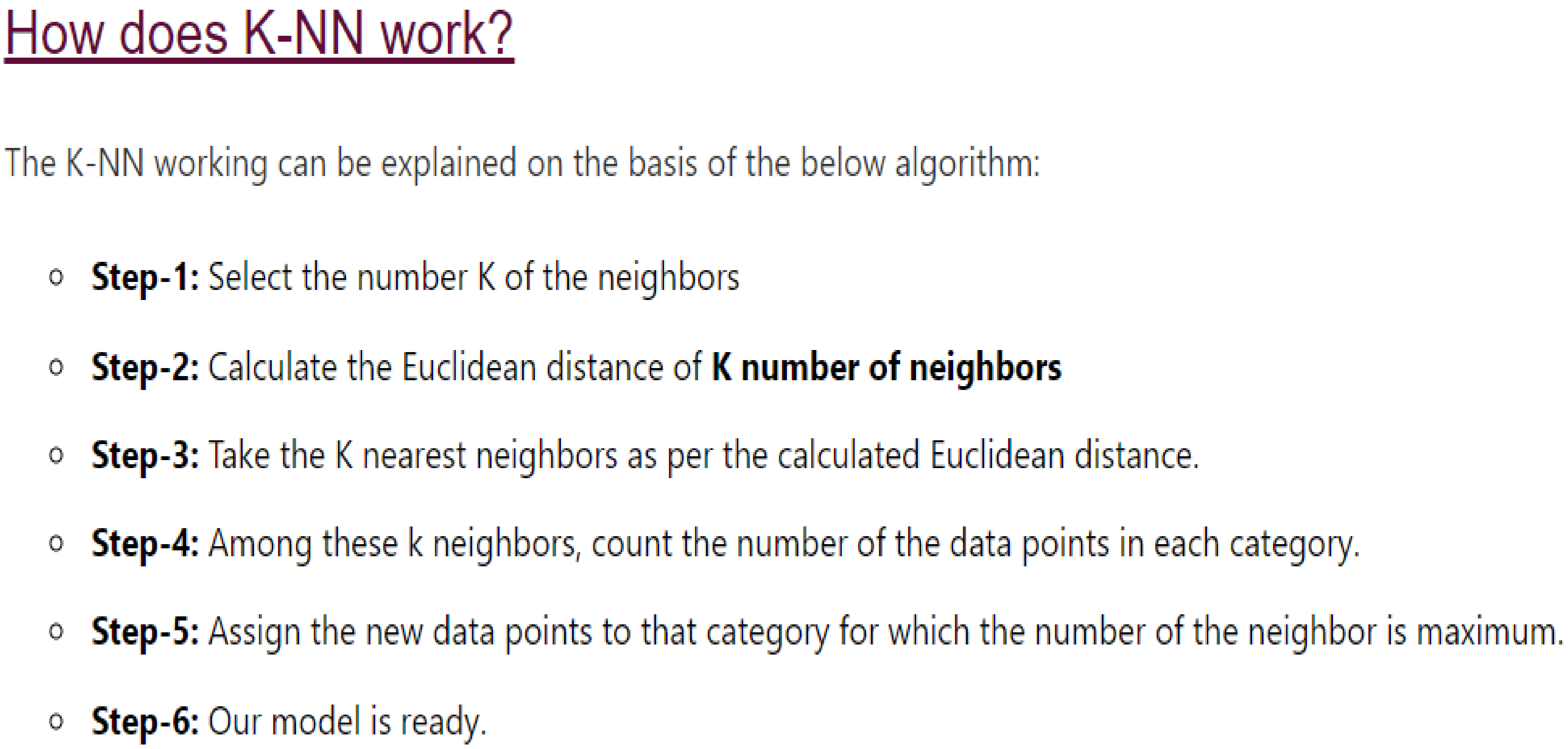
- Choosing the value of *k*:
  - If *k* is too small, sensitive to noise points
  - If *k* is too large, neighborhood may include points from other classes



1-nearest neighbor

Voronoi diagram

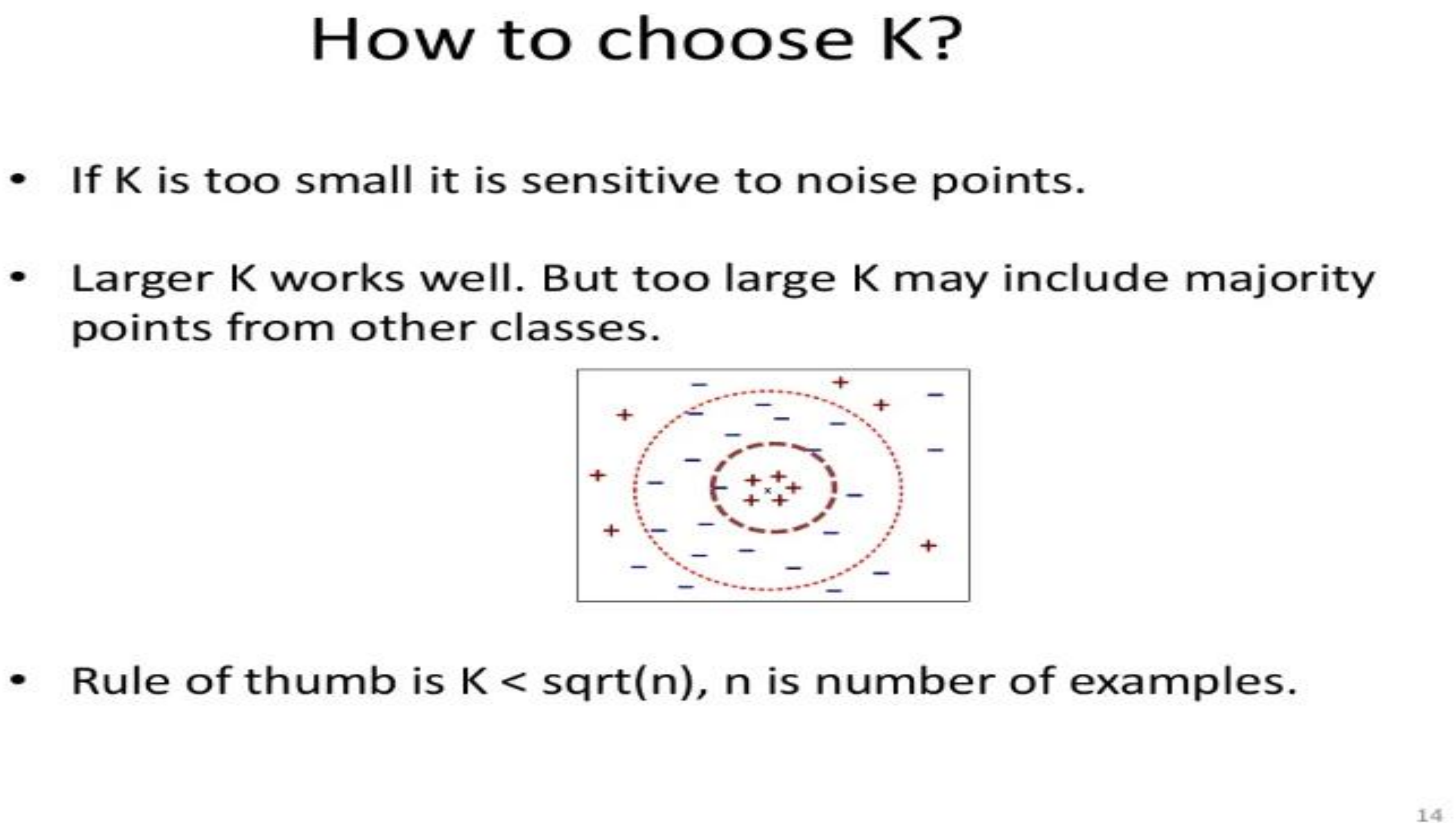




How does K-NN work?

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

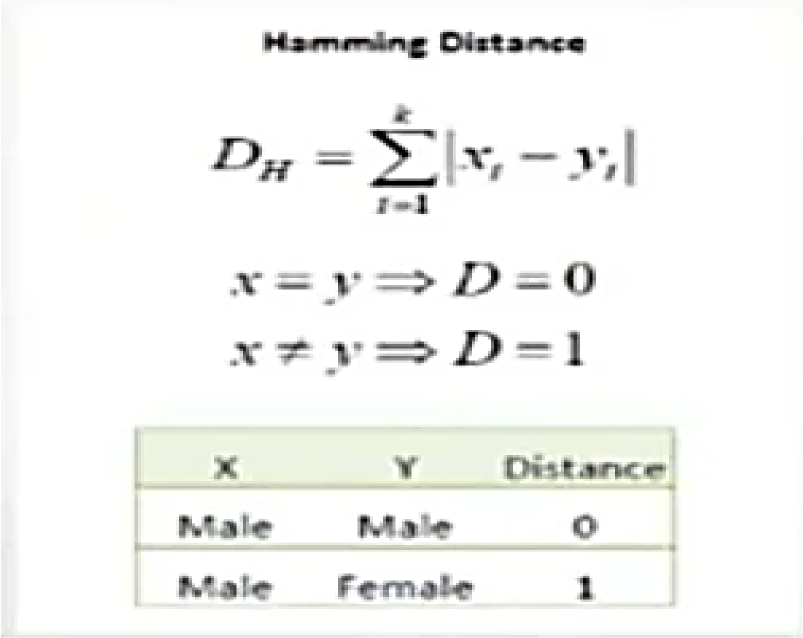


How to choose K?

- If K is too small it is sensitive to noise points.
- Larger K works well. But too large K may include majority points from other classes.
- Rule of thumb is K < sqrt(n), n is number of examples.

Nominal/Categorical Data

- Distance works naturally with numerical attributes.
- Binary value categorical data attributes can be regarded as 1 or 0.



Hamming Distance

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

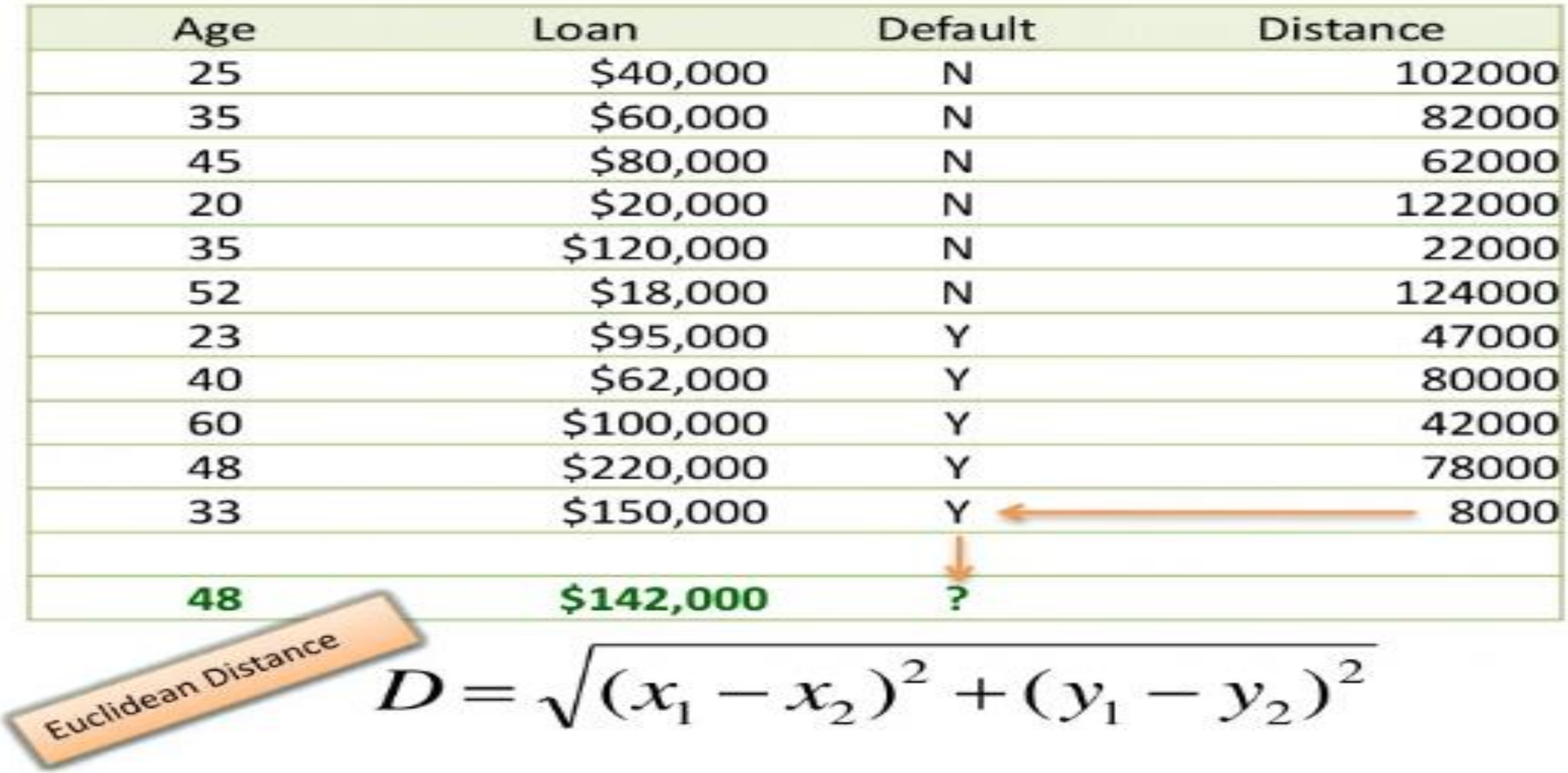
$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

19

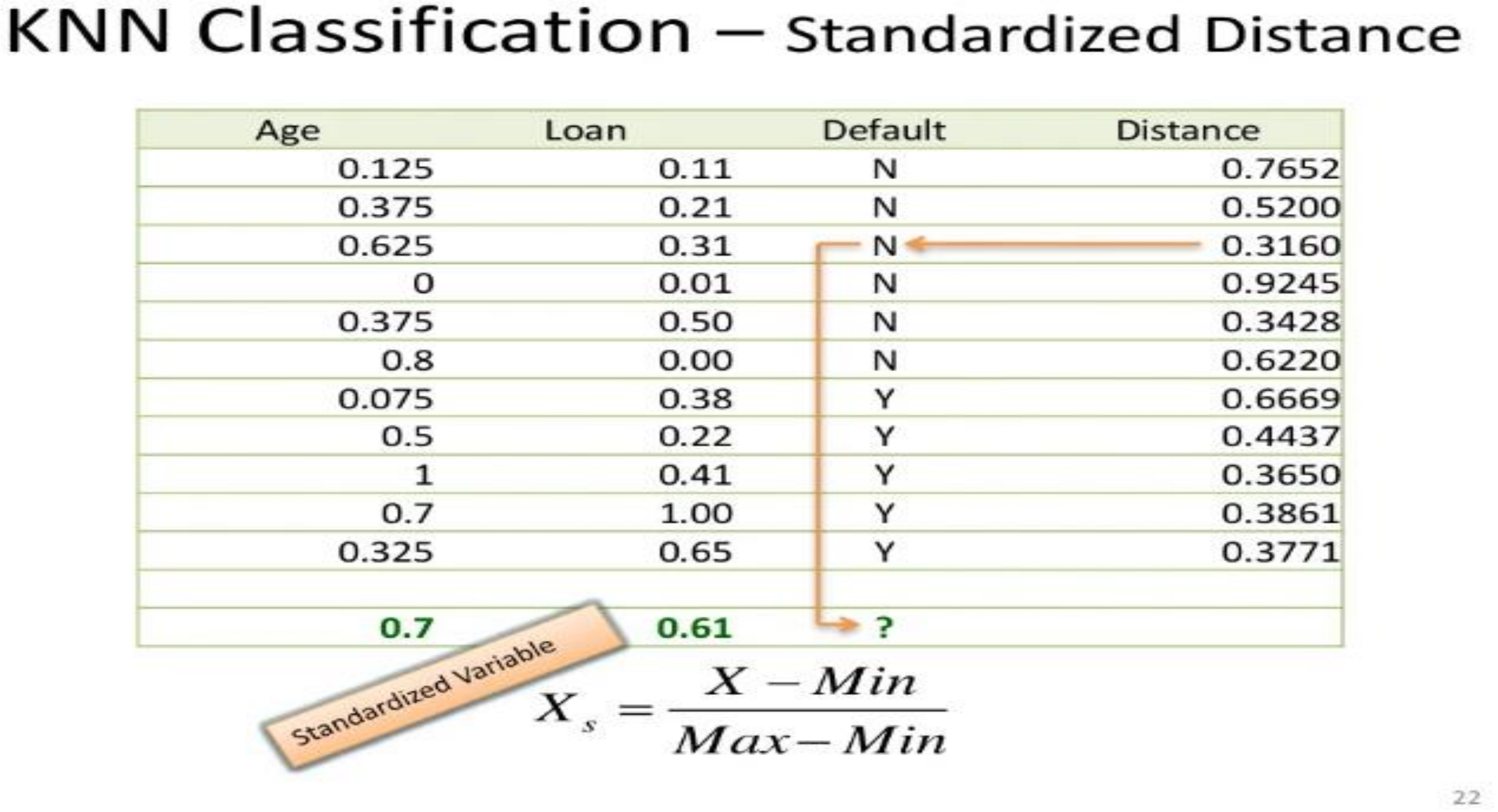
KNN Classification — Distance



Age	Loan	Default	Distance
25	$40,000	N	102000
35	$60,000	N	82000
45	$80,000	N	62000
20	$20,000	N	122000
35	$120,000	N	22000
52	$18,000	N	124000
23	$95,000	Y	47000
40	$62,000	Y	80000
60	$100,000	Y	42000
48	$220,000	Y	78000
33	$150,000	Y	8000
48	$142,000	?	

Euclidean Distance

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



KNN Classification – Standardized Distance

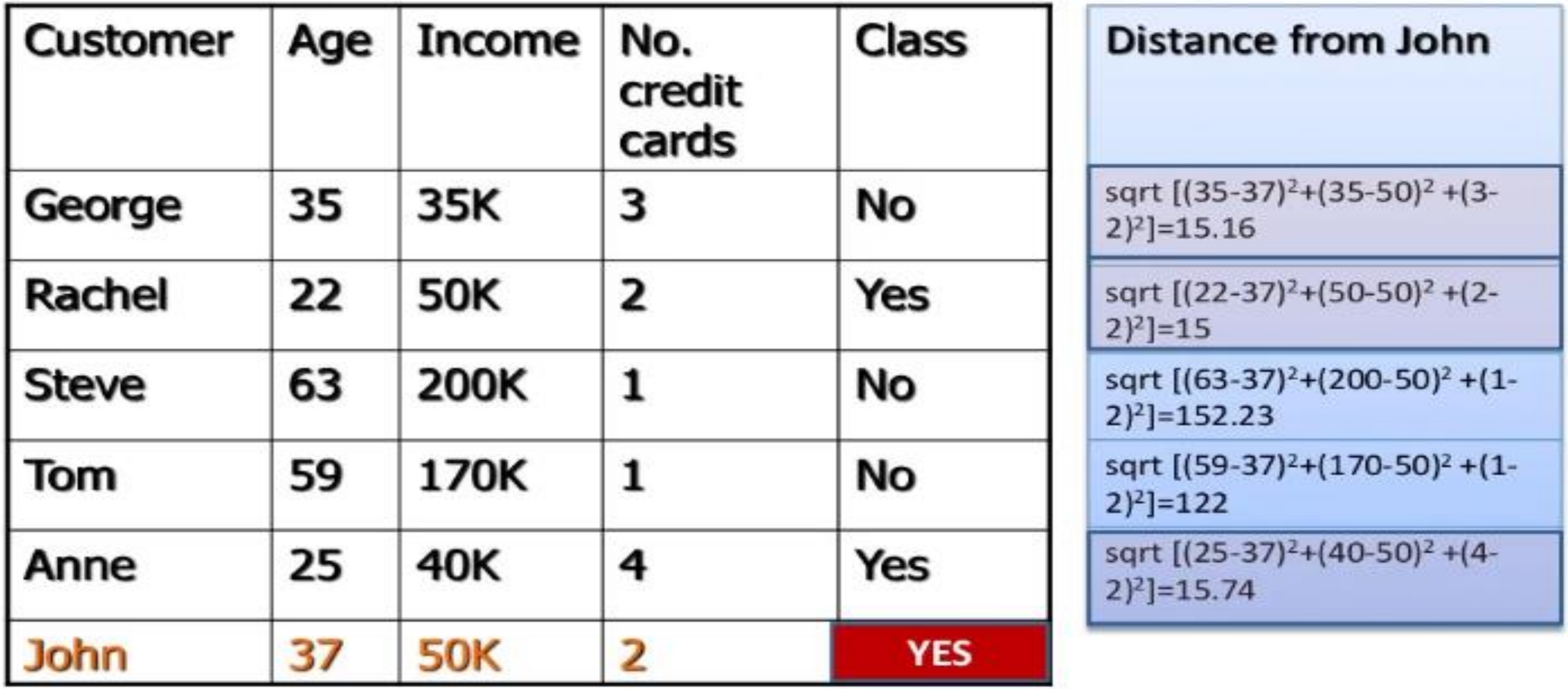
Age	Loan	Default	Distance
0.125	0.11	N	0.7652
0.375	0.21	N	0.5200
0.625	0.31	N	0.3160
0	0.01	N	0.9245
0.375	0.50	N	0.3428
0.8	0.00	N	0.6220
0.075	0.38	Y	0.6669
0.5	0.22	Y	0.4437
1	0.41	Y	0.3650
0.7	1.00	Y	0.3861
0.325	0.65	Y	0.3771
0.7	0.61	?	

Standardized Variable

$$X_s = \frac{X - Min}{Max - Min}$$

22

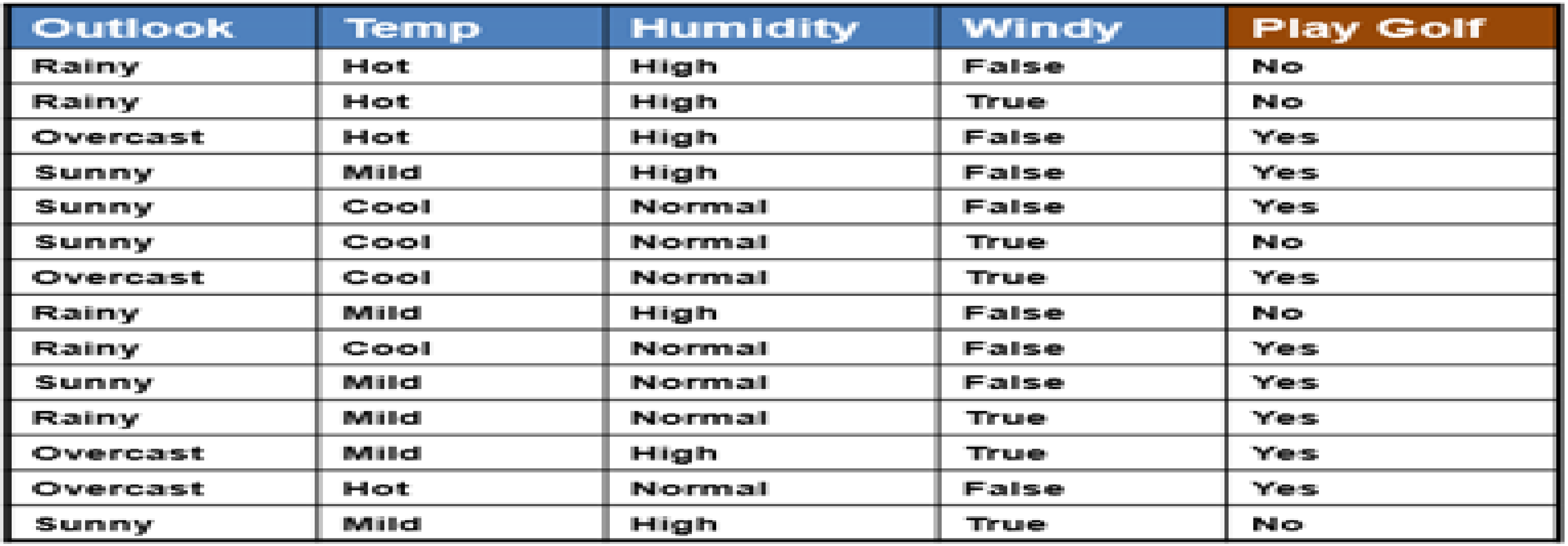
3-KNN: Example(1)



Customer	Age	Income	No. credit cards	Class
George	35	35K	3	No
Rachel	22	50K	2	Yes
Steve	63	200K	1	No
Tom	59	170K	1	No
Anne	25	40K	4	Yes
John	37	50K	2	YES

Distance from John
sqrt $[(35-37)^2+(35-50)^2+(3-2)^2]=15.16$
sqrt $[(22-37)^2+(50-50)^2+(2-2)^2]=15$
sqrt $[(63-37)^2+(200-50)^2+(1-2)^2]=152.23$
sqrt $[(59-37)^2+(170-50)^2+(1-2)^2]=122$
sqrt $[(25-37)^2+(40-50)^2+(4-2)^2]=15.74$

Problem Statement



Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Identify the Play Golf possibility if Outlook=‘Overcast’, Temp=‘Cool’, Wicndy=‘True’.