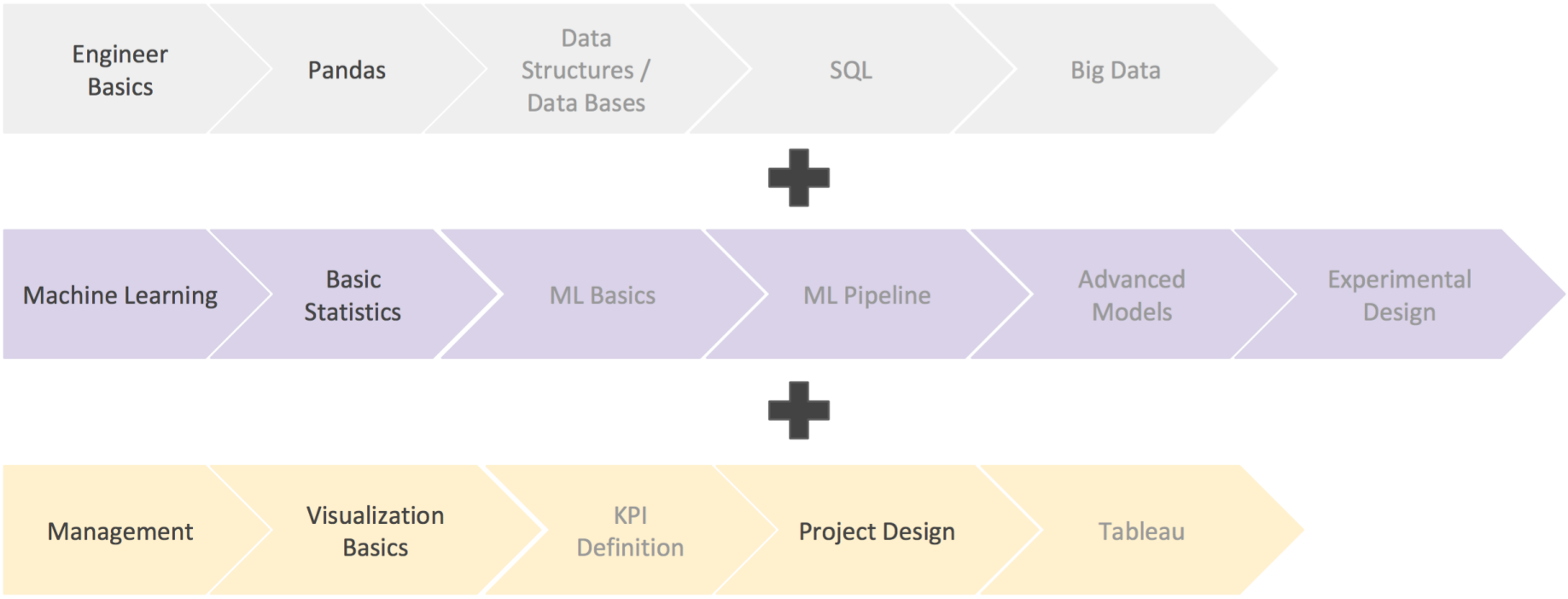
Data Science with Python

February 26, 2018

Topics of the day

- Syllabus of this course
- What is machine learning?
- A first model $\rightarrow$ linear_classifier.ipynb

Course overview

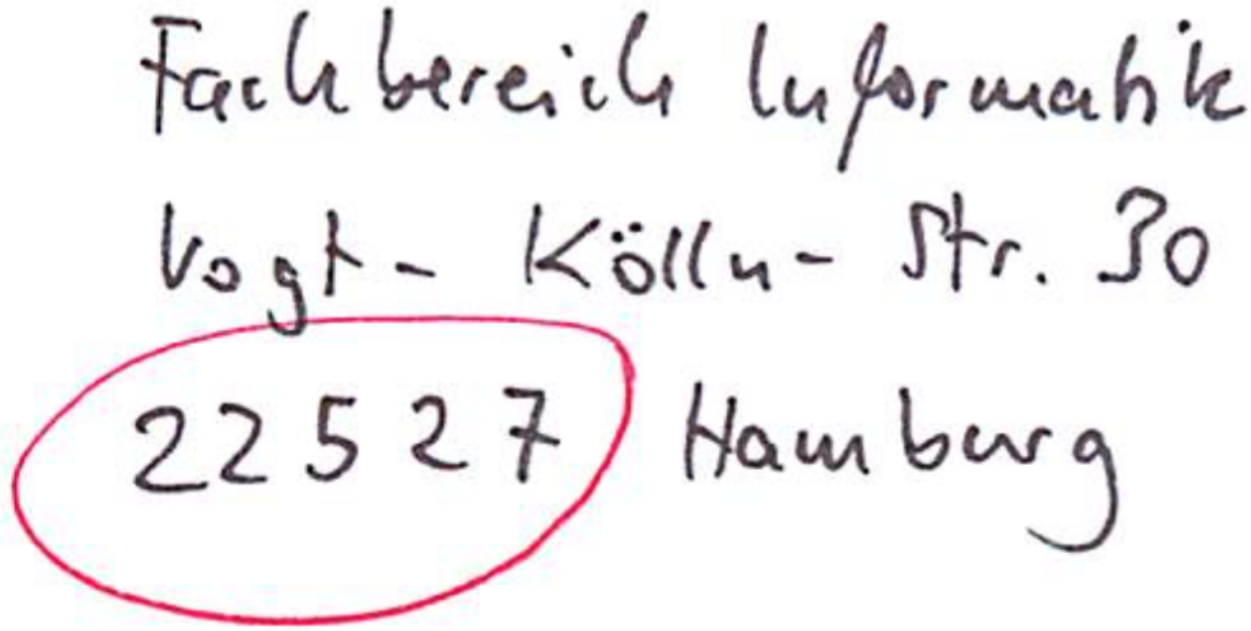


What is machine learning?

Machine learning is a set of methods to automatically detect patterns in data and to use those patterns to help in decision making.

from Murphy (2012)

Example: Digit recognition



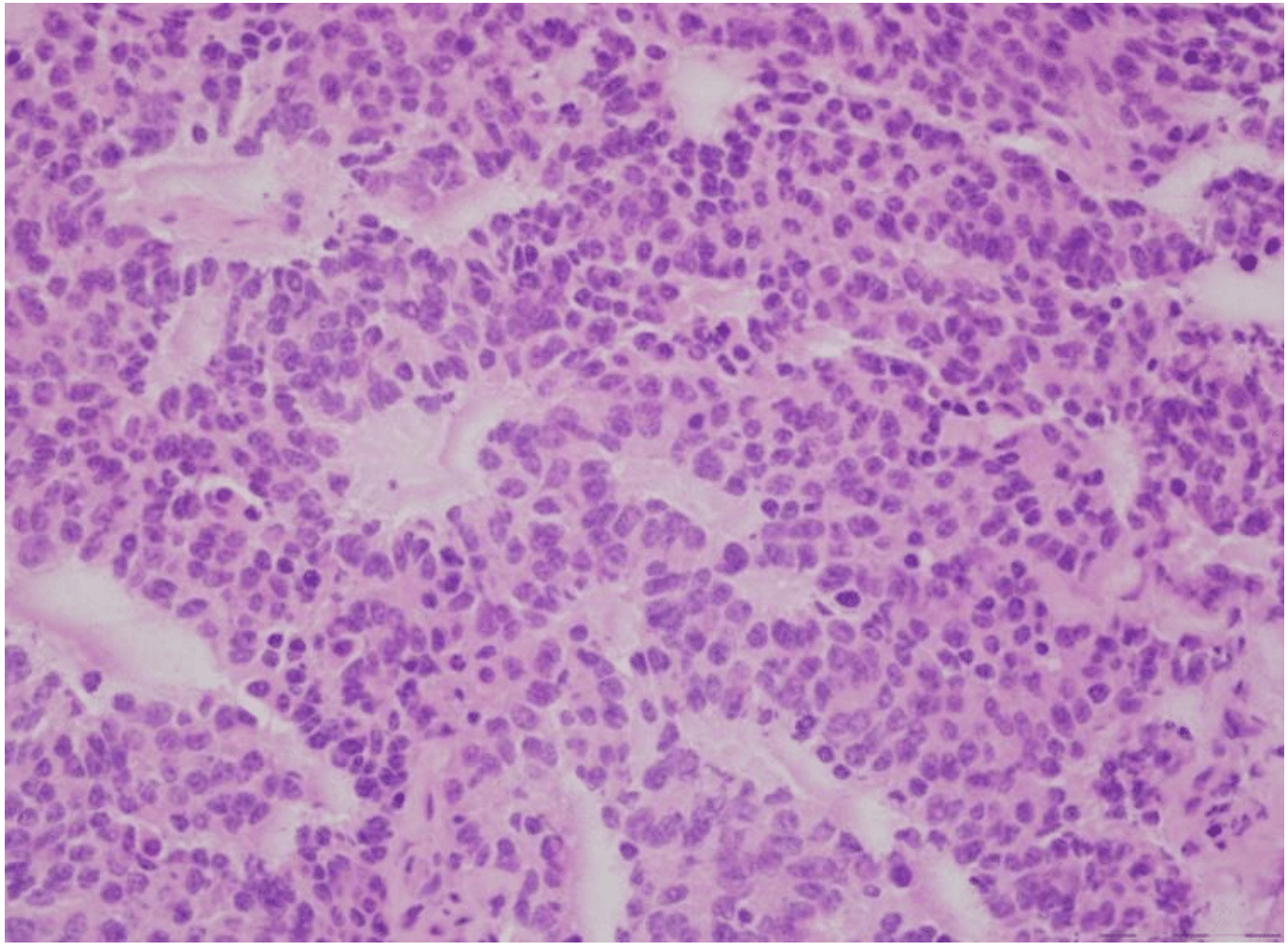
e.g., the MNIST data set

There is too much variation to design a rule by hand.

Example: Autonomous driving

https://www.tesla.com/autopilot

Example: Cancer detection



Automatic detection and classification of cell nuclei in microscopy images

Example: Machine translation

Skype Translator

Break down language barriers with your friends, family and colleagues.

Our online translator can help you communicate more clearly. Our voice translator currently works in 8 languages, and our text translator is available in more than 50 languages for instant messaging.

Skype Translator uses machine learning. So the more you use it, the better it gets.

Get Skype for Windows desktop

What is machine learning?

Machine learning is about "algorithms that allow a computer to learn specific tasks from training examples. [...]

Ideally, the computer should use the examples to extract a general rule" and not just memorize the seen examples.

from Prof. Ulrike von Luxburg's slides on ML

An algorithm is something like a recipe, a set of rules to follow for solving a class of (computational) problems.

What is machine learning?

How do you learn a rule from examples?

Assume your task is to distinguish different species of flowers from the genus Iris.

Iris Setosa

Iris Versicolor

Iris Virginica

What is machine learning?

How do you learn a rule from examples?

You could gather **samples** and measure different **features**/properties they have, for example the length and width of their "leaves" (aka petals and sepals) and see if that is useful for distinguishing them.

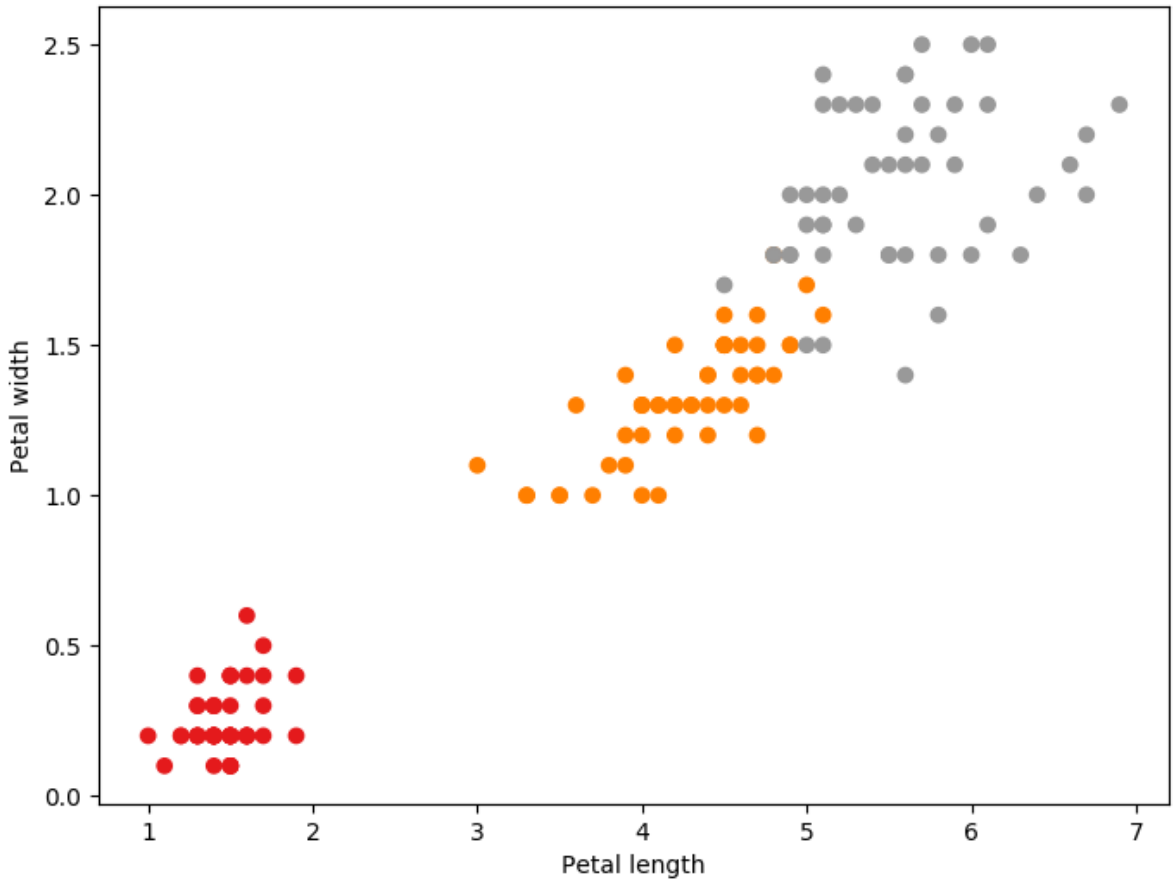
What is machine learning?

How do you learn a rule from examples?

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
```

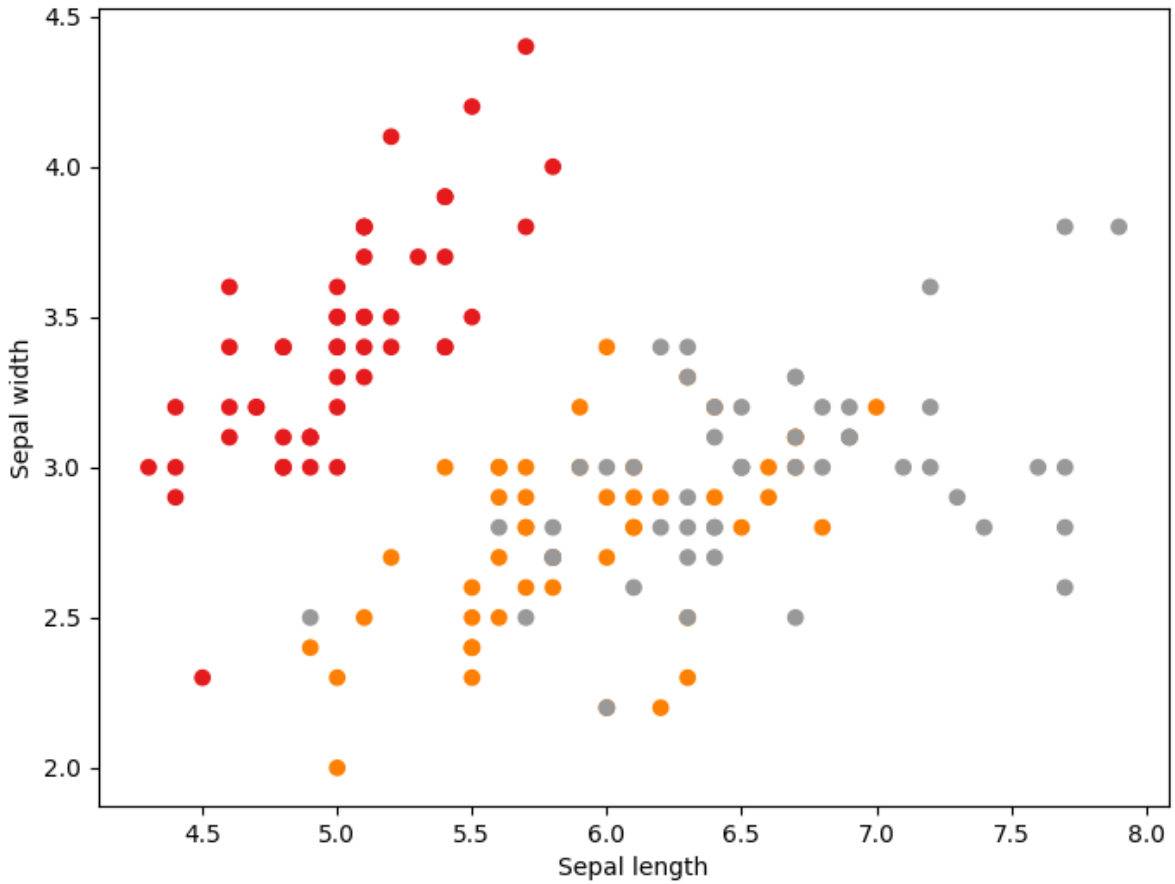
What is machine learning?

How do you learn a rule from examples?



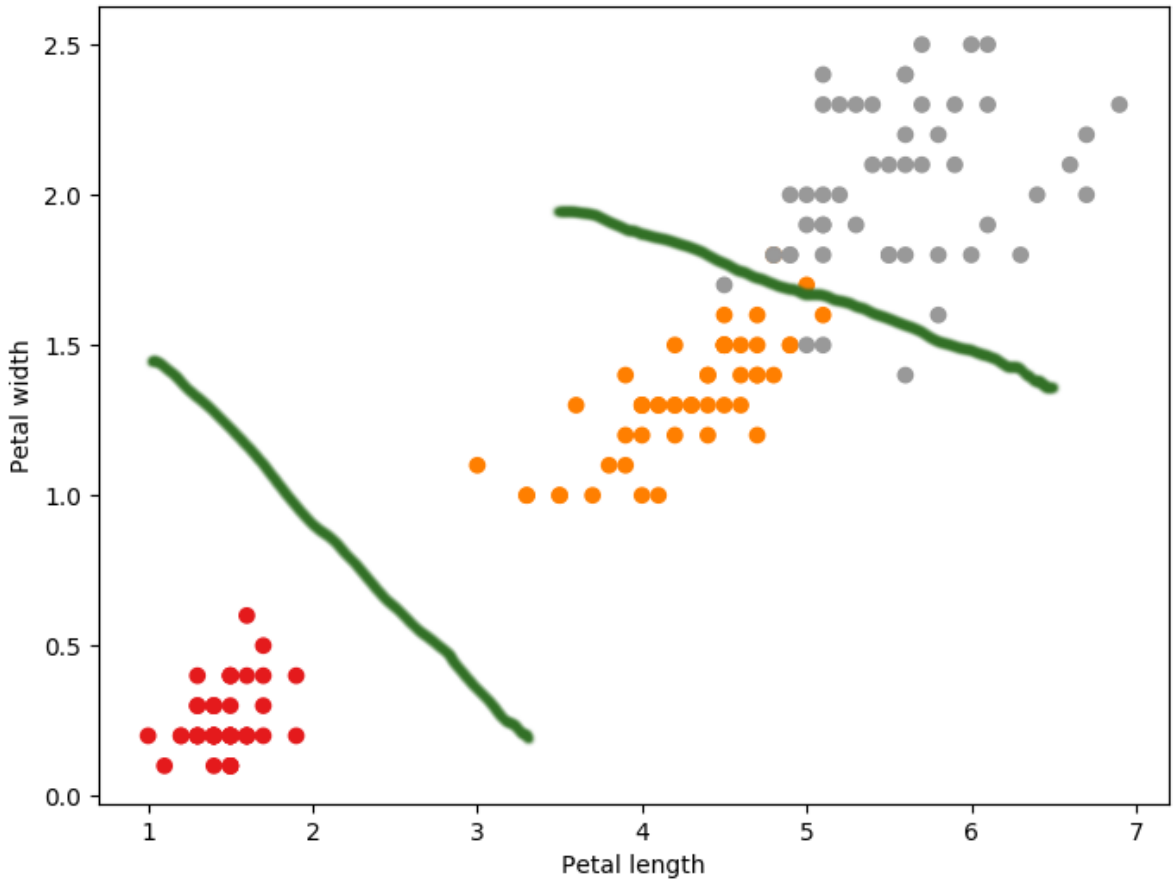
What is machine learning?

How do you learn a rule from examples?



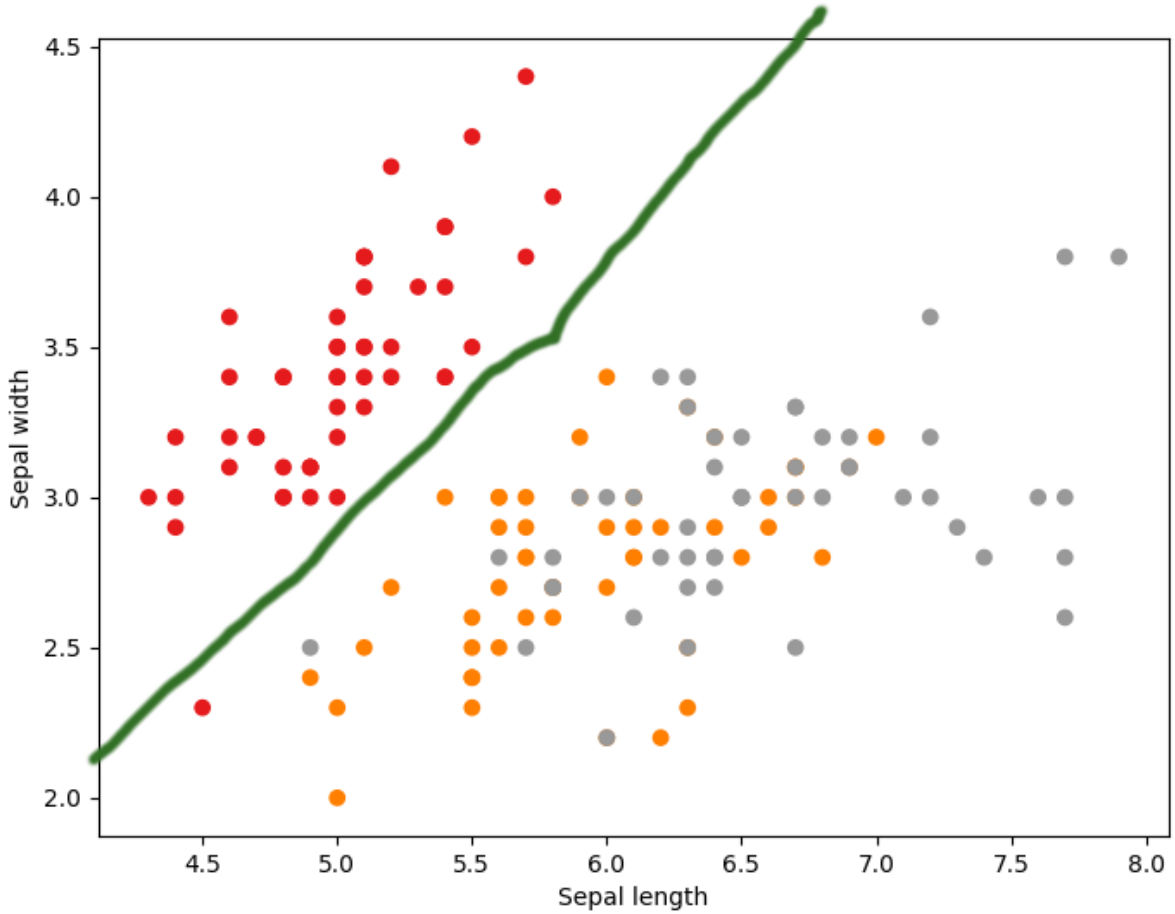
What is machine learning?

How do you learn a rule from examples?



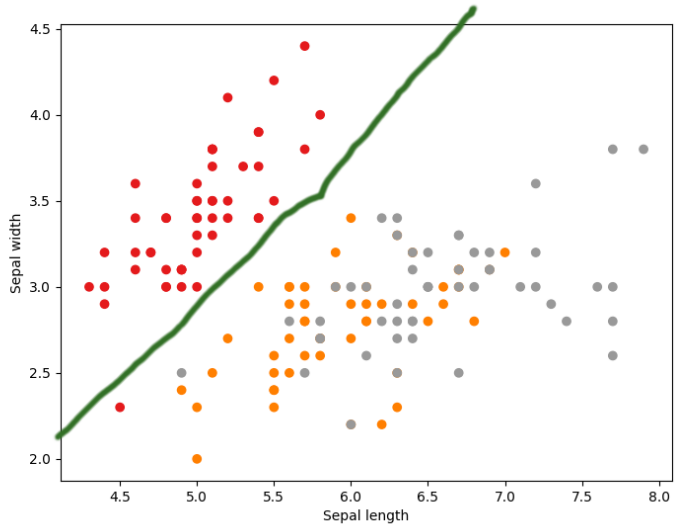
What is machine learning?

How do you learn a rule from examples?



What is machine learning?

How do you express such a rule such that a computer would understand?



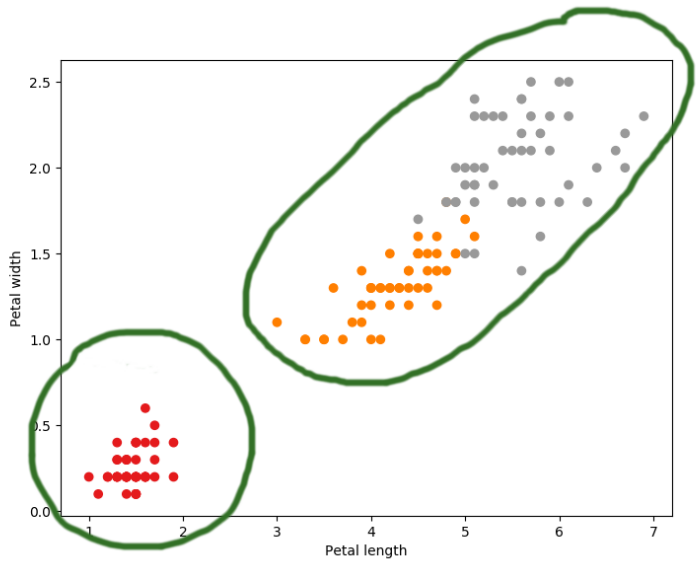
e.g., a linear equation: $y = f(x) = a \cdot x + b$

So the task of a computer could be to find a and b such that the resulting lines/rules separate the given examples into their resp. classes.

What is machine learning?

We call this type of machine learning in which we know which **class**/**label** each examples belongs to **supervised machine learning.**

There is also **unsupervised machine learning** in which the labels of samples are not known, so instead of **classifying** data we could try to **cluster** them.



Some further questions

- How do you know which of the features you have designed and chosen are useful? ← feature selection

  (Sneak preview: some more advanced models can learn features, so you don't have to manually design them.)

- How do you know which of a variety of possible rules that solve the task is the best one? How to measure how good a rule is? ← model evaluation, performance measures

- How do you tell what kinds of rules to try in the first place? E.g., straight line vs. parabola vs. etc. And how to tell when you have collected enough samples such that the desire rule can be learned at all? ← bias-variance tradeoff, overfitting, underfitting

A first model

→ linear_classifer.ipynb