EUROGRAPHICS 2018

Deep Learning for Graphics

Niloy Mitra
UCL

Iasonas Kokkinos
UCL/Facebook

Paul Guerrero
UCL

Vladimir Kim
Adobe Research

Kostas Rematas
U Washington

Tobias Ritschel
UCL

UCL

facebook
Artificial Intelligence Research

Adobe®

UNIVERSITY of WASHINGTON

People

Niloy Mitra

People

Niloy Mitra

Iasonas Kokkinos

People

Niloy Mitra

Iasonas Kokkinos

Paul Guerrero

People

Niloy Mitra

Iasonas Kokkinos

Paul Guerrero

Vladimir Kim

People

Niloy Mitra

Iasonas Kokkinos

Paul Guerrero

Vladimir Kim

Kostas Rematas

People

Niloy Mitra

Iasonas Kokkinos

Paul Guerrero

Vladimir Kim

Kostas Rematas

Tobias Ritschel

Course Overview

Course Overview

- Part I: Introduction and ML Basics

Course Overview

- Part I: Introduction and ML Basics

- Part II: Supervised Neural Networks: Theory and Applications

Course Overview

- Part I: Introduction and ML Basics
- Part II: Supervised Neural Networks: Theory and Applications
- Part III: Unsupervised Neural Networks: Theory and Applications

Course Overview

- Part I: Introduction and ML Basics
- Part II: Supervised Neural Networks: Theory and Applications
- Part III: Unsupervised Neural Networks: Theory and Applications
- Part IV: Beyond Image Data

Course Overview

- Part I: Introduction and ML Basics
- Part II: Supervised Neural Networks: Theory and Applications
- Part III: Unsupervised Neural Networks: Theory and Applications
- Part IV: Beyond Image Data

Four 1.5hr sessions with breaks as per EG timetable.

Code Examples

PCA/SVD basis

Linear Regression

Polynomial Regression

Stochastic Gradient Descent vs. Gradient Descent

Multi-layer Perceptron

Edge Filter ‘Network’

Convolutional Network

Filter Visualization

Weight Initialization Strategies

Colorization Network

Autoencoder

Variational Autoencoder

Generative Adversarial Network

http://geometry.cs.ucl.ac.uk/dl4g/

Two-way Communication

Two-way Communication

- *This tutorial is given for the first time!*

Two-way Communication

- *This tutorial is given for the first time!*
- Our aim is to convey what we found to be relevant so far.

Two-way Communication

- *This tutorial is given for the first time!*

- Our aim is to convey what we found to be relevant so far.

- You are invited/encouraged to give feedback
  - On-line form
  - Speakup. Please send us your criticism/comments/suggestions
  - Ask questions, please!

Two-way Communication

- *This tutorial is given for the first time!*

- Our aim is to convey what we found to be relevant so far.

- You are invited/encouraged to give feedback
  - On-line form
  - Speakup. Please send us your criticism/comments/suggestions
  - Ask questions, please!

- **Thanks to many people who helped so far with slides/comments.**

Course Overview

- **Part I: Introduction and ML Basics**
- Part II: Supervised Neural Networks: Theory and Applications
- Part III: Unsupervised Neural Networks: Theory and Applications
- Part IV: Beyond Image Data

Representations in CG

Representations in CG

- Images (e.g., pixel grid)

Representations in CG

- Images (e.g., pixel grid)
- Volume (e.g., voxel grid)

Representations in CG

- Images (e.g., pixel grid)
- Volume (e.g., voxel grid)
- Meshes (e.g., vertices/edges/faces)

Representations in CG

- Images (e.g., pixel grid)

- Volume (e.g., voxel grid)

- Meshes (e.g., vertices/edges/faces)

- Animation (e.g., skeletal positions over time; cloth dynamics over time)

Representations in CG

- Images (e.g., pixel grid)
- Volume (e.g., voxel grid)
- Meshes (e.g., vertices/edges/faces)
- Animation (e.g., skeletal positions over time; cloth dynamics over time)
- Pointclouds (e.g., point arrays)

Representations in CG

- Images (e.g., pixel grid)
- Volume (e.g., voxel grid)
- Meshes (e.g., vertices/edges/faces)
- Animation (e.g., skeletal positions over time; cloth dynamics over time)
- Pointclouds (e.g., point arrays)
- Physics simulations (e.g., fluid flow over space/time)

Problems in Computer Graphics

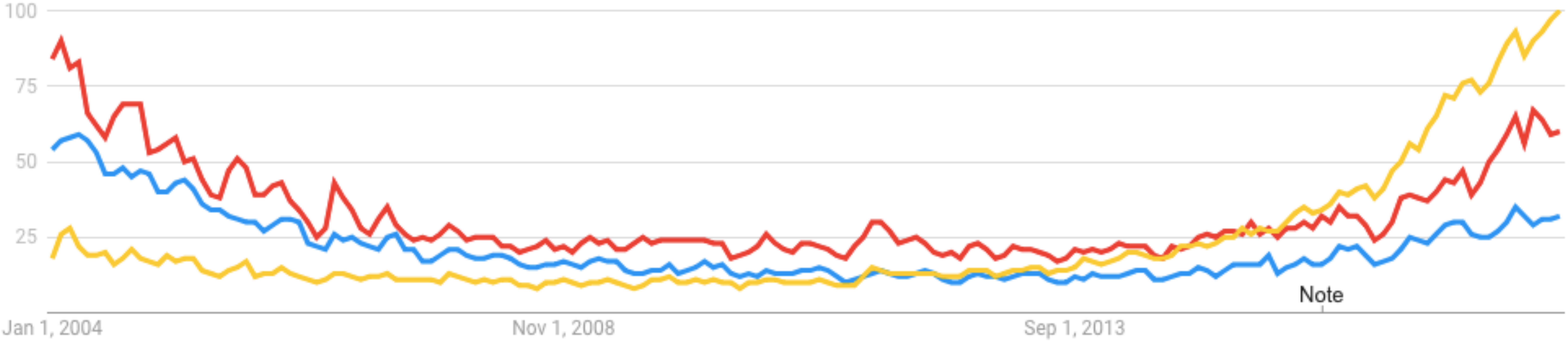
- Feature detection (image features, point features) $\mathbb{R}^{m \times m} \rightarrow \mathbb{Z}$
- Denoising, Smoothing, etc. $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$
- Embedding, Distance computation $\mathbb{R}^{m \times m, m \times m} \rightarrow \mathbb{R}^{d}$
- Rendering $\mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$
- Animation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Physical simulation $\mathbb{R}^{3m \times t} \rightarrow \mathbb{R}^{3m}$
- Generative models $\mathbb{R}^{d} \rightarrow \mathbb{R}^{m \times m}$

Rise of Machine Learning

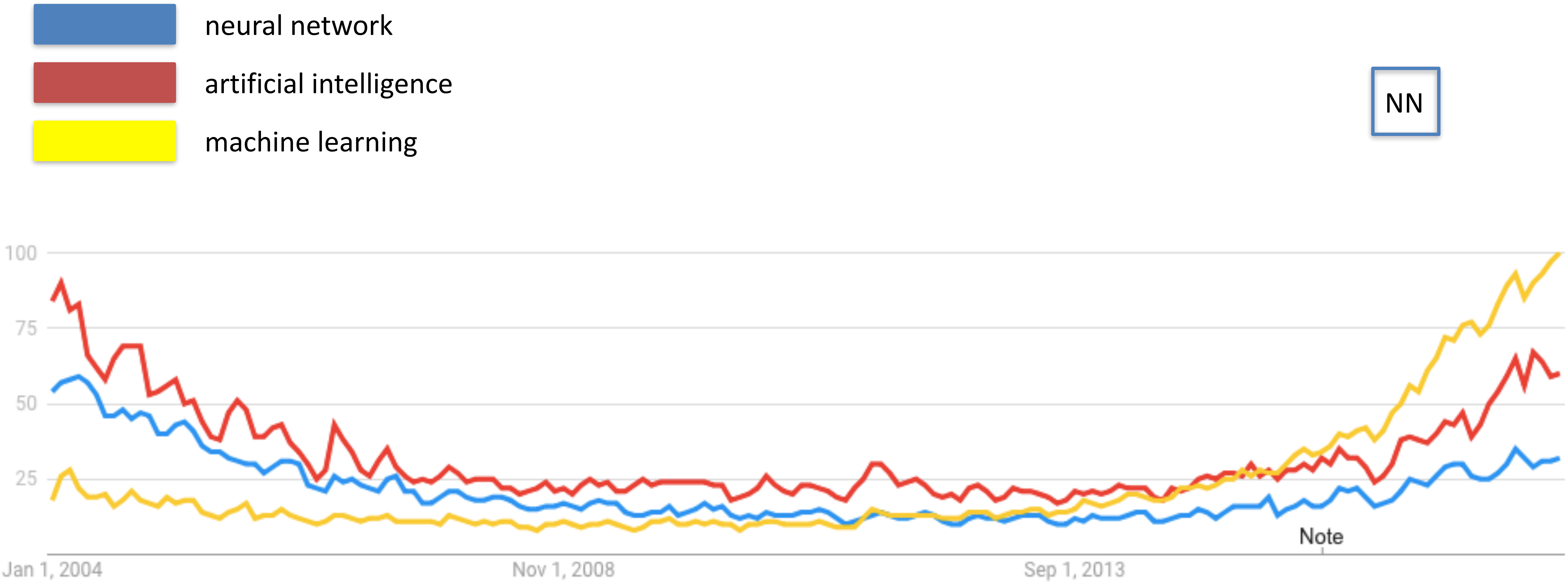
🟦 neural network

🟥 artificial intelligence

🟨 machine learning

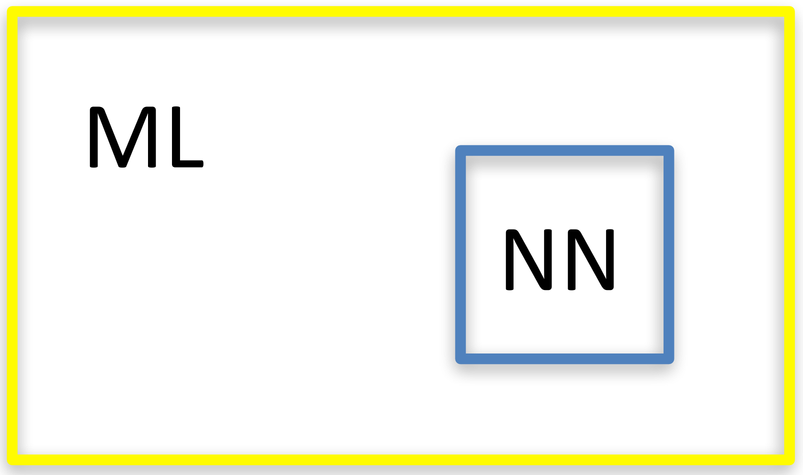


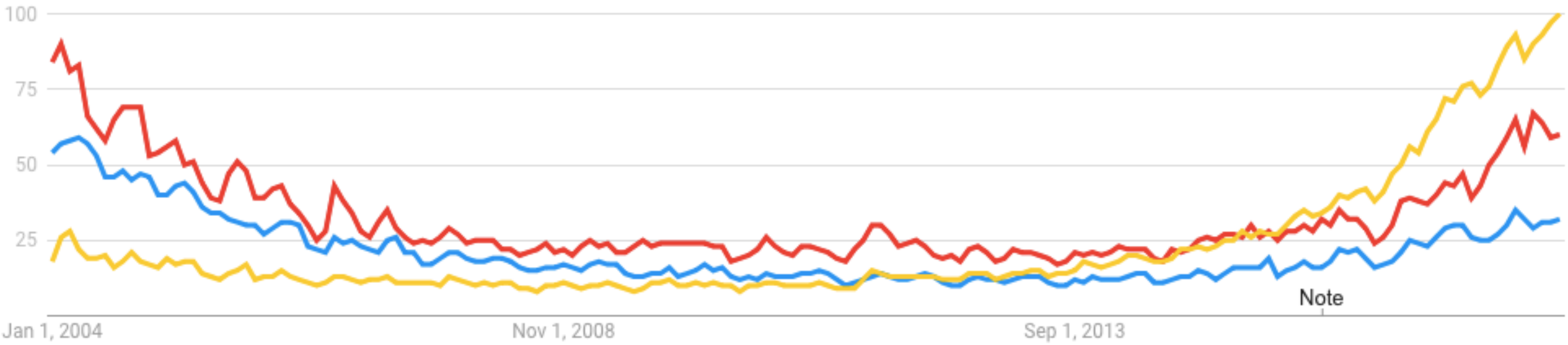
Rise of Machine Learning



Rise of Machine Learning

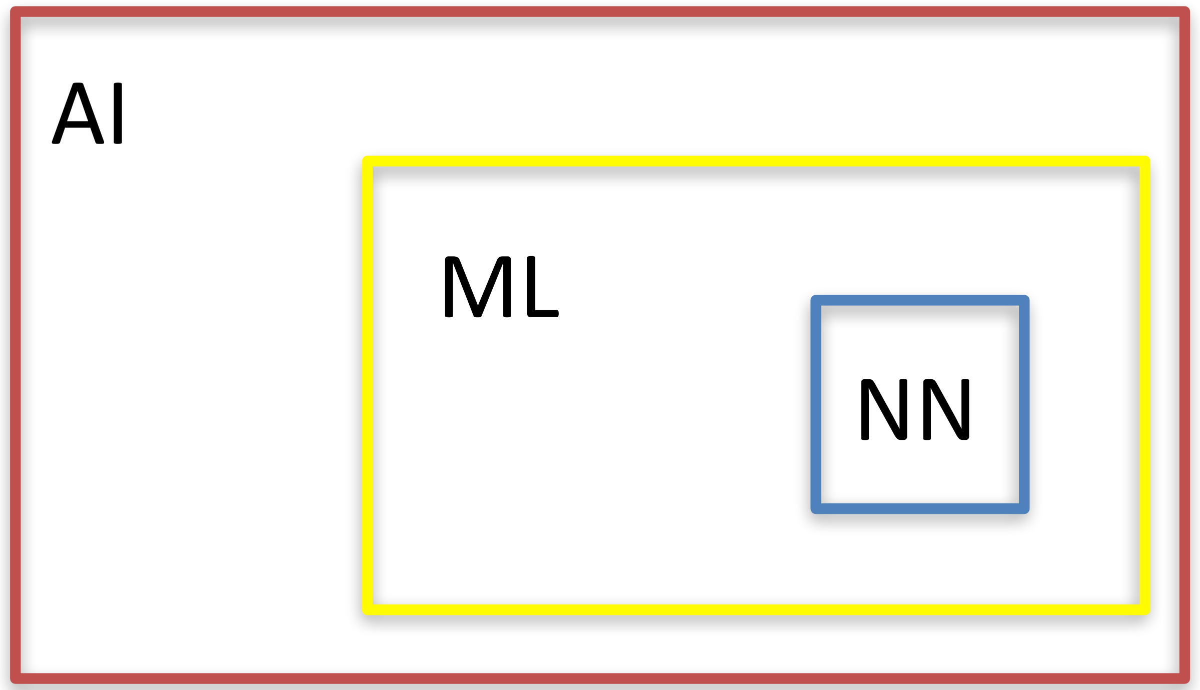
- 🟦 neural network
- 🟥 artificial intelligence
- 🟨 machine learning





Rise of Machine Learning

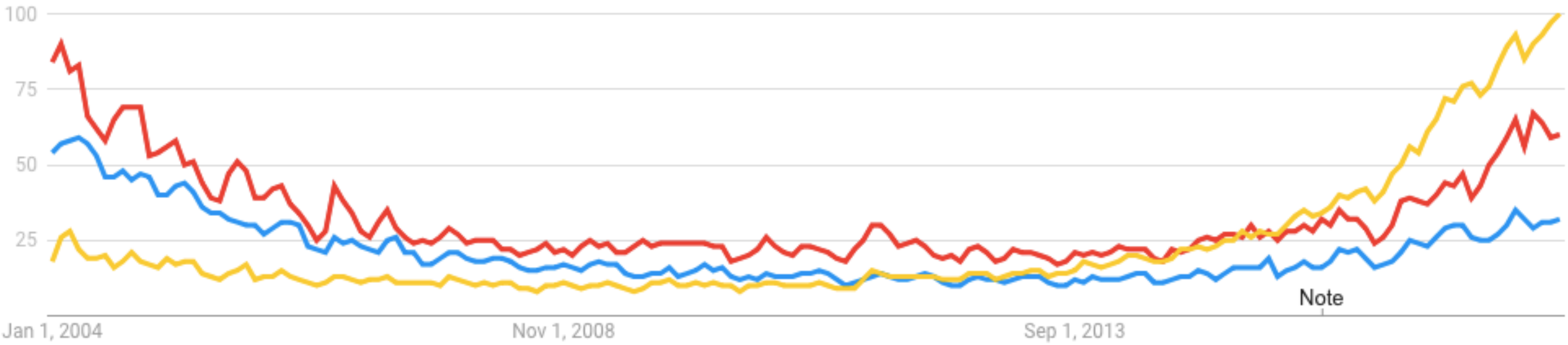
- 🟦 neural network
- 🟥 artificial intelligence
- 🟨 machine learning



AI

ML

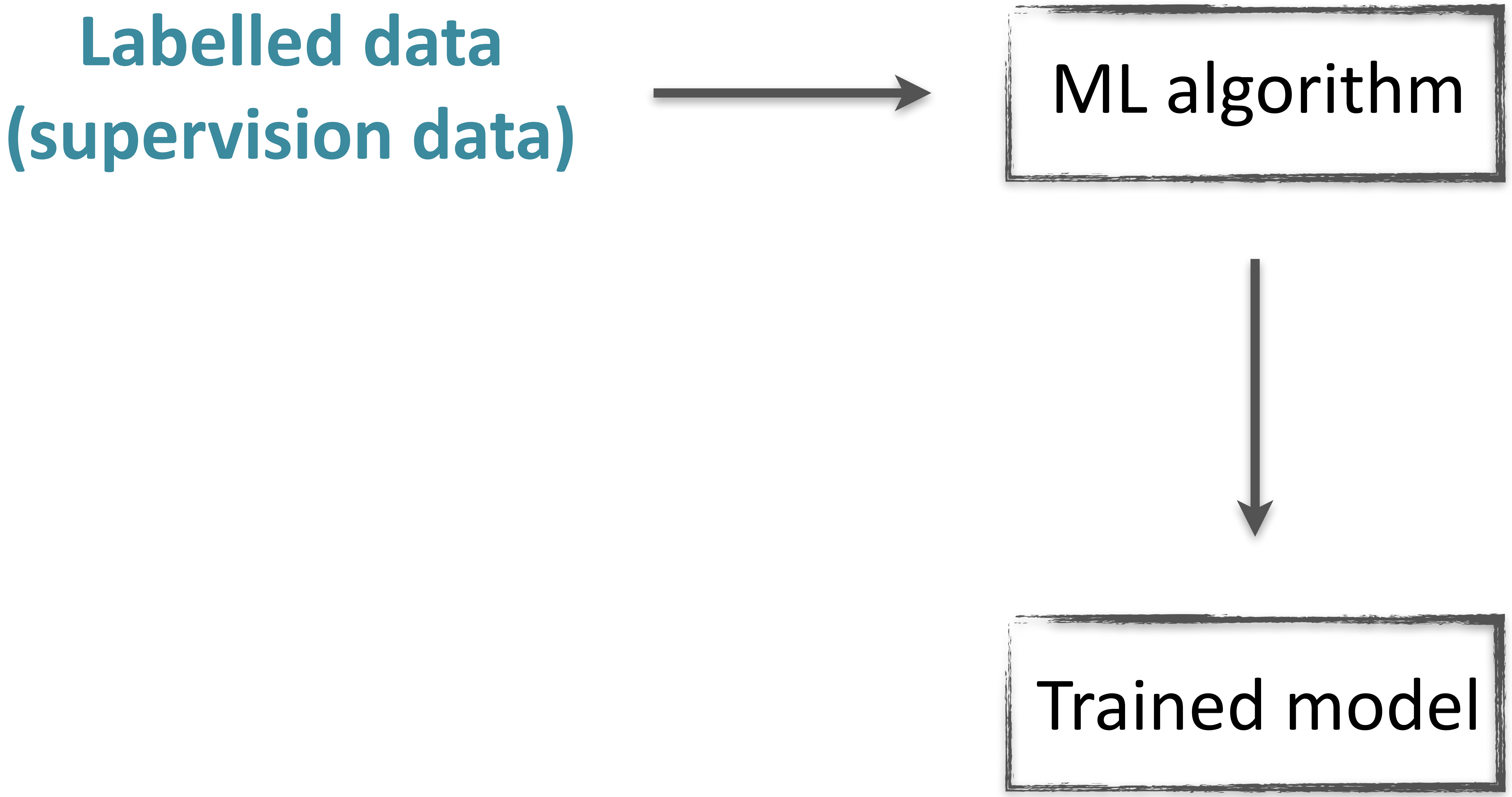
NN



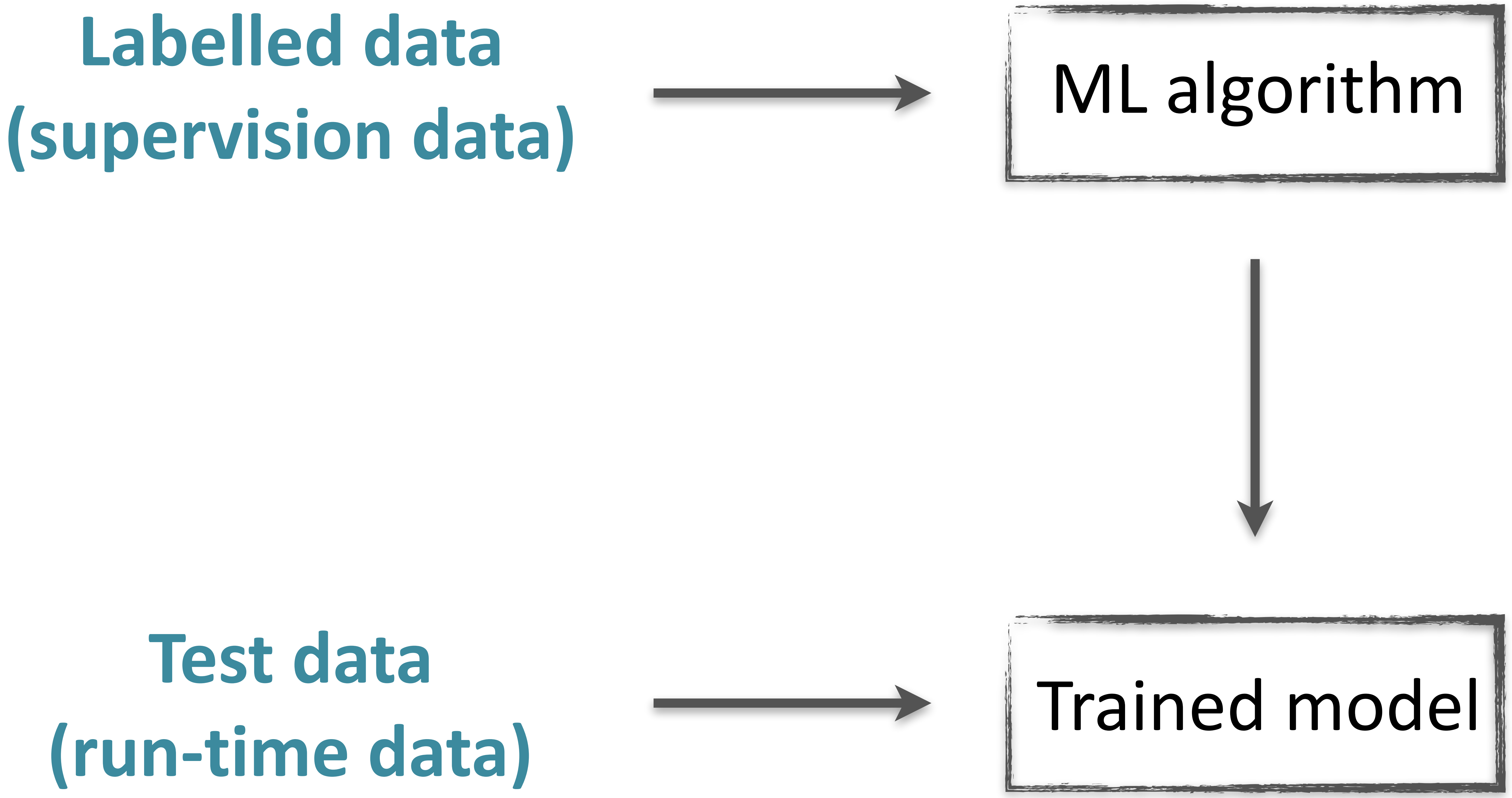
Data-driven Algorithms (Supervised)

Labelled data
(supervision data)

Data-driven Algorithms (Supervised)



Data-driven Algorithms (Supervised)



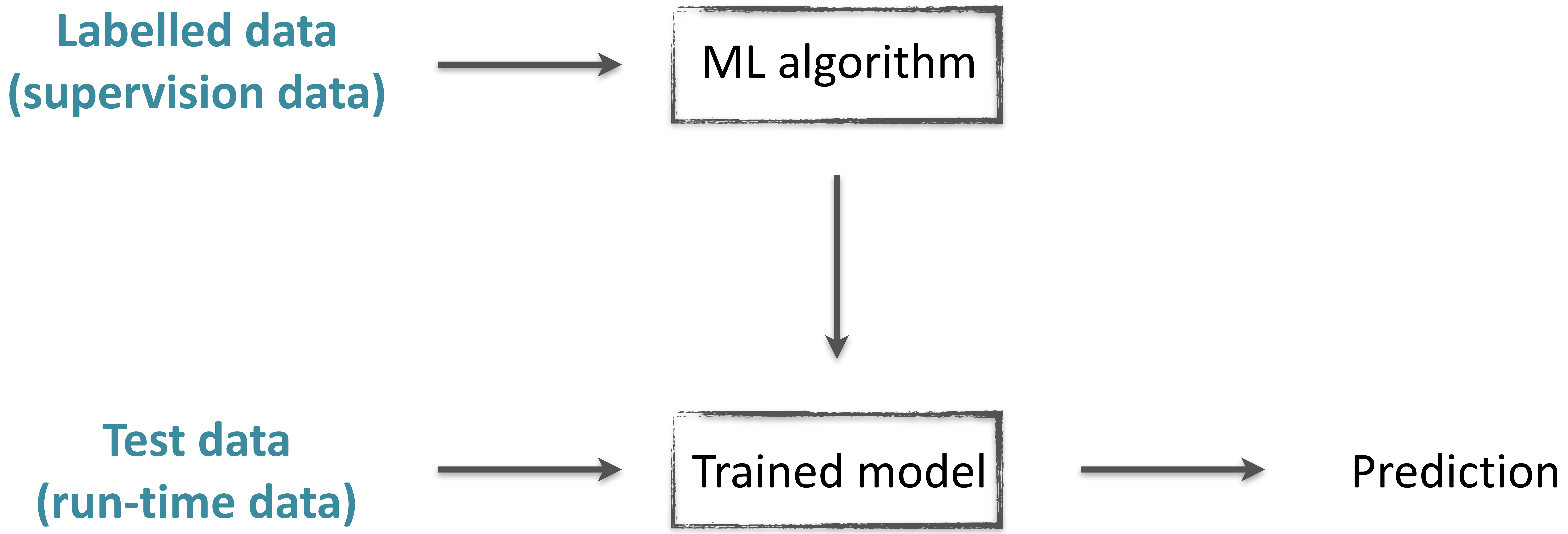
Labelled data
(supervision data)

ML algorithm

Trained model

Test data
(run-time data)

Data-driven Algorithms (Supervised)



Labelled data
(supervision data)

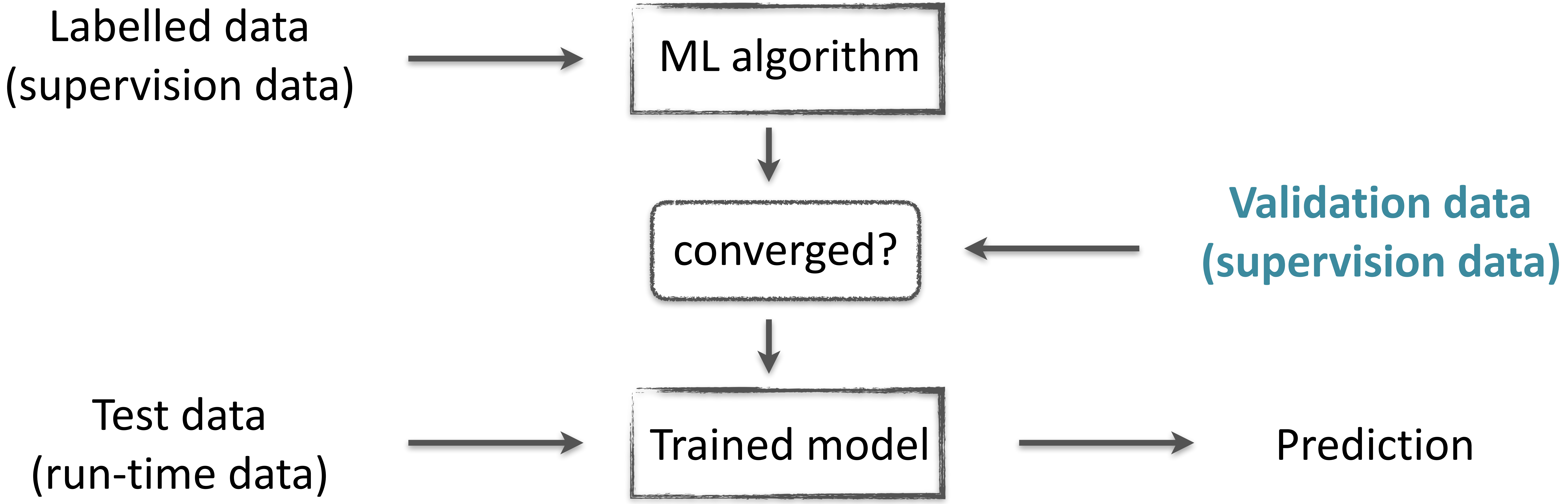
ML algorithm

Test data
(run-time data)

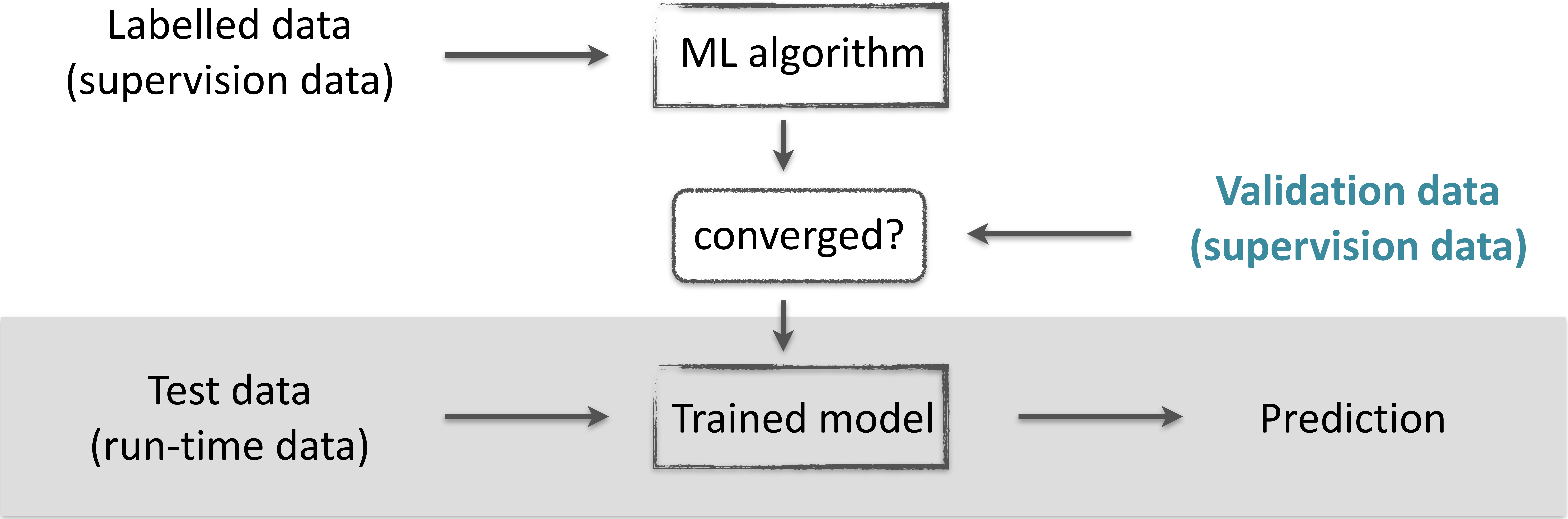
Trained model

Prediction

Data-driven Algorithms (Supervised)



Data-driven Algorithms (Supervised)



Labelled data (supervision data)

ML algorithm

converged?

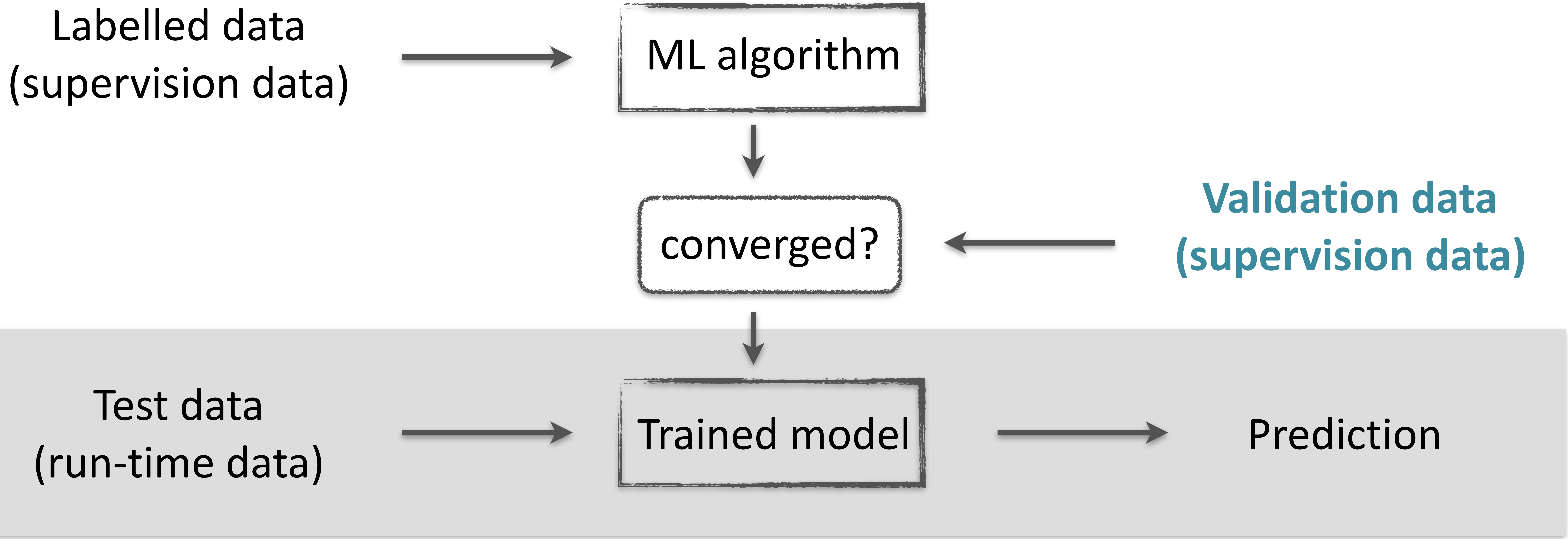
Validation data (supervision data)

Test data (run-time data)

Trained model

Prediction

Data-driven Algorithms (Supervised)


Labelled data
(supervision data)

ML algorithm

converged?

Validation data
(supervision data)

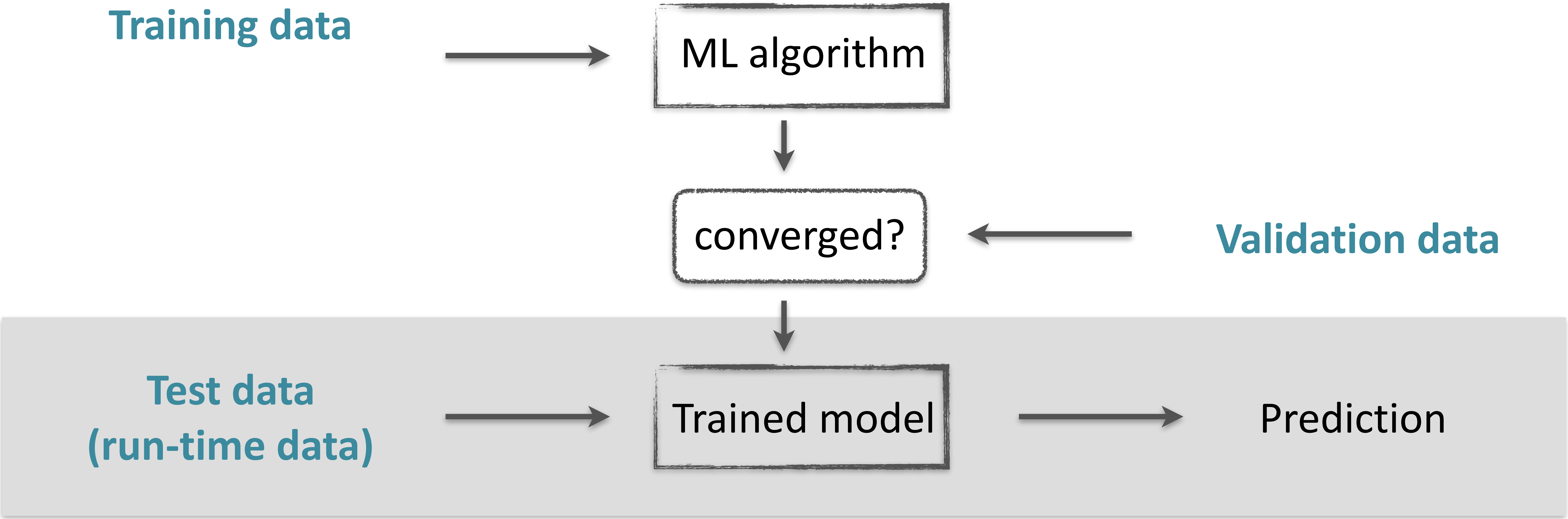
Test data
(run-time data)

Trained model

Prediction

Implementation Practice: Training: 70%; Validation: 15%; Test 15%

Data-driven Algorithms (Unsupervised)



Training data

ML algorithm

converged?

Validation data

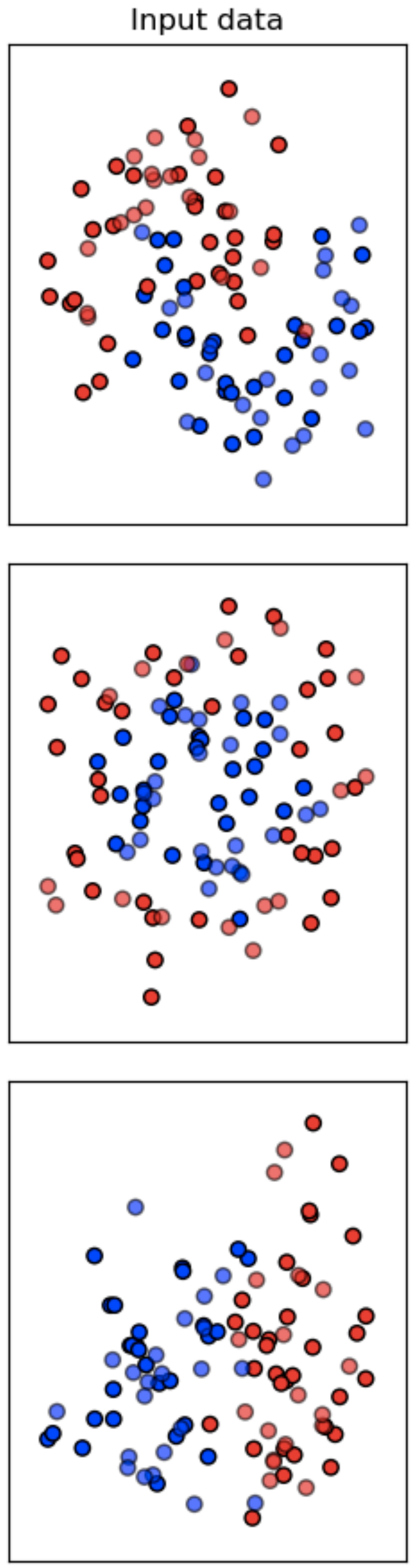
Test data
(run-time data)

Trained model

Prediction

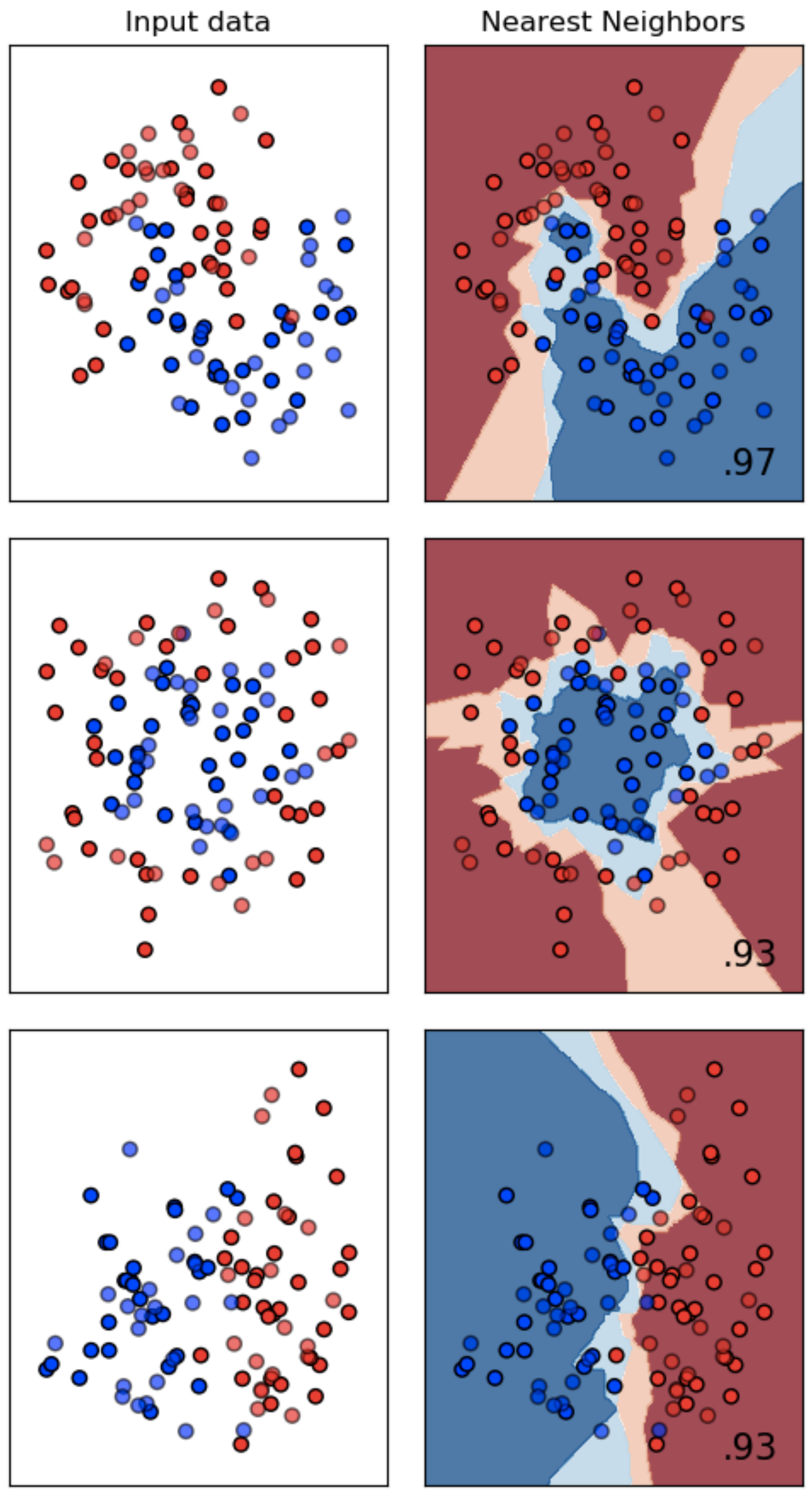
Implementation Practice: Training: 70%; Validation: 15%; Test 15%

Various ML Approaches (Supervised approaches)



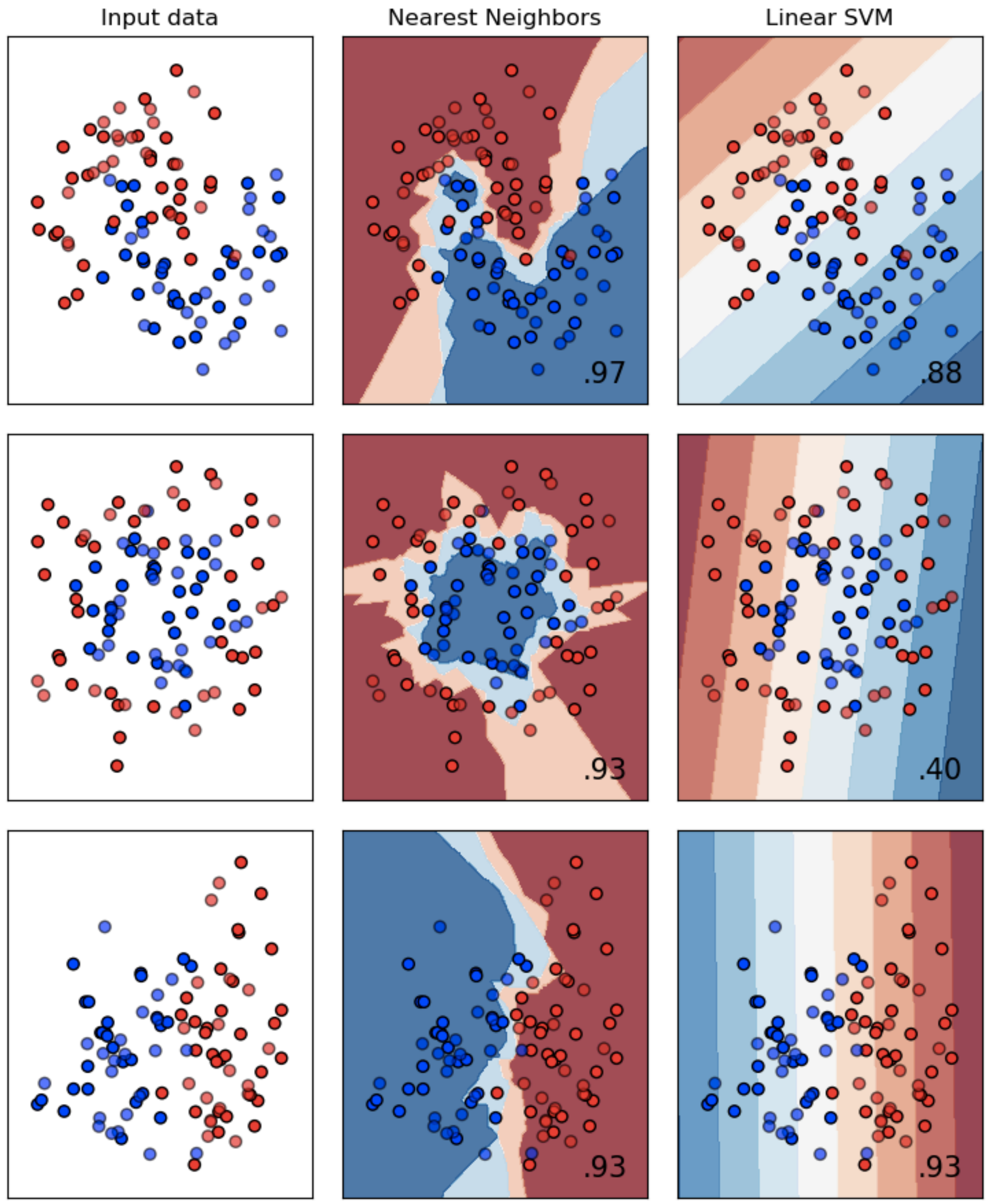
http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Various ML Approaches (Supervised approaches)



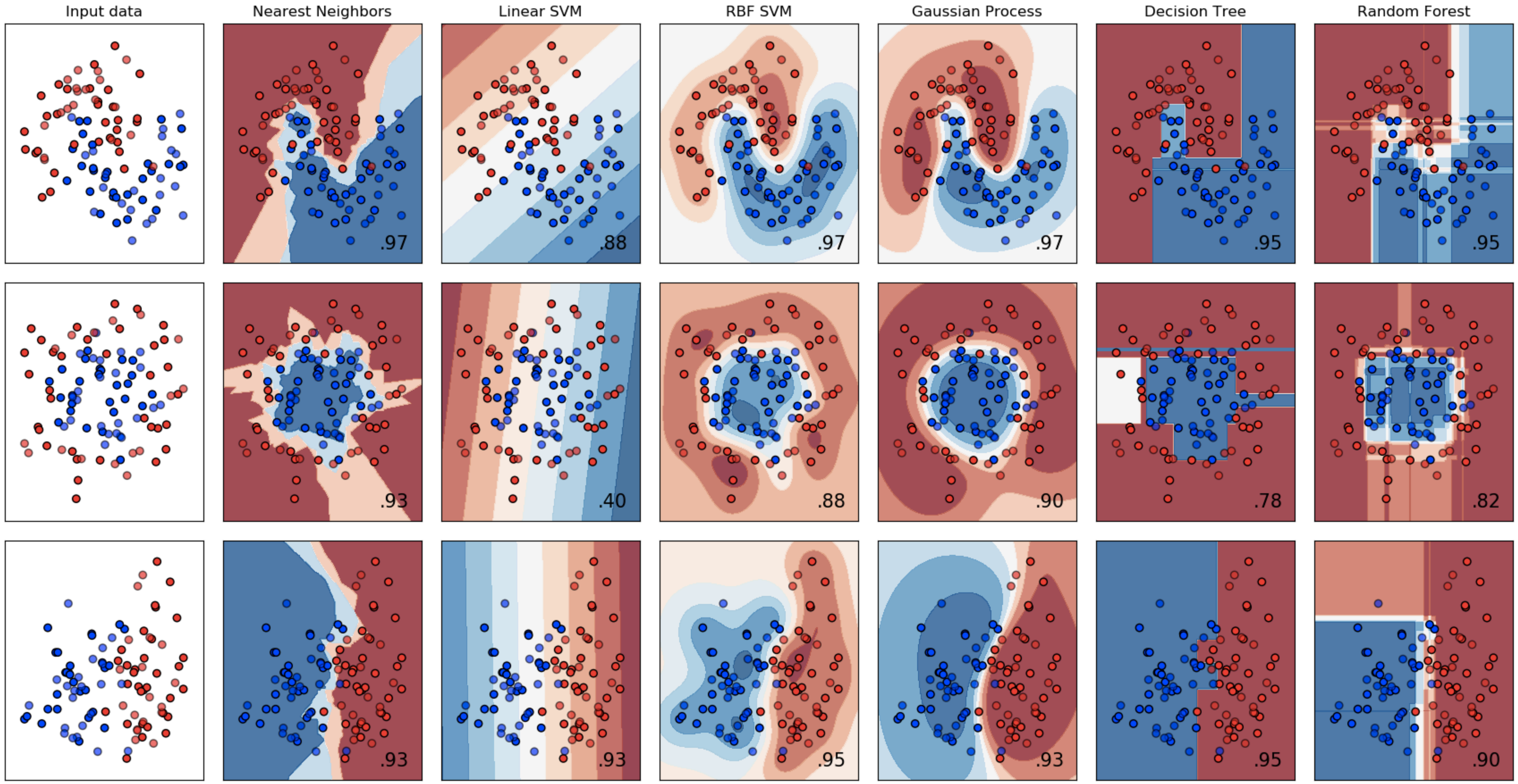
http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Various ML Approaches (Supervised approaches)



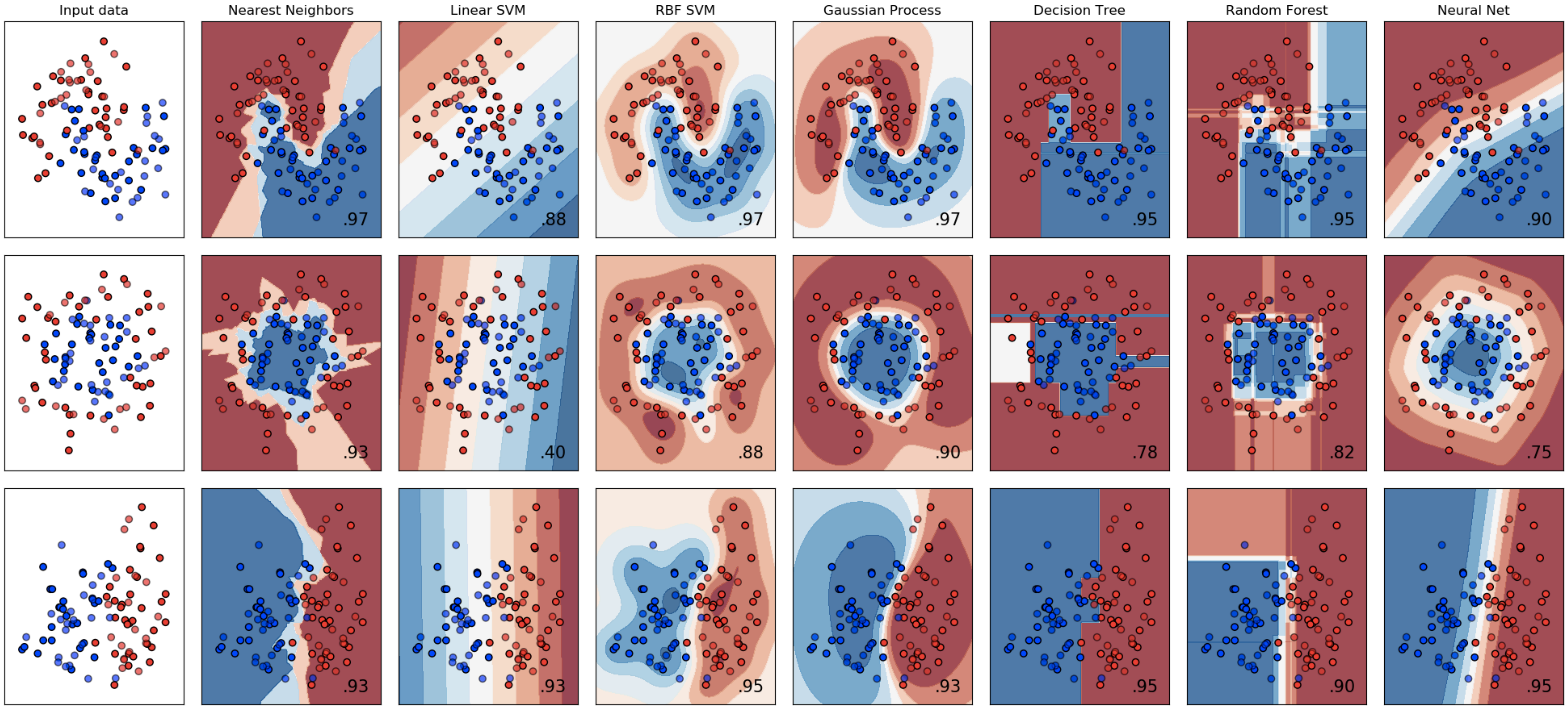
http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Various ML Approaches (Supervised approaches)



http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Various ML Approaches (Supervised approaches)

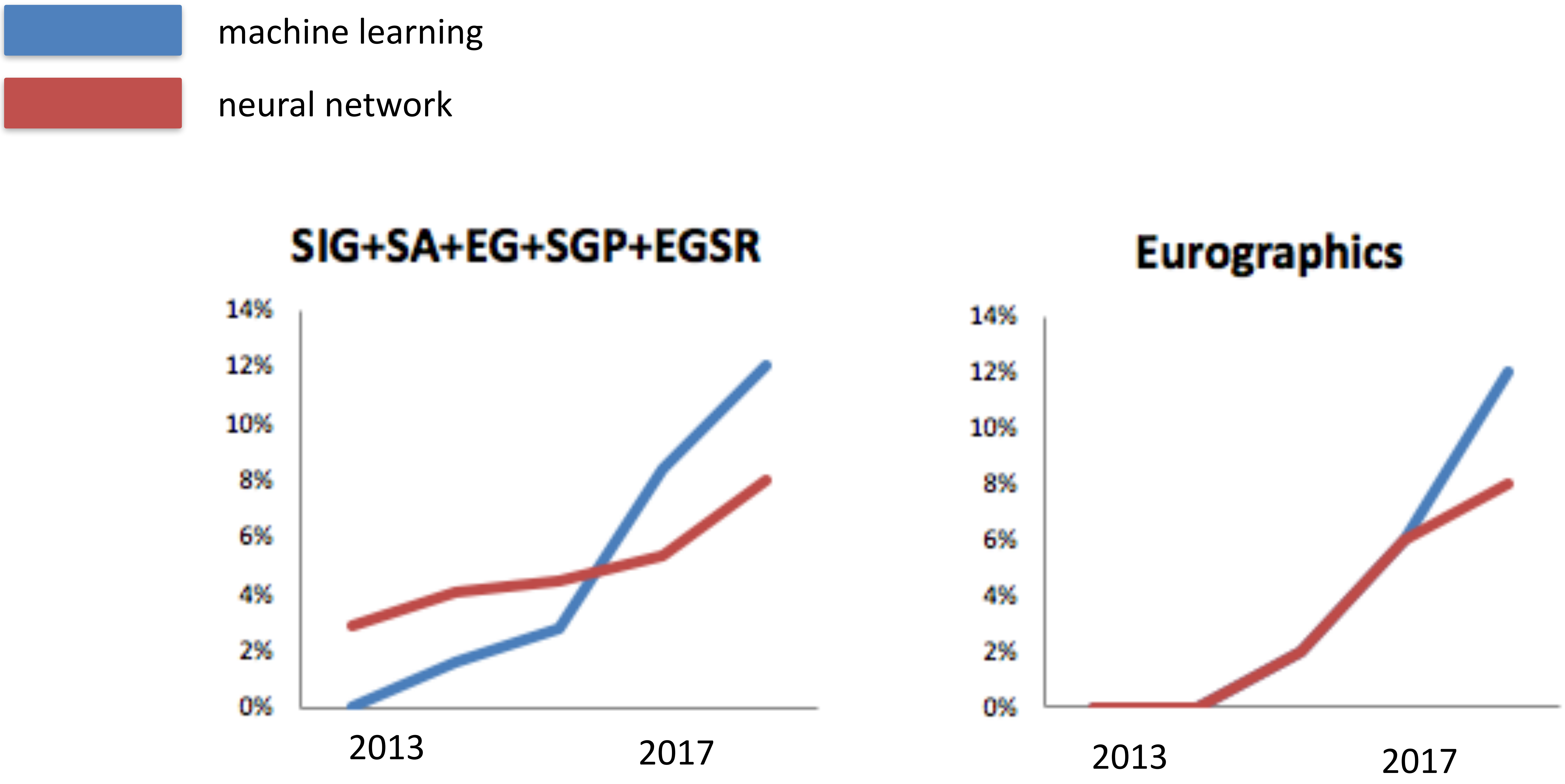


http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Rise of Learning

- 1958: Perceptron
- 1974: Backpropagation
- 1981: Hubel & Wiesel wins Nobel prize for ‘visual system’
- 1990s: SVM era
- 1998: CNN used for handwriting analysis
- **2012: AlexNet wins ImageNet**

Rise of Machine Learning (in Graphics)



What is Special about Graphics?

What is Special about Graphics?

- **Image Processing** (image translation tasks)

What is Special about Graphics?

- **Image Processing** (image translation tasks)
- Many sources of input data — **model building** (e.g., images, scanners, motion capture)

What is Special about Graphics?

- **Image Processing** (image translation tasks)
- Many sources of input data — **model building** (e.g., images, scanners, motion capture)
- Many sources of **synthetic data** — can serve as supervision data (e.g., rendering, animation)

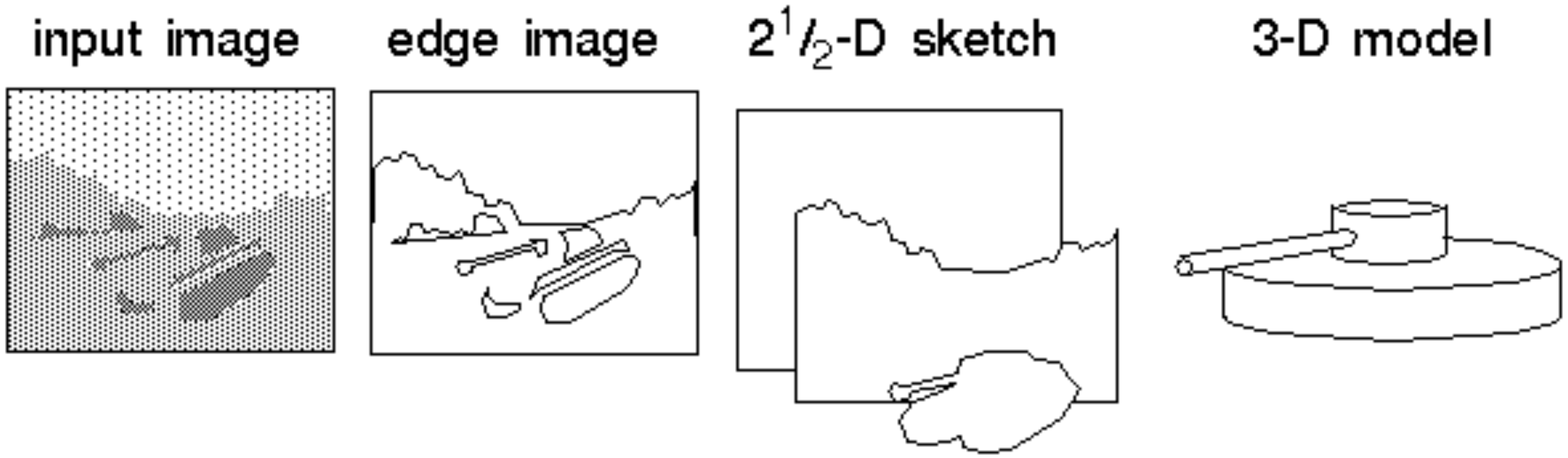
What is Special about Graphics?

- **Image Processing** (image translation tasks)
- Many sources of input data — **model building** (e.g., images, scanners, motion capture)
- Many sources of **synthetic data** — can serve as supervision data (e.g., rendering, animation)
- Many problems in **generative models**

End-to-end: **Features**

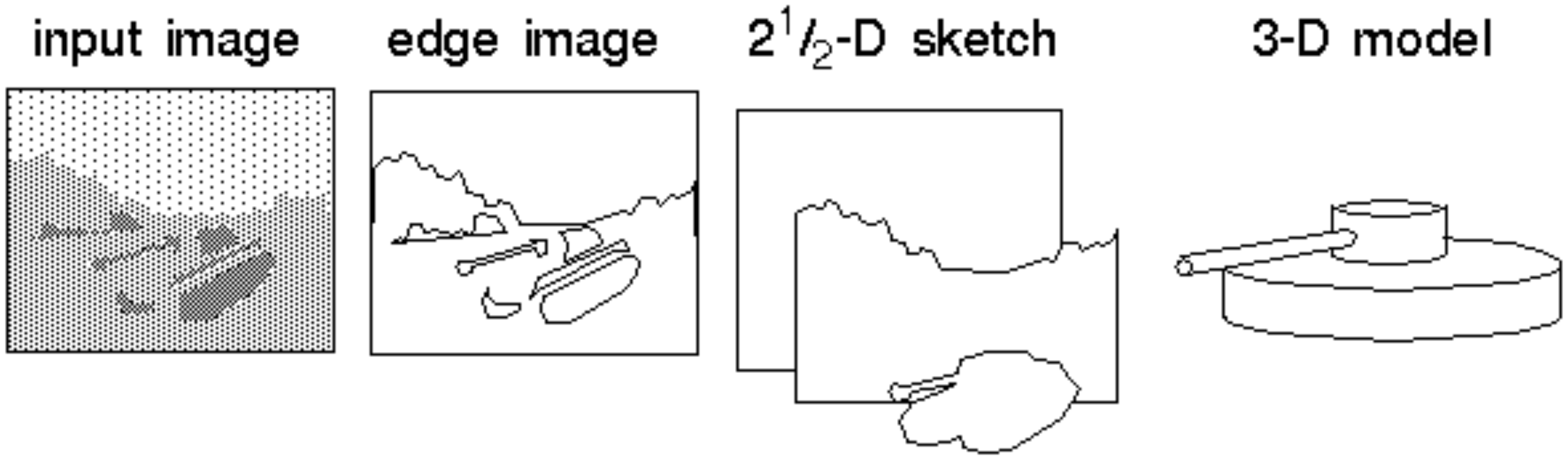
End-to-end: **Features**

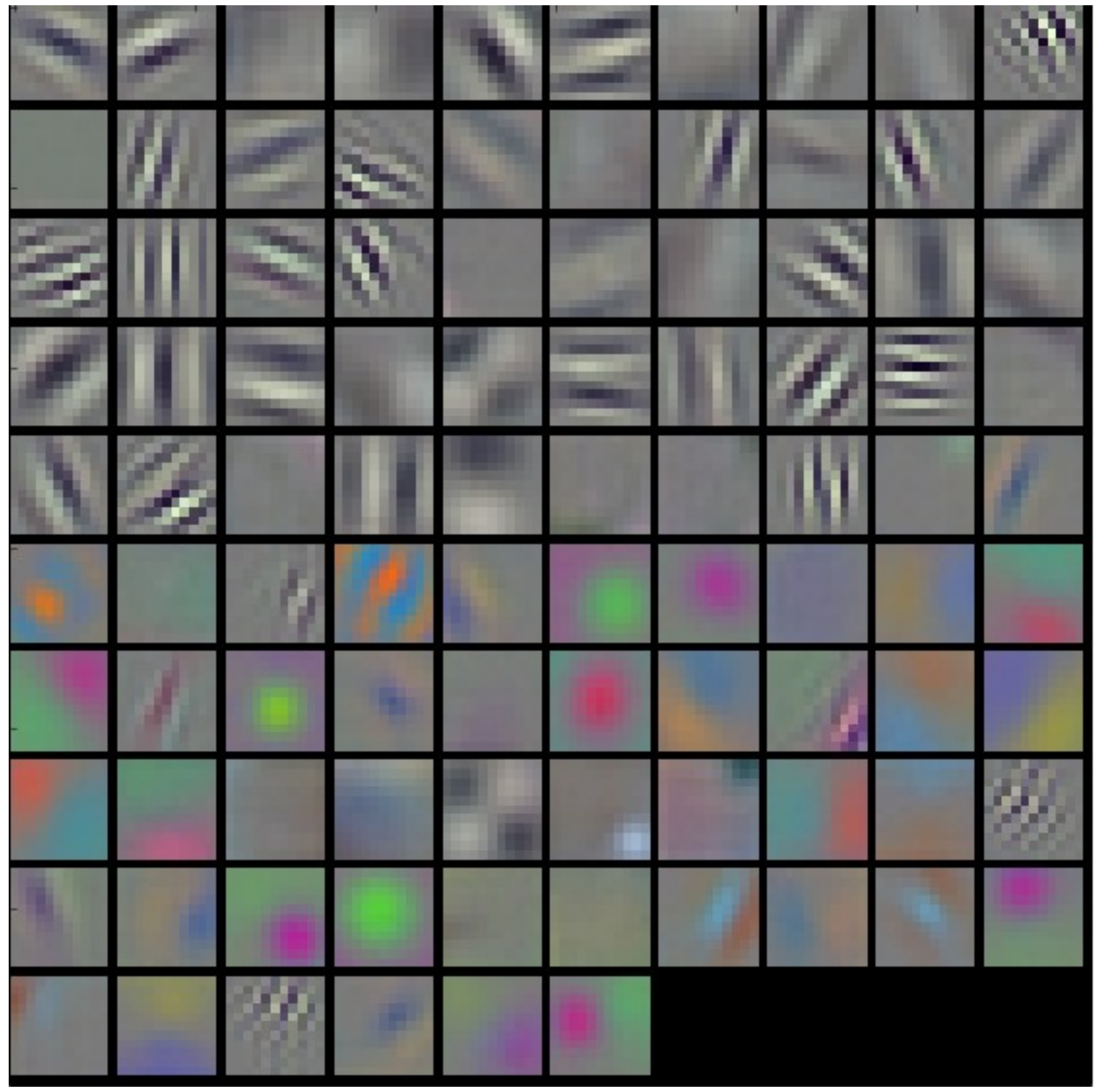
- *Old days*
  - First some handy features were extracted, e.g. edges or corners (hand-crafted)
  - Second, some AI was ran on that features (optimized)



End-to-end: **Features**

- *Old days*
  - First some handy features were extracted, e.g. edges or corners (hand-crafted)
  - Second, some AI was ran on that features (optimized)
- *Now*
  - End-to-end
  - Move away from hand-crafted representations





End-to-end: **Loss**

End-to-end: **Loss**

- *Old days*
  - Evaluation came after
  - It was a bit optional:
    - You might still have a good algorithm without a good way of quantifying it
    - Evaluation helped publishing

End-to-end: **Loss**

- *Old days*
  - Evaluation came after
  - It was a bit optional:
    - You might still have a good algorithm without a good way of quantifying it
    - Evaluation helped publishing
- *Now*
  - It is essential and build-in
  - If the loss is not good, the result is not good
  - Evaluation happens automatically

End-to-end: **Loss**

- *Old days*
  - Evaluation came after
  - It was a bit optional:
    - You might still have a good algorithm without a good way of quantifying it
    - Evaluation helped publishing
- *Now*
  - It is essential and build-in
  - If the loss is not good, the result is not good
  - Evaluation happens automatically
- While still much is left to do, this makes graphics much more reproducable

End-to-end: **Data**

End-to-end: **Data**

- *Old days*
  - Test with some toy examples
  - Deploy on real stuff
  - Maybe collect some data later

End-to-end: **Data**

- *Old days*
  - Test with some toy examples
  - Deploy on real stuff
  - Maybe collect some data later

- *Now*
  - Test and deploy need to be as identical as you can
  - Need to collect data first
  - No two steps

Examples in Graphics

Geometry

Image manipulation

Animation

Rendering

Examples in Graphics

Sketch simplification

Colorization

Image manipulation

BRDF estimation

Real-time rendering

Rendering

Denoising

Geometry

Procedural modelling

Mesh segmentation

Learning deformations

Animation

Boxification

Fluid

Animation

Facial animation

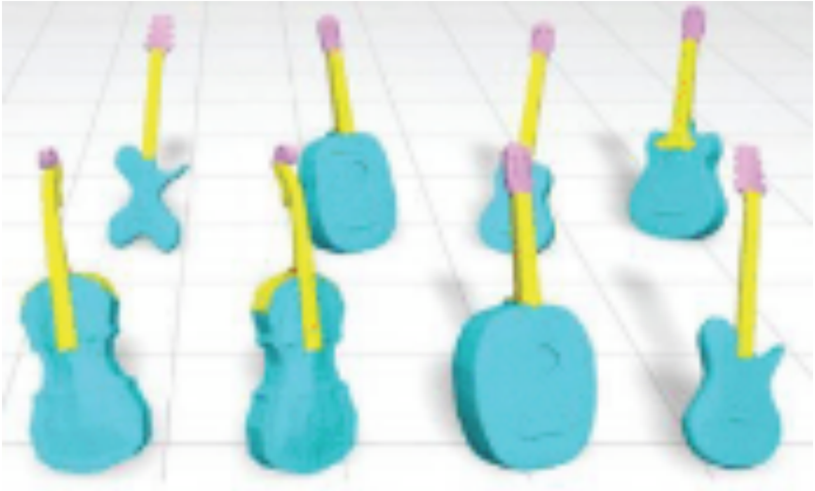
PCD processing

Examples in Graphics

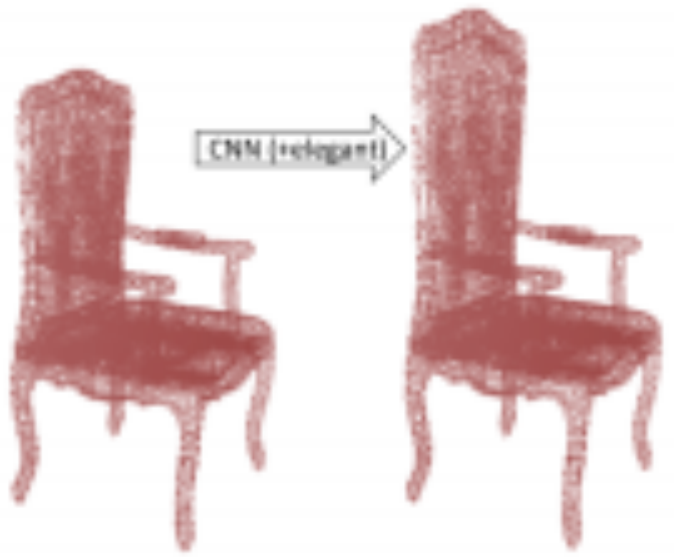
Sketch simplification

Colorization

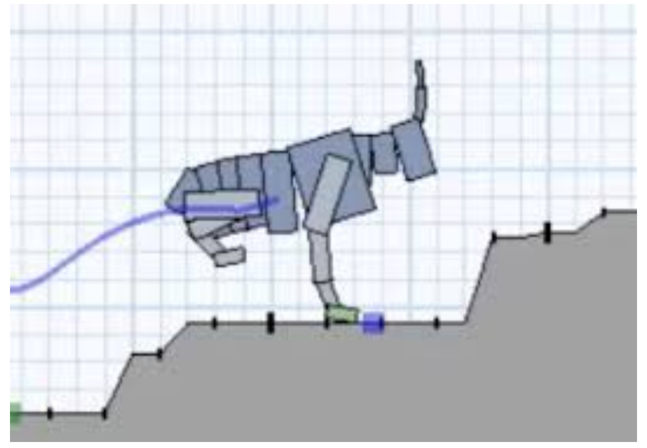
Procedural modelling



Mesh segmentation



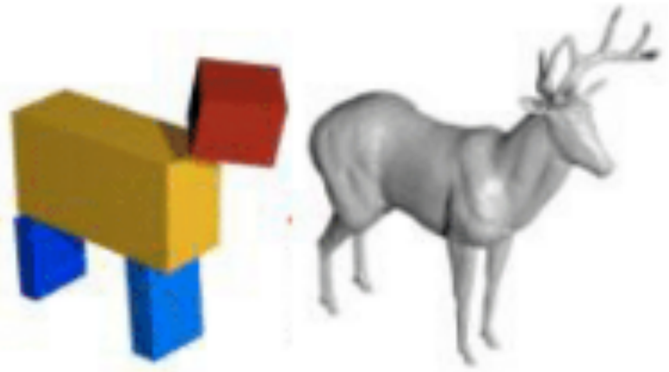
Learning deformations



Animation

BRDF estimation

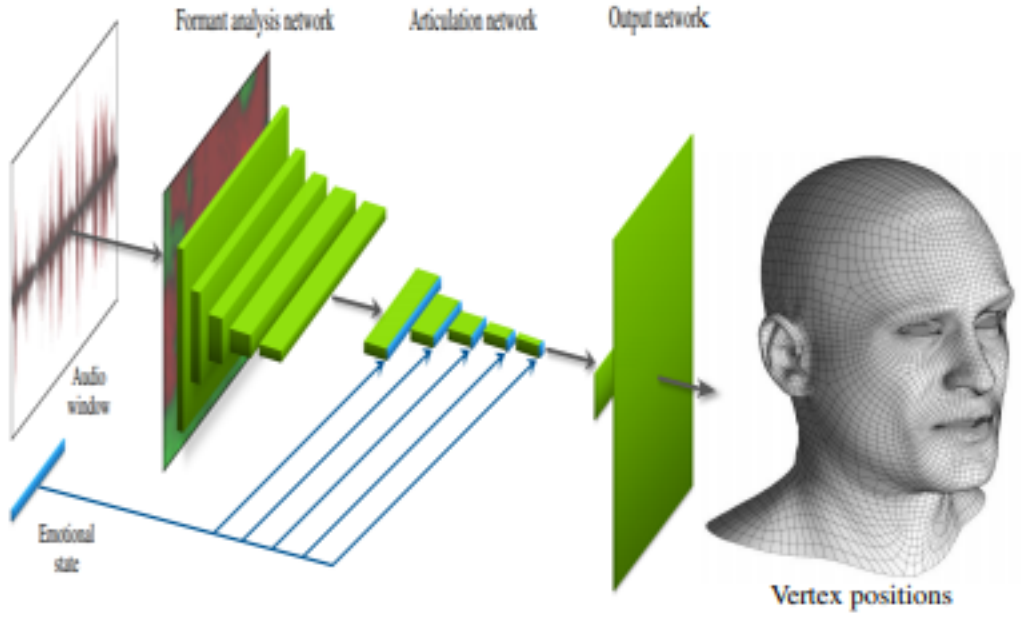
Fluid



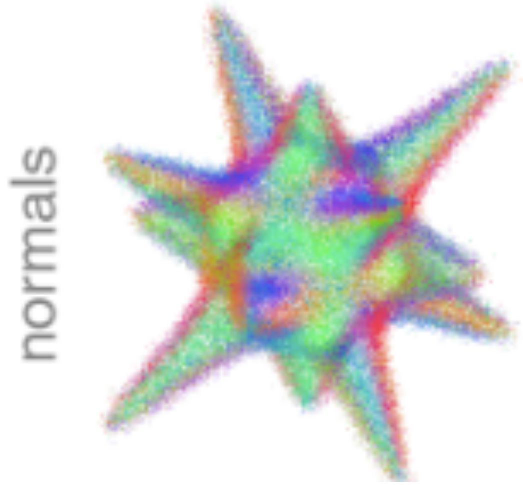
Boxification

Real-time rendering

Denoising

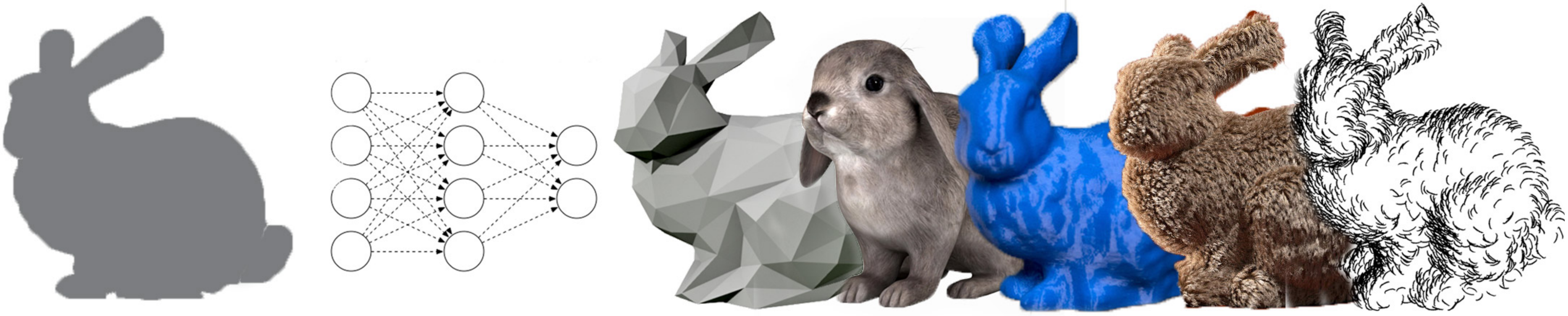


Facial animation



PCD processing

Course Information (slides/code/comments)



http://geometry.cs.ucl.ac.uk/dl4g/