



Artificial Intelligence and Data Analytics for Engineers (AIDAE)

Lecture 5
May, 29th

Anas Abdelrazeq

Andrés Posada

Marco Kemmerling

Today's Lecturer

Vladimir Samsonov

Learning Objectives



Learning Objective w.r.t. Knowledge/Understanding. After successfully completing this lecture, the students will have achieved the following learning outcomes:

- Have an understanding what a feature is
- Know about the curse of dimensionality
- Have an understanding about the differences between feature selection and feature extraction

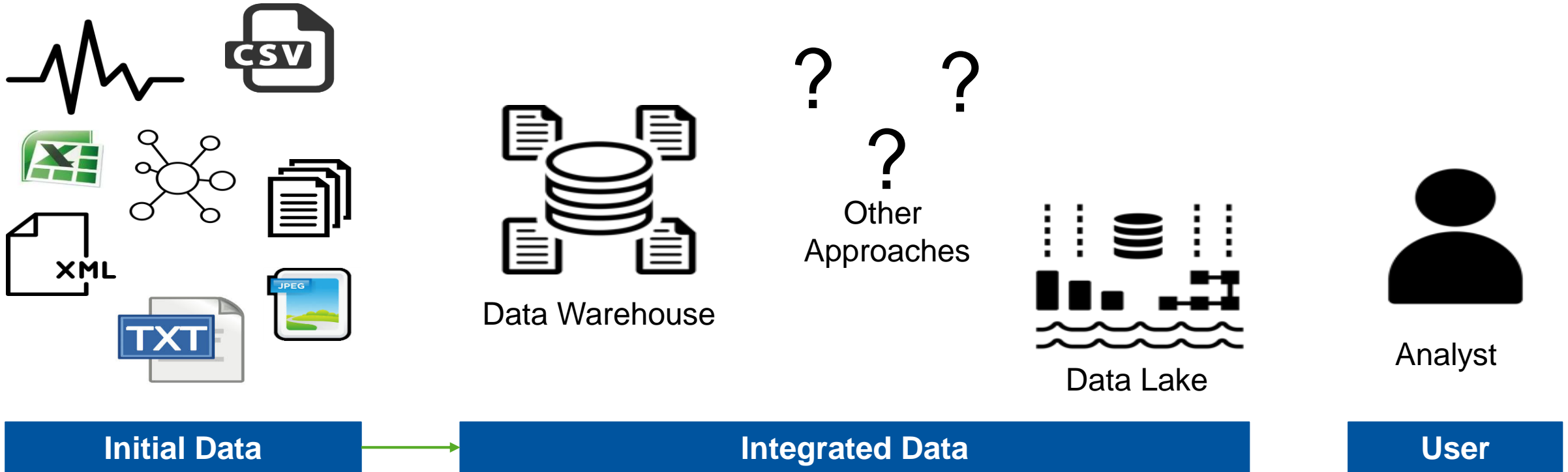
Recap Lecture 4

What is Data Integration?



Working Definition

Data **integration** is the task of combining different (possible heterogeneous) data from various sources to enable users a unified access to them (e.g. for data analysis).

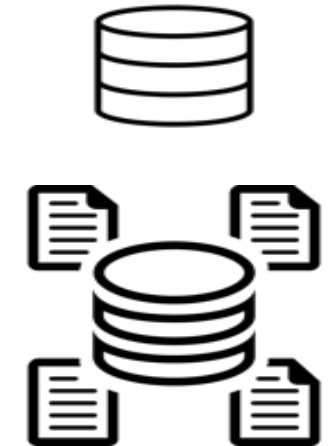
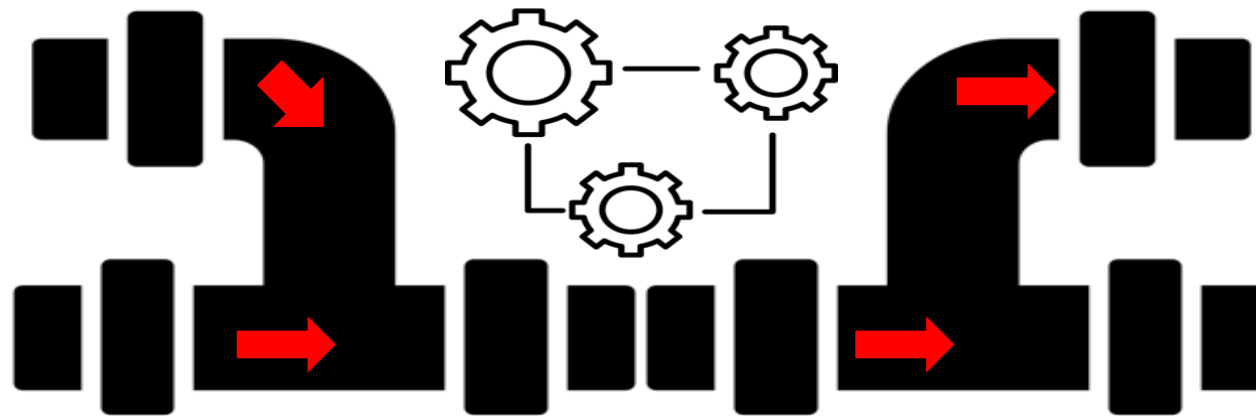


Data Pipelines



Working Definition

A data pipeline is any software system that takes data from one or more inputs and transforms it in some way before writing it to one or more outputs [Tyler Akidau, Google]



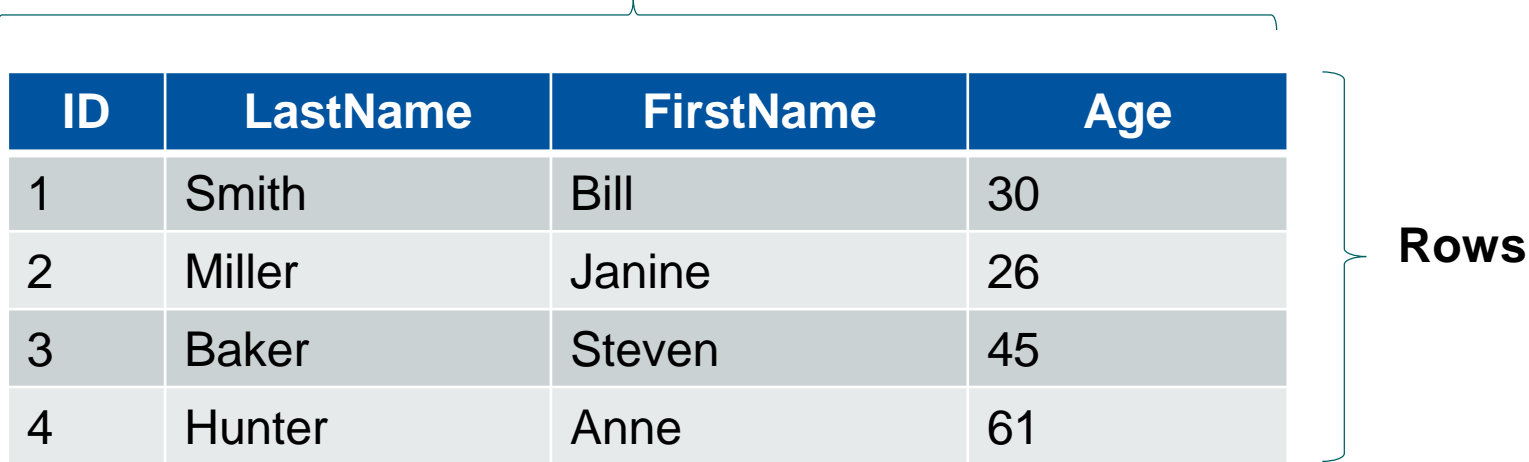
Pipelines realize integration implementation and are highly dependent on a company's IT-environment!

Storage?
Transformation?
Transportation?

Structure of Relational Databases

- Each **Database** consists of multiple **Tables**
- Example Table: Persons

Columns



ID	LastName	FirstName	Age
1	Smith	Bill	30
2	Miller	Janine	26
3	Baker	Steven	45
4	Hunter	Anne	61

Rows

Extract, Transform, Load: The ETL Process



Working Definition

ETL is a process in which data is first extracted (E) from various data sources, then transformed (T), e.g. cleaned or schema adjusted, and finally loaded (L) into the target system, e.g. a data warehouse.

Extract

- Various (heterogeneous data sources)
- Synchronously
- Asynchronously
 - Periodically
 - Event-based
 - Query-based

Transform

- Uniform schema (Schema-Mapping)
- Syntactic Transform
- Semantic Transform, e.g. duplicate removal
- Data Preprocessing, Cleansing (see lecture 3)

Load

- Overwrite
- Update
- Versioning

Companies



INFORMATICA

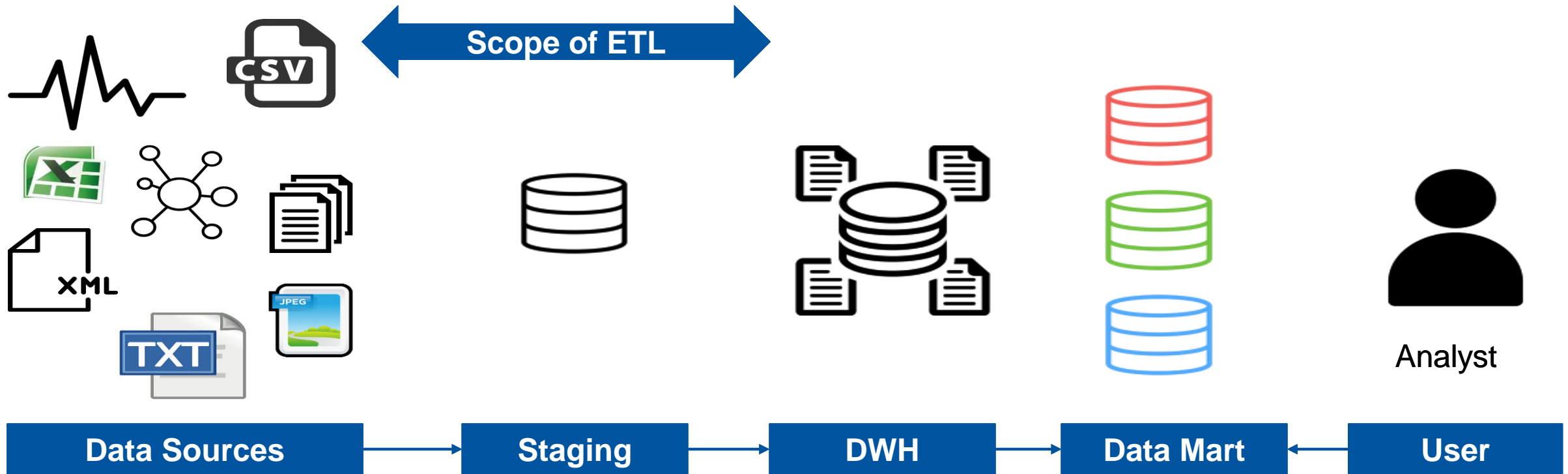


Data Warehouse



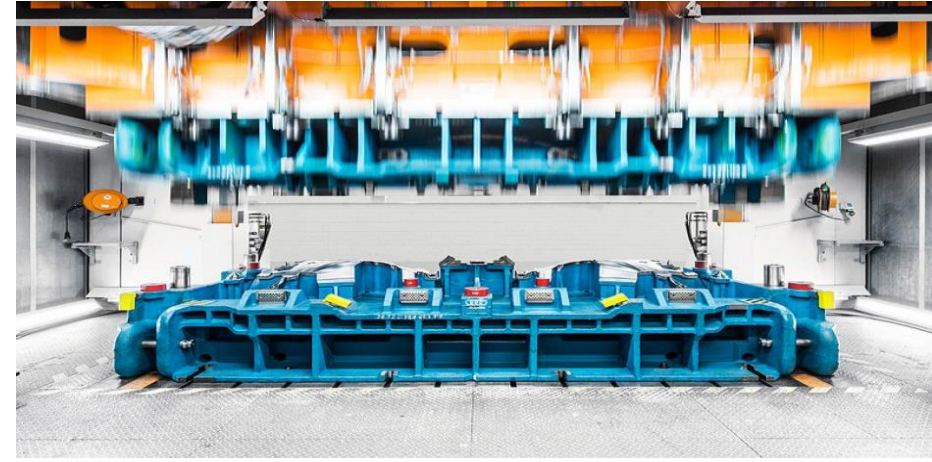
Working Definition

A data warehouse (DWH) is a central database system which integrates data from all kinds of company-wide operational data sources, e.g. production, for subsequent analysis purposes.



Motivation and Introduction

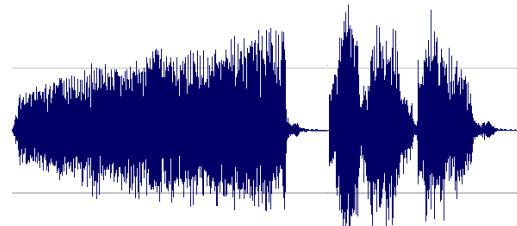
How do you represent the “real world”?



Sensors generate (numeric) measurements



Unstructured textual error reports



Sound Data



Images

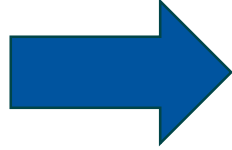
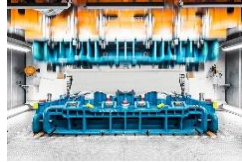
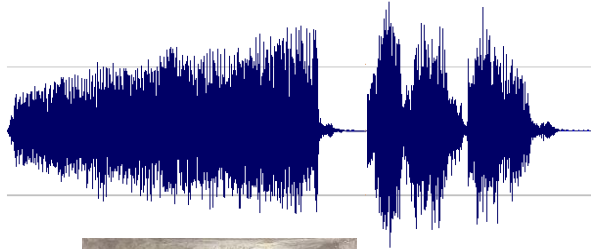


Digital Representation of the real world is “easy” ...

Remember lecture 1? There are many different ways to represent data, each of which has its own merits and drawbacks.

- **Tables:** Excel sheets, .csv-files, etc...
- **Plain text:** Word documents, etc...
- **Structured text:** .xml-files, .html-files, etc...
- **Nested objects:** .mat-files, .hdf5-files, etc...
- **Image:** .jpeg-files, .gif-files, .eps-files, .svg-files, etc...
- **Video:** .mp4-files, .avi-files, etc...
- **Audio:** .mp3-files, .wav-files, etc...
- ...

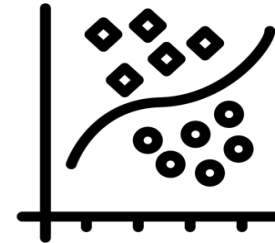
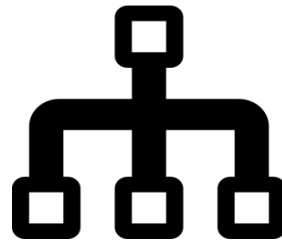
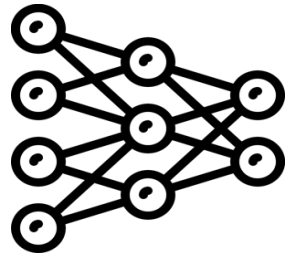
Learning Pipeline?



- **Tables:** Excel sheets, .csv-files, etc...
- **Plain text:** Word documents, etc...
- **Structured text:** .xml-files, .html-files, etc...
- **Nested objects:** .mat-files, .hdf5-files, etc...
- **Image:** .jpeg-files, .gif-files, .eps-files, .svg-files, etc...
- **Video:** .mp4-files, .avi-files, etc...
- **Audio:** .mp3-files, .wav-files, etc...
- ...



Your opinion: Good Idea?
Sneak Preview: No!

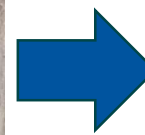


Features

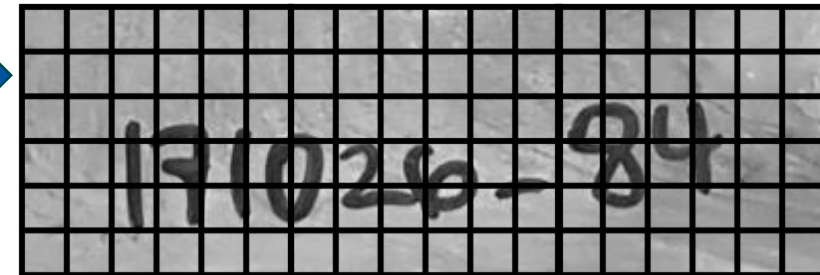


Working Definition

A feature is an individual measurable property or characteristic of a phenomenon being observed. [Christopher Bishop]



Grey-scale



Features for
pixels

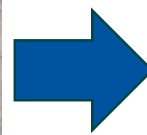


- Location of each pixel (e.g. (3,5))
- Intensity (0 (black), ..., 1 (white))
- Intensity of neighboring pixels?

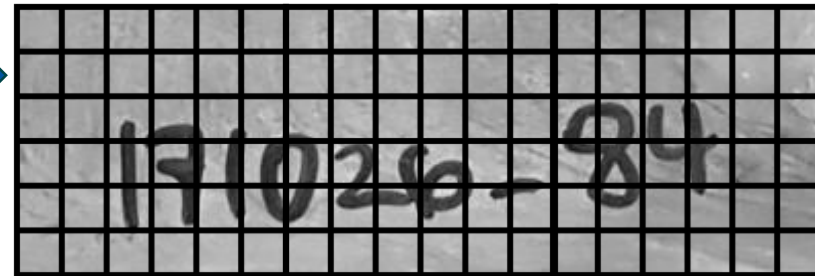


Working Definition

A feature vector is used to represent the features of an (real world) object in a mathematical form. Multiple feature vectors form the feature space.



Grey-scale

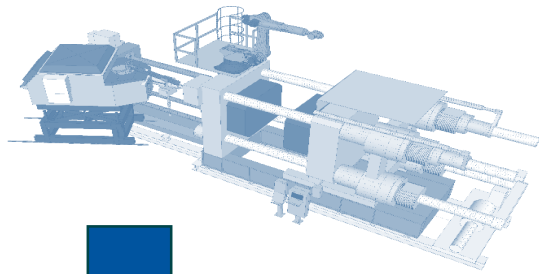


Features for
pixels



$$\vec{X} = \begin{pmatrix} 3 \\ 5 \\ 0.3 \\ 0.6 \end{pmatrix} \in \text{Feature Space}$$

Features



$$\vec{X} = \begin{pmatrix} 3 \\ 5 \\ 0.3 \\ 0.6 \\ 432 \\ 12 \end{pmatrix}$$

Feature Vector

Maybe the classes
correlate
easily with one or
two features ...

Class Labels

OK

... or they are a complex
functions of lots of
features

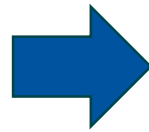
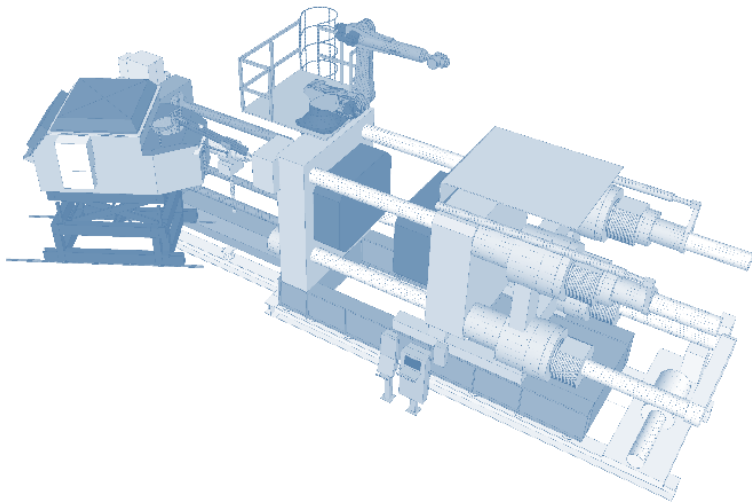
FAIL



Working Definition

Feature extraction is the process of deriving features from an real world object. The features should both be **informative** as well as **non redundant**.

So, what are good features for a given domain, e.g. high-pressure die casting? Let's say we want to predict the quality of the final product.



That highly depends on the task at hand!

Raw data from the casting process yields roughly 20 features: Start process [timestamp], end process [timestamp], pressure [bar], closing force [kN], plunger speed [m/s] ...

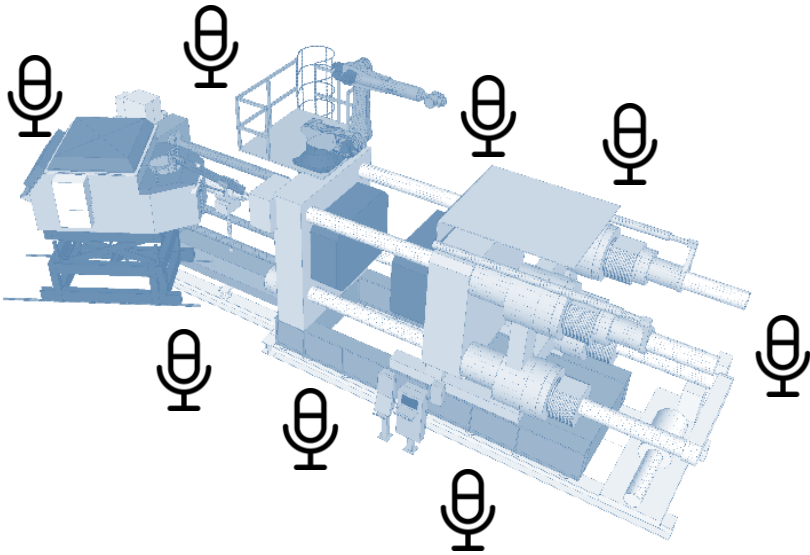
Easy to represent as a numeric vector!



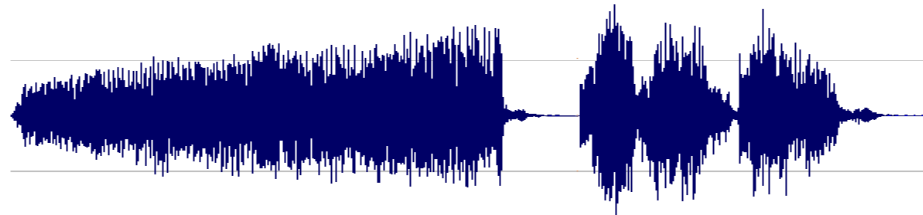
Working Definition

Feature extraction is the process of deriving features from an real world object. The features should both be informative as well as non redundant.

So, what are good features for a given domain, e.g. high-pressure die casting? Let's say we want to predict the quality of the final product.



... or you equip the machine with micro-phones.



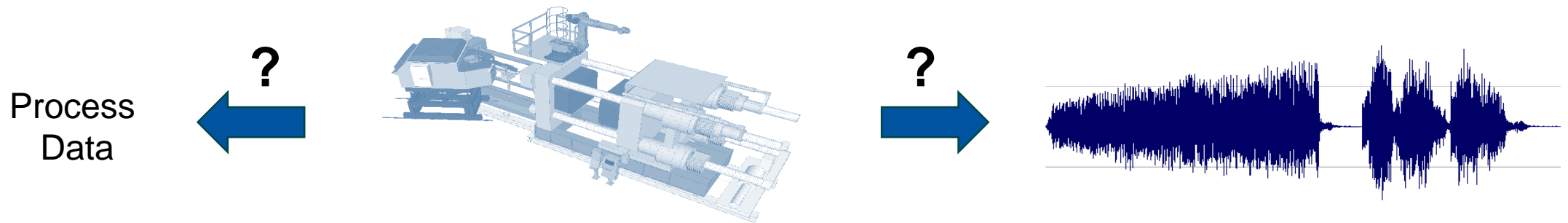
Energy, intensity, extreme values, means, peaks, zero crossings ...



Working Definition

Feature extraction is the process of deriving features from an real world object. The features should both be informative as well as non redundant.

So, what are good features for a given domain, e.g. high-pressure die casting? Let's say we want to predict the quality of the final product.



Working Definition

Feature engineering is the task of creating features for the task at hand using domain knowledge.

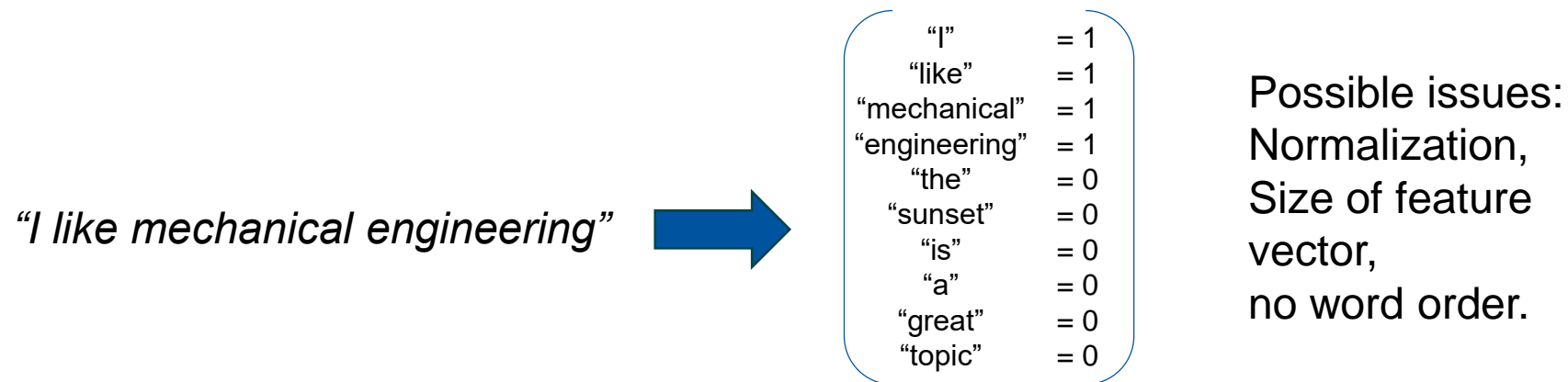
Bag-of-Words

How do you extract features of a text? For example:

“I like mechanical engineering”, “I like the sunset”, “Mechanical engineering is a great topic”

Method:

1. Define a vocabulary of words, e.g. a subset of the English language (texts in a given (training) corpus).
2. Count the number of occurrences of every word in the vocabulary with respect to the text. That yields a vector resembling a histogram.



Question: What features would you pick for image data of digits?



0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9

Have a look at the MNIST
data set for digit recognition:

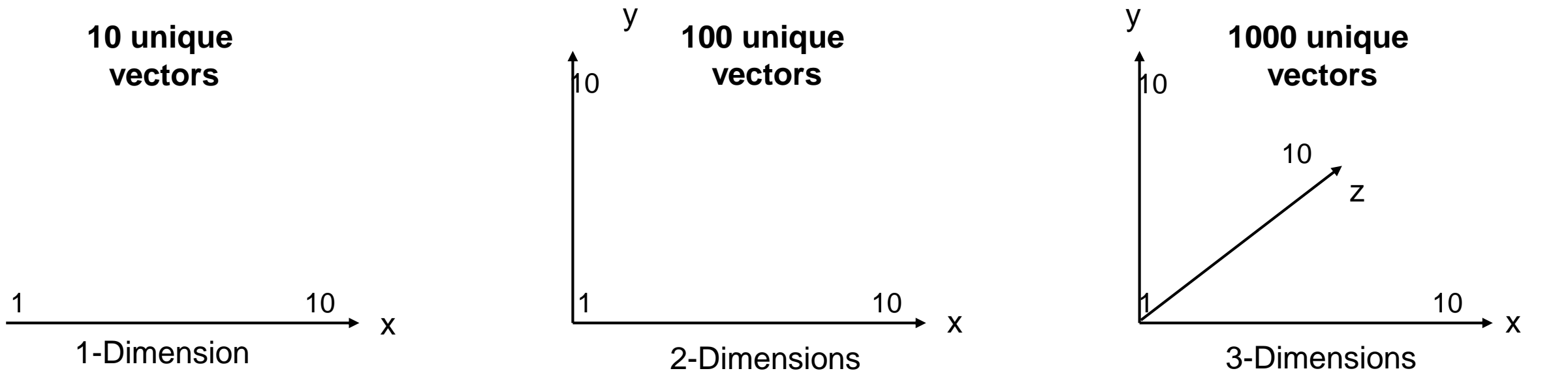
<https://www.kaggle.com/c/digit-recognizer>

Feature Selection

Would it be a good idea to generate as many features as possible given a specific task?



Curse of Dimensionality

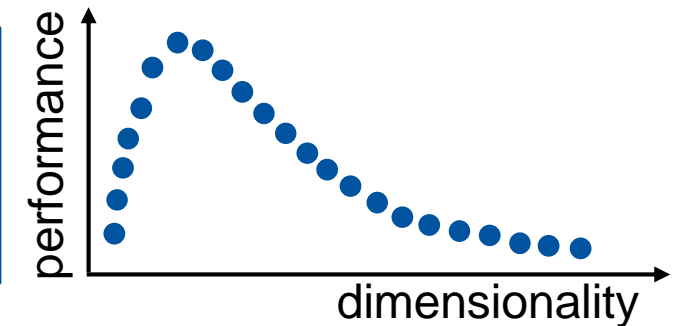


As the Dimensions increase, the feature space grows exponentially...



Working Definition

The curse of dimensionality refers to the exponential growths of the feature space with every dimension/feature added, while the number of samples decreases



Feature Selection



Working Definition

Feature selection is the process of picking only a subset of the features and not the whole set. Some of the features might not be relevant to the task at hand while others might be redundant (e.g. highly correlated)

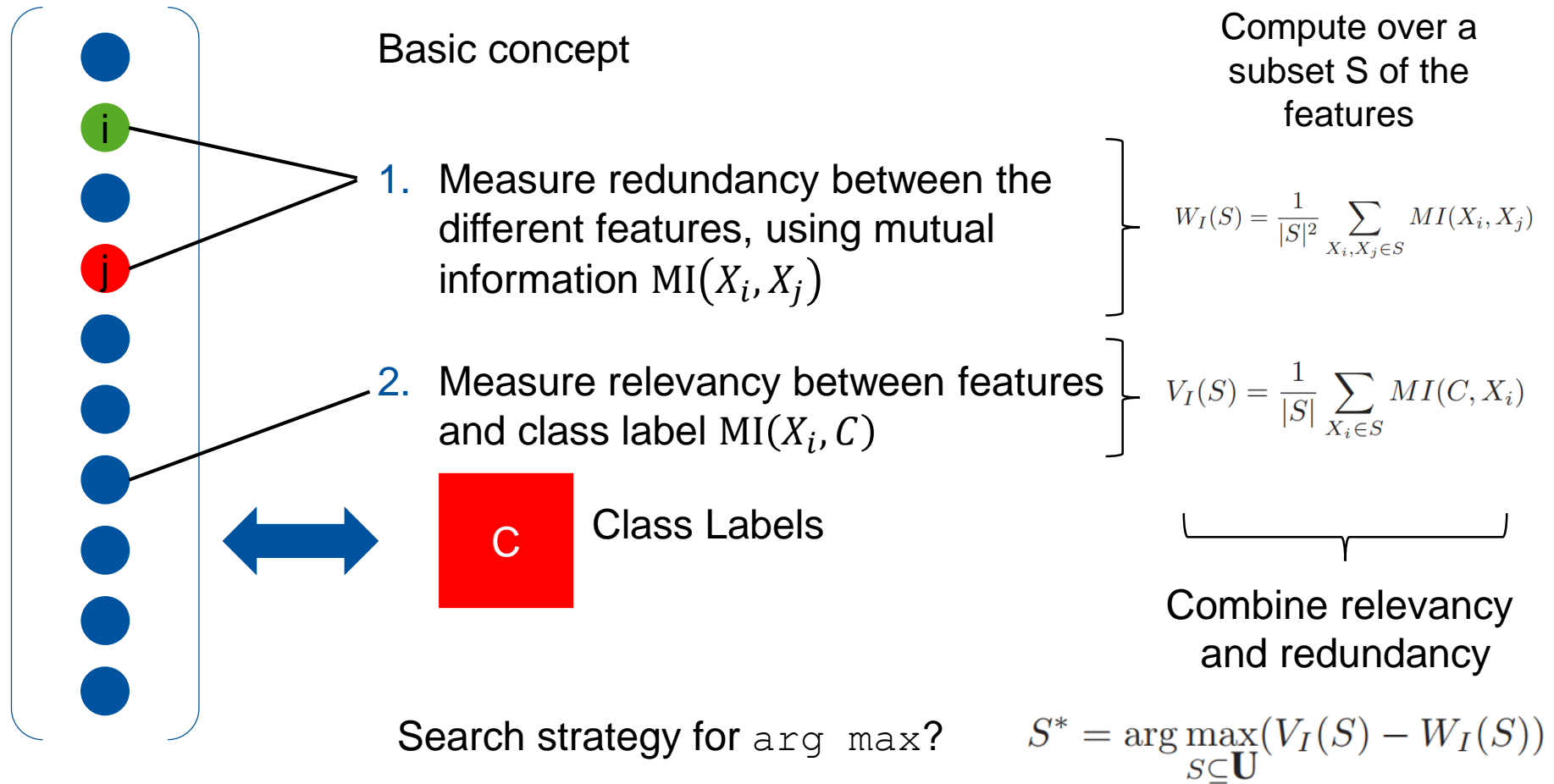
1. Moderate the curse of dimensionality
2. Reduction of training time in model generation (because of less data)
3. Generalization (features are less redundant)



Working Definition

Dimensionality reduction (feature projection) is a the process of reducing a (high) dimensional space of features in such a way, that they are represented well in a low-dimensional space

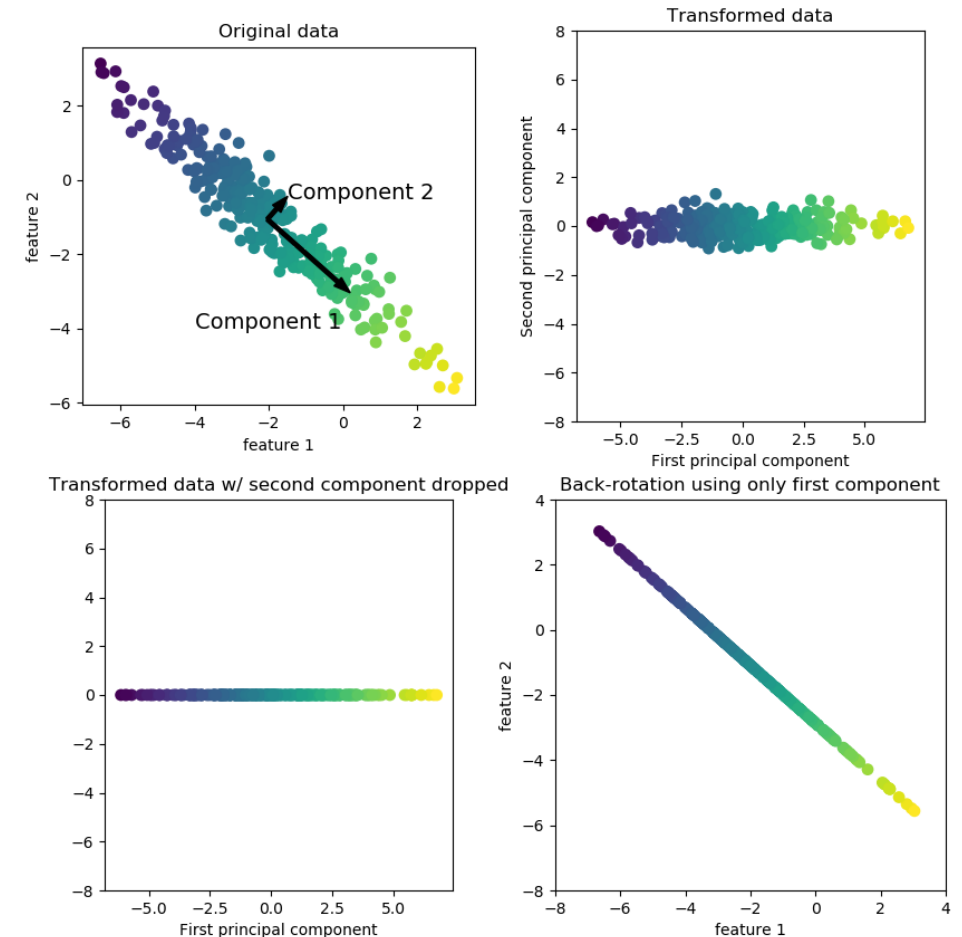
Maximum Relevancy Minimum Redundancy



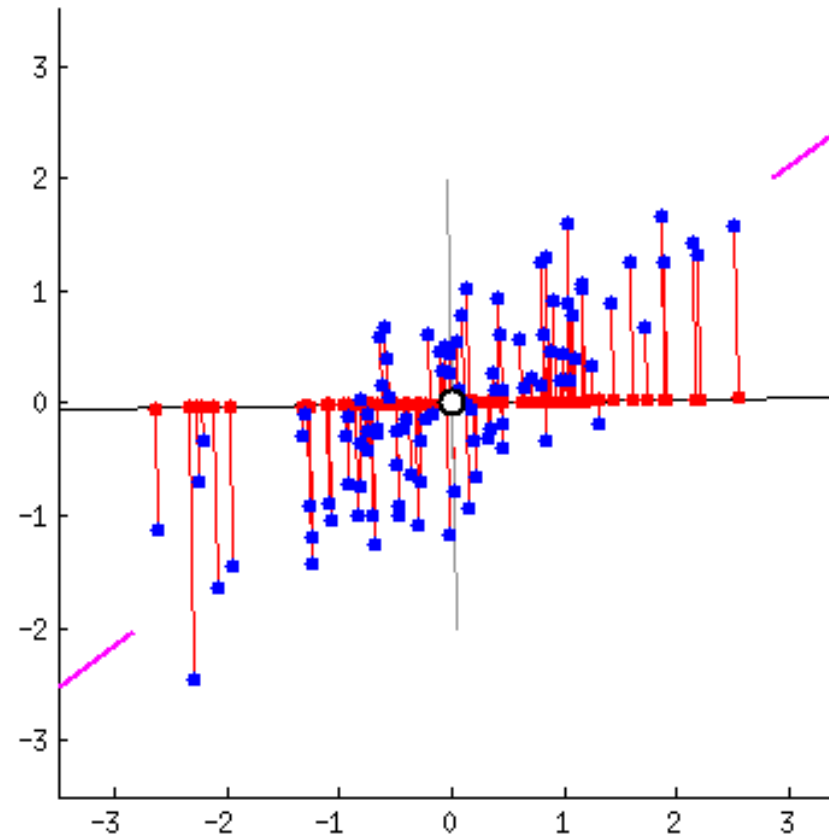
Assumption: Directions with highest variance are the most important

Basic concept:

1. Find rotation such that highest variance of data is along first axis
2. Find axis orthogonal to the first axis having second highest variance & make it the second axis
3. Rinse and repeat for all additional axes'
4. Drop axes, starting with last axis



Intuition: Finding the first principle component



Source: <https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579#140579>

What is this transformation mathematically?

- We want to transform our data into a space where coordinates are not correlated



The correlation of X and Y is defined as their covariance, normalized by the product of the standard deviations of X and Y.

Therefore: If the **correlation is zero**, then the **covariance will be zero**.

That means: We want to project our data such that the data's covariance matrix is diagonalized, w.r.t. the new base vectors.

$$\begin{pmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \cdots & \text{Cov}(X_1, X_n) \\ \text{Cov}(X_2, X_1) & \text{Var}(X_2) & \cdots & \text{Cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(X_n, X_1) & \text{Cov}(X_n, X_2) & \cdots & \text{Var}(X_n) \end{pmatrix} \xrightarrow{X \rightarrow X'} \begin{pmatrix} \text{Var}(X'_1) & 0 & \cdots & 0 \\ 0 & \text{Var}(X'_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \text{Var}(X'_n) \end{pmatrix}$$

How do we perform such a projection?



Theorem

For any symmetric matrix A there is always an orthogonal matrix S such that

$$D = S^T A S$$

where D is a diagonal matrix consisting of the eigenvalues of A :

$$\lambda_1 \dots \lambda_n$$

and S is the column-wise concatenation of the Eigenvectors of A .

Furthermore $S^{-1} = S^T$ holds.

How does that help us?

If we consider the Covariance matrix to be A , then we have a way of diagonalizing it, given that

$$\text{cov}(S^T X) = S^T \text{cov}(X) S.$$

The multiplication by S^T corresponds to a base transformation, where the corresponding space is spanned by the base vectors contained in S .

Principal Component Analysis (PCA)

Naïve implementation

1. Move the centre of your data to the origin of the coordinate system

In other words: The mean of the coordinates must be zero

Otherwise you would rotate around the wrong point

2. Calculate the covariance matrix of the translated points
3. Calculate the eigenvectors of the covariance matrix
4. Build a new matrix containing the eigenvectors as columns
Sort eigenvectors by their eigenvalues, in descending order
5. Transpose the resulting matrix and left-multiply it to your data.



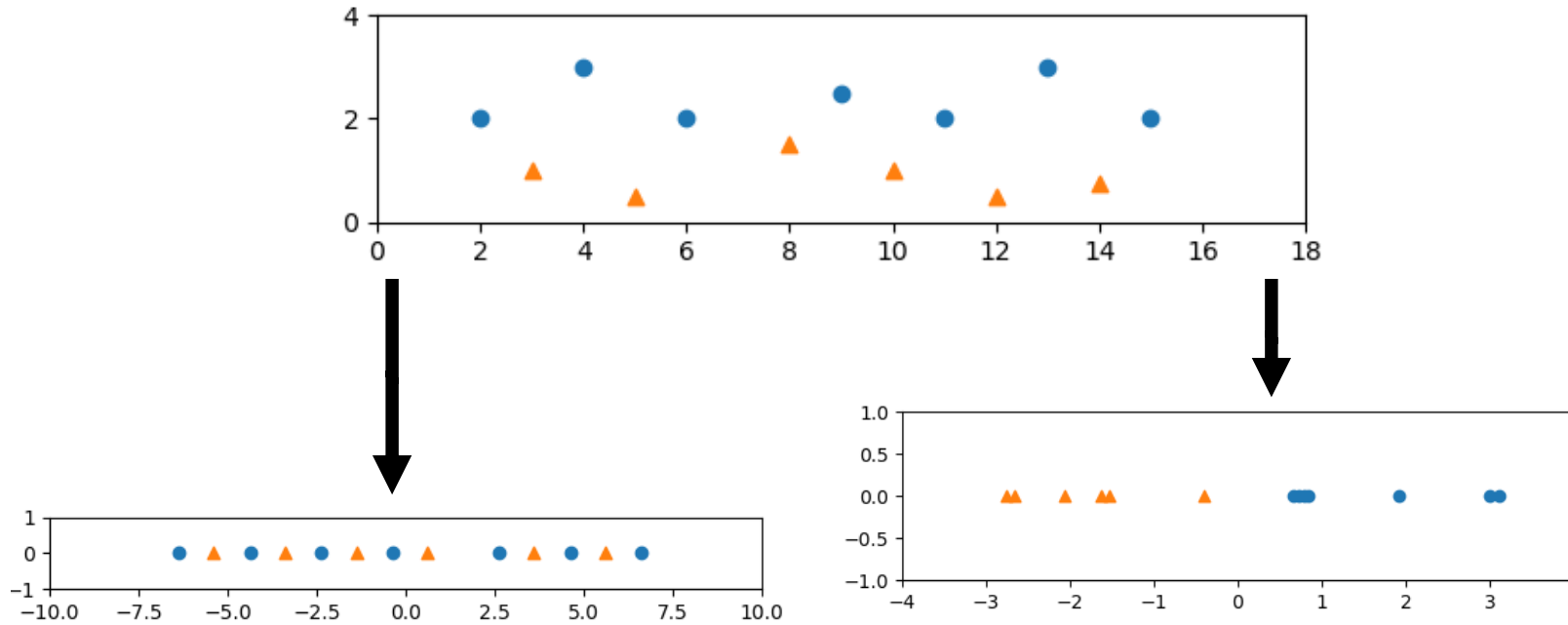
This is a bad implementation. The covariance matrix grows quadratically to the amount of dimensions of your data.

Most libraries, such as sklearn, implement optimized versions of this algorithm.

Reminder - Assumption of PCA:

“Dimensions with highest variance are the most important”

What if the variance is not correlated with the labels?



PCA Result

What we probably want

We want to optimize the representation of data w.r.t. to the labels.

Optimization goals of LDA:

Given n sets of data points belonging to n distinct labels

1. Maximize **distance between the mean coordinates** of each set after projection.
2. Minimize **scatter** within each set after projection.



Scatter

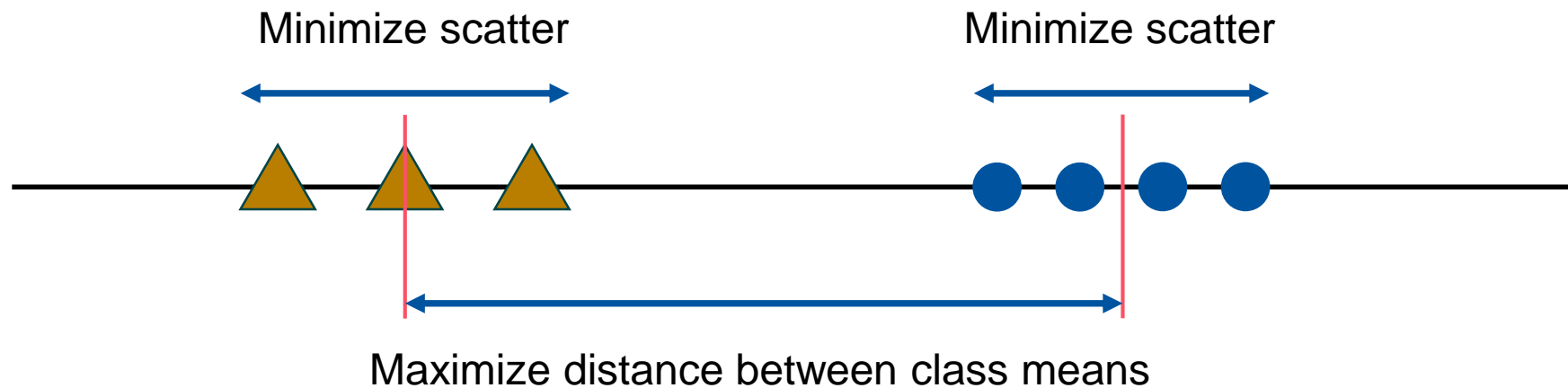
Let X_i be a set of points. Let ϕ be a projection into a less dimensional subspace. Then the scatter of X_i is defined as:

$$\tilde{s}_i^2 = \sum_{y \in \phi(X_i)} (y - \text{avg}(\phi(X_i)))^2$$

Linear Discriminant Analysis (LDA)

Optimization goals graphically

Desired result, after projection:



Linear Discriminant Analysis (LDA)

Naïve implementation

Similar to the implementation of PCA:

1. Move data to the centre of the coordinate system, similar to PCA
2. Compute scatter matrices for each class and add them to each other
3. Compute differences in means for each coordinate and multiply that with it's transposed
4. Multiply both matrices from steps 2 and 3 and solve the eigenvalue problem

We will spare you the implementation details of LDA.



Again, this is a simplistic approach. Highly optimized implementations can be found in almost any statistics library. Do not implement this yourself.

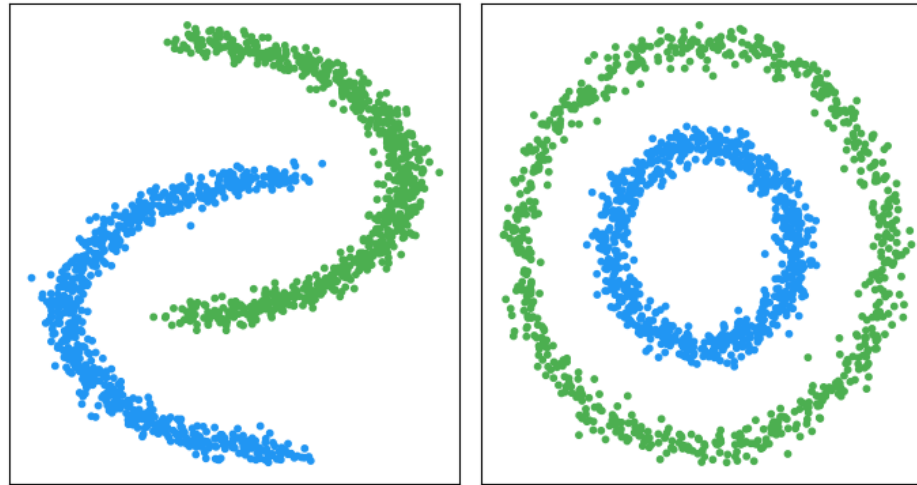
LDA and PCA in comparison

	PCA	LDA
Requirements	Unsupervised: Does not require labels	Supervised: Requires class labels
Optimization goal	Maximize variance	Minimize intra class scatter maximize difference in class means
Technique	Transformation into lower dimensional linear subspace via rotation	

LDA and PCA: Shared disadvantage

PCA and LDA possibly erase (useful) information

Especially holds true if the data cannot be accurately projected into a linear subspace, such as the following two data sets:

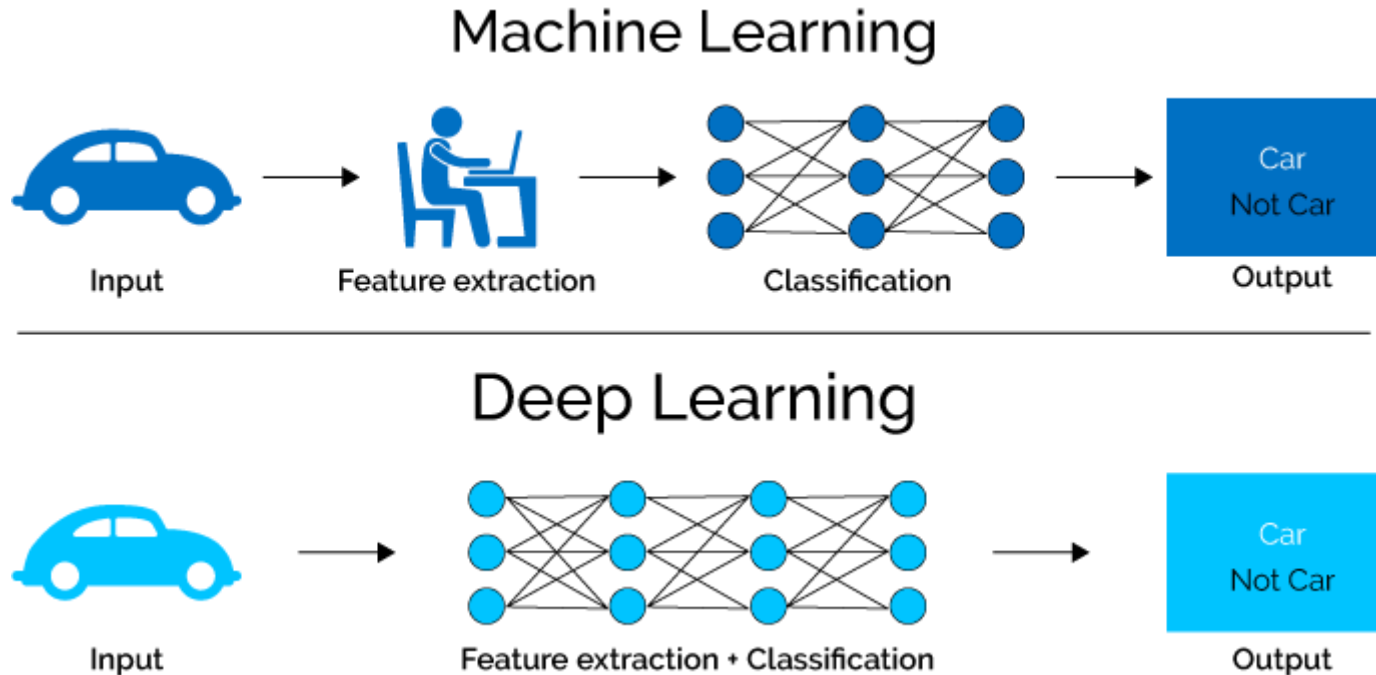


PCA and LDA are not always a good idea. If and which of the techniques can be applied depends on the data and the rest of your analysis or ML pipeline. When in doubt: Try it and compare the results.

Sneak Preview: Features & Neural Networks

Features & NN (sneak preview)

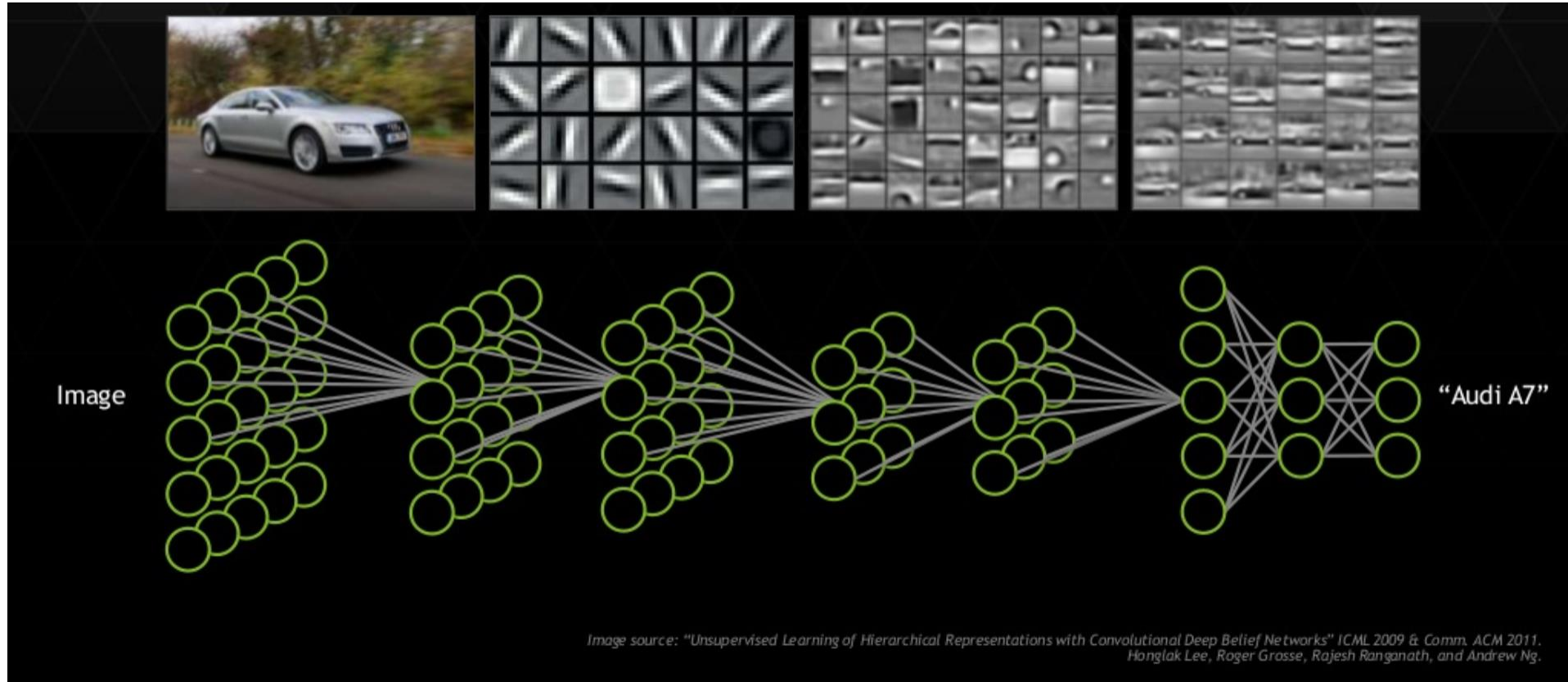
NN can use part of its layers to perform feature extraction.



- + The network will try to extract the features that are more relevant to the tasks.
- The model has to bear the extra complexity (+time +resources).
- The internal features of the NN are limited to the type of units used in the model.

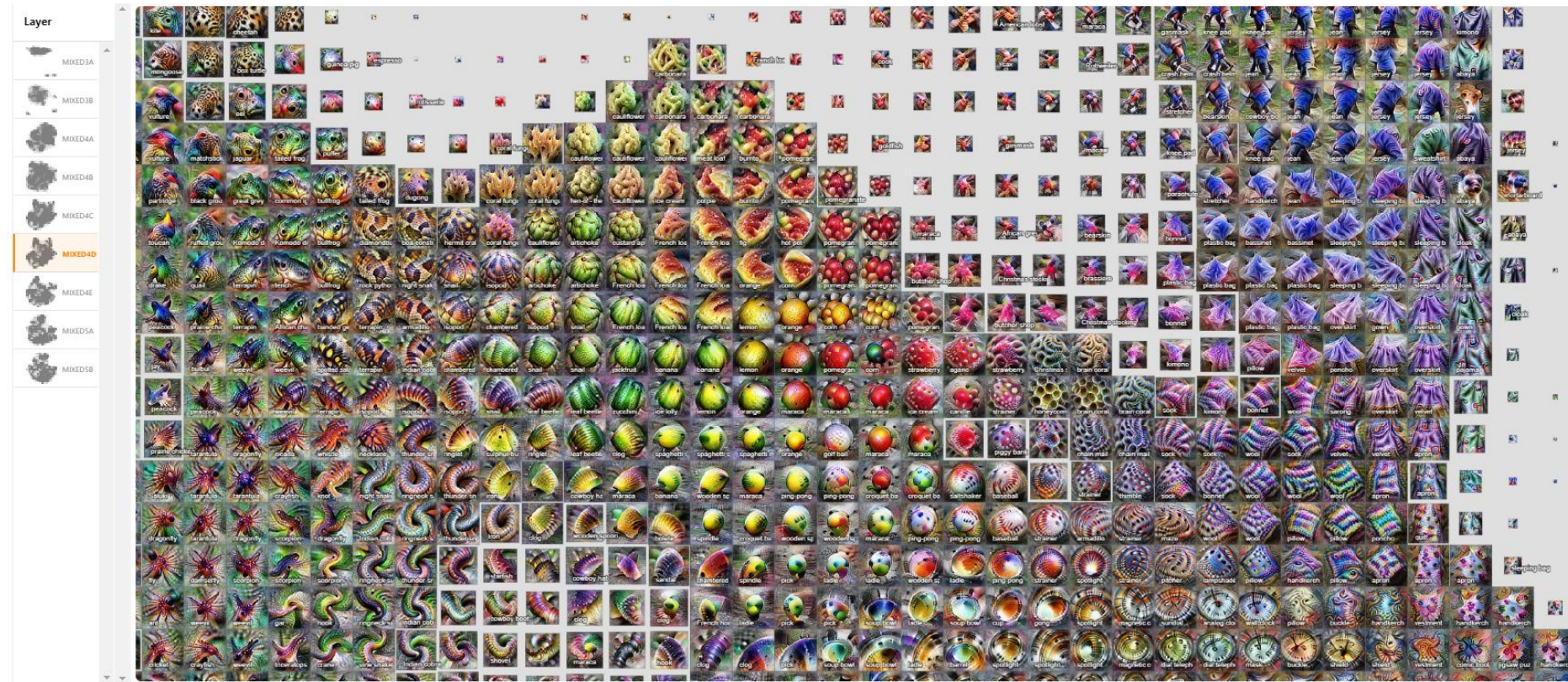
Features & NN (sneak preview)

CNN construct features from simple edges to more complex structures through their layers.



Features & NN (sneak preview)

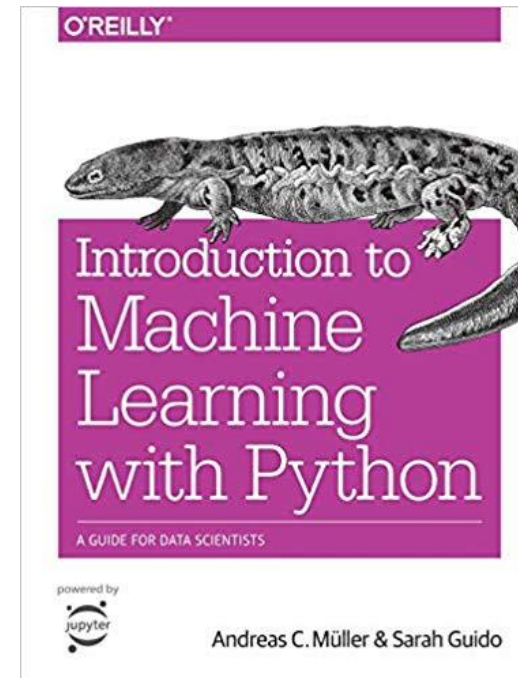
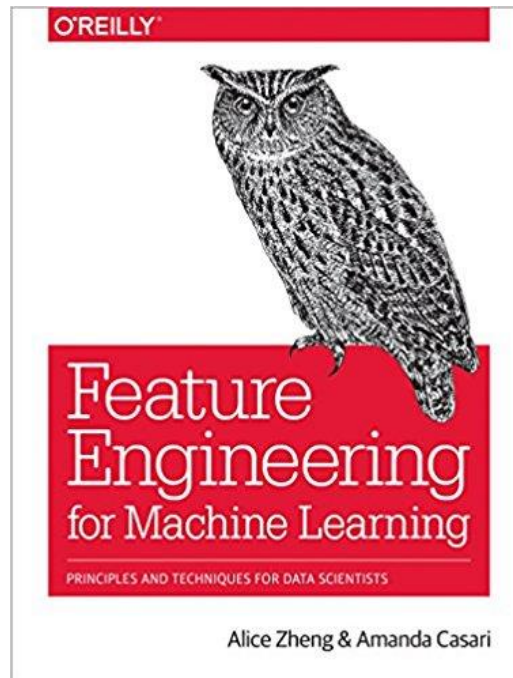
Explaining the internal features extracted by NN by exploration.



Source: <https://distill.pub/2019/activation-atlas/>

Further Reading Material

<https://www.youtube.com/watch?v=leTyvBPhYzw&t=171s> (Art of Feature Engineering for Data Science)
<https://www.youtube.com/watch?v=78RUW9kuDe4> (Practical Machine Learning: 2.3 - Feature Engineering)





Thank you for your attention!

Lecture Team AIDAE
aidae@ima-ifu.rwth-aachen.de