Scikit-Learn Workshop Series

This workshop will begin at 13:02

Materials for today:

To work on your own computer:

- Go to https://github.com/nuitrcs/scikit-learn-workshop
- Click on the green Code button, choose Download ZIP, unzip the folder, open your IDE of choice and navigate to the folder you downloaded.

To work on the cloud:

- Go to https://github.com/nuitrcs/scikit-learn-workshop
- On the instructions page, click on the link that says "Google Colab Notebook: The Data Science Pipeline".

The Data Science Pipeline

efrén cruz cortés and Ritika Giri

Spring 2024

This workshop is brought to you by

Northwestern IT
Research Computing and Data Services

Got a machine learning, data science, statistics, or visualization question about your research?

We're here to help.

Go to bit.ly/rcdsconsult to request a FREE consultation.

LOGISTICS

- **Ask questions** in the Zoom chat – if you know the answer to a question, you can answer it (we will politely correct you if you're wrong).

- **If my internet goes out**, everyone gets a 5-minute break and we will meet back in the same Zoom room (shouldn't happen, but I like to have a plan).

Scikit-learn series

- April 22: Part 1 - The Data Science Pipeline
- April 29: Part 2 - Supervised Learning – Regression
- May 6: Part 3 - Supervised Learning – Classification
- May 13: Part 4 - Unsupervised Learning and Beyond

Warning!

When using Google Colab, you're sending your data to Google's servers.

We recommend that you **don't upload any information that you wouldn't publicly share** (data, code, etc).

It is okay to use Google Colab for today's workshop.

The Data Science Pipeline

efrén cruz cortés and Ritika Giri

Spring 2024

Today's workshop

- Half will be concept heavy.
- Half will be coding heavy.

Why?

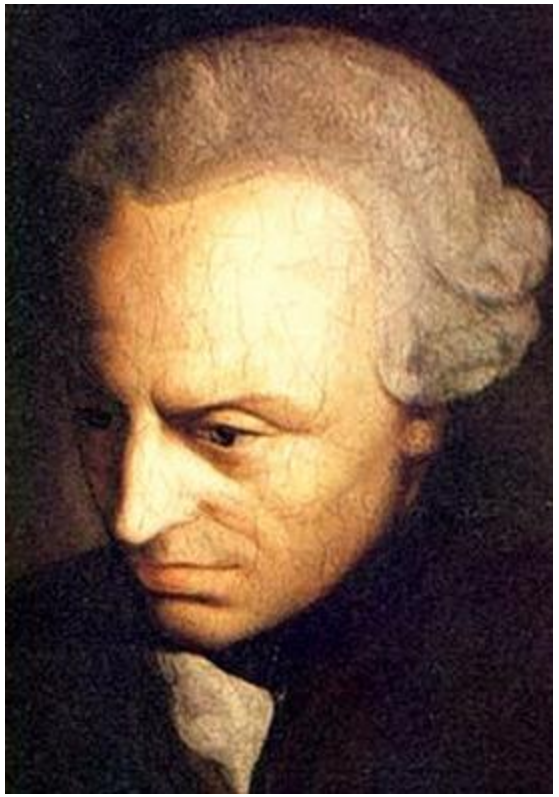
The rest of the sessions will be coding heavy.

What is learning?

How do concepts enter our mind?

That all our knowledge begins with **experience** there can be no doubt.

For . . . objects … affect our senses, and … produce representations … [and] rouse our powers of understanding … to compare, to connect, or to separate these, and so to convert the raw material of our sensuous impressions into a knowledge of objects, which is called [experience].

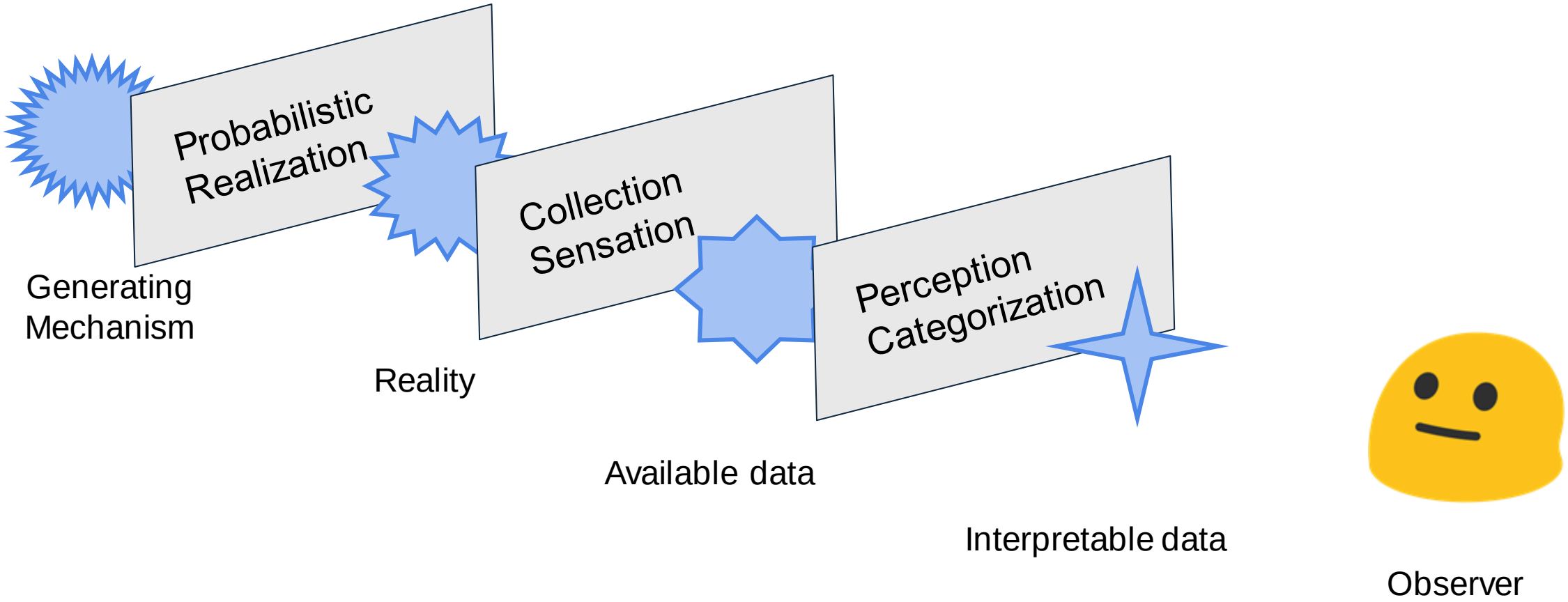


Immanuel Kant, Critique of Pure Reason

Toy model

- Data generating mechanism
- Probabilistic realization (including possible noise)
- Collection of data: through the body senses, through scientific apparatus (including possible measurement error)
- Abstraction of data for our understanding: make it into numbers, categories, ranks, etc.

Toy model



cruz cortés, “Machine Learning for the Perplexed”

Examples

- A person's health → set of specific measurements
- A disease → patient's symptoms
- Development of a cancer type → estimates of gene expression
- Economic standing of a community → individual's income
- Cultural dynamics → music preference
- Weather patterns → (Temp, Press, Wind Sp) at different locations
- State of the road → pixels representing ppl

How do we access the data generating mechanism?

1. After passing through the filters of sensation and perception, we make “observations” of the generating mechanism.
2. We assign attributes to these “observations”.
3. We make conclusions about what the generative law may be.

How do we understand the generating mechanism?

There are … two ways of investigating and discovering truth. The one hurries on rapidly from the senses and particulars to the most general axioms … For the mind is fond of starting off to generalities…

Francis Bacon, Novum Organum

How do we understand the generating mechanism?

The other constructs its axioms from the senses and particulars, by ascending continually and gradually, till it finally arrives at the most general axioms … The formation of notions and axioms on the foundation of true induction, is the only fitting remedy.

Francis Bacon, Novum Organum

Basis of evidence-based inference

- “Particulars” enter our understanding after a series of filters.

- From the particulars, we “build up” laws, from lower to higher levels of generality.

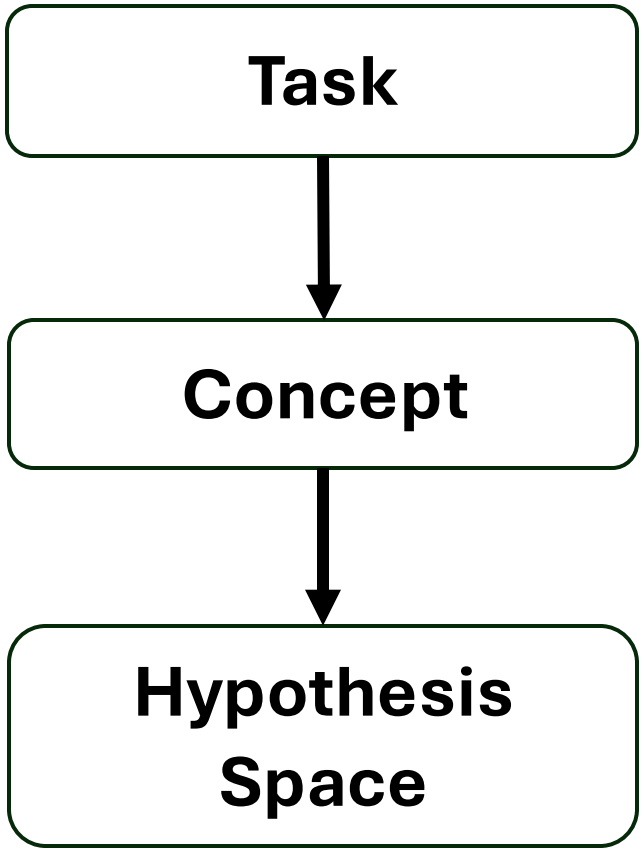
Example

1. We observe effects of a drug on a few patients.
    - Hypothesize effectiveness to overall population.
2. We test on larger and larger populations.
3. We test it holds over different conditions.
4. We find the general law: drug X cures disease Y.

What do we want to learn in ML

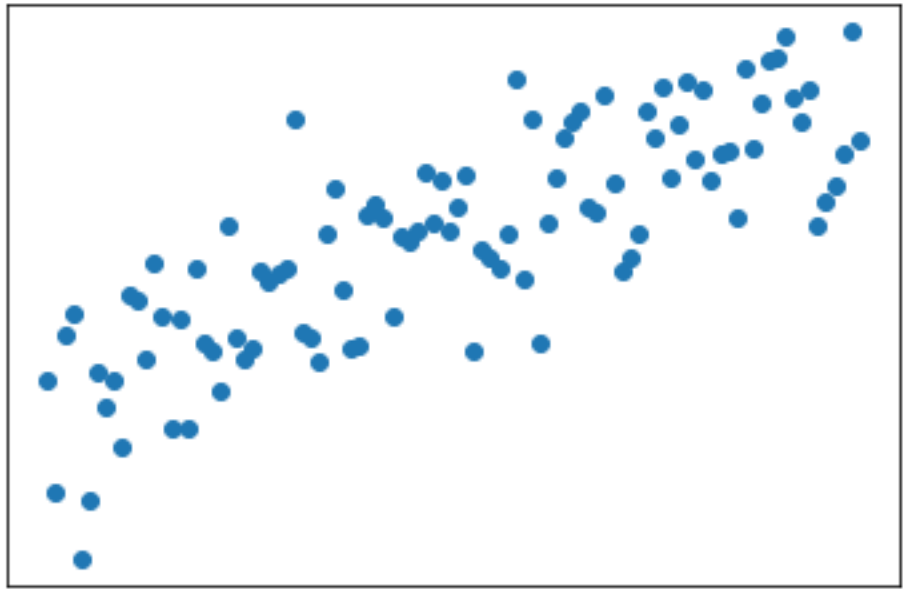
We will take the **task** approach:

- There is a task we want the machine to **do**.
- To perform this task, the machine must learn an underlying **concept**.
- We'll try to learn this concept by searching over a **space of hypotheses**.



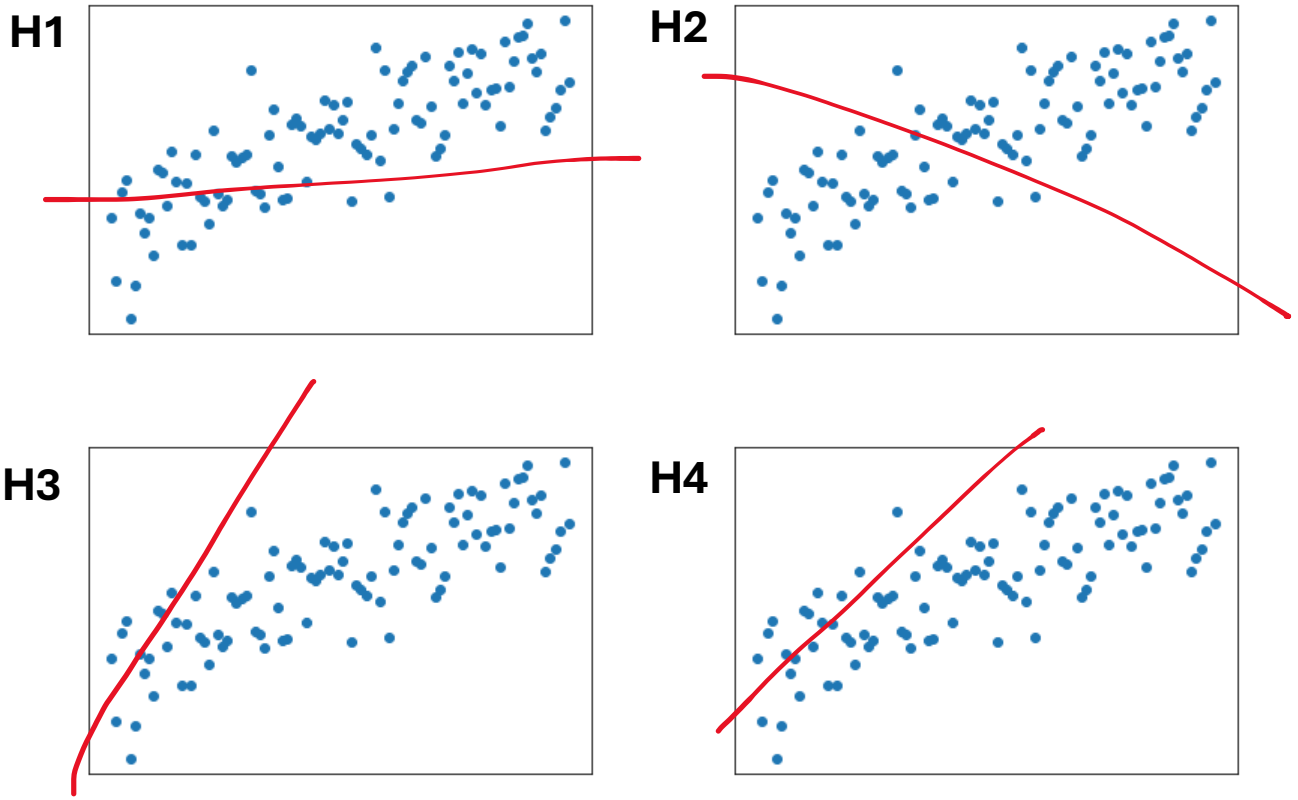
Learning a linear relationship

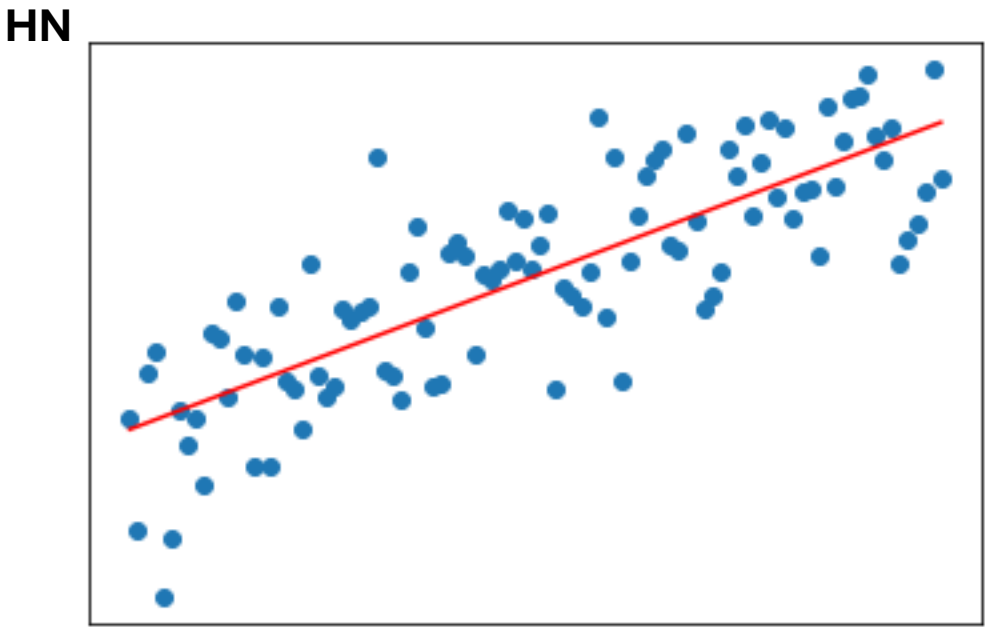
Imagine we observe the variation of one variable with respect to another. We presume the relationship is linear:



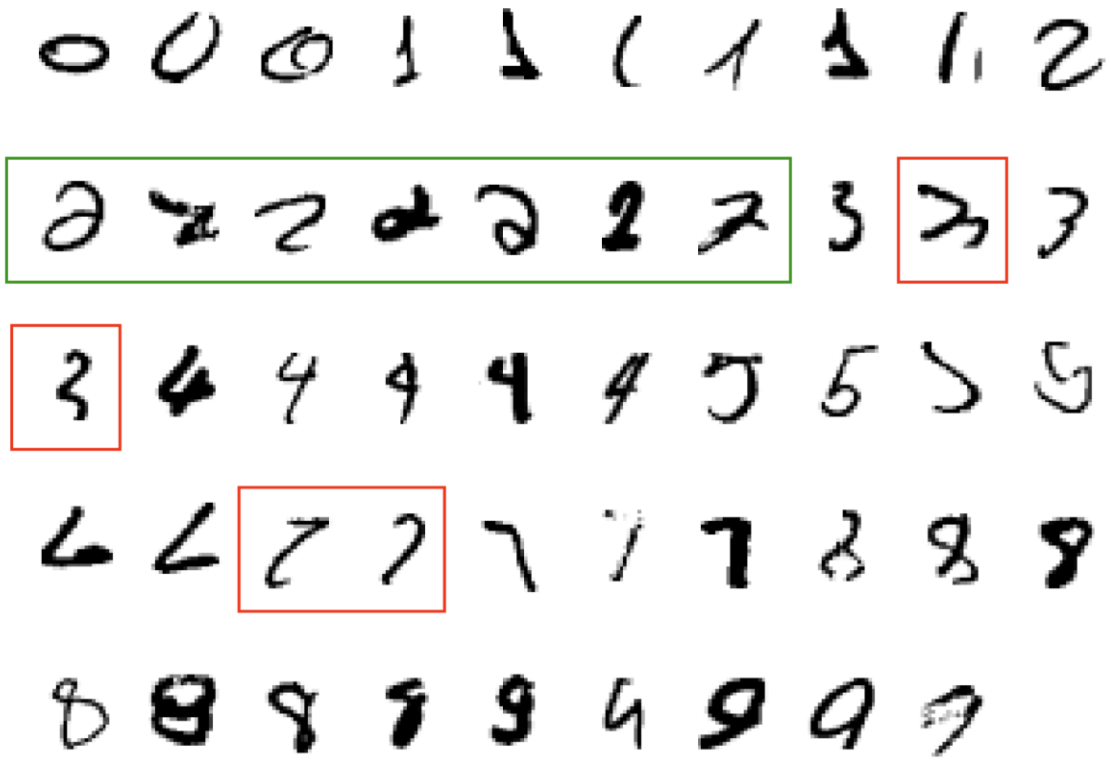
Learning a linear relationship

Our hypothesis space is the set of all straight lines:





A classic example of a task requiring ML



What makes a **2**?

Examples

Task	Concept	Hypothesis*
Recognize pedestrians	Pedestrian (image of)	Set of images representing pedestrians
Generate images from prompt	Entities in the prompt	Function from words to images
Identify a cancer	Cancer types	Subsets of genetic signatures
Allocate resources	Communities in need	Maps of (income, infrastructure) pairs to allocation amount.

* In practice take the form of functions: classifiers, regressors, generative models, etc.

When Do We Use Machine Learning?

- **ML is used when:**
  - Human expertise does not exist (navigating on Mars)
  - Humans can’t explain their expertise (speech recognition)
  - Models must be customized (personalized medicine)
  - Models are based on huge amounts of data (genomics)

- **When you should not use ML:**
  - There is no need to “learn” to calculate payroll.
  - When it is unethical!

More examples of ML tasks

- **Recognizing patterns:**
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- **Generating patterns:**
  - Generating images or motion sequences
- **Recognizing anomalies:**
  - Unusual credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant
- **Prediction:**
  - Future stock prices or currency exchange rates

Types of Learning

We can bundle many of the previous examples into types:

- Supervised Learning → Provide labeled examples
  - Classification, Regression
- Unsupervised Learning → Provide unlabeled examples
  - Dimensionality Reduction, Clustering
- Reinforcement Learning → Interact and learn from environment
- Latent Space Learning → Generative AI

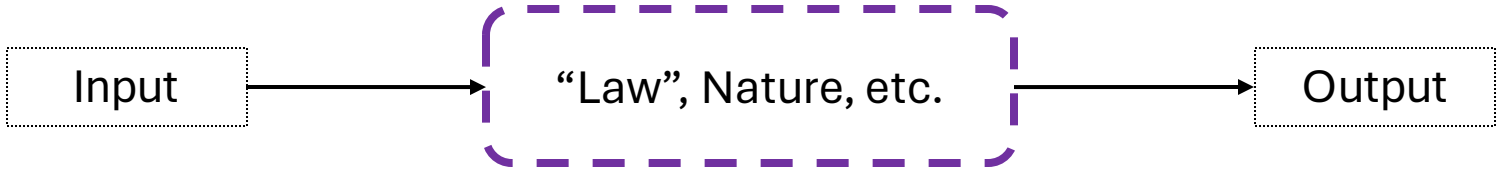
In reality, these paradigms are not strictly separated, but it is a useful categorization.

In this series

- Session 2: Supervised Learning – Regression.
- Session 3: Supervised Learning – Classification.
- Session 4: Towards unsupervised learning and beyond.

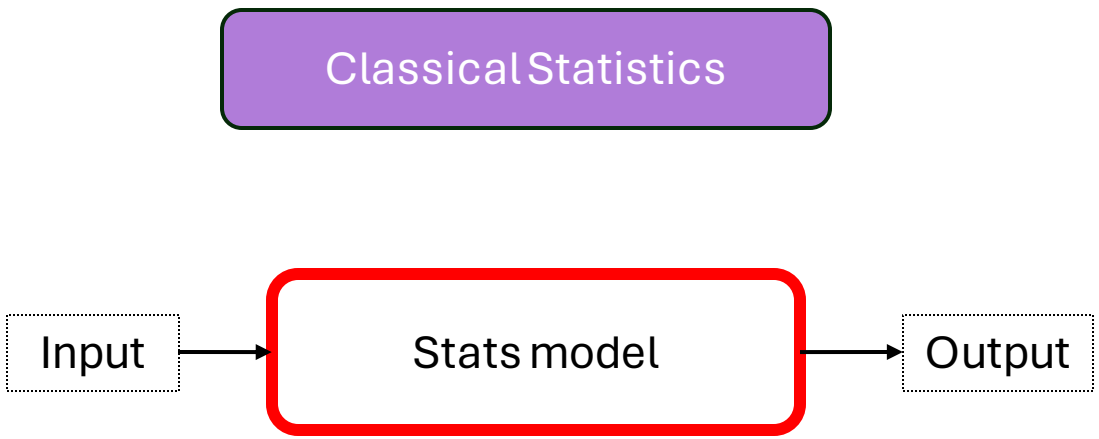
What's new about machine learning?

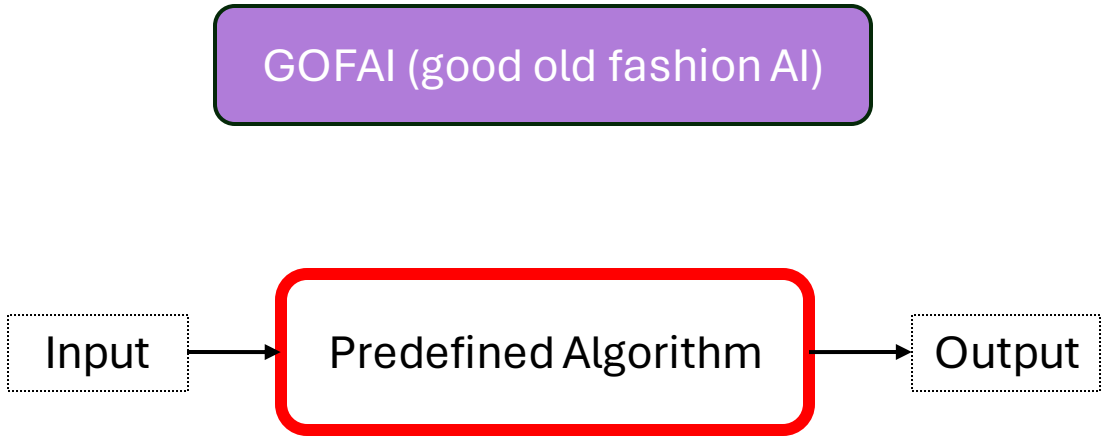
Say we want to capture the following relationship:



See Leo Breiman's "The Two Cultures"

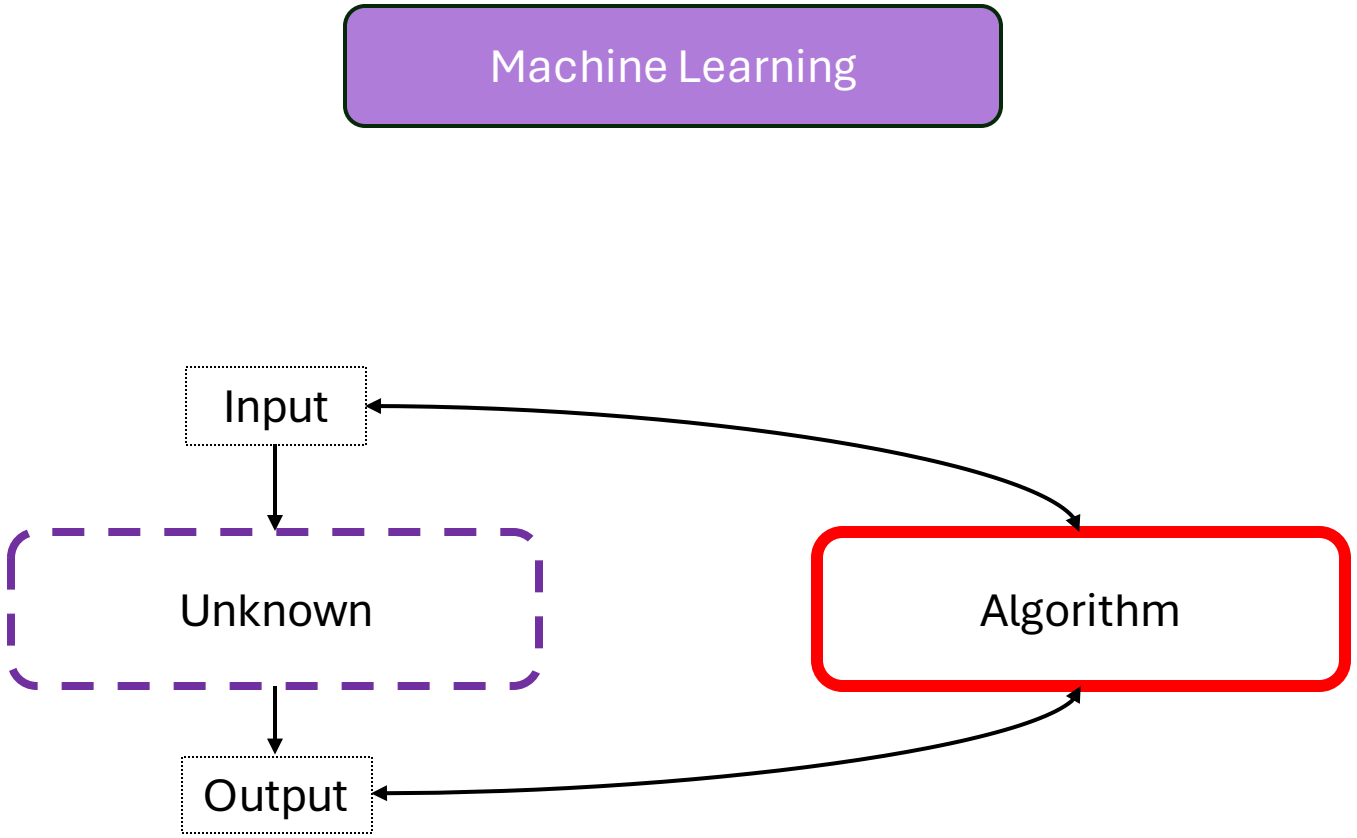
What's new about machine learning?





See Leo Breiman's "The Two Cultures"

What's new about machine learning?

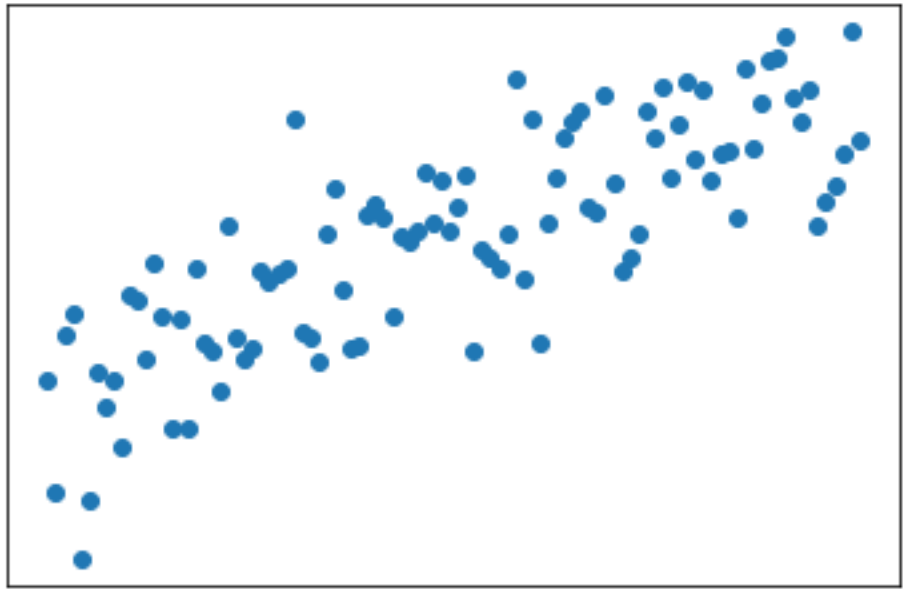


See Leo Breiman's "The Two Cultures"

How is learning achieved?

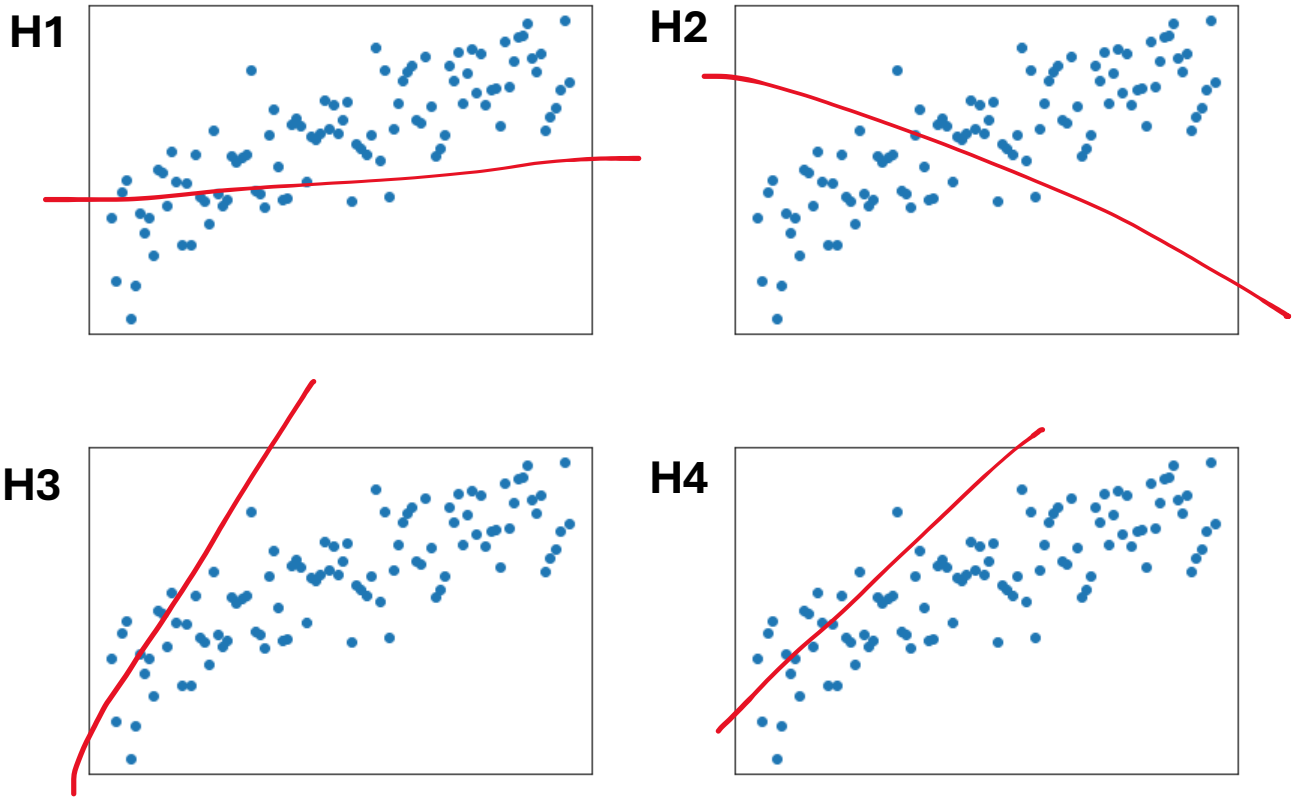
Learning a linear relationship

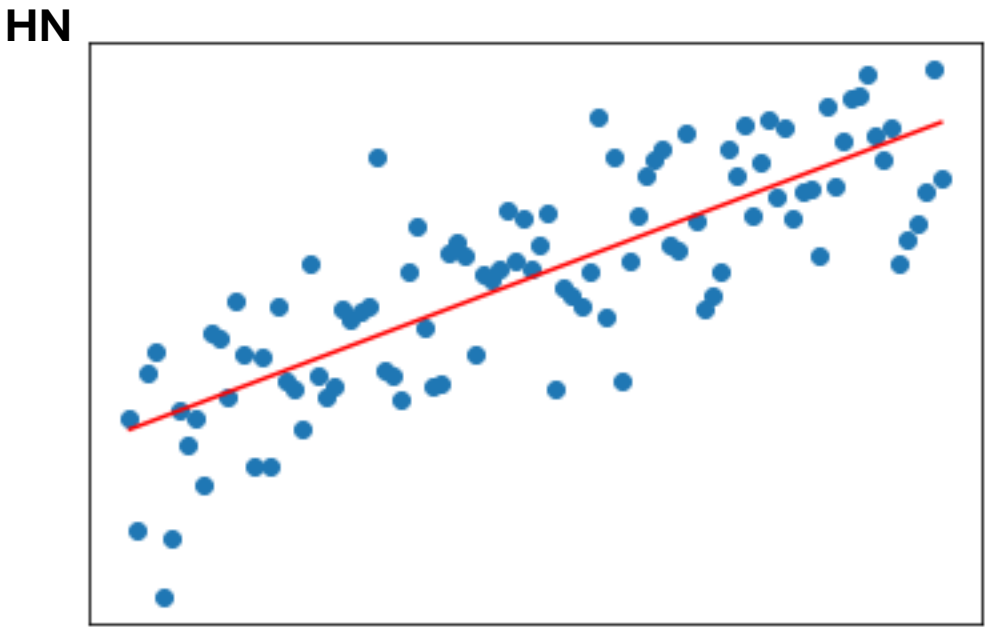
Imagine we observe the variation of one variable with respect to another. We presume the relationship is linear:



Learning a linear relationship

Our hypothesis space is the set of all straight lines:

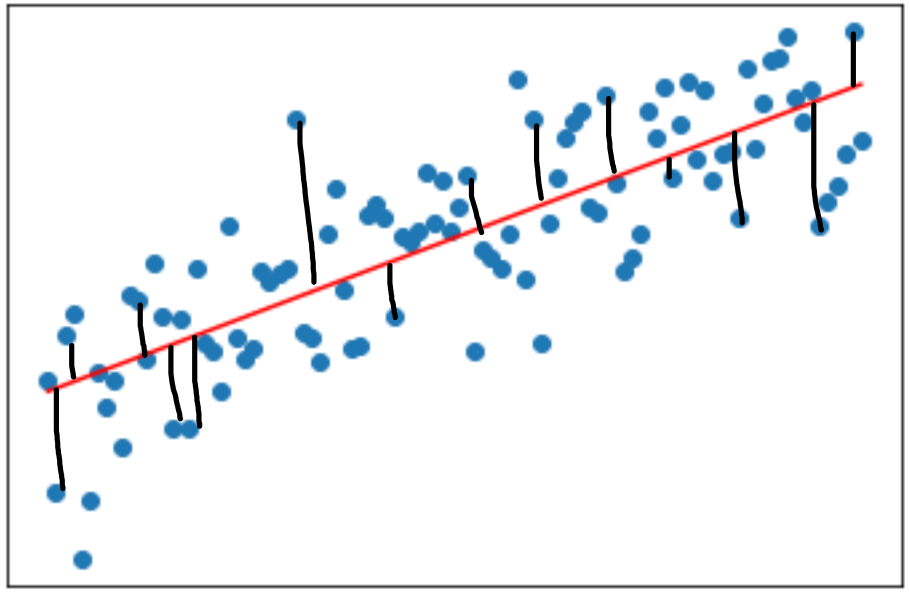




Which hypothesis is better?

Quantifying **error**: how close is each line to our observation?

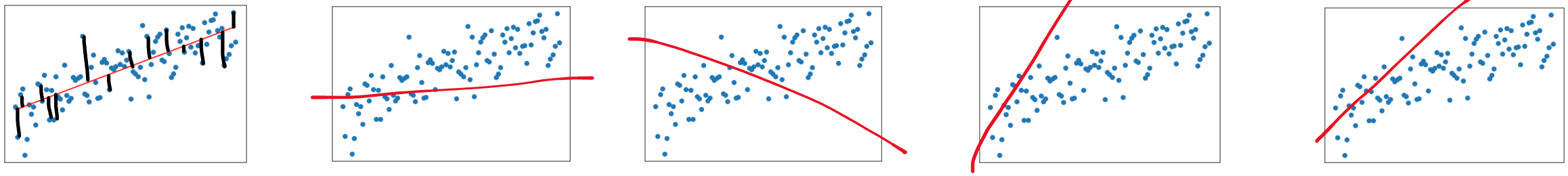
As measured by … ? → Sum of individual deviations?



Quantifying error

Adding up all individual errors will give you **one number**.

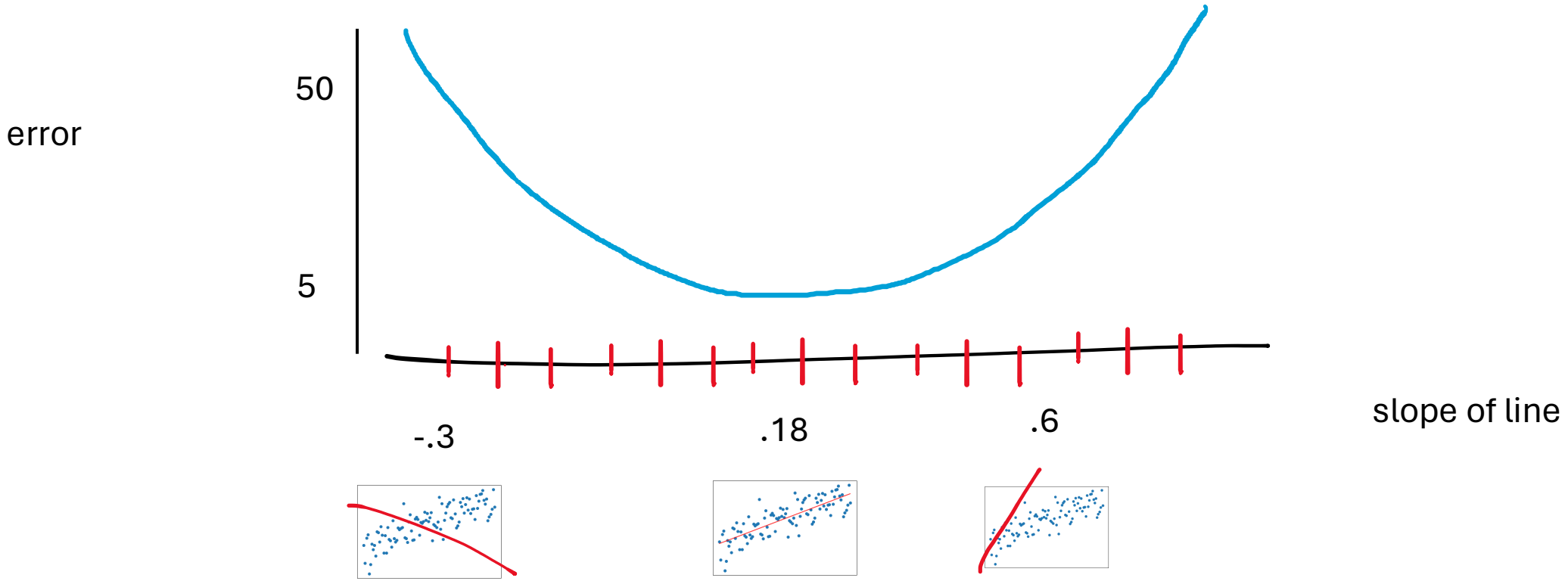
→ Pick the line that minimizes that number!!



5 10 50 15 10

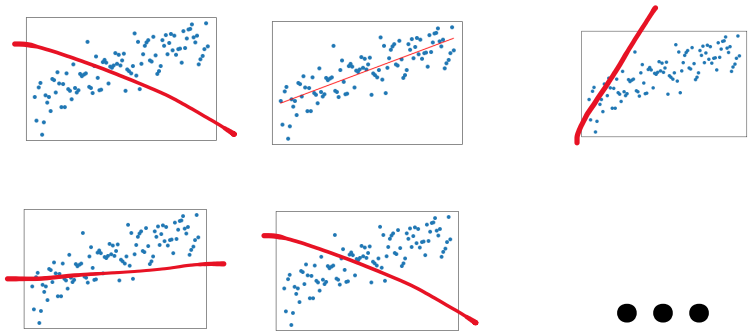
Finding the best line

Remember: we can fully determine each line by knowing its **slope.**

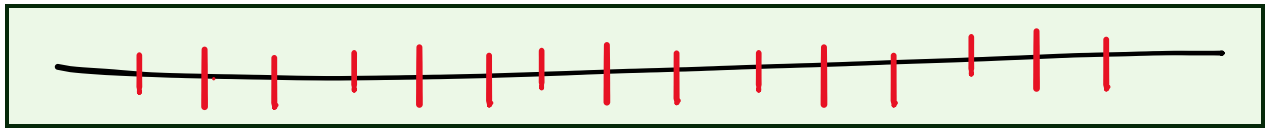


Hypothesis Space

Note how we have moved from space of all straight lines to one dimension of “slopes”:



Hypothesis space: set of all straight lines



Amenable version of Hyp. Space:
The real line (set of all slopes)

General learning principle

- Describe your Hypothesis space as a mathematically tractable space.
- Formulate a way to quantify error.
- Find the hypothesis that minimizes your error!

Choices

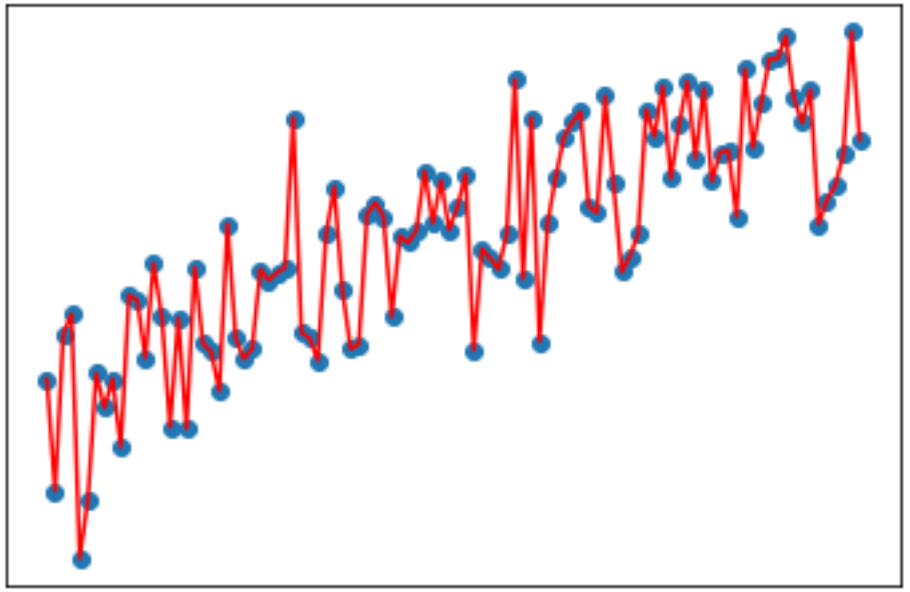
- Choosing the error is not trivial, but there are well known error types you can use for most tasks.

- Choosing hypothesis space is also not trivial.
  - Why did we choose set of straight lines instead of set of all possible lines?
  - Thoughts?

Should we perfectly fit our observations?

We run the risk of overfitting!!

- While the line minimizes the error…
  - It is unlikely it will match any new observation.



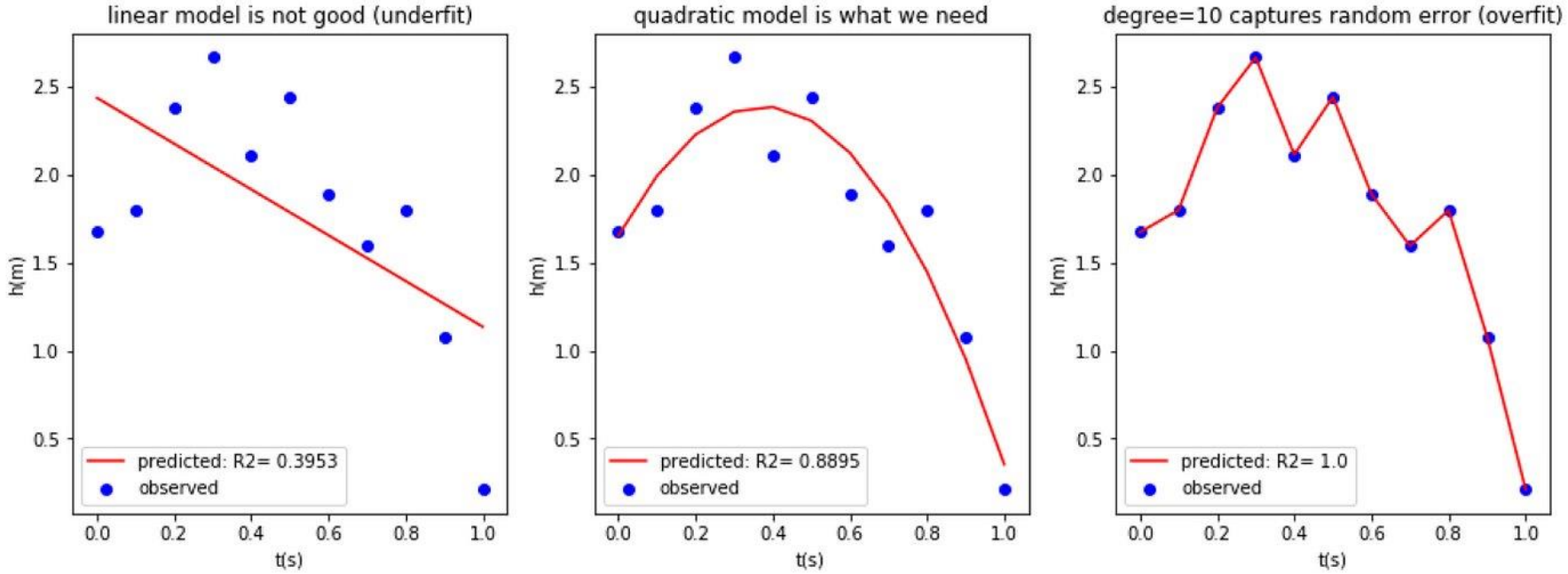
We must leave space for new experiences.

There are … two ways of investigating and discovering truth. The one hurries on rapidly from the senses and particulars to the most general axioms … For **the mind is fond of starting off to generalities**…

Francis Bacon, Novum Organum

Example:

- Say our data looks like the blue dots.
  - **(1)** A straight line is too simple of a hypothesis. Error is large.
  - **(3)** A complicated line relies too much on specific data, will fail with new data. Error is zero.
  - **(2)** A quadratic line is the perfect middle ground: gives space for new observations, keeps the error small.



Generalization

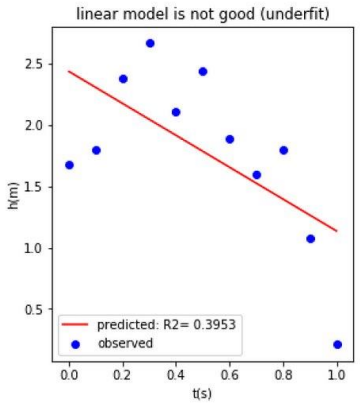
The problem of finding the middle ground between under and overfitting is called the **bias-variance trade-off**.

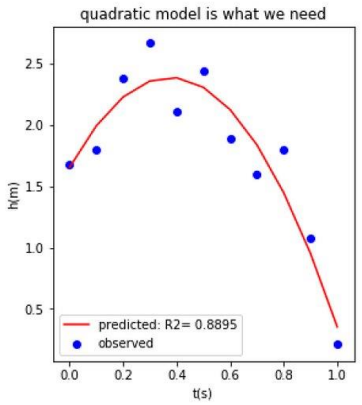
- A simple model will be biased towards a specific form, and it will vary little with observations (like a hard-headed person).
- A complicated model will have low bias, but vary widely with observations (like someone who keeps modifying and embellishing their theories just to be right).

- A good balanced hypothesis will have the power of **generalization**: it will, with some small error, generalize to new observations.

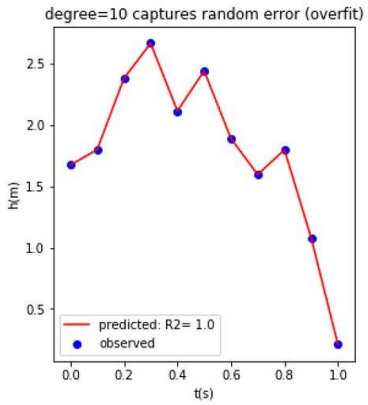
Bias-Variance trade-off

Underfitting





Overfitting



High Bias
Low Variance

Low Bias
High Variance

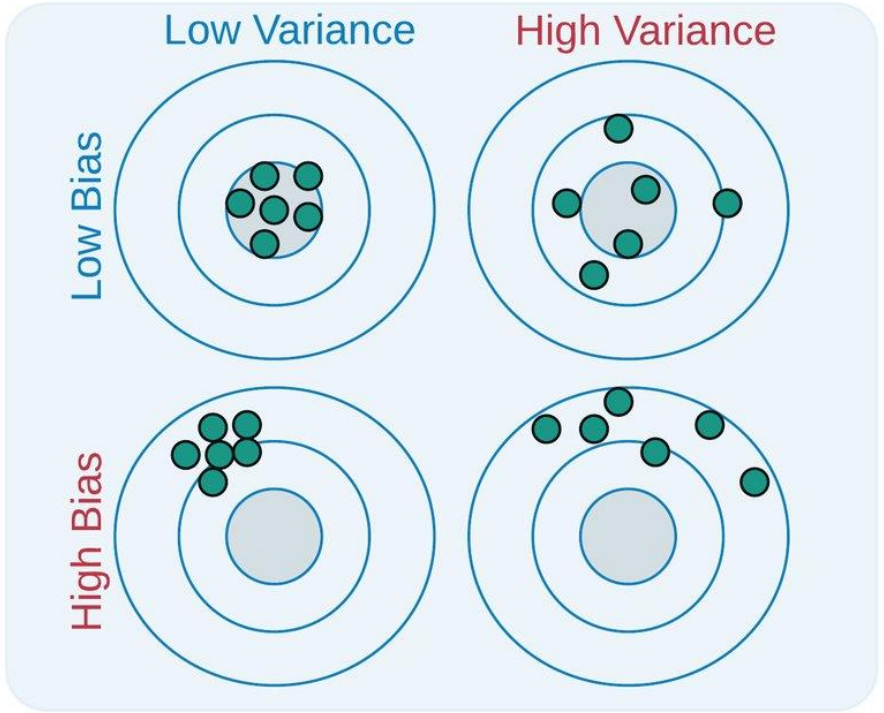
Bias:
Systematic prejudice in the model
Simple model = High bias

Variance:
Change in the model's prediction, when the dataset is changed a little bit
Complex model = High variance

Bias-Variance trade-off

Imagine different research groups get different samples of the same phenomenon. Each **green dot** represents the found hypothesis **H** for different sets of observations.

The **grey** middle circle represents the optimal point.



High Bias, Low Var:

- The **H**s will be similar, but off from the optimal point.

High Variance, Low Bias:

- Some **H**s will be close to the optimal point, but it varies, widely.

Low Bias and Variance:

- While ideal, many times impossible. Often there are fundamental limitations.

Validation

Notes on terminology

Note we have used the word “hypothesis” in a different way than classical statistics.

From now on we will use “model” instead of “hypothesis”.

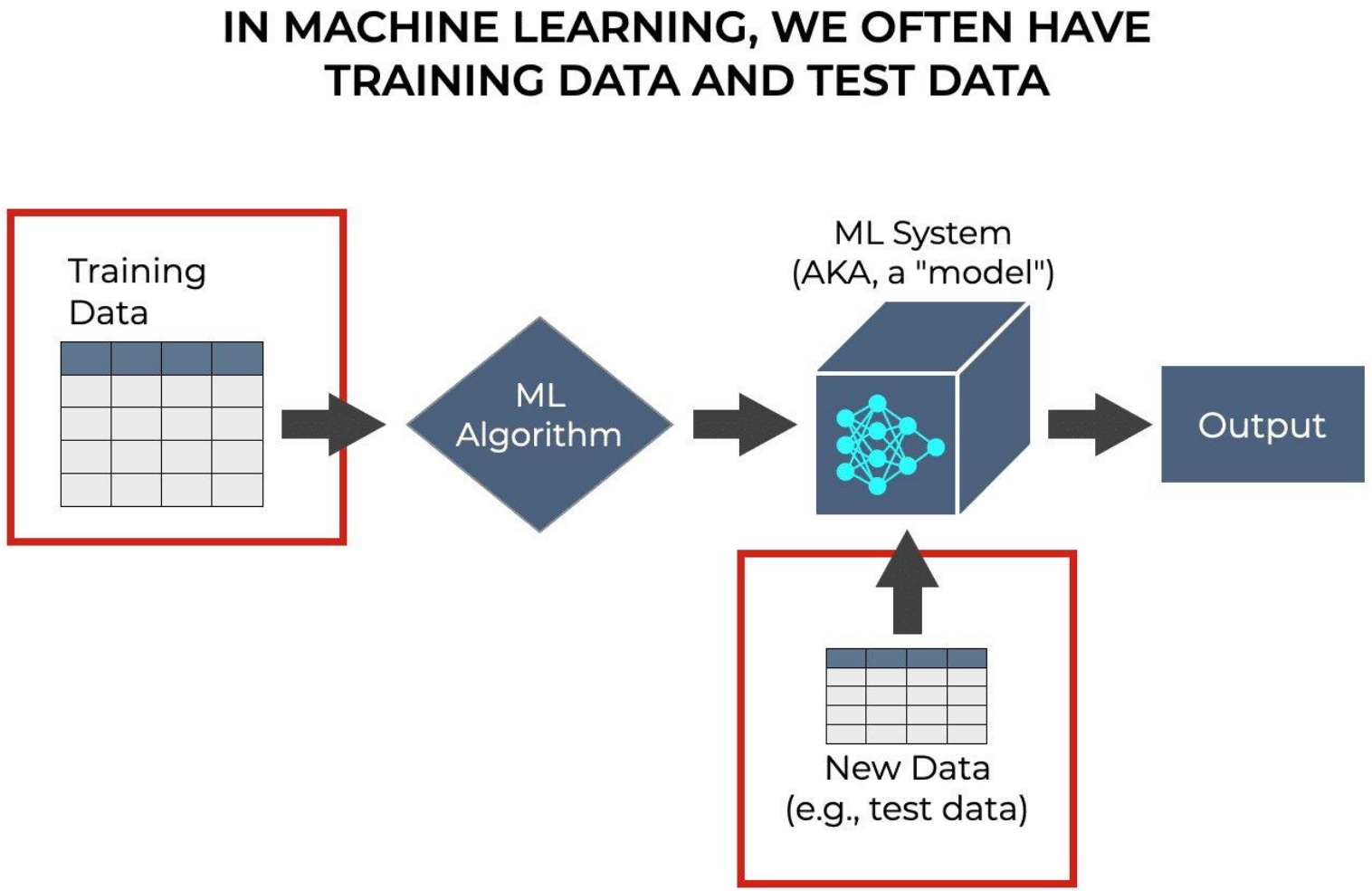
How to find a generalizable hypothesis?

- Instead of minimizing the error on our observations:
  - Save a portion of the data and keep it secret, unobserved.
  - Formulate the model.
  - Test the model on the secret data.
- This procedure helps us build models with good generalization properties.

Training data: observed data, used to formulate our model.

Test data: held out, secret data, used to test the model.

Train-Test split



The ML Pipeline

Summary

- Formulate a **task**.
- Obtain **interpretable data** (after passing through a few reality filters).
- Choose and search over a **hypothesis space** according to the principle of **generalization**.
  - One way to accomplish this: train-test split.

Machine Learning workflow.

We left a few gaps, which we will fill in the following sessions. Here is a possible set of steps in the ML pipeline:

1. Should I use ML on this problem?	Is there a pattern to detect? Can I solve it analytically? Do I have data?
2. Gather and organize data.	Preprocessing, cleaning, visualizing.
3. Establishing a baseline.	
4. Choosing a model, loss, regularization, ...	
5. Optimization (could be simple, could be a PhD...).	
6. Hyperparameter search.	
7. Analyze performance & mistakes and iterate back to step 4 (or 2).	

And with that…

On to scikit-learn!