

Lecture 01 – Classifying images

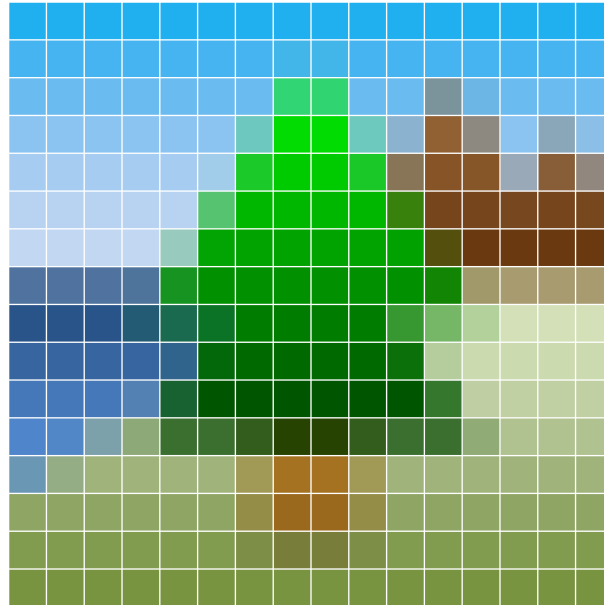
Prof. João Fernando Mari

joaofmari.github.io

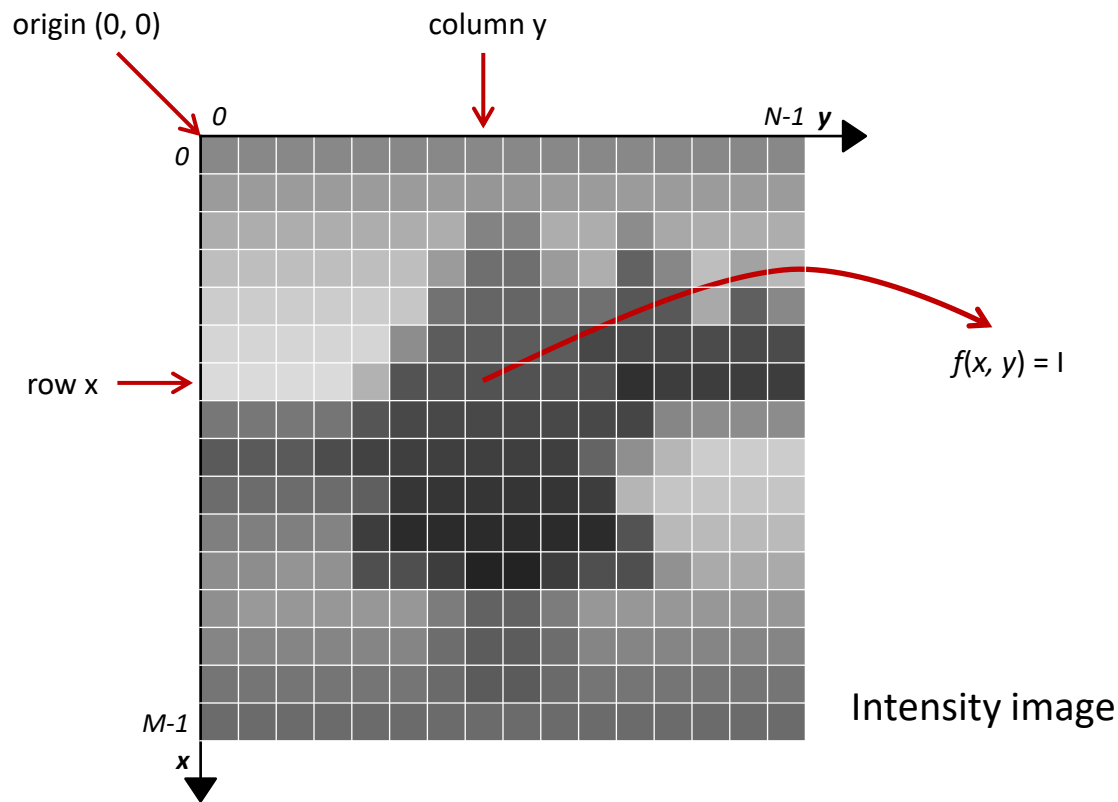
joaof.mari@ufv.br

Agenda

- Digital images
- A classification problem
- Classification pipelines
- Learning models
- Cross-validation
- Classification evaluation

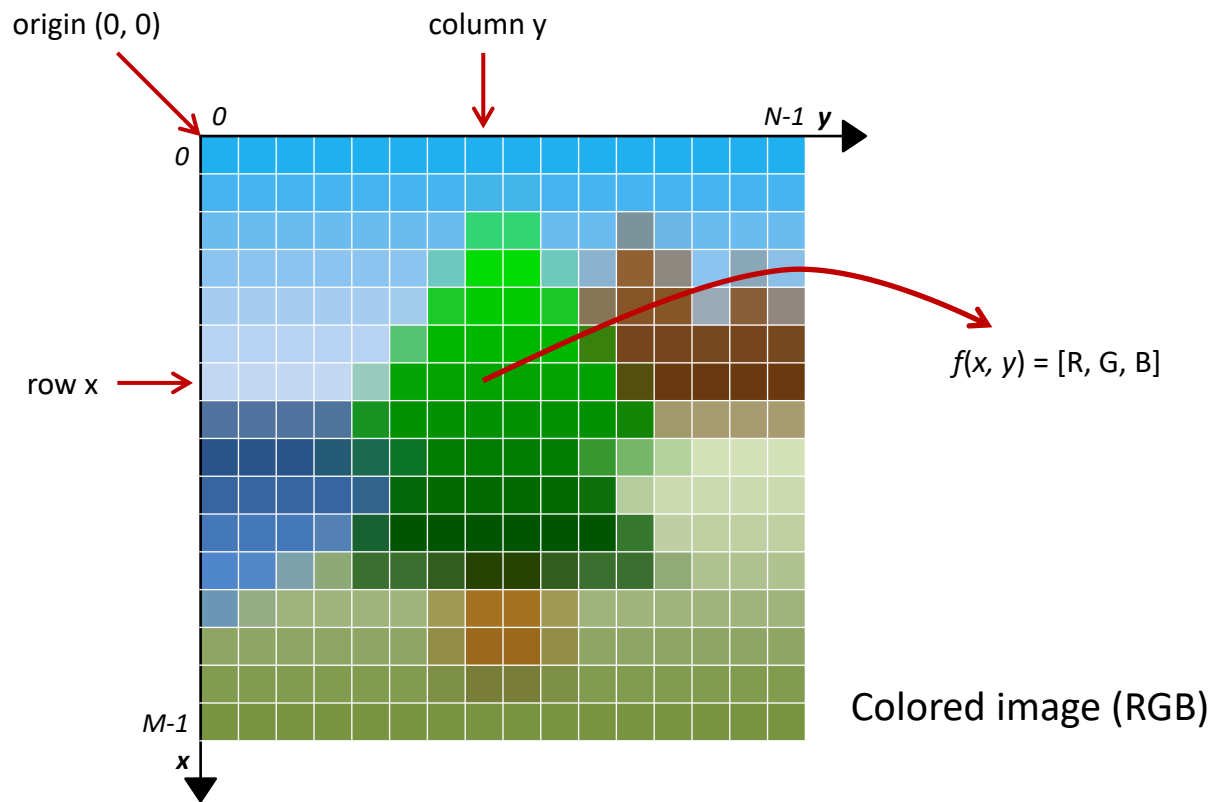


Digital images



M rows
N columns
M x N pixels

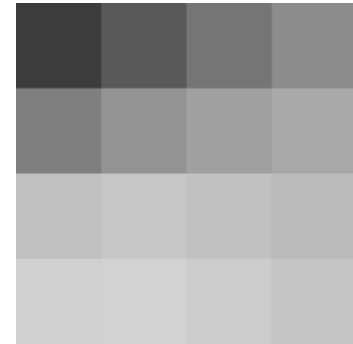
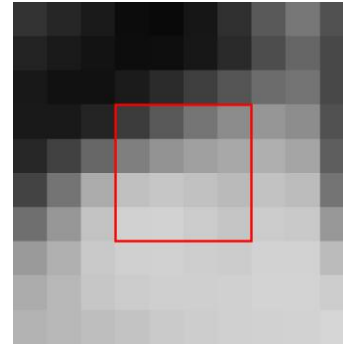
Digital images



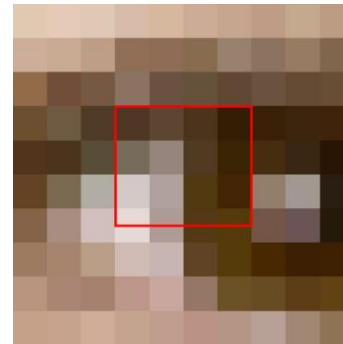
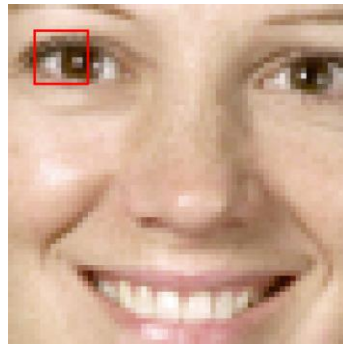
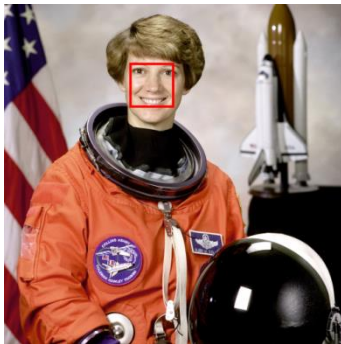
M rows
N columns
M x N pixels

Colored image (RGB)

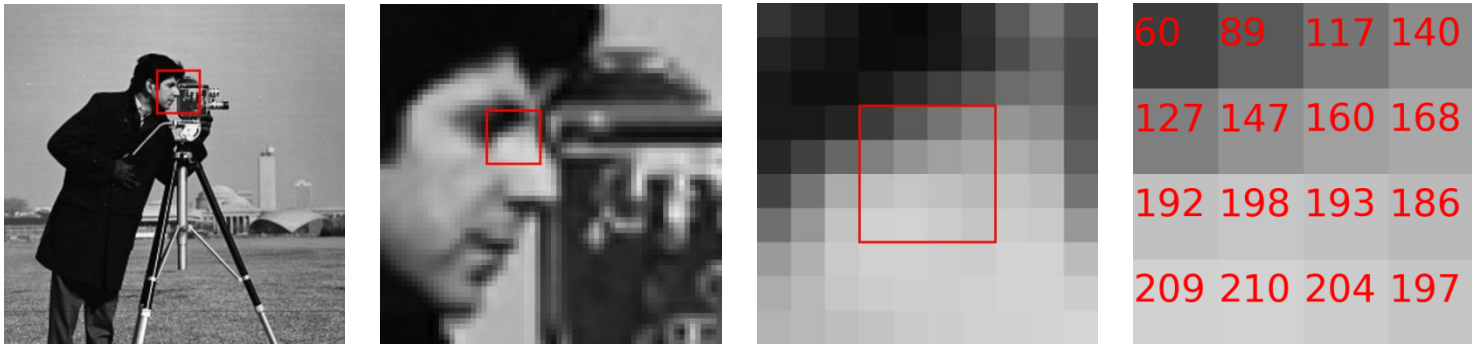
Intensity image (gray levels):



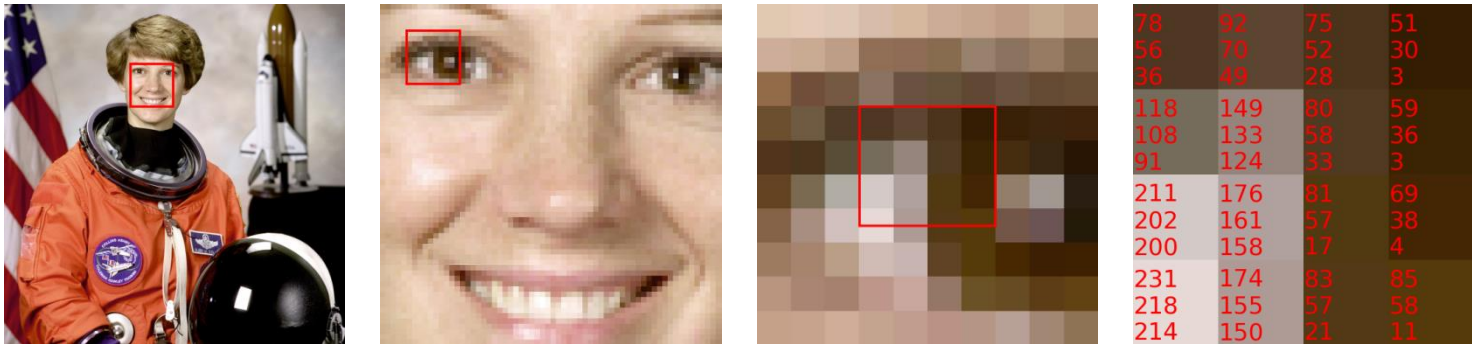
Colored image (RGB):



Intensity image (gray levels):



Colored image (RGB):





vermelho – R (red)



verde – G (green)



azul – B (blue)





red – R (red)



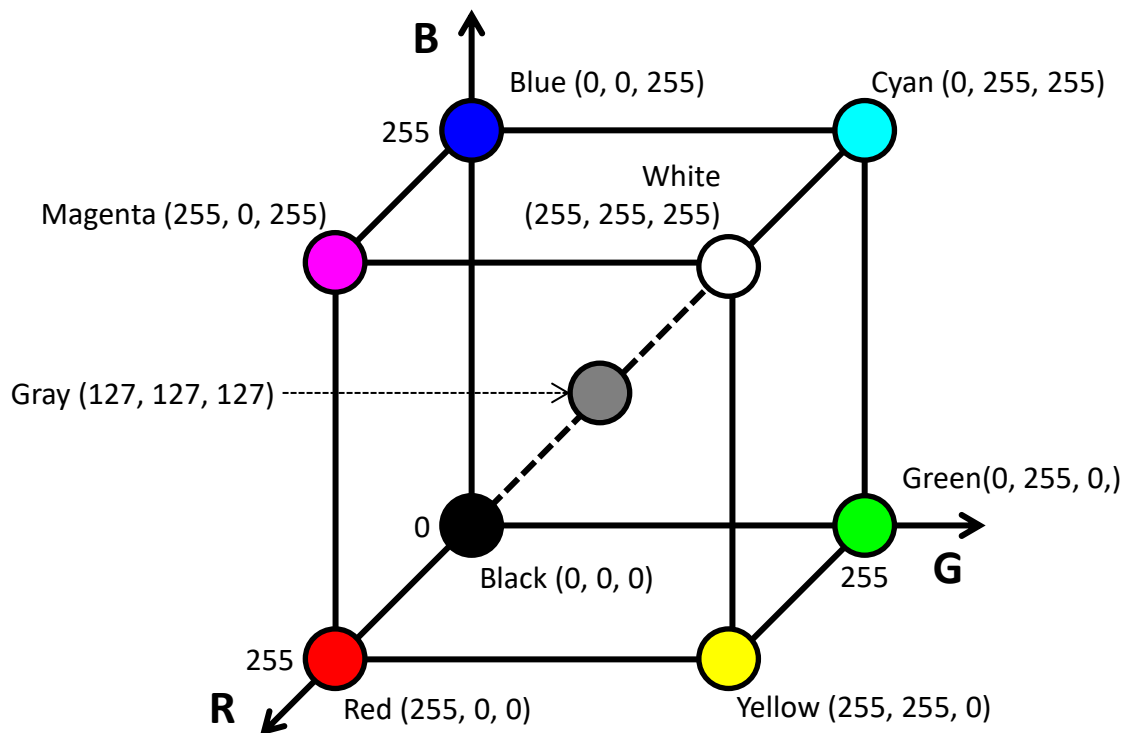
green – G (green)



blue – B (blue)



Colored images – RGB



A CLASSIFICATION PROBLEM

A classification problem

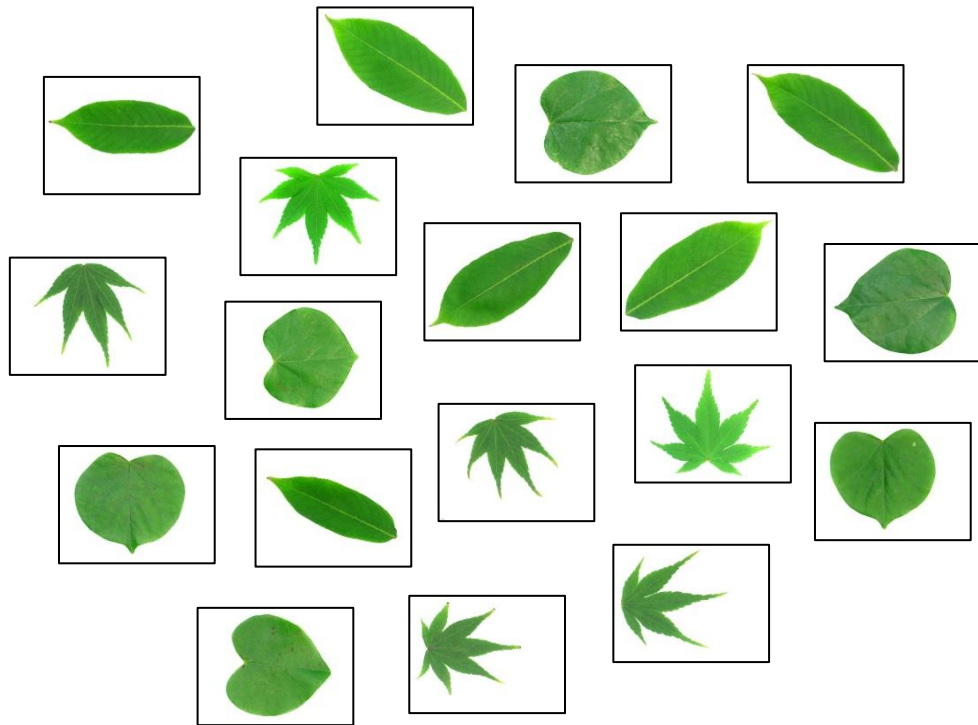
- Learning to classify three types (classes) of leaves from images.

- Flavia leaf dataset:

- <http://flavia.sourceforge.net/>
- 1,907 images
- 33 classes

- We selected 3 classes:

- *aesculus chinensis*
- *acer palmatum*
- *cercis chinensis*



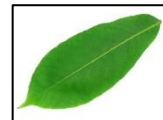
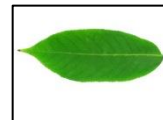
A classification problem

- Learning to classify three types (classes) of leaves from images.

- Feature extraction:
 - Select features from the images that can be used to distinguish between the classes.

- Features can be:
 - Shapes
 - Colors
 - Textures
 - Histogram of gradients (HoG)
 - Bag of Visual Words*
 - Fisher Vectors*
 - ...

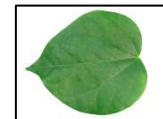
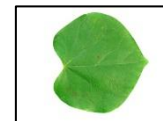
aesculus chinensis



acer palmatum



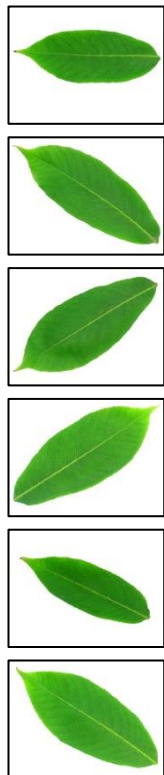
cercis chinensis



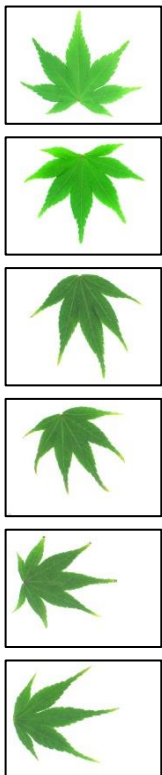
A classification problem

- Some feature shapes:

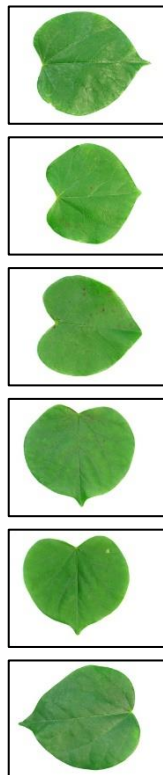
aesculus chinensis



acer palmatum



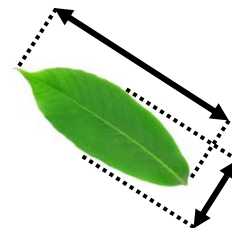
cercis chinensis



Area:

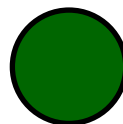


Axis:



Roundness:

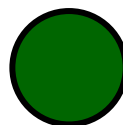
1,0



0,0

Excentricity:

0,0



1,0

Solidity:



÷



÷

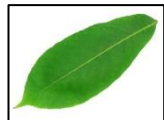
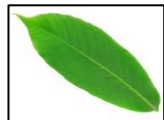
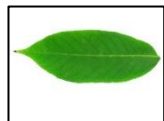


<https://scikit-image.org/docs/stable/api/skimage.measure.html#skimage.measure.regionprops>

A classification problem

- Some feature shapes:

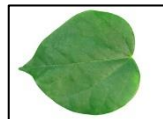
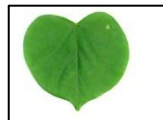
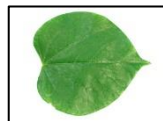
aesculus chinensis



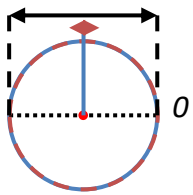
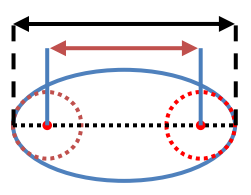
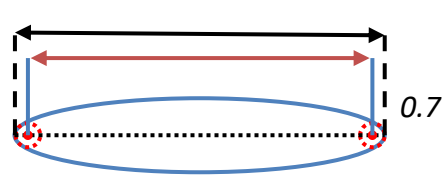
acer palmatum



cercis chinensis






- Area:** the number of pixels of the shape scaled by pixel-area.
- Axis:** the length of the major and minor axis of the ellipse with the same normalized second central moments as the region.
- Roundness:** a function of the perimeter and the area of the region
 - $$roundness = \frac{4 \times \pi \times area}{perimeter^2}$$
- Eccentricity:** the ratio of the focal distance over the major axis length of the ellipse with the same second-moments.

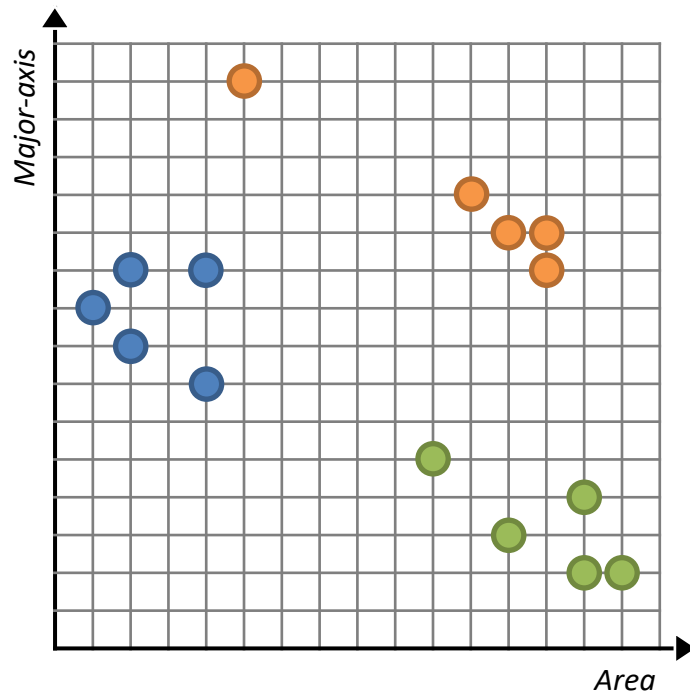
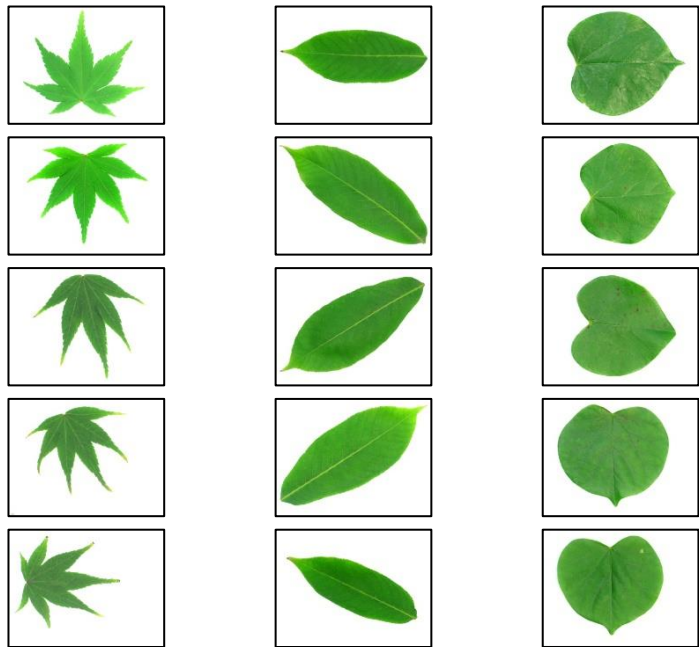



- Solidity:** the ratio of pixels in the region to pixels of the convex hull image.
 - Convex hull:** the smallest convex polygon that encloses the region.

<https://scikit-image.org/docs/stable/api/skimage.measure.html#skimage.measure.regionprops>

A classification problem

- Learning to classify three types (classes) of leaves from images.

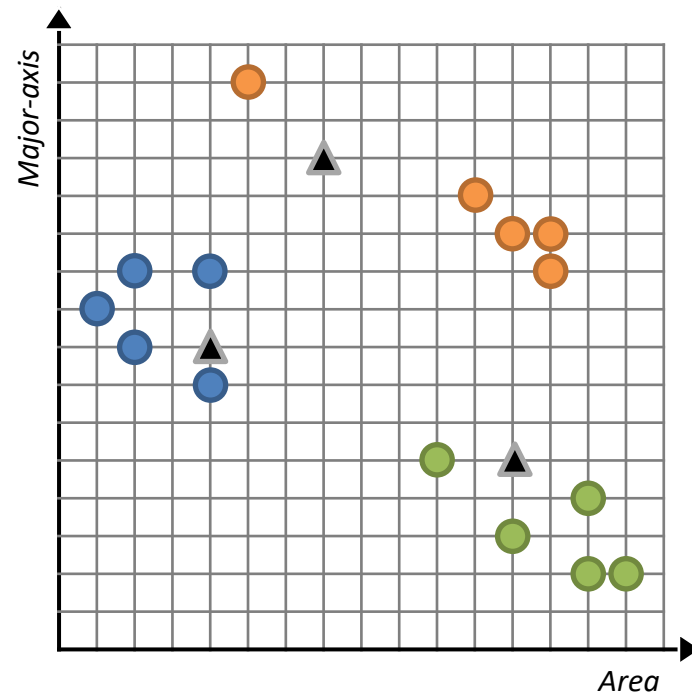
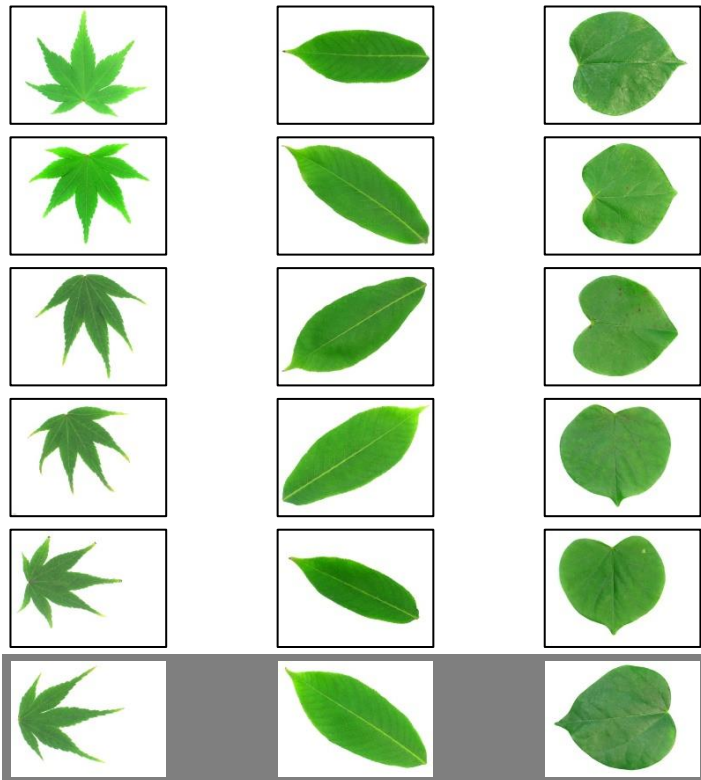
 *acer palmatum*
 *aesculus chinensis*
 *cercis chinensis*



A classification problem

- Learning to classify three types (classes) of leaves from images.

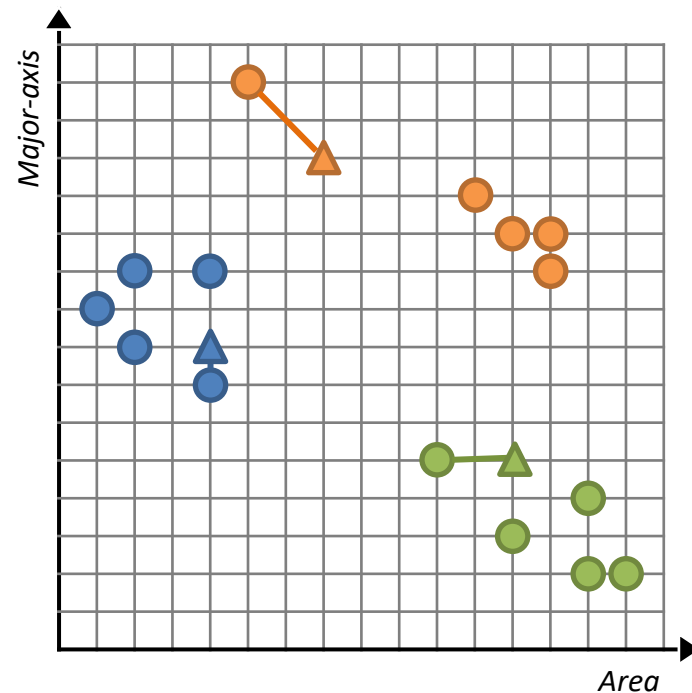
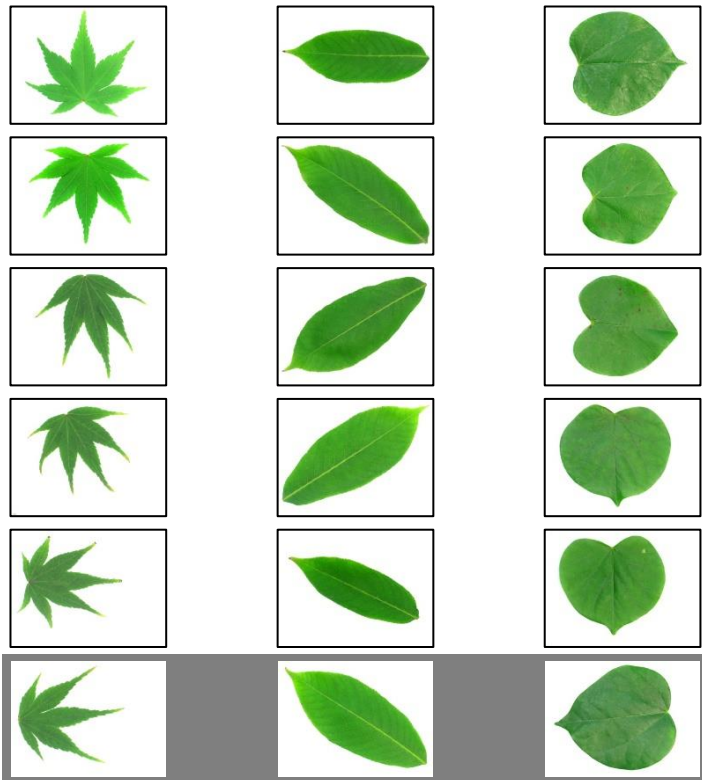
● *acer palmatum*
● *aesculus chinensis*
● *cercis chinensis*



The nearest neighbor method

- Learning to classify three types (classes) of leaves from images.

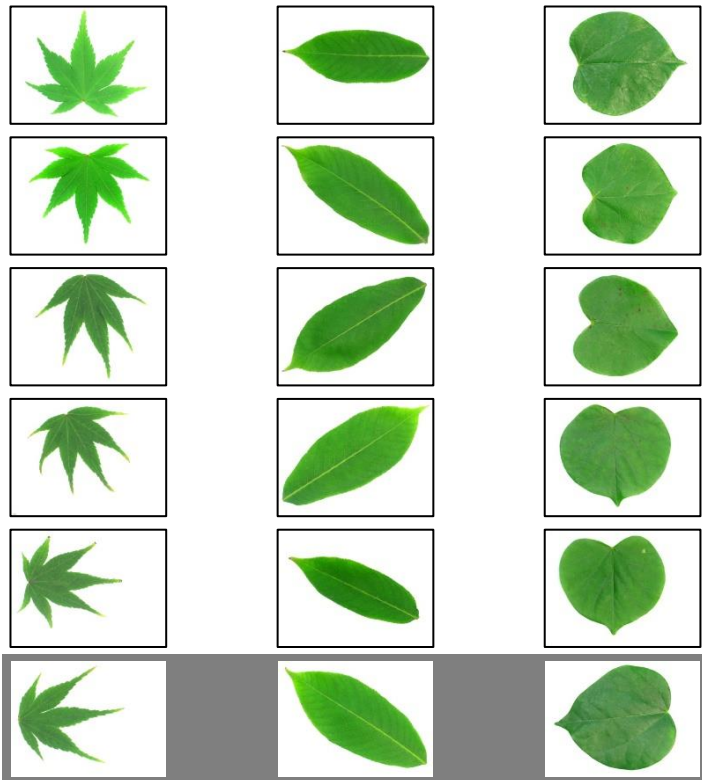
● *acer palmatum*
● *aesculus chinensis*
● *cercis chinensis*



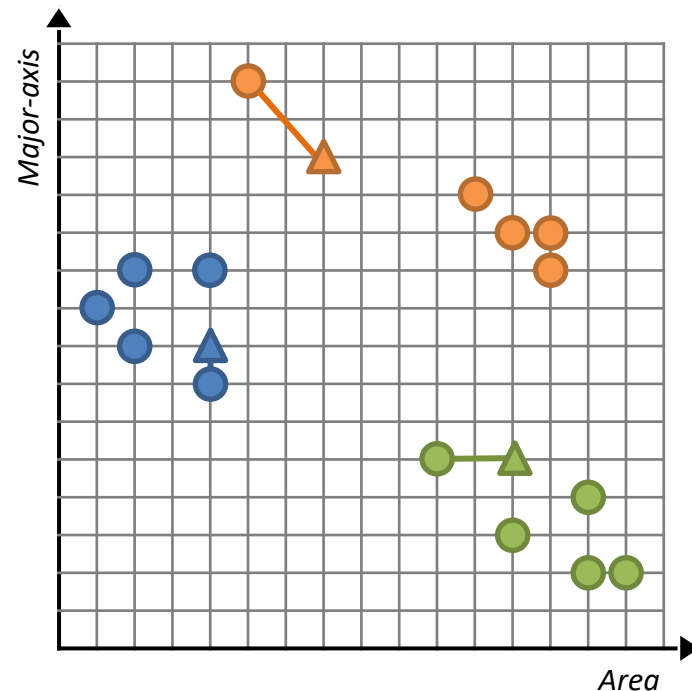
K-nearest neighbors – K-NN

- Learning to classify three types (classes) of leaves from images.

● *acer palmatum*
 ● *aesculus chinensis*
 ● *cercis chinensis*






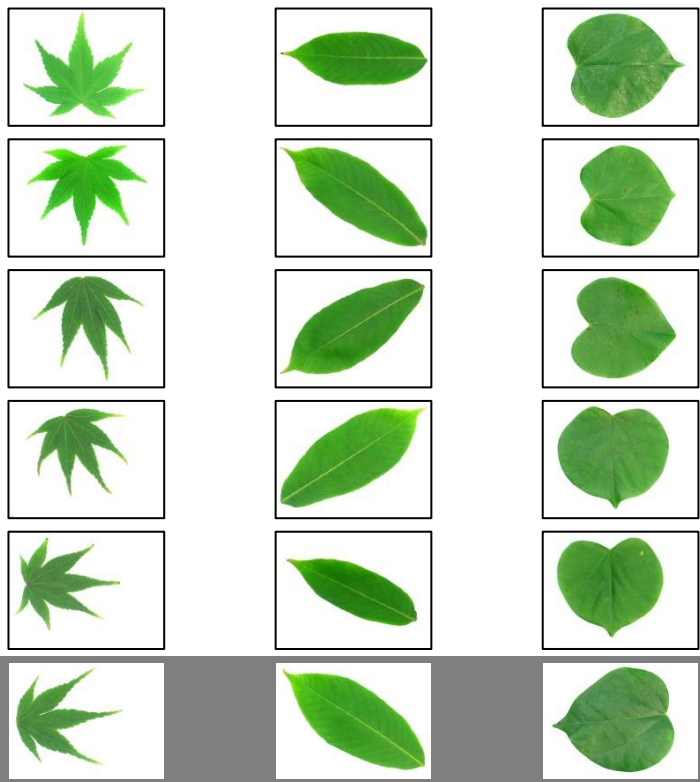
$k = 1$



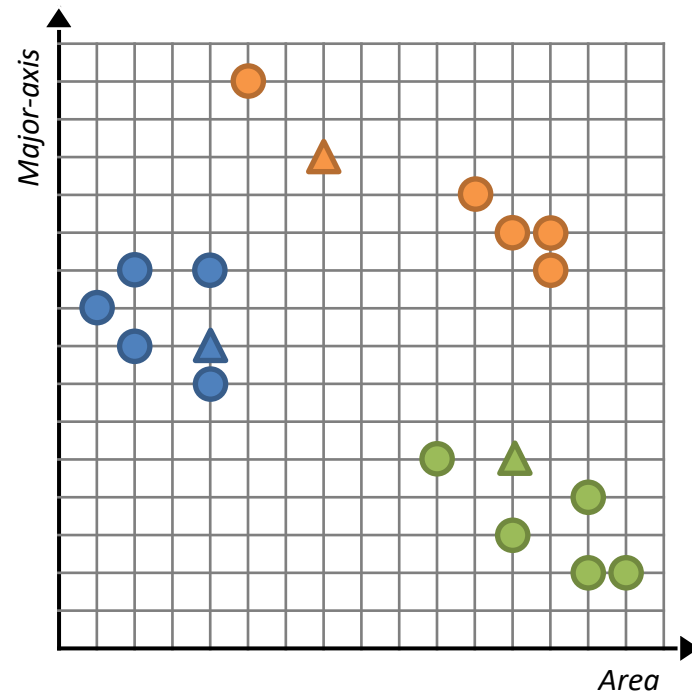
K-nearest neighbors – K-NN

- Learning to classify three types (classes) of leaves from images.

 *acer palmatum*
 *aesculus chinensis*
 *cercis chinensis*

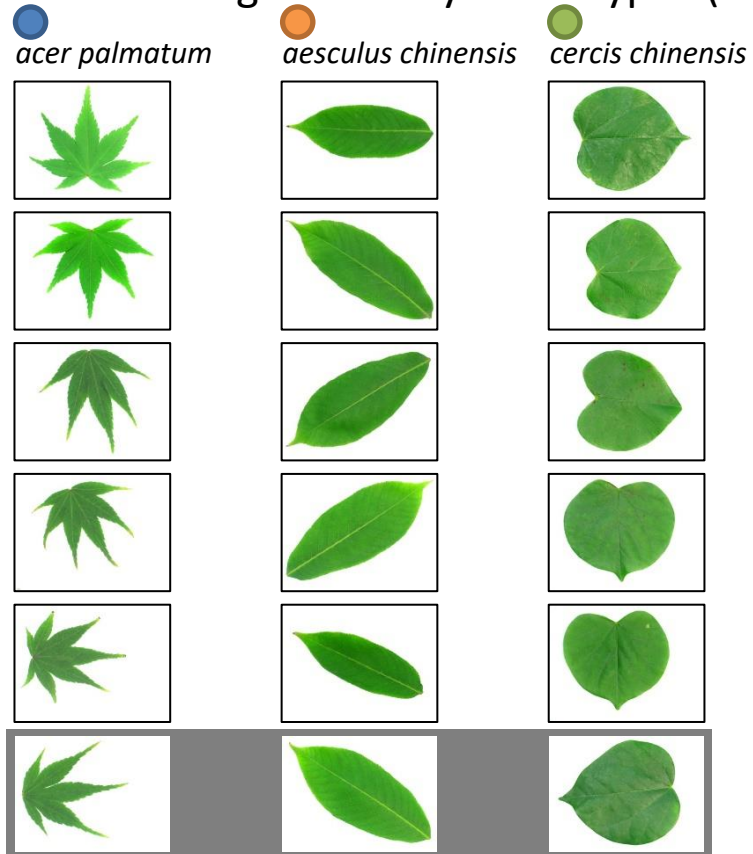


$k = 3$

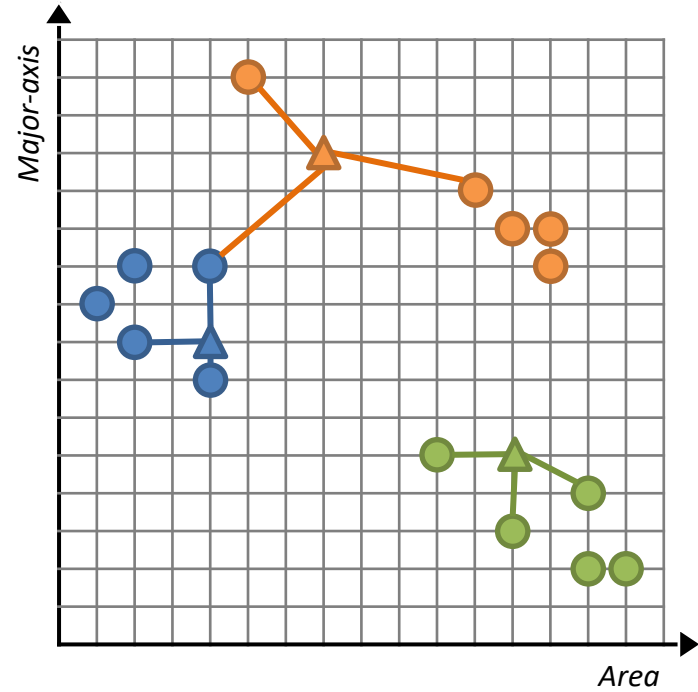


K-nearest neighbors – K-NN

- Learning to classify three types (classes) of leaves from images.



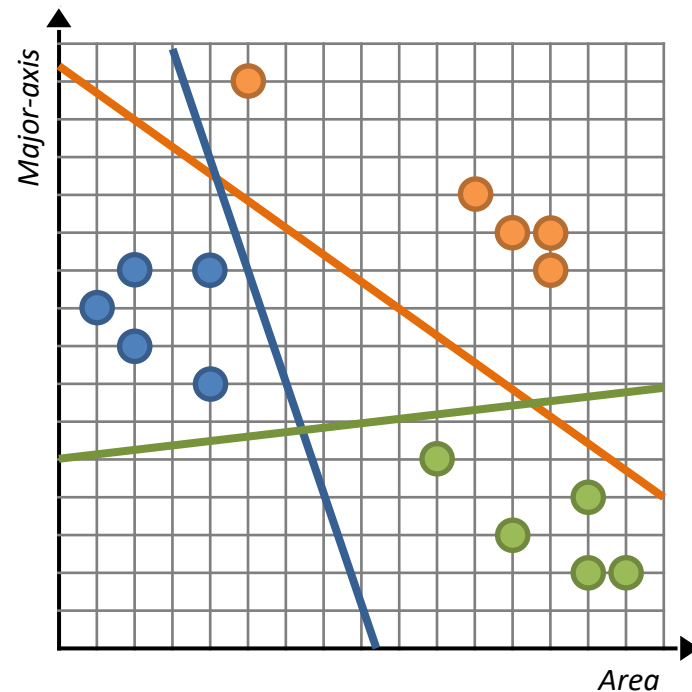
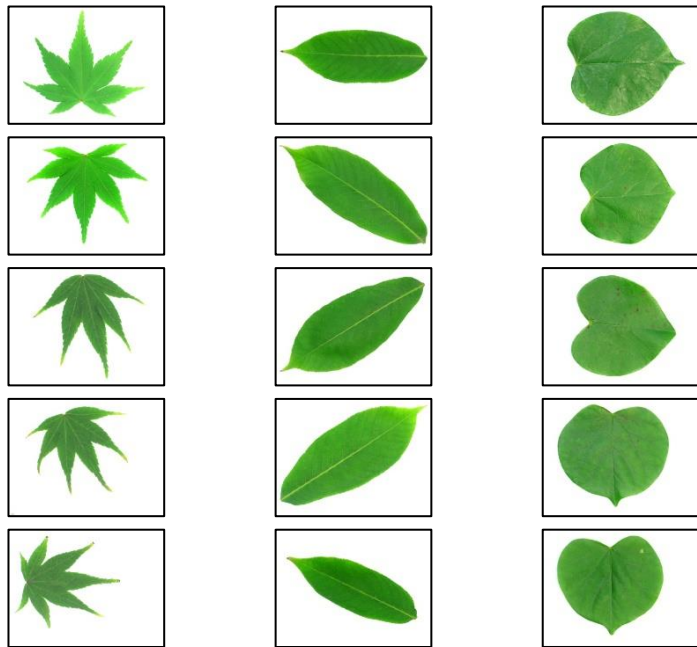
$k = 3$



Linear functions (Perceptrons)

- Learning to classify three types (classes) of leaves from images.

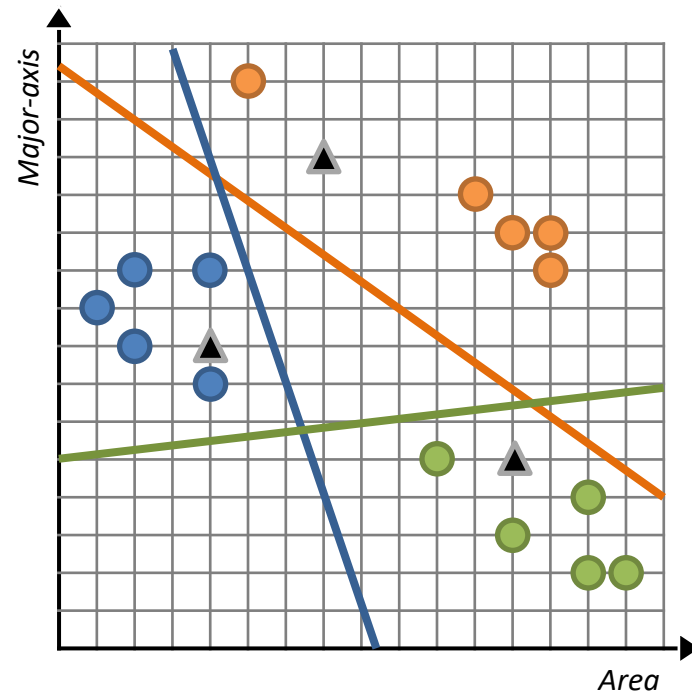
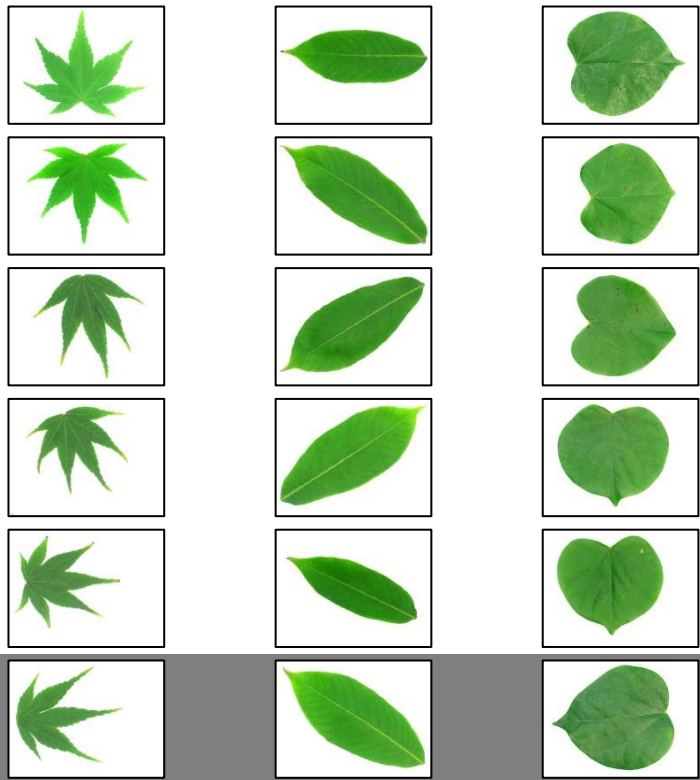
● *acer palmatum*
● *aesculus chinensis*
● *cercis chinensis*



Linear functions (Perceptrons)

- Learning to classify three types (classes) of leaves from images.

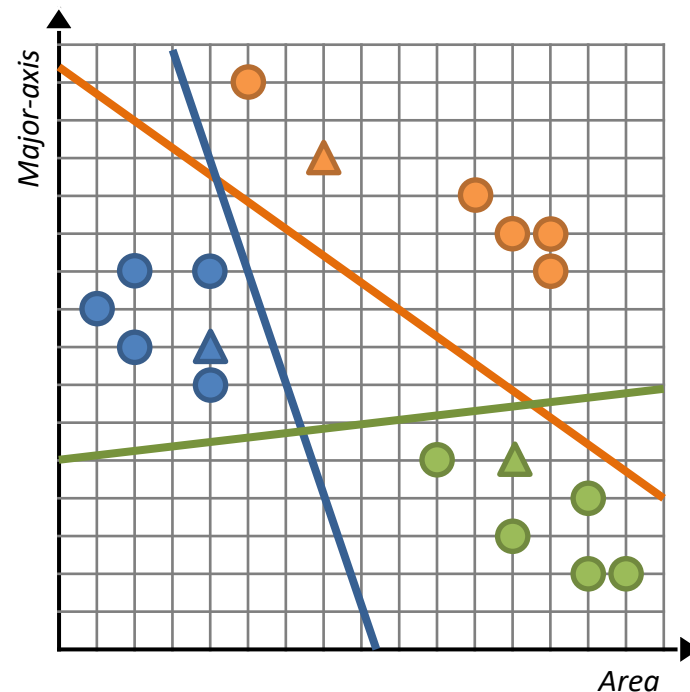
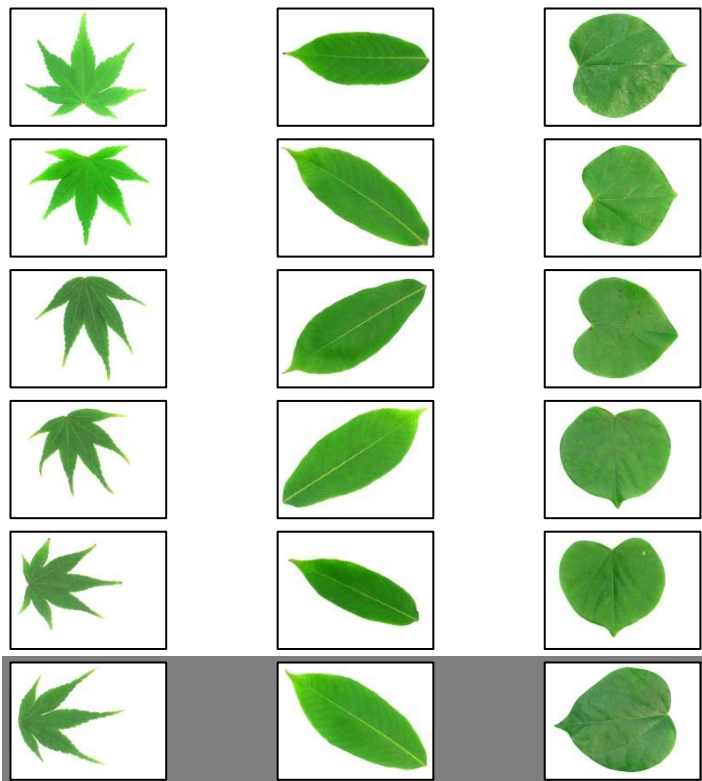
● *acer palmatum*
● *aesculus chinensis*
● *cercis chinensis*



Linear functions (Perceptrons)

- Learning to classify three types (classes) of leaves from images.

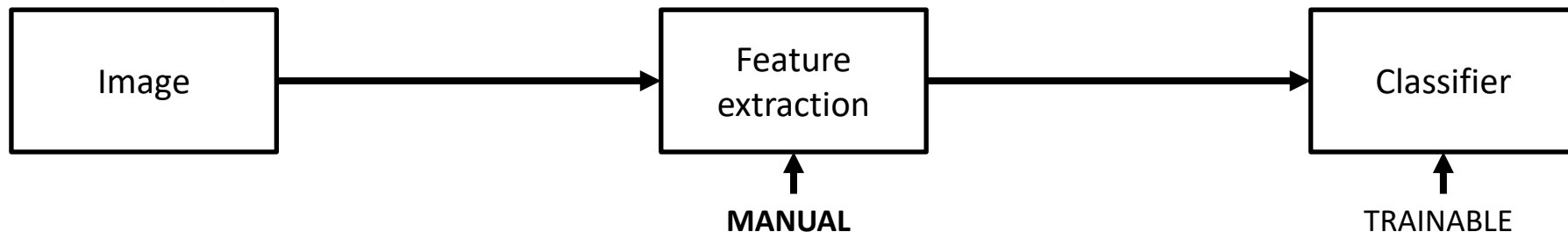
● *acer palmatum*
● *aesculus chinensis*
● *cercis chinensis*



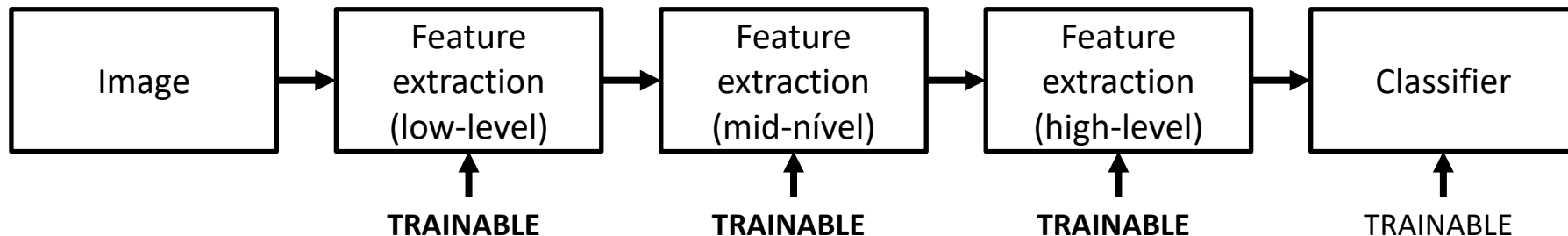
CLASSIFICATION PIPELINES

Classification pipelines

The classic image classification pipeline



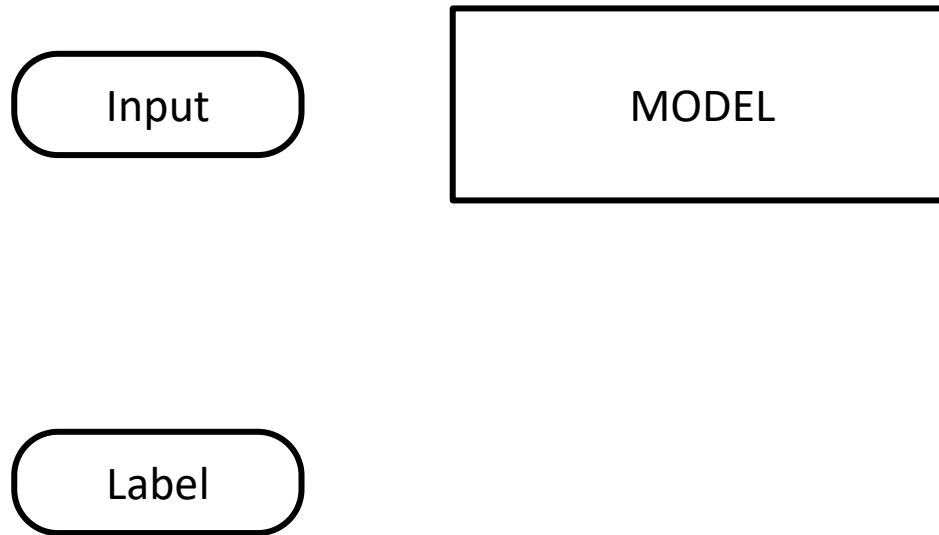
Deep Learning

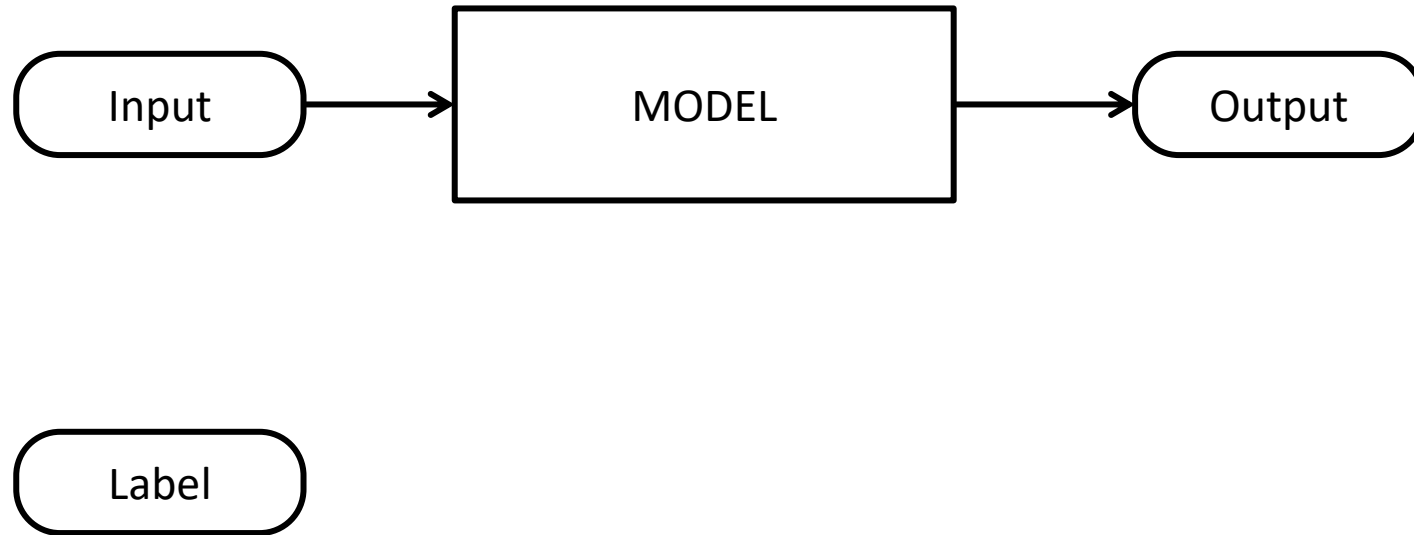


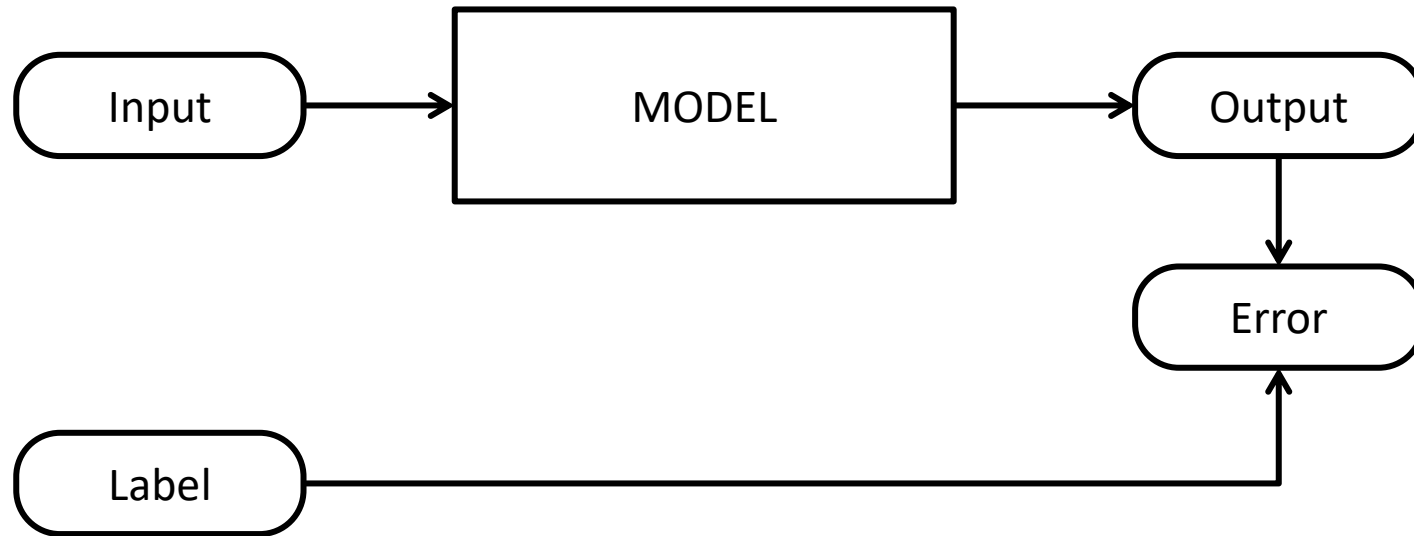
LEARNING MODELS

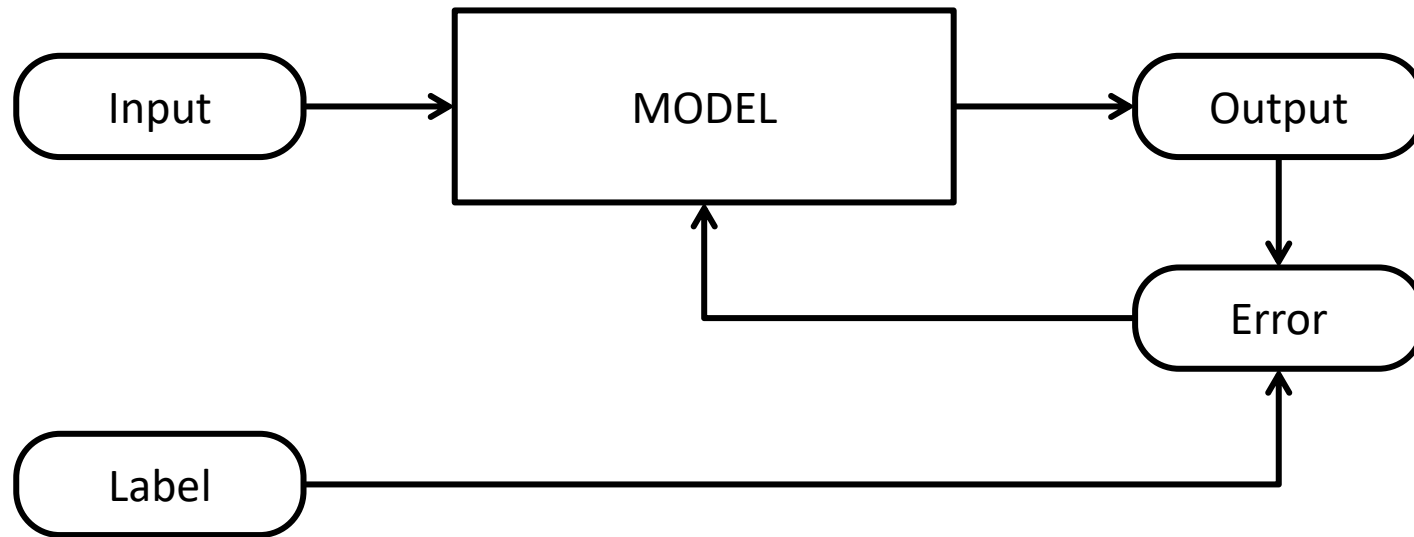
Learning models

- Supervised learning
- Unsupervised learning
- Reinforcement learning
- Semi-supervised learning
- Self-supervised learning

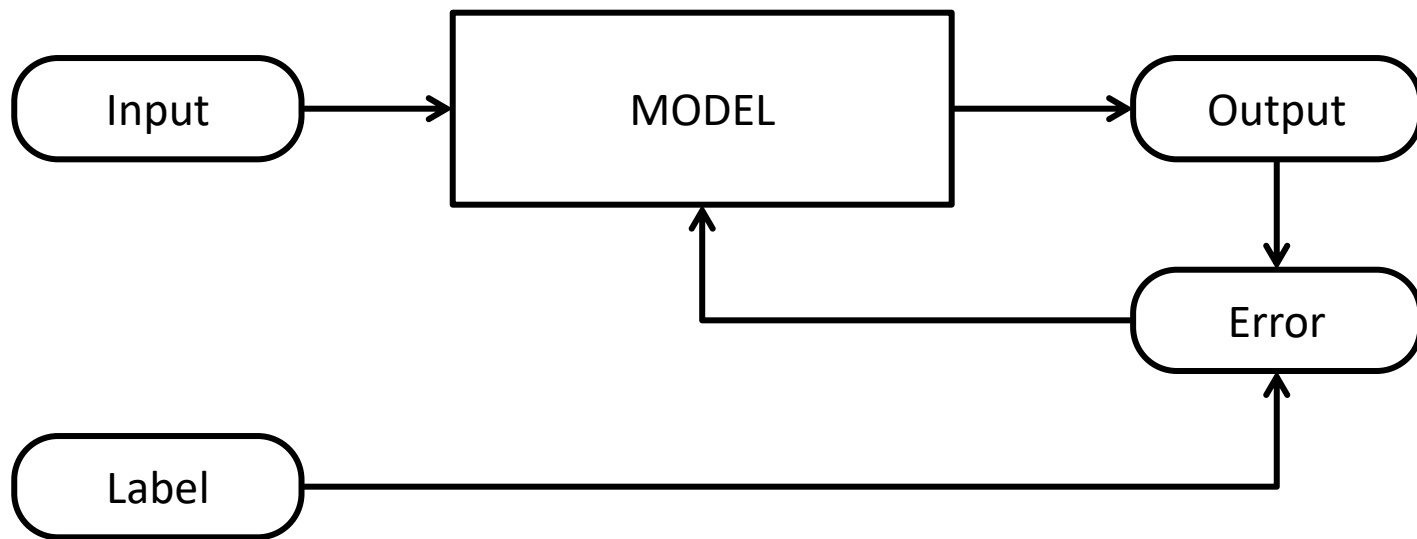
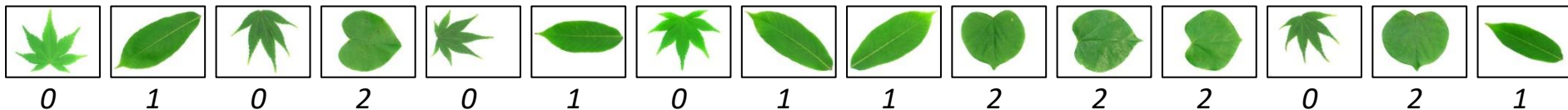









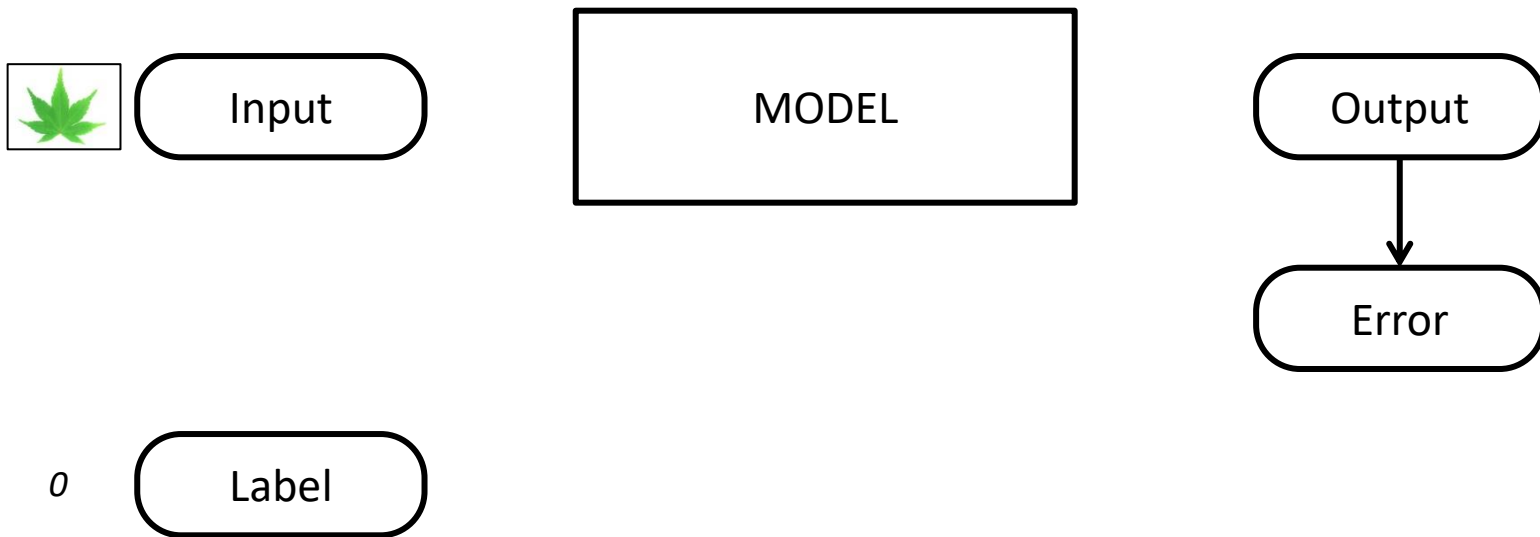
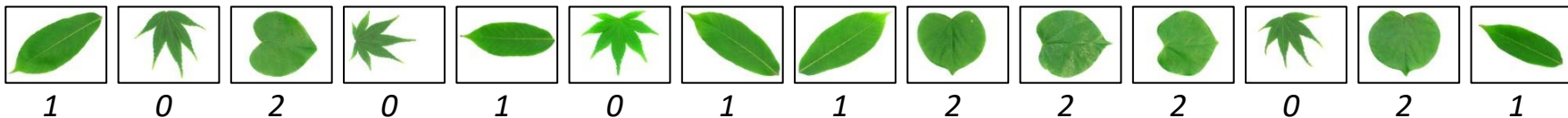





Example: Supervised learning



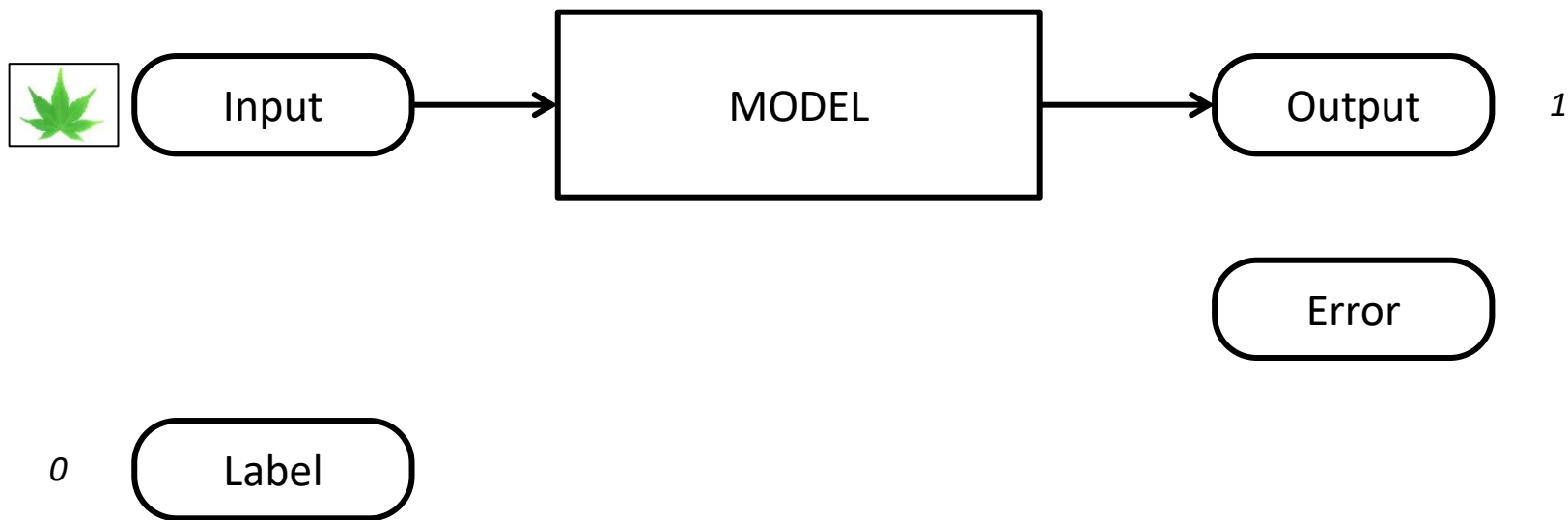
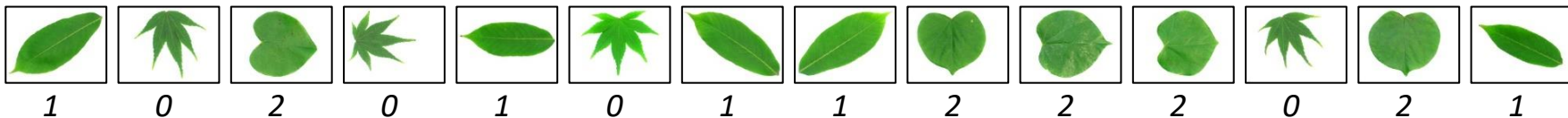
 0: *acer palmatum*
 1: *aesculus chinensis*
 2: *cercis chinensis*




Example: Supervised learning



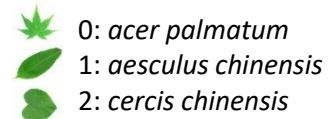
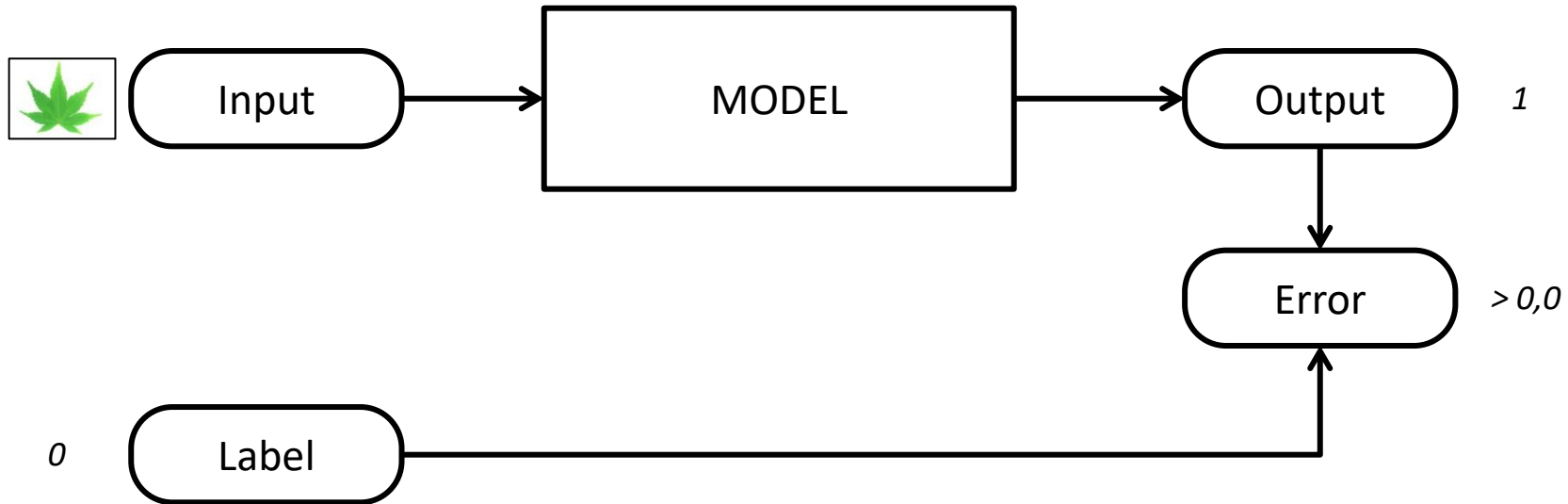
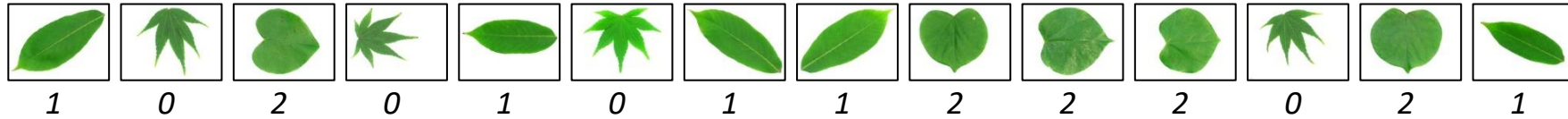
 0: *acer palmatum*
 1: *aesculus chinensis*
 2: *cercis chinensis*

Example: Supervised learning

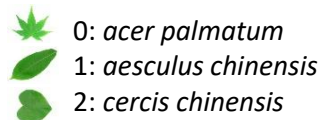
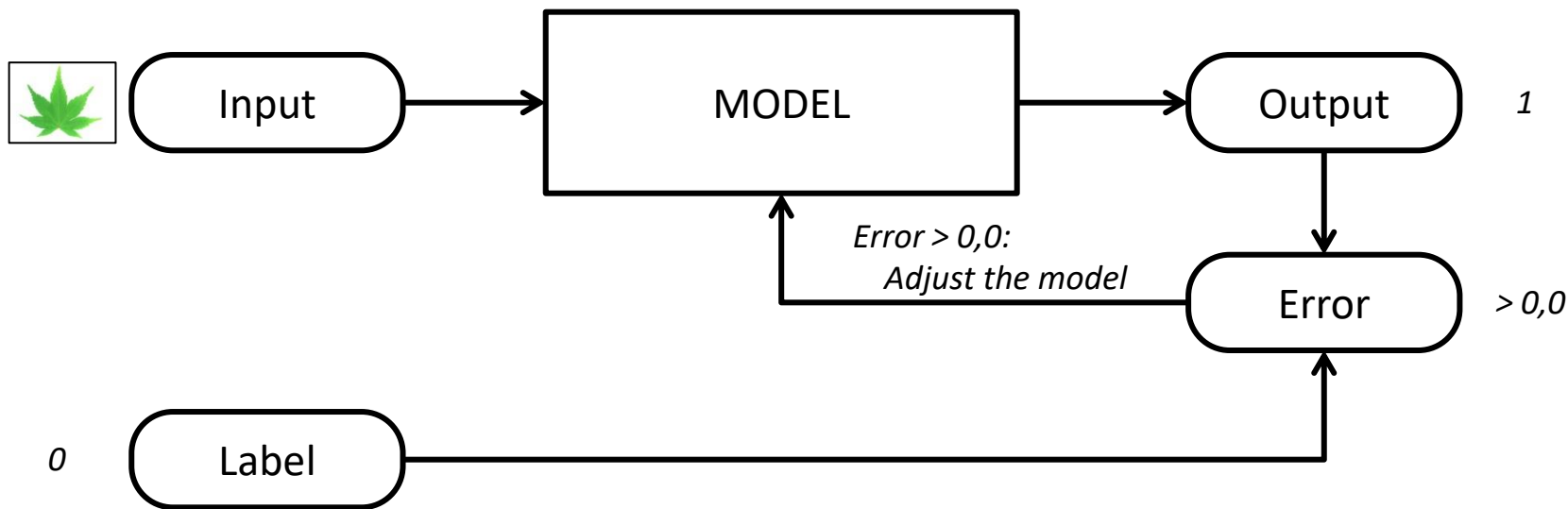
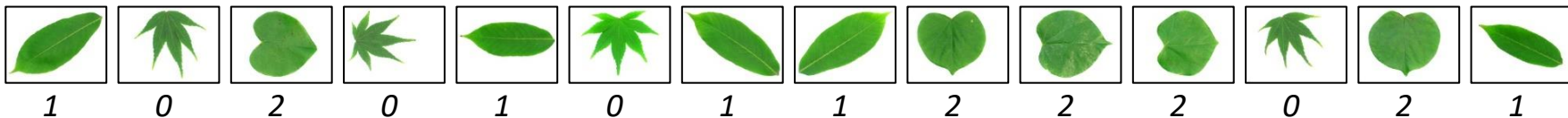


 0: *acer palmatum*
 1: *aesculus chinensis*
 2: *cercis chinensis*

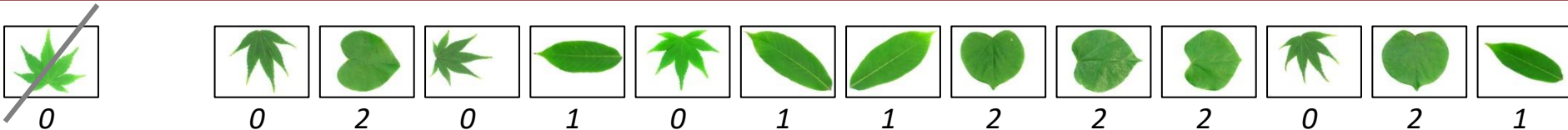
Example: Supervised learning



Example: Supervised learning



Example: Supervised learning



Input




MODEL

Output

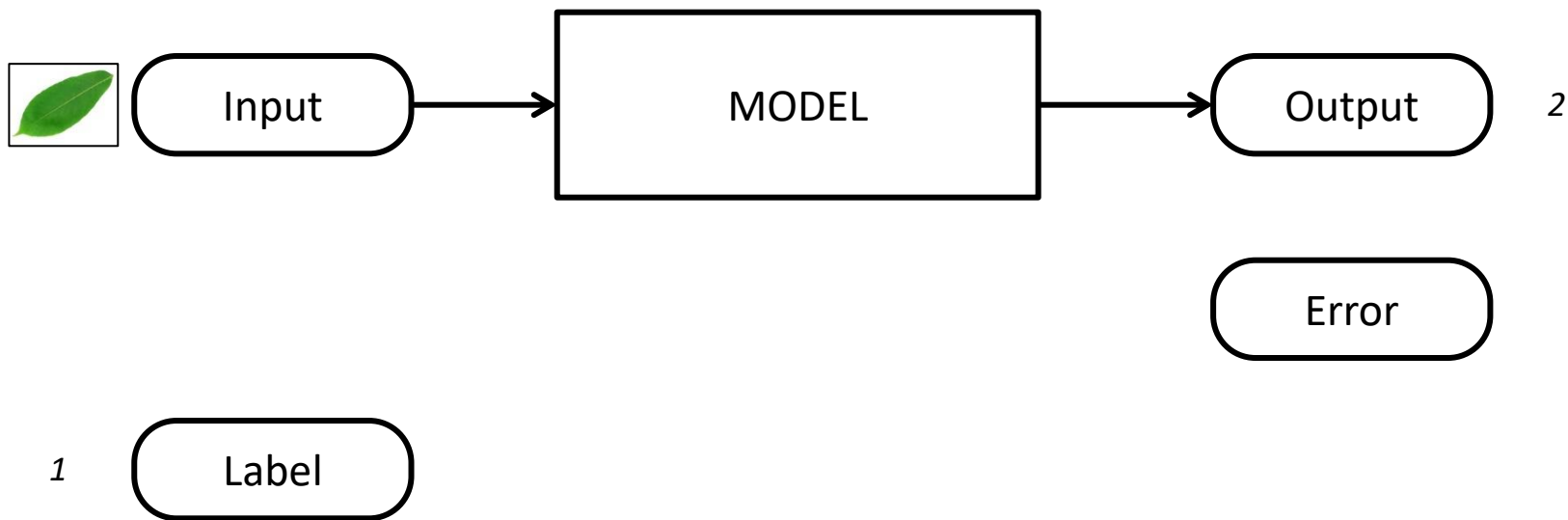
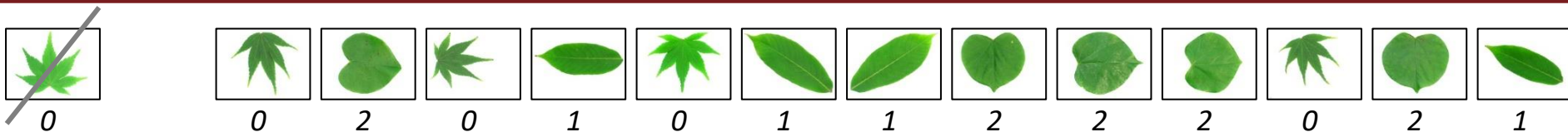
Error




1

Label

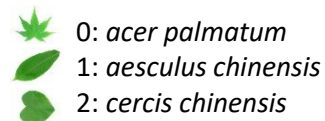
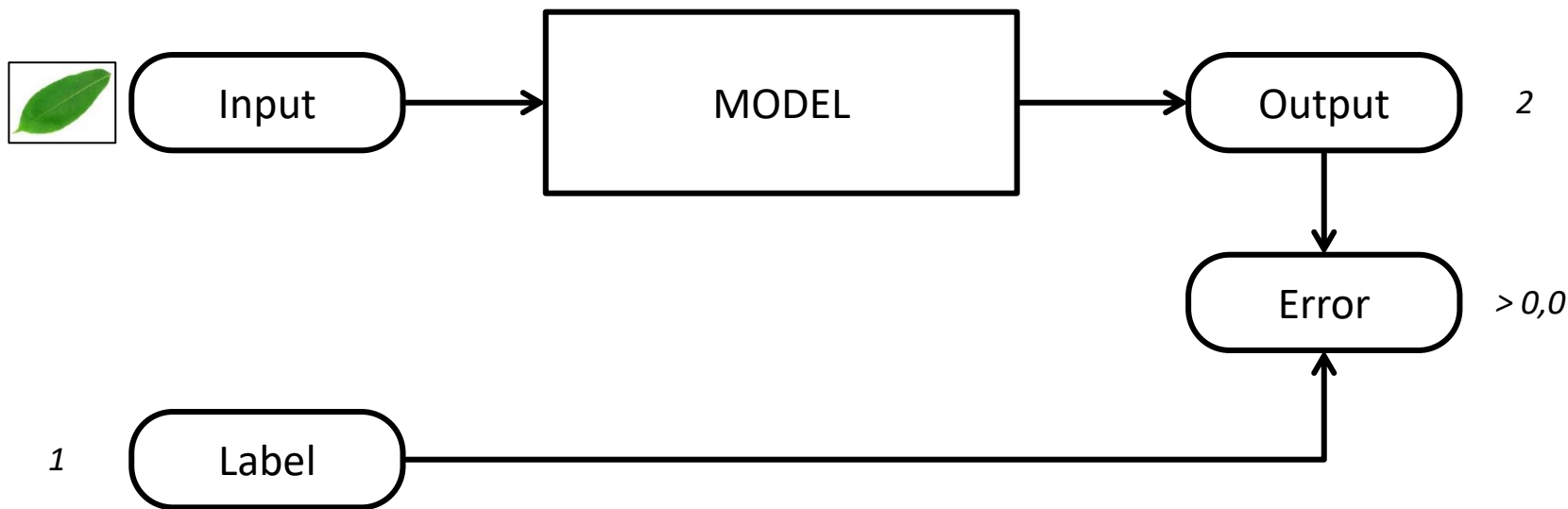
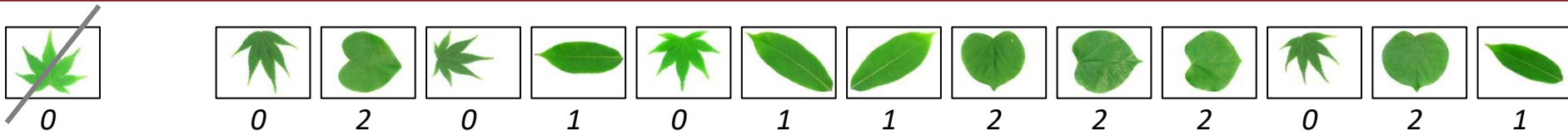
 0: *acer palmatum*
 1: *aesculus chinensis*
 2: *cercis chinensis*

Example: Supervised learning

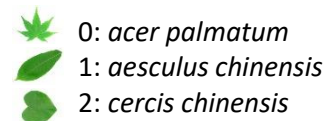
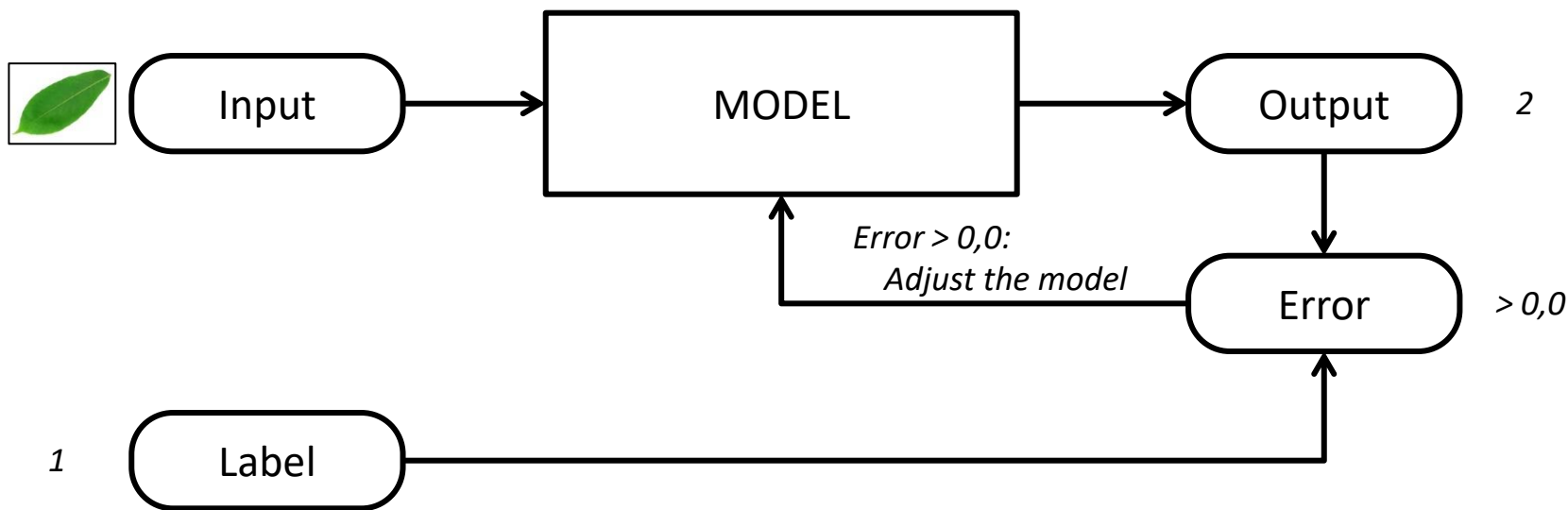
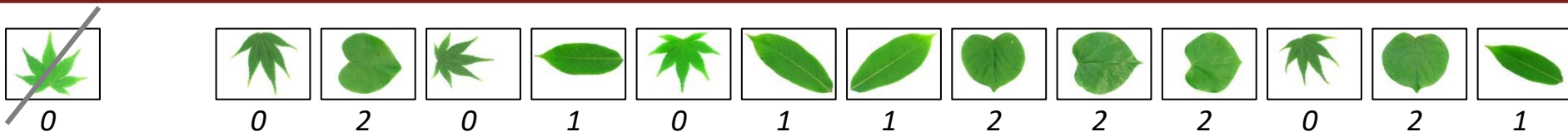


 0: *acer palmatum*
 1: *aesculus chinensis*
 2: *cercis chinensis*

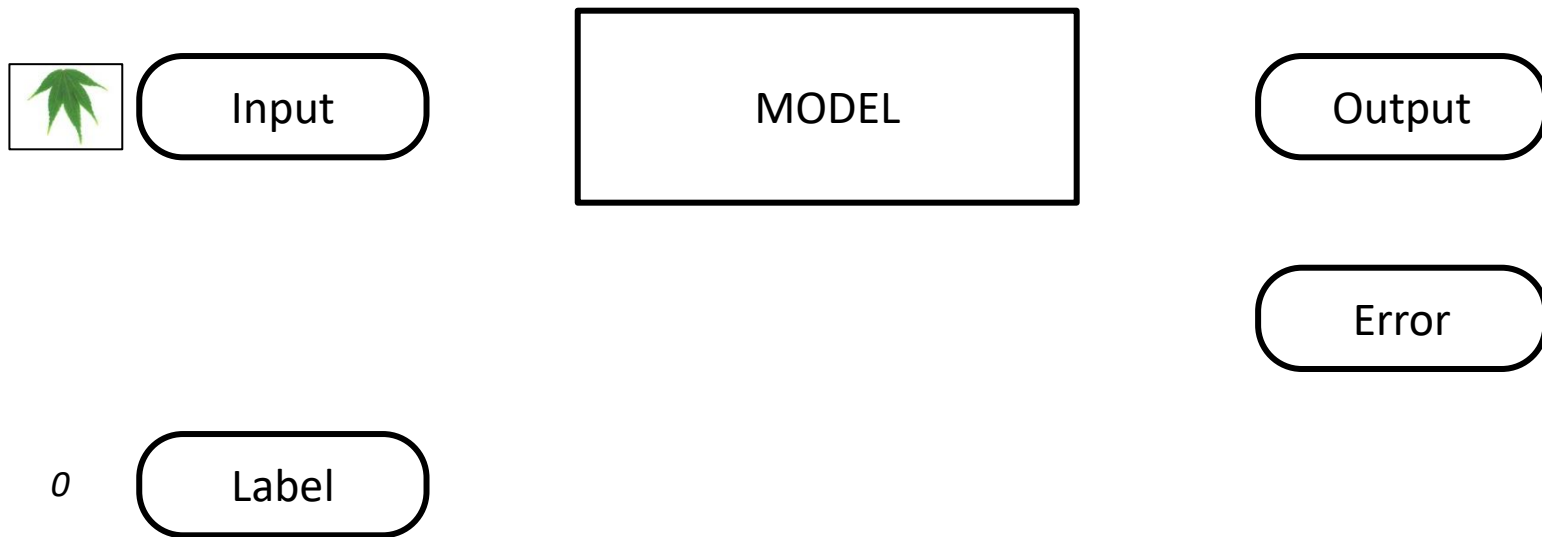
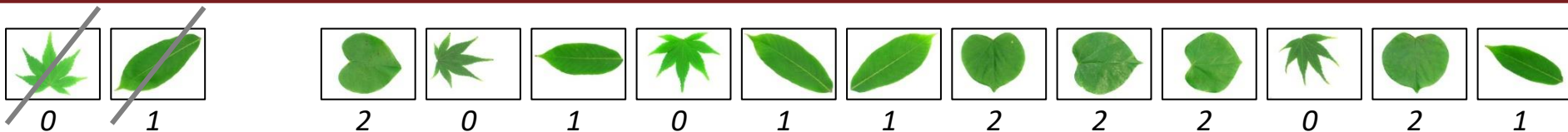
Example: Supervised learning






Example: Supervised learning

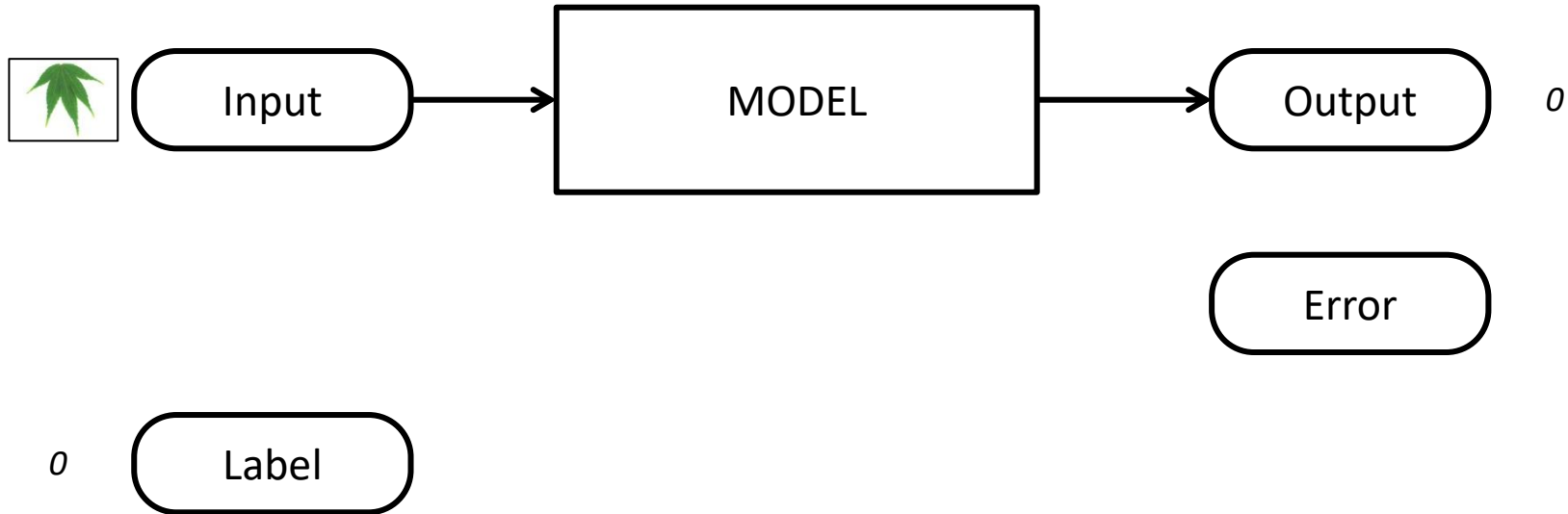
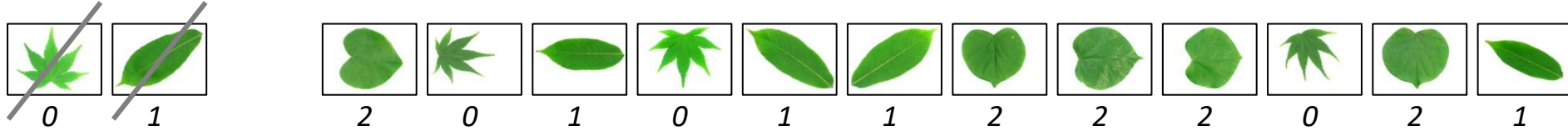





Example: Supervised learning



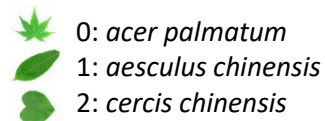
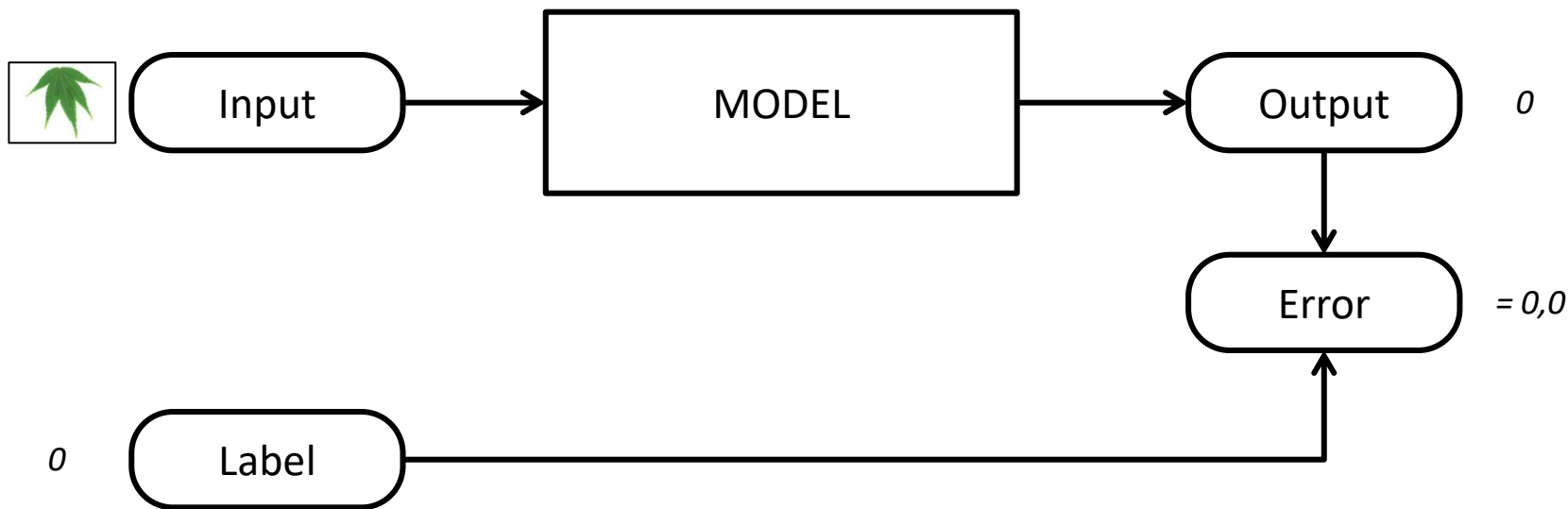
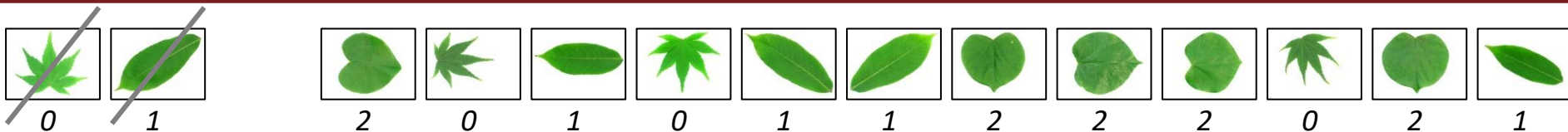
 0: *acer palmatum*
 1: *aesculus chinensis*
 2: *cercis chinensis*

Example: Supervised learning

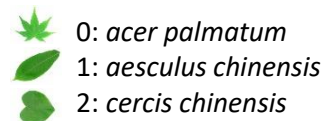
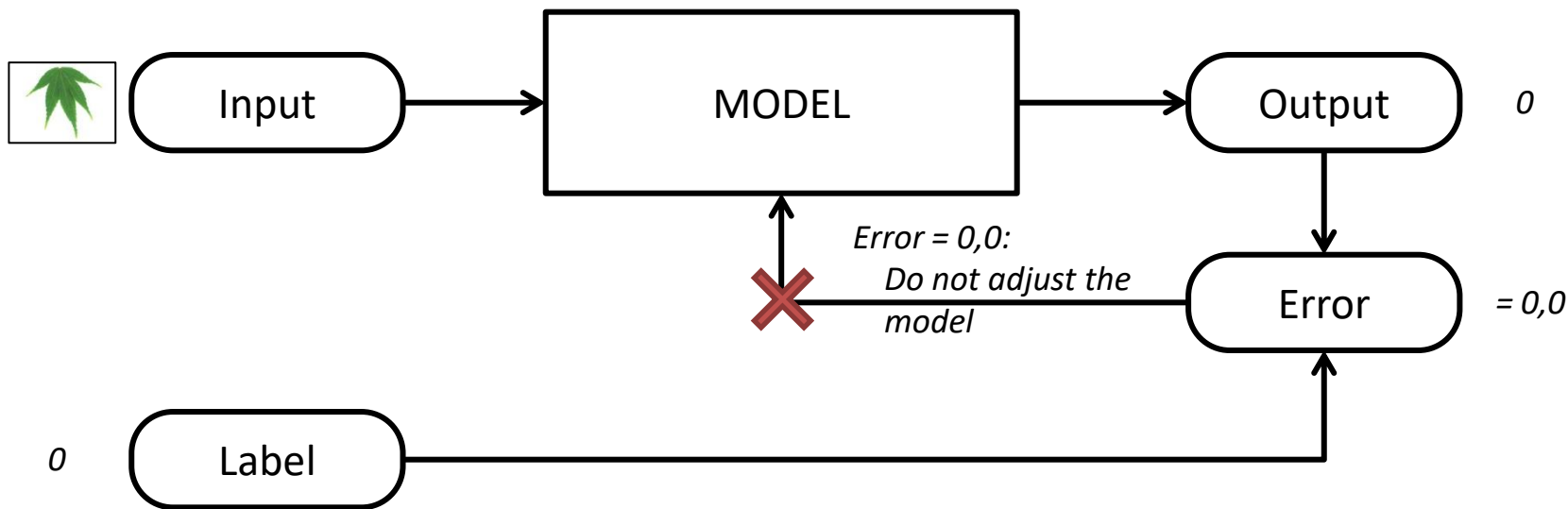
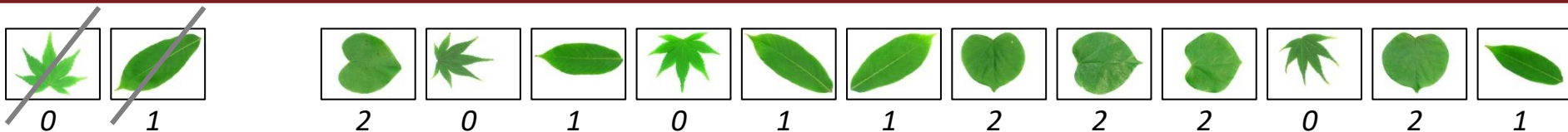


 0: *acer palmatum*
 1: *aesculus chinensis*
 2: *cercis chinensis*

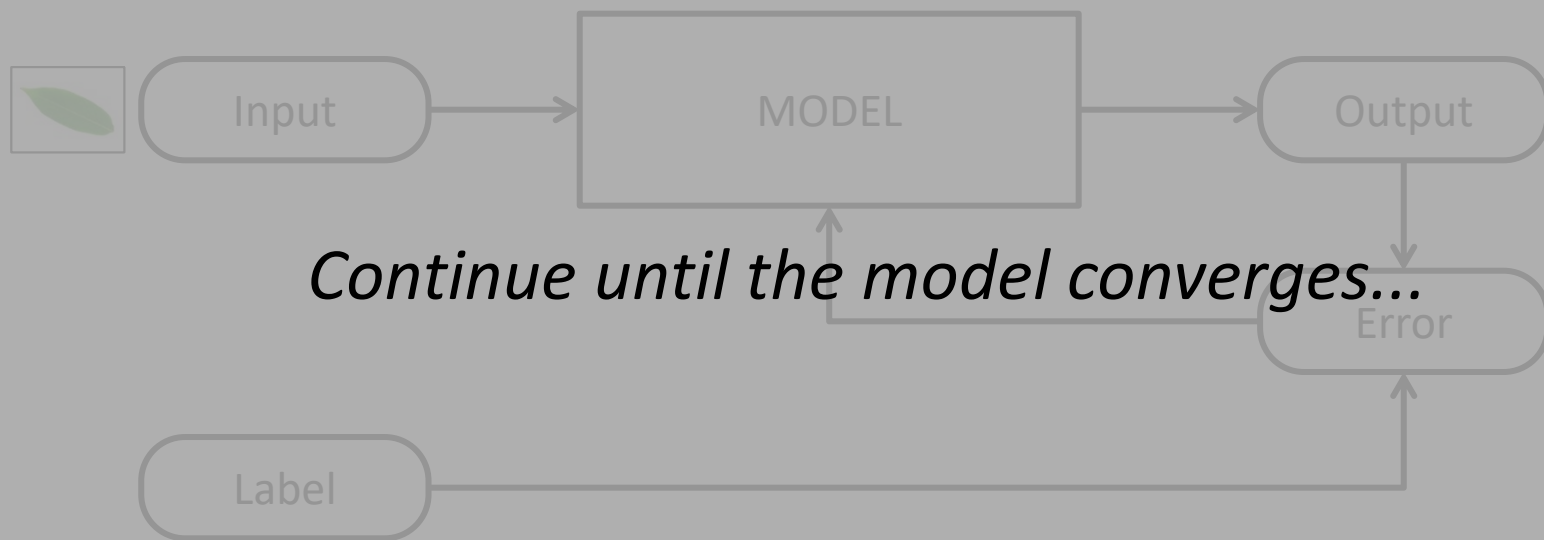
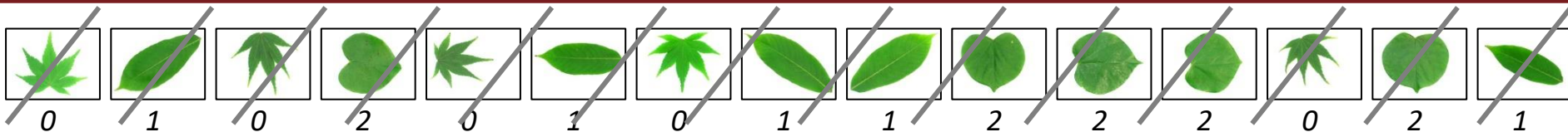
Example: Supervised learning






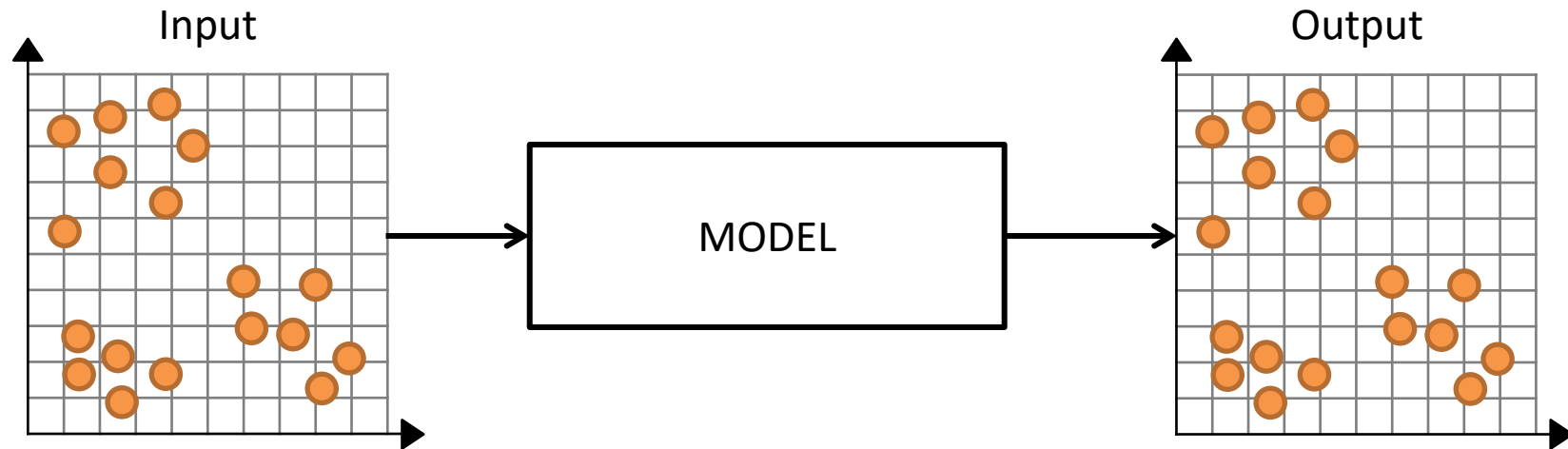
Example: Supervised learning

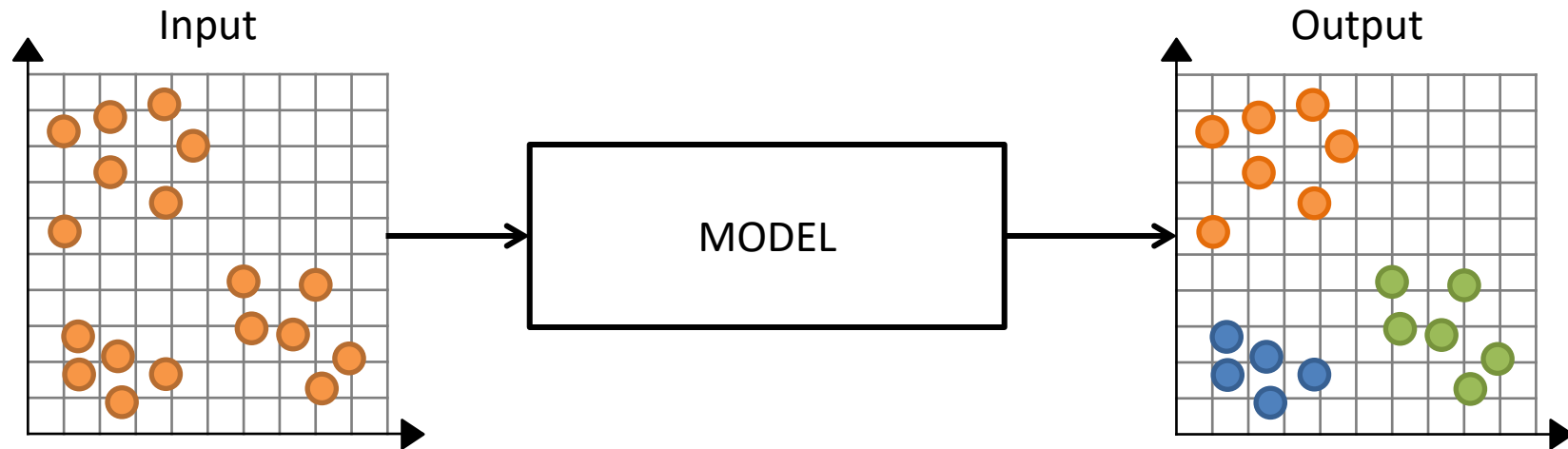


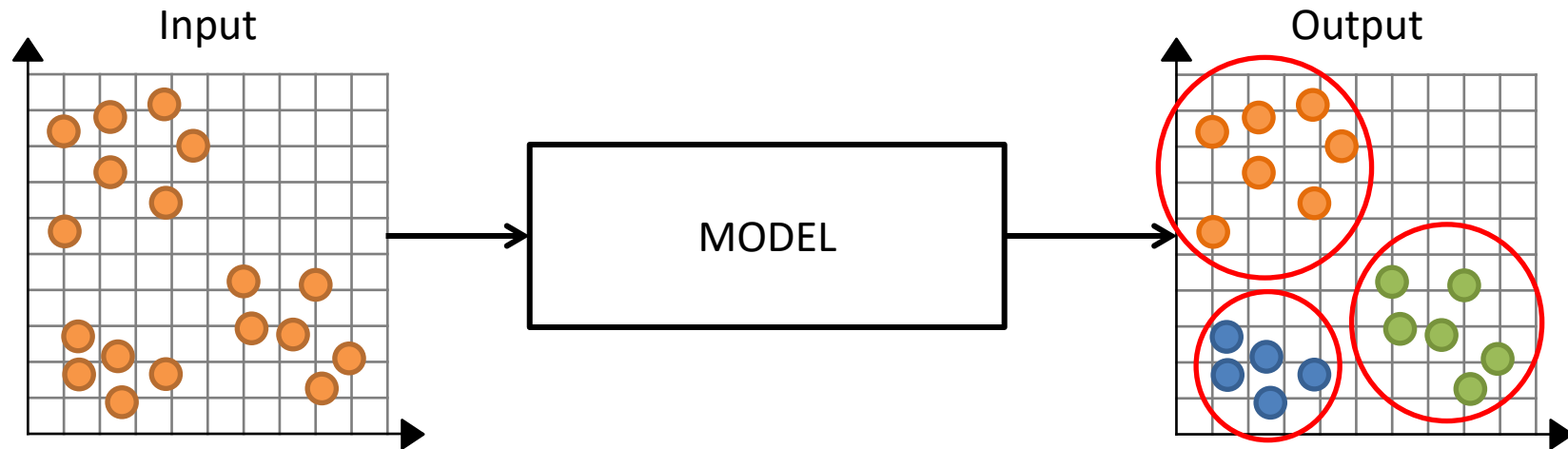
Example: Supervised learning



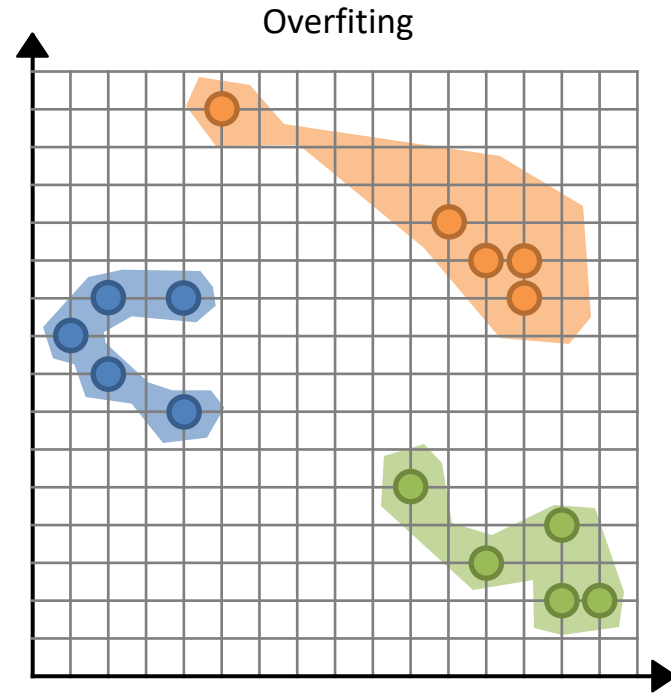
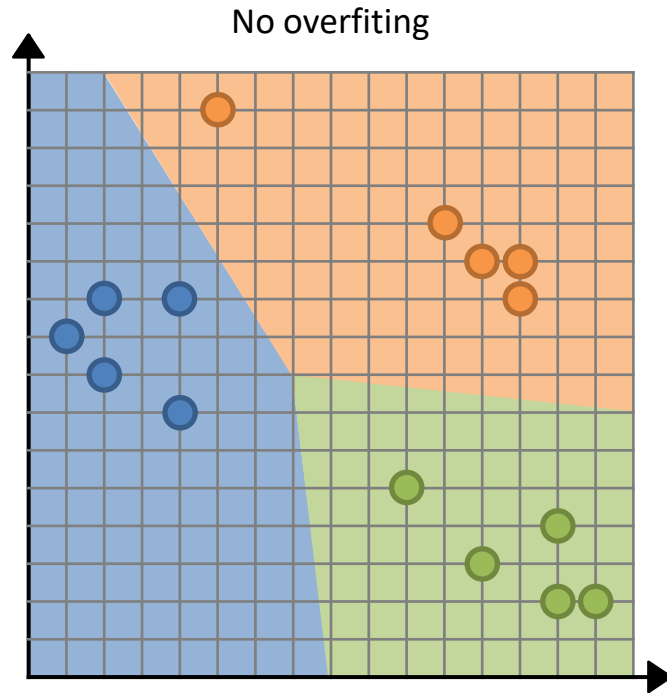
-  0: acer palmatum
-  1: aesculus chinensis
-  2: cercis chinensis

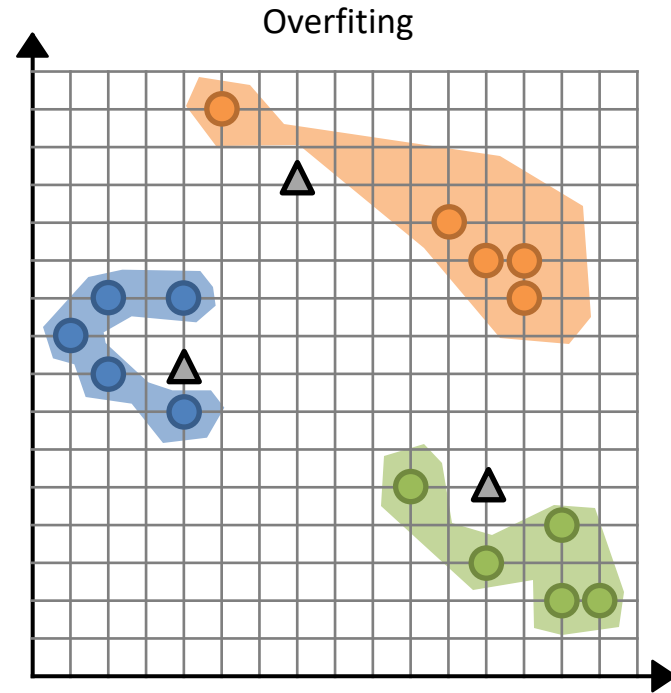
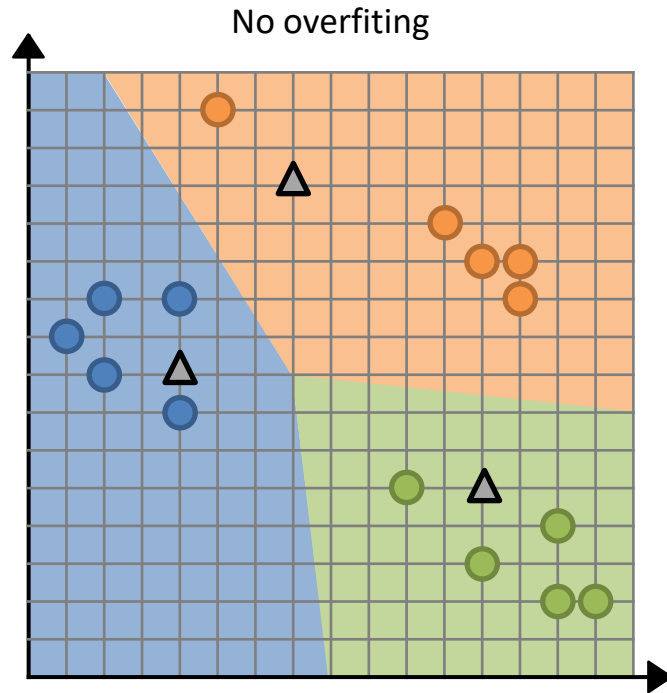


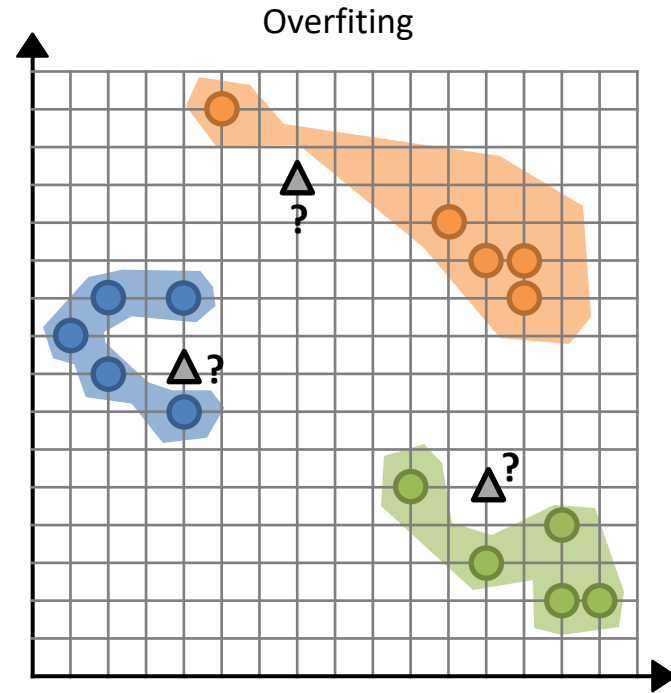
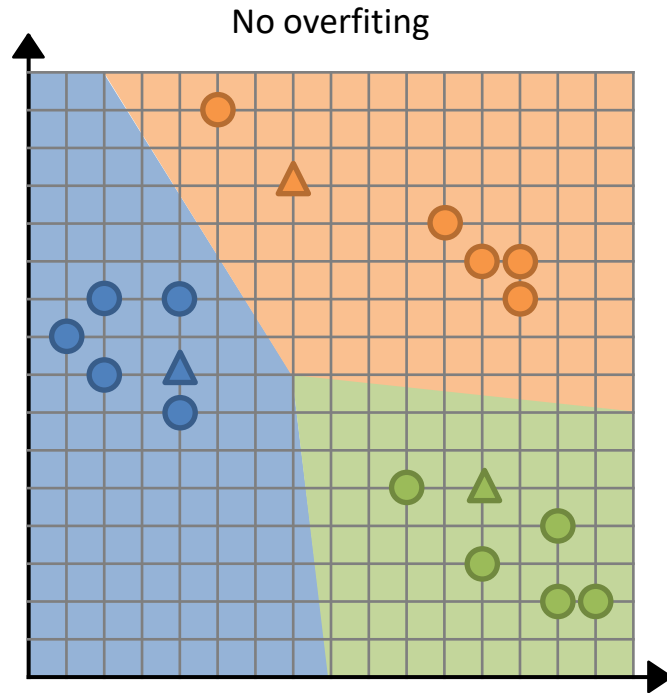


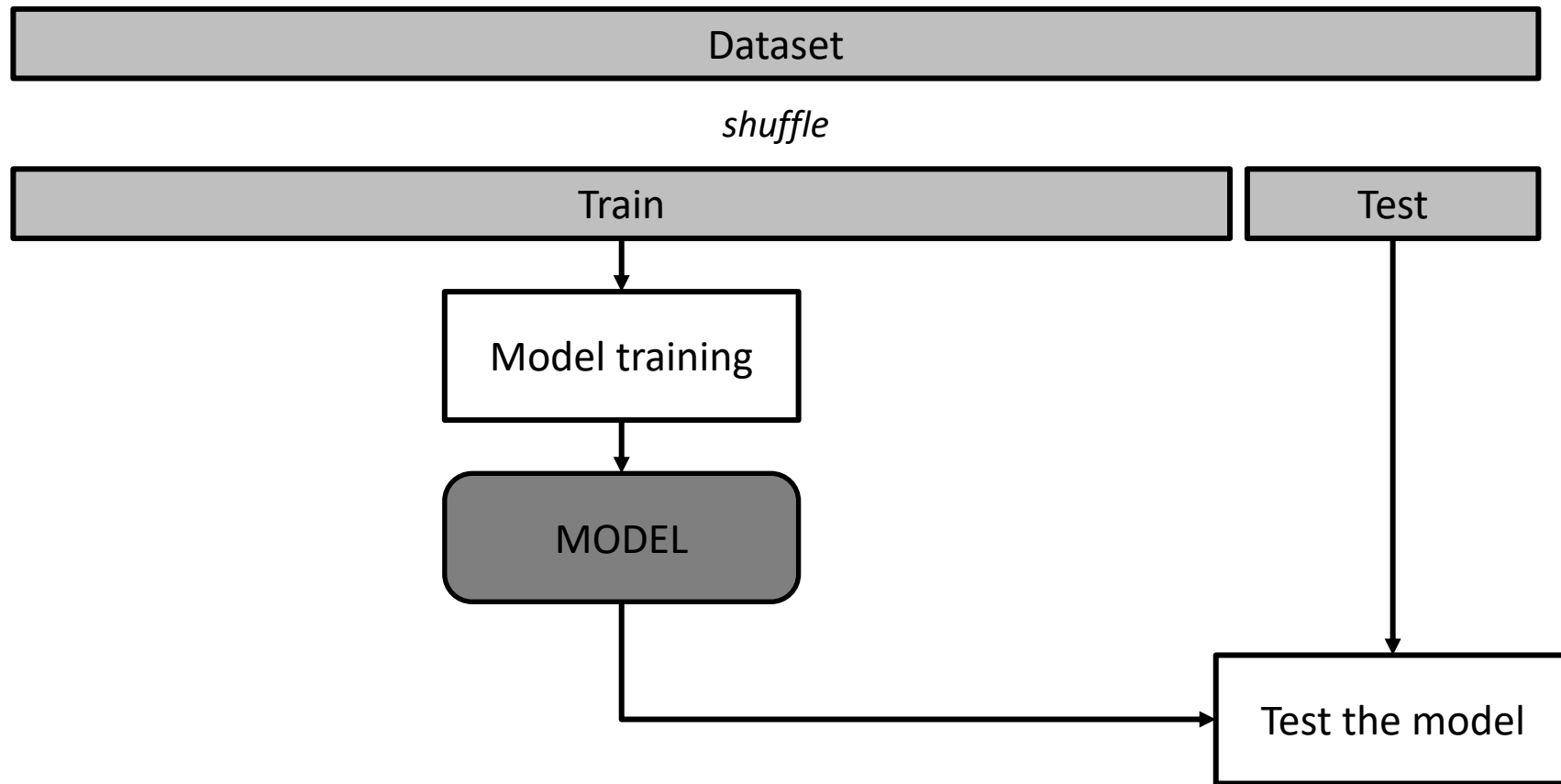


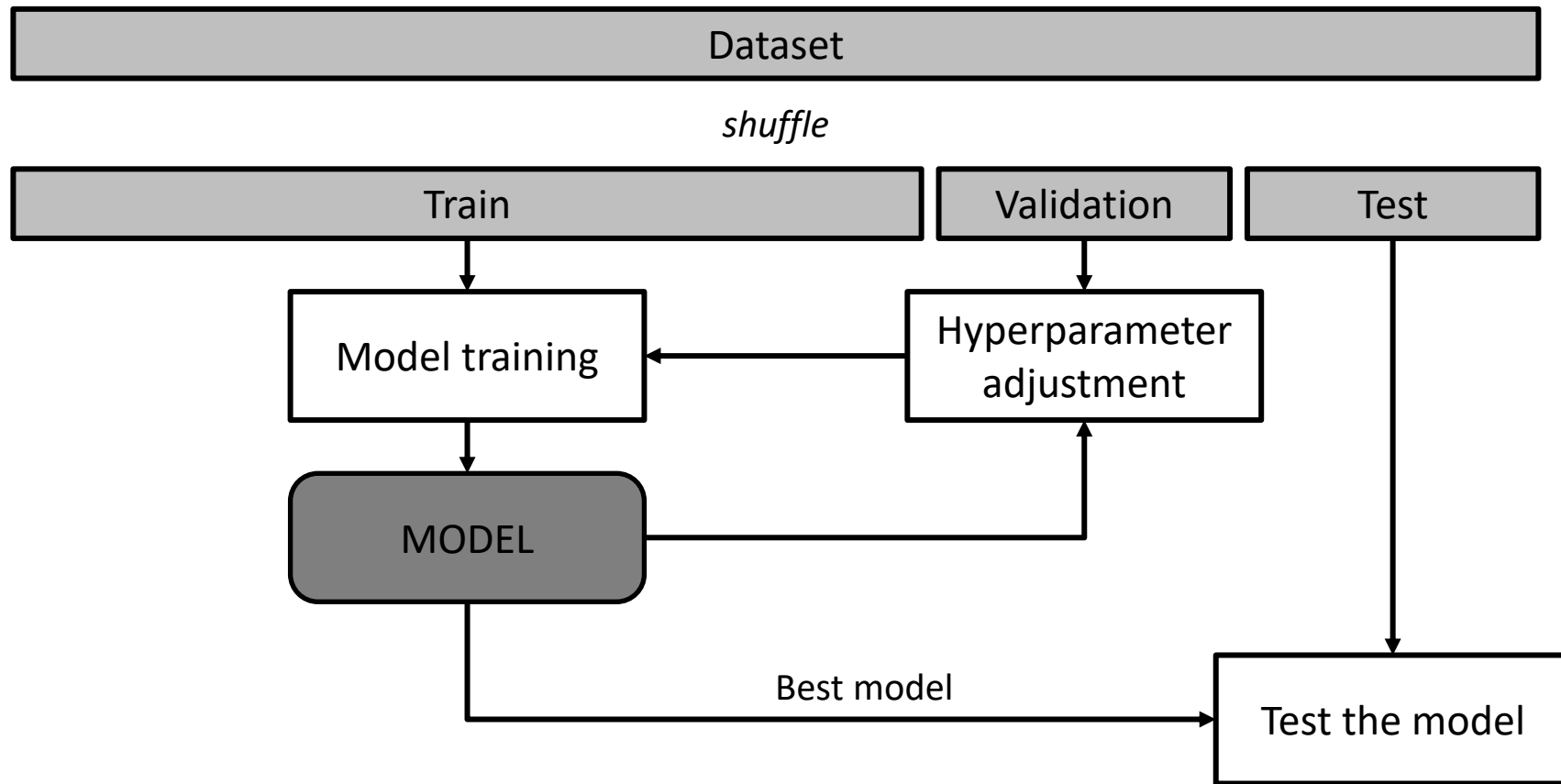
CROSS-VALIDATION











Hold-out cross-validation

Dataset

Image	excentricidade	área	classe
0	0,1	250	0
1	0,6	450	2
2	0,3	350	0
3	0,2	550	1
4	0,5	800	1
5	0,2	100	0
6	0,7	200	2
7	0,7	750	1
8	0,4	400	0
9	0,8	150	2
10	0,9	300	2
11	0,8	700	1
12	0,4	150	
13	0,4	300	
14	0,2	200	
15	0,7	250	

shuffle

Shuffled dataset

Image	excentricidade	área	classe
13	0,5	300	0
11	0,8	700	1
8	0,4	400	0
9	0,8	150	2
2	0,3	350	0
15	0,7	250	2
4	0,5	800	1
7	0,7	750	1
10	0,9	300	2
12	0,4	150	0
3	0,6	550	1
6	0,6	200	2
0	0,4	700	1
1	0,8	400	2
5	0,3	100	0
14	0,2	200	0

split

Train

Image	excentricidade	área	classe
13	0,5	300	0
11	0,8	700	1
8	0,4	400	0
9	0,8	150	2
2	0,3	350	0
15	0,7	250	2
4	0,5	800	1
7	0,7	750	1
10	0,9	300	2
12	0,4	150	0
3	0,6	550	1
6	0,6	200	2

Test

Image	excentricidade	área	classe
0	0,4	700	1
1	0,8	400	2
5	0,3	100	0
14	0,2	200	0

K-fold cross-validation



shuffle



NORMALIZATION

Normalization

- Normal Feature Transform (Standard Scaler)

Training (X_{train})

Imagem	excentricidade	área
13	0,50	300
11	0,80	700
8	0,40	400
9	0,80	150
2	0,30	350
15	0,70	250
4	0.50	800
7	0,70	750
10	0,90	300
12	0,40	150
3	0,60	550
6	0,60	200
Mean:	0.60	408.33
Std. Dev.:	0.1859	234.35

Normalized Training (X'_{train})

Imagem	excentricidade	área
13		
11		
8		
9		
2		
15		
4		
7		
10		
12		
3		
6		
Mean:		
Std. Dev.:		

Test (X_{test})

Imagem	excentricidade	área
0	0,4	700
1	0,8	400
5	0,3	100
14	0,2	200

Normalized test (X'_{test})

Imagem	excentricidade	área
0		
1		
5		
14		
Mean:		
Std. Dev.:		

Normalization

- Normal Feature Transform (Standard Scaler)

Training (X_{train})

Imagem	excentricidade	área
13	0,50	300
11	0,80	700
8	0,40	400
9	0,80	150
2	0,30	350
15	0,70	250
4	0.50	800
7	0,70	750
10	0,90	300
12	0,40	150
3	0,60	550
6	0,60	200
Mean:	0.60	408.33
Std. Dev.:	0.1859	234.35

Normalized Training (X'_{train})

Imagem	excentricidade	área
13		
11		
8		
9		
2		
15		
4		
7		
10		
12		
3		
6		
Mean:		
Std. Dev.:		

Test (X_{test})

Imagem	excentricidade	área
0	0,4	700
1	0,8	400
5	0,3	100
14	0,2	200

Normalized test (X'_{test})

Imagem	excentricidade	área
0		
1		
5		
14		
Mean:		
Std. Dev.:		

$$X'_{train} = \frac{X_{train} - \text{mean}(X_{train})}{\text{std}(X_{train})}$$

$$X'_{test} = \frac{X_{test} - \text{mean}(X_{train})}{\text{std}(X_{train})}$$

Normalization

- Normal Feature Transform (Standard Scaler)

Training (X_{train})

Imagem	excentricidade	área
13	0,50	300
11	0,80	700
8	0,40	400
9	0,80	150
2	0,30	350
15	0,70	250
4	0.50	800
7	0,70	750
10	0,90	300
12	0,40	150
3	0,60	550
6	0,60	200
Mean:	0.60	408.33
Std. Dev.:	0.1859	234.35

Normalized Training (X'_{train})

Imagem	excentricidade	área
13	-0.5380	-0.4622
11	1.0760	1.2445
8	-1.0760	-0.0355
9	1.0760	-1.1022
2	-1.6140	-0.2489
15	0.5380	-0.6756
4	-0.5380	1.6712
7	0.5380	1.4578
10	1.6140	-0.4622
12	-1.0760	-1.1022
3	0.0000	0.6044
6	0.0000	-0.8889
Mean:	0.00	1.00
Std. Dev.:	0.00	1.00

Test (X_{test})

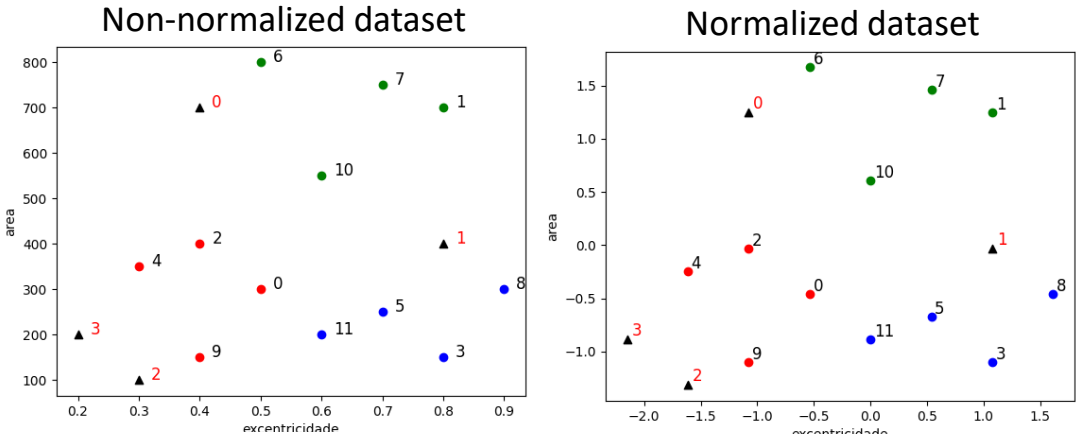
Imagem	excentricidade	área
0	0,4	700
1	0,8	400
5	0,3	100
14	0,2	200

Normalized test (X'_{test})

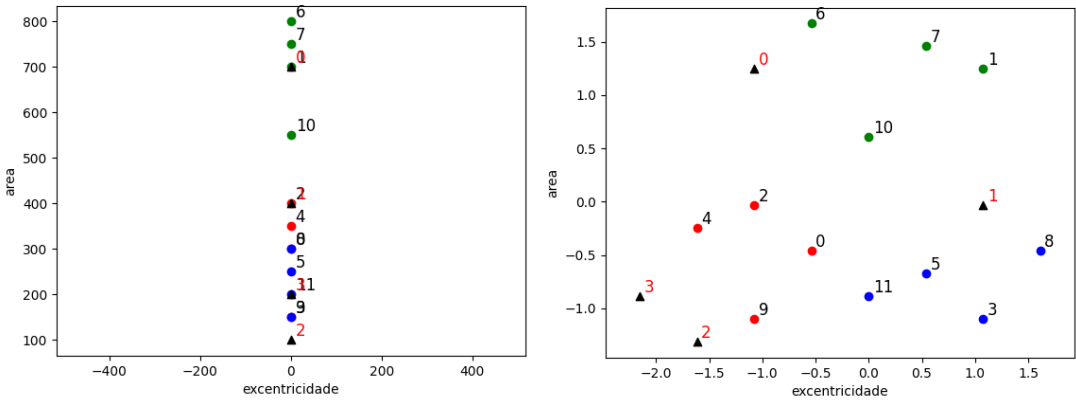
Imagem	excentricidade	área
0	-1.0760	1.244528
1	1.0760	-0.035558
5	-1.6140	-1.315644
14	-2.1521	-0.888949
Mean:	-0.9415	-0.2489
Std. Dev.:	1.4150	1.1289

$$X'_{train} = \frac{X_{train} - \text{mean}(X_{train})}{\text{std}(X_{train})} \quad X'_{test} = \frac{X_{test} - \text{mean}(X_{train})}{\text{std}(X_{train})}$$

Adjusted aspect ratio



Real aspect ratio



CLASSIFICATION EVALUATION

Confusion matrix

- **True positive (TP):**
 - Objects of class C1 classified as C1.
- **True negative (TN):**
 - Objects of other classes (C2 and C3) classified as not being C1.
- **False positive (FP) (type I error):**
 - Objects classified as C1 but belonging to other classes (C2 or C3).
- **False negative (FN) (type II error):**
 - Objects of class C1 classified as other classes (C2 or C3).

		Classification			
		Class C1	Class C2	Class C3	Sum
Real class	Class C1	5	3	0	8
	Class C2	2	3	1	6
	Class C3	0	2	11	13
	Soma	7	8	12	

Confusion matrix

- **True positive (TP):**
 - Objects of class C1 classified as C1.
- **True negative (TN):**
 - Objects of other classes (C2 and C3) classified as not being C1.
- **False positive (FP) (type I error):**
 - Objects classified as C1 but belonging to other classes (C2 or C3).
- **False negative (FN) (type II error):**
 - Objects of class C1 classified as other classes (C2 or C3).

Class C1		Classification			
		Class C1		Others	
Real class	Class C1	5	TP	3	FN
	Others	2	FP	17	TN

		Classification			
		Class C1	Class C2	Class C3	Sum
Real class	Class C1	5	3	0	8
	Class C2	2	3	1	6
	Class C3	0	2	11	13
	Soma	7	8	12	

Confusion matrix

- **True positive (TP):**
 - Objects of class C1 classified as C1.
- **True negative (TN):**
 - Objects of other classes (C2 and C3) classified as not being C1.
- **False positive (FP) (type I error):**
 - Objects classified as C1 but belonging to other classes (C2 or C3).
- **False negative (FN) (type II error):**
 - Objects of class C1 classified as other classes (C2 or C3).

Class C1		Classification	
		Class C1	Others
Real class	Class C1	5 TP	3 FN
	Others	2 FP	17 TN

		Classification			Sum
		Class C1	Class C2	Class C3	
Real class	Class C1	5	3	0	8
	Class C2	2	3	1	6
	Class C3	0	2	11	13
	Soma	7	8	12	

Confusion matrix

- **True positive (TP):**
 - Objects of class C1 classified as C1.
- **True negative (TN):**
 - Objects of other classes (C2 and C3) classified as not being C1.
- **False positive (FP) (type I error):**
 - Objects classified as C1 but belonging to other classes (C2 or C3).
- **False negative (FN) (type II error):**
 - Objects of class C1 classified as other classes (C2 or C3).

		Classification			
		Class C1	Class C2	Class C3	Sum
Real class	Class C1	5	3	0	8
	Class C2	2	3	1	6
	Class C3	0	2	11	13
	Soma	7	8	12	

Class C1		Classification			
		Class C1		Others	
Real class	Class C1	5	TP	3	FN
	Others	2	FP	17	TN

Class C2		Classification			
		Class C2		Others	
Real class	Class C2	3	TP	3	FN
	Others	5	FP	16	TN

Confusion matrix

- **True positive (TP):**
 - Objects of class C1 classified as C1.
- **True negative (TN):**
 - Objects of other classes (C2 and C3) classified as not being C1.
- **False positive (FP) (type I error):**
 - Objects classified as C1 but belonging to other classes (C2 or C3).
- **False negative (FN) (type II error):**
 - Objects of class C1 classified as other classes (C2 or C3).

		Classification			
		Class C1	Class C2	Class C3	Sum
Real class	Class C1	5	3	0	8
	Class C2	2	3	1	6
	Class C3	0	2	11	13
	Soma	7	8	12	

Class C1		Classification			
		Class C1		Others	
Real class	Class C1	5	TP	3	FN
	Others	2	FP	17	TN

Class C2		Classification			
		Class C2		Others	
Real class	Class C2	3	TP	3	FN
	Others	5	FP	16	TN

Class C3		Classification			
		Classe C3		Others	
Real class	Class C3	11	TP	2	FN
	Others	1	FP	13	TN

Accuracy, precision, recall, and F1-score

- Accuracy:

- $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

- Precision:

- $Precision = \frac{TP}{TP+FP}$

- Recall:

- $Recall = \frac{TP}{TP+FN}$

- F1-score:

- $F1 - score = \frac{2 \times TP}{2 \times TP + FP + FN}$

- Support:

- $Support = TP + FN$

Accuracy, precision, recall, and F1-score

- Accuracy:
 - *How close the classification is to the true value.*
 - $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
- Precision:
 - *The ability of the classifier not to label a negative sample as positive.*
 - $Precision = \frac{TP}{TP+FP}$
- Recall:
 - *The ability of the classifier to find all the positive samples.*
 - $Recall = \frac{TP}{TP+FN}$
- F1-score:
 - *The weighted harmonic mean of the precision and recall.*
 - $F1 - score = \frac{2 \times TP}{2 \times TP + FP + FN}$
- Support:
 - *The number of occurrences of each real (true) class.*
 - $Support = TP + FN$

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

Accuracy, precision, recall, and F1-score

- Accuracy:

$$- \text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

- Precision:

$$- \text{Precision} = \frac{TP}{TP+FP}$$

- Recall:

$$- \text{Recall} = \frac{TP}{TP+FN}$$

- F1-score:

$$- \text{F1-score} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

- Support:

$$- \text{Support} = TP + FN$$

Classes	TP	TN	FP	FN	Accuracy	Precision	Recall	F1-Score	Support
C1	5	17	2	3					
C2	3	16	5	3					
C3	11	13	1	2					
MEAN									
STD. DEV.									

Accuracy, precision, recall, and F1-score

- Accuracy:

$$- \text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

- Precision:

$$- \text{Precision} = \frac{TP}{TP+FP}$$

- Recall:

$$- \text{Recall} = \frac{TP}{TP+FN}$$

- F1-score:

$$- \text{F1-score} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

- Support:

$$- \text{Support} = TP + FN$$

Classes	TP	TN	FP	FN	Accuracy	Precision	Recall	F1-Score	Support
C1	5	17	2	3	0.8148	0.7143	0.6250	0.6667	8
C2	3	16	5	3	0.7037	0.3750	0.5000	0.4286	6
C3	11	13	1	2	0.8889	0.9167	0.8462	0.8800	13
MEAN									
STD. DEV.									

Accuracy, precision, recall, and F1-score

- Accuracy:

$$- \text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

- Precision:

$$- \text{Precision} = \frac{TP}{TP+FP}$$

- Recall:

$$- \text{Recall} = \frac{TP}{TP+FN}$$

- F1-score:

$$- \text{F1-score} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

- Support:

$$- \text{Support} = TP + FN$$

Classes	TP	TN	FP	FN	Accuracy	Precision	Recall	F1-Score	Support
C1	5	17	2	3	0.8148	0.7143	0.6250	0.6667	8
C2	3	16	5	3	0.7037	0.3750	0.5000	0.4286	6
C3	11	13	1	2	0.8889	0.9167	0.8462	0.8800	13
MEAN					0.8025	0.6687	0.6571	0.6584	
STD. DEV.					0.0761	0.2235	0.1431	0.1844	

Accuracy Vs. Precision

- *Accuracy:*

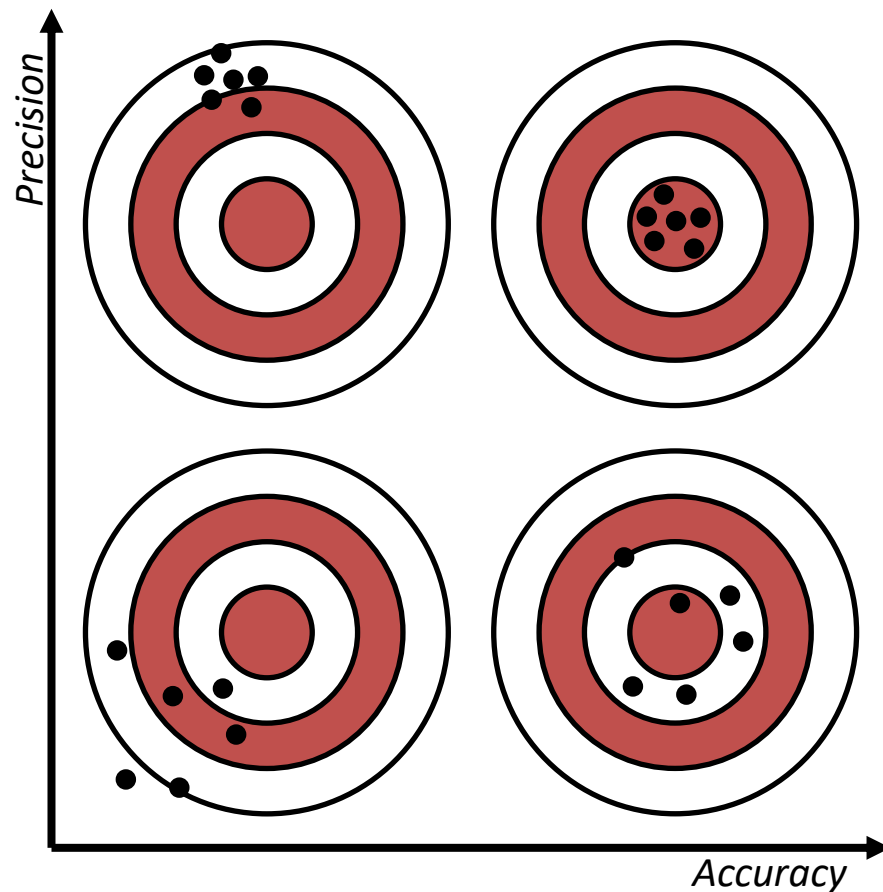
- $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

- *How close the classification is to the true value.*

- *Precision:*

- $Precision = \frac{TP}{TP+FP}$

- *The variation between different predictions.*



- GONZALEZ, R.C.; WOODS, R.E.; **Processamento Digital de Imagens**. 3ª edição. Editora Pearson, 2009.
- COSTA, L. DA F.; CESAR-JR., R. M. **Shape analysis and classification: theory and practice**. CRC Press, 2000. Chapter 8.
- Yann LeCun', Alfredo Canziani. **Yann LeCun's Deep Learning Course at CDS - SPRING 2021**
 - <https://cds.nyu.edu/deep-learning/>
- **scikit-learn – User Guide**.
 - https://scikit-learn.org/stable/user_guide.html

THE END