# Machine Learning and Pervasive Computing

Stephan Sigg

13.04.2015

# Overview and Structure

| | |
|---|---|
| 13.04.2015 | Organisation |
| 13.04.2015 | Introduction |
| 20.04.2015 | Rule-based learning |
| **27.04.2015** | A simple Supervised learning algorithm |
| 04.05.2015 | Excursion: Avoiding local optima with random search |
| 11.05.2015 | – |
| **18.05.2015** | High dimensional data |
| 25.05.2015 | – |
| 01.06.2015 | Artificial Neural Networks |
| **08.06.2015** | Decision Trees |
| 15.06.2015 | k-Nearest Neighbour methods |
| **22.06.2015** | Probabilistic models |
| 29.06.2015 | Topic models |
| **06.07.2015** | Unsupervised learning |
| **13.07.2015** | Anomaly detection, Online learning, Recom. systems |

# Outline

# What is machine learning ?

# What is machine learning ?

## Wikipedia

*Machine learning is a subfield of computer science and artificial intelligence that deals with the construction and study of systems that can learn from data, rather than follow only explicitly programmed instructions.*

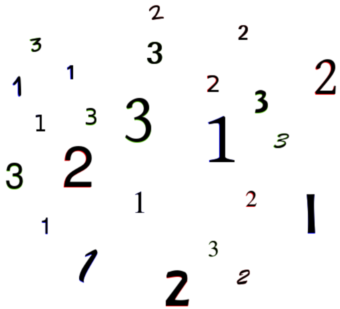
# What is machine learning ?

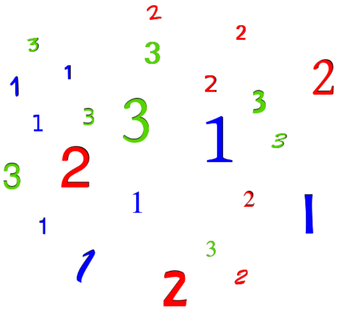
## Wikipedia

*Machine learning is a subfield of computer science and artificial intelligence that deals with the construction and study of systems that can learn from data, rather than follow only explicitly programmed instructions.*

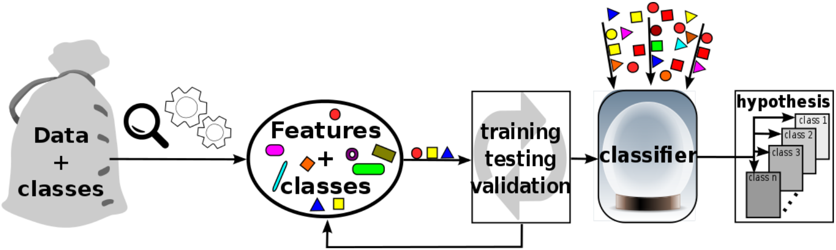
## Tom M. Mitchell, The Discipline of Machine Learning, 2006

*Machine Learning seeks to answer the question how we can build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes'*



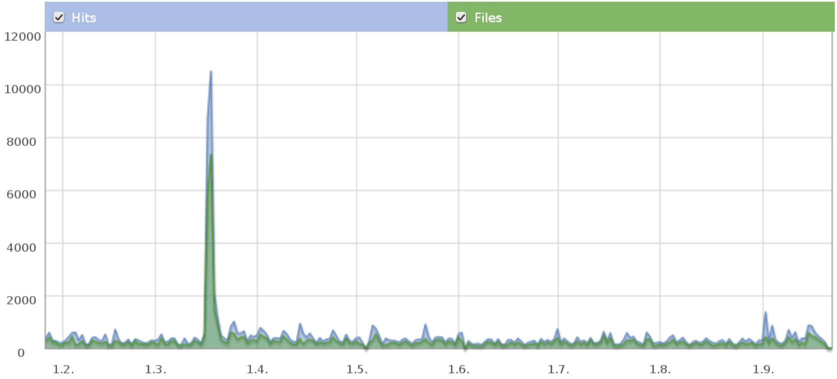


# Toolchain for Machine learning algorithms



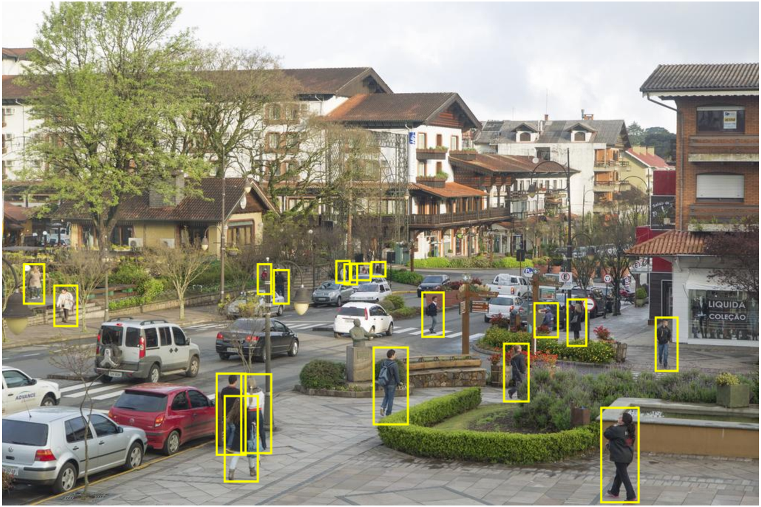
# Artificial intelligence

# Anomaly detection



# Image segmentation

# Image segmentation



# Image segmentation

# Image segmentation

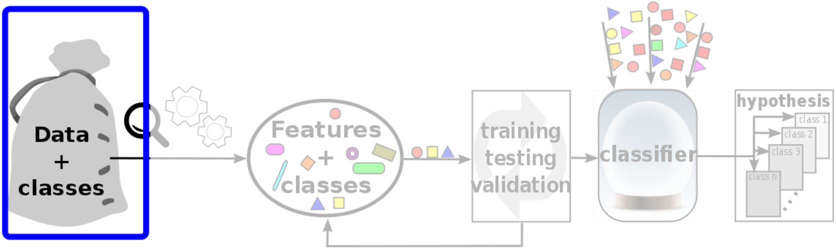
# Image segmentation

# Recommender systems

Here: http://www.movielens.org

# Talking about data

## Typical input for Machine-learning algorithms

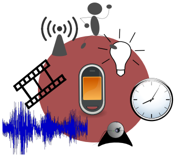


# Sensors and sensor classes

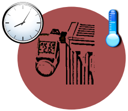
Typical input for Machine learning algorithms might originate from sensors

## Sensing modalities

→ Smartphones
→ Smart environments
→ Sensor networks
→ IoT
→ Cars/vehicles
→ CCTV



Time
RF
Temperature
Light
Audio
Video
Smoke
Fingerprint
...



# Example sensors and classes

# Sensors and sensor classes

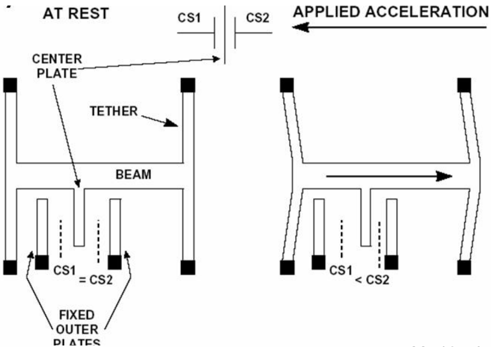
## MEMS acceleration sensors

**E.g.** Analogue Devices ADXL

$\rightarrow$ Low energy consumption, small, cheap, medium precision

MEMS = Micro-mechanical System: Mechanic in Silicon (Silizium)

**Principle** Comparison of capacity CS1 and CS2 leads to acceleration

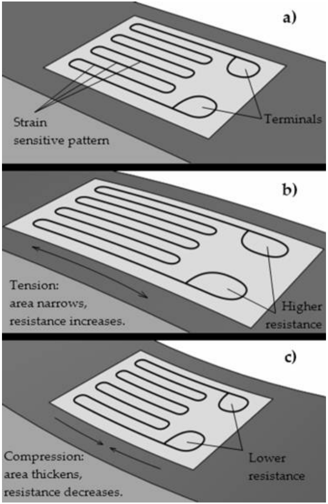


# Sensors and sensor classes

## Pressure sensors

E.g. IEE about 3-10 Euro

→ Very imprecise



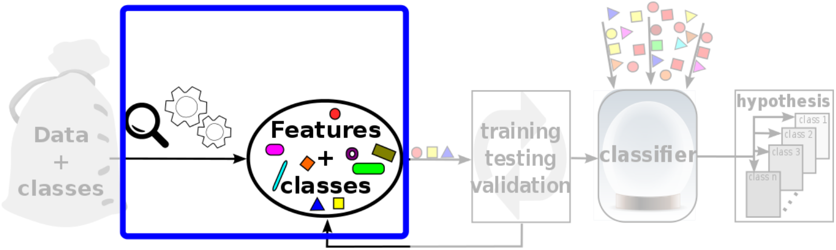
# Sensors and sensor classes

Output of sensors has to be further processed typically

- Raw electrical signals
- Interpretation of signals as electric values
- Binary or Real valued representation
- Further identification of features
- Feature extraction and interpretation

# Outline

# Features and feature selection




**Problem** Typically, raw data from sensors is not well suited as input for Machinle learning approaches

→ Noisy

→ Not meaningful/expressive

**Feature extraction** Process the data in order to obtain features which are meaningful and correlated to respective classes

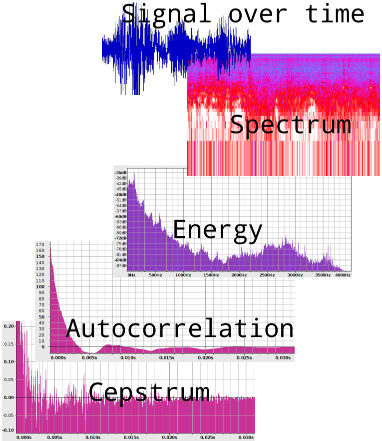
# Features and feature extraction

## What is a feature?

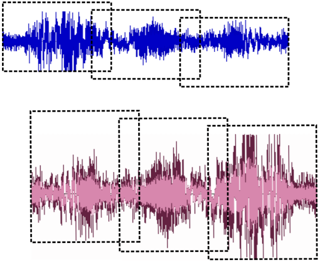
- Sensor data might be hard to interpret
- Multiple aspects might be contained in a single data stream (hard to distinguish by ML algorithm)

e.g. Audio

- → Loudness
- → Energy on frequency bands
- → Zero crossings
- → Direction changes



# Feature extraction

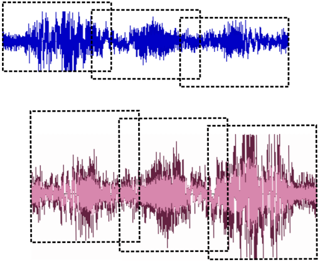


Pre-process data

→ Framing

→ Normalisation

# Feature extraction



Pre-process data

→ Framing

→ Normalisation

Domain knowledge?
-> better set of ad-hoc features

Features commensurate?
-> normalise

Pruning of input required?
-> if no, create disjunctive features or weithted sums of features

Independent features?
-> construct conjunctive features or products of features

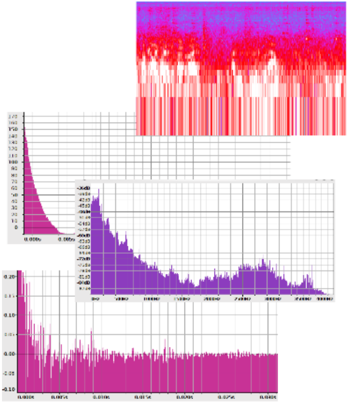
Is the data noisy?
-> detect outlier examples

Do you know what to do first?
-> If not, use a linear predictor

# Feature extraction

→ Identify meaningful features

- remove irrelevant/redundant features

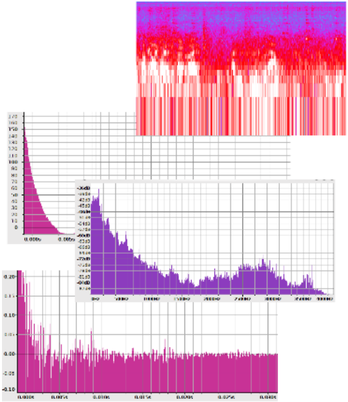


# Feature extraction

→ Identify meaningful features

- remove irrelevant/redundant features

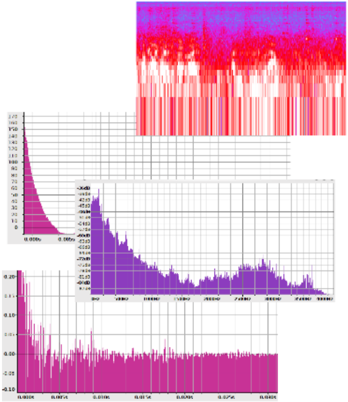
→ Features can be contradictory!



# Feature extraction

**Task: Short presentation**

- Which are good/common features for Machine learning?



# Feature selection algorithms

## How to identify good/meaningful features?

**Feature selection**

How to find a good subset of features which is best suited to distinguish between the classes considered?

# Feature selection algorithms

## How to identify good/meaningful features?

### Feature selection

How to find a good subset of features which is best suited to distinguish between the classes considered?

### Las Vegas Filter

Repeatedly generate random feature subsets and computes their classification performance

# Feature selection algorithms

## How to identify good/meaningful features?

### Feature selection

How to find a good subset of features which is best suited to distinguish between the classes considered?

### Focus algorithm

1. Evaluate each singleton feature set
2. Evaluate each set of two features

⋮

Until consistent solution is found

# Feature selection algorithms

## How to identify good/meaningful features?

### Feature selection

How to find a good subset of features which is best suited to distinguish between the classes considered?

### Relief algorithm

Given a single feature, compute for all samples

$$\frac{\text{Closest distance to all other samples of the same class}}{\text{Closest distance to all samples not in that class}}$$

Rationale: Feature is more relevant the more it separates a sample from samples associated to other classes and the less it separates it from samples belonging to the same class

# Feature selection algorithms

## How to identify good/meaningful features?

### Feature selection

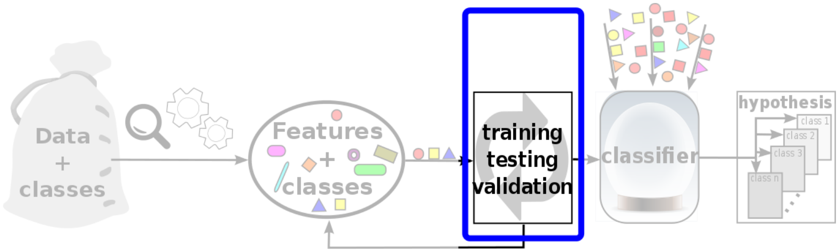
How to find a good subset of features which is best suited to distinguish between the classes considered?

### Pearson Correlation Coefficient

$$\varrho(X, Y) = \frac{\mathrm{Cov}(X, Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}$$

- Identifies linear relation between input variables $x_i$ and an output $y$

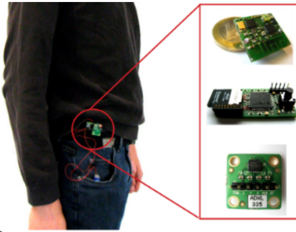
# Outline


Data
+
classes
Features
+
classes
training
testing
validation
classifier
hypothesis
class 1
class 2
class 3
class n

# Preparation: Sampling of sufficient data

## To train a ML algorithm, sufficient training data is required

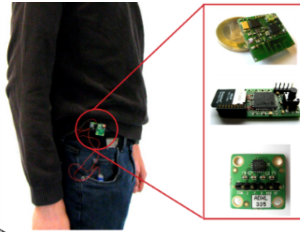
- → Annotated! (Ground-truth)
- → Multiple subjects
- → Various environmental conditions (time of day, weather, ...)



# Preparation: Sampling of sufficient data

## To train a ML algorithm, sufficient training data is required

- → Annotated! (Ground-truth)
- → Multiple subjects
- → Various environmental conditions (time of day, weather, ...)



### Example

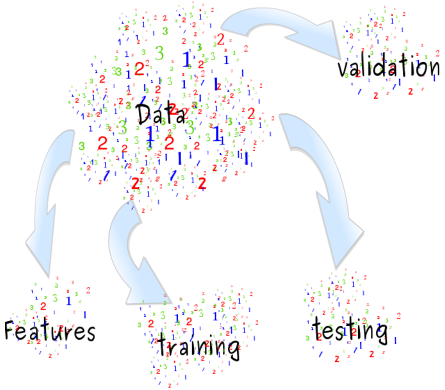
Electric supply data over 15 years covers 5000 days but only 15 christmas days

→ Especially critical events like accidents (e.g. plane, car, earthquake) are scarce

# Further processing: Importance of sufficient data

## Feature selection, training, testing, validation sets

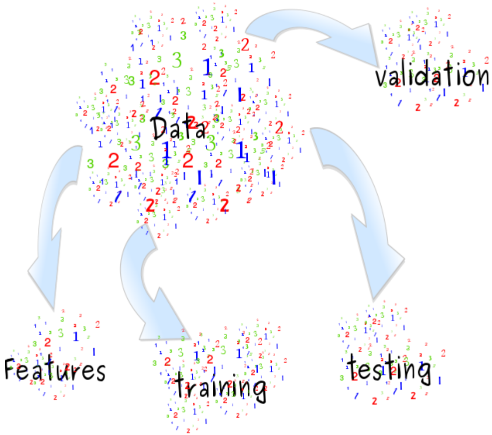
- Utilise dedicated data sets for feature selection and classifier training
- Pay attention that a single raw-data sample could not impact features in both these sets
- Don't train the features on the training- or test-data-set



# Selection of Training, Testing and Validation sets

**How to separate the data to train the classifier**

- Utilise dedicated test-, training- and validation-data-sets
- Pay attention that a single raw-data sample could not impact features in both these sets



# Evaluation of classification performance

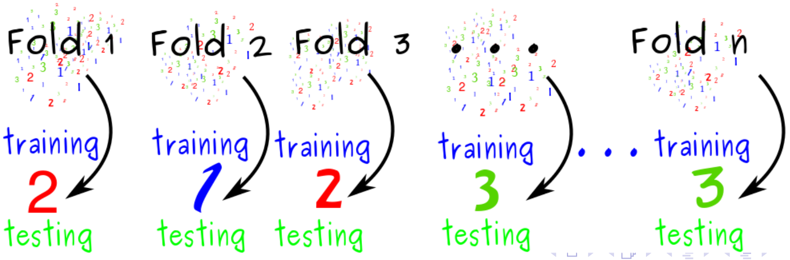
## k-fold cross-validation

- Standard: k=10

# Evaluation of classification performance

## Leave-one-out cross-validation

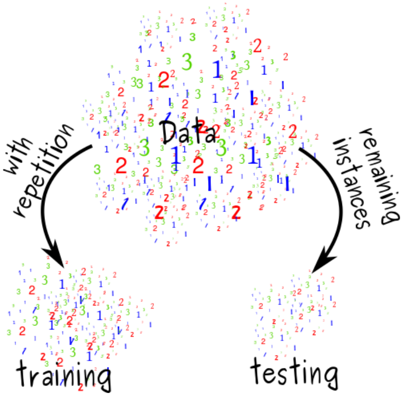
- n-fold cross validation where n is the number of instances in the data-set
- Each instance is left out once and the algorithm is trained on the remaining instances
- Performance of left-out instance (success/failure)



# Evaluation of classification performance

## 0.632 Bootstrap

- Form training set by choosing n instances from the data-set with replacement
- All not picked instances are used for testing
- Probability to pick a specific instance: $1-\left(1-\frac{1}{n}\right)^n \approx 1-e^{-1} \approx 0.632$



# Outline

# Evaluation of classification performance

## Evaluation of classification performance

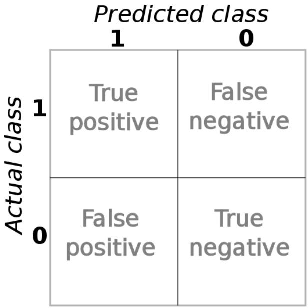
### Classification accuracy

- Confusion matrices
- Precision
- Recall
- $F_1$-score

| | Classification | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Aw | No | To | Sb | Sl | Sr | St | $\sum$ |
| Aw | **52** | | 3 | 6 | 0 | 17 | 22 | 100 |
| No | | **436** | 25 | 7 | 6 | 17 | 9 | 500 |
| To | | 40 | **59** | | | | 1 | 100 |
| Sb | 15 | 22 | | **32** | 4 | 22 | 5 | 100 |
| Sl | 12 | 11 | 1 | 6 | **48** | 8 | 14 | 100 |
| Sr | 4 | 15 | | 6 | 1 | **67** | 7 | 100 |
| St | 3 | 18 | 1 | 1 | 24 | 10 | **43** | 100 |
| $\sum$ | 92 | 551 | 86 | 65 | 94 | 129 | 83 | |

| | Classification | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Aw | No | To | Sb | Sl | Sr | St | recall |
| Aw | **.58** | .09 | | .13 | .11 | .05 | .04 | .58 |
| No | | **.872** | .05 | .014 | .012 | .034 | .018 | .872 |
| To | | .4 | **.59** | | | | .01 | .59 |
| Sb | .15 | .22 | | **.32** | .04 | .22 | .05 | .32 |
| Sl | .12 | .11 | .01 | .06 | **.48** | .08 | .14 | .48 |
| Sr | .04 | .15 | | .06 | .01 | **.67** | .07 | .67 |
| St | .03 | .18 | .01 | .01 | .24 | .1 | **.43** | .43 |
| prec | .630 | .791 | .686 | .492 | .511 | .519 | .518 | |

# Evaluation of classification performance



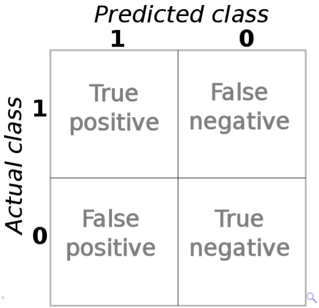
# Evaluation of classification performance

## Precision

Of all samples that were predicted with $y = 1$, what fraction actually belongs to class 1?

## Recall

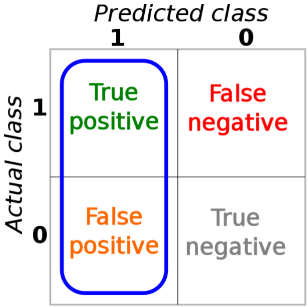
Of all samples that actually belong to class 1, which fraction has been correctly predicted with $y = 1$?



# Evaluation of classification performance

## Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$



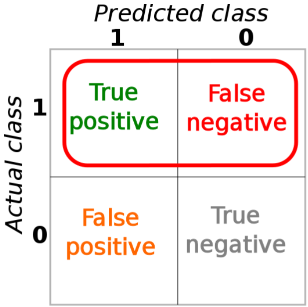
# Evaluation of classification performance

## Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

## Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



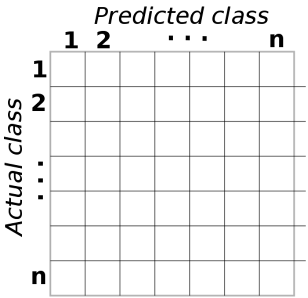
# Evaluation of classification performance

## Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

## Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



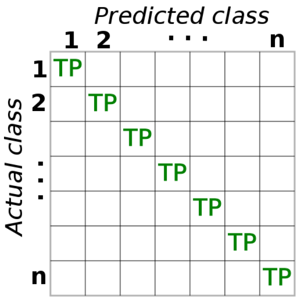
# Evaluation of classification performance

## Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

## Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



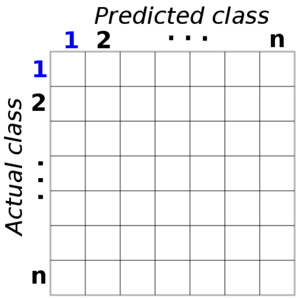
# Evaluation of classification performance

## Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

## Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



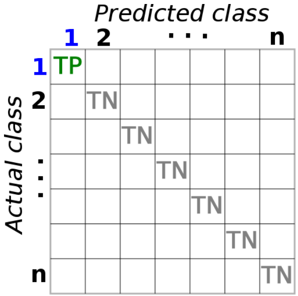
# Evaluation of classification performance

## Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

## Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



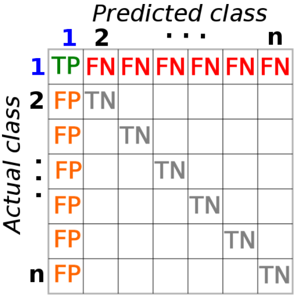
# Evaluation of classification performance

## Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

## Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



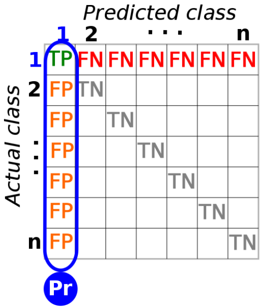
# Evaluation of classification performance

## Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

## Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



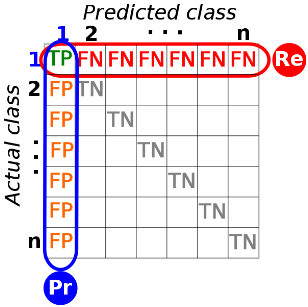
# Evaluation of classification performance

## Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

## Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



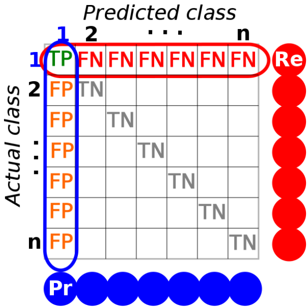
# Evaluation of classification performance

## Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

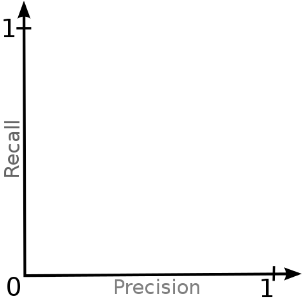
## Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$



# Evaluation of classification performance

Tradeoff between precision and recall



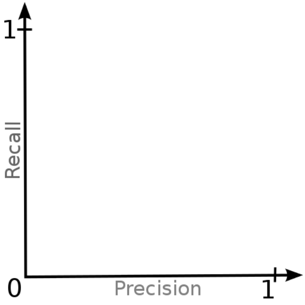
# Evaluation of classification performance

## Tradeoff between precision and recall

### Precision

$$\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

When we want to predict a particular class only if very confident $\Rightarrow$ High precision, Low recall



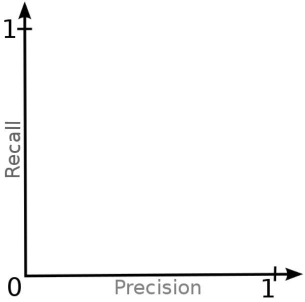
# Evaluation of classification performance

## Tradeoff between precision and recall

### Recall

$$\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$

When we want to minimise false negatives ⇒ Low precision, High recall



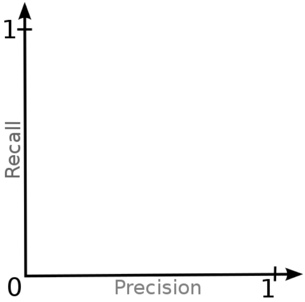
# Evaluation of classification performance

Tradeoff between precision and recall

## $F_1$ Score

In order to combine precision and recall into a single decision variable, we can use the

$$F_1 \text{ Score: } 2 \frac{\text{Precision}}{\text{Precision} + \text{Recall}}$$



# Training of the classifier

## Evaluation of classification performance

### Information score

Let C be the correct class of an instance and $\mathcal{P}(C)$, $\mathcal{P}'(C)$ be the prior and posterior probability of a classifier
We define:[1]

$$I_i = \begin{cases} -\log(\mathcal{P}(C)) + \log(\mathcal{P}'(C)) & \text{if } \mathcal{P}'(C) \geq \mathcal{P}(C) \\ -\log(1-\mathcal{P}(C)) + \log(1-\mathcal{P}'(C)) & \text{else} \end{cases}$$

The information score is then

$$\mathrm{IS} = \frac{1}{n}\sum_{i=1}^{n} I_i$$

[1] I. Kononenko and I. Bratko: Information-Based Evaluation Criterion for Classifier's Performance, Machine Learning, 6, 67-80, 1991.

# Training of the classifier

## Evaluation of classification performance

### Brier score

The Brier score is defined as

$$\text{Brier} = \sum_{i=1}^{n} (t(x_i) - p(x_i))^2$$

where

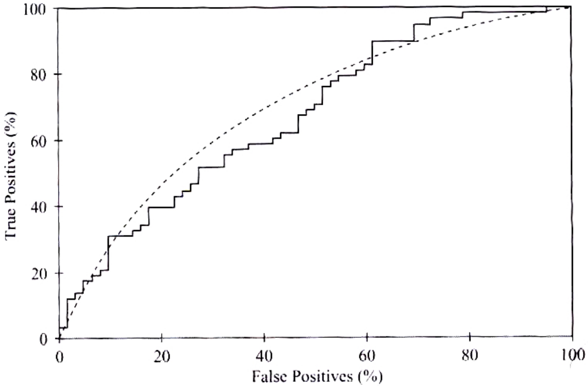
$$t(x_i) = \begin{cases} 1 & \text{if } x_i \text{ is the correct class} \\ 0 & \text{else} \end{cases}$$

and $p(x_i)$ is the probability the classifier assigned to the class $x_i$.

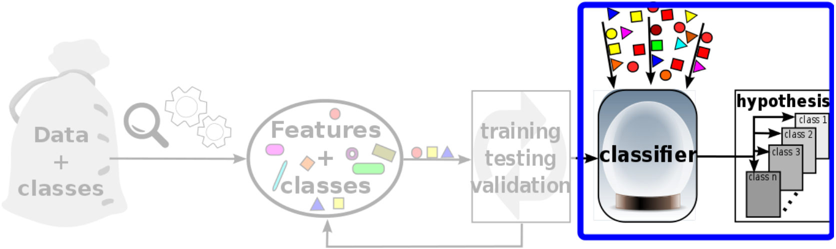
# Training of the classifier

## Evaluation of classification performance

### Area under the receiver operated characteristic (ROC) curve (AUC)




| Rank | Predicted | Actual Class |
|---|---|---|
| 1 | 0.95 | *yes* |
| 2 | 0.93 | *yes* |
| 3 | 0.93 | *no* |
| 4 | 0.88 | *yes* |
| 5 | 0.86 | *yes* |
| 6 | 0.85 | *yes* |
| 7 | 0.82 | *yes* |
| 8 | 0.80 | *yes* |
| 9 | 0.80 | *no* |
| 10 | 0.79 | *yes* |
| 11 | 0.77 | *no* |
| 12 | 0.76 | *yes* |
| 13 | 0.73 | *yes* |
| 14 | 0.65 | *no* |
| 15 | 0.63 | *yes* |
| 16 | 0.58 | *no* |
| 17 | 0.56 | *yes* |
| 18 | 0.49 | *no* |
| 19 | 0.48 | *yes* |
| ... | ... | ... |




Continued from next week ...

# Outline

# Questions?

Stephan Sigg
stephan.sigg@cs.uni-goettingen.de

# Literature

- C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.
- R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.