Master in Computer Vision *Barcelona*

Module: M3. Machine Learning for Computer Vision

Lecture: Experimental Setup

Lecturer: **Ramon Baldrich**
ramon.baldrich@uab.cat

Machine Learning

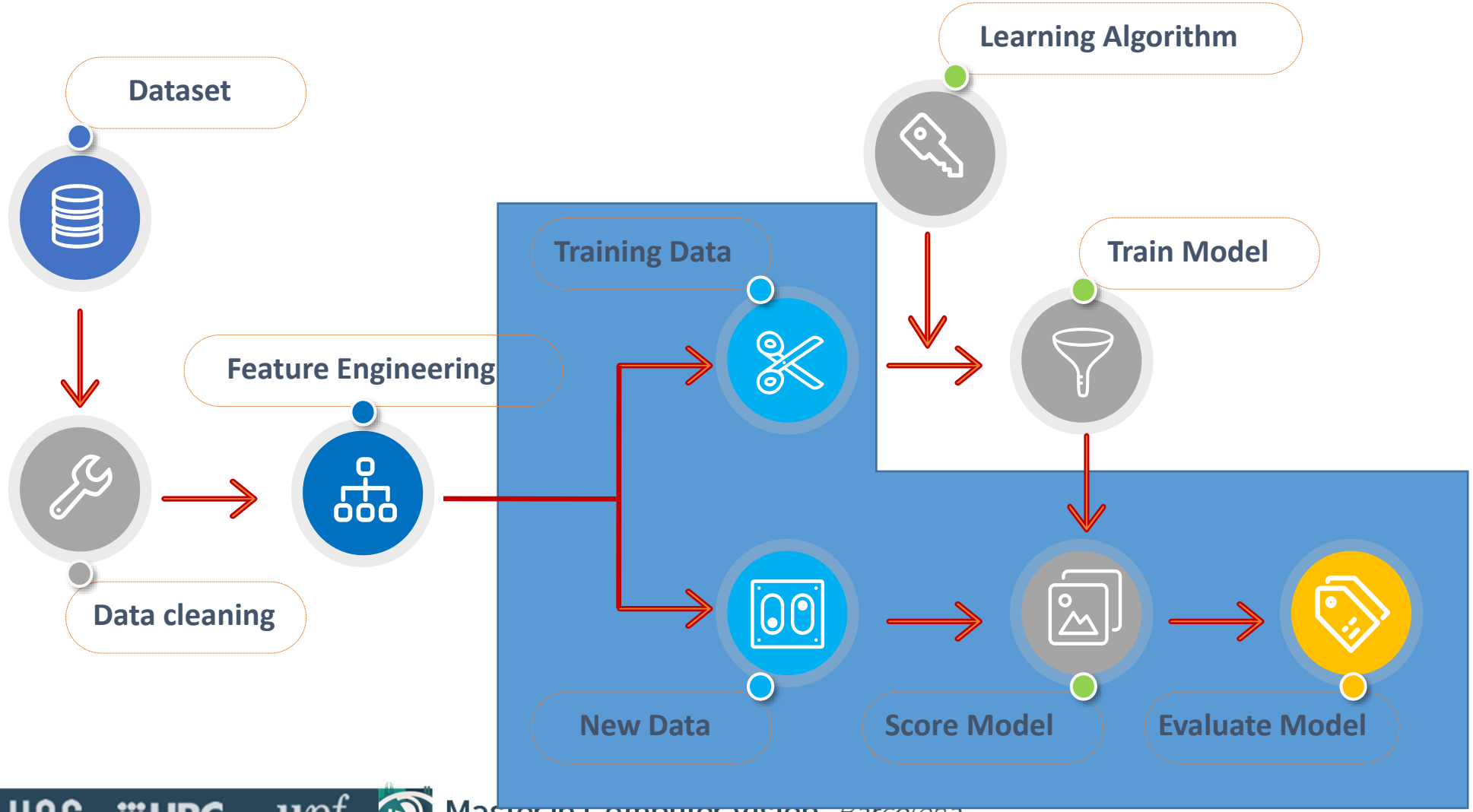
Introduction to model evaluation

Ramon Baldrich

Universitat Autònoma de Barcelona

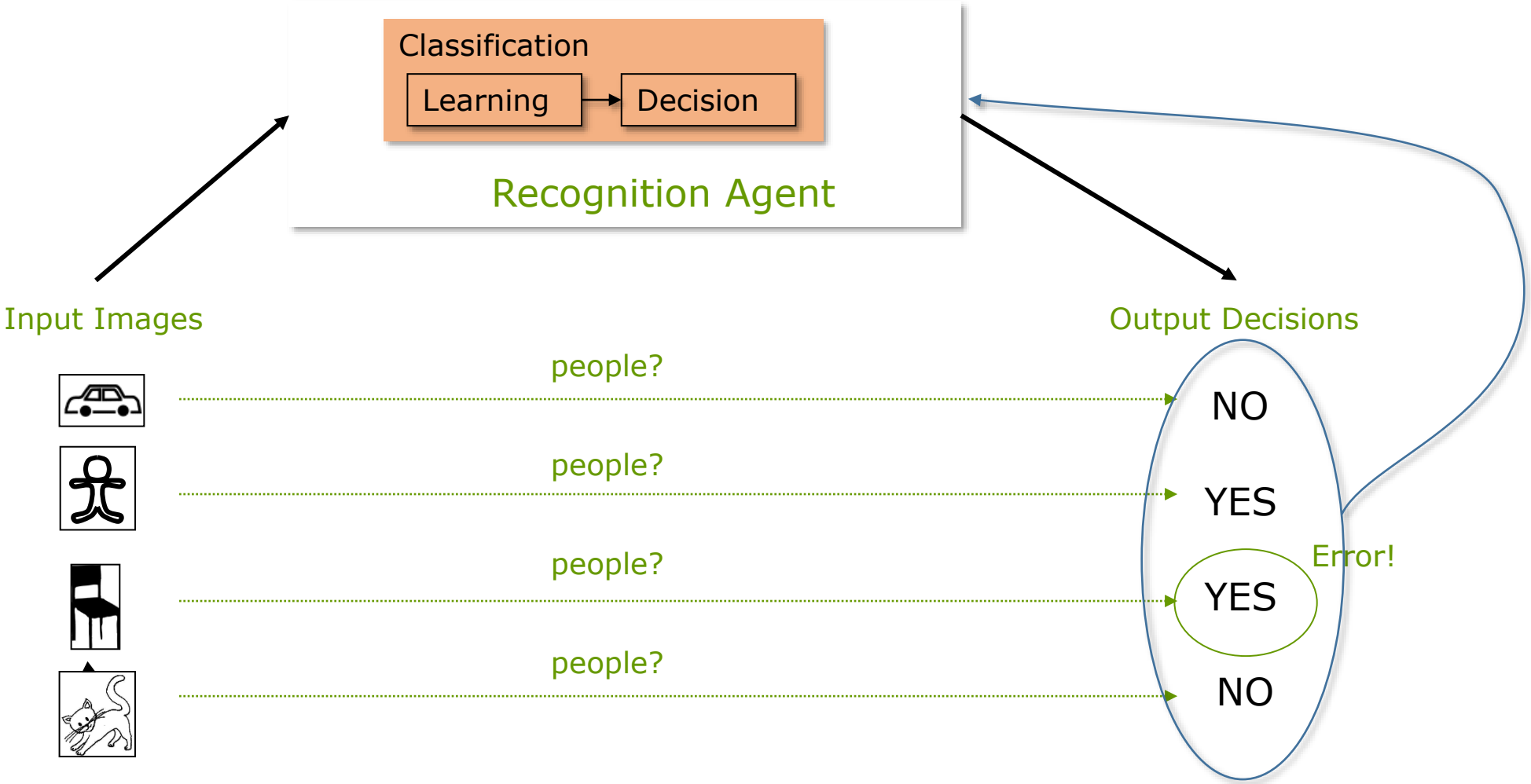
Credit to: Maria Vanrell, Fernando Vilariño

Learning process from data



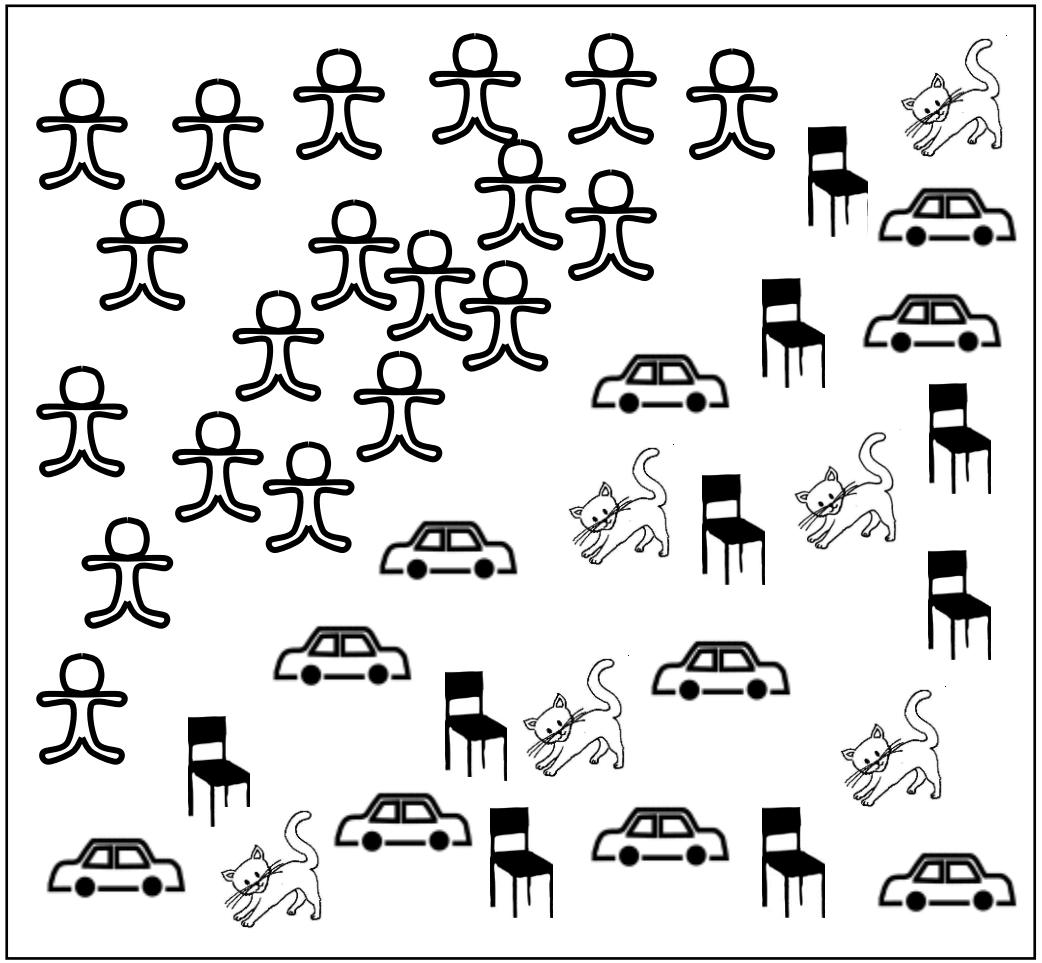
Goal: How to evaluate the performance of a recognition agent?

Goal: How to evaluate the performance of our recognition agent?

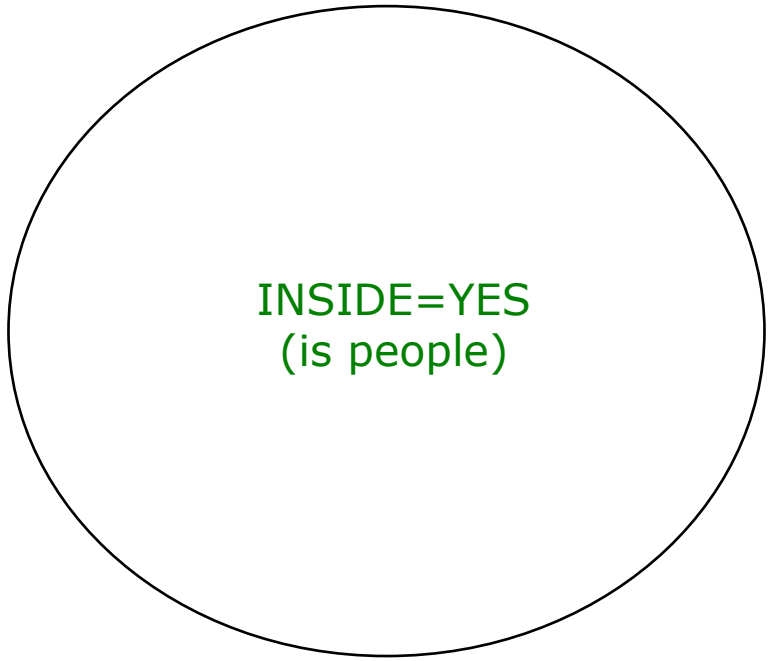


Example:

Given an image dataset:

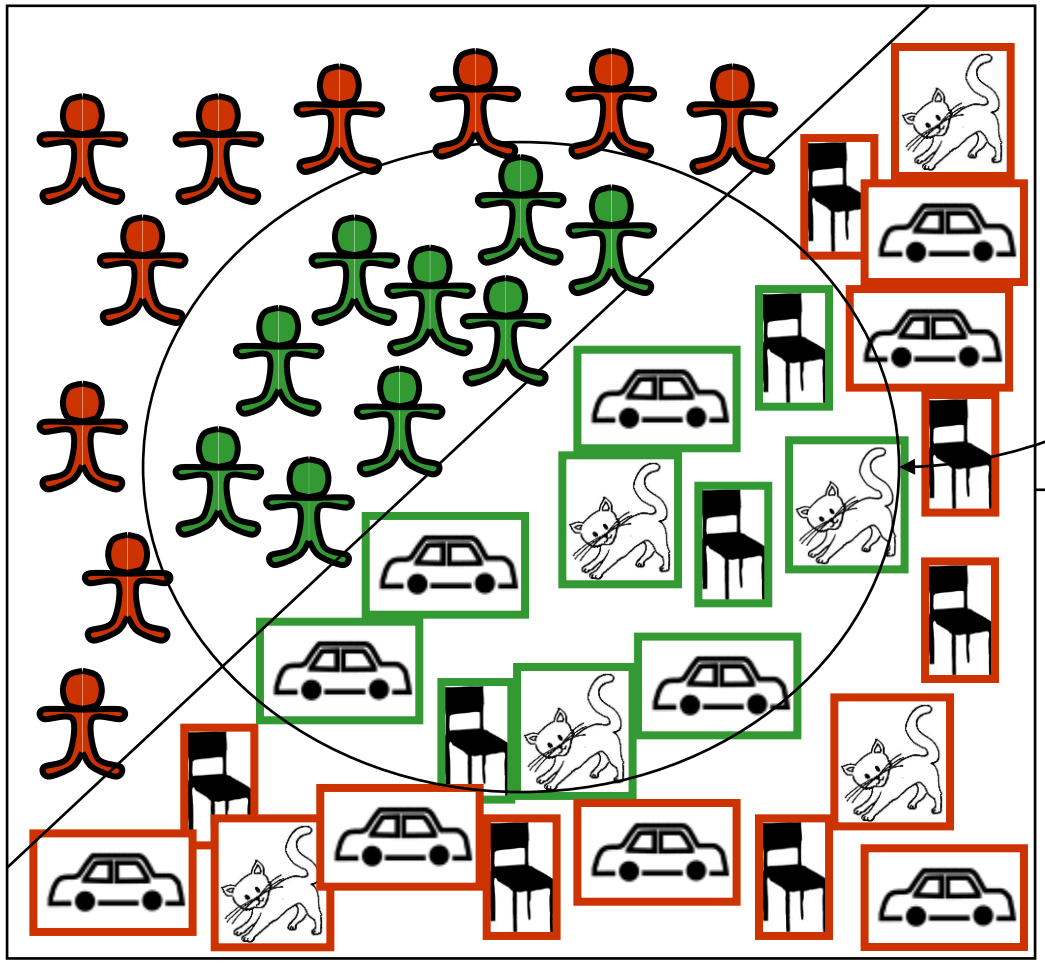


The task of our **people recognizer** is given by:



Example:

Given an image dataset:



Our recognizer:

Is it a good result?

Ground Truth

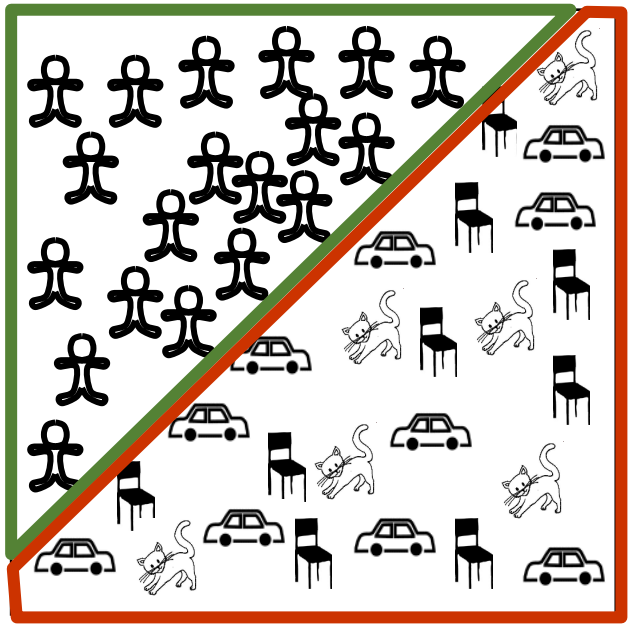
Ground truth is a term used in cartography, meteorology, analysis of aerial photographs, satellite imagery and a range of other remote sensing techniques in which data are gathered at a distance.

Ground truth may refer to a process in which a pixel on a satellite image is compared to what is there in reality (at the present time) in order to verify the contents of the pixel on the image

(from Wikipedia)

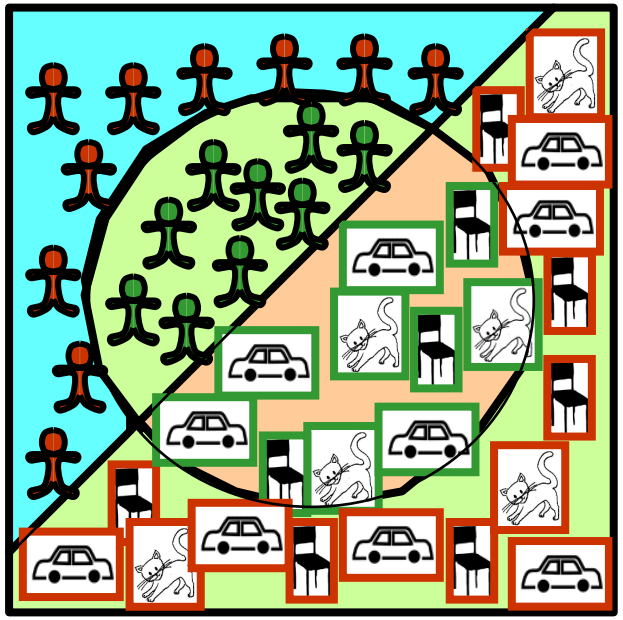
Confusion matrix (or Contingency table)

RECOGNIZER OUTPUT / CORRECT RECOGNITION	PEOPLE	NON-PEOPLE
PEOPLE	True Positive	False Negative
NON-PEOPLE	False Positive	True Negative



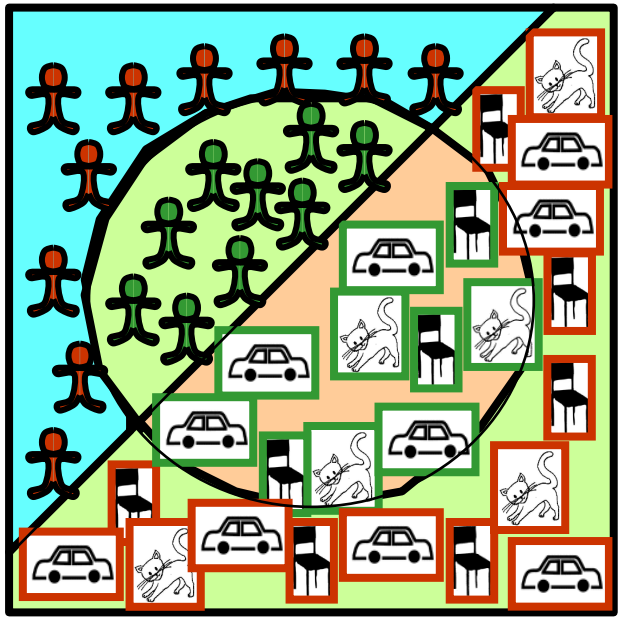
Confusion matrix (or Contingency table)

RECOGNIZER OUTPUT / CORRECT RECOGNITION	PEOPLE	NON-PEOPLE
PEOPLE	True Positive	False Negative
NON-PEOPLE	False Positive	True Negative



Exercise: Build the Confusion matrix of this recognizer

CORRECT RECOGNITION \ RECOGNIZER OUTPUT	PEOPLE	NON-PEOPLE
PEOPLE	True Positive	False Negative
NON-PEOPLE	False Positive	True Negative



CORRECT RECOGNITION \ RECOGNIZER OUTPUT	PEOPLE	NON-PEOPLE
PEOPLE	9	10
NON-PEOPLE	10	35

Studying the confusion matrix:

RECOGNIZER OUTPUT / CORRECT RECOGNITION	PEOPLE	NON-PEOPLE
PEOPLE	True Positive	False Negative
NON-PEOPLE	False Positive	True Negative

- **Accuracy:** Goodness of the recogniser in **true results.**

$$Accuracy = \frac{TruePositive + TrueNegative}{All}$$

- **Precision:** Quality of the recogniser of **true people** in the result

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

- **Sensitivity (Recall):** Efficiency in recognizing **all true people**

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative}$$

- **Specificity (1-Fall·out):** Efficiency in leaving out **all non-people**

$$Specificity = \frac{TrueNegative}{FalsePositive + TrueNegative}$$

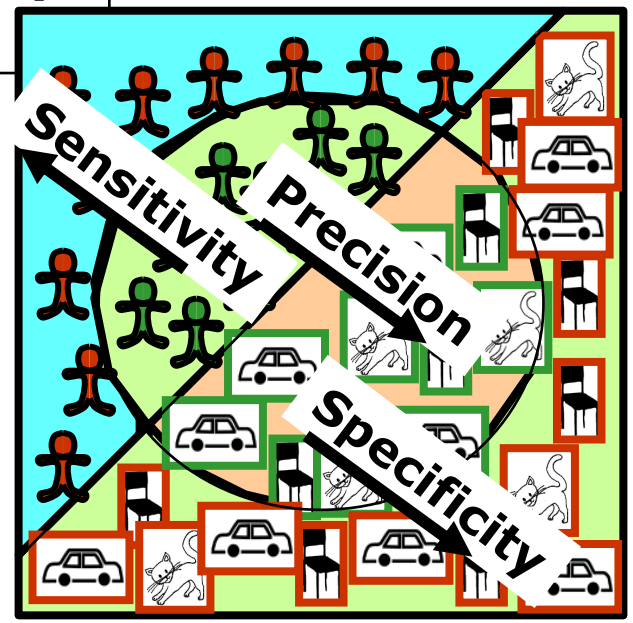
Fall·out: False Positive Rate

(Rate in including **all non-people**)

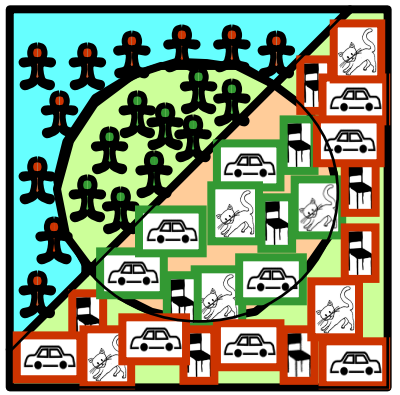
https://en.wikipedia.org/wiki/Confusion_matrix

Measures on the confusion matrix:

RECOGNIZER OUTPUT / CORRECT RECOGNITION	PEOPLE	NON-PEOPLE	
PEOPLE	True Positive	False Negative	**Sensitivity (Recall)**
NON-PEOPLE	False Positive	True Negative	**Specificity (1-Fall·out)**
	Precision		**Accuracy**



Understanding measures



		OUTPUT CLASSIFICATION	
		TRUE	FALSE
CORRECT CLASSIFICATION	POSITIVE	True Positive	False Negative
	NEGATIVE	False Positive	True Negative

Sensitivity (recall)=100%

All people is classified as people

No idea about how non-people has been classified

Specificity =100%

All non-people is classified as non-people

No idea about how people has been classified

Precision=100%

All classification is formed by people.

No idea about how many people has been left

Accuracy=100%

Perfect classification

Fall·out =100%

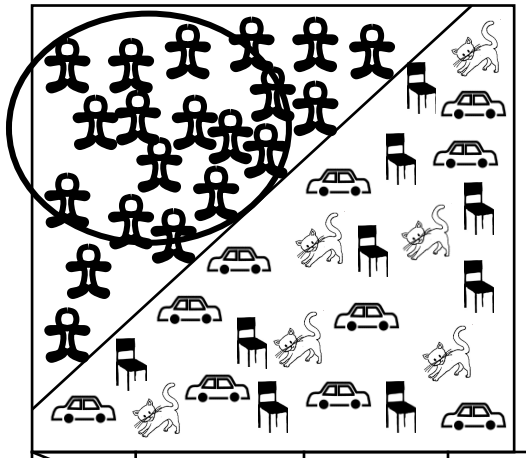
All non-people is classified as people

Assume you have a "positive" class called 1 and a "negative" class called 0.
P is your estimate of the true class label *Y*.

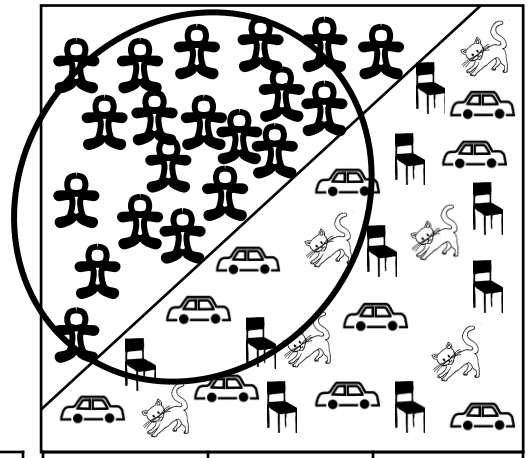
Then:

Precision	$P(Y=1 \mid P=1)$
Recall=Sensitivity	$P(P=1 \mid Y=1)$
Specificity	$P(P=0 \mid Y=0)$

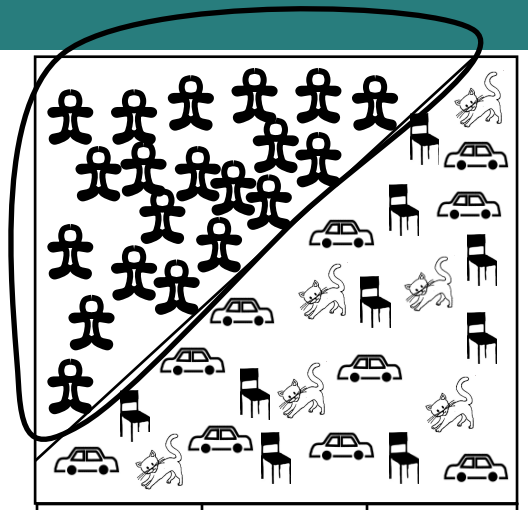
Exercise 1: Describe the following classifiers in terms of their confusion matrices
[#People=20 , #Non-People=25]



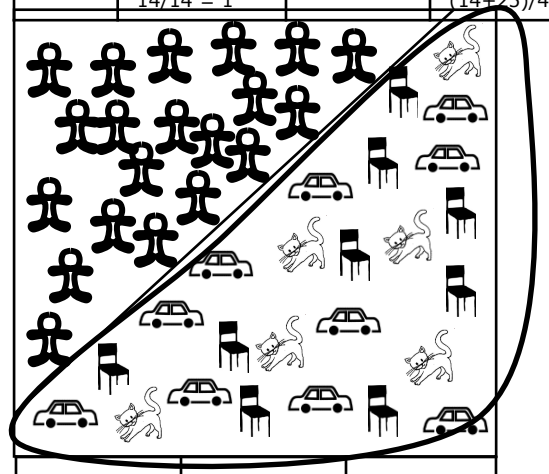
	PEOPLE	NON-PEOPLE	
Positive	TP: 14	FN: 6	Sensitivity: 14/20 = 0,7
Negative	FP: 0	TN: 25	Specificity: 25/25=1
	Precision: 14/14 = 1		Accuracy: (14+25)/45= 0,86



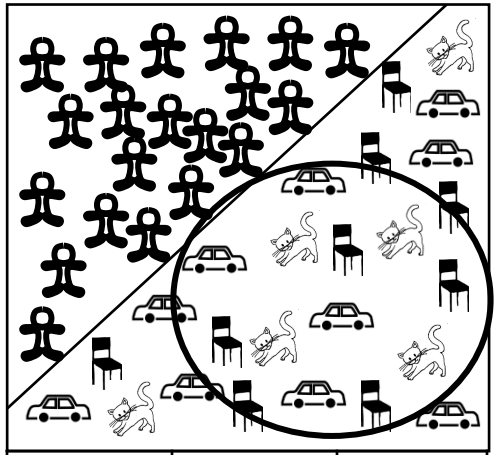
TP:	FN:	
FP:	TN:	



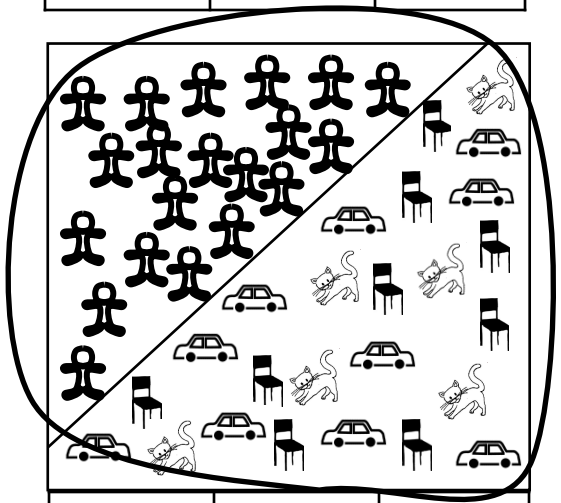
TP:	FN:	
FN:	TN:	



TP:	FP:	
FP:	TN:	

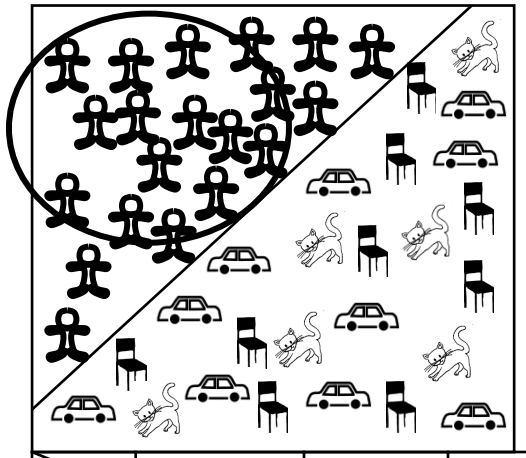


TP:	FN:	
FP:	TN:	

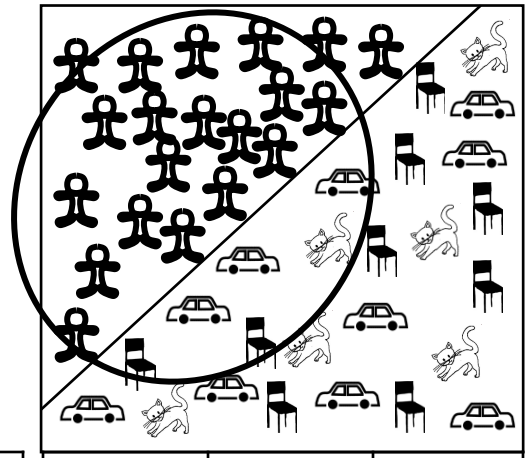


TP:	FN:	
FP:	TN:	

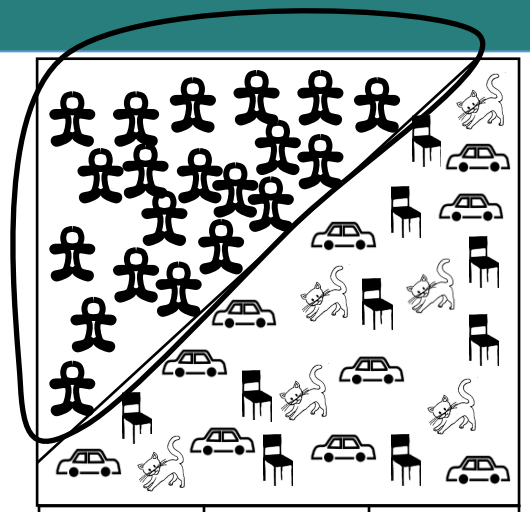
Exercise 1: Describe the following classifiers in terms of their confusion matrices
[#People=20 , #Non-People=25]



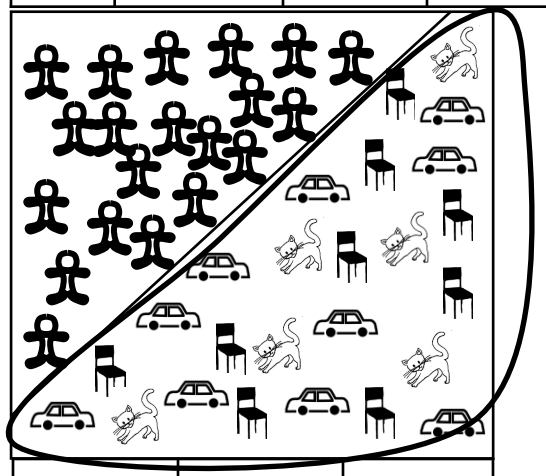
	PEOPLE	NON-PEOPLE	
Positive	TP: 14	FP: 0	Precision: 14/14 = 1
Negative	FN: 6	TN: 25	
	Sensitivity: 14/20 = 0,7	Specificity: 25/25=1	Accuracy: (14+25)/45= 0,86



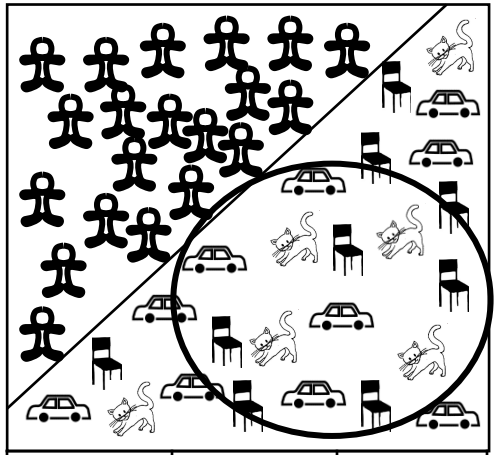
TP: 17	FP: 3	0, 85
FN: 6	TN: 19	0,76
0,74		0,80



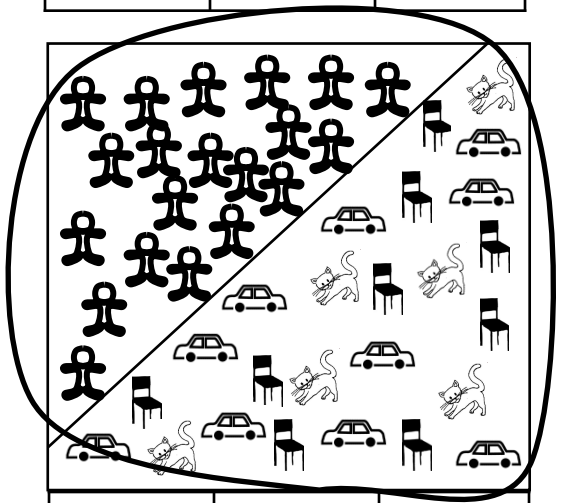
TP: 20	FP: 0	1
FN: 0	TN: 25	1
1		1



TP: 0	FP: 20	0
FN: 25	TN: 0	0
0		0

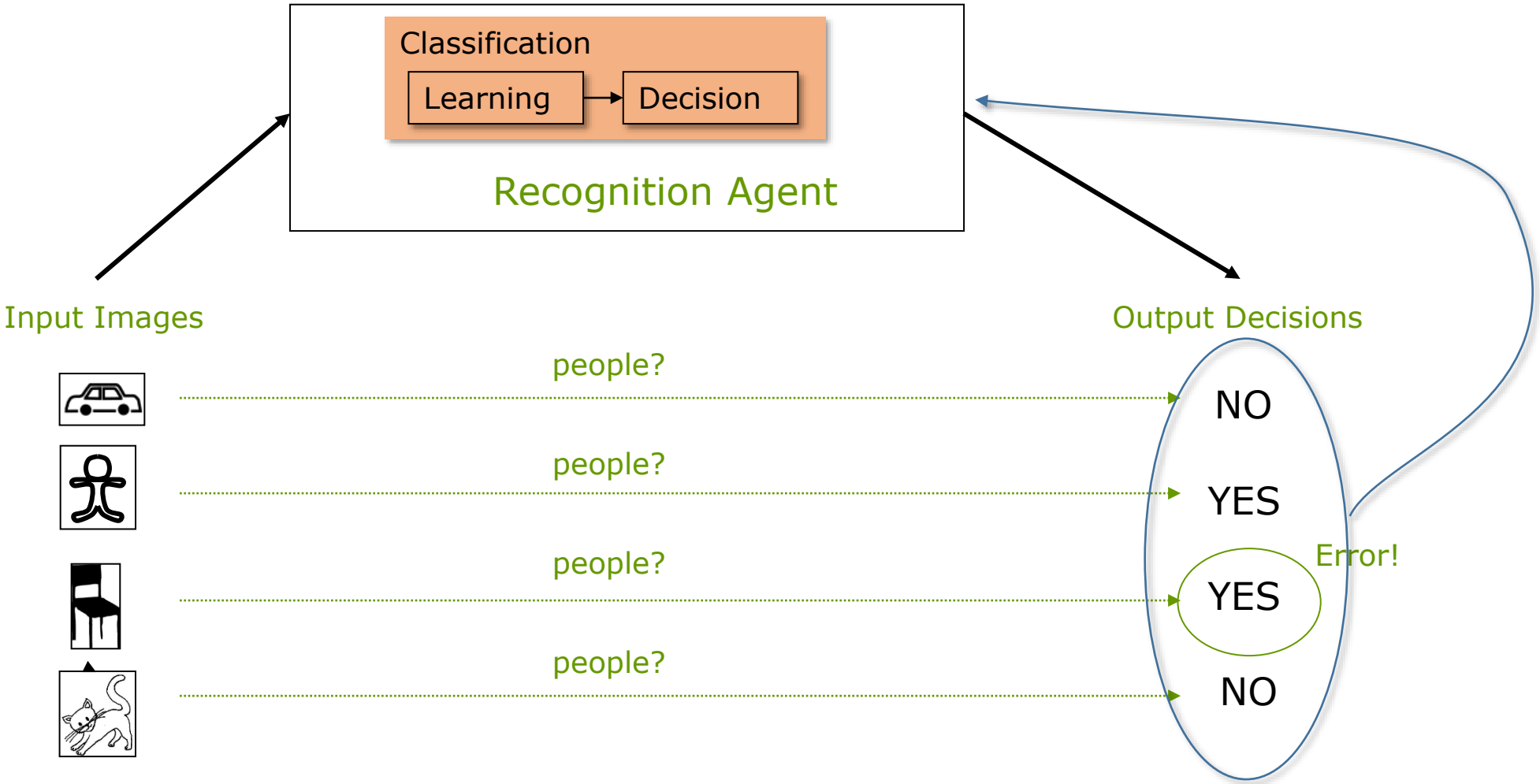


TP: 0	FP: 20	0
FN: 13	TN: 12	0,48
0		0,27

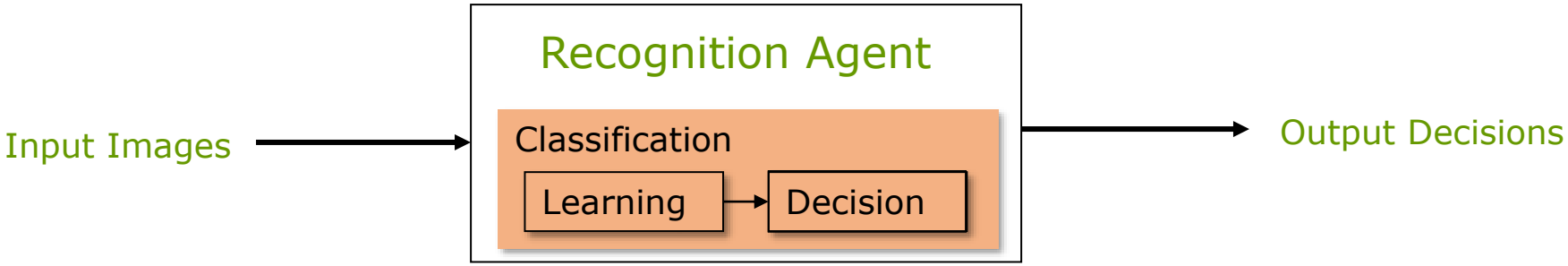


TP: 20	FP: 0	1
FN: 25	TN: 0	0
0,44		0,44

Goal: How to evaluate the performance of our recognition agent?



Question: How to use the peformance measures to tune our Decision Module?



Decision Module, usually needs a threshold value to achieve a final decision,

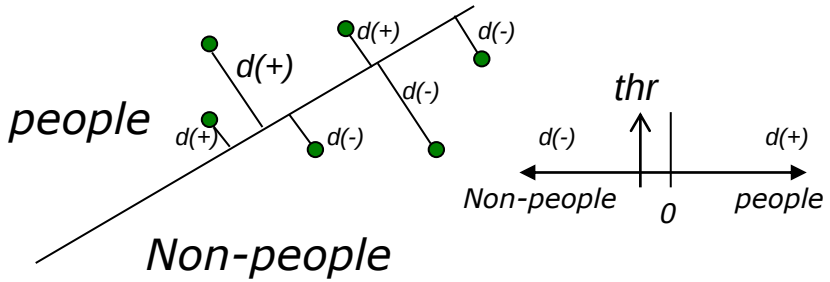
Generative learning:

$P(people/I)$
$P(car/I)$
$P(cat/I)$
$P(chair/I)$

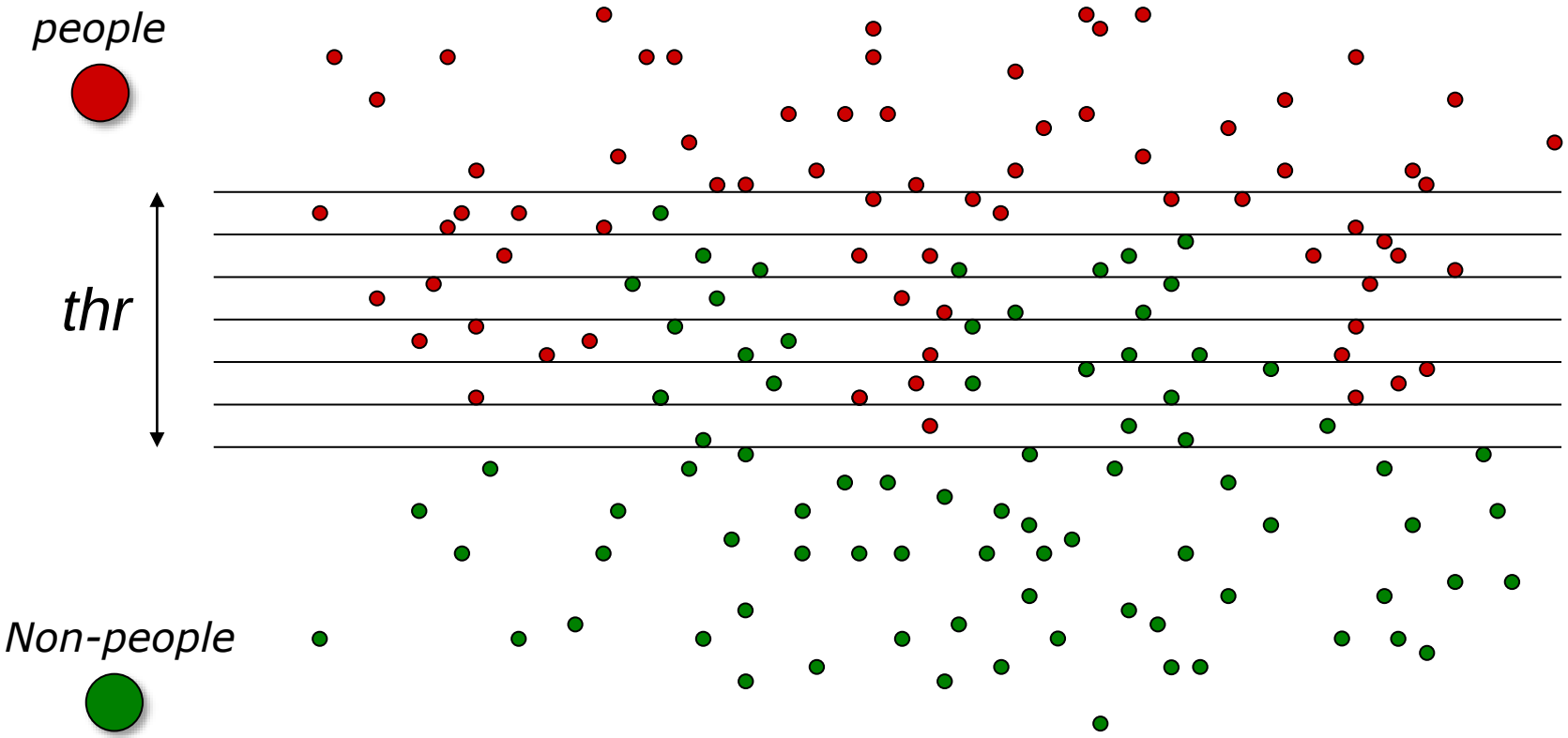
$p(people/I) > thr$

$\sum p(class/I) = 1$

Discriminative learning:



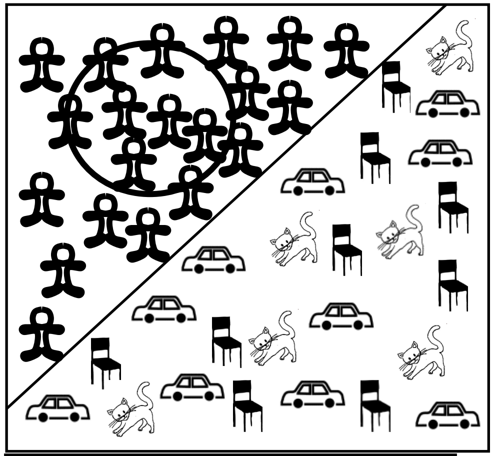
Different thresholds



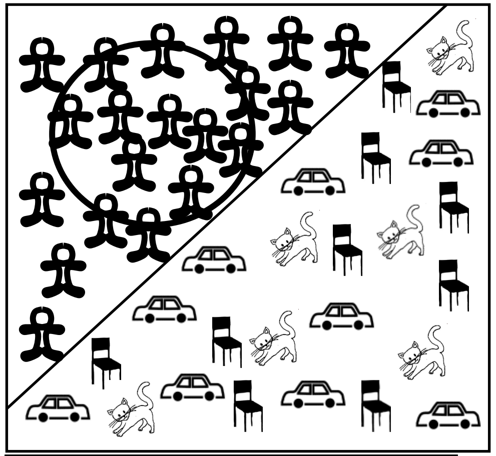
Different thresholds → Different confusion matrices

Accuracy
Precision
Sensitivity (Recall)
Specificity (Fall-out)

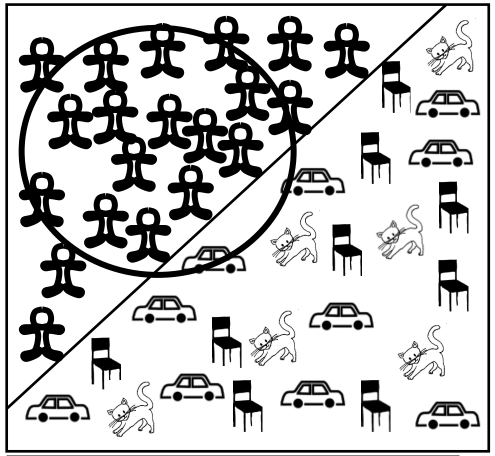
Exercise 2: Describe the following classifiers in terms of their confusion matrices
[#People=20 , #Non-People=25]



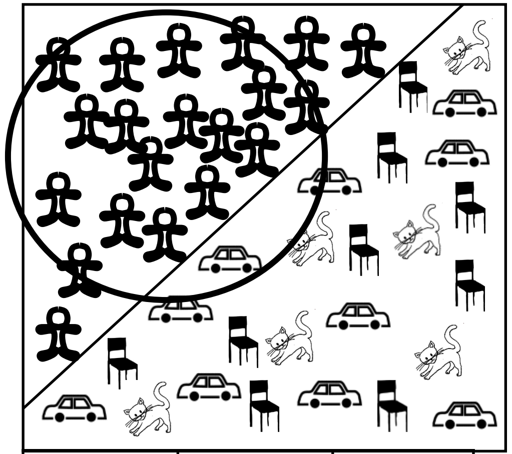
TP:	FN:	
FP:	TN:	



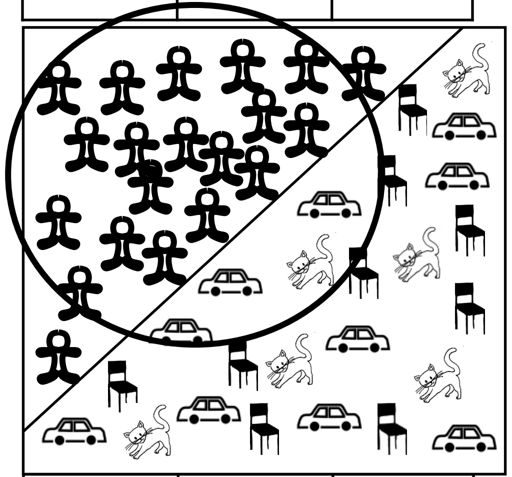
TP:	FN:	
FP:	TN:	



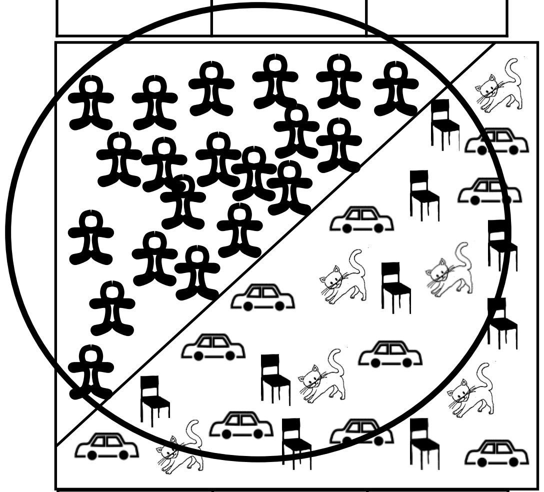
TP:	FN:	
FP:	TN:	



TP:	FN:	
FP:	TN:	

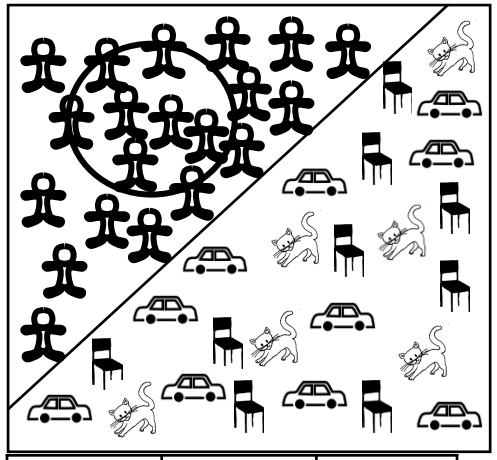


TP:	FN:	
FP:	TN:	

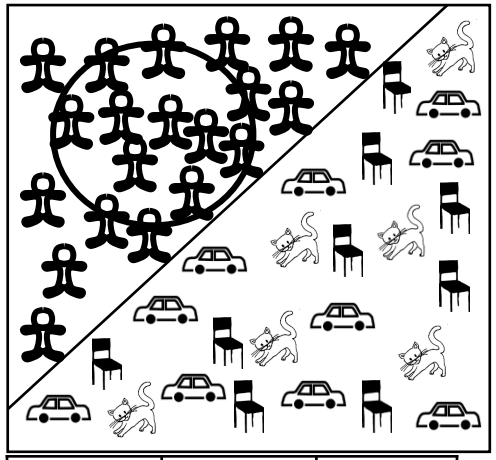


TP:	FN:	
FP:	TN:	

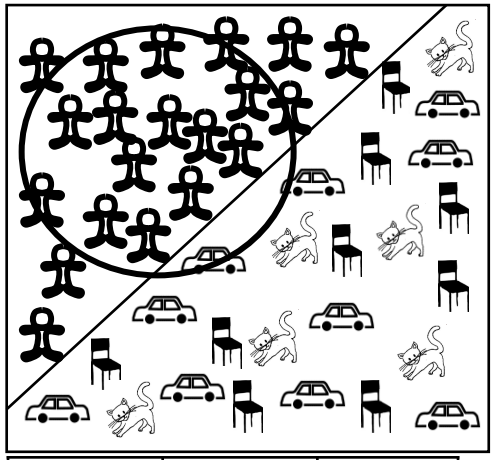
Exercise 2: Describe the following classifiers in terms of their confusion matrices
[#People=20 , #Non-People=25]



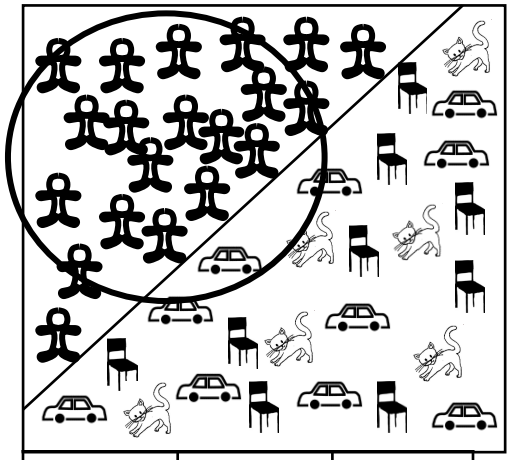
TP: 7	FN: 13	0,35
FP: 0	TN: 25	1
1		0,71



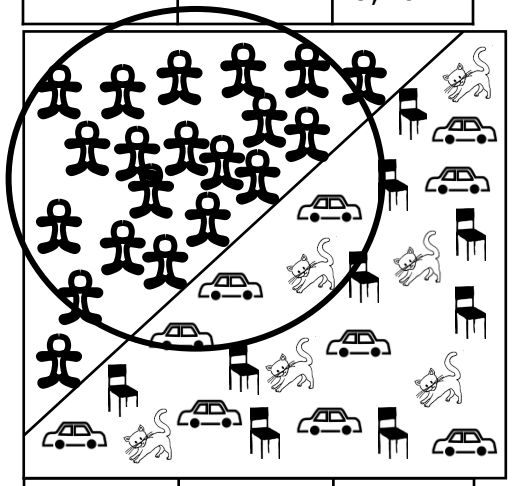
TP: 9	FN: 11	0,45
FP: 0	TN: 25	1
1		0,76



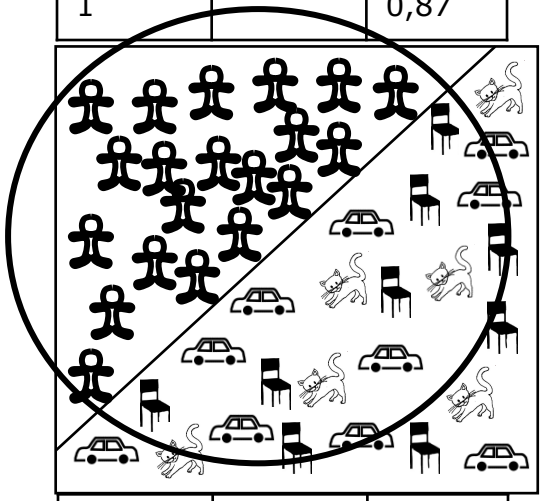
TP: 14	FN: 6	0,70
FP: 0	TN: 25	1
1		0,87



TP: 17	FN: 3	0, 85
FP: 1	TN: 24	0,96
0,94		0,91

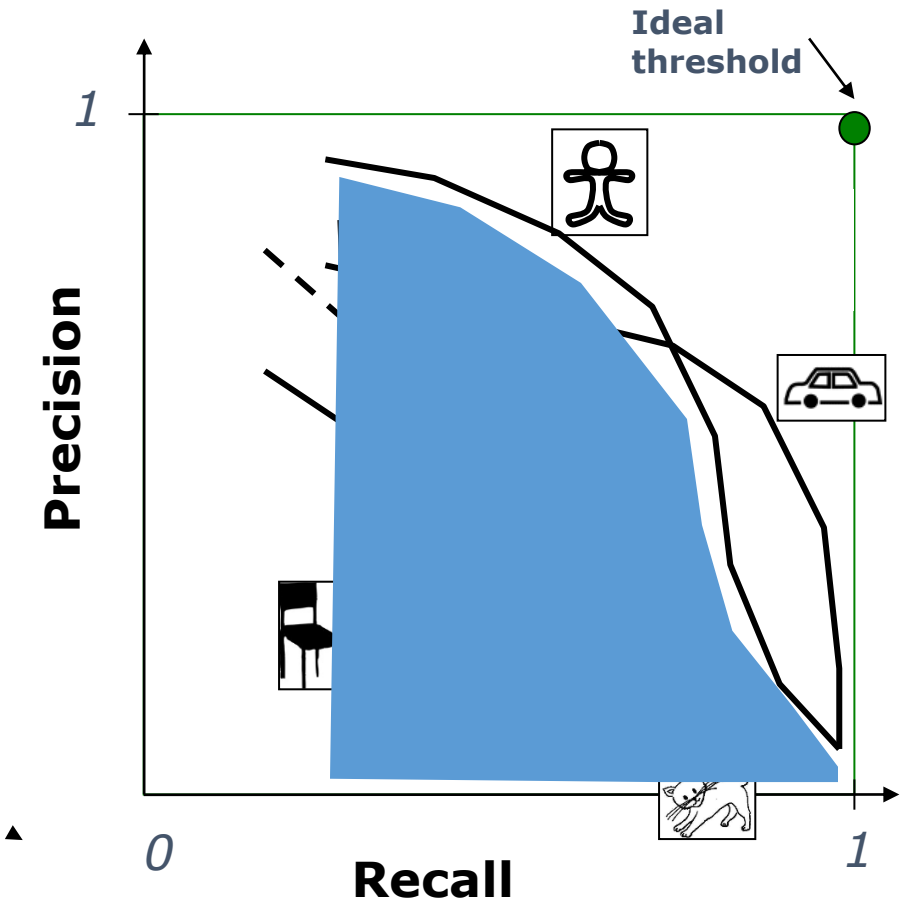


TP: 18	FN: 2	0,90
FP: 4	TN: 21	0,84
0,82		0,87

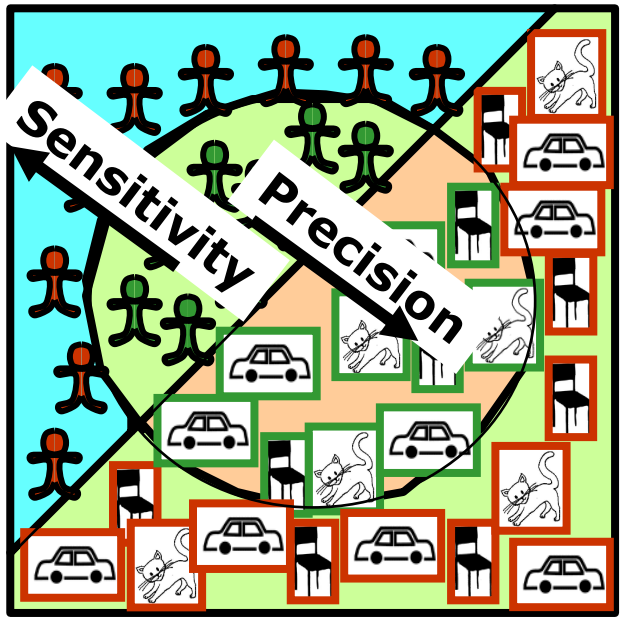


TP: 20	FN: 0	1
FP: 17	TN: 8	0,32
0,54		0,62

Plotting measures in the space of possible thresholds.
It fully defines your classifier



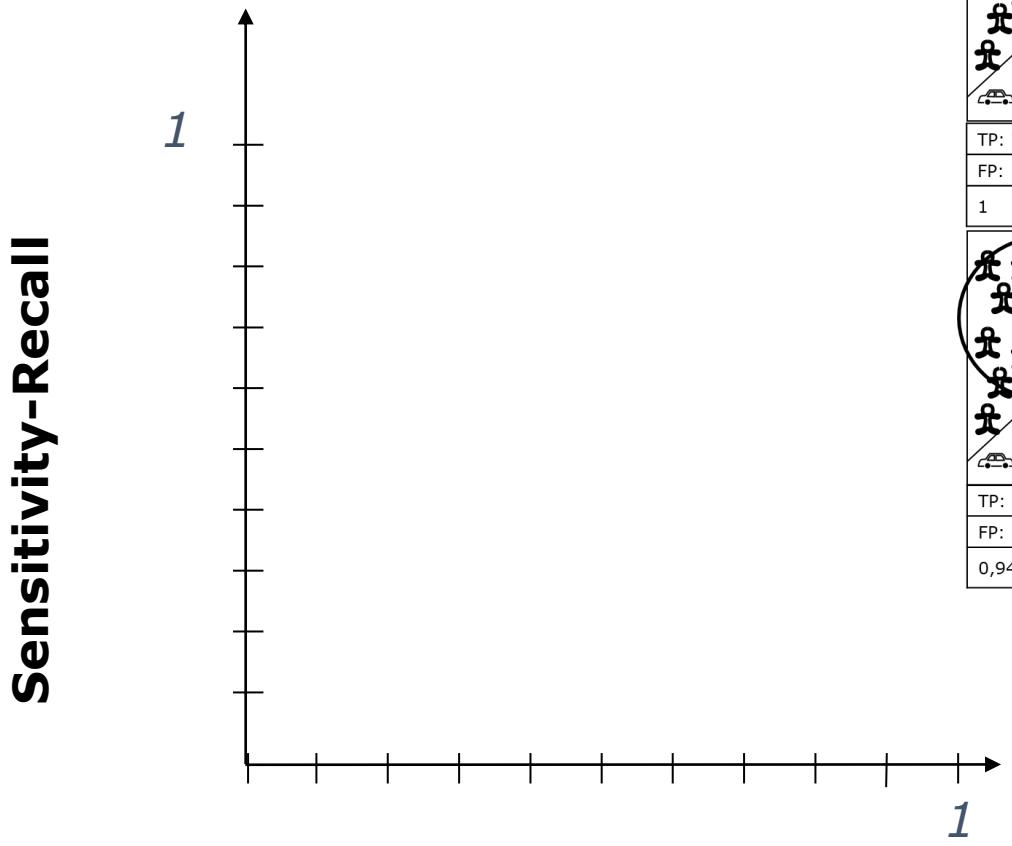
Precision / Recall

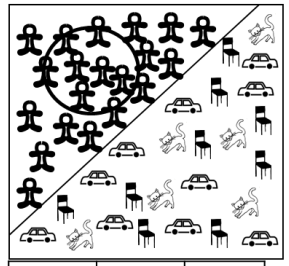


RECOGNIZER OUTPUT / CORRECT RECOGNITION	TRUE	FALSE	
POSITIVE	True Positive	False Negative	**Recall**
NEGATIVE	False Positive	True Negative	
	Precission		

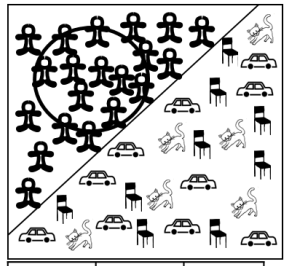
Average Precision:
averages precision over the entire range of recall

Exercise 3: Describe the Precision/Recall curve according to the threshold provided by the radius of the recognizer [#People=20 , #Non-People=25]

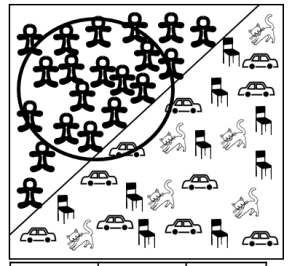




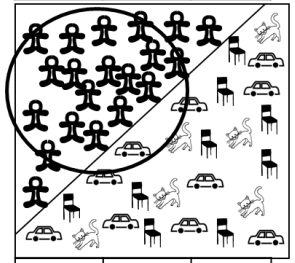
TP: 7	FN: 13	0,35
FP: 0	TN: 25	1
1		0,71



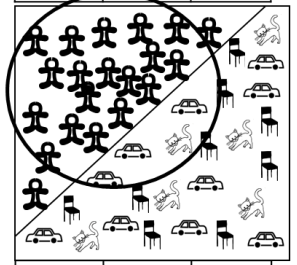
TP: 9	FN: 11	0,45
FP: 0	TN: 25	1
1		0,76



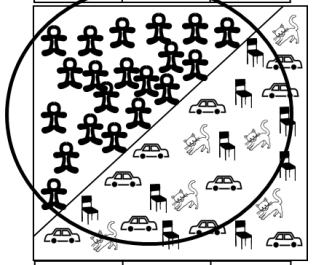
TP: 14	FN: 6	0,70
FP: 0	TN: 25	1
1		0,87



TP: 17	FN: 3	0, 85
FP: 1	TN: 24	0,96
0,94		0,91

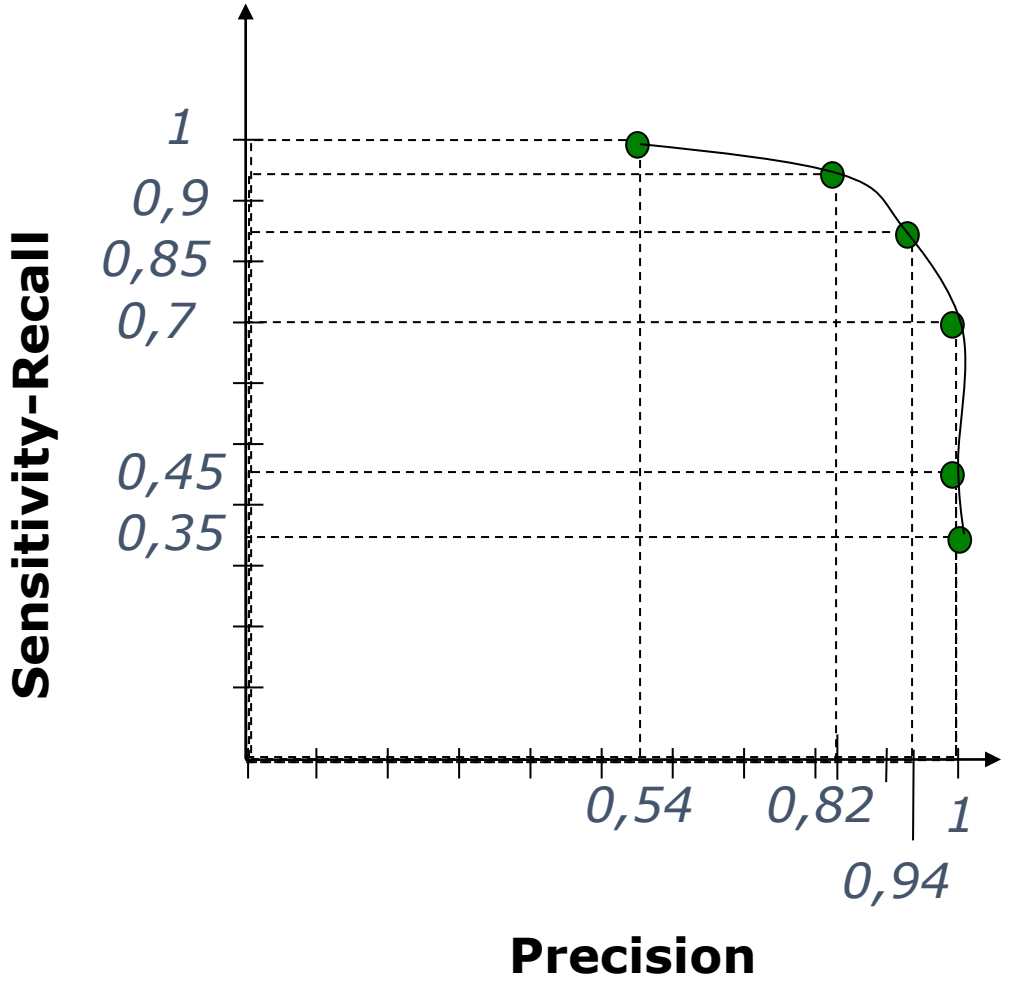


TP: 18	FN: 2	0,90
FP: 4	TN: 21	0,84
0,82		0,87

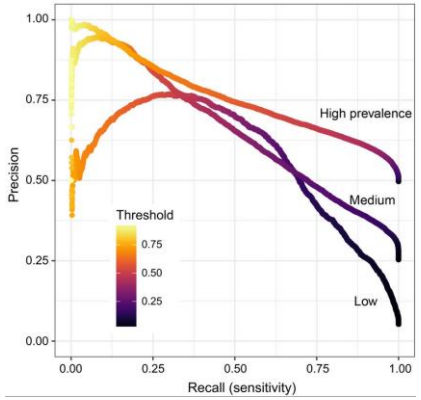


TP: 20	FN: 0	1
FP: 17	TN: 8	0,32
0,54		0,62

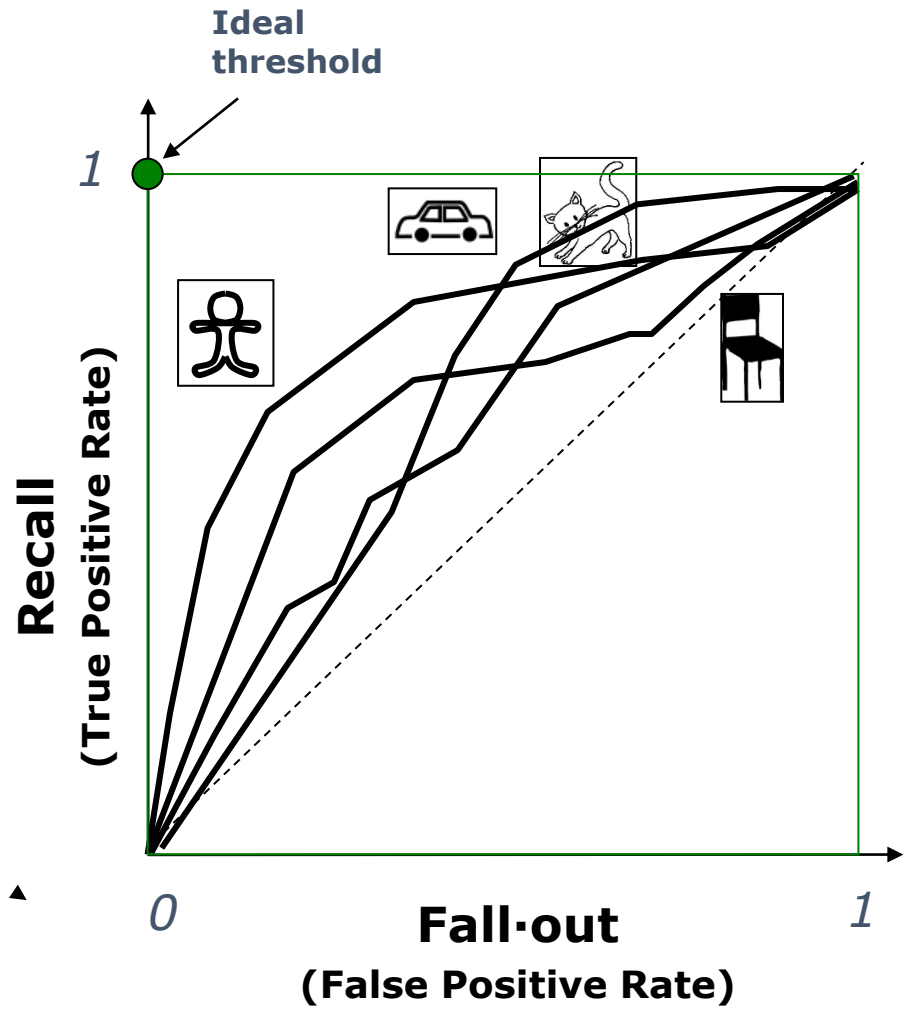
Exercise 3: Describe the Precision/Recall curve according to the threshold provided by the radius of the recognizer [#People=20, #Non-People=25]



Real data

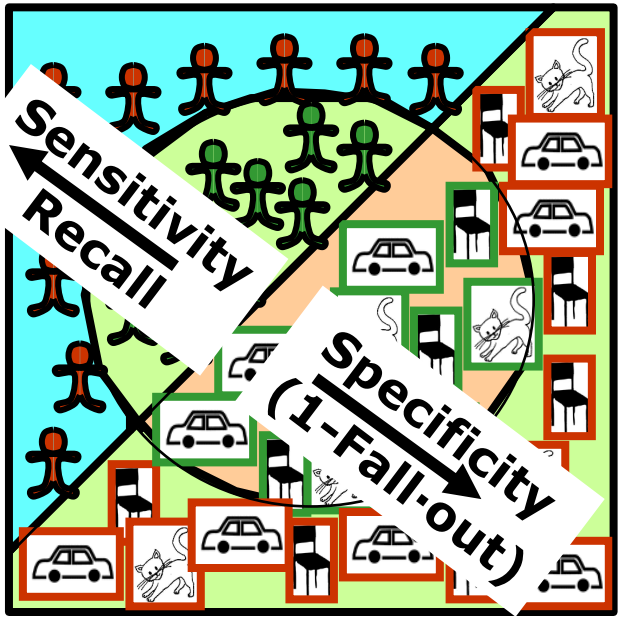


Plotting measures: ROC curves
RECEIVER OPERATING CARACTERISTIC (ROC)



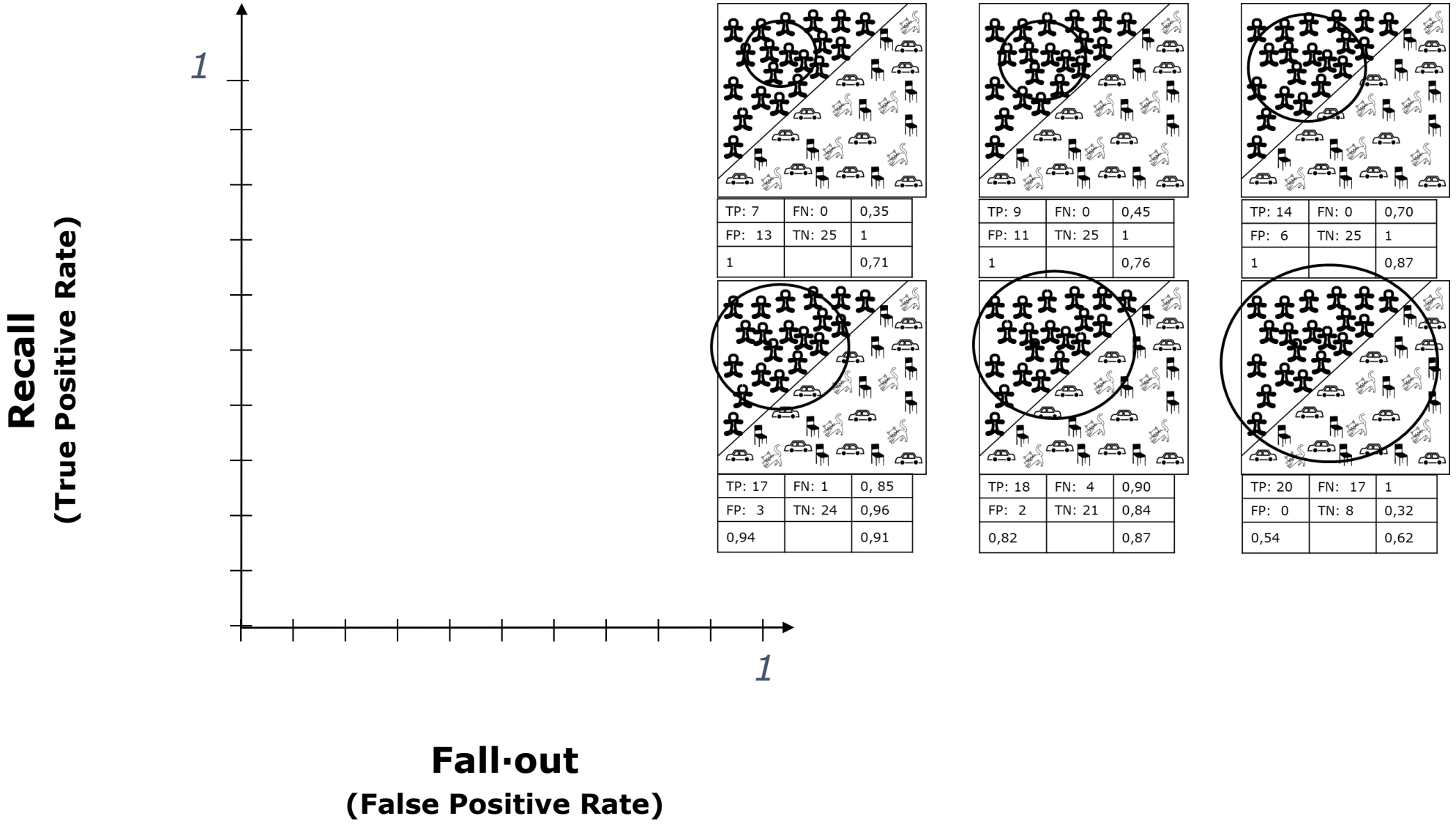
Good thresholds → **Recall > Fall·out**

Fall-out / Recall

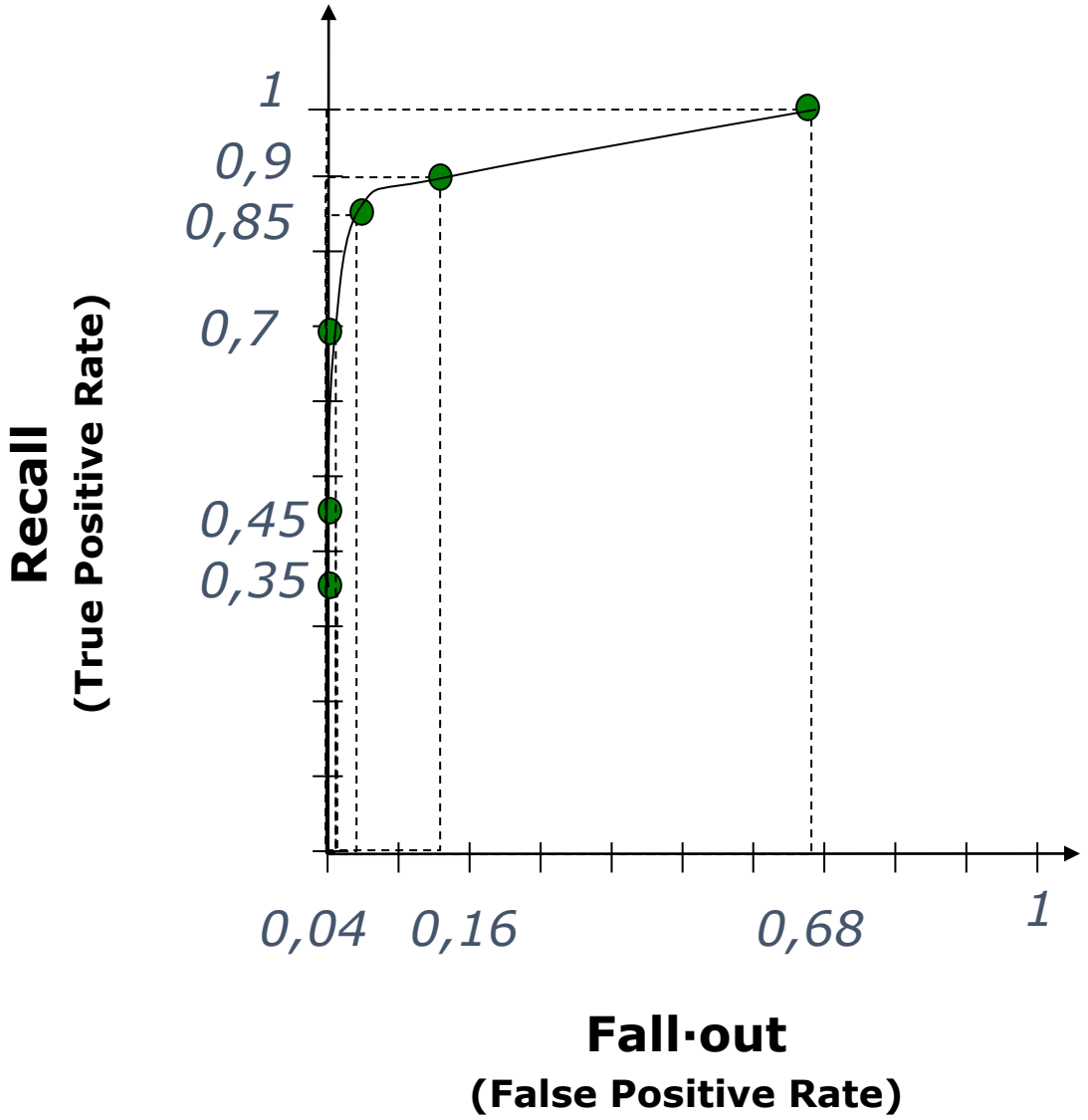


CORRECT RECOGNITION / RECOGNIZER OUTPUT	TRUE	FALSE	
POSITIVE	True Positive	False Positive	**Recall**
NEGATIVE	False Negative	True Negative	**Fall-out**

Exercise 3: Describe the ROC curve according to the threshold provided by the radius of the recognizer [#People=20 , #Non-People=25]



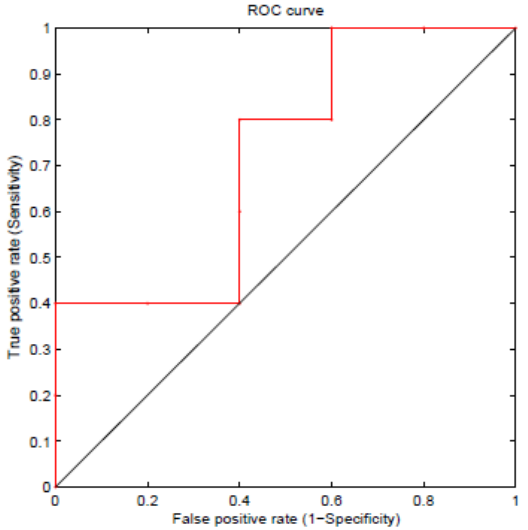
Exercise 3: Describe the ROC curve according to the threshold provided by the radius of the recognizer [#People=20 , #Non-People=25]



Exercise 1: Answer the following question.

16. (JW) A system which was designed to retrieve sailing-images gave the following retrieval result:

The whole data set contains five sail boats and five non-sailboats. Draw the ROC curve (horizontal: false positive rate and vertical true positive rate).



Contingency Matrices:

N: Retrieved Im.

True Pos	*False Pos*
False Neg	*True Neg*

(Recall,Fall-out)

$N = 1$

1	0
4	5

$(0.2, 0)$

$N = 2$

2	0
3	5

$(0.4, 0)$

$N = 3$

2	1
3	4

$(0.4, 0.2)$

$N = 4$

2	2
3	3

$(0.4, 0.4)$

$N = 5$

3	2
2	3

$(0.6, 0.4)$

$N = 6$

4	2
1	3

$(0.8, 0.4)$

$N = 7$

4	3
1	2

$(0.8, 0.6)$

$N = 8$

5	3
0	2

$(1, 0.6)$

$N = 9$

5	4
0	1

$(1, 0.8)$

$N = 10$

5	5
0	0

$(1, 1)$

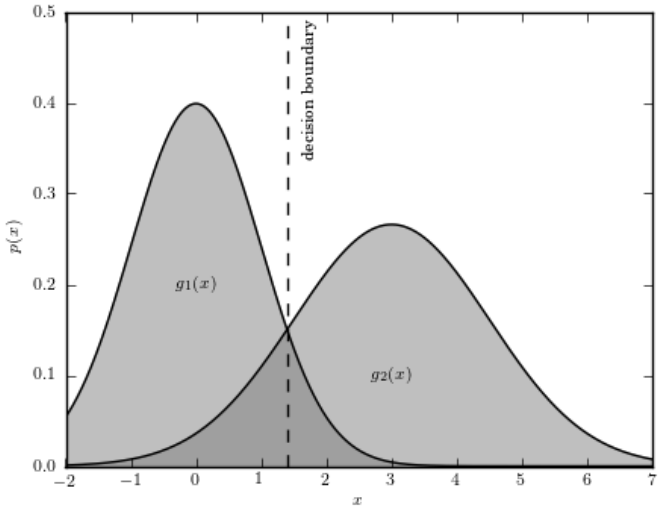
Experimetal setup: Some basic concepts

1. Training, test and validation datasets.
    1. Generalization power of a classifier.
    2. Asymptotical error.
    3. Difference between train and test set. Implications in over fitting.
    4. The validation set.
    5. The data size.
2. Cross-validation and leave-one out.
    1. Data set splitting.
    2. Cross validation. Average value and standard deviation. Confidence interval.
    3. Leave one out.
3. Classical performance metrics.
    1. Error, sensitivity, specificity, precision, f-measure.
    2. ROC curves, AUC and operation point of the system.
    3. Precision-recall curves.
4. Obtaining inference from the data.
    1. Hypothesis testing: Null hypothesis, alternative hypothesis, statistic, p-value.
    2. Example of t-test function.

Training, test and validation datasets - 1/4

- We want to **assess the performance** of a given classifier.
- The process of adapting the classifier to a specific problem is called **training**. We say that we **train** the classifier.
- We expect that the trained classifier can **generalise**.
  - This means that it can classify any sample in a correct way.
  - That for any previously unseen sample (not present in the training set), the classifier will have a correct prediction.
- A 100% perfect performance is not always possible for different reasons, among which:
  - The classifier cannot generalise the separable underlying model.
    - Because the model is too complex for the classifier.
    - Because we do not have enough samples to learn the underlying model.
  - The model is not separable.

*The **best** performance of any classifier is asymptotically bounded by the Bayesian error.*



http://www.astroml.org/book_figures/chapter9/fig_bayes_DB.html

Training, test and validation datasets - 2/4

- The data set used to train a classifier is called **training set**.
- The measure of the classification error on the training set is called **resubstitution error**.
- The resubstitution error provides us with useful information
- We can find 2 main problems (which are in essence the same one) when using the resubstitution error:
  - The performance of the classifier is assessed only with training samples, which could **or could not be representative enough** of all the population.
  - The classifier can learn particularities of the training set, that are not shared by the population, **over fitting** to the training set.

The error on the training set provides useful information, but for practical effects, it is not an estimate of the real performance of the classifier, due to lack of generalisation.

Training, test and validation datasets - 3/4

- We can use training and **test sets** in order to avoid over fitting.
- The training set and the test set **must be disjunctive** sets (no sample belongs to both sets).
- The test set provides the **performance of the system**.
- **All the parameters are tuned with the training** set.
- All the **unseen** samples are in the test set

Usually 80% training and 20% test is used... This rule is not carved on stone.

The larger the training set, the larger the options for the classifier to generalise well.

Datasets with low number of samples may not be representative of the underlying population.

Increasing the number of samples will not be a solution if unseen samples can arise which differ from the learnt model (e.g.: new videos arriving from a new hospital with a slightly different protocol).

Training, test and validation datasets - 4/4

- We can use training, **validation** and test sets in order to identify the best performing classifier.
- For the best performing approach we need to **sacrifice** some data. This is the validation data set.
- **The test set provides the final performance** of the system trained with the training. This classifier was the one that obtained the best performance on the validation set.

Usually, 60% is used for training and 20% for validation and 20% for test.
This is just a guideline, many options and variations can be frequently found.

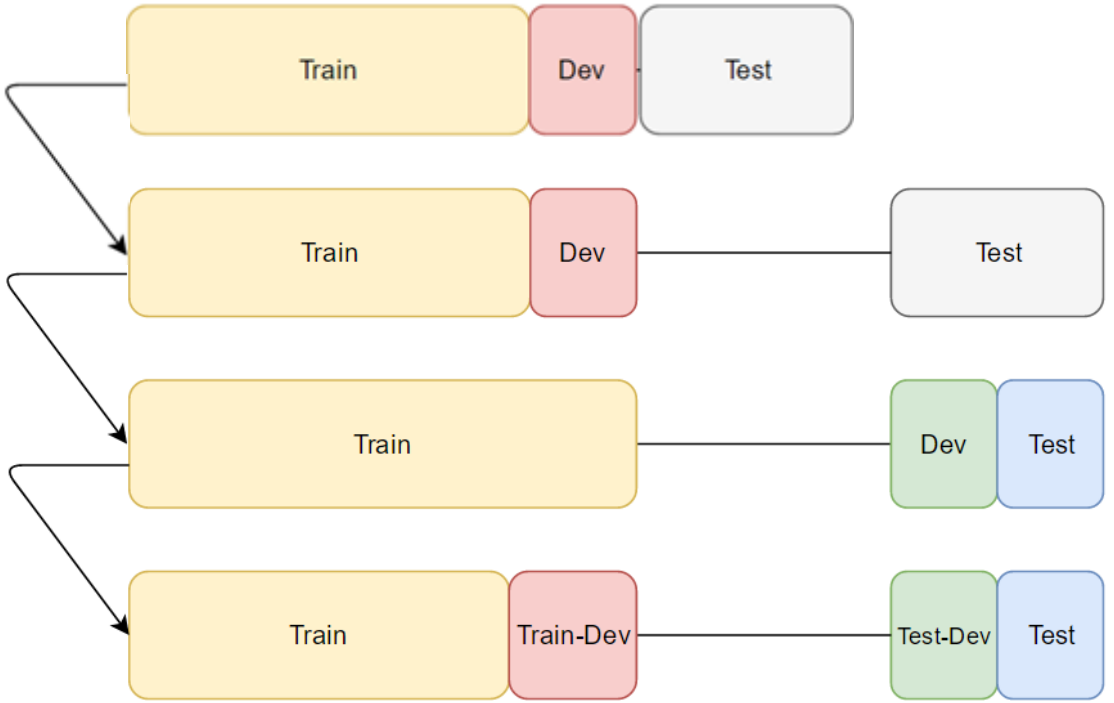
The validation set is used to select the classifier, trained with the training set.

The test set provides us with the final performance.

Training, test and validation datasets add-on

When applying models to a **new context**:

- We can use **test** on a **different** domain.
- We can **validate** and **test** on a **different** domain.
- We can set the model on a given domain and modify it (**validate** and **test)** on a **different** domain.



Nuts and Bolts of Applying Deep Learning (Andrew Ng)

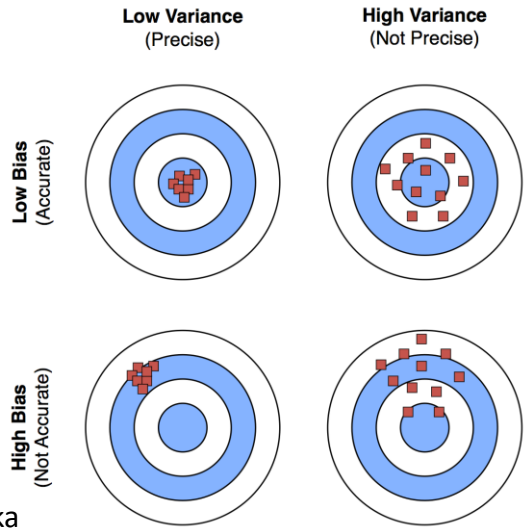
https://www.youtube.com/watch?v=F1ka6a13S9I&t=3044s

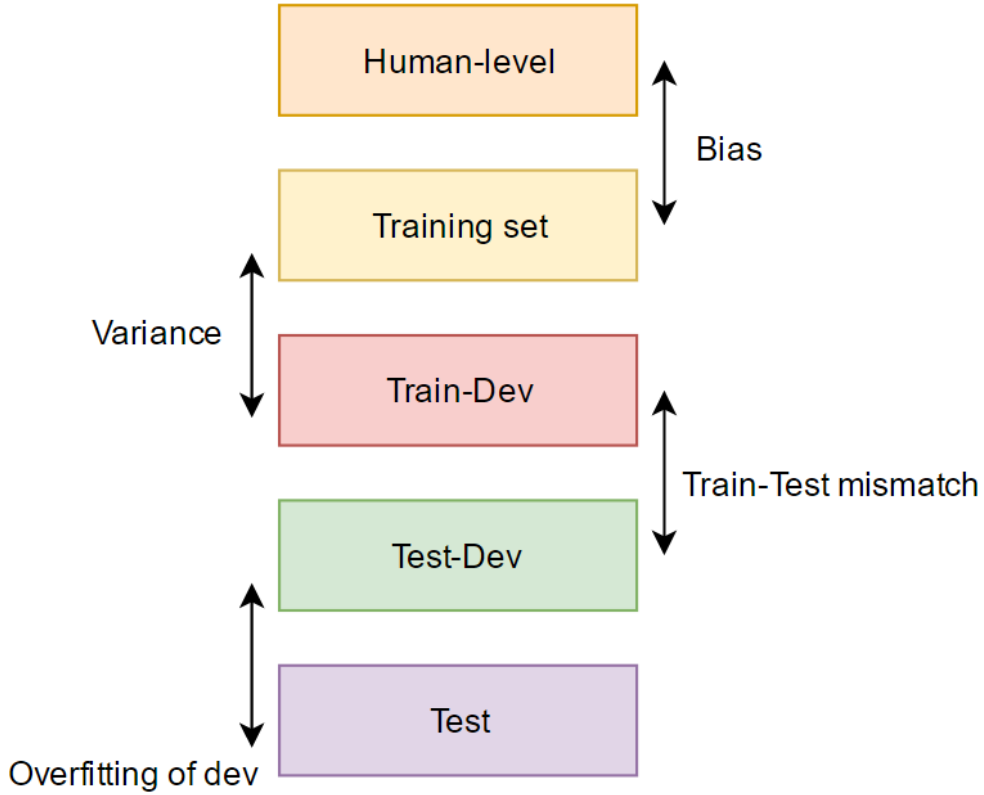
https://kevinzakka.github.io/2016/09/26/applying-deep-learning/

Bias-Variance

The **bias** is error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

The **variance** is error from sensitivity to small fluctuations in the training set. High variance can cause overfitting: modeling the random noise in the training data, rather than the intended outputs.



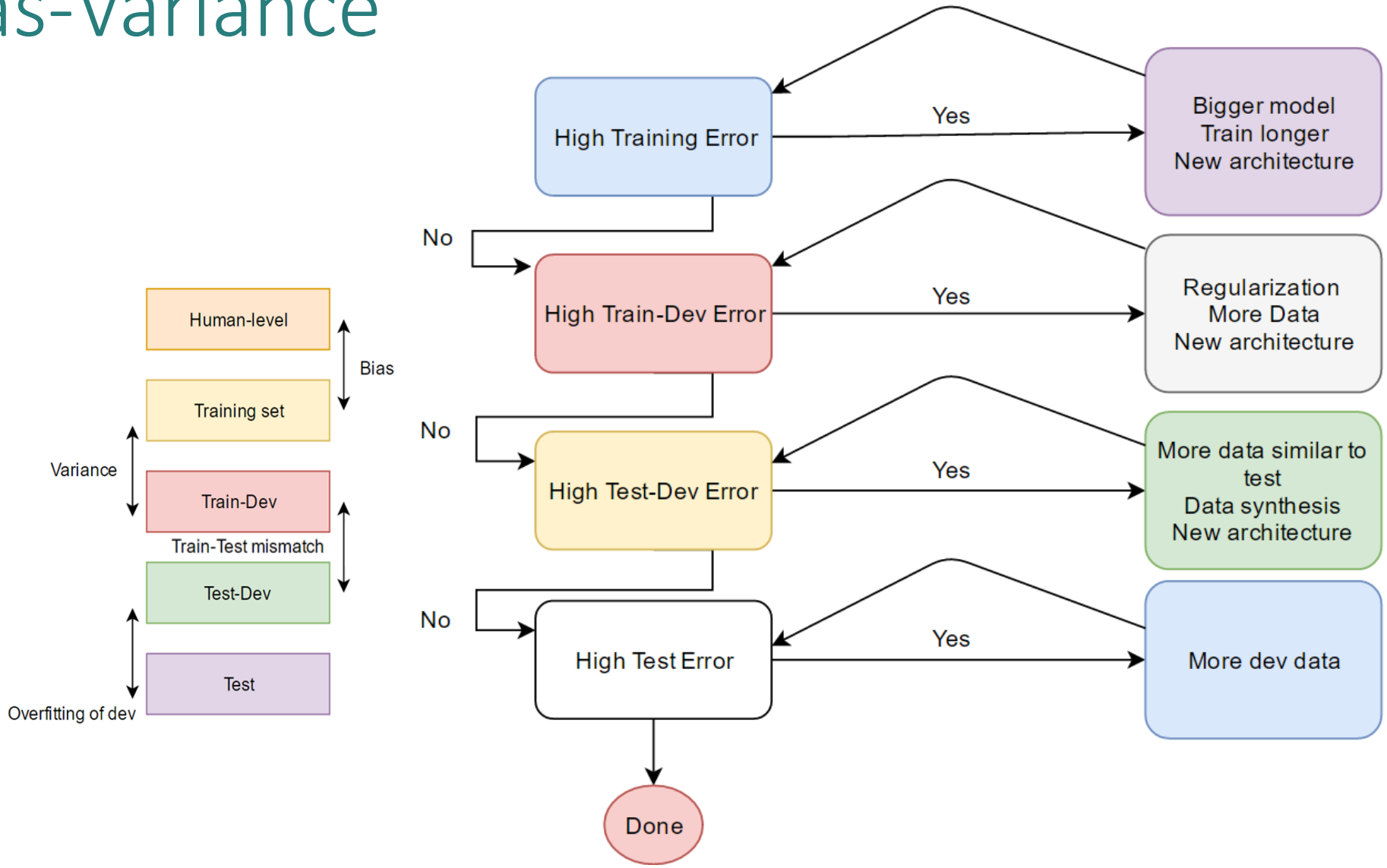


Nuts and Bolts of Applying Deep Learning (Andrew Ng)

https://www.youtube.com/watch?v=F1ka6a13S9I&t=3044s

https://kevinzakka.github.io/2016/09/26/applying-deep-learning/

Bias-Variance



Cross-validation

The **splitting** of the dataset in training, validation and test reduces the number of samples.

We now know that the reduction in the number of samples may **affect the power** of generalization of the classifier.

A potential solution is the use of **cross-validation** as follows:

1. Random splitting of data in k-subsets of equal or similar size.
2. Pick one subset and train the classifier with the remaining k-1 subsets.
3. Repeat k times until all subsets are used for testing.

Show the final performance as: **Average (standard deviation)**

*The **average value** is an estimate of the asymptotical performance of the classifier.*

*The **true performance** value of the system is found within +/- two standard deviation distance from the average value, with a 95% of certainty.*

*This defines a **confidence interval** in which we can guarantee the presence of the real average value.*

Leave-on-out

In the extreme, we can pick only one sample and train with the remaining samples. This is the **leave-one-out** approach.

- In the leave-one-out approach, **we maximise** all the date we have for training.
- However, all the **trained classifiers are very similar**, and we do not have a measure of the dispersion (no standard deviation).
- For this reason, we do not have information regarding the **potential uncertainty** of the result.

In the leave-one-out approach, we train N classifiers for N samples in our training set, which can be highly cost effective.

Bootstrapping validation.

Bootstrapping

Uniform sampling with replacement of available items (once a sample is selected, it goes back to the training sample and could be selected once again).

- **0.632 bootstrap**: Givena set of X data, we took X smaples. The data not selected as training sample will be the tets set.

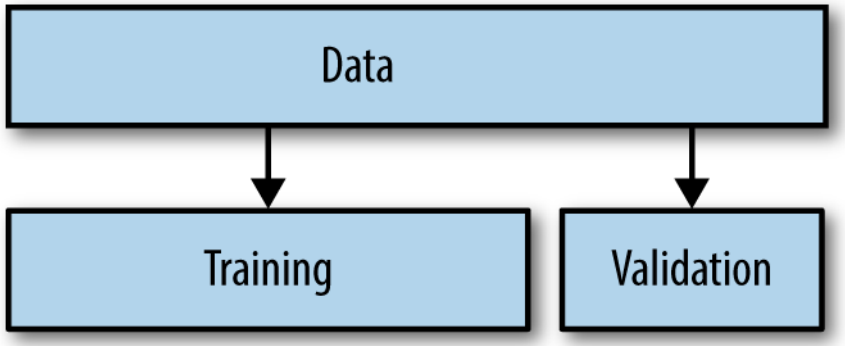
  Around 63.2% of the samples will be in the “bootstrap” (training set) and 36.8% in the test set:

  $(1-1/x)^x \approx e^{-1} = 0.368$

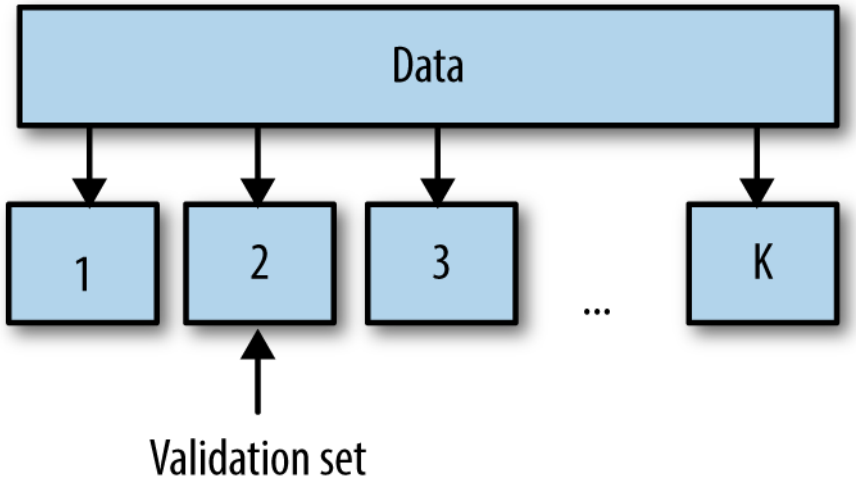
  If we repeat the process k times:

$$acc(M) = \sum_{i=1}^{k} (0.632 \times acc(M_i)_{test\_set} + 0.368 \times acc(M_i)_{train\_set})$$

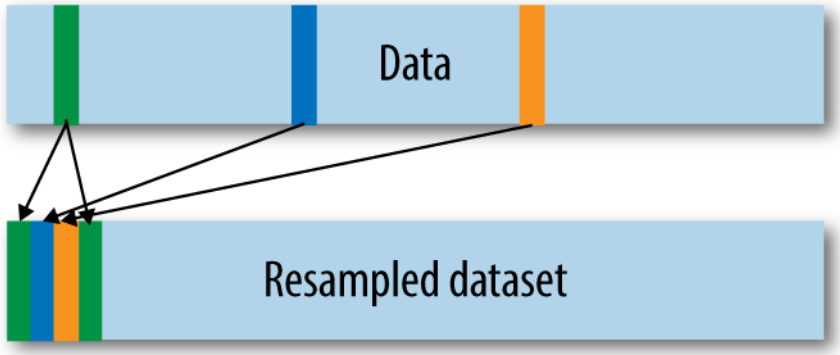
Hold-out validation



K-fold cross validation



Bootstrap resampling



Some classical performance metrics

Accuracy – Error:

```
Acc = (TP + TN)/(ALL)
Err = (FP + FN)/(ALL)
```

Sensitivity (*aka* Recall):

```
Sens = TP / (TP + FN)
```

Specificity:

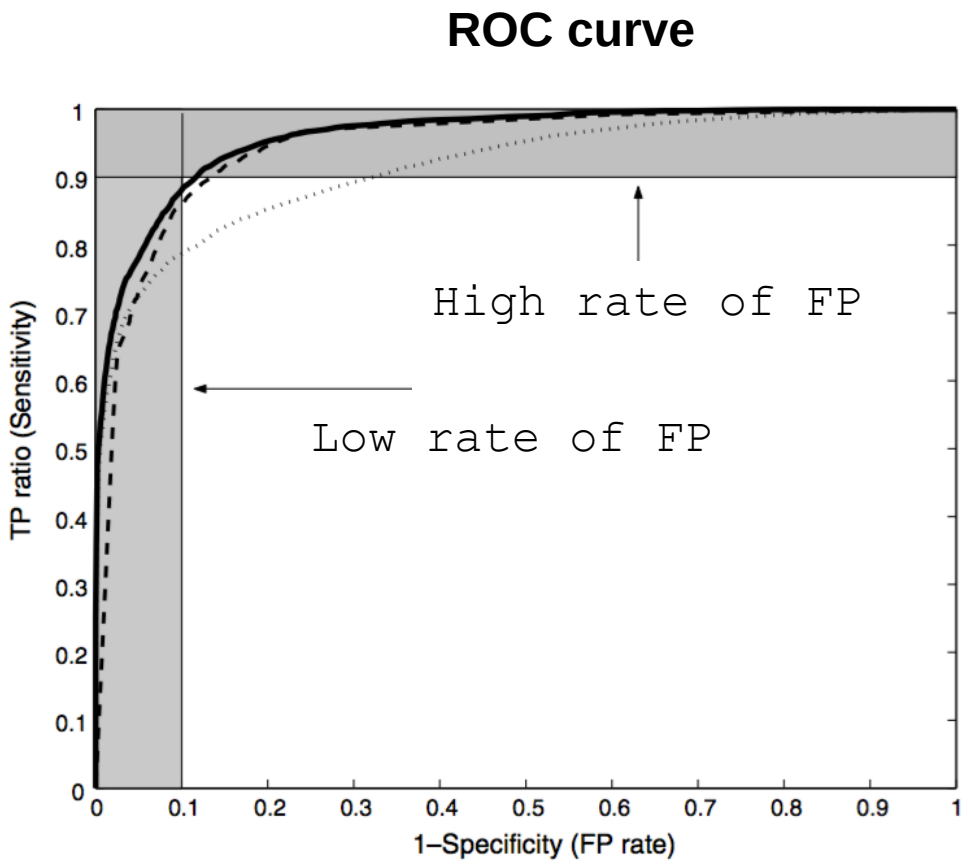
```
Spec = TN / (TN + FP)
```

Precision:

```
Prec = TP / (TP + FP)
```

F-measure:

```
F=2*sens*prec/(sens+prec)
```



The area under the ROC curve AUC is a metric for the classifier profile as a whole, and it is typically used for ranking.

Nevertheless, once in production, the classifier will always work on a single specific operation point of the ROC curve.

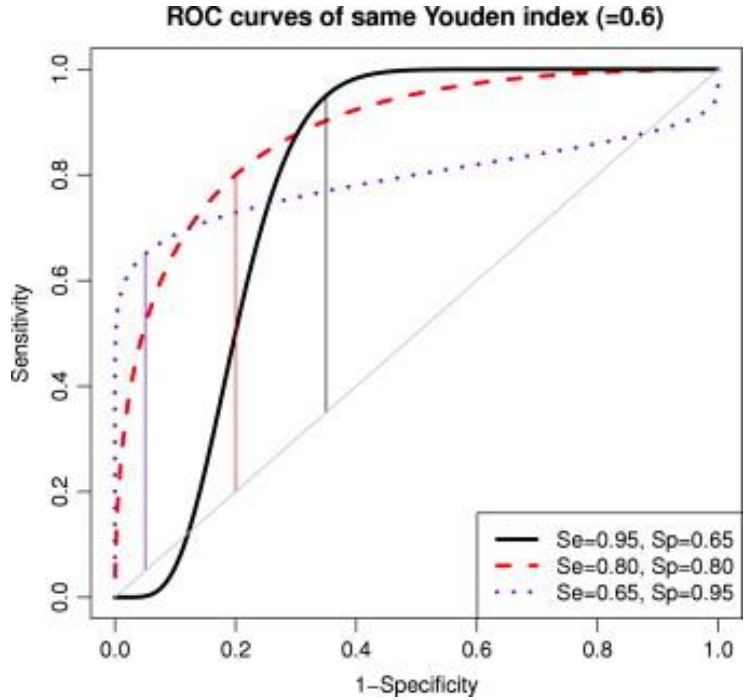
Better cut-off threshold

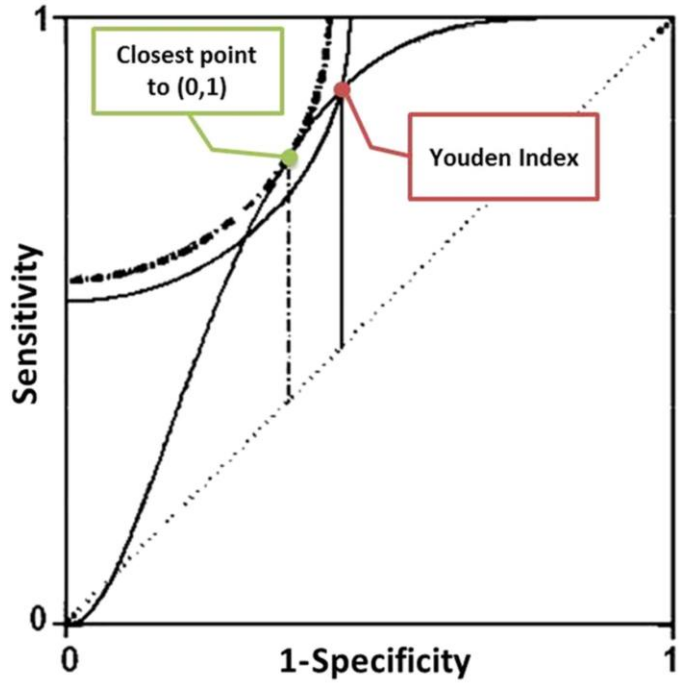
Youden index:

```
J = sensitivity + specificity - 1
Thr = arg max(J)
```

(0,1) distance:

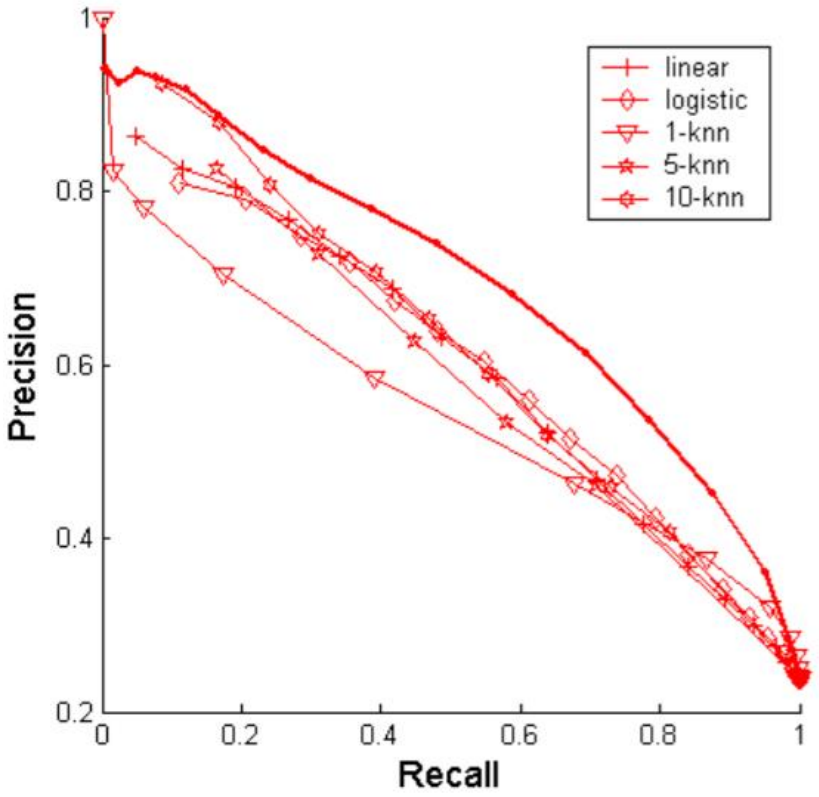
```
D = ( (1-sensitivity)²+(1-specificity)² )
Thr = arg min(D)
```





Precision-recall curve

- In order to have a view of the weight of the FP (particularly unbalanced problems) the **precision-recall** PR-curves are useful.
- The PR-curve does not have the same properties as the ROC curve, since it is not monotonically growing down.
- In addition the PR-curve **can start** at an arbitrary working point.
- For that reasons curves are more **difficult to compare**.



Summary

- ✓ Accumulate as much data from different sources as possible.
- ✓ Split your dataset in training, validation and test set.
- ✓ Use the training data to train different classifiers. Use the validation for selecting the optimal version of the classifier. Use the test to obtain the final performance.
- ✓ Choose appropriate metrics to asses the performance of your classifier and analyze its behavior both quantitatively and qualitatively.
- ✓ Provide ROC or PR curves for the identification of potentially interesting operational points.
- ✓ Obtain aggregated results in terms of average, standard deviation and confidence intervals.

Interesting links for further reading

1. *Frank Keller*. **Frank Keller´s tutorial on Evaluation** http://www.coli.uni-saarland.de/~crocker/Teaching/Connectionist/lecture11_4up.pdf
2. *Thacker et al*. **Performance characterization in computer vision: A guide to best practices**. Computer Vision and Image Understanding 109 (2008) 305–334. http://www.csd.uwo.ca/faculty/barron/PAPERS/CVIU2008.pdf
3. *Adrian F. Clark and Christine Clark*. **Performance Characterization in Computer Vision. Technical report**. http://vase.essex.ac.uk/talks/performance-evaluation.pdf