Introduction to Machine Learning

Dr. Nikos Nikolaou,

Extrasolar Planets Group, Astrophysics

What is Machine Learning?

What are computers good for?

'**Computing**'… flawlessly execute trillions of FLOP/s

Can solve a problem as long as **programmed** to do so

'**Program**' ('**algorithm**'):
a finite, well-defined series of steps for solving a problem

Human devises algorithms, computer executes

But for many problems hard to devise an algorithm...

(many variables – some irrelevant – others related in complicated ways,
noise/stochasticity involved, ability & time limited, …)

An easy vs. a hard task to program

Easy: ‘ Find all primes $<$ n ’

Example algorithm: **Sieve of Eratosthenes** (c. 250BC)

1. List all integers from 2 to n
2. Let $p = 2$ (the smallest prime)
3. Enumerate all multiples of p, counting to n from $2p$ in increments of p; Mark them in the list
4. Find the first number $p' > p$ in the list that is not marked;
   If there was no such number, stop;
   Otherwise, let $p = p'$ (this is the next prime), and repeat from step 3
5. When algorithm terminates, the numbers remaining unmarked in the list are all the primes below n

Others: **Sieve of Sundaram** (1934), **Sieve of Atkins**(2004), …

An easy vs. a hard task to program

Hard: ‘ Is this a cat or a dog? ’

How to write an algorithm to solve this?

i.e. how to define in precise mathematical terms what constitutes a cat / a dog in an image?

Machine Learning

Can a computer learn on its own how to solve problems?

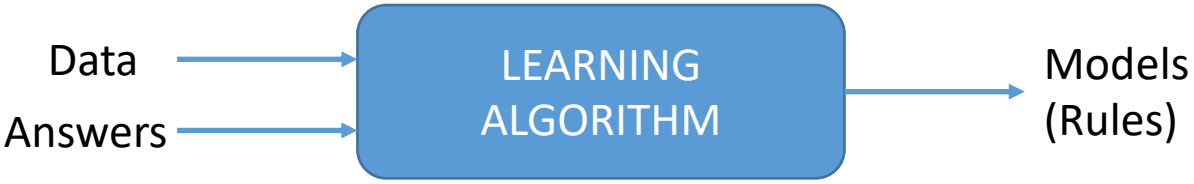
Can we automate the process of devising an algorithm?
(e.g. for telling apart cats from dogs)

The study of algorithms that learn algorithms

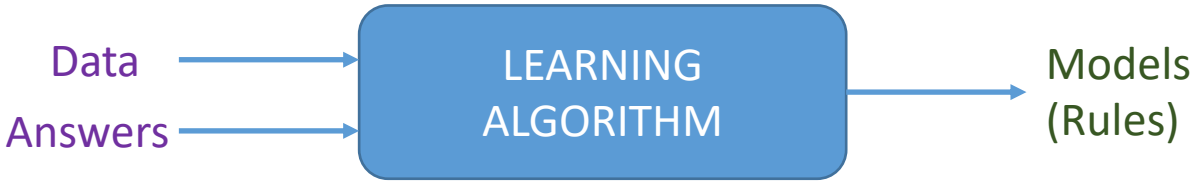
Learning from Data

Traditional Programming

Machine Learning



Inferring Rules from Observations



Given examples (observations),
find **a likely** rule (model, hypothesis) that generated them

A common theme in science

No **inference** w/o **assumptions**

Hence: All machine learning algorithms make assumptions!

So, ML…

AI subfield: automates algorithm design,
machine (re)programs itself
adapts to experience / data / feedback

Automates discovery of rules (models) **given data** (examples)

Central part of modern knowledge discovery / data analysis / computational statistics

many observations, many variables – some irrelevant – others related in complicated ways, noise/stochasticity, human ability & time limited

All ML methods make assumptions!

Types of Machine Learning Tasks

Types of Learning

Supervised Learning

Given **examples of input** & **output**, **find a 'good' mapping between them**

Unsupervised Learning

Given **examples of input**, **discover 'structure' in data**

underlying probability distribution
interesting subsets of (similar) examples
(or interesting regions of the underlying input space) ,
summarize data for visualization / get rid of irrelevant info,
find outliers,
identify interesting features

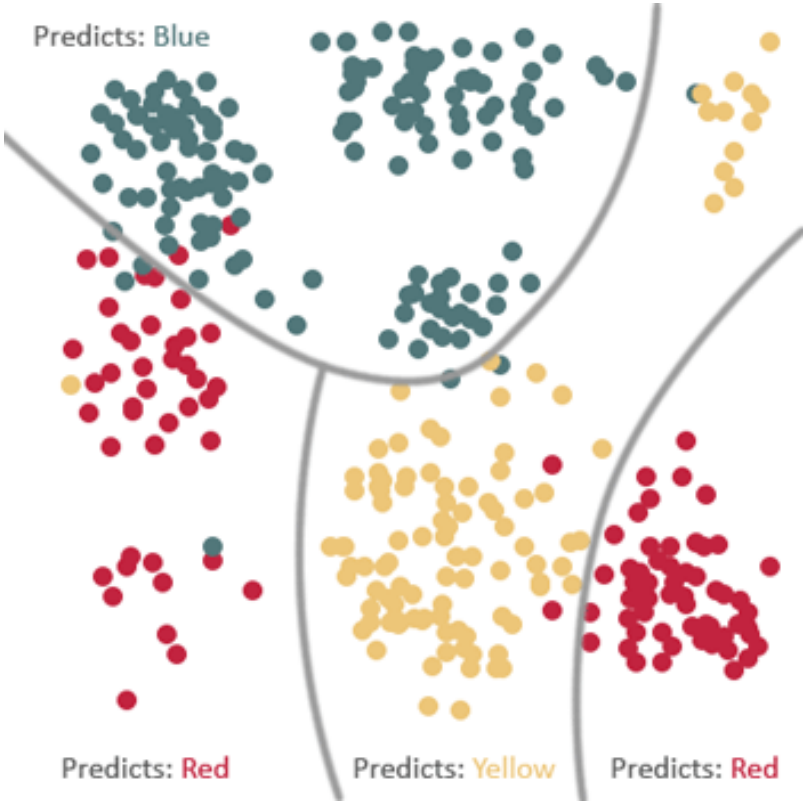
Reinforcement Learning

Given **a set of actions** & **reward feedback** from 'environment',
find 'good' sequences of actions

Supervised Learning (1)

Given examples of input X & output Y, find a 'good' mapping $Y = F(X)$

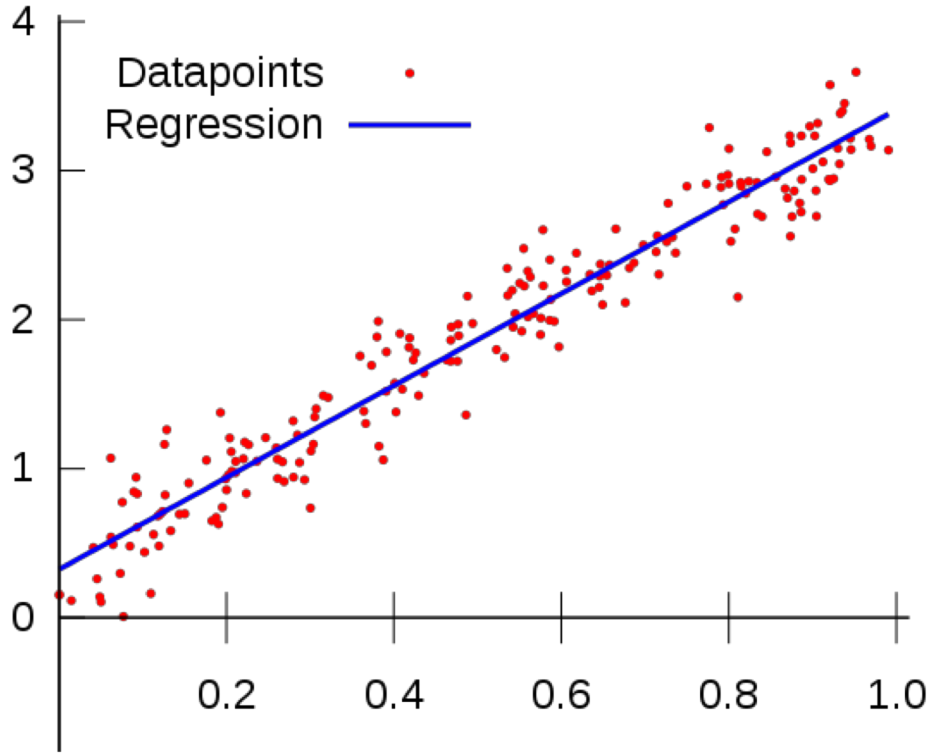
Classification: $Y \in$ a finite set



Supervised Learning (2)

Given examples of input X & output Y, find a 'good' mapping $Y = F(X)$

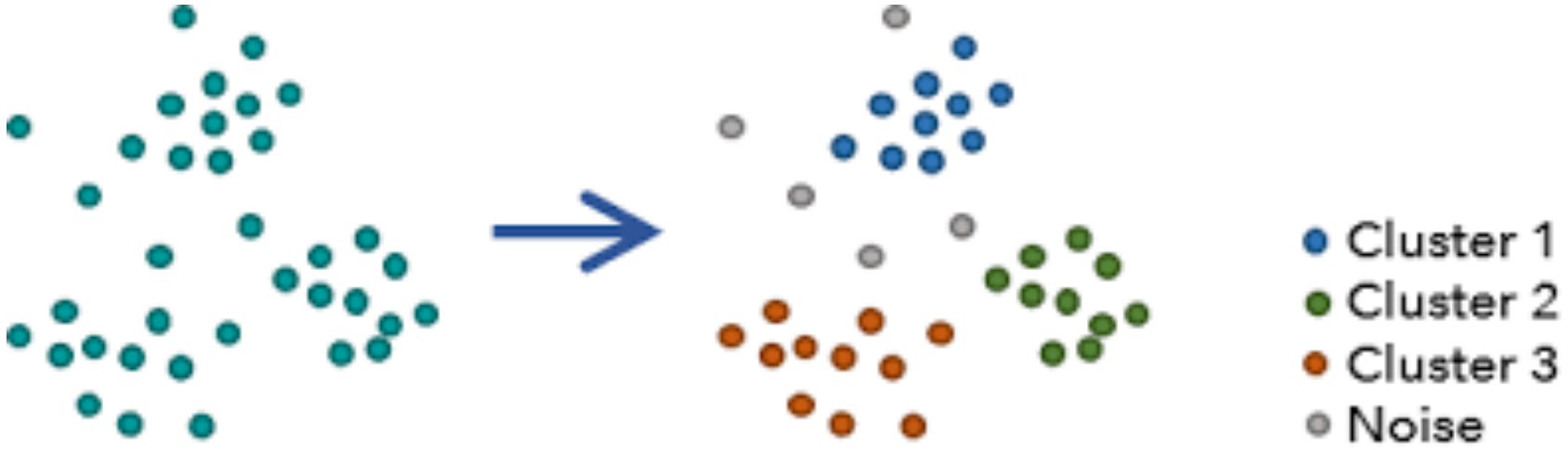
Regression: $Y \in \mathbb{R}$



Unsupervised Learning (1)

Given examples of input, find structure in data

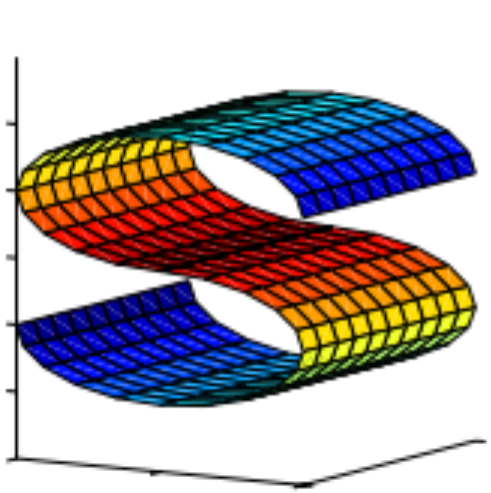
Clustering: Find subsets of ‘similar’ datapoints /
Separate input space in sub-regions

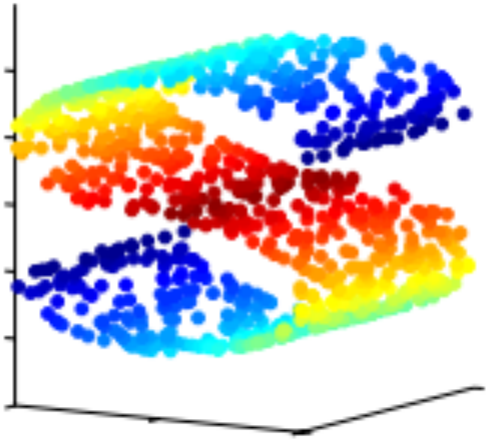


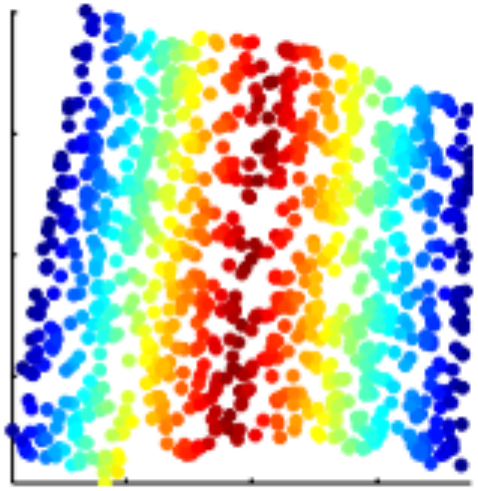
Unsupervised Learning (2)

Given examples of input, find structure in data

Dimensionality Reduction: Project data to some informative lower dimensional space



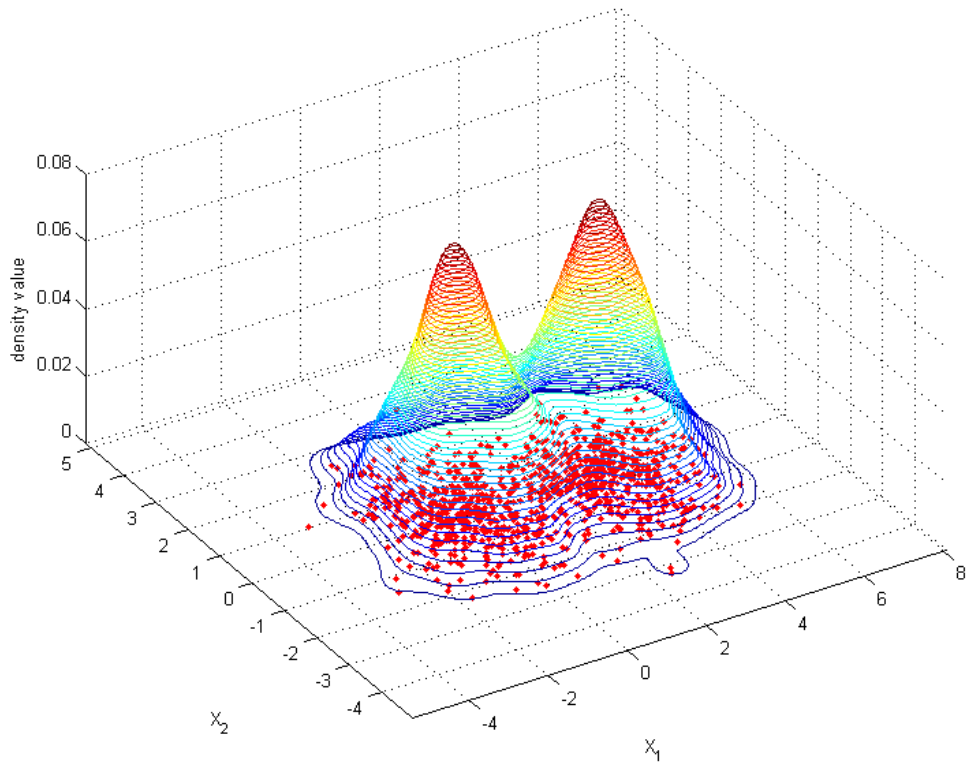




Unsupervised Learning (3)

Given examples of input, find structure in data

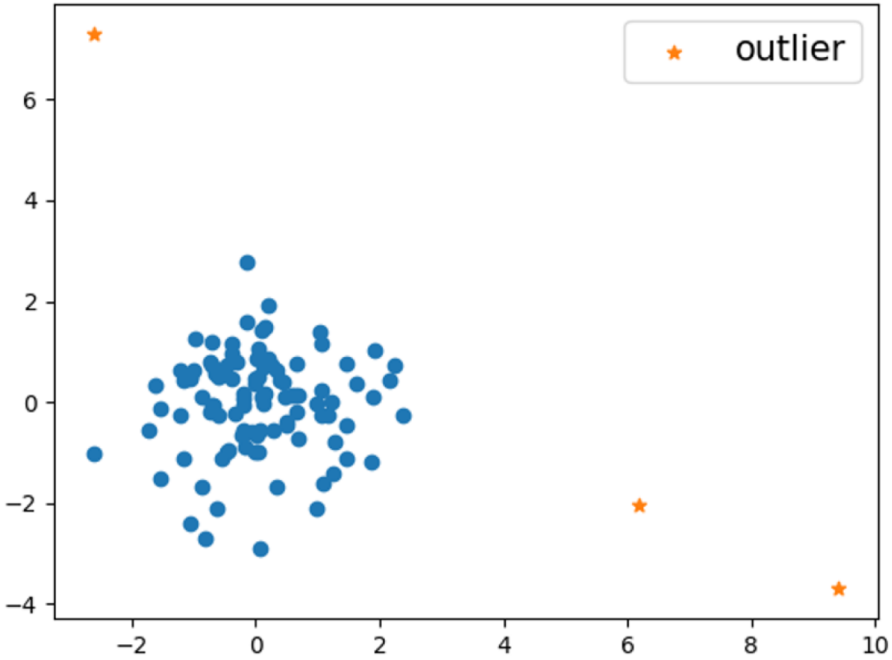
Density Estimation: Find underlying distribution generating data

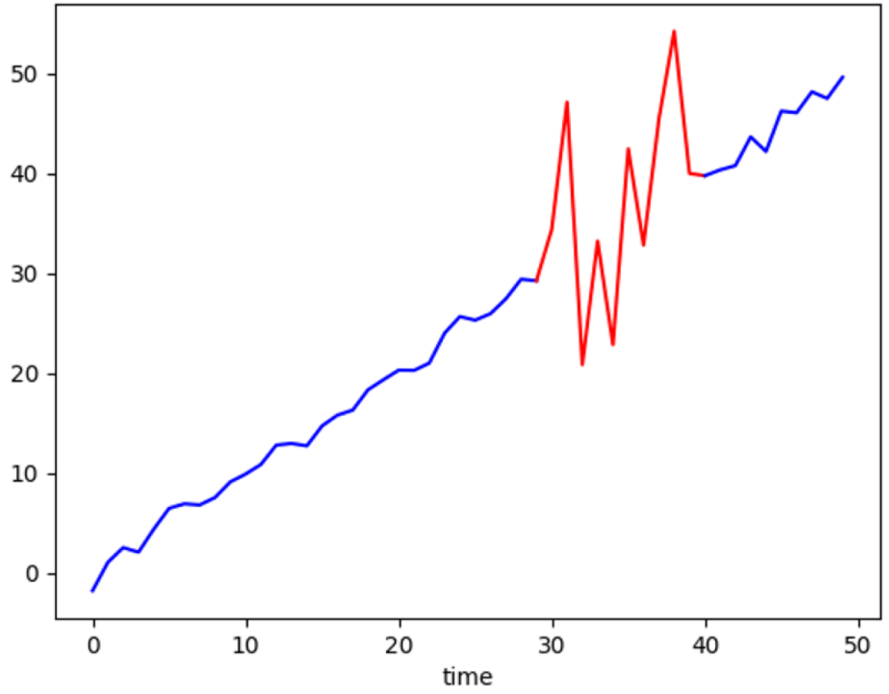


Unsupervised Learning (4)

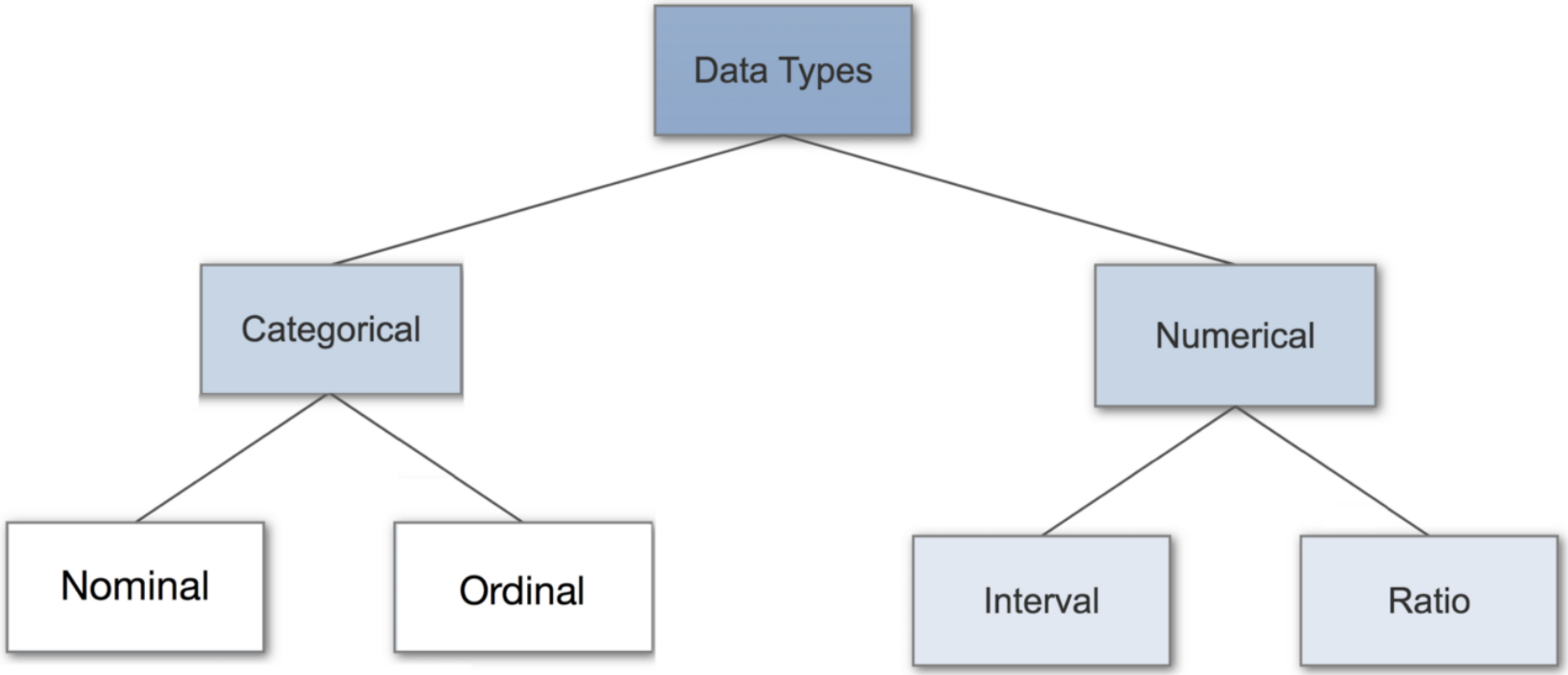
Given examples of input, find structure in data

Anomaly Detection: Find outliers in data





What can my ‘data’ be?



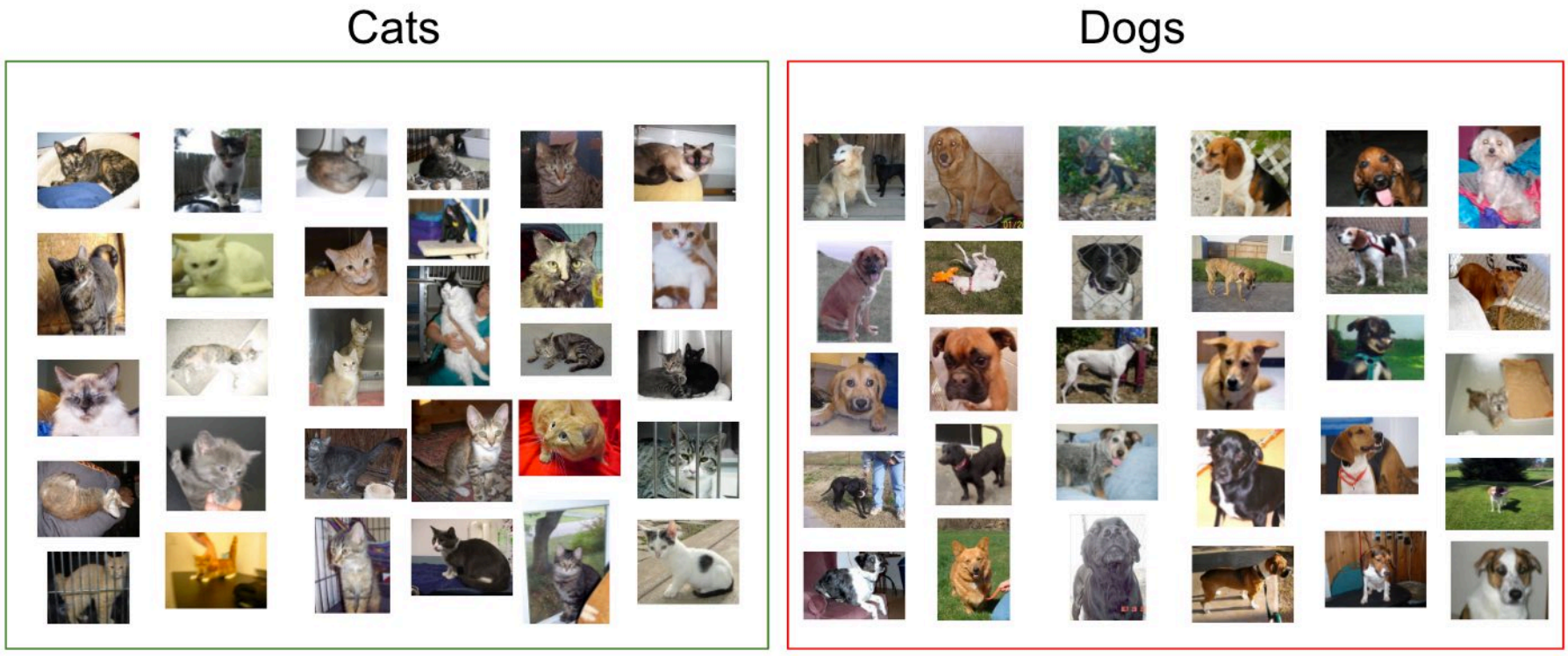
Any dimensionality (both input & output)

Can be **structured** (e.g. image, video, graphs, sequences, …)

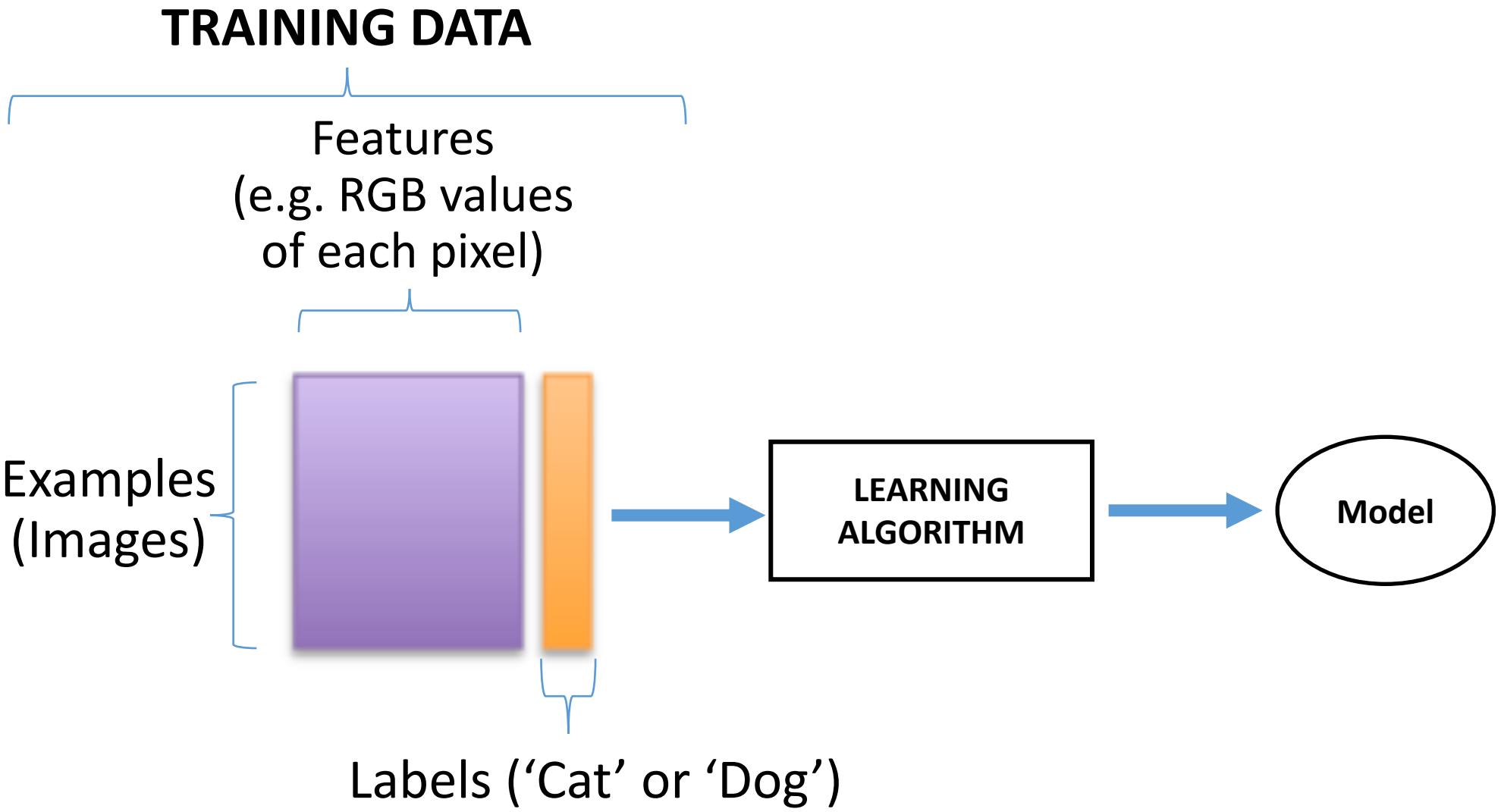
Might contain **missing values**

Example: Binary Classification

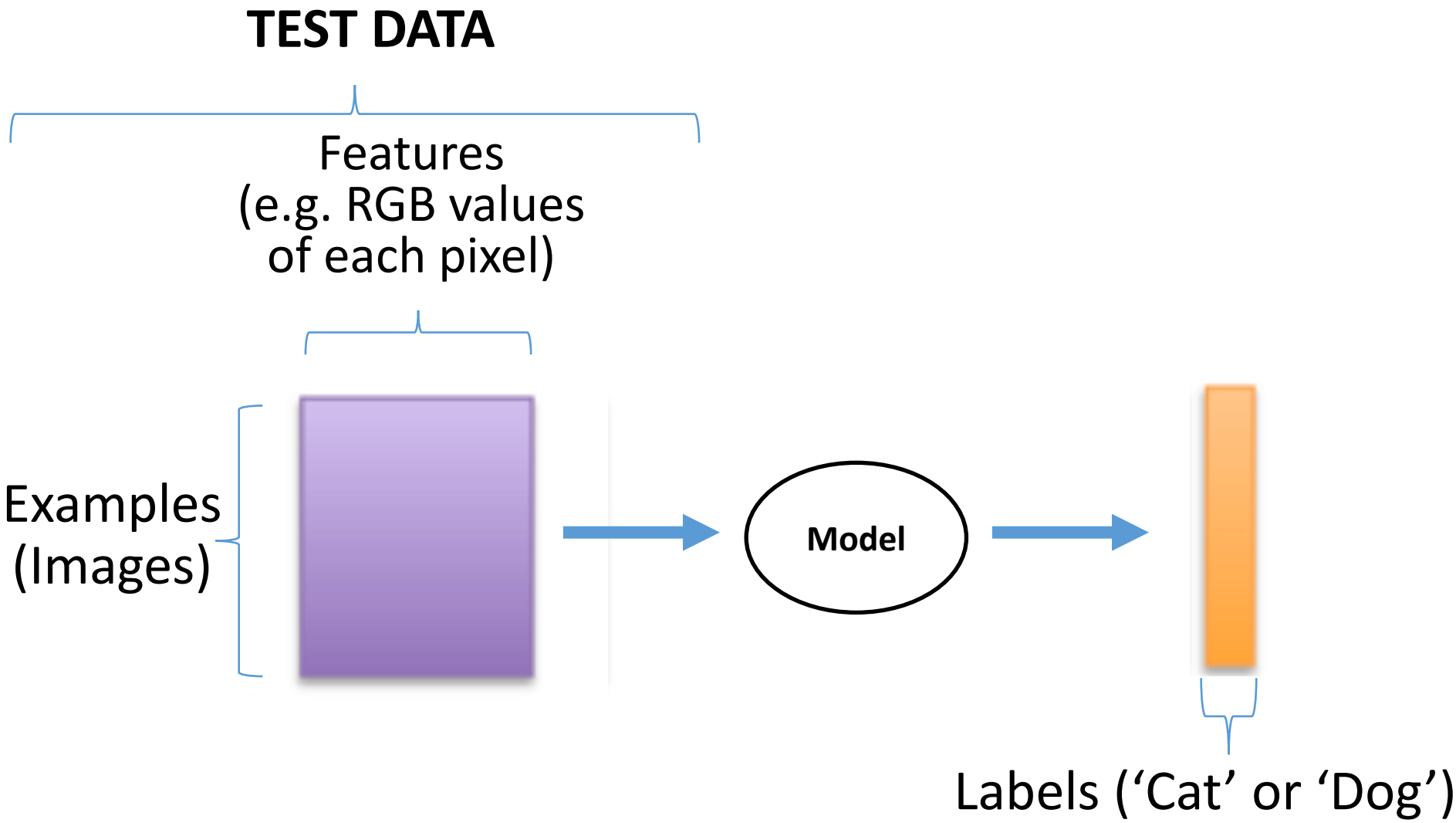
Dog vs. Cat Classifier



Training the Model



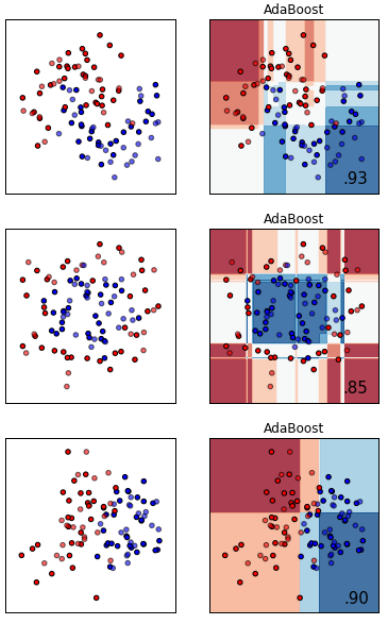
Obtaining Predictions from a Model



Learning Algorithms
&
Evaluating Models

Learning Algorithm's Job (in Classification)

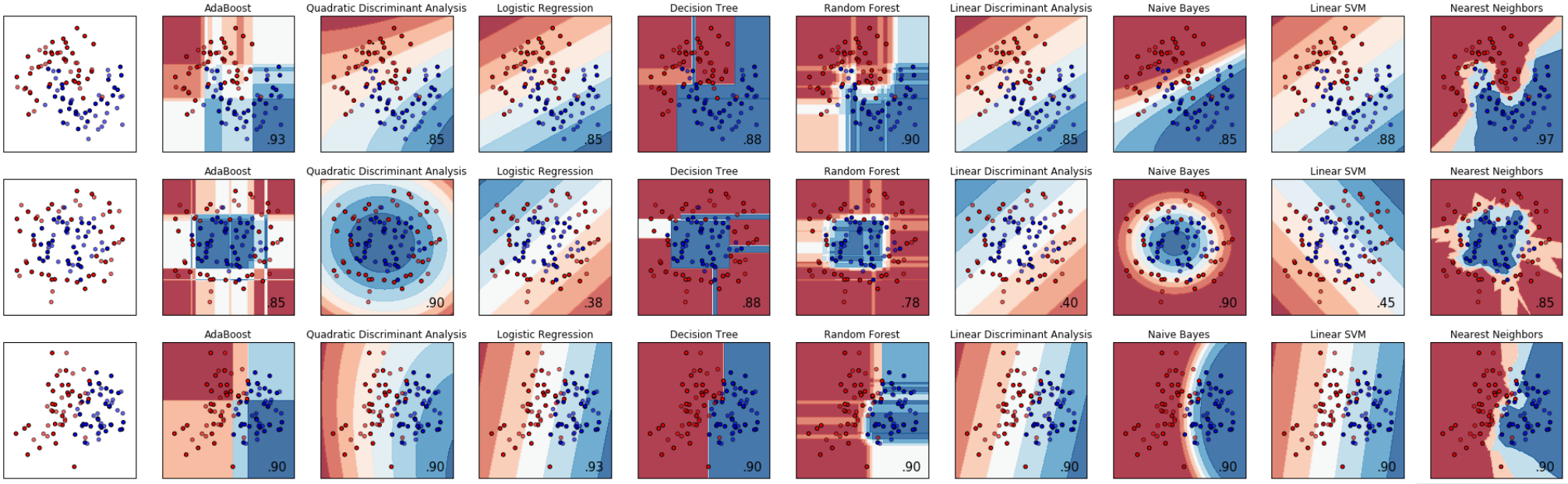
Given a set of points in some space belonging to different classes…



Learn **decision surface** that **'best' separates classes**

Many Learning Algorithms

Each with its own **assumptions**
(statistical, probabilistic, mathematical, topological, geometrical, …)



Goal of Learning: Generalization

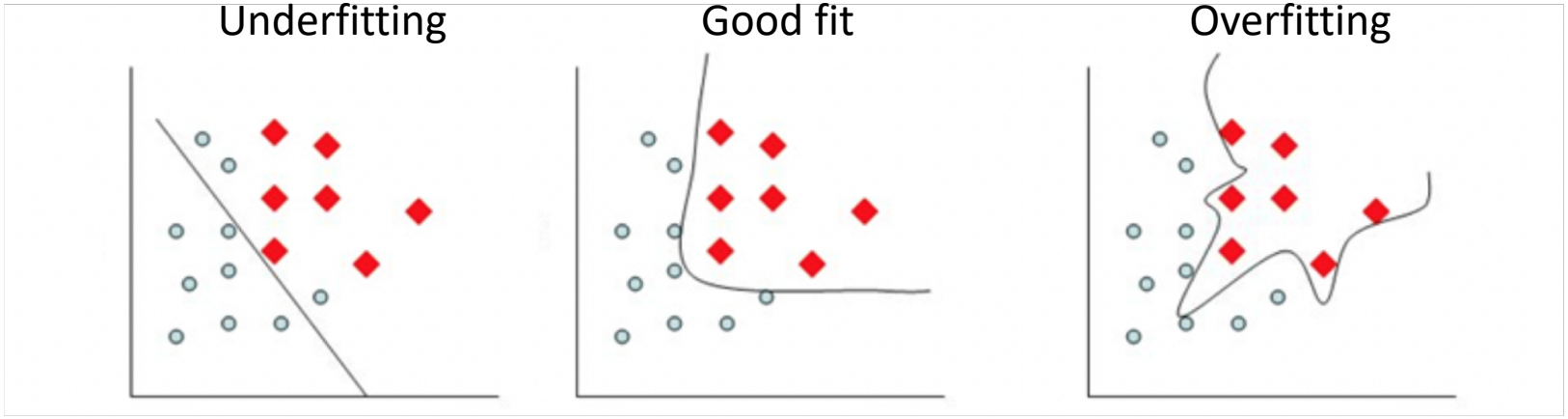
Want a model that performs well on new datapoints!

Learning ≠ memorizing your examples!

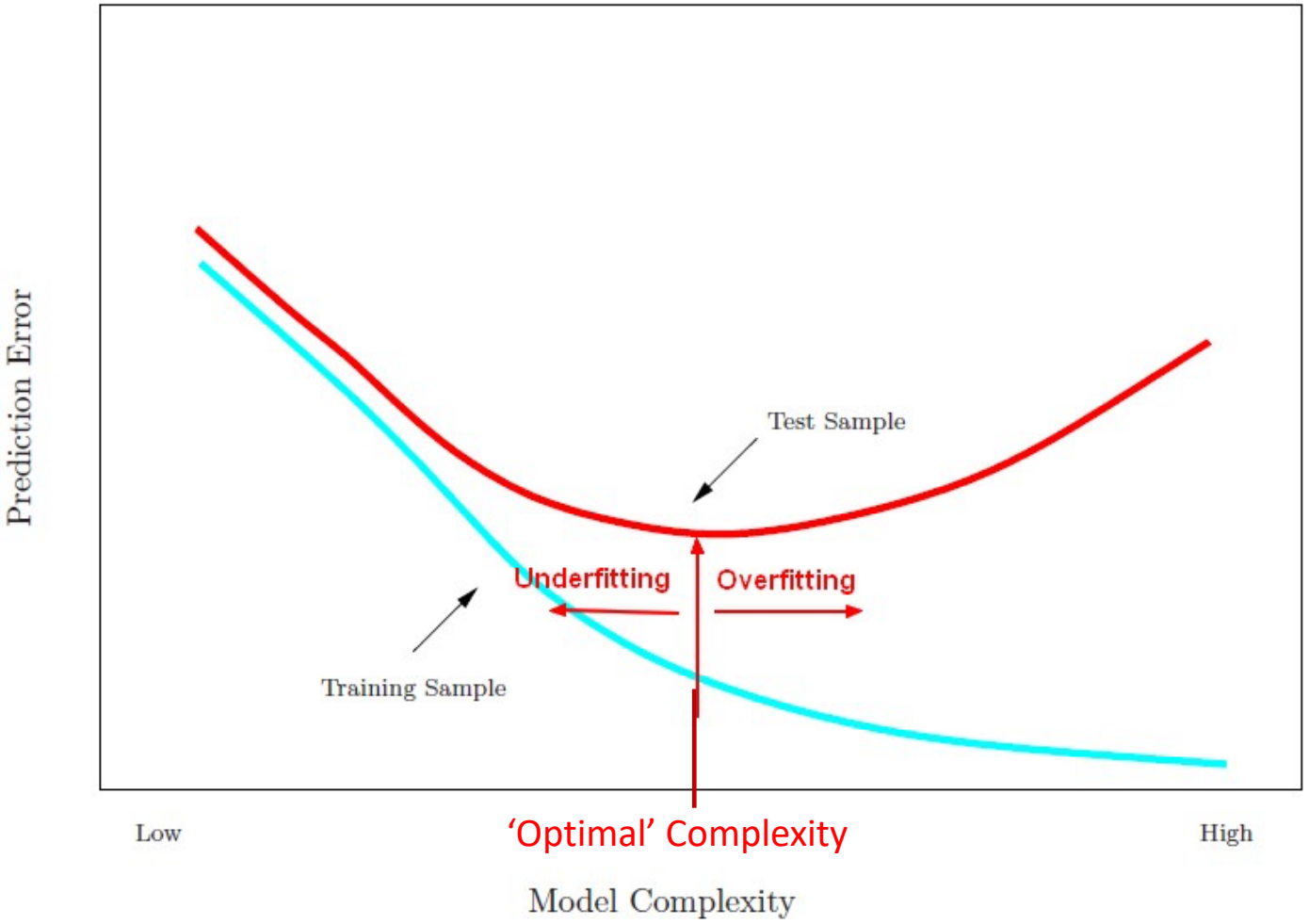
Overfitting vs. Underfitting

Learning ≠ memorizing your examples!

Machine learning ≠ just curve fitting!



Overfitting vs. Underfitting



Avoiding Overfitting

Get **more data**

Add some **noise** in data / optimization

Regularization: Penalize complexity (**Occam's Razor**)

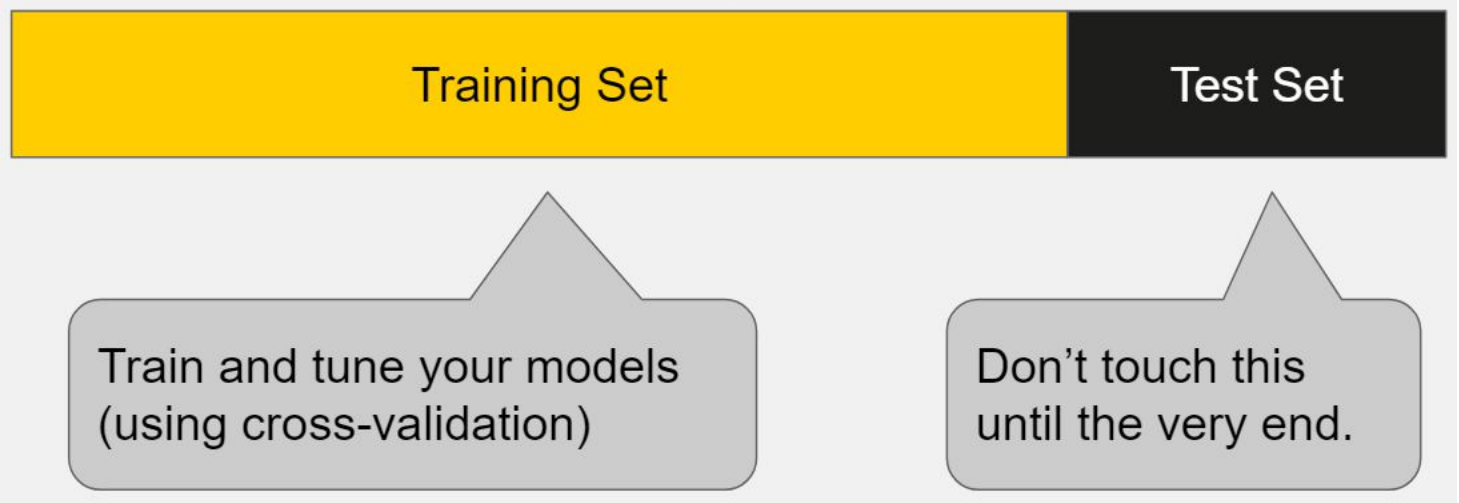
Hyperparameter Optimization:

ML algorithms allow **adjusting complexity of model**

Some do so 'on their own' (**minimal hyperparameter tuning**)

Model Evaluation

By now clear that:



Metric depends on **problem characteristics**

e.g. classification: accuracy

But if imbalanced / cost-sensitive: Precision, Recall, AUROC, FDR
Expected Cost, Neyman-Pearson,…

Simplicity, robustness, interpretability, computational cost, …

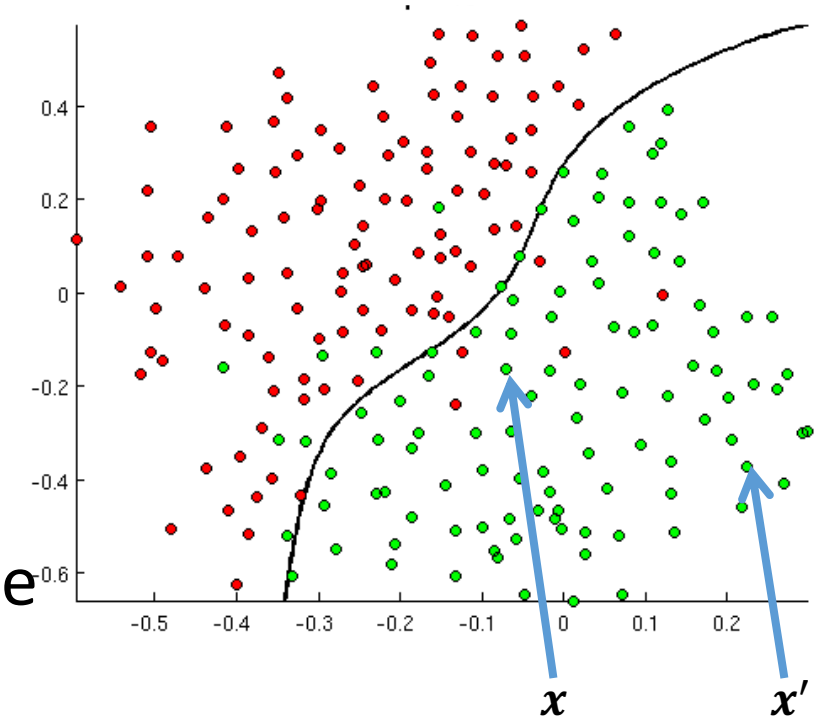
Classification Algorithms Build Models to…

Classify examples

Is x a dog?

Rank examples

Is x 'more of a dog' than x'?



Output a **score** for each example

'How much of a dog' is x?

Output a **probability estimate** for each example

What is the (estimated) probability that x is a dog?

Identify what **features contributed most to the prediction**

'What makes you think x is a dog? '

Can I Use ML in my Research?

Want to **learn rule for mapping inputs to desired outputs / finding structure in data**?

Have reason to **believe such a mapping / structure exists**?

Is mapping /structure **difficult to capture by a program / analytical solution**?

Do you **have / can you collect / simulate data**?

Do you have a way to **verify / evaluate your model**?

Then Yes!

Which ML algortithm should I use?

Type of problem?

Type of data?

Amount of data?

Computational resources?

Assumptions can/should make?

Missing data? **Outliers**?

Ultimately, **best choice is dataset specific**; my suggestions:

Structured: **Deep Learning (**e.g. **CNNs** for images**)**

Unstructured: **Gradient Boosting**, **Random Forests**

Low data scenarios: **SVMs**, **kNN**

1st approach: **Logistic/Linear regression**… **START SIMPLE!**

Where do I start?

Lots of **tutorials**, **MOOCs**, **books** (feel free to ask for suggestions!)

Talk to us – we are interested in applying ML to problems in the wider area of Physics!

Hands-on experience:

- Programming Languages: Python, R, Matlab, …
- Specialized libraries: Scikit-learn,
  Tensorflow, Pytorch, Keras (for Deep Learning)

Thank you!