DSHL Summer School 2022
Day 2: Features

Martin Antenreiter

Features

Some machine learning projects succeed and some fail.

What makes the difference?

Easily the most important factor is the features used.

Prof. Pedro Domingos

Feature Engineering

- Feature engineering is a process
  - that uses domain knowledge,
  - in order to generate features
  - with which machine learning algorithms work.

Feature engineering is important for the application of machine learning!

Feature Engineering – Conventional ML

~70% of the time is invested in feature engineering!

Machine learning:

Success depends on how you prepare the data!

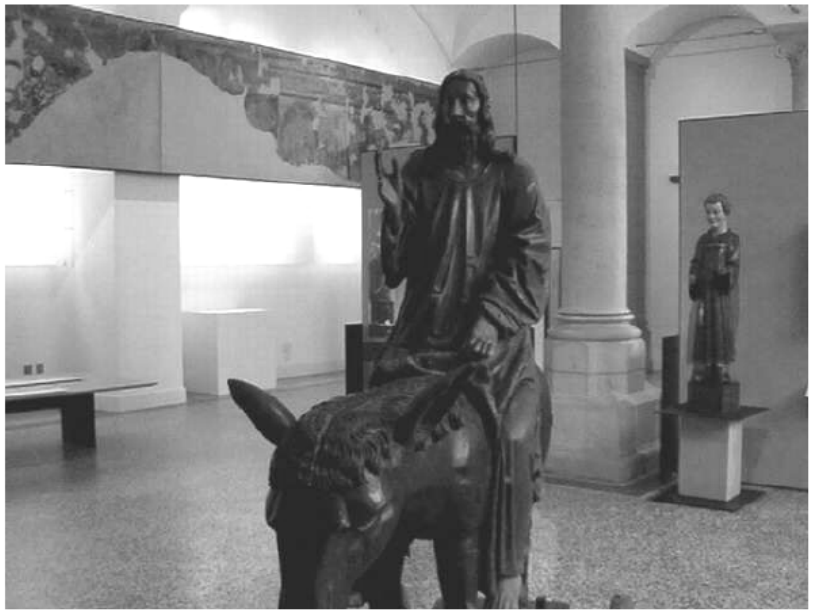
Example: Image Processing

- Output data very different
  =>content very similar

Image Processing

- Differences
  - Shooting point
  - Color temperature / illumination

Image Processing - Normalization



- Color Temperature / Illumination
  - Normalization
  - Discard color information

Image Processing Library

- OpenCV is a fast C/C++ library

  It has a python interface!

- Drawback
  - Interfaces change over time!
  - The installation procedure is sometimes difficult.
  - Therfore, define the exact library version!

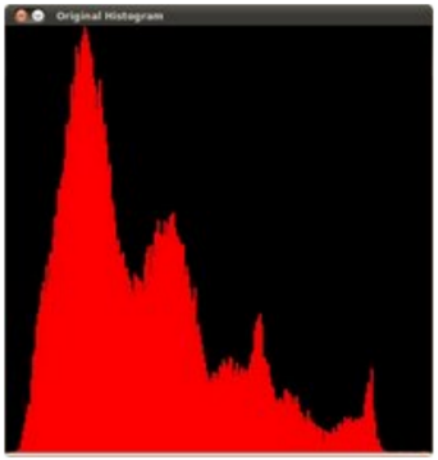
Example: Histogram Equalization

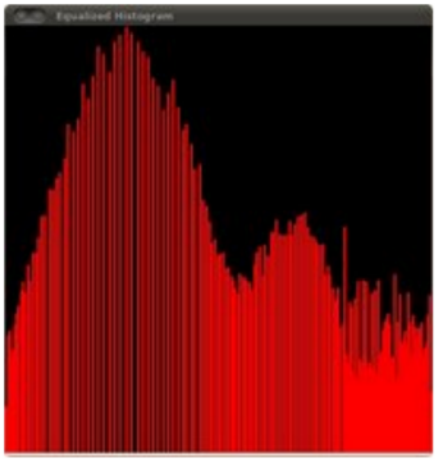
```
import cv2 as cv

filename = 'myimage.jpg'
src = cv.imread(filename)
if src is None:
    print(f'Could not open image: {filename}!')
    exit(0)
src = cv.cvtColor(src, cv.COLOR_BGR2GRAY)
dst = cv.equalizeHist(src)
```

OpenCV: equalizeHist

Image has more contrast



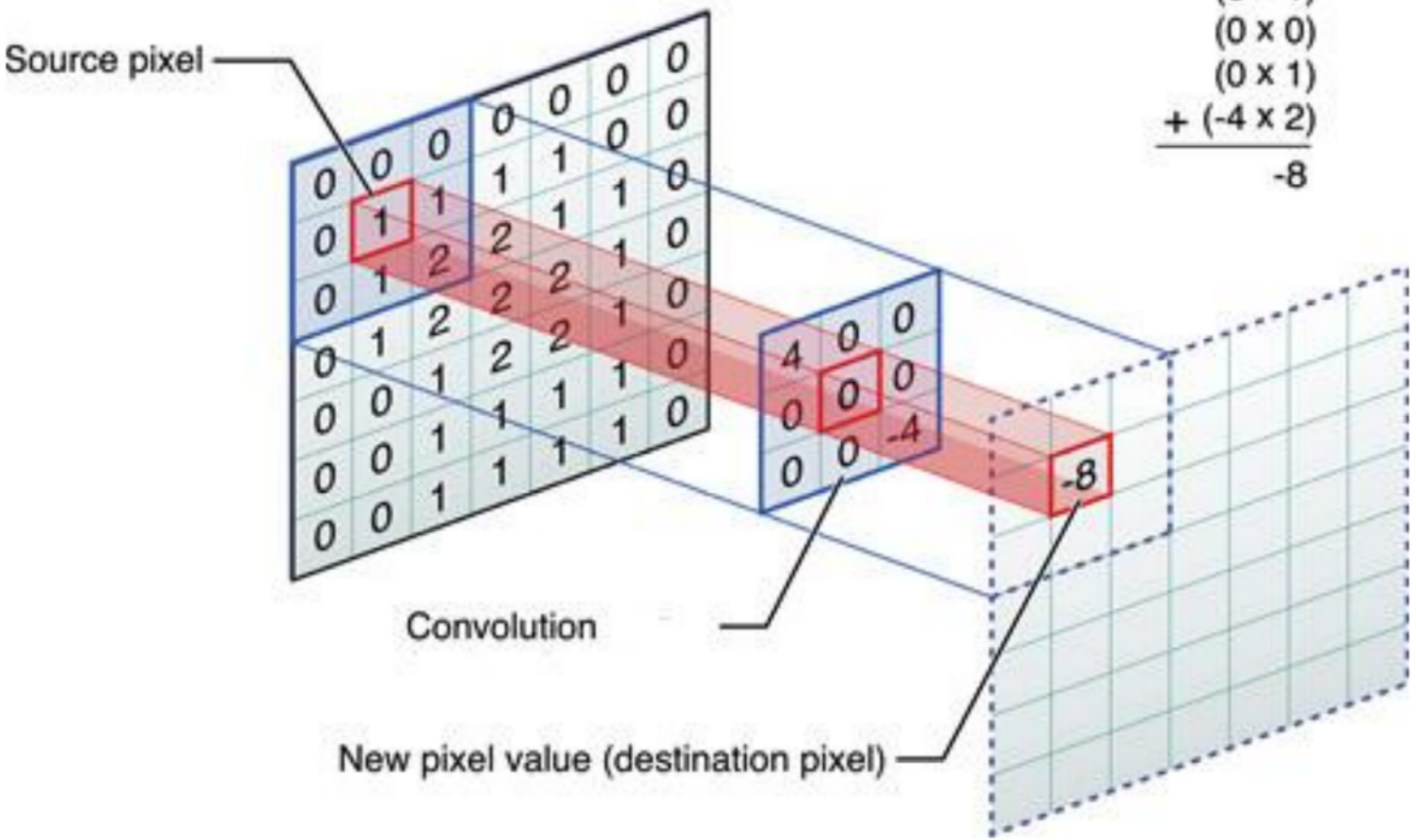


Histograms

2D Convolution

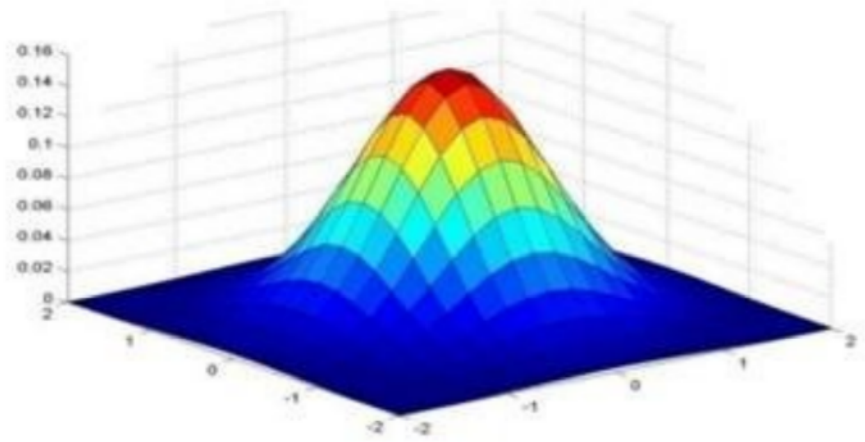
Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

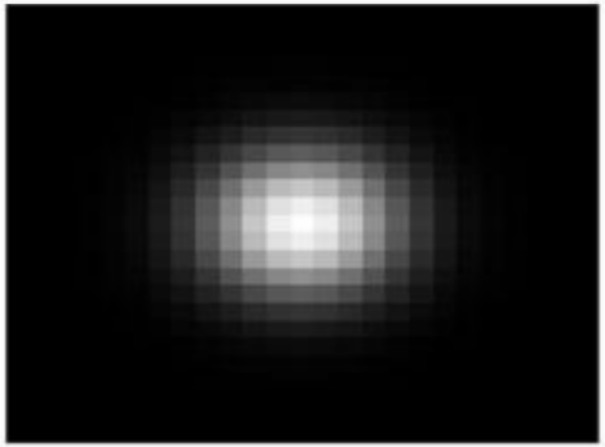
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8



Gaussian Filter

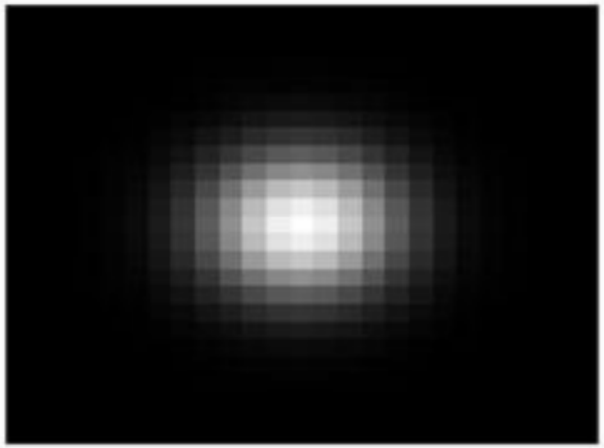
- A visualization of the weights generated by a 2D Gaussian distribution





Gaussian Filter Size

- Filter (kernel) size 3x3, 5x5, 7x7,...



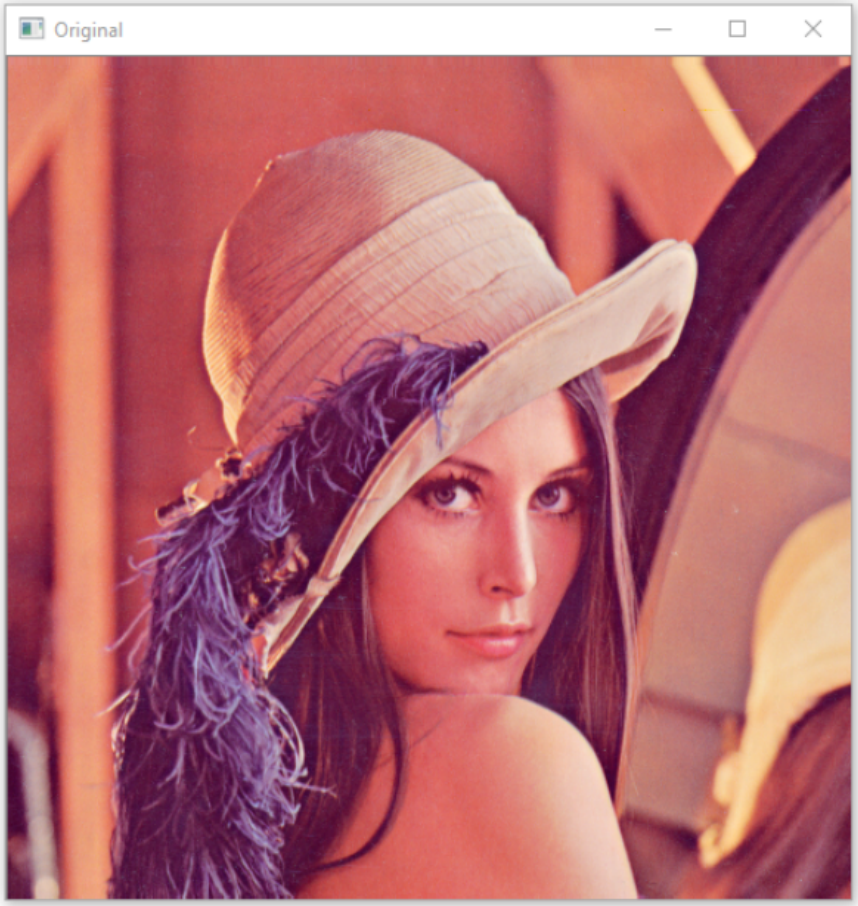
$$\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

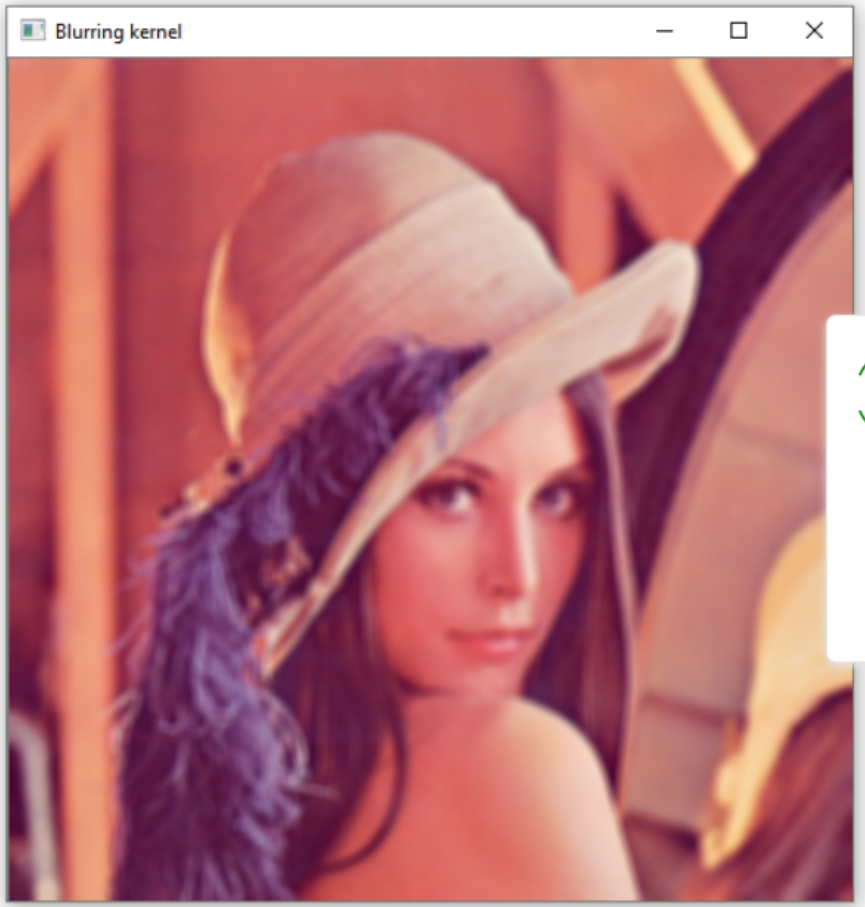
Or

$$\frac{1}{256}\begin{bmatrix}1 & 4 & 6 & 4 & 1\\ 4 & 16 & 24 & 16 & 4\\ 6 & 24 & 36 & 24 & 6\\ 4 & 16 & 24 & 16 & 4\\ 1 & 4 & 6 & 4 & 1\end{bmatrix}$$

Example: Gaussian Filter

- Blurring image with a filter size 7x7





Sobel Operator x-Direction

- Filter for edge detection

  x-Direction kernel

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I$$

Sobel Operator y-Direction

- Filter for edge detection

  y-Direction kernel

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I$$

Sobel Operator Gradient

- At each point of the image we calculate an approximation of the gradient

$$G = \sqrt{G_x^2 + G_y^2}$$

- Sometimes the following simpler equation is used:

$$G = |G_x| + |G_y|$$

Sobel Example

150	150	150	255	255
150	150	255	255	1
150	255	255	1	1
255	255	1	1	1
255	1	1	1	1

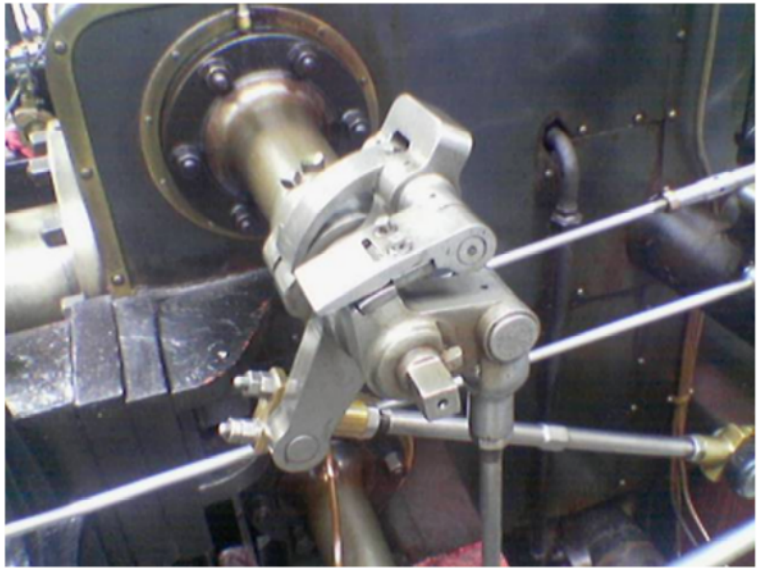
-150	-300	-150
0	0	0
150	510	255

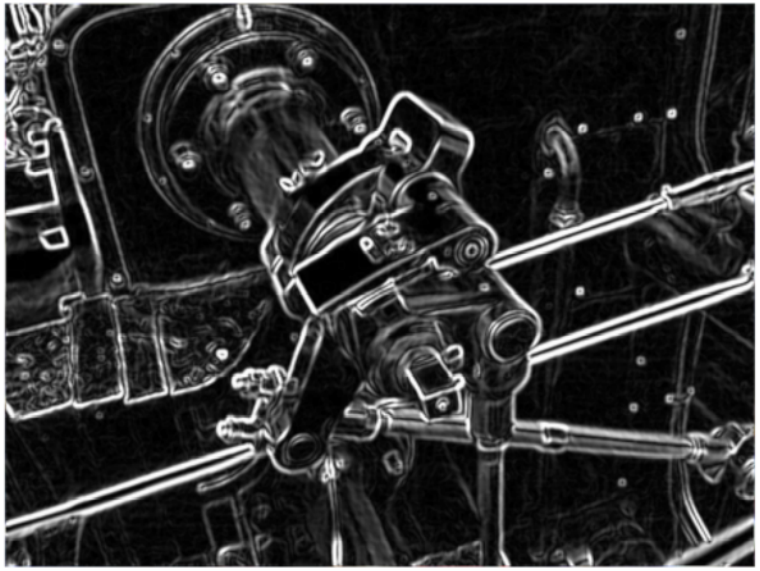
y-direction kernel

-1	-2	-1
0	0	0
1	2	1

g = -150-300-150
+150+510+255
g = 315

Sobel Result





Feature Engineering

Which filters (kernels) should we use for our problem?

Feature Engineering Loop

1) Brainstorming
   - Use knowledge about the domain!
   - What could work?
2) Create feature
3) Adjust features (thresholds)
4) Learn hypothesis
5) Performance measurement

Repeat 1-5

Feature Engineering and Error Estimation

- Feature Engineering Loop => Overfitting is possible!
- The true error can only be estimated with new unseen test data!
- Tip: Test data is from a "safe" for the final evaluation!

Feature Categorization

- Numeric variables
  - balance: 2.000€
  - weight: 4,0t
- Category variables
  - with order:
    - grade: very good > good > satisfactory
  - without order
    - gender: female / male
    - location: Graz, Leoben, Vienna

Discretization

- Input: numeric values
- Output: discrete values

  see sklearn.preprocessing.KBinsDiscretizer

- Split strategies
  - Distribution over range, each range is of equal length
  - Distribution using number of items per bin
  - Distribution using label information

Discrete Values Converted to Numeric Values

- Input type: Category
- Output: For each category a numerical value

Location
Graz
Leoben
Wien

Location
1
2
3

- Problem
  - Values specify an order!!

Discrete Values Converted to Numeric Values (2)

Location
Graz
Leoben
Wien

Location
1
2
3

- Apartments are more expensive in Vienna than in Graz.
- What might a learned model predict about housing prices in Leoben?

Discrete Values and One-Hot Encoding

- Input type: Category
- Output: Separate column for each category

  see sklearn.preprocessing.OneHotEncoder

Location	Value
Graz	1
Leoben	2
Wien	3

→

o1	o2	o3
1	0	0
0	1	0
0	0	1

Multi-hot Encoding for Streets

Street
Alte Poststraße
Eggenberger Straße
Eggenberger Allee

Location	s1	s2	s3
FH	1	1	0
Siemens	0	1	0
Otto	1	0	0

Feature Scaling

- Features have different value ranges!

- Why should you scale features?
  - Algorithms do not weight features equally

    e.g. k-Nearest Neighbors with Euclidean distance
  - Optimization algorithms converge faster (SVM, SGD,...)

Feature Scaling

- Min-Max Scaling
  (see sklearn.preprocessing.MinMaxScaler)

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

- Normalization using mean and with Min-Max

$$x' = \frac{x - \bar{x}}{\max(x) - \min(x)}$$

Feature Scaling

- Normalization using mean and standard deviation (see sklearn.preprocessing.StandardScaler)

$$x' = \frac{x - \bar{x}}{\sigma}$$

- Unit vector (see sklearn.preprocessing.normalize)

$$x' = \frac{x}{||x||}$$

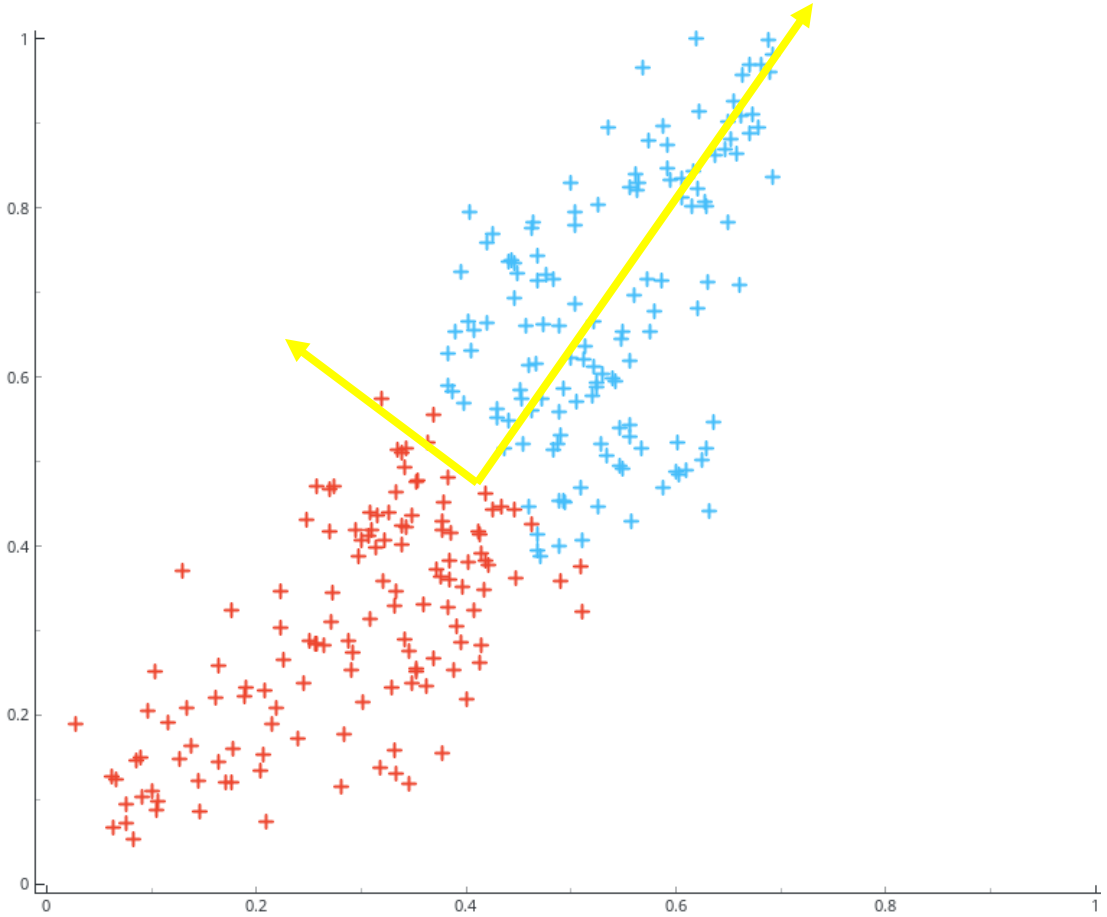
Principal Component Analysis - PCA

- Method for dimensionality reduction
- It is an unsupervised method
- Assumption of PCA:
  - The direction with the highest variance contains the most information.

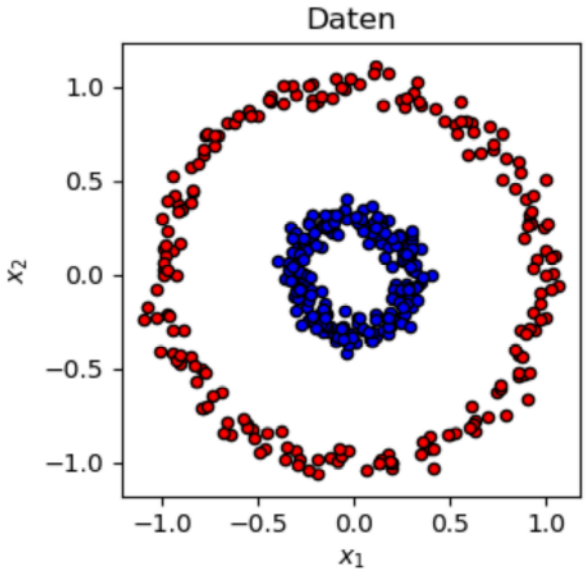
- Discard the components with low variance (=dimension reduction)
- see sklearn.decomposition.PCA

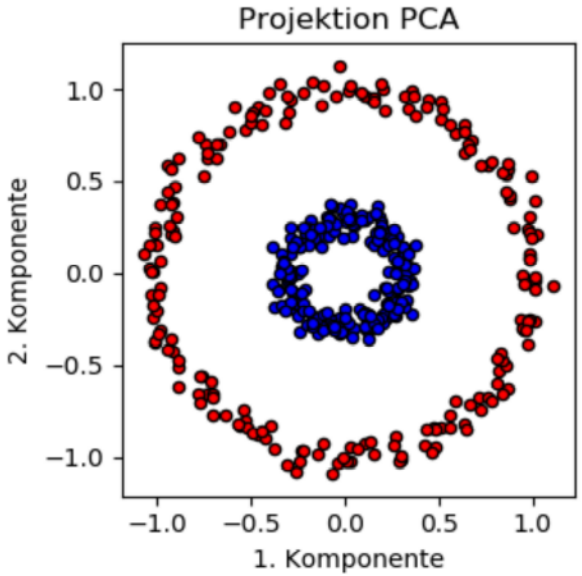
PCA Example

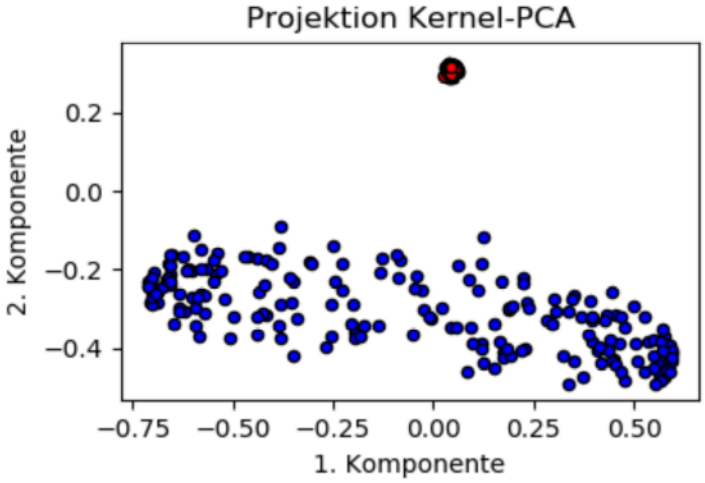
- Example



Principal Component Analysis – Kernel PCA







Missing Data

- Reasons
  - User did not want to enter data in a survey
  - Sensor defects
  - Measurement was forgotten
  - ...

Missing Data - Processing

- Delete data row
  - but data is valuable!

- You need domain knowledge which can give an insight into how to preprocess data with missing values!
- Replace missing data entries with
  - mean or median
  - delete columns only
  - interpolation between two measured values
  - prediction of missing values: use a learned model for prediction of the missing values

Incorrect Data

- Reasons:
  - No/incorrect calibration of the measuring devices
  - Excel import/export error

    e.g. csv data with , or .

- Systematic errors
  - e.g. offset at the measuring device
  - Correction possible if you recognize the error
  - Daylight saving time change!

Data Snooping

Data that influenced the learning process can no longer be used for performance measurement.

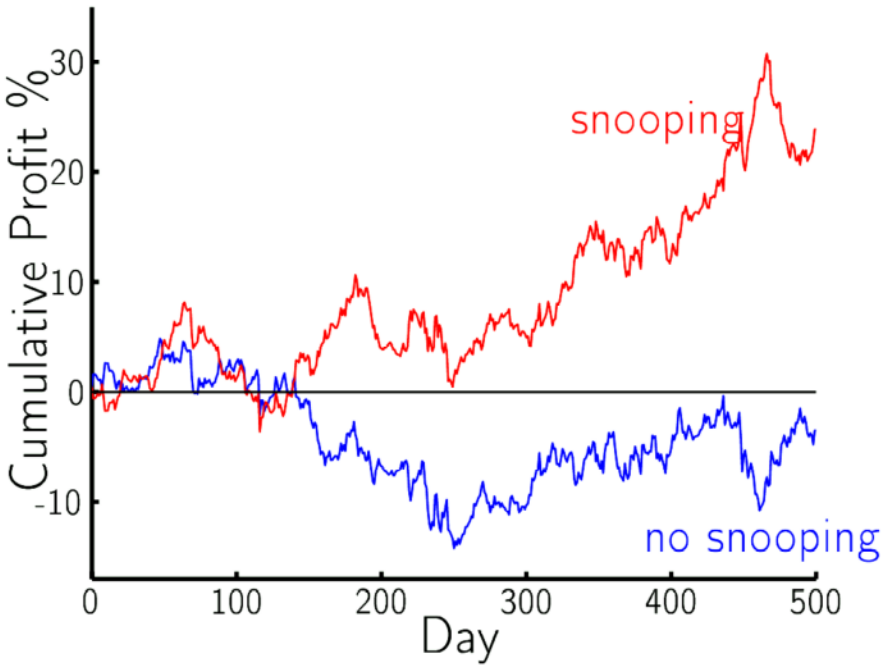
Most common mistake in practice!

Data Snooping Financial Data

- Exchange Rate Forecast
  - US Dollar vs. Pound Sterling
- Input for a forecast:
  - Exchange rate fluctuations over the last 20 days
- Output:
  - Expected exchange rate fluctuation
- Goal:
  - Profit from exchange rate fluctuations

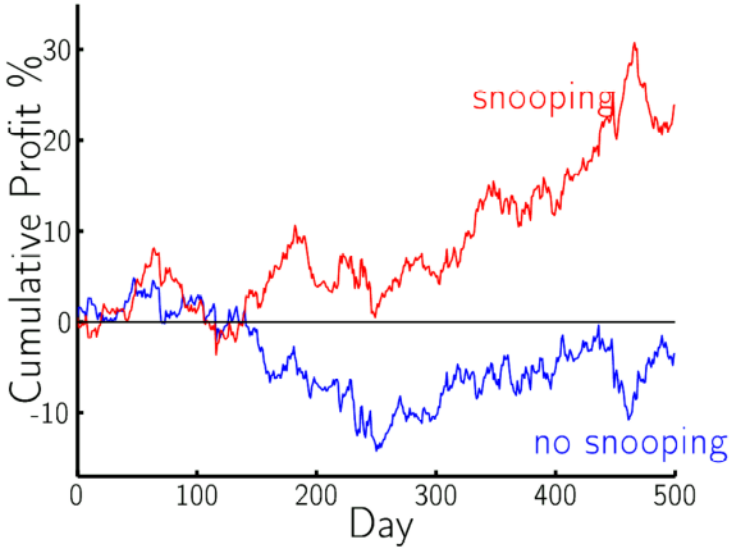
Data Snooping Financial Data - Processing

- Processing for ML
  - Normalization of the data
  - Splitting into training, validation, and test data
  - Learning
  - Evaluation



Data Snooping Pre-Processing

- Problems in the pre-processing
  - Normalization (mean value calculation, scaling)
  - Splitting into training, validation, and test data



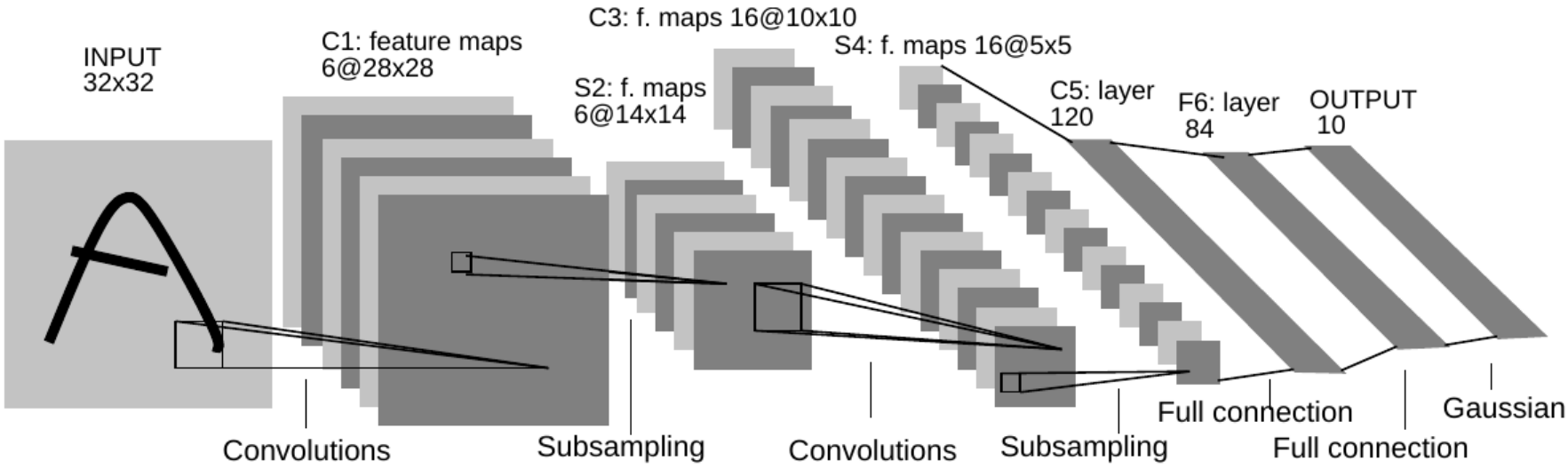
=> The mean value calculation includes future (test) values!

Feature Learning

Learning the features from the data should make "feature engineering" obsolete!

Feature Learning
CNN

- 2D Convolutional Neural Networks
  - 1989: Handwriting recognition (digits)
  - 1991: Face recognition
  - 1993: Vehicle detection
  - 2011: Object detection (GPU-based)



Feature Learning

- Feature learning works when
  - a lot of data is available
  - and enough computing capacity

- Examples:
  - Computer Vision (CNN)
  - AlphaGo Zero (CNN): Only with the rules of the game and trained by playing games against itselves!

Alpha Go Zero

Input: Image of the board