VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

AIML231/DATA302 — Techniques in ML

Week 1

Machine Learning Overview

Dr Qi Chen

School of Engineering and Computer Science

Victoria University of Wellington

Qi.Chen@vuw.ac.nz

Week Overview

★ AI and Machine Learning

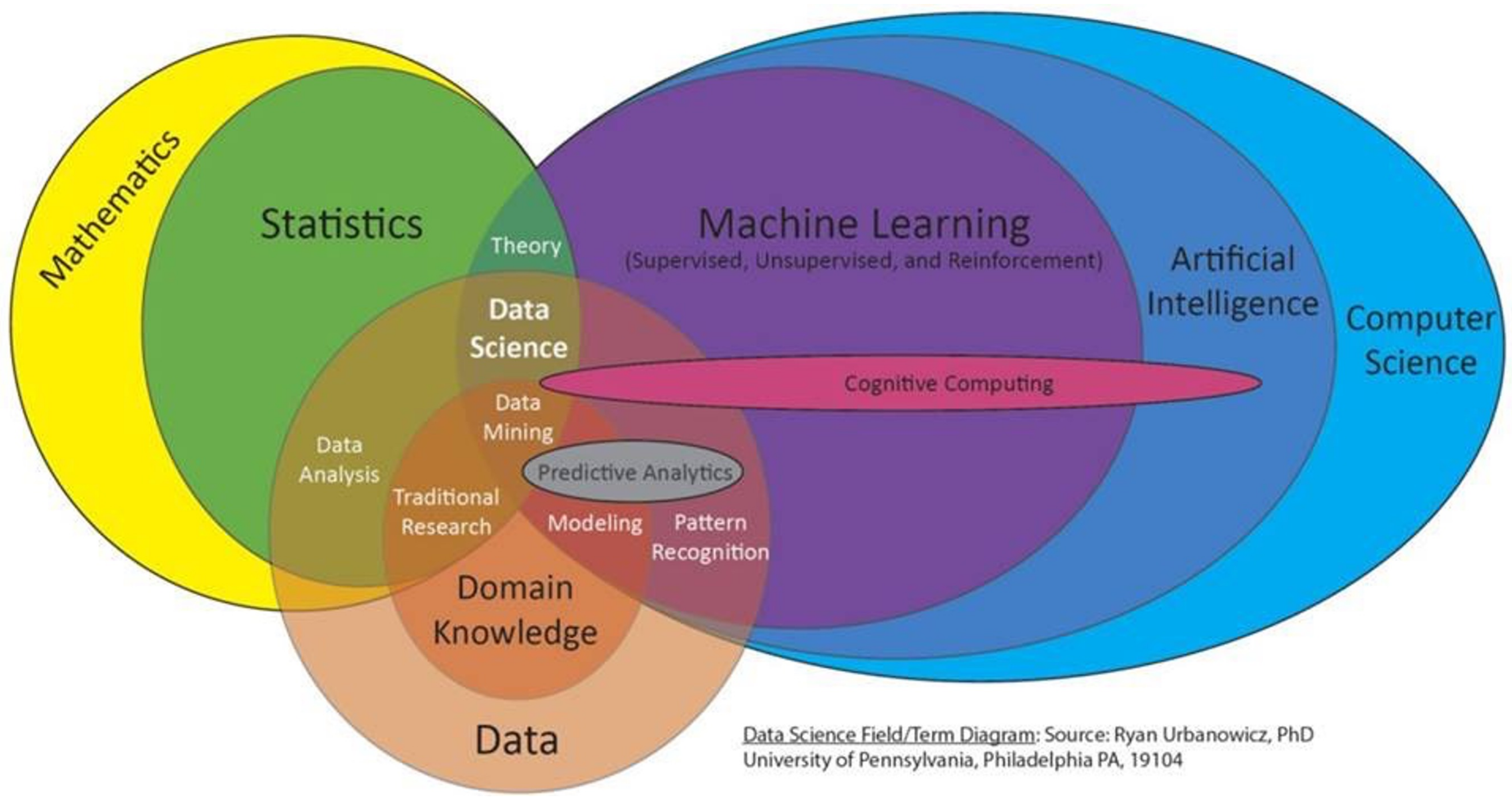
★ Machine Learning Scope: Data, Task, Model, and Algorithm

★ Data in Machine Learning

★ Machine Learning Tasks

Artificial Intelligence and Machine Learning

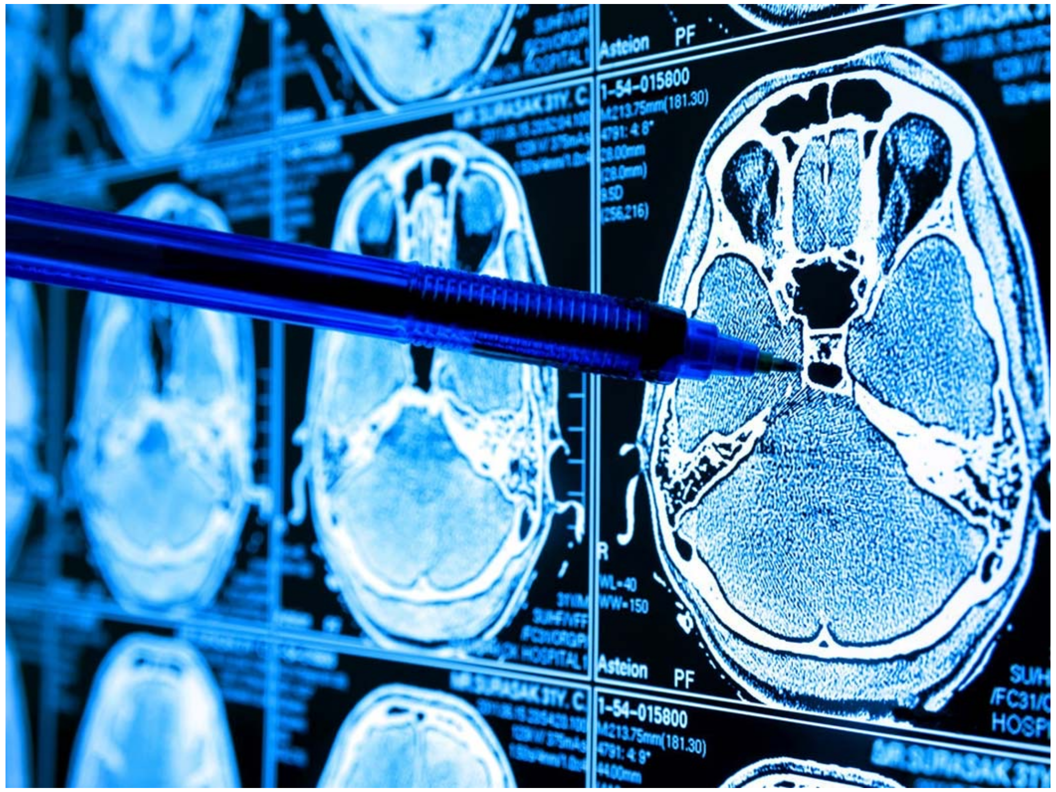
- Artificial Intelligence is the broader concept of machines being able to carry out tasks in a way that we would consider “smart”.
- Machine Learning is a current application of AI based around the idea that giving machines access to data and let them learn for themselves.

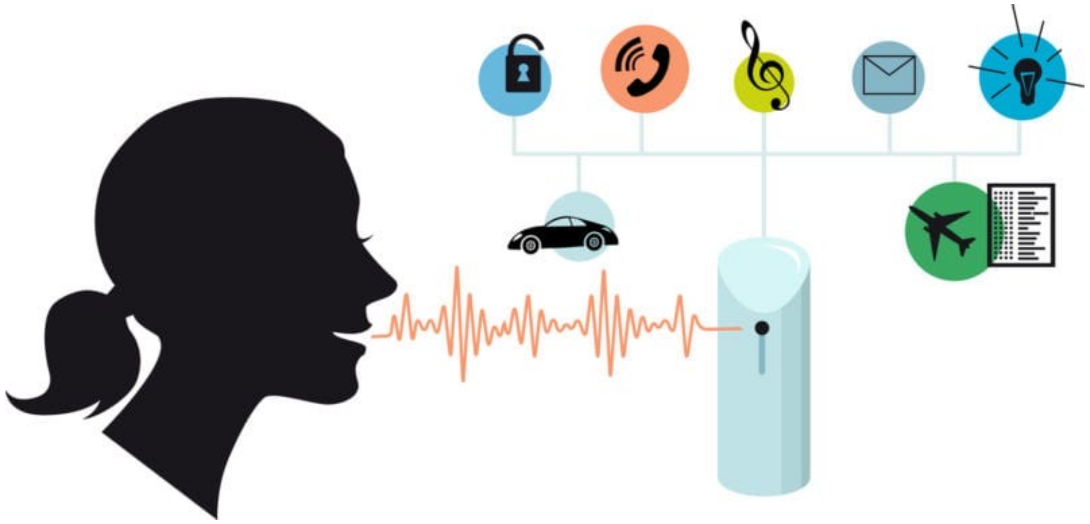


Machine Learning

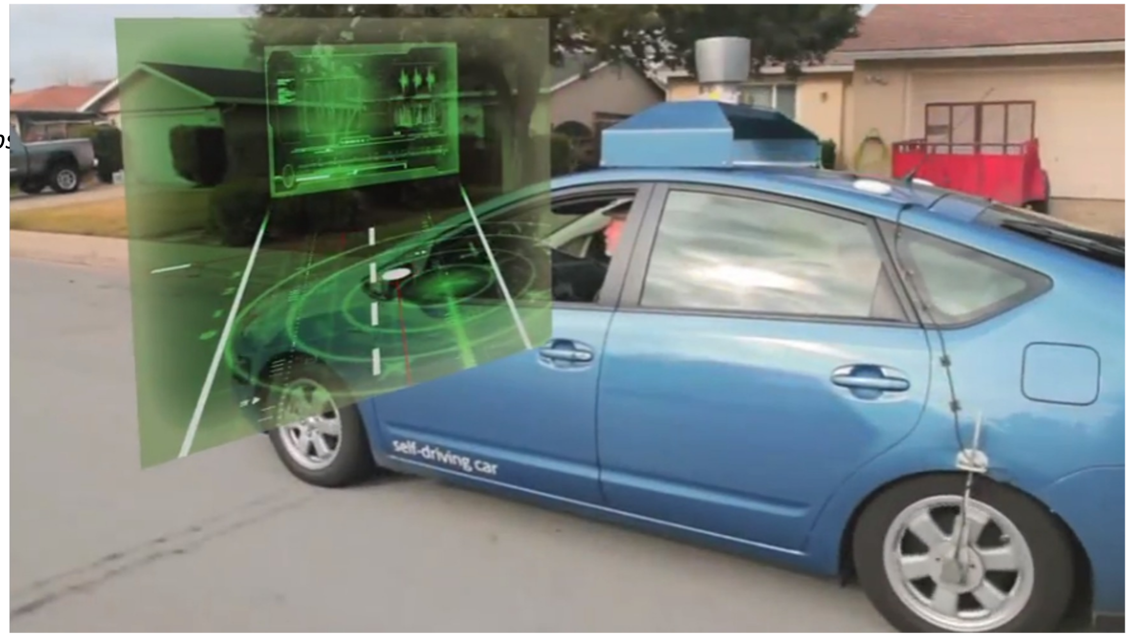
- science of getting computers to act without being explicitly programmed.
- a branch of artificial intelligence focuses on the development of algorithms and statistical models
- based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.
- a method of data analysis that automates analytical model building.

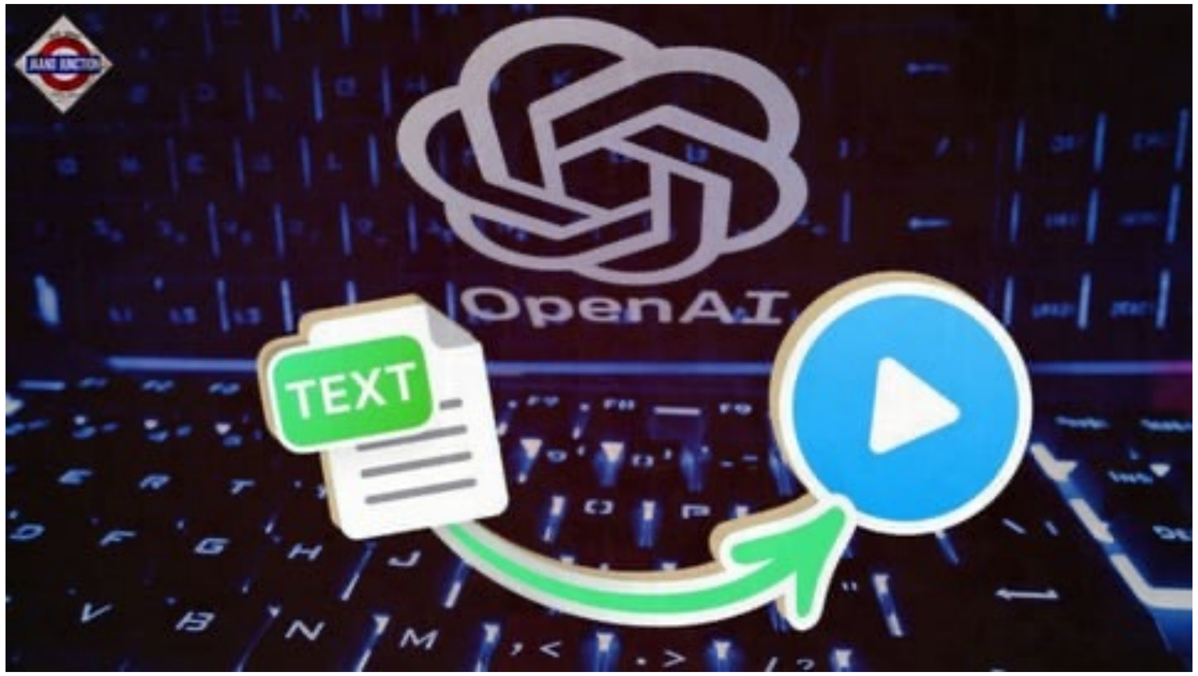
Machine Learning Applications



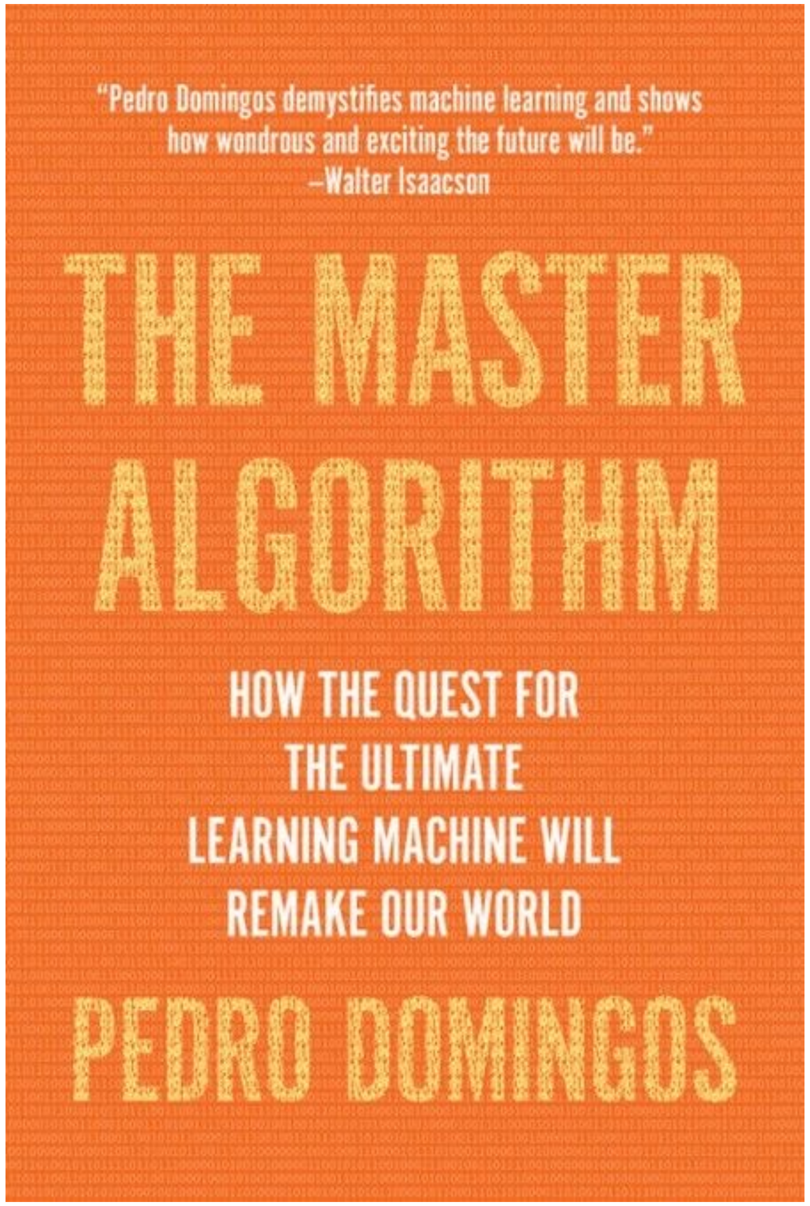


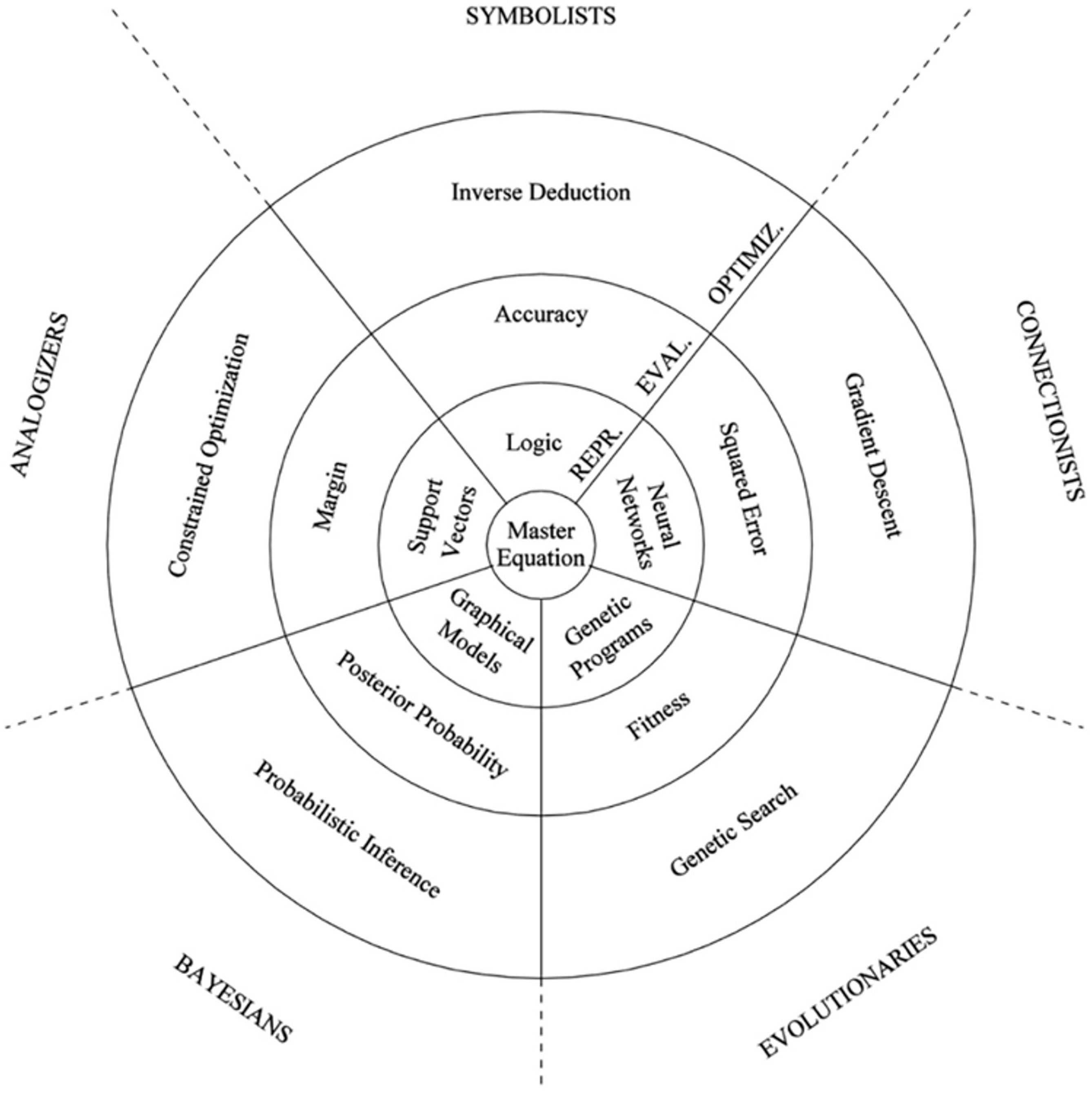
https





Five Tribes of Machine Learning





Domingos, Pedro. "A few useful things to know about machine learning." *Communications of the ACM* 55, no. 10 (2012): 78-87.

The Scope of Machine Learning

ML involves a *wide variety* of each of these:

- data
- task
- model
- algorithm

- Today we'll address a couple of aspects of data.
- Next lecture: the tasks in Machine Learning
- Then in the following weeks: on to the most common models and algorithms

Data

- datahub.io
- openml.org
- Kaggle, UCI

e.g.

  - iris -- https://www.openml.org/d/61
  - penguins -- https://www.openml.org/d/42585
  - diabetes -- https://www.openml.org/d/37
  - banknotes -- https://www.openml.org/d/1462
  - ...

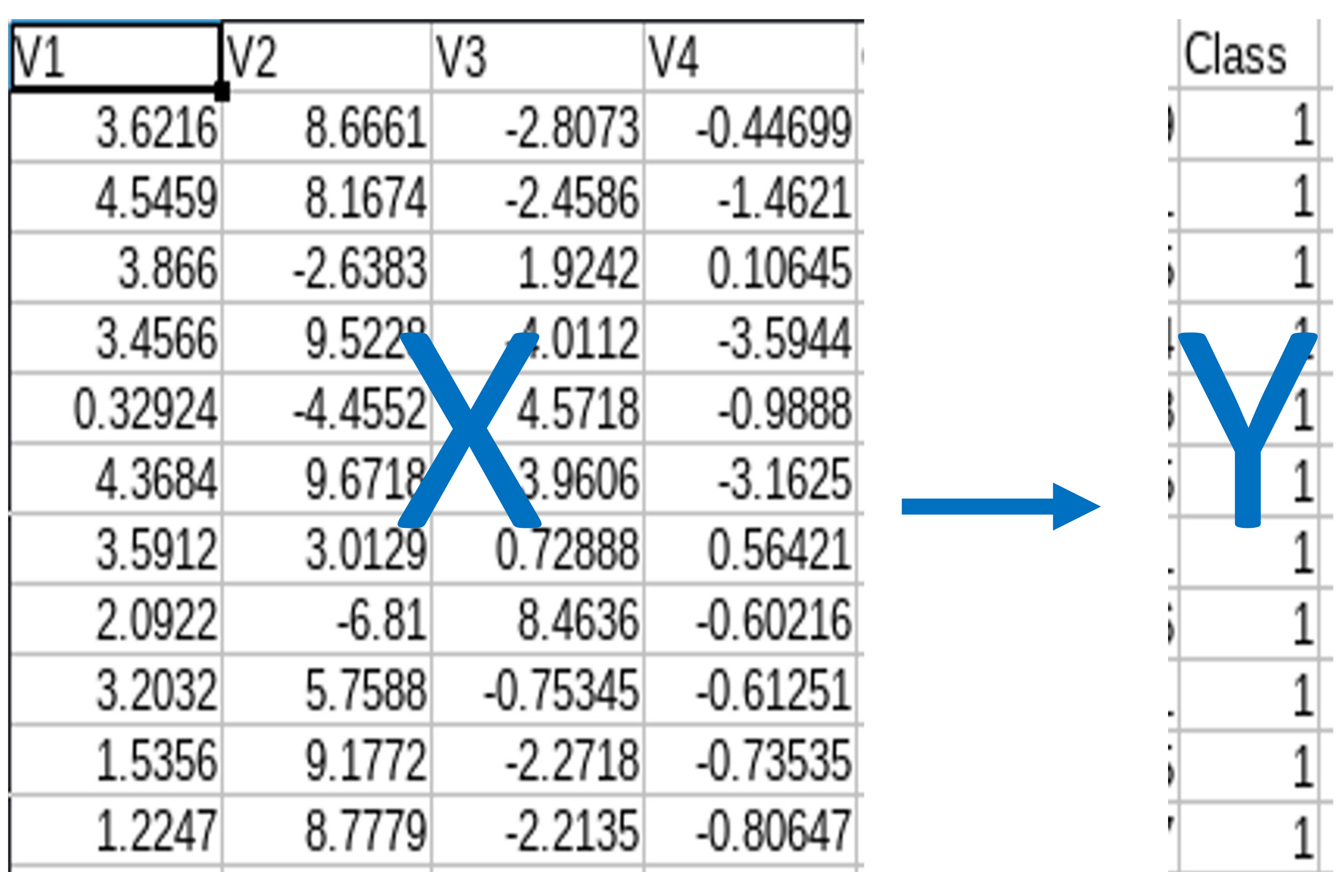
Data

- there are lots of new tools all the time, but near-generic tools at the moment:
  - python
  - numpy
  - sklearn
  - pandas
  - matplotlib
  - jupyter notebooks

Data

- Consider banknotes.csv:
- V1-V4 are values of 4 “features”
- the Class is 1 (legit) or 2 (forged)
- Common to talk separately about X and Y:

V1	V2	V3	V4	Class
3.6216	8.6661	-2.8073	-0.44699	1
4.5459	8.1674	-2.4586	-1.4621	1
3.866	-2.6383	1.9242	0.10645	1
3.4566	9.5228	-4.0112	-3.5944	1
0.32924	-4.4552	4.5718	-0.9888	1
4.3684	9.6718	-3.9606	-3.1625	1
3.5912	3.0129	0.72888	0.56421	1
2.0922	-6.81	8.4636	-0.60216	1
3.2032	5.7588	-0.75345	-0.61251	1
1.5356	9.1772	-2.2718	-0.73535	1
1.2247	8.7779	-2.2135	-0.80647	1



~1300 rows in this case. The “Class=2” ones are further down.

Data as "Vectors" in a "Space"

- each row is one data item, here consisting of a pairing:

V1	V2	V3	V4	Class
3.6216	8.6661	-2.8073	-0.44699	1

$$X \rightarrow y$$

- we might talk about **X** = (x_0, x_1, x_2, x_3) instead of the names v_1, etc specific to this dataset.
- ***X*** is a 4-dimensional *vector*
- x_i is the value for the i^{th} *dimension*

Data as points in a space

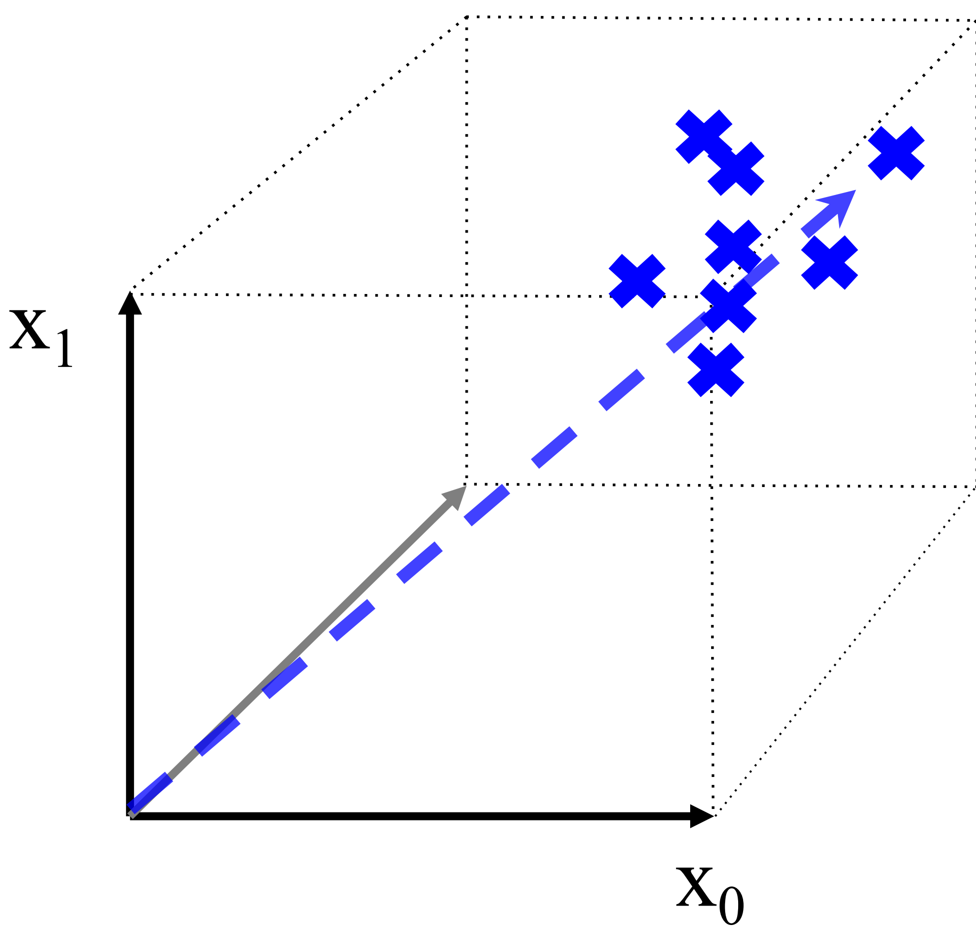
A "row" can be thought of as a "point" in a "space" of data

It's easy to visualise when dimensionality is low:

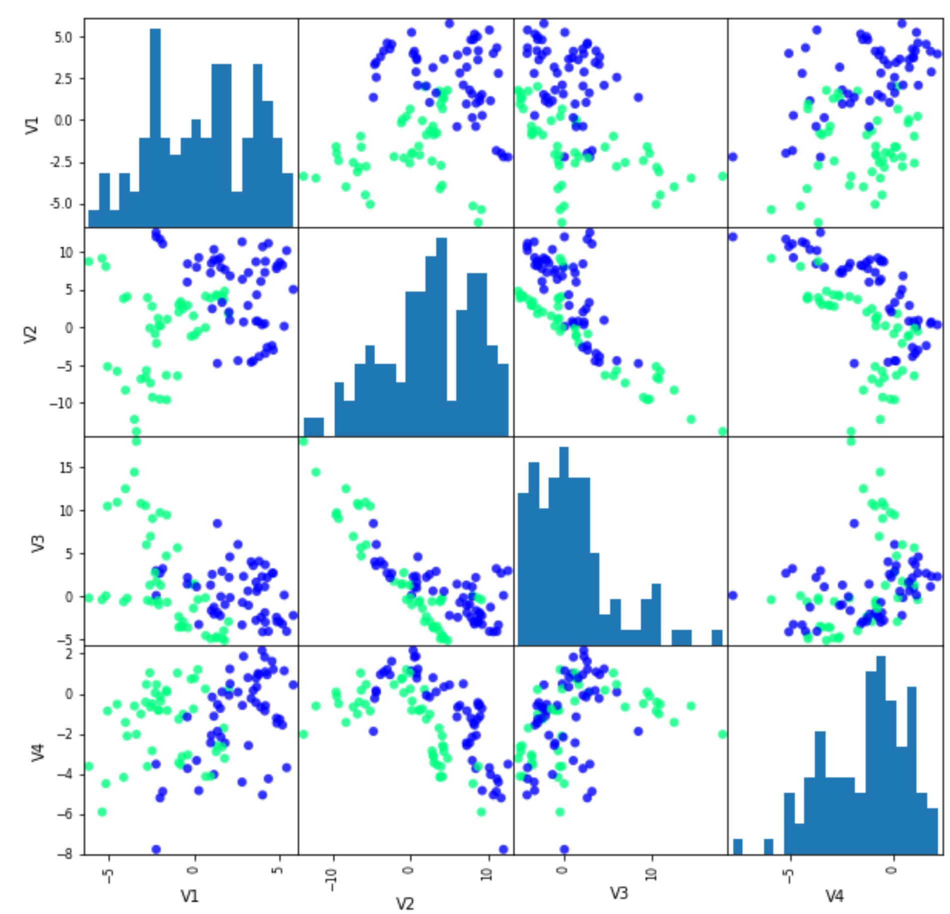
- 1 dimension, e.g. just V1
- 2 dimensions, easy
- 3 dimensions, easy

V1	V2	V3	V4	Class
3.6216	8.6661	-2.8073	-0.44699	1
4.5459	8.1674	-2.4586	-1.4621	1
3.866	-2.6383	1.9242	0.10645	1
3.4566	9.5228	-4.0112	-3.5944	1
0.32924	-4.4552	4.5718	-0.9888	1
4.3684	9.6718	-3.9606	-3.1625	1
3.5912	3.0129	0.72888	0.56421	1
2.0922	-6.81	8.4636	-0.60216	1
3.2032	5.7588	-0.75345	-0.61251	1
1.5356	9.1772	-2.2718	-0.73535	1
1.2247	8.7779	-2.2135	-0.80647	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Data as Points in a Vector Space



V1	V2	V3	V4	Class
3.6216	8.6661	-2.8073	-0.44699	1
4.5459	8.1674	-2.4586	-1.4621	1
3.866	-2.6383	1.9242	0.10645	1
3.4566	9.5228	-4.0112	-3.5944	1
0.32924	-4.4552	4.5718	-0.9888	1
4.3684	9.6718	-3.9606	-3.1625	1
3.5912	3.0129	0.72888	0.56421	1
2.0922	-6.81	8.4636	-0.60216	1
3.2032	5.7588	-0.75345	-0.61251	1
1.5356	9.1772	-2.2718	-0.73535	1
1.2247	8.7779	-2.2135	-0.80647	1



https://sporf.neurodata.io/demos/openml_banknote

Humans can’t see in more than 3d

Some of our intuitions hold, some fail

Data in High dimensions

Eg: I made a million data points that were *d*-dimensional, with each dimension being randomly chosen in range -1 to +1. i.e. in a “box” of side 2.

```
for dims in range(1,21):
    print(dims, countInside(1000000,dims))
```

```
1 1000000
2 785776
3 523488
4 307377
5 164873
6 80816
7 37145
8 15879
9 6370
10 2581
11 915
12 324
13 113
14 41
15 11
16 3
17 1
18 0
19 0
20 0
```

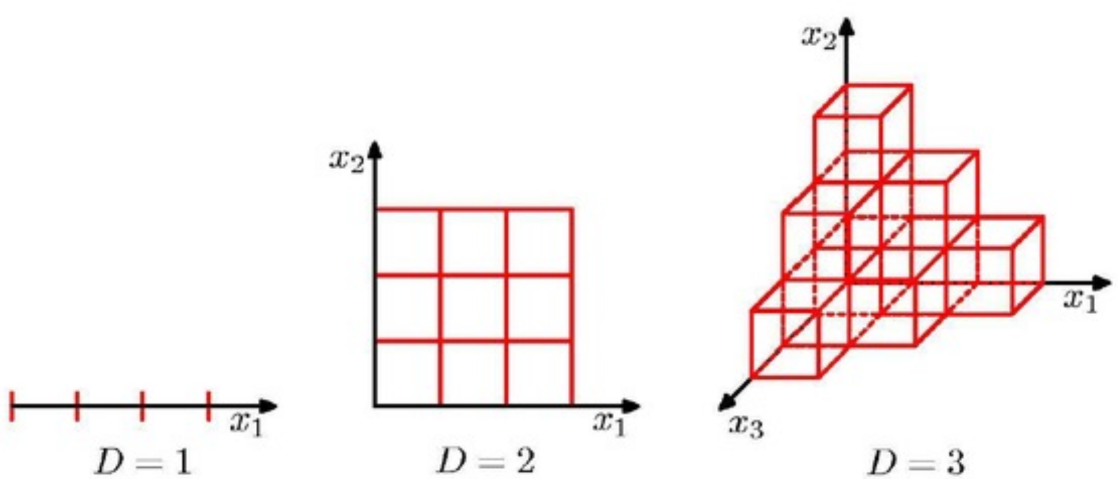
none

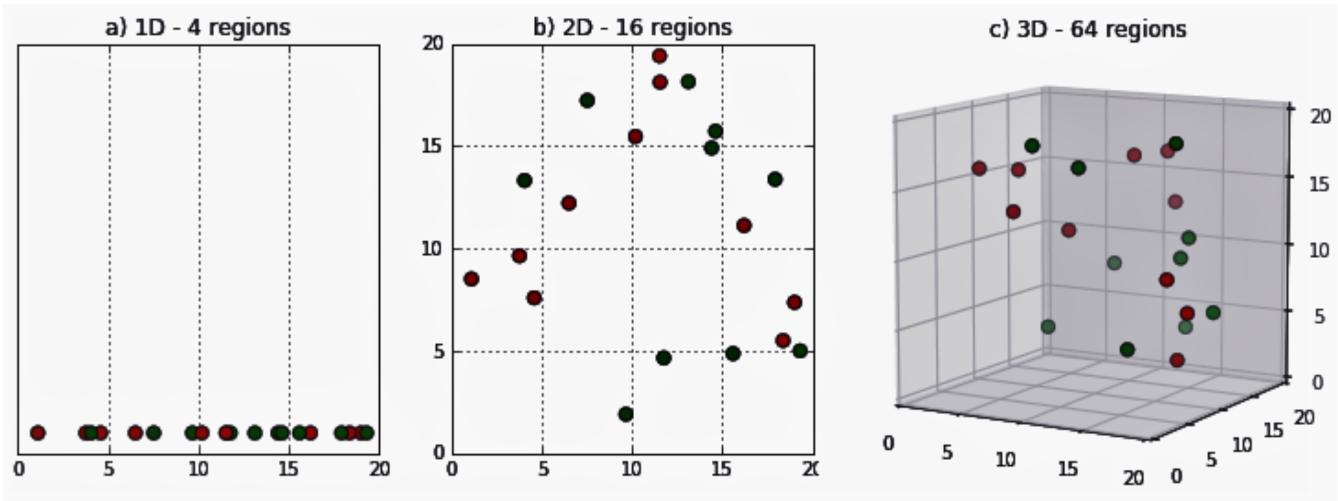
A ball with diameter 2 fits snugly inside this box. Out of a million random points, how many land inside the ball?

The Curse of Dimensionality

Challenges and limitations when dealing with high-dimensional data

- data sparsity, difficult to find meaningful correlations in data
- training the model becomes much slower
- multicollinearity： two or more variables are found to be highly correlated with one another





https://medium.com/@soumikdhar222/curse-of-dimensionality-771dce1d1fbc