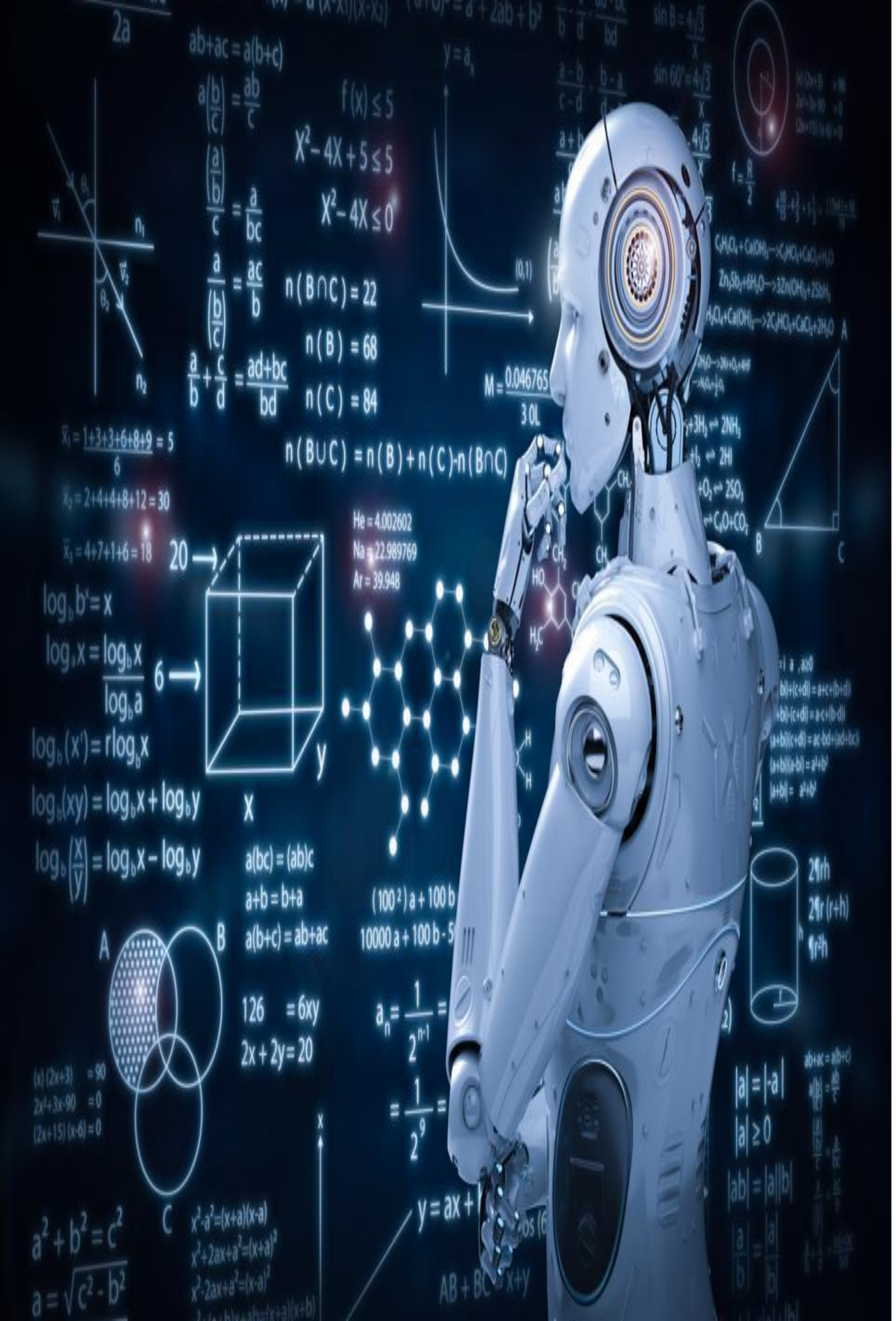


Practical Machine Learning

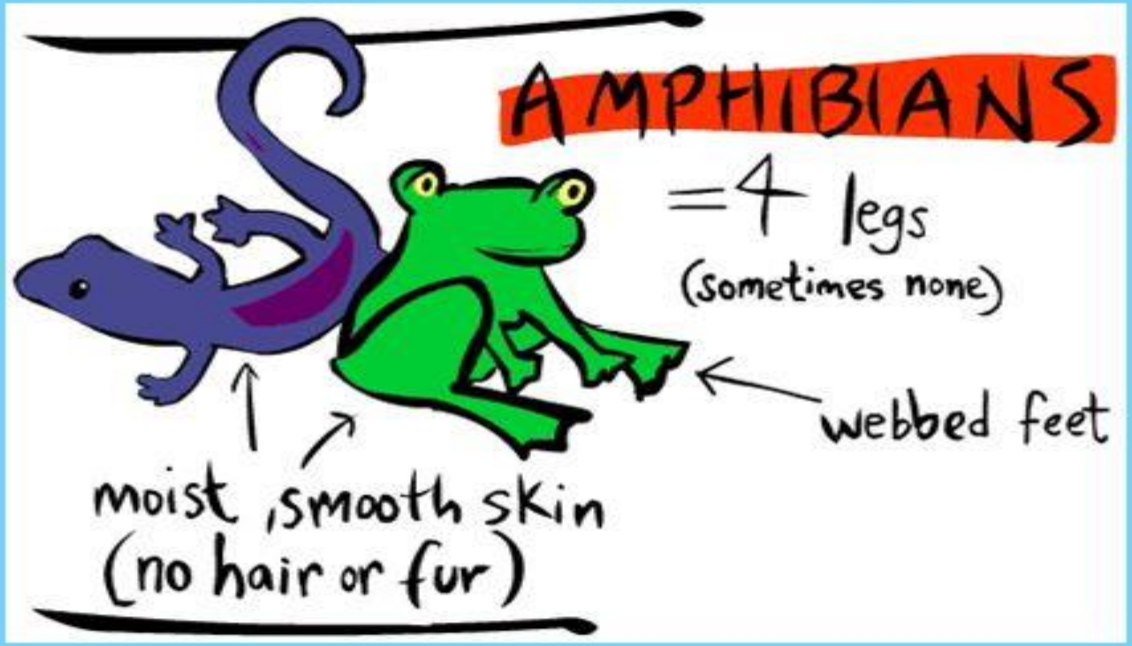
Day 9: Sep22 DBDA

Kiran Waghmare

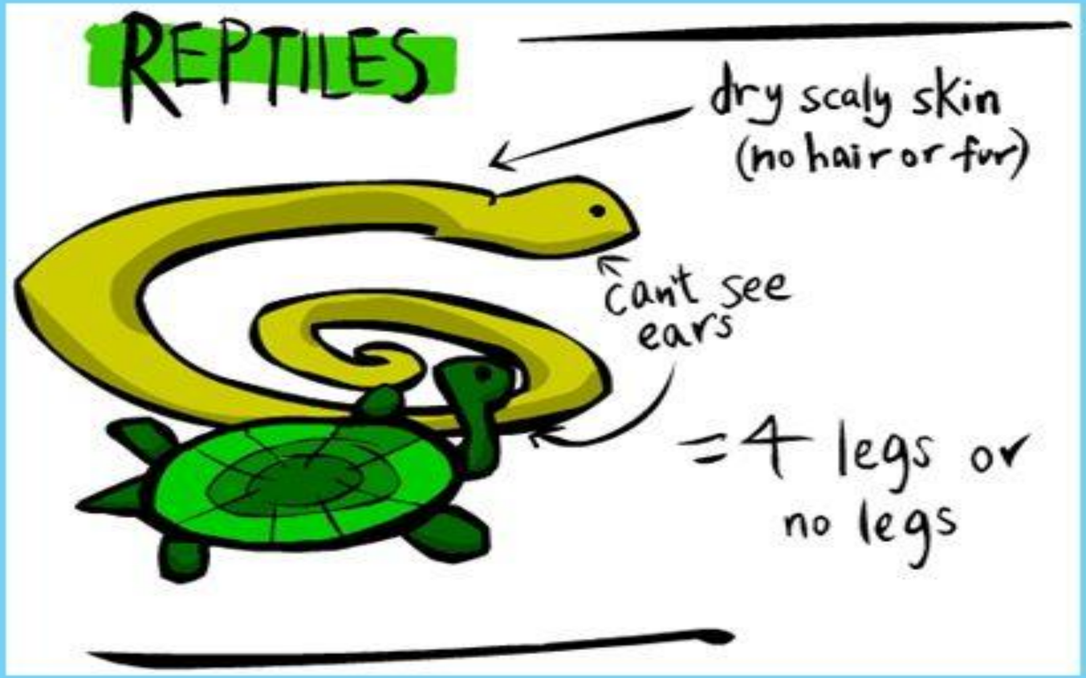
Agenda

- Classification
- Measures for classification
- KNN

Amphibians



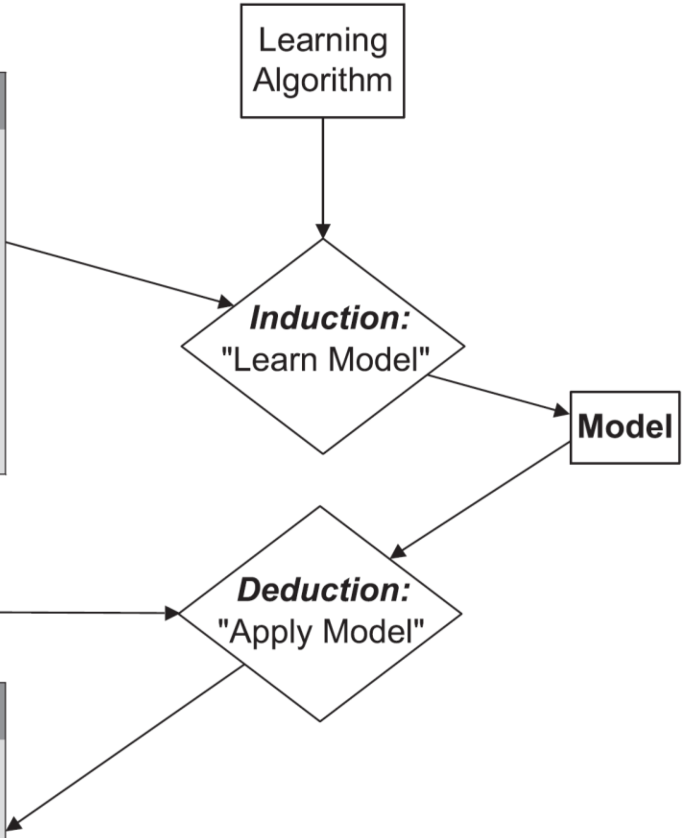
Reptiles



General Approach for Building Classification Model

Training Set

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
11	No	Married	55K	?
12	Yes	Divorced	80K	?
13	Yes	Single	110K	?
14	No	Single	95K	?
15	No	Married	67K	?

Test Set

Figure 3.3. General framework for building a classification model.

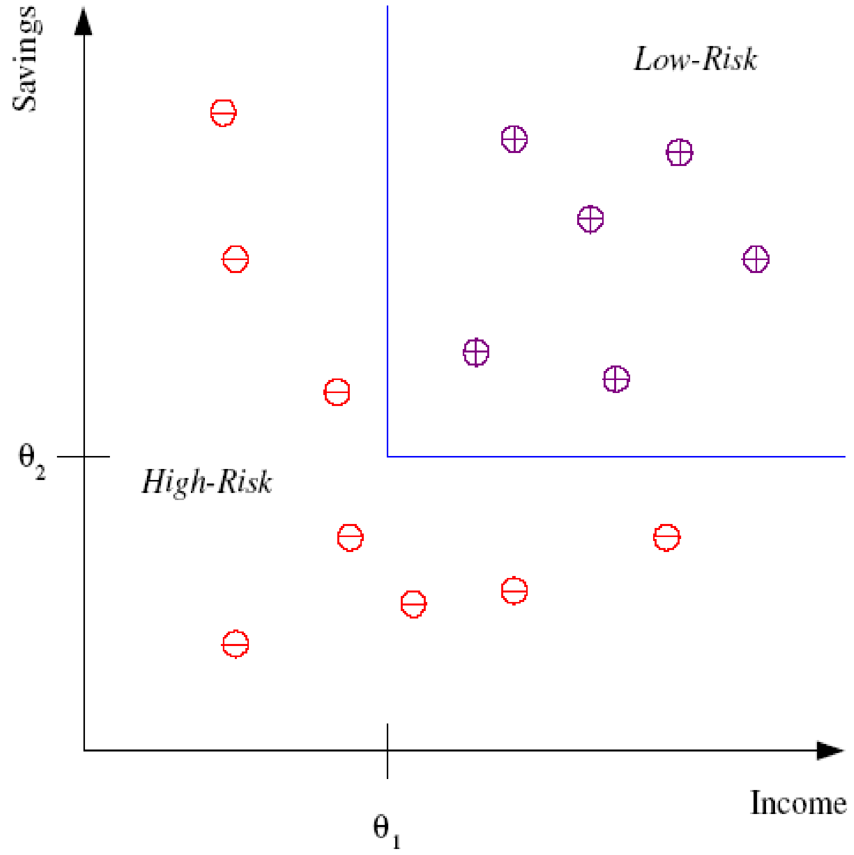
Classification Techniques

- Base Classifiers
  - Decision Tree based Methods
  - Rule-based Methods
  - Nearest-neighbor
  - Naïve Bayes and Bayesian Belief Networks
  - Support Vector Machines
  - Neural Networks, Deep Neural Nets

- Ensemble Classifiers
  - Boosting, Bagging, Random Forests

Classification

- Example: Credit scoring
- Differentiating between low-risk and high-risk customers from their *income* and *savings*



Discriminant: IF *income* $> \theta_1$ AND *savings* $> \theta_2$
THEN low-risk ELSE high-risk

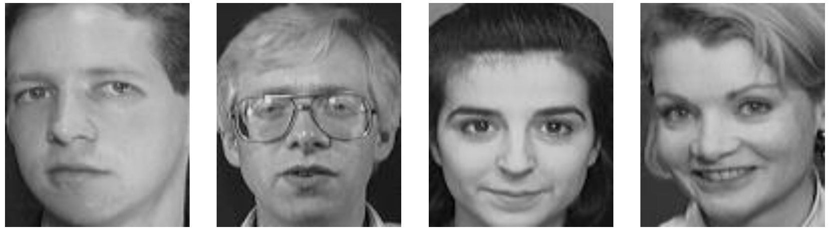
Classification: Applications

- Aka Pattern recognition
- Face recognition: Pose, lighting, occlusion (glasses, beard), make-up, hair style
- Character recognition: Different handwriting styles.
- Speech recognition: Temporal dependency.
  - Use of a dictionary or the syntax of the language.
  - Sensor fusion: Combine multiple modalities; eg, visual (lip image) and acoustic for speech
- Medical diagnosis: From symptoms to illnesses
- ...

Face Recognition

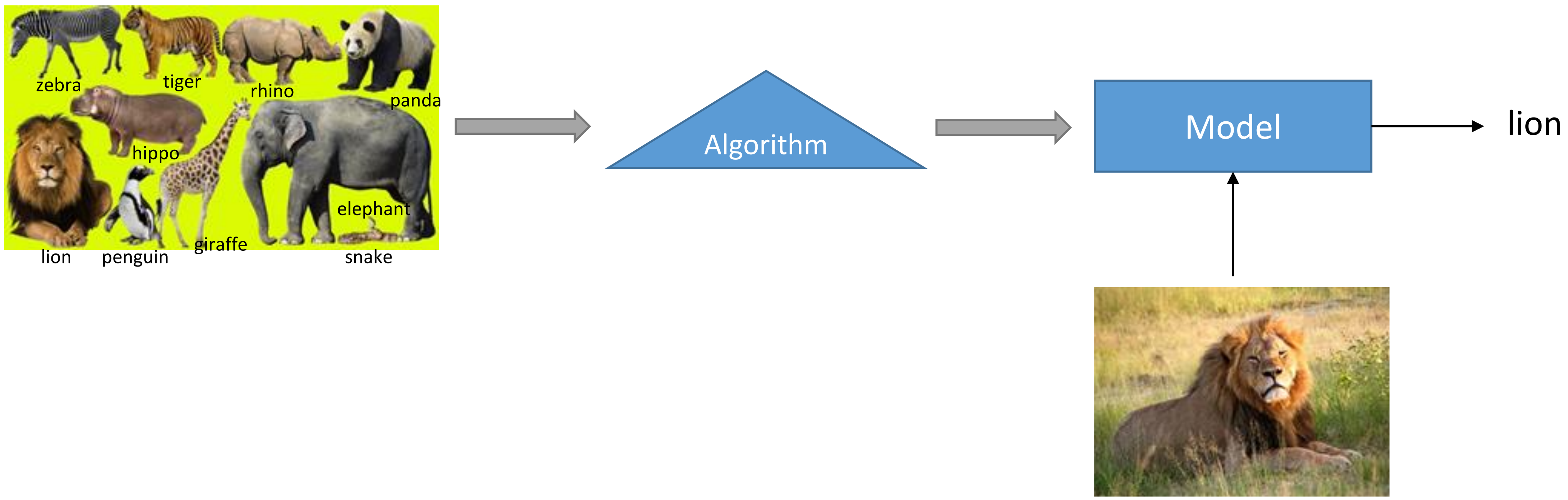
Training examples of a person

Test images

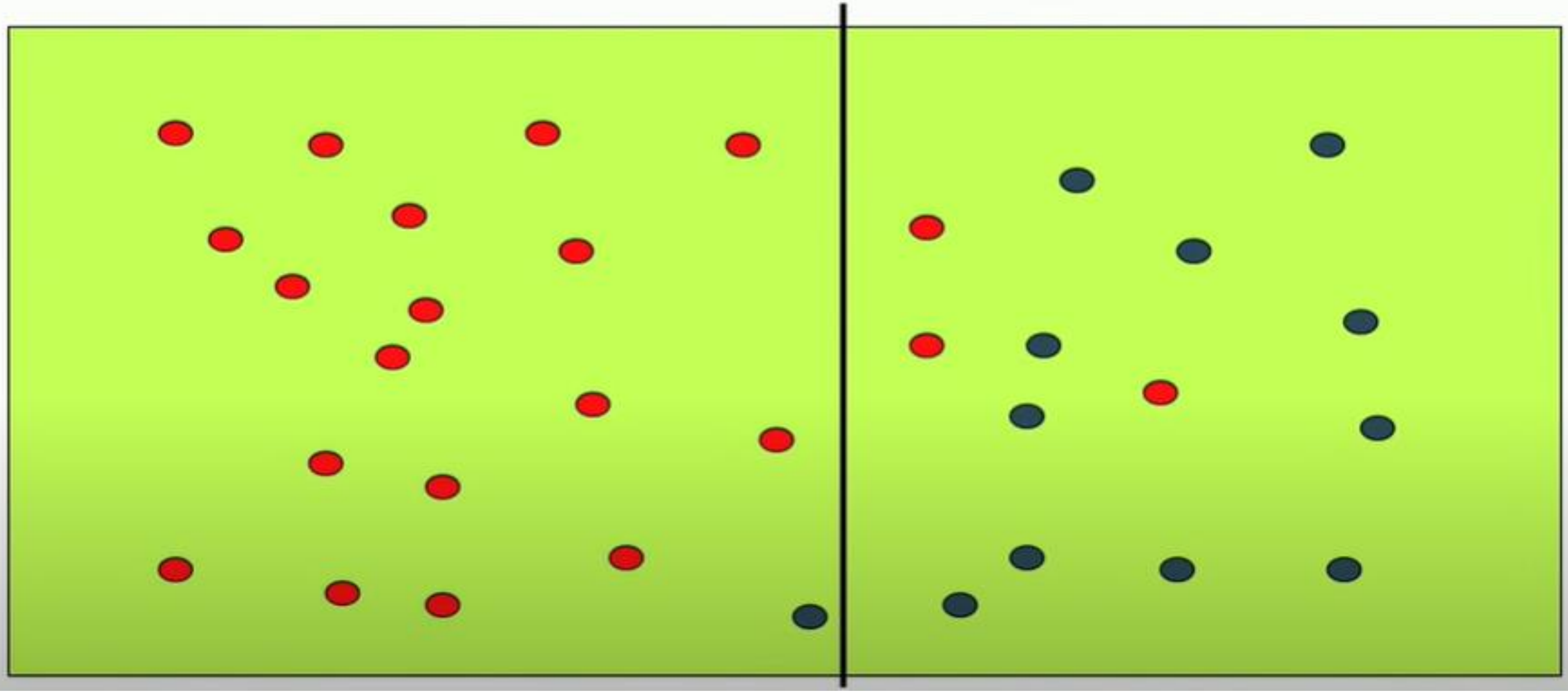


AT&T Laboratories, Cambridge UK
http://www.uk.research.att.com/facedatabase.html

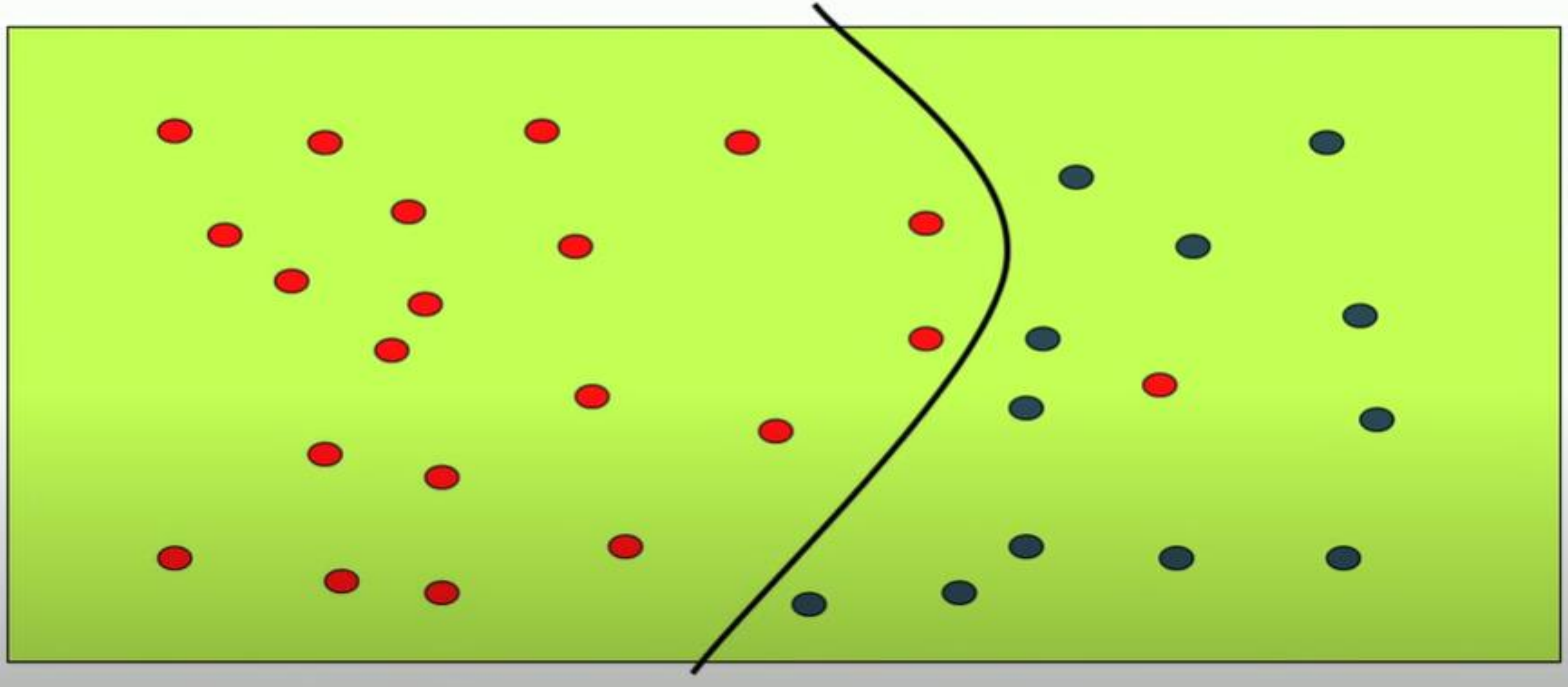
Classification



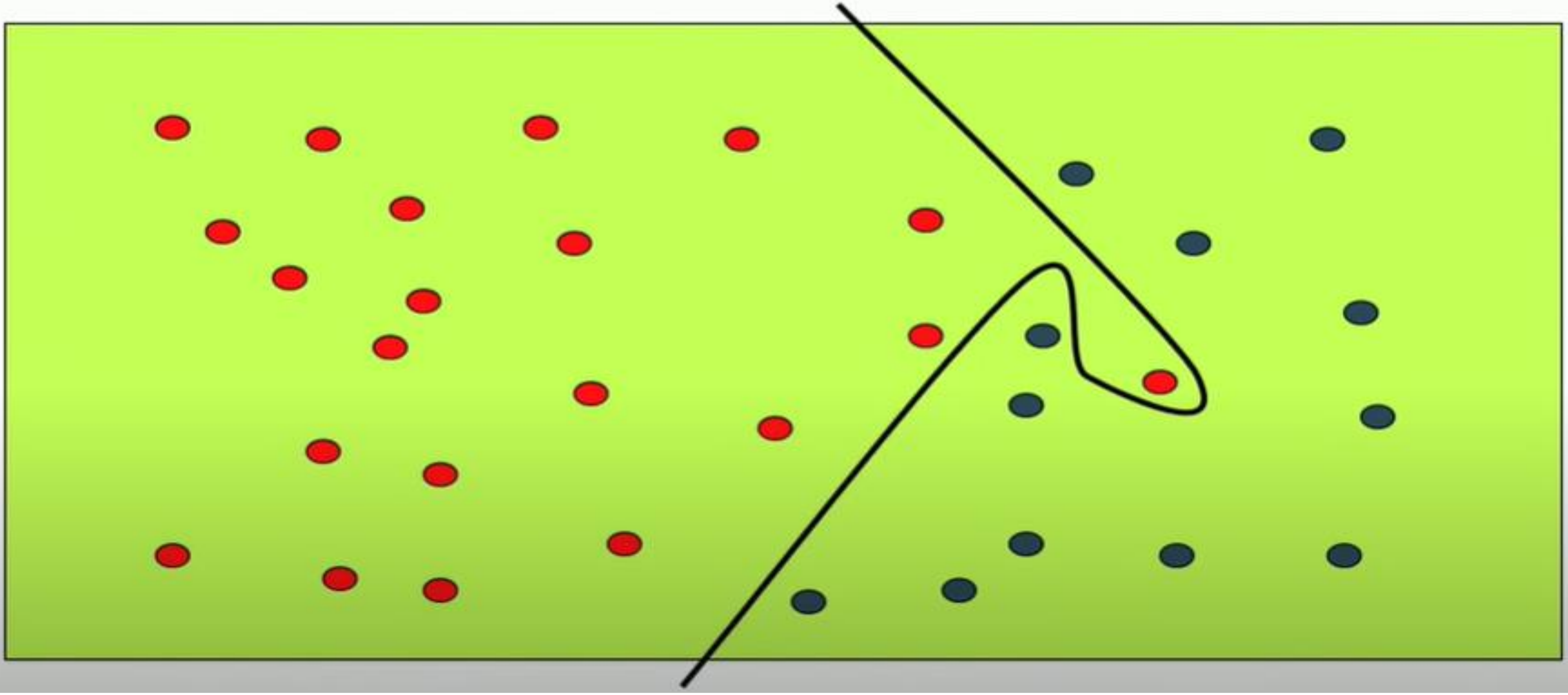
Possible Classifiers



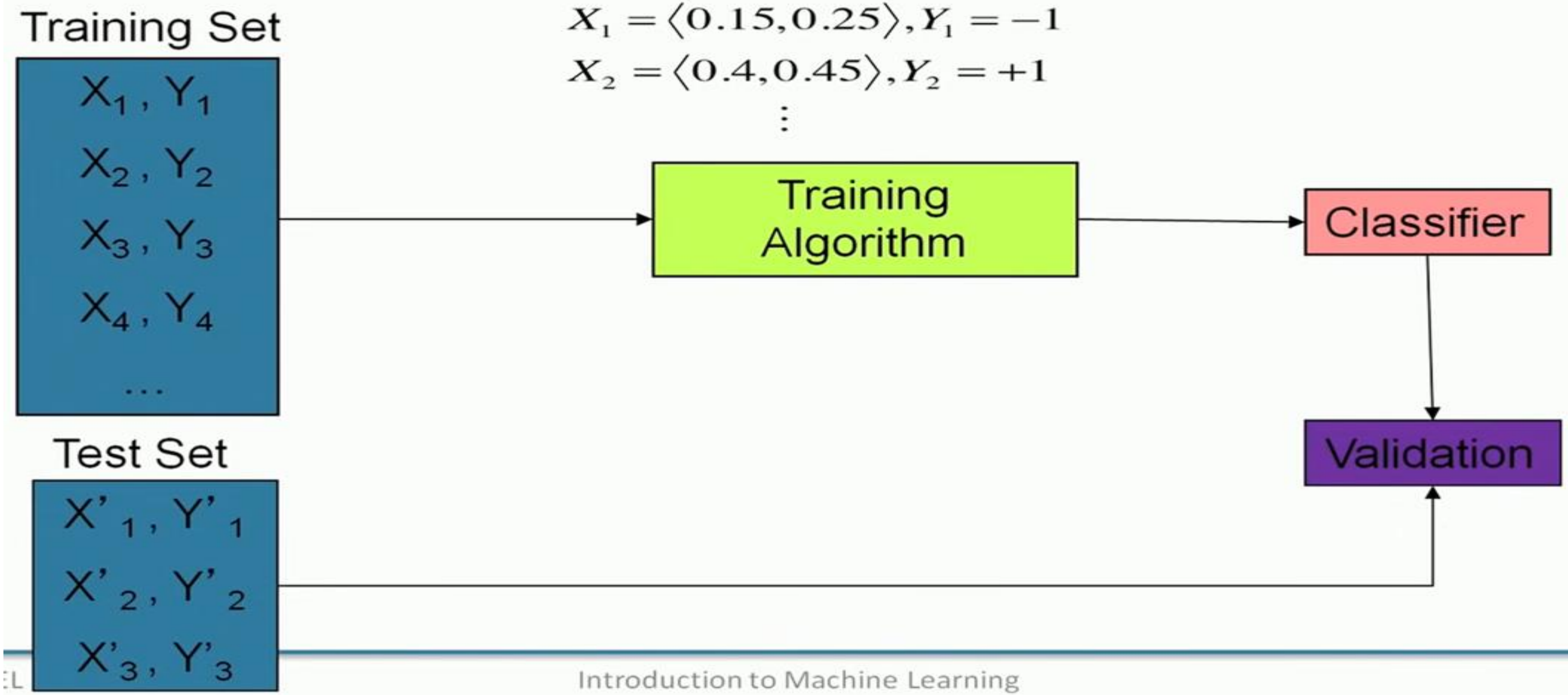
Possible Classifiers



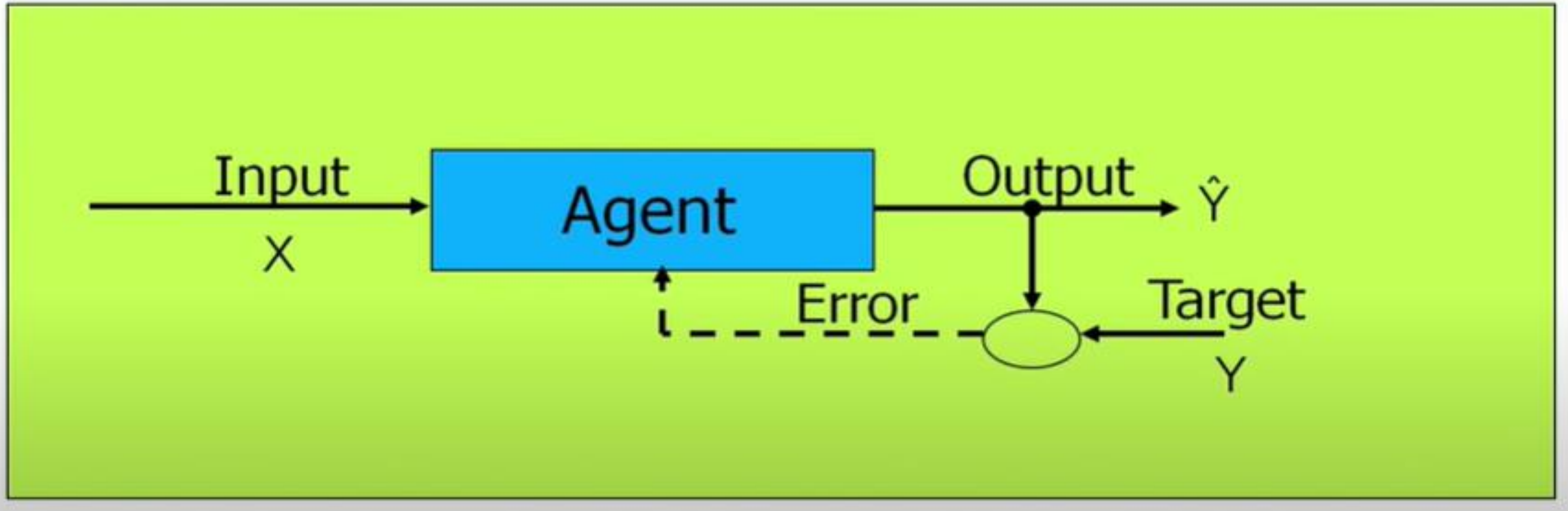
Possible Classifiers



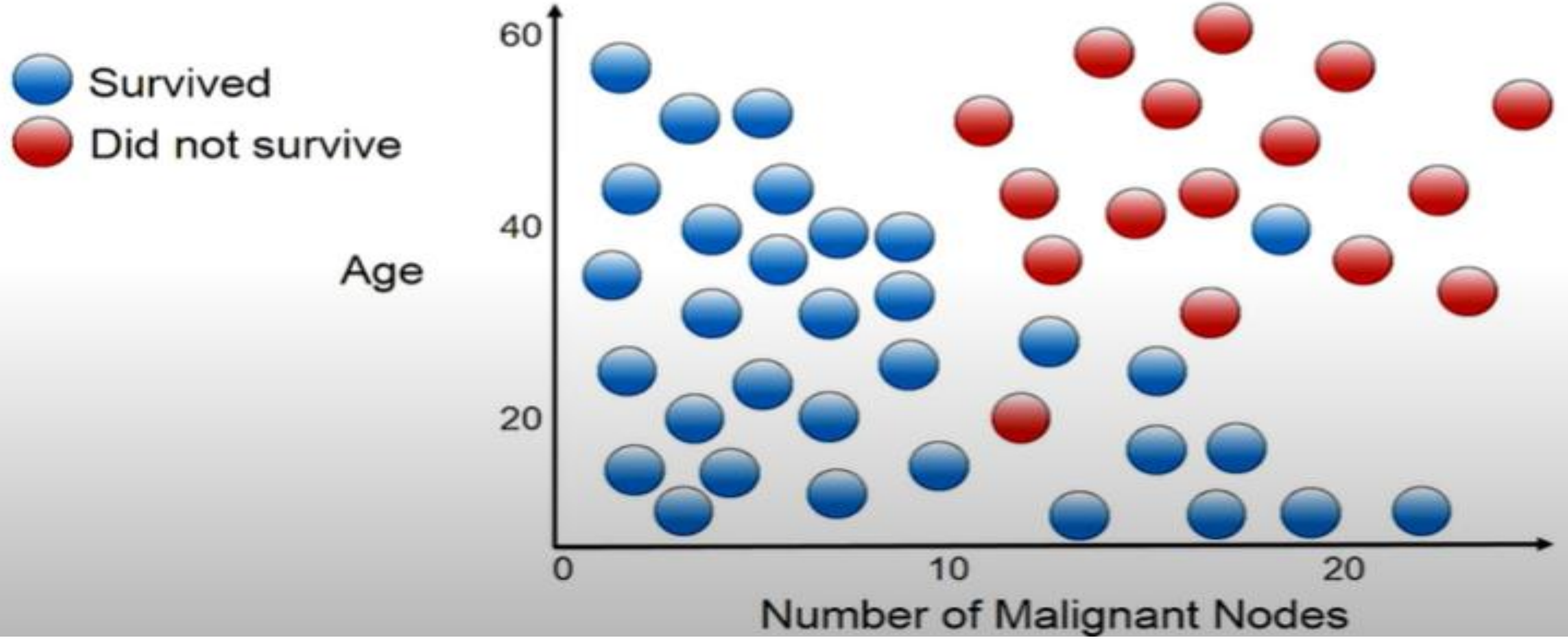
The Process


Training Set
X_1 , Y_1
X_2 , Y_2
X_3 , Y_3
X_4 , Y_4
...
$X_1 = \langle 0.15, 0.25 \rangle, Y_1 = -1$
$X_2 = \langle 0.4, 0.45 \rangle, Y_2 = +1$
⋮
Training Algorithm
Classifier
Validation
Test Set
X'_1 , Y'_1
X'_2 , Y'_2
X'_3 , Y'_3

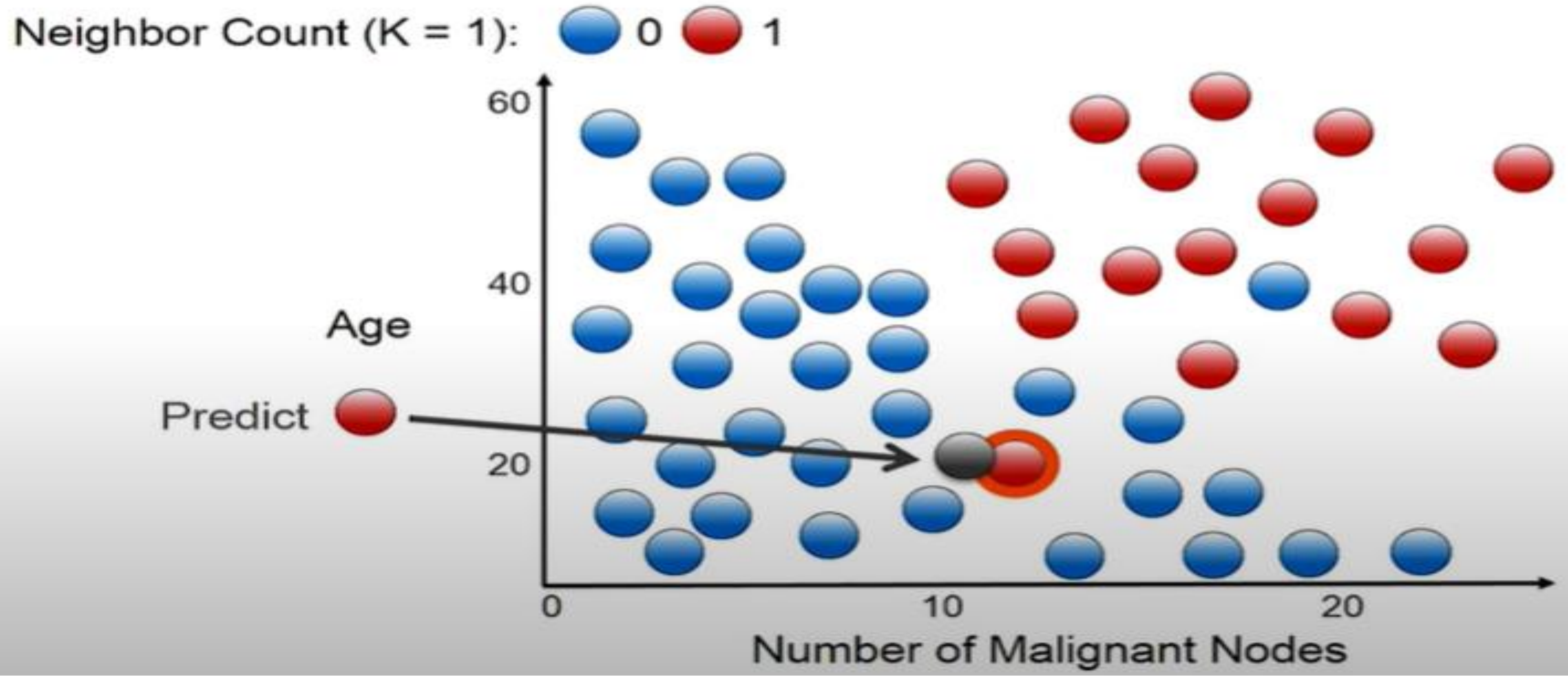
Training



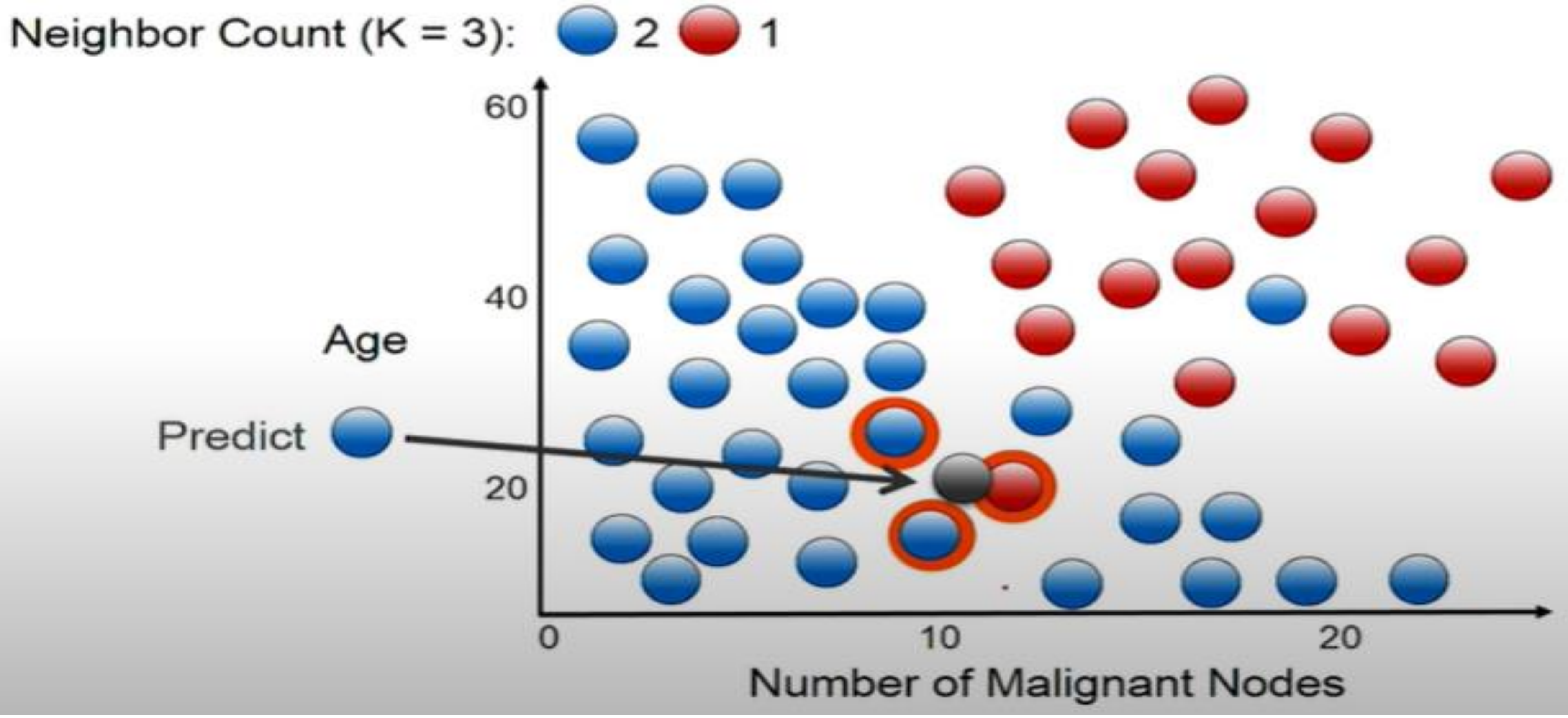
K-Nearest Neighbour- Classification



.. Classification



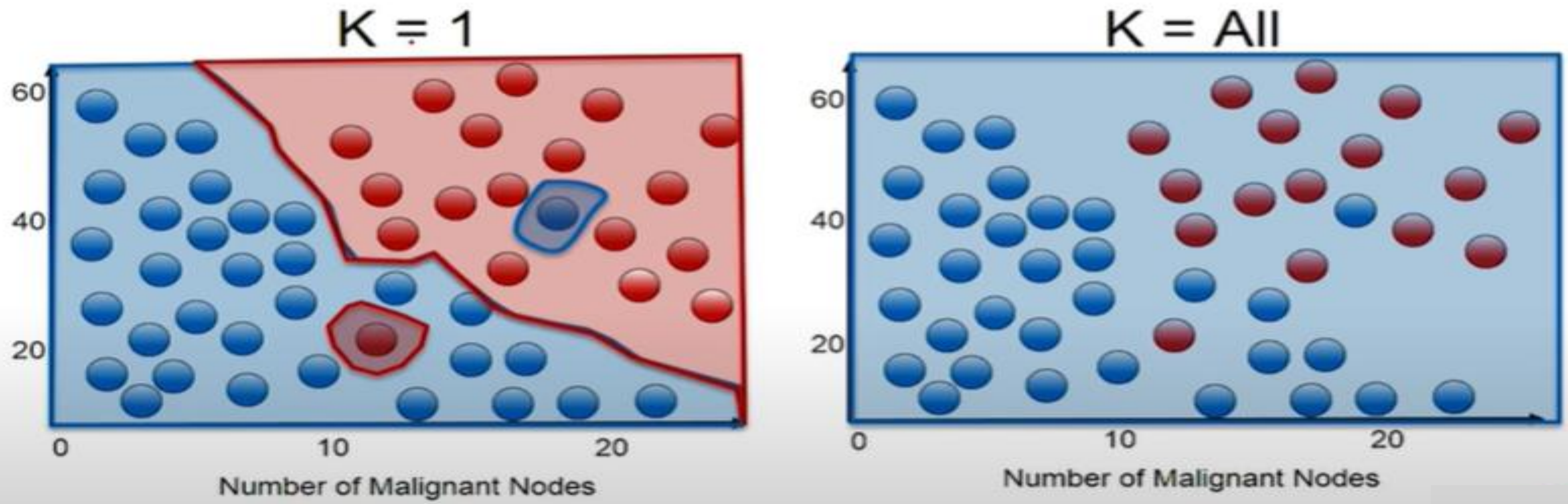
...Classification



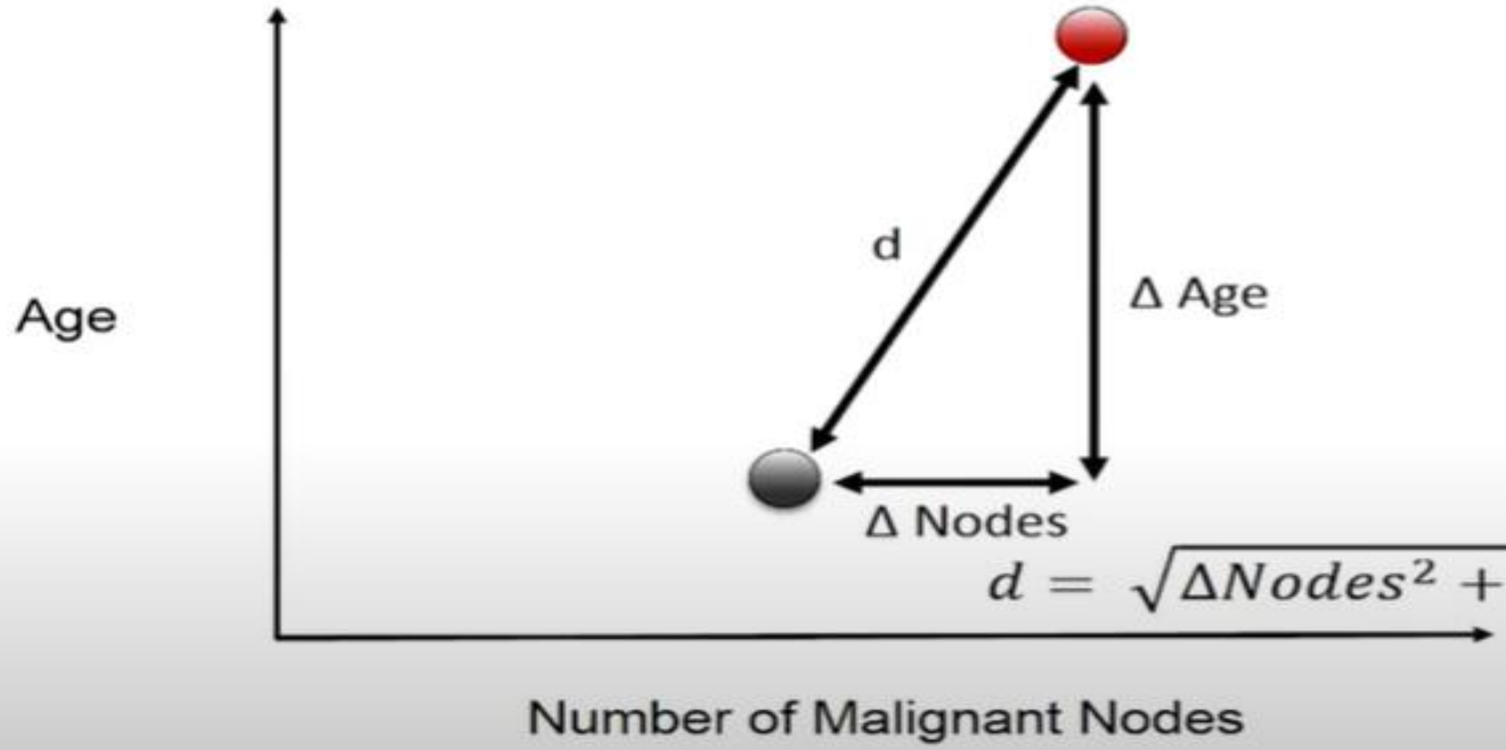
KNN parameters

- K – nearest neighbours
- Distance metric

Choosing K


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

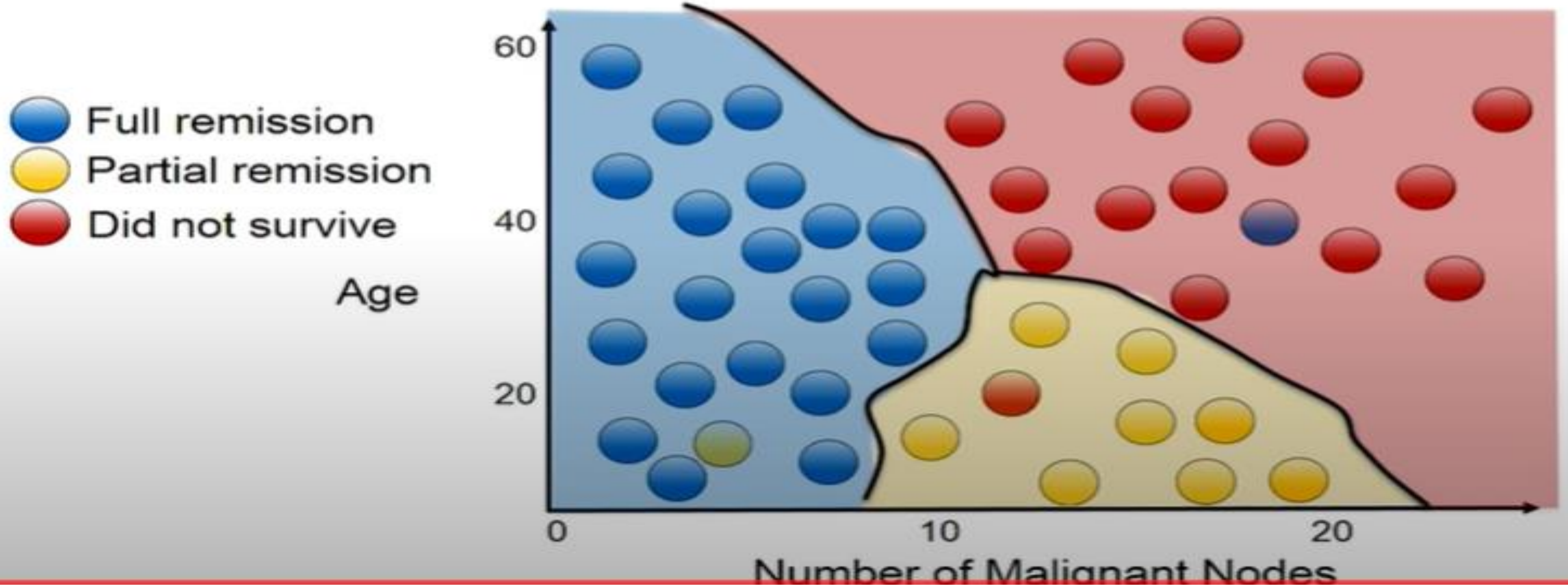
Distance Metric- Euclidean Distance



$$d = \sqrt{\Delta Nodes^2 + \Delta Age^2}$$

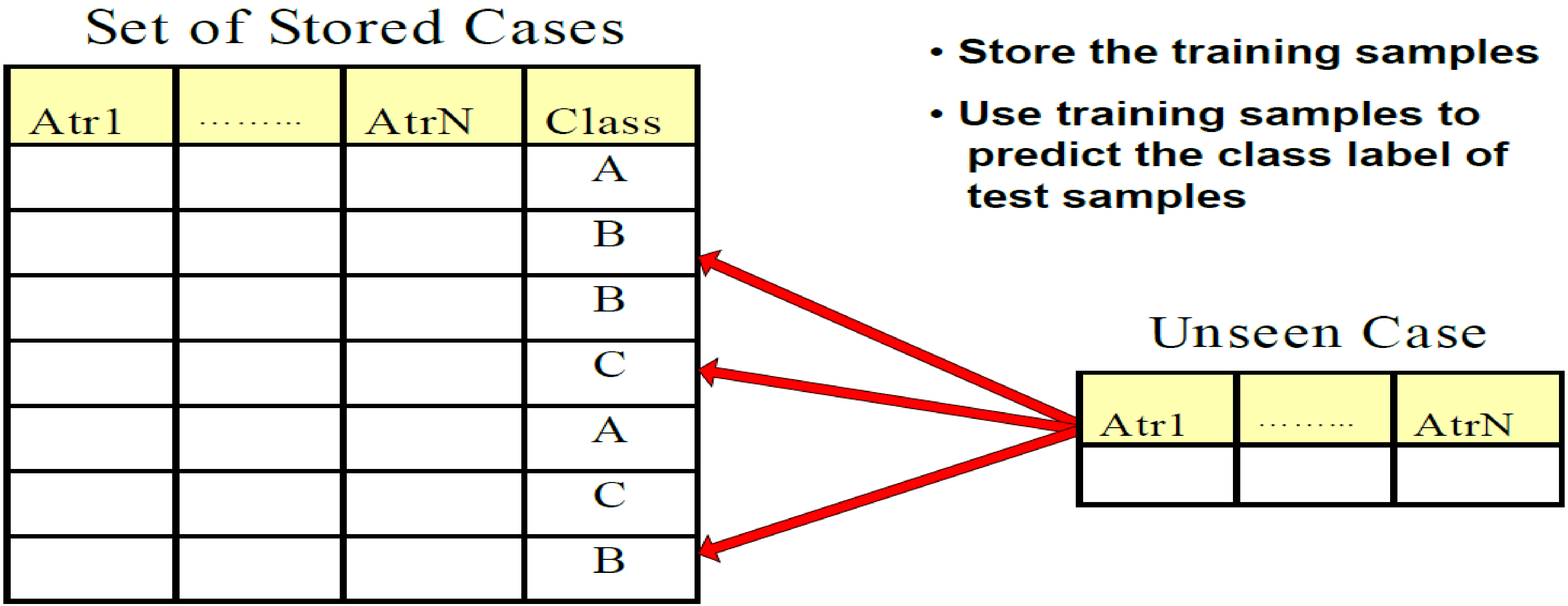
Multiple Classes

K = 5



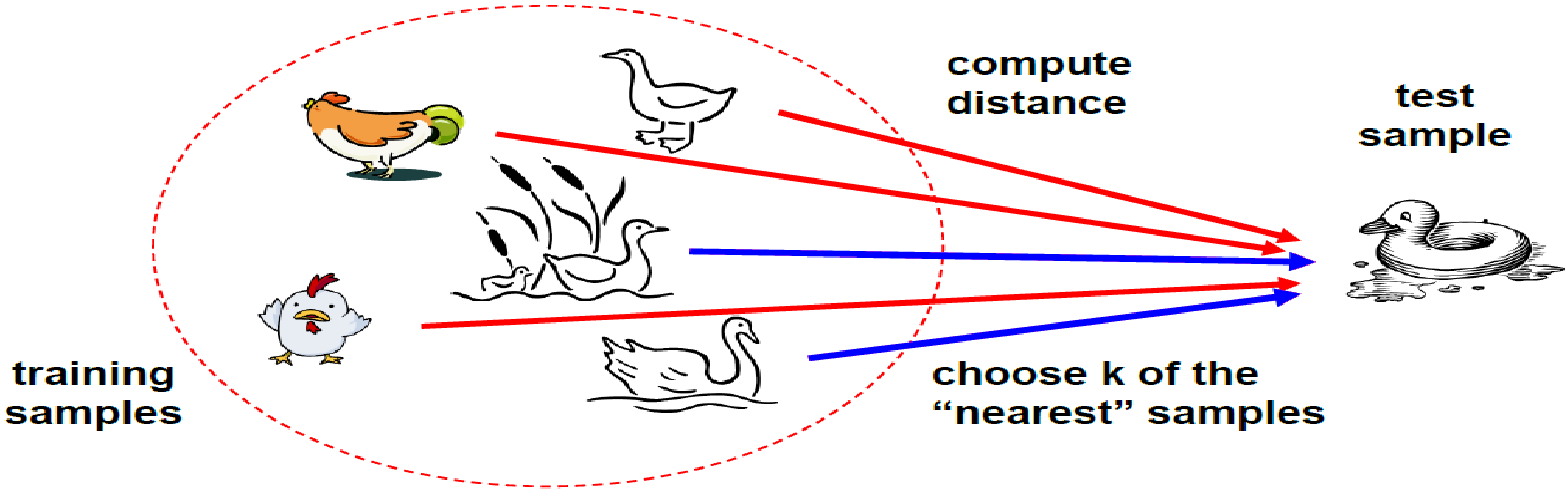
Instance based classifiers

- Store the training samples
- Use training samples to predict the class label of test samples



Nearest Neighbor Classifiers

- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



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).

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.

Simple Analogy..

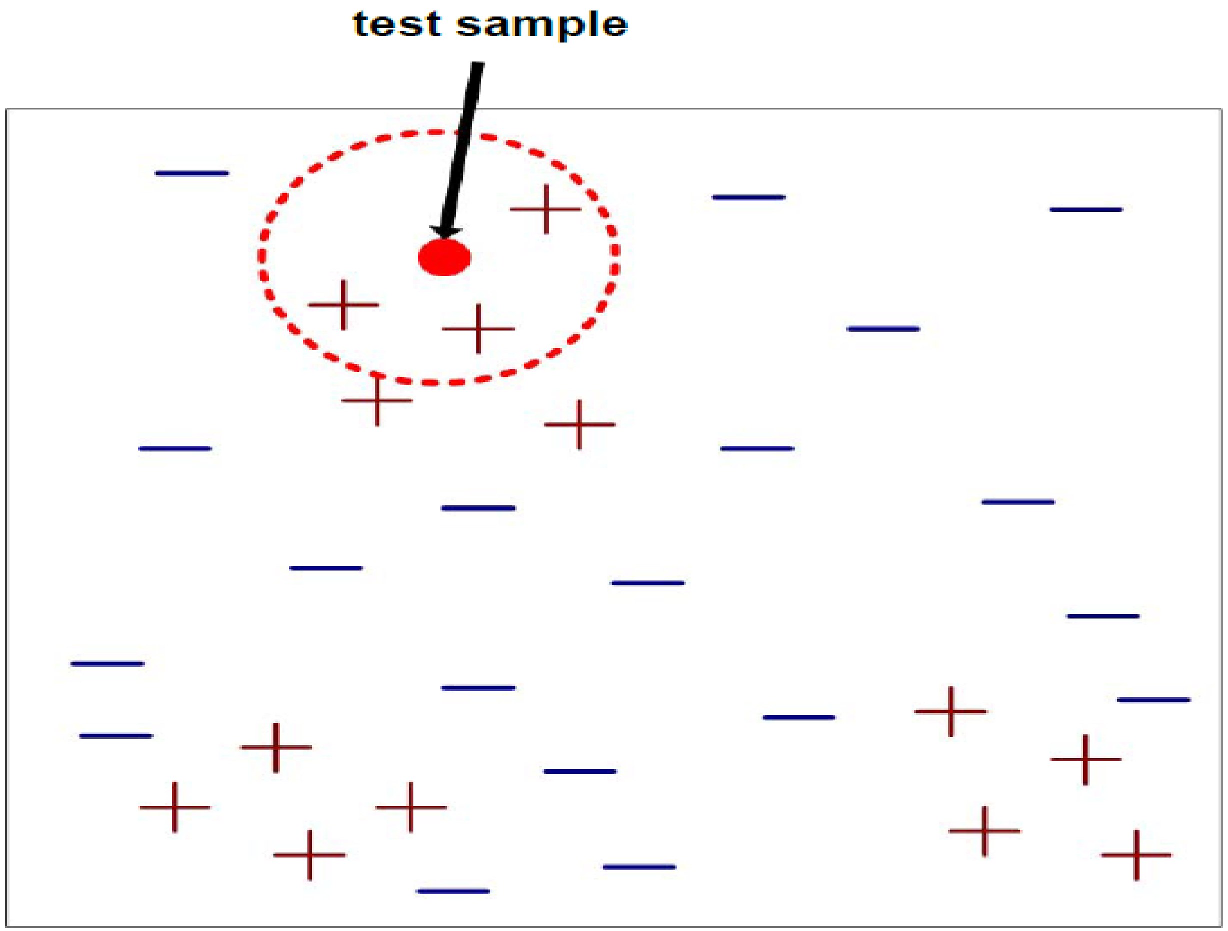
- Tell me about your friends(*who your neighbors are*) and *I will tell you who you are*.






3

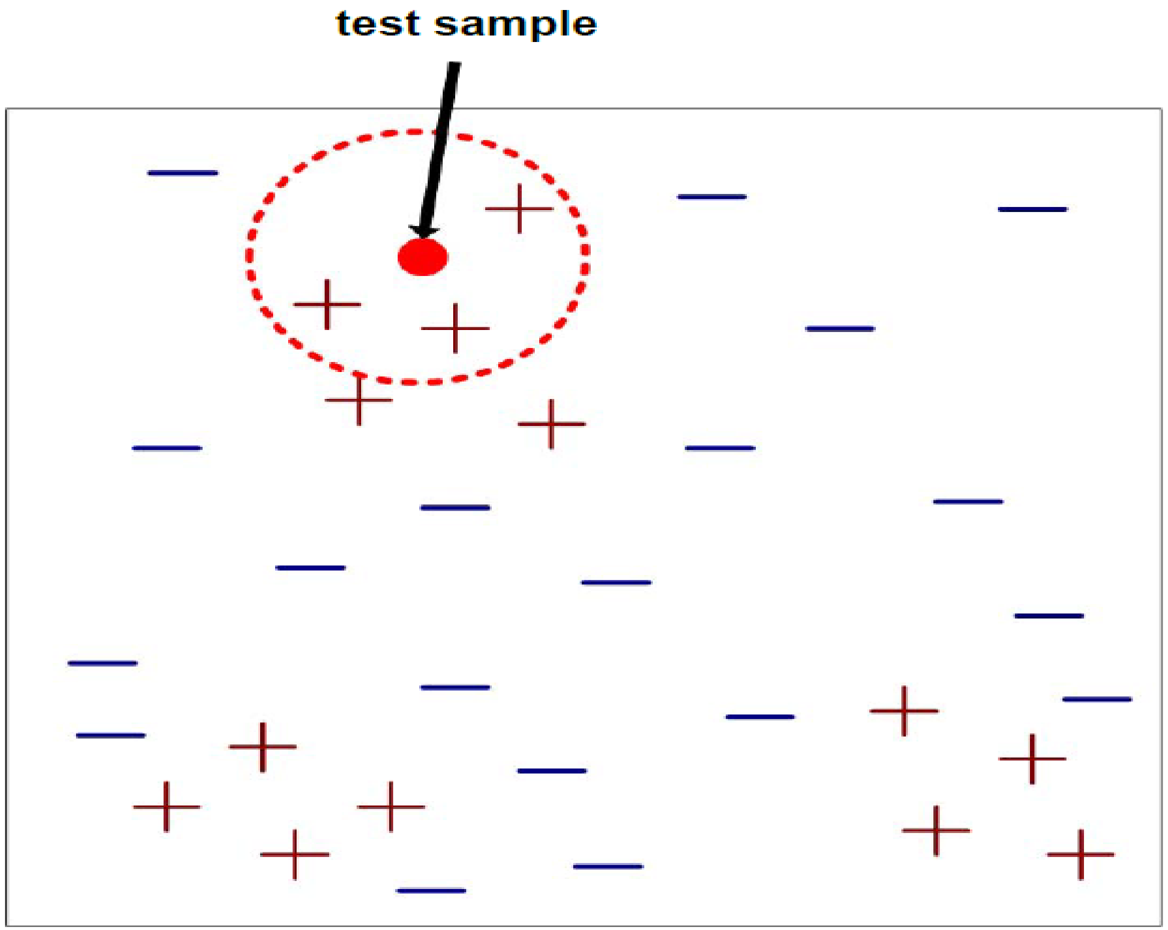
Nearest Neighbor Classifiers



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

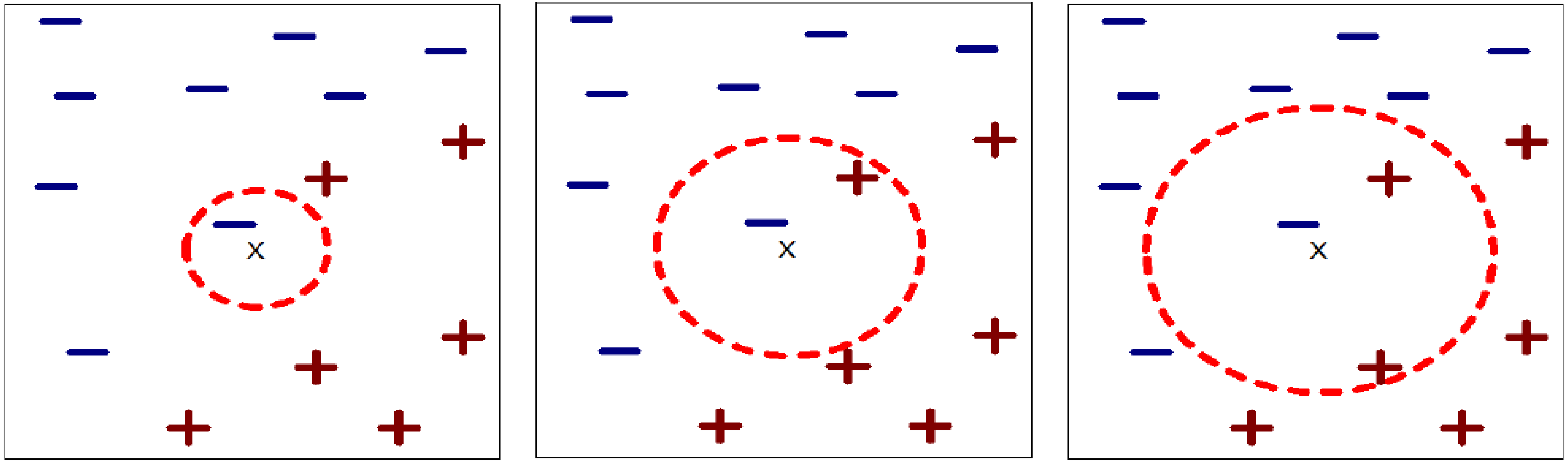


To classify test sample:

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

Distance measure for Continuous Variables

Distance functions

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

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

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

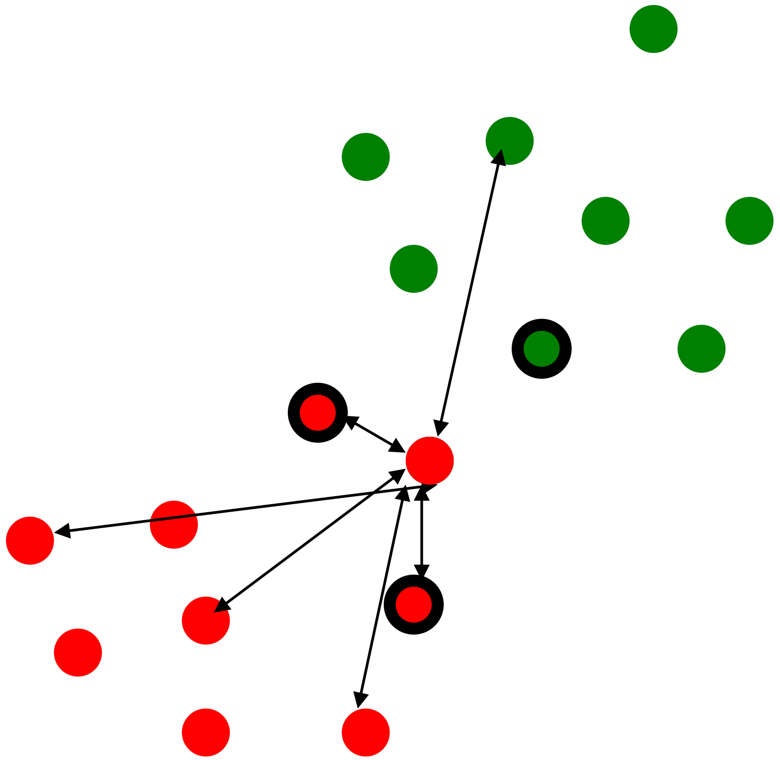
K-NN classifier schematic

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

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

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



Distance Between Neighbors

- Calculate the distance between new example (E) and all examples in the training set.

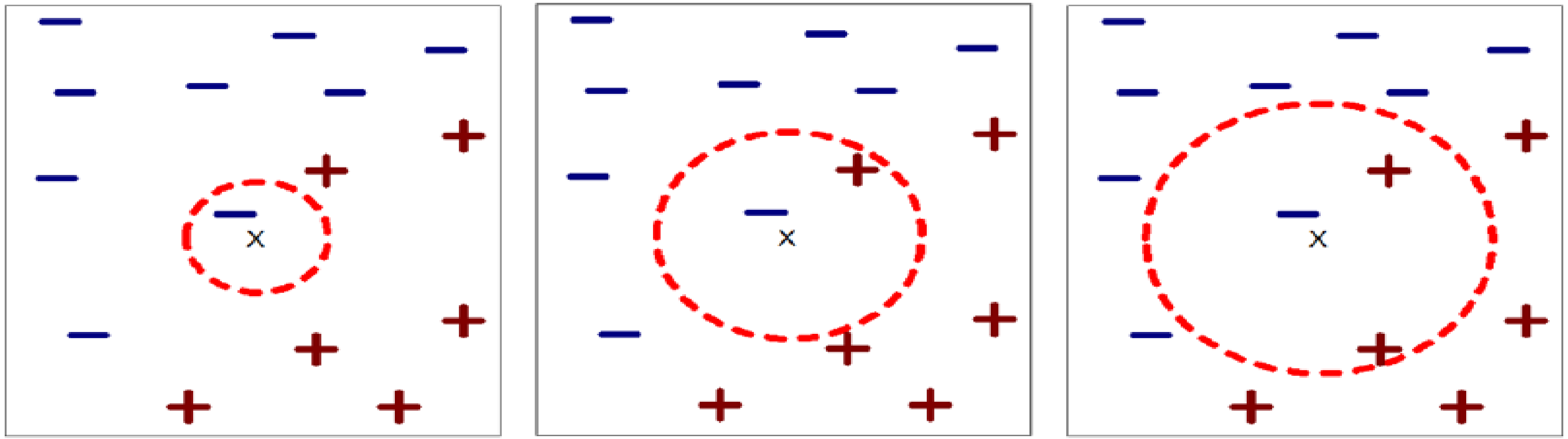
- *Euclidean* distance between two examples.
  - $X = [x_1, x_2, x_3, .., x_n]$
  - $Y = [y_1, y_2, y_3, ..., y_n]$

  - The Euclidean distance between *X* and *Y* is defined as:

$$D(X,Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

1

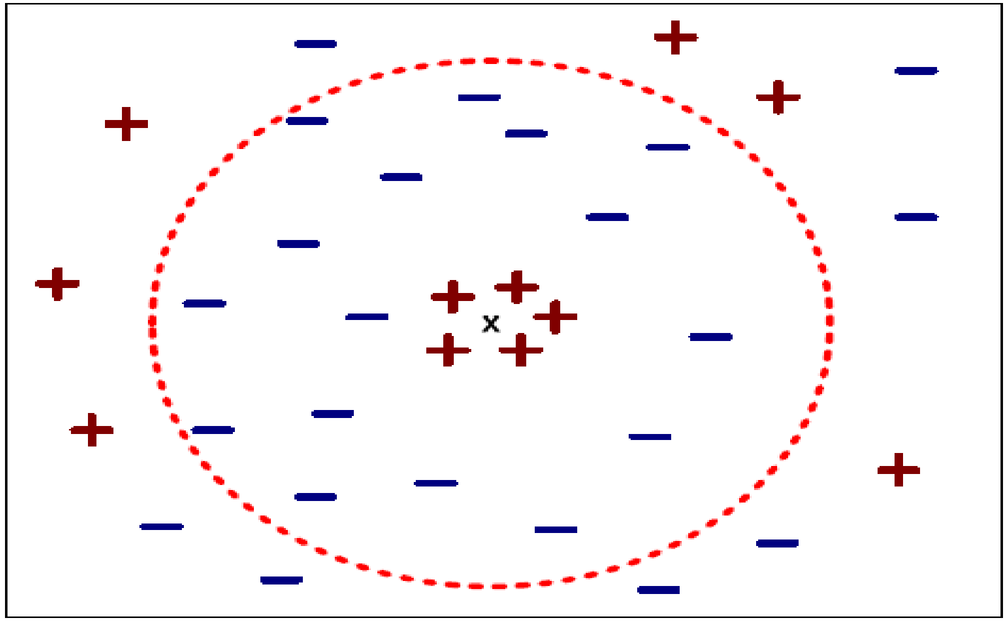
Predicting class from nearest neighbors



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

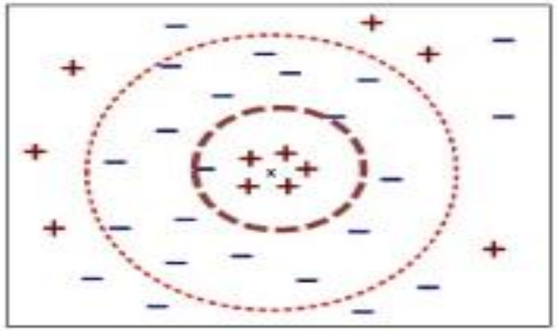


K-Nearest Neighbor Algorithm

- All the instances correspond to points in an n-dimensional feature space.
- Each instance is represented with a set of numerical attributes.
- Each of the training data consists of a set of vectors and a class label associated with each vector.
- Classification is done by comparing feature vectors of different K nearest points.
- Select the K-nearest examples to E in the training set.
- Assign E to the most common class among its K-nearest neighbors.

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.

14

KNN Feature Weighting

- Scale each feature by its importance for classification

$$D(a,b) = \sqrt{\sum_k \boxed{w_k} (a_k - b_k)^2}$$

- Can use our prior knowledge about which features are more important
- Can learn the weights $\mathbf{w_k}$ using **cross-validation** (to be covered later)

Feature Normalization

- Distance between neighbors could be dominated by some attributes with relatively large numbers.
  - e.g., income of customers in our previous example.

$$a_i = \frac{v_i - \min v_i}{\max v_i - \min v_i}$$

- Arises when two features are in different scales.

- Important to normalize those features.
  - Mapping values to numbers between 0 – 1.

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

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

3-KNN: Example(

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

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Example: PEBLS

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Distance between nominal attribute values:

d(Single,Married)

= | 2/4 – 0/4 | + | 2/4 – 4/4 | = 1

d(Single,Divorced)

= | 2/4 – 1/2 | + | 2/4 – 1/2 | = 0

d(Married,Divorced)

= | 0/4 – 1/2 | + | 4/4 – 1/2 | = 1

d(Refund=Yes,Refund=No)

= | 0/3 – 3/7 | + | 3/3 – 4/7 | = 6/7

Class	Marital Status		
	Single	Married	Divorced
Yes	2	0	1
No	2	4	1

Class	Refund	
	Yes	No
Yes	0	3
No	3	4

$$d(V_1, V_2) = \sum_i \left| \frac{n_{1i}}{n_1} - \frac{n_{2i}}{n_2} \right|$$

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