Machine learning:
introductory crash course

Andrei Zinovyev

institutCurie

Computational Systems
Biology of Cancer group

PR[AI]RIE
PaRis Artificial Intelligence Research InstitutE

Objectives of the course

- Provide basic vocabulary of machine learning
- Coarse-grained understanding of machine learning concepts
- Some hints on application of machine learning in genomic data analysis

PS: These slides will be available at :
https://auranic.github.io/teaching/2021-ml_intro

Plan of the course

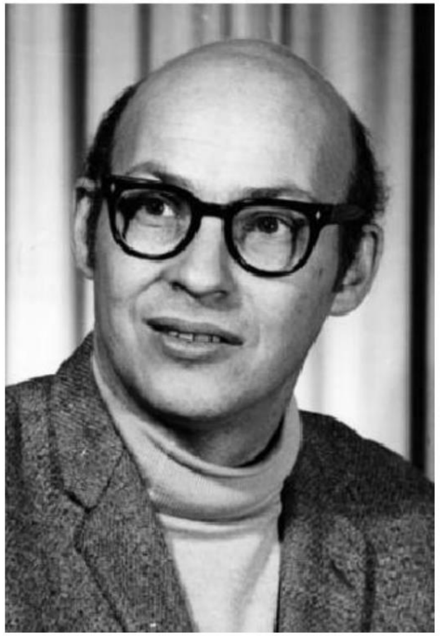
Part I. Introductory notions

Part II. Supervised approach

Part III. Unsupervised approach

What is the difference between statistics, machine learning, artificial intelligence and deep learning?

Artificial intelligence at Dartmouth workshop in 1956 : 2 months, 10 great minds



Marvin Minsky

"An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves.."

John McCarthy

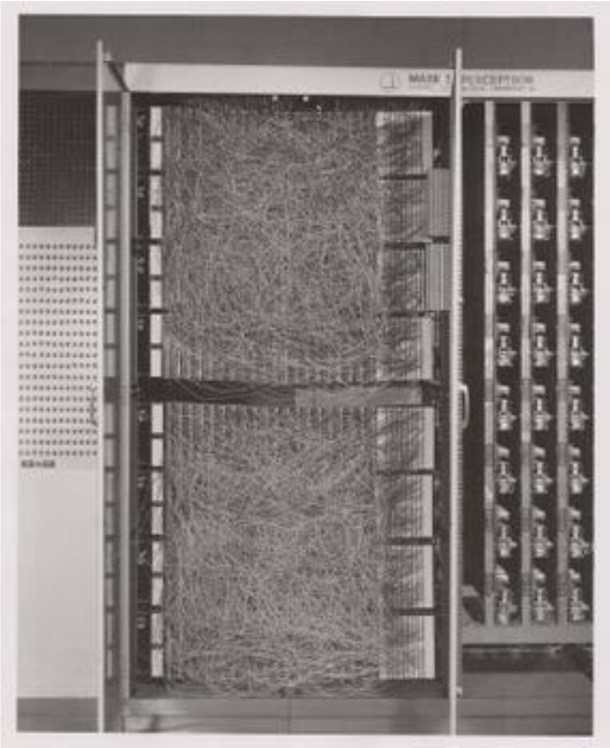
Minsky's pragmatic problems

- *Search*
- *Pattern-Recognition*
- *Learning*
- *Planning*
- *Induction*

Thinking machines:
i) The Knowledge base which has rules and facts.
ii) And the inference engine which applies rules to the already known facts from the knowledge base to infer new facts.

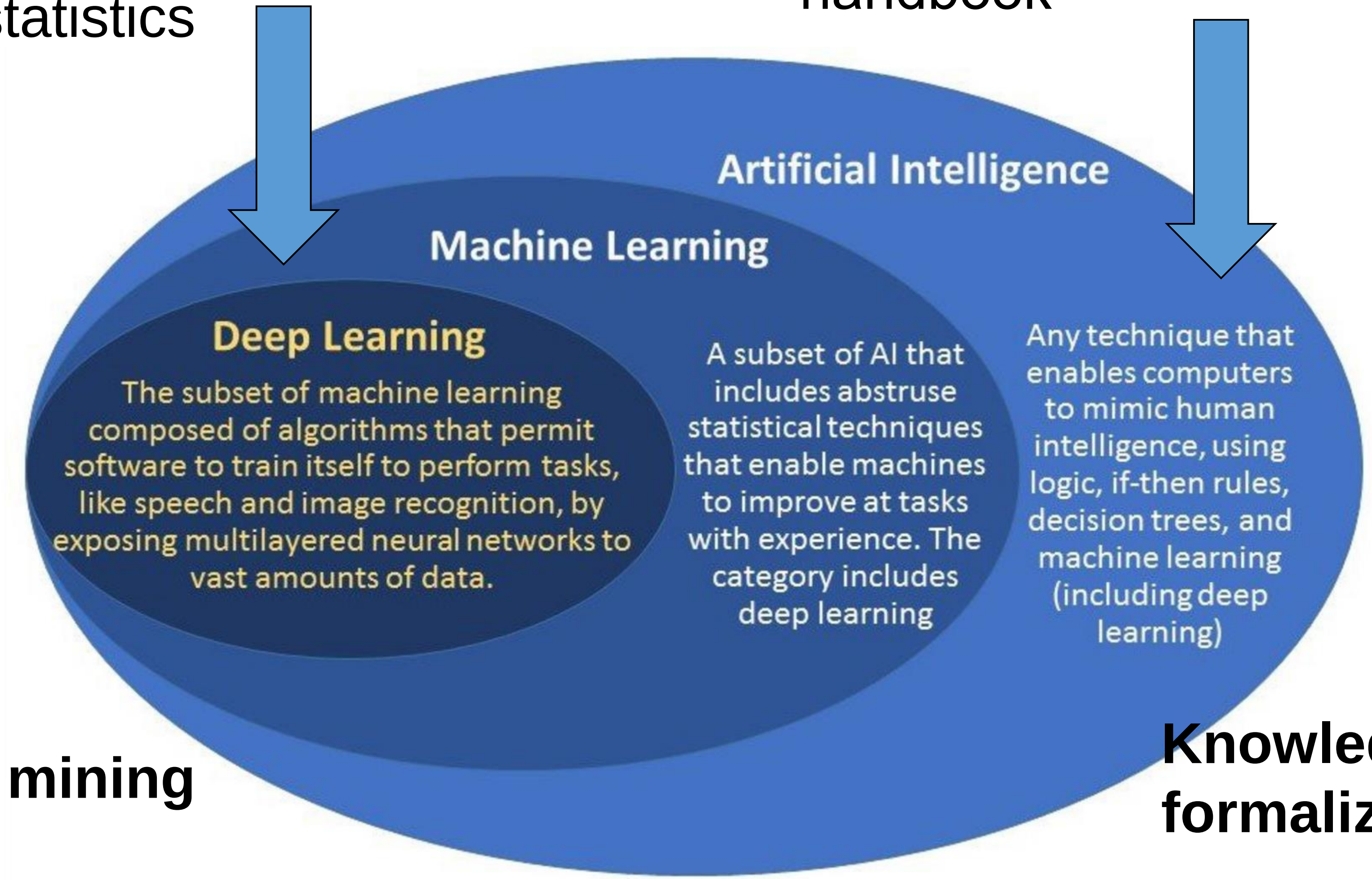
Frank Rosenblatt, inventor of perceptron

- Cornell University, PhD in 1956
- Psychologist, head of cognitive systems section
- Constructor of Mark I Perceptron (simplified perceptron)
- Theory of multi-layered perceptron (aka deep neural network)



A.I. based on data
Very advanced form of statistics

A.I. automating reasoning and knowledge retrieval
Very advanced form of a "handbook"



Data mining

Knowledge formalization

Notion of *machine learning model*

Wikipedia: Machine learning algorithms build **a model** based on training data, in order **to make predictions or decisions** *without being explicitly programmed to do so*.

If the model uses, as a part of training and construction, the notion of **probability distribution** then we talk about statistical inference and **statistical model**

In other cases, model is just a mathematical function characterized by a number of **model parameters** which converts a sample of data into a set of numbers or labels

Machine learning model is a mathematically defined function with (many) parameters

Data

Prediction

$$F_{\beta_1,\beta_2,\ldots,\beta_k}(x_1, x_2, \ldots, x_p) = y$$

Parameters

Fit the model = define its parameters

Machine learning model is a mathematically defined function with (many) parameters

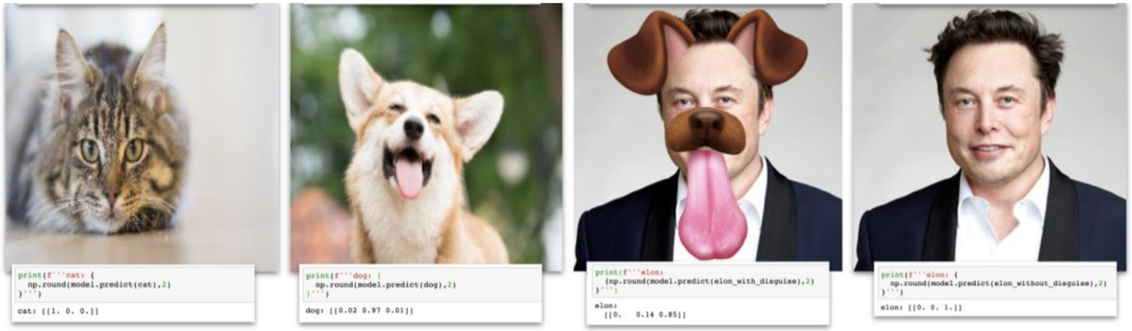
Data must be represented as a set of numbers!

$$F_{\beta_1,\beta_2,\ldots,\beta_k}(x_1, x_2, \ldots, x_p) = y$$

0 – cat

1 - dog

Machine learning model is a mathematically defined function with (many) parameters



$$F_{\beta_1,\beta_2,\ldots,\beta_k}(x_1, x_2, \ldots, x_p) = y$$

100% cat

97% dog

14% dog
85% Elon Musk

100% Elon Musk

Parameters and *hyperparameters*

- **Parameters** are derived via training
- **Hyperparameter** controls the learning process, they are not derived from training the model
- Example of hyperparameter : *the topology and size of a neural network*
- Example of hyperparameter : *the way the data are pre-processed*
- *Type of model* can be also considered a hyperparameter of learning

What is *data* in machine learning?

What is *data* in machine learning?

- Any set of **observations (samples, examples)** that can be described by a common set of **features**
- Features must be represented by numbers
- Most of the existing data are NOT numbers
- Even if the data look like numbers, it almost always require some preparation (**cleaning** and **preprocessing)**!

Data in Machine Learning = Table with numbers

Variables (features)

Objects (samples, measurements)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	ID	GSM26804	GSM26867	GSM26868	GSM26869	GSM26870	GSM26871	GSM26872	GSM26873	GSM26874	GSM26875	GSM26876	GSM26877	GSM26878	GSM26879	GSM26880
2	1007_s_at	10.1865219	8.55465039	10.0171922	9.62855164	8.98179716	9.32096544	9.47013224	8.95127564	9.96641442	10.4723245	9.24634157	9.02814158	9.80726386	10.0884552	9.42789917
3	1053_at	7.14041117	7.9214253	7.19382145	6.33955085	7.0908807	7.14601906	7.11899363	6.1405604	7.07155598	7.54040306	7.13747501	6.68022907	7.3384041	7.06154974	8.10872116
4	117_at	3.82411386	4.04754597	3.79189557	3.84224583	3.92385016	4.86869941	3.88504756	3.76331375	4.32971859	3.89711353	3.81477514	3.86303976	3.75730583	3.90036158	3.7273577
5	121_at	3.61027455	3.54508217	4.54816259	3.74454054	3.61249215	3.92550296	3.6694669	3.52652939	3.64293119	4.04713877	3.46597877	3.49245376	3.67221448	3.66359582	3.61227108
6	1255_g_at	1.88973308	1.83203391	2.04186476	1.89308074	1.91040953	1.91591151	1.95901919	1.83514593	1.91134886	1.98236692	1.89657927	1.91074736	1.9468854	2.00801479	1.87033852
7	1294_at	2.76750098	2.78550183	2.86012235	2.84959436	3.26397282	2.88519676	3.16642211	3.26979855	2.96513014	3.01209778	3.7258176	3.24593083	2.89258523	4.22469552	2.65138576
8	1316_at	3.56186724	6.00938132	5.47627387	3.46082345	3.5589646	3.55022131	3.6495575	3.52929593	3.81489528	3.80151472	3.65353504	3.64297291	5.49390683	3.65494323	3.1776103
9	1320_at	2.73909575	2.68207678	2.97410312	2.73471052	2.78817658	2.79770738	2.90340693	2.67748734	2.78673884	2.94813241	2.74922119	2.78593559	2.88668564	2.98050986	2.62360657
10	1405_i_at	6.56570279	6.28698926	4.91331257	7.08328018	8.85548288	8.73393312	7.00368174	9.20074992	7.56290044	7.08242829	8.62383444	6.68093219	6.64318345	9.43959551	7.59805121
11	1431_at	2.8344133	2.78755371	3.18847354	2.88404293	2.93762587	2.89823055	3.05244607	2.78417436	2.90076657	3.09872342	2.90011368	2.90453628	3.00948297	3.1228764	2.74311179
12	1438_at	2.08209982	2.05046004	2.1380021	2.08249533	2.09277912	2.1099077	2.11854206	2.04375093	2.09150681	2.13821066	2.0847717	2.09495798	2.13115924	2.1353399	2.04584187
13	1487_at	5.54120155	5.35862078	5.46869731	5.52103094	5.51418122	5.55106929	5.4161482	5.44489428	5.24818751	5.56301699	5.42549692	5.54960823	5.82915837	5.56467106	5.50830277
14	1494_f_at	2.54757724	2.37930712	2.62709071	2.38194831	2.44028963	2.4526832	2.4825064	2.4207785	2.60409103	2.49857683	2.43723118	5.2354071	2.48110506	2.49964028	2.41921899
15	1598_g_at	2.7304057	2.67040188	2.59698585	7.93551881	5.34425285	3.13179926	6.57015445	4.4323031	5.18399788	3.88981767	3.85670525	4.88119006	2.70978966	3.85692387	2.75953351
16	160020_at	2.1655937	2.14026455	2.21194547	2.16062823	2.17141169	2.17996571	2.2008294	2.1242019	2.18214481	2.2125988	2.1687426	2.43832316	2.19630922	2.21189546	2.12666118
17	1729_at	7.01826581	6.8620684	6.2748978	5.90084028	6.41997144	6.40378323	6.47535055	6.56605198	6.69687512	6.47743846	6.83935011	6.77296396	7.34317394	6.89120616	6.7314662
18	1773_at	1.65915684	1.63701805	1.72741313	1.65439452	1.67083716	1.67811596	1.70139307	1.64332524	1.67628101	1.71880406	1.6714433	1.67212824	1.70672522	1.71772136	1.6204299
19	177_at	2.94878496	2.86836877	3.14969855	2.97643251	2.98608845	3.03205184	3.08209486	2.89669887	2.97919094	3.13159394	2.92393653	3.02575255	3.12900366	3.1146516	2.95474175
20	179_at	0.57716722	0.55275837	0.63200969	0.57298874	0.58419168	0.59124817	0.61105933	0.56274132	0.59422142	0.62795537	0.58159784	0.58517916	0.61999536	0.61528153	0.5432499
21	1861_at	1.18690202	1.15813312	1.22122377	1.17375236	1.18429212	1.20030196	1.27557097	1.15859558	1.19207924	1.65247824	1.18805205	1.19209823	1.22668581	1.2303746	1.15380531
22	200000_s_at	9.20648723	9.16145477	8.7773438	8.87165851	8.61164901	9.11532903	7.49798068	8.6501605	8.65648402	8.50846148	8.23676007	9.0088335	8.48443715	8.47810052	8.67504714
23	200001_at	10.2111295	9.64241927	8.49184651	9.32048593	9.55080931	9.54725821	9.48348667	9.20829652	9.94634018	9.95504495	9.78220873	9.51833134	10.0545938	9.27885752	9.13860085
24	200002_at	11.7416844	12.5435781	12.5946606	11.2449107	11.7915808	11.4243596	12.3739699	11.5708209	10.6073152	12.4039151	11.1801336	12.3501075	11.8337089	12.0351735	12.0298037
25	200003_s_at	11.9080732	12.7295141	11.8924837	11.8114427	11.9696242	12.0234239	12.1696299	12.4044847	11.5106517	12.6009712	11.214454	13.10743	12.5458678	12.3421479	11.8707809
26	200004_at	12.8626281	13.0318466	12.3226364	12.9112874	12.5629091	13.1340588	13.0250779	12.8029198	12.9787753	13.1286809	12.748781	13.0629905	13.0935061	13.030989	13.4212022
27	200005_at	11.2365327	11.0171526	11.7152353	10.4233686	11.1230332	11.294694	10.7547452	10.900953	10.4631057	10.5860537	10.8269418	10.8355385	11.3292254	10.9910538	11.8222214
28	200006_at	13.4345486	13.07559	13.5937822	13.4856798	13.0994422	13.4686359	13.5762938	13.3161896	13.4856942	13.4639962	13.5249391	13.2203125	13.0822576	13.2736093	13.2935
29	200007_at	13.4323845	13.8222834	13.8399309	13.5619045	12.9873835	13.1472475	13.6921953	13.5192546	13.8453793	14.0467732	13.594668	13.7081125	13.3744476	13.8363235	13.4141853

+ object annotation + variable annotation

GenOMIC data: result of sequencing

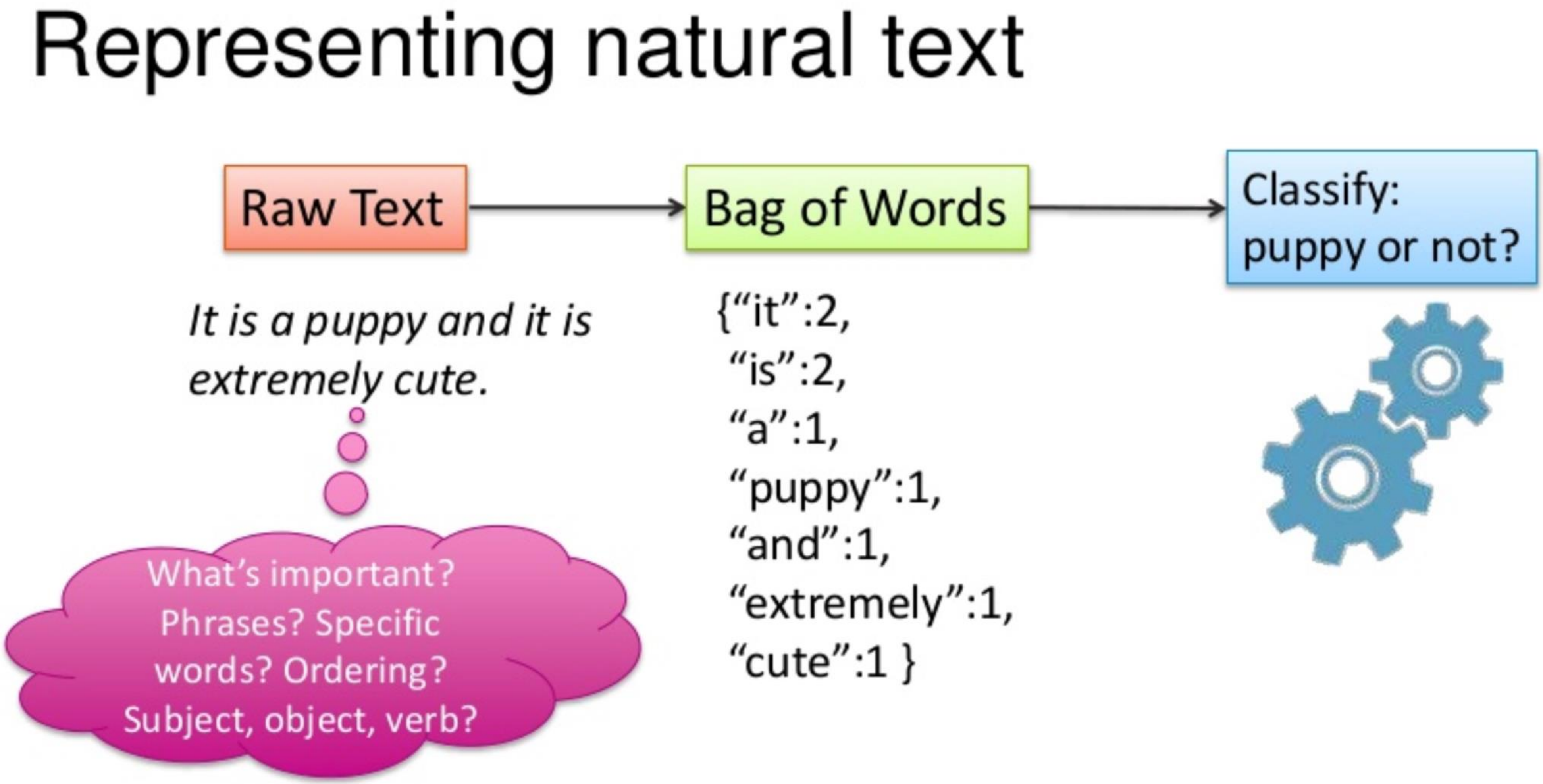
Bag of sequences

Various features types:

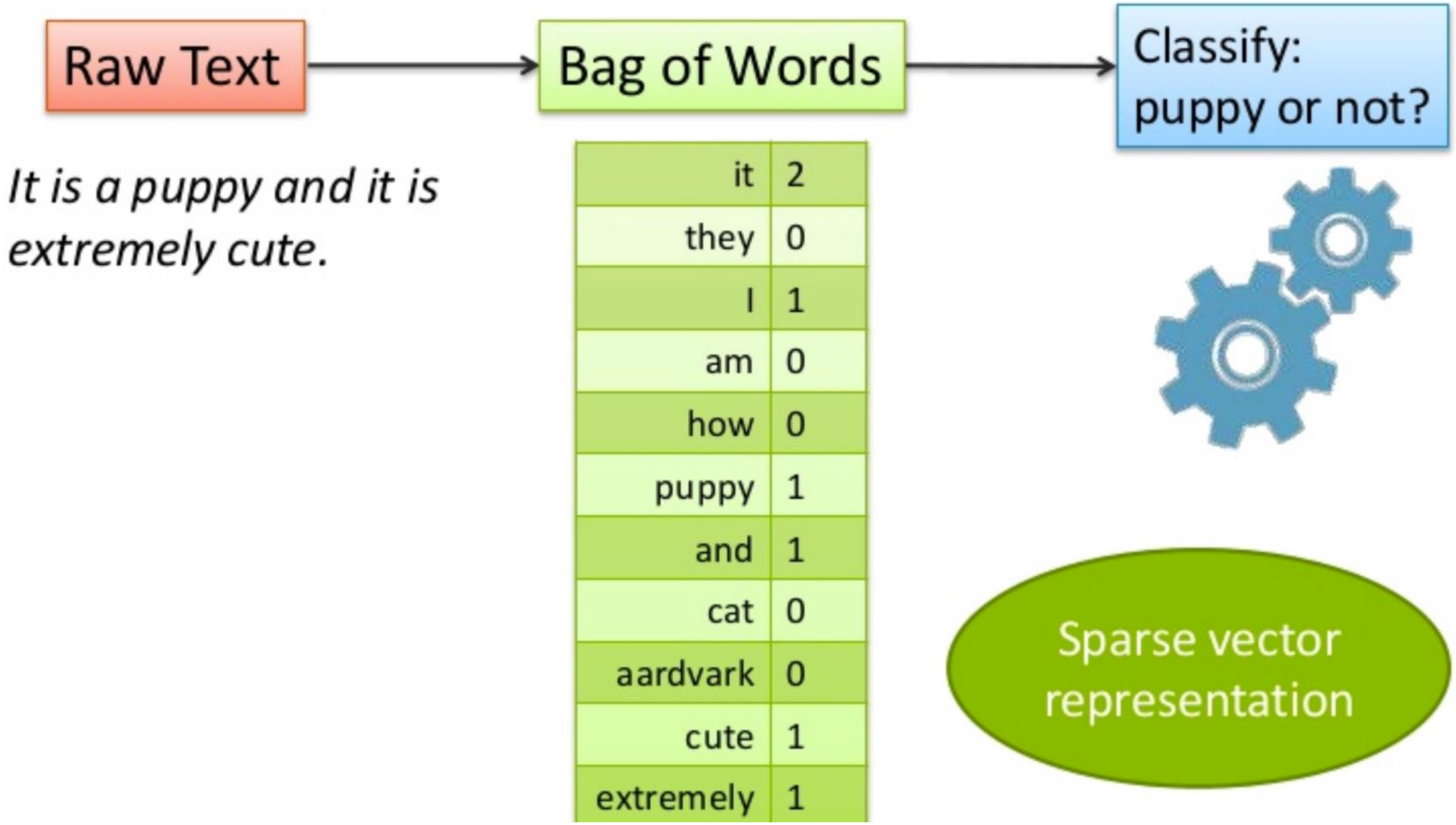
Counts, Peaks, Profiles, kmer frequencies, Hits, Connections between sites

Each technology and problem leads to specific set of features

Other data types: raw data -> numerical table



Representing natural text (e.g., clinical record)



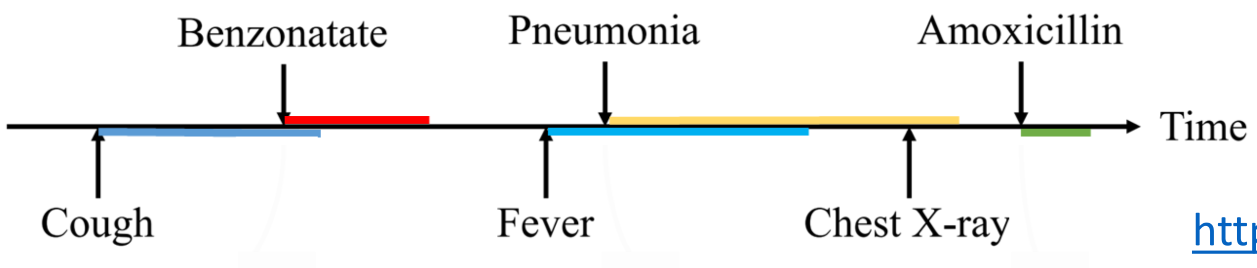
Representing images

Raw Image → Bag of Visual Words → Classify: person or animal?

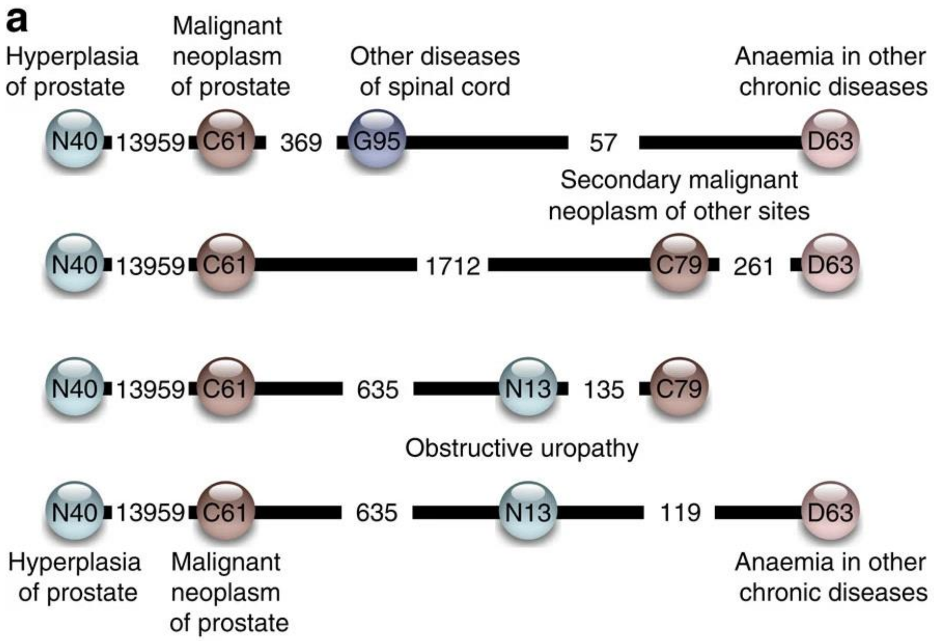
Raw image:
millions of RGB triplets,
one for each pixel

Image source: "Recognizing and learning object categories,"
Li Fei-Fei, Rob Fergus, Anthony Torralba, ICCV 2005—2009.

Encoding disease trajectories from electronic health records

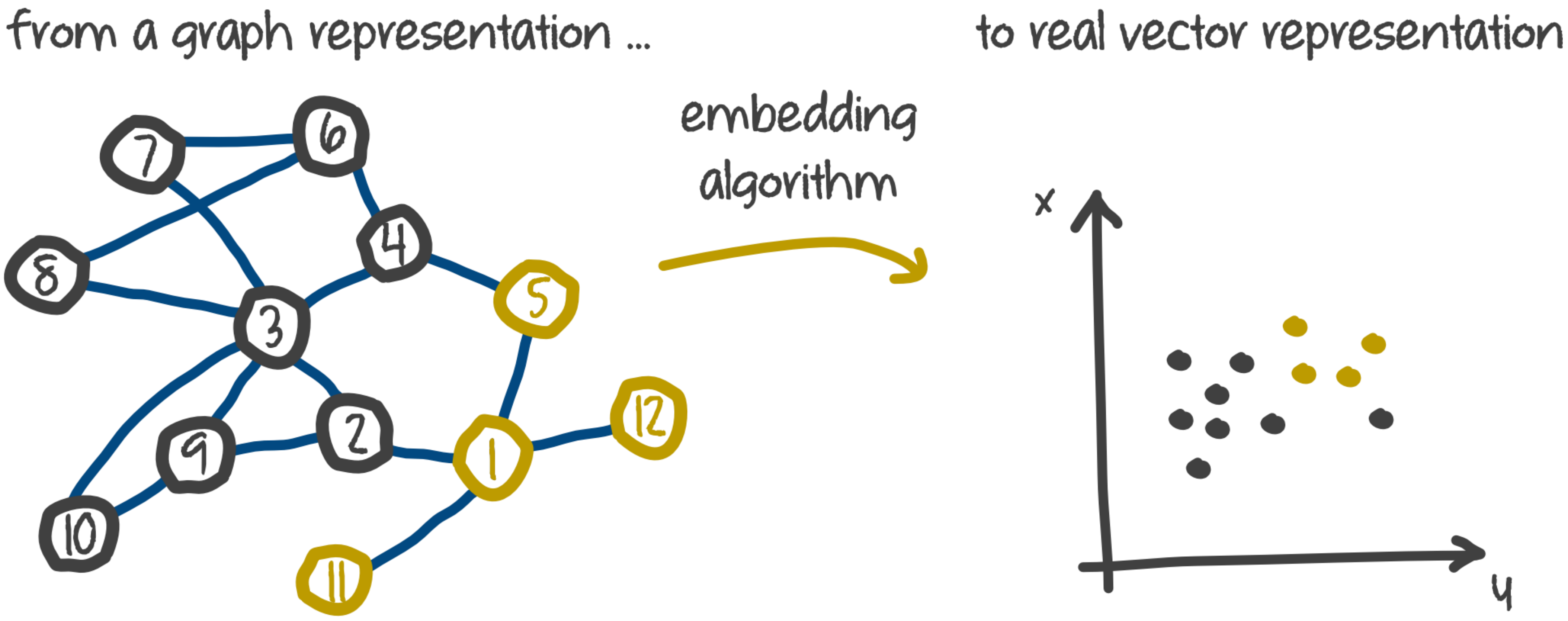


http://proceedings.mlr.press/v116/kumar20a.html



https://www.nature.com/articles/ncomms5022

Graph embedding



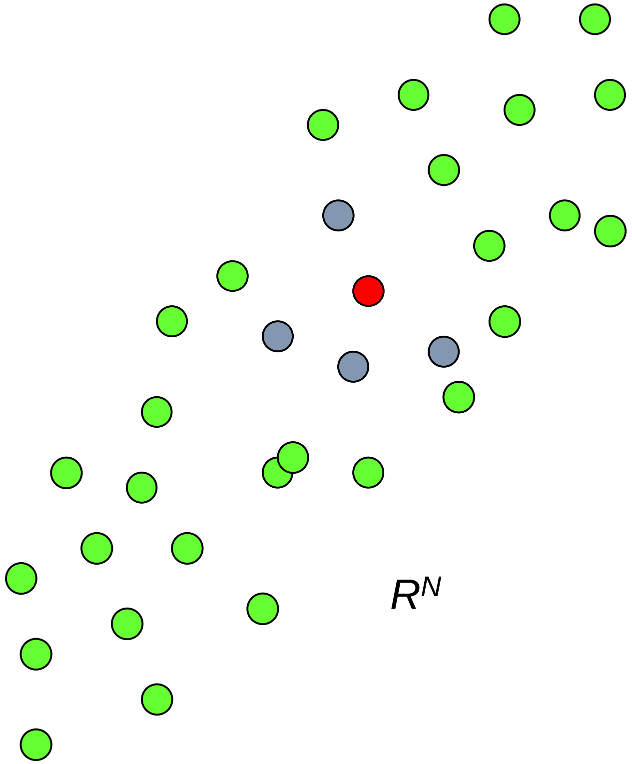
Example: recommendation systems

Geometrical point of view: Analysis of numerical tables = study of a cloud of points in multidimensional space

Variables (features)

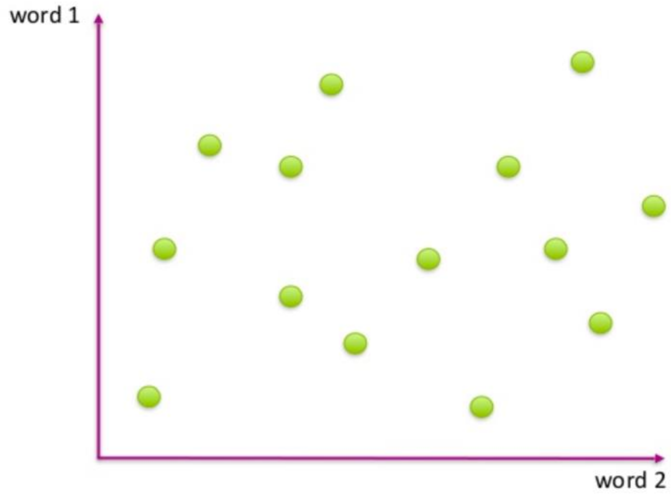
Objects (samples, measurements)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	ID	GSM26804	GSM26867	GSM26868	GSM26869	GSM26870	GSM26871	GSM26872	GSM26873	GSM26874	GSM26875	GSM26876	GSM26877	GSM26878	GSM26879	GSM26880
2	1007_s_at	10.1865219	8.55465039	10.0171922	9.62855164	8.98179716	9.32096544	9.47013224	8.95127564	9.96641442	10.4723245	9.24634157	9.02814158	9.80726386	10.0884552	9.42789917
3	1053_at	7.14041117	7.9214253	7.19382145	6.33955085	7.0908807	7.14601906	7.11899363	6.1405604	7.07155598	7.54040306	7.13747501	6.68022907	7.3384041	7.06154974	8.10872116
4	117_at	3.82411386	4.04754597	3.79189557	3.84224583	3.92385016	4.86869941	3.88504756	3.76331375	4.32971859	3.89711353	3.81477514	3.86303976	3.75730583	3.90036158	3.7273577
5	121_at	3.61027455	3.54508217	4.54816259	3.74454054	3.61249215	3.92550296	3.6694669	3.52652939	3.64293119	4.04713877	3.46597877	3.49245376	3.67221448	3.66359582	3.61227108
6	1255_g_at	1.88973308	1.83203391	2.04186476	1.89308074	1.91040953	1.91591151	1.95901919	1.83514593	1.91134886	1.98236692	1.89657927	1.91074736	1.9468854	2.00801479	1.87033852
7	1294_at	2.76750098	2.78550183	2.86012235	2.84959436	3.26397282	2.88519676	3.16642211	3.26979855	2.96513014	3.01209778	3.7258176	3.24593083	2.89258523	4.22469552	2.65138576
8	1316_at	3.56186724	6.00938132	5.47627387	3.46082345	3.5589646	3.55022131	3.6495575	3.52929593	3.81489528	3.80151472	3.65353504	3.64297291	5.49390683	3.65494323	3.1776103
9	1320_at	2.73909575	2.68207678	2.97410312	2.73471052	2.78817658	2.79770738	2.90340693	2.67748734	2.78673884	2.94813241	2.74922119	2.78593559	2.88668564	2.98050986	2.62360657
10	1405_i_at	6.56570279	6.28698926	4.91331257	7.08328018	8.85548288	8.73393312	7.00368174	9.20074992	7.56290044	7.08242829	8.62383444	6.68093219	6.64318345	9.43959551	7.59805121
11	1431_at	2.8344133	2.78755371	3.18847354	2.88404293	2.93762587	2.89823055	3.05244607	2.78417436	2.90076657	3.09872342	2.90011368	2.90453628	3.00948297	3.1228764	2.74311179
12	1438_at	2.08209982	2.05046004	2.1380021	2.08249533	2.09277912	2.1099077	2.11854206	2.04375093	2.09150681	2.13821066	2.0847717	2.09495798	2.13115924	2.1353399	2.04584187
13	1487_at	5.54120155	5.35862078	5.46869731	5.52103094	5.51418122	5.55106929	5.4161482	5.44489428	5.24818751	5.56301699	5.42549692	5.54960823	5.82915837	5.56467106	5.50830277
14	1494_f_at	2.54757724	2.37930712	2.62709071	2.38194831	2.44028963	2.4526832	2.4825064	2.4207785	2.60409103	2.49857683	2.43723118	5.2354071	2.48110506	2.49964028	2.41921899
15	1598_g_at	2.7304057	2.67040188	2.59698585	7.93551881	5.34425285	3.13179926	6.57015445	4.4323031	5.18399788	3.88981767	3.85670525	4.88119006	2.70978966	3.85692387	2.75953351
16	160020_at	2.1655937	2.14026455	2.21194547	2.16062823	2.17141169	2.17996571	2.2008294	2.1242019	2.18214481	2.2125988	2.1687426	2.43832316	2.19630922	2.21189546	2.12666118
17	1729_at	7.01826581	6.8620684	6.2748978	5.90084028	6.41997144	6.40378323	6.47535055	6.56605198	6.69687512	6.47743846	6.83935011	6.77296396	7.34317394	6.89120616	6.7314662
18	1773_at	1.65915684	1.63701805	1.72741313	1.65439452	1.67083716	1.67811596	1.70139307	1.64332524	1.67628101	1.71880406	1.6714433	1.67212824	1.70672522	1.71772136	1.6204299
19	177_at	2.94878496	2.86836877	3.14969855	2.97643251	2.98608845	3.03205184	3.08209486	2.89669887	2.97919094	3.13159394	2.92393653	3.02575255	3.12900366	3.1146516	2.95474175
20	179_at	0.57716722	0.55275837	0.63200969	0.57298874	0.58419168	0.59124817	0.61105933	0.56274132	0.59422142	0.62795537	0.58159784	0.58517916	0.61999536	0.61528153	0.5432499
21	1861_at	1.18690202	1.15813312	1.22122377	1.17375236	1.18429212	1.20030196	1.27557097	1.15859558	1.19207924	1.65247824	1.18805205	1.19209823	1.22668581	1.2303746	1.15380531
22	200000_s_at	9.20648723	9.16145477	8.7773438	8.87165851	8.61164901	9.11532903	7.49798068	8.6501605	8.65648402	8.50846148	8.23676007	9.0088335	8.48443715	8.47810052	8.67504714
23	200001_at	10.2111295	9.64241927	8.49184651	9.32048593	9.55080931	9.54725821	9.48348667	9.20829652	9.94634018	9.95504495	9.78220873	9.51833134	10.0545938	9.27885752	9.13860085
24	200002_at	11.7416844	12.5435781	12.5946606	11.2449107	11.7915808	11.4243596	12.3739699	11.5708209	10.6073152	12.4039151	11.1801336	12.3501075	11.8337089	12.0351735	12.0298037
25	200003_s_at	11.9080732	12.7295141	11.8924837	11.8114427	11.9696242	12.0234239	12.1696299	12.4044847	11.5106517	12.6009712	11.214454	13.10743	12.5458678	12.3421479	11.8707809
26	200004_at	12.8626281	13.0318466	12.3226364	12.9112874	12.5629091	13.1340588	13.0250779	12.8029198	12.9787753	13.1286809	12.748781	13.0629905	13.0935061	13.030989	13.4212022
27	200005_at	11.2365327	11.0171526	11.7152353	10.4233686	11.1230332	11.294694	10.7547452	10.900953	10.4631057	10.5860537	10.8269418	10.8355385	11.3292254	10.9910538	11.8222214
28	200006_at	13.4345486	13.07559	13.5937822	13.4856798	13.0994422	13.4686359	13.5762938	13.3161896	13.4856942	13.4639962	13.5249391	13.2203125	13.0822576	13.2736093	13.2935
29	200007_at	13.4323845	13.8222834	13.8399309	13.5619045	12.9873835	13.1472475	13.6921953	13.5192546	13.8453793	14.0467732	13.594668	13.7081125	13.3744476	13.8363235	13.4141853

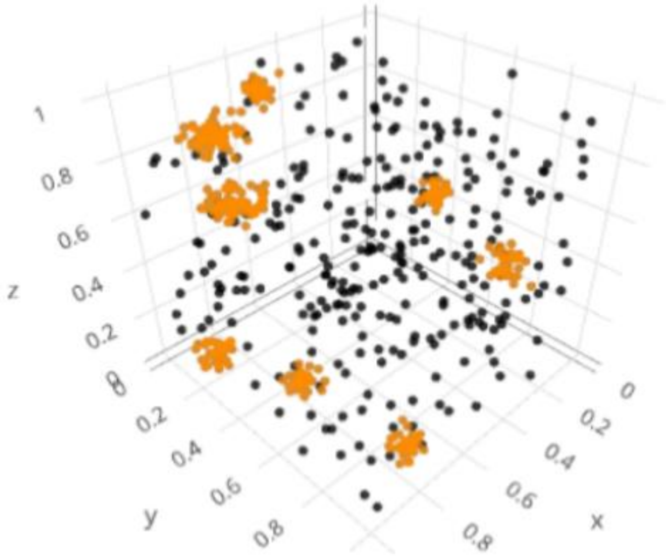


Data point cloud in R^N

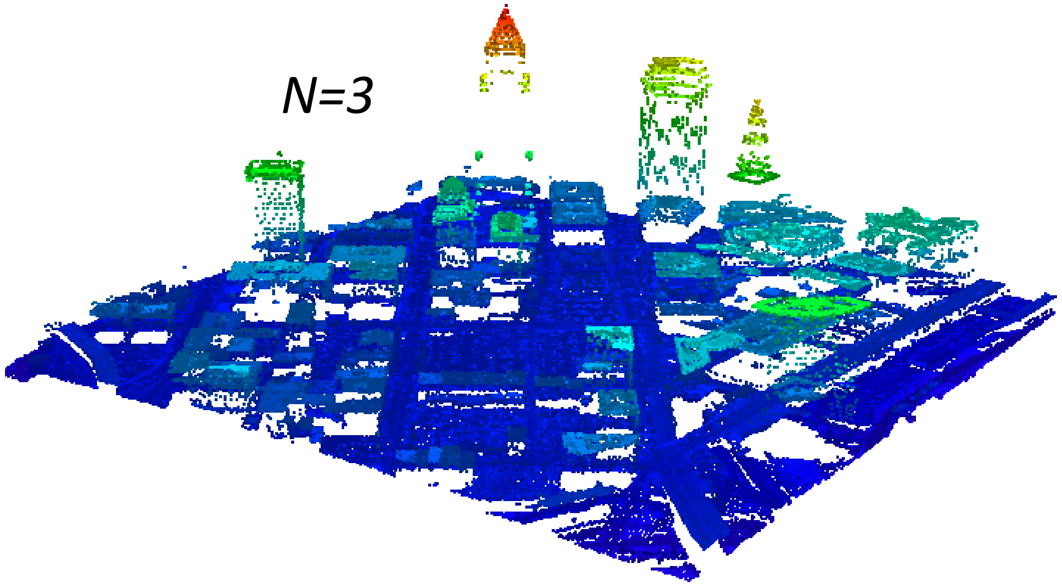
N=2



N=3

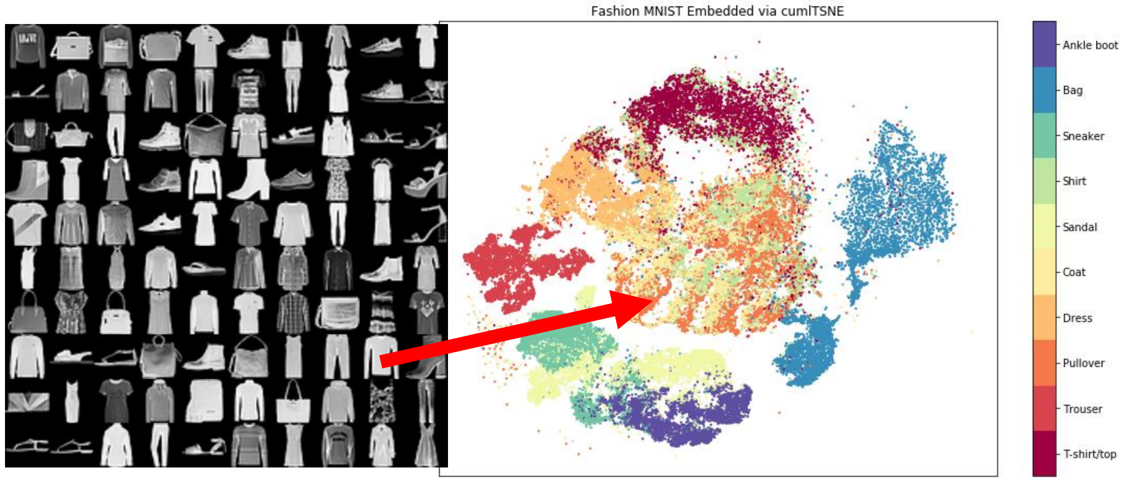


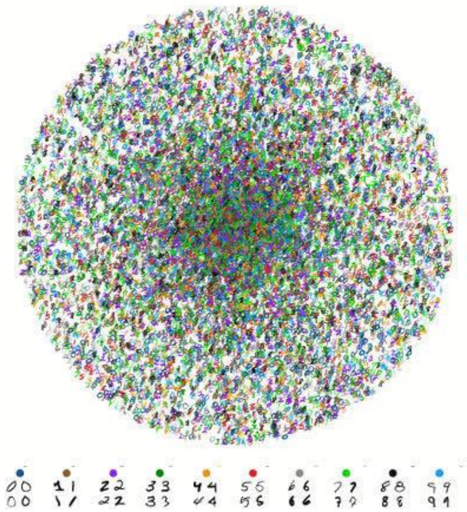
N=3



LIDAR Data point cloud

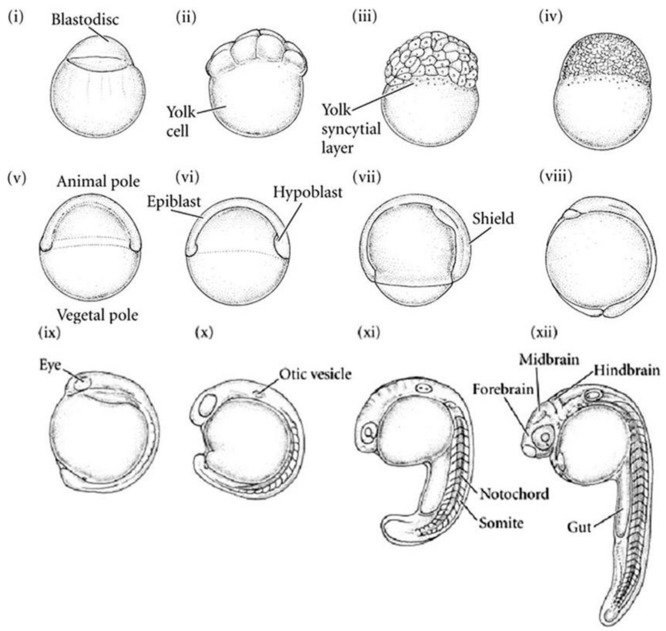
N=784





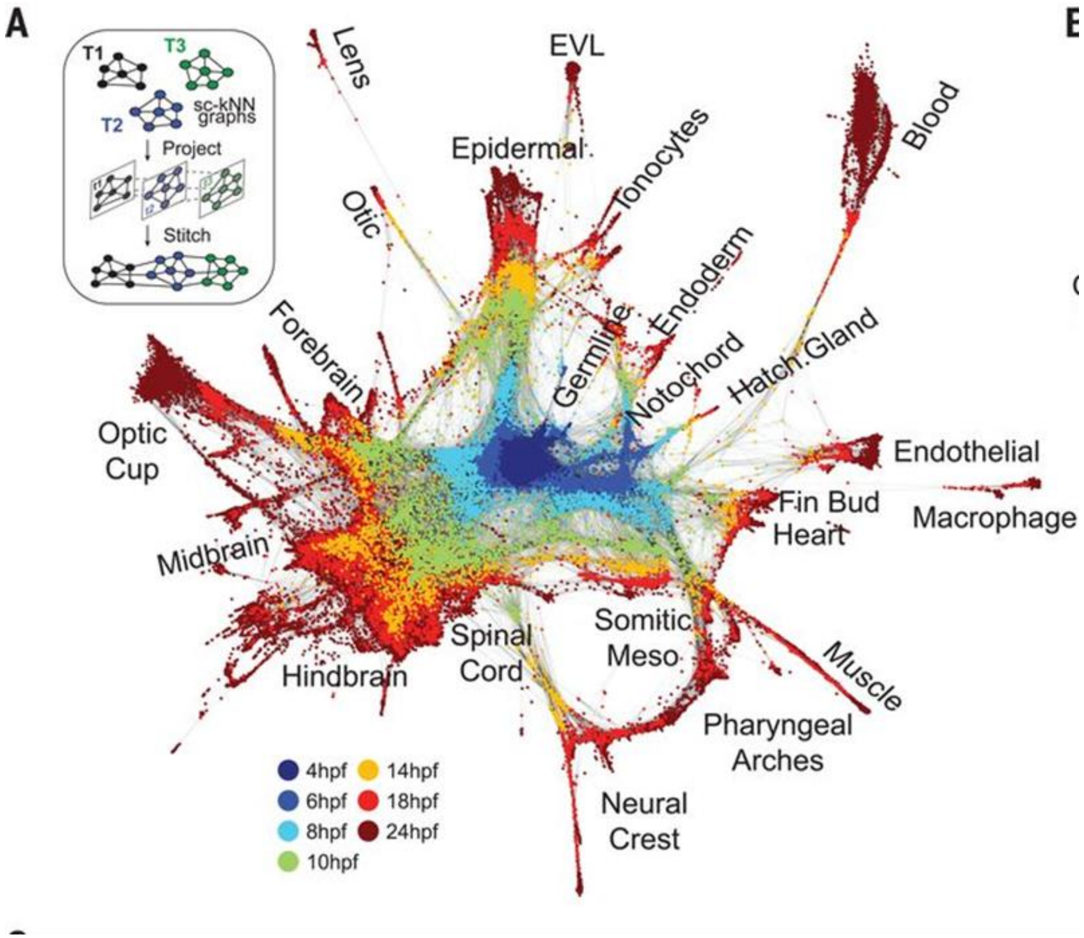
Single cell omics: biology becomes a field of data science

Biological (physical) view



Single cell omics view

N ~ 20000



From Wagner et al, Science, 2018

Data types: most of the world data are not numbers!

1) Numerical
    - Example: *weight, height*
2) Categorical:
    - Ordinal
        - Example: age range (infant, toddler, teenager, young, adult, senior)
    - Nominal
        - Example: *eye color, mothertongue*

Simplest data type: BINARY (Yes/No, False/True, 0/1)

All *raw data* – even that which looks like numbers – must be prepared for machine learning algorithms!

GENES

SAMPLES

	ISG15	AURKAIP1	MRPL20	NOC2L	UBE2J2	CPSF3L	AGRN	PUSL1	CCNL2	AP006222.2	SAMD11	DVL1	TNFRSF18	ANKRD65	C1orf159	SDF4	PLEKHN1
c4	1197	502	270	38	0	0	38	0	0	0	38	0	0	0	0	0	0
c5	1837	481	437	131	43	43	87	0	0	0	43	0	0	0	0	0	0
c7	1607	459	114	0	114	114	0	0	0	0	0	0	0	0	0	0	0
c13	2852	547	429	0	39	117	0	0	0	78	0	0	0	0	0	0	0
c16	4574	424	196	98	32	196	0	0	0	0	32	0	0	0	0	0	0
c17	2619	545	72	36	0	36	0	36	0	0	0	0	0	0	0	36	0
c22	6100	169	169	0	0	0	169	0	0	0	0	0	0	0	0	0	0
c25	1829	228	114	0	57	57	57	0	57	0	0	0	0	0	0	0	0
c36	1973	529	721	48	48	0	48	48	0	48	0	48	0	0	0	0	0
c37	3646	487	205	102	25	51	0	51	0	0	0	0	51	77	0	0	0
c43	7961	433	166	66	66	0	99	0	0	0	0	0	33	0	0	33	0
c57	4222	408	363	0	0	0	0	136	45	0	0	0	0	0	0	0	45
c61	6313	564	333	102	25	25	25	25	0	0	0	0	0	51	0	0	0
c65	1636	459	459	86	86	0	28	57	28	57	28	86	0	0	0	28	0
c66	2807	561	280	62	31	0	62	31	0	0	31	31	93	0	31	0	0
c68	904	620	212	35	35	53	35	35	35	0	35	0	17	53	0	0	17
c73	3490	427	213	71	35	0	106	0	0	0	71	0	0	0	0	0	71
c79	453	517	647	64	129	0	129	0	0	0	0	64	0	0	64	0	0
c86	2948	368	0	0	61	61	61	0	0	0	0	0	0	0	0	0	0
c88	4105	696	274	18	54	36	73	18	18	18	18	0	0	0	18	0	0
c90	6037	862	246	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c93	2966	679	247	30	30	0	0	61	0	0	0	30	0	0	30	0	0
c96	1814	753	393	136	51	102	0	34	34	34	34	17	0	0	0	51	0
c98	3541	108	272	108	54	54	0	54	54	0	0	0	0	0	0	0	0
c104	2593	420	175	140	0	0	70	0	35	0	0	0	0	35	0	0	0
c111	5239	698	174	0	58	0	0	58	58	0	0	0	0	0	0	0	0
c112	3293	542	271	0	0	0	77	0	38	38	0	0	0	38	0	0	0
c136	3756	352	117	117	0	0	0	0	0	0	0	0	58	0	58	0	0
c137	2909	513	285	57	0	0	0	0	0	0	0	0	0	0	0	0	0
c138	4524	421	268	0	76	0	0	76	38	0	0	0	0	0	0	0	0
c140	961	601	120	0	0	0	240	0	0	0	0	0	0	0	0	0	0
c150	2639	434	310	108	124	46	31	46	31	15	15	15	0	0	0	0	15
c151	1477	396	144	144	72	108	0	0	0	0	0	72	0	36	0	36	0

Example: RNASeq count table

All *raw data* – even that which looks like numbers – must be prepared for machine learning algorithms!

Illusion 1: that these numbers are directly usable

Illusion 2: that this is the “natural” choice of numerical features (data representation)

GENES

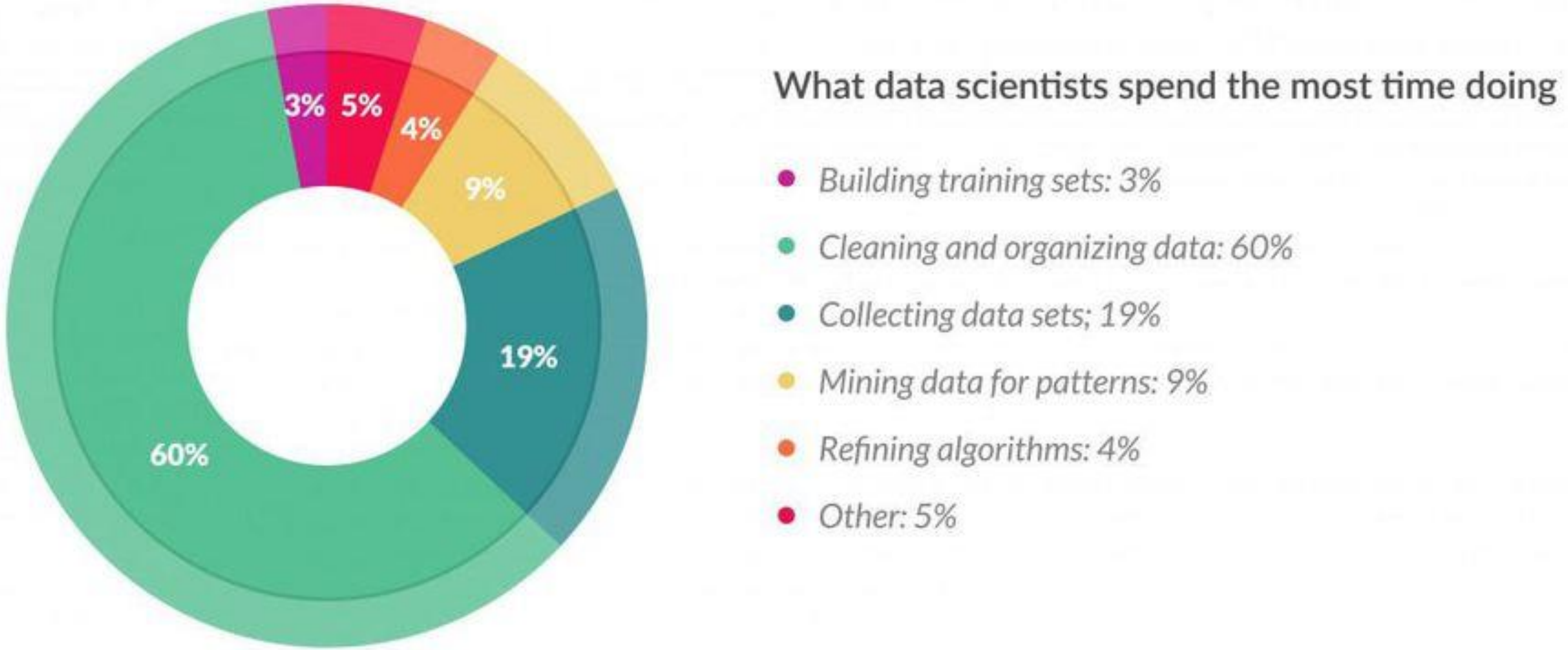
SAMPLES

	ISG15	AURKAIP1	MRPL20	NOC2L	UBE2J2	CPSF3L	AGRN	PUSL1	CCNL2	AP006222.2	SAMD11	DVL1	TNFRSF18	ANKRD65	C1orf159	SDF4	PLEKHN1
c4	1197	502	270	38	0	0	38	0	0	0	38	0	0	0	0	0	0
c5	1837	481	437	131	43	43	87	0	0	0	43	0	0	0	0	0	0
c7	1607	459	114	0	114	114	0	0	0	0	0	0	0	0	0	0	0
c13	[illegible]	547	429	0	39	117	0	0	0	78	0	0	0	0	0	0	0
c16	4574	[illegible]	[illegible]	[illegible]	32	96	0	[illegible]	[illegible]	0	[illegible]	0	[illegible]	0	0	0	0
c17	2619	[illegible]	72	[illegible]	0	36	0	[illegible]	0	0	[illegible]	0	[illegible]	0	0	36	0
c22	6100	169	169	0	0	0	169	0	0	0	0	0	0	0	0	0	0
c25	1829	228	114	0	57	57	57	0	57	0	0	0	0	0	0	0	0
c36	[illegible]	529	721	48	48	0	48	48	0	48	0	48	0	0	0	0	0
c37	3646	[illegible]	205	102	[illegible]	51	0	51	0	0	0	0	51	77	0	0	0
c43	7961	433	166	66	66	0	99	0	0	0	0	0	33	0	0	33	0
c57	4222	408	363	0	0	0	0	136	45	0	0	0	0	0	0	0	45
c61	6313	564	333	102	25	25	25	25	0	0	0	0	0	51	0	0	0
c65	1636	459	459	86	86	0	28	57	28	57	28	86	0	0	0	28	0
c66	2807	561	280	62	31	0	62	31	0	0	31	31	93	0	31	0	0
c68	904	620	212	35	35	53	35	35	35	0	35	0	17	53	0	0	17
c73	3490	427	213	71	35	0	106	0	0	0	71	0	0	0	0	0	71
c79	453	517	647	64	129	0	129	0	0	0	0	64	0	0	64	0	0
c86	2948	[illegible]	[illegible]	[illegible]	61	61	61	0	0	[illegible]	0	0	[illegible]	0	0	0	0
c88	4105	696	[illegible]	18	54	36	[illegible]	18	18	[illegible]	18	0	[illegible]	0	18	0	0
c90	6037	862	246	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c93	2966	679	247	30	30	0	0	61	0	0	0	30	0	0	30	0	0
c96	1814	753	393	[illegible]	51	102	0	34	34	34	34	17	0	0	0	51	0
c98	3541	[illegible]	[illegible]	[illegible]	54	54	0	54	54	[illegible]	0	0	0	0	0	0	0
c104	2593	420	175	[illegible]	0	0	70	[illegible]	35	[illegible]	0	0	[illegible]	35	0	0	0
c111	5239	698	174	0	58	0	0	58	58	0	0	0	0	0	0	0	0
c112	3293	542	271	0	0	0	77	0	38	38	0	0	0	38	0	0	0
c136	3756	352	117	117	[illegible]	0	0	0	0	0	0	0	58	0	58	0	0
c137	2909	[illegible]	[illegible]	57	[illegible]	[illegible]	0	0	0	0	0	0	0	0	0	0	0
c138	4524	421	[illegible]	[illegible]	76	0	0	76	38	0	0	0	0	0	0	0	0
c140	961	601	120	0	0	0	240	0	0	0	0	0	0	0	0	0	0
c150	2639	434	310	108	124	46	31	46	31	15	15	15	0	0	0	0	15
c151	1477	396	144	144	72	108	0	0	0	0	0	72	0	36	0	36	0

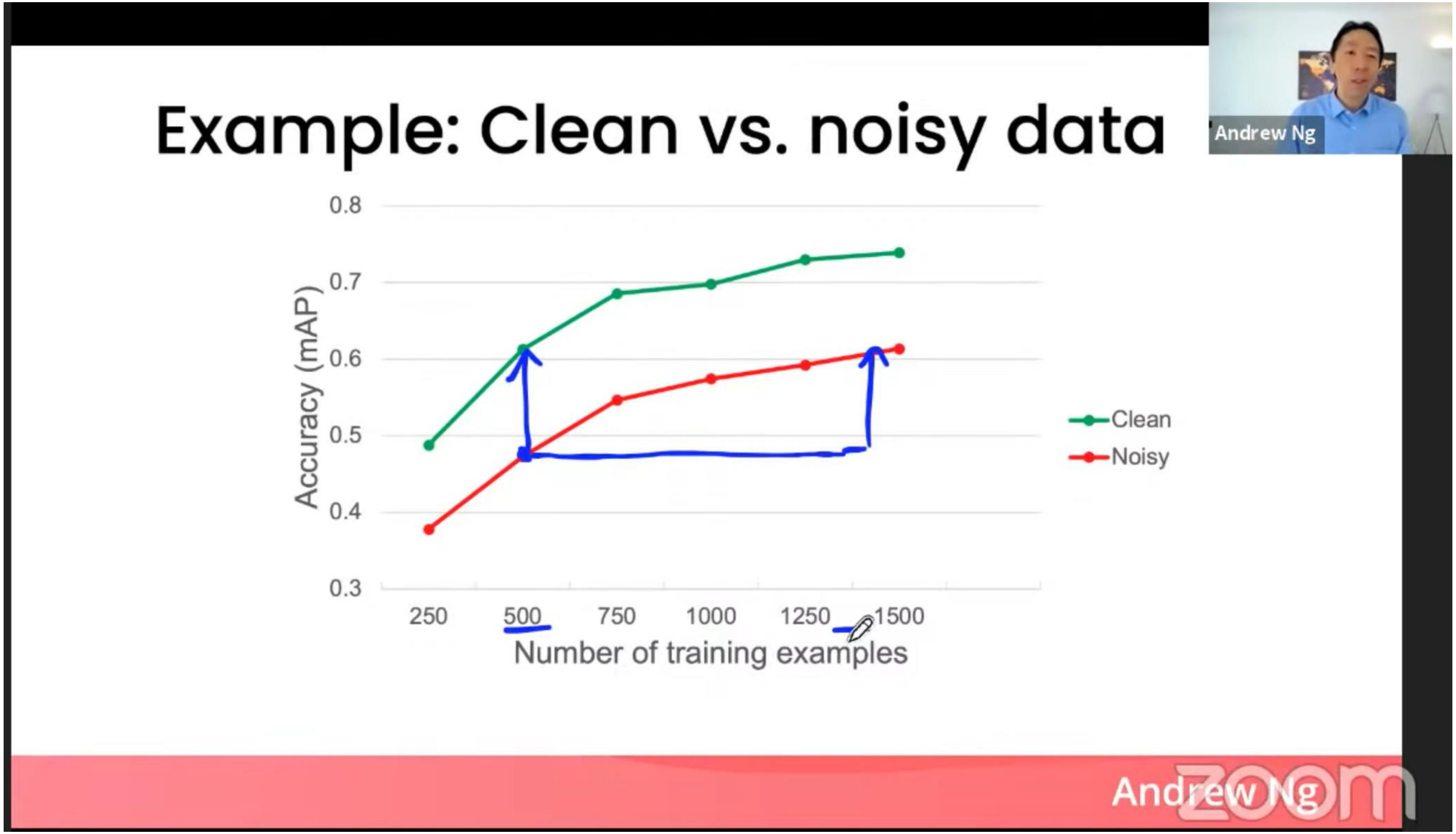
Example: RNASeq count table

Data cleaning/preprocessing/representation

Data Preprocessing is a technique that is used to convert the raw data into a “clean” data set



https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#58fdfc6f637d



https://www.youtube.com/watch?v=06-AZXmwHjo

What is BIG DATA?

BIG DATA, many definitions and aspects

Volume

- Large number of observations (but how many, 1000, 1000000?)
- Large number of object features
- Large volume : difficult to manipulate on a single computer

Variety

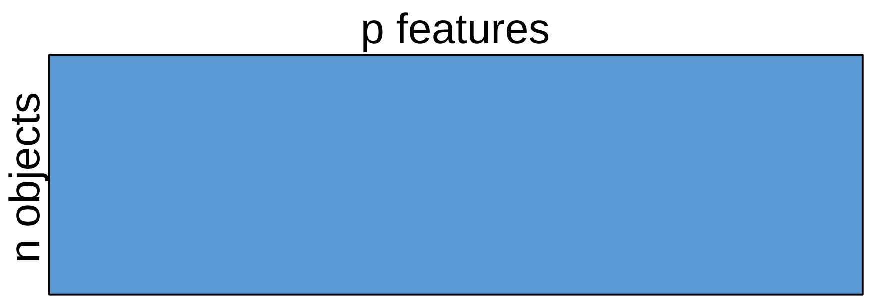
- Large variety of feature types (completeness of object characterization)

Velocity

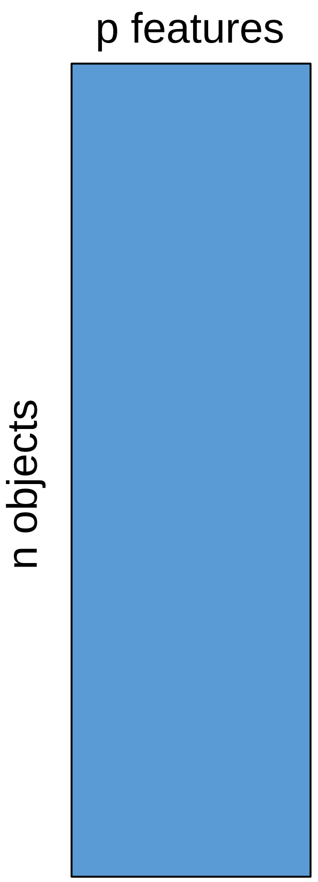
- The speed at which the data is generated and processed

Large p, small n

Wide data



Classical statistics



BIG DATA: n >> 1
WIDE DATA: p>>n
BIG DATA IN GENOMICS: p>>n>>1 (frequently)

WIDE DATA IN GENOMICS: p>>n

p	*n*
Length of genome : 3×10^9	
Number of genes : $\sim 10^4$	Number of tumors in a typical retrospective study: $\sim 10^2$
Number of proteforms : $\sim 10^5$	
Number of SNPs : $\sim 10^6$	
Number of CpGs : $\sim 10^7$	

Special case: single cell datasets (question: is it a "big data" or not)

D. Donoho, from Stanford University webpage

High-dimensional post-classical world: Big Data, Bigger Dimension

- **The number of attributes *p* >> The number of examples *N***
- This *post-classical* world is different from the '*classical world*'.
- The classical methodology was developed for the 'classical world' based on the assumption of $p < N$, and $N \to \infty$.
- These results all fail if $p > N$.
- **The *p* > *N* case is not anomalous; it is the generic case.**

Donoho, D.L. High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. Invited Lecture at Mathematical Challenges of the 21st Century, AMS.

What is “curse of dimensionality”?

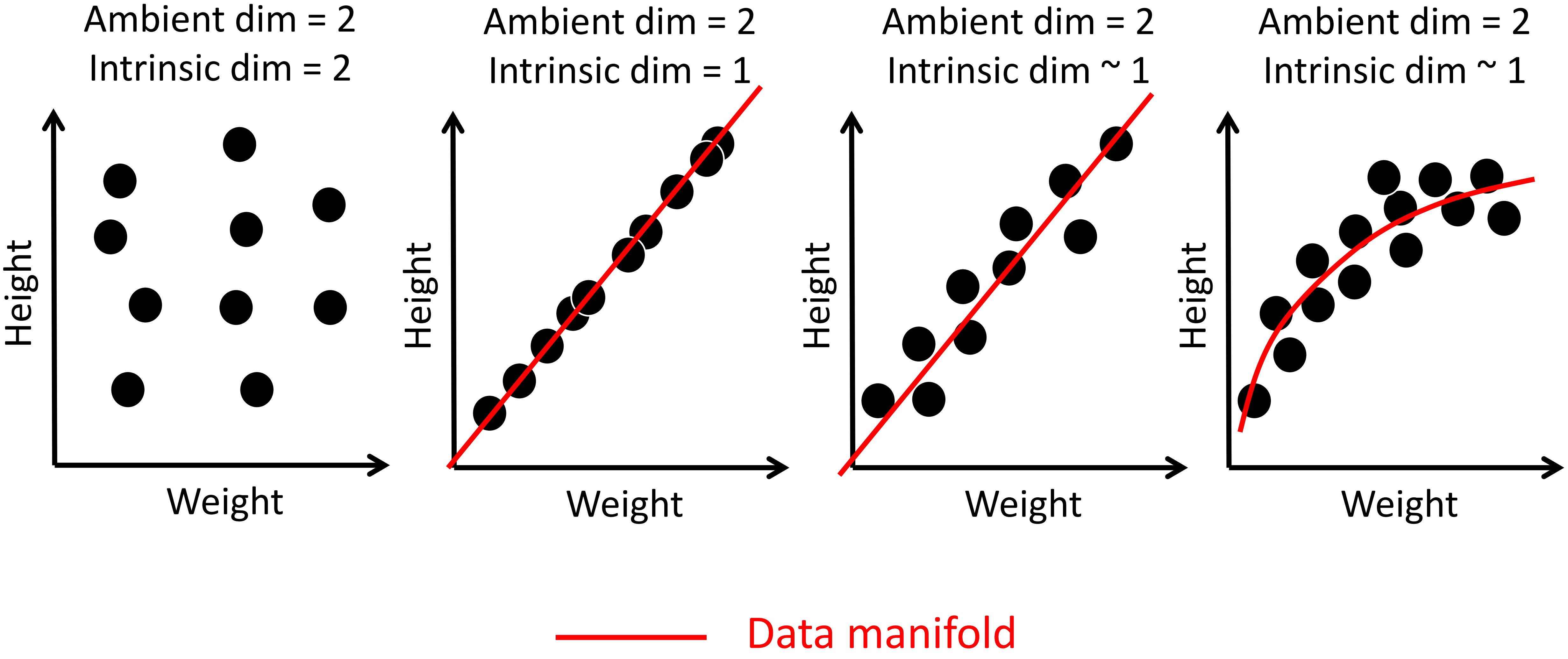
Curse of dimensionality and intrinsic data dimension

- Curse of dimensionality : various phenomena that arise when analyzing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings

- p = **ambient (full) dimensionality** (number of variables after data preprocessing)

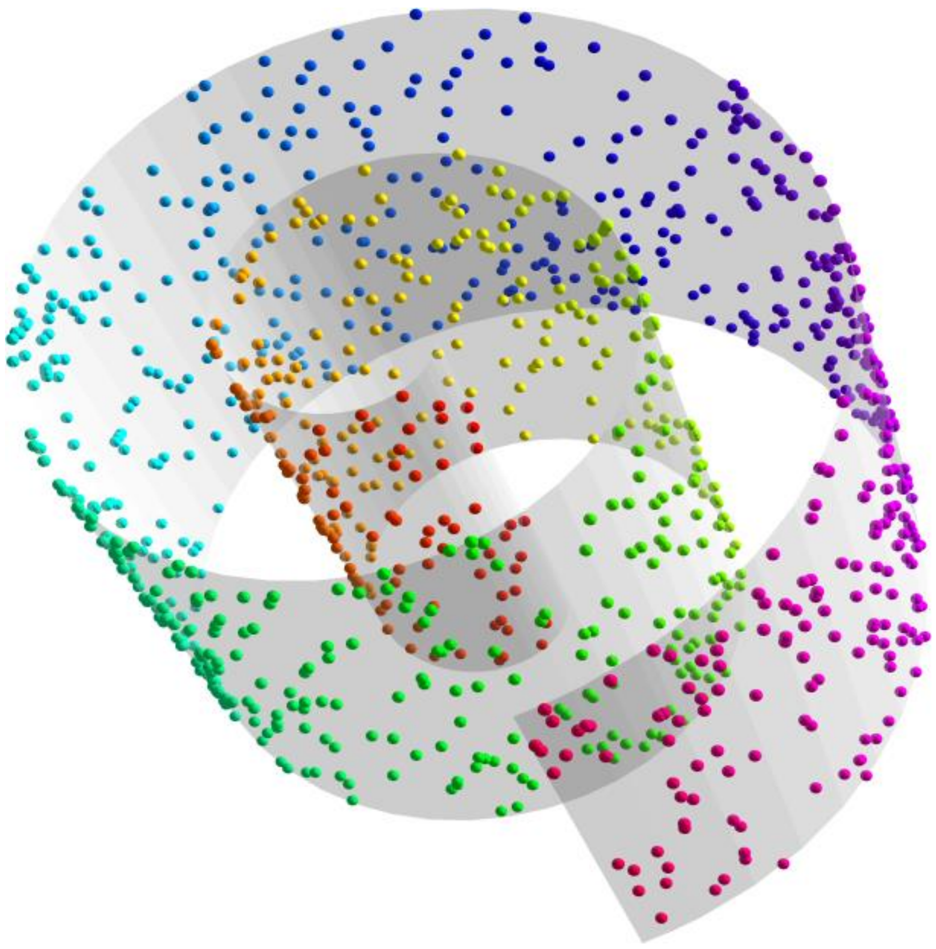
- However, in many cases, variables contain partially redundant information

- Intrinsic dimensionality (ID): ‘how many variables are needed to generate a good approximation of the data’



Ambient dim = 2
Intrinsic dim = 2
Height
Weight
Ambient dim = 2
Intrinsic dim = 1
Height
Weight
Ambient dim = 2
Intrinsic dim ~ 1
Height
Weight
Ambient dim = 2
Intrinsic dim ~ 1
Height
Weight
Data manifold

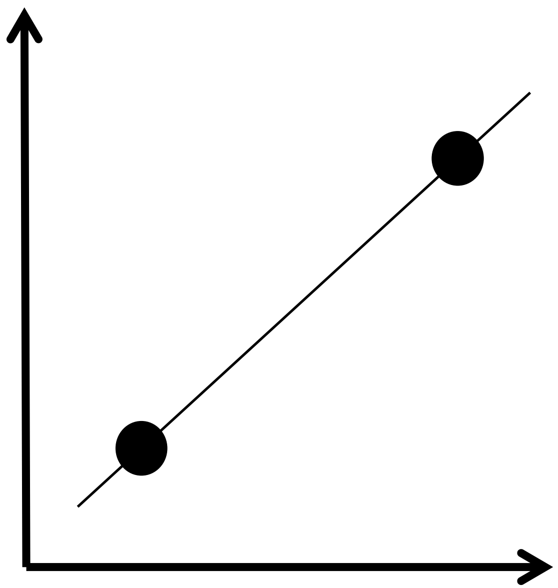
Swiss roll dataset



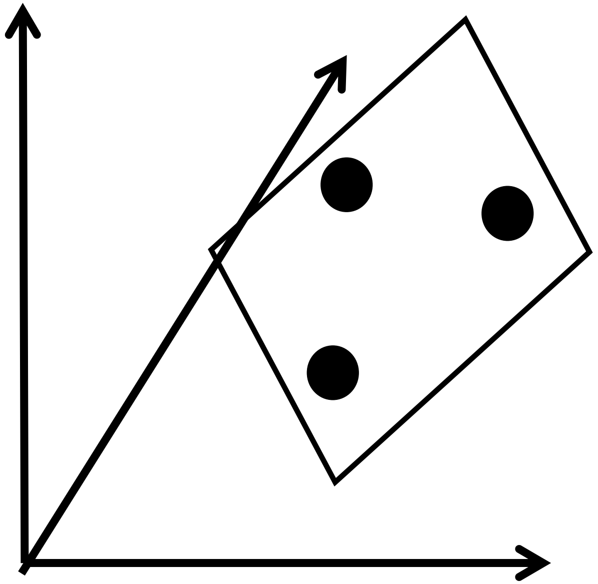
Ambient dim = 3
Intrinsic dim = 2

Intrinsic dimensionality can not be bigger than the number of data points minus 1

Number of data points = 2
Ambient dim = 2
Intrinsic dim = 1



Number of data points = 3
Ambient dim = 3
Intrinsic dim = 2



Curse of dimensionality and genomics data

~~When number of features >> number of objects~~

When the *intrinsic dimension of the data* > log2(number of objects)

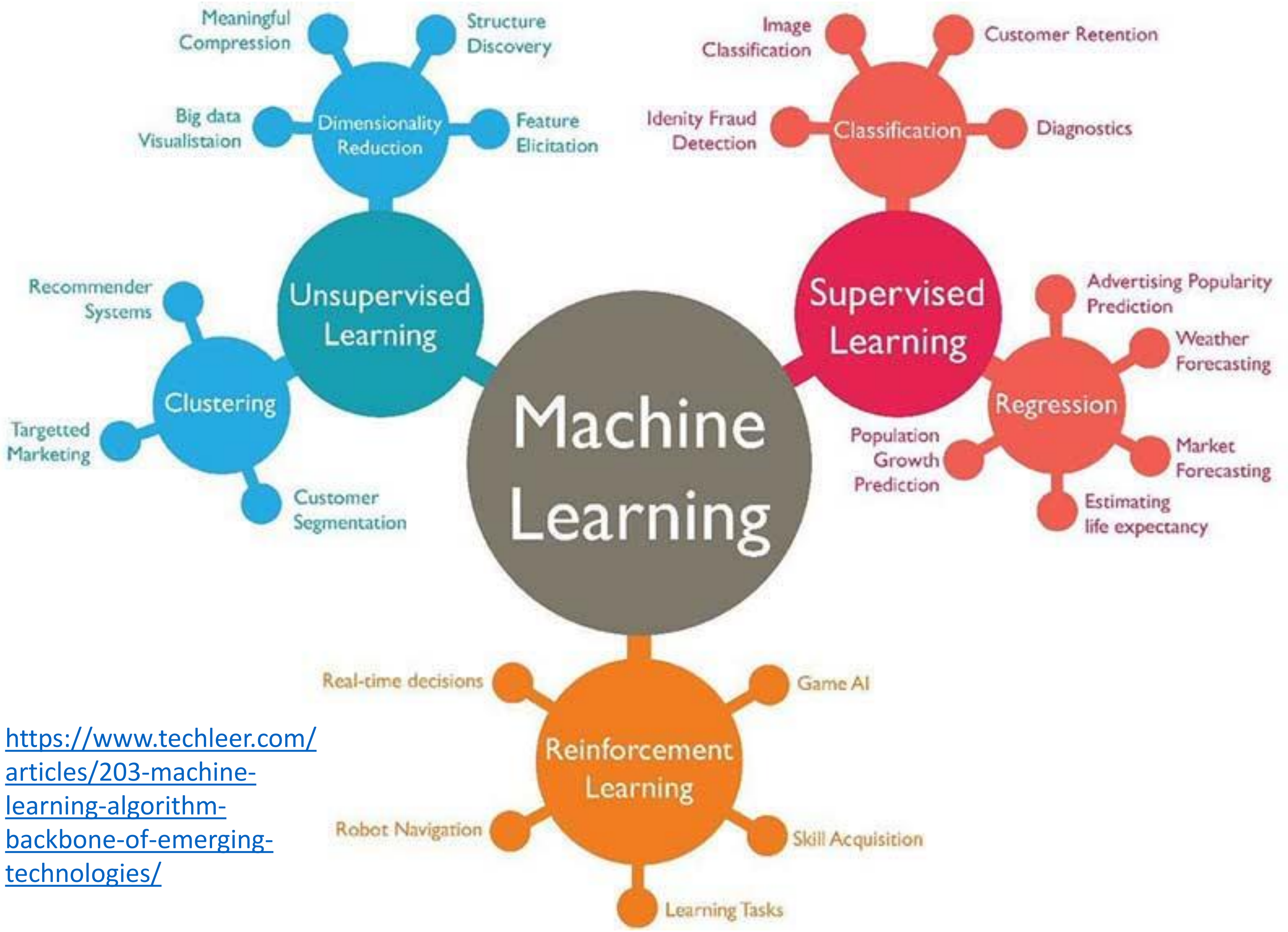
Fortunately, genomics data are frequently intrinsically relatively low-dimensional

For example, ID of typical transcriptomic datasets can be estimated in 20-30 (may be, even much less)

Some types of genomics data are intrinsically high-dimensional (e.g., mutation matrices)

What are the types of machine learning models?

Types of machine learning approaches



https://www.techleer.com/articles/203-machine-learning-algorithm-backbone-of-emerging-technologies/

Self-supervised learning: Pretend there is a part of the input you do not know and predict that [Y.Le Cun] *Language models, watching videos and predicting the future frames, AlphaZero …*

Flavors or special tricks: *Representation learning, transfer learning, one-shot learning, semi-supervised learning etc…*

What is the difference between classification and regression?

Supervised learning

"Data"

Independent or explanatory variables : X

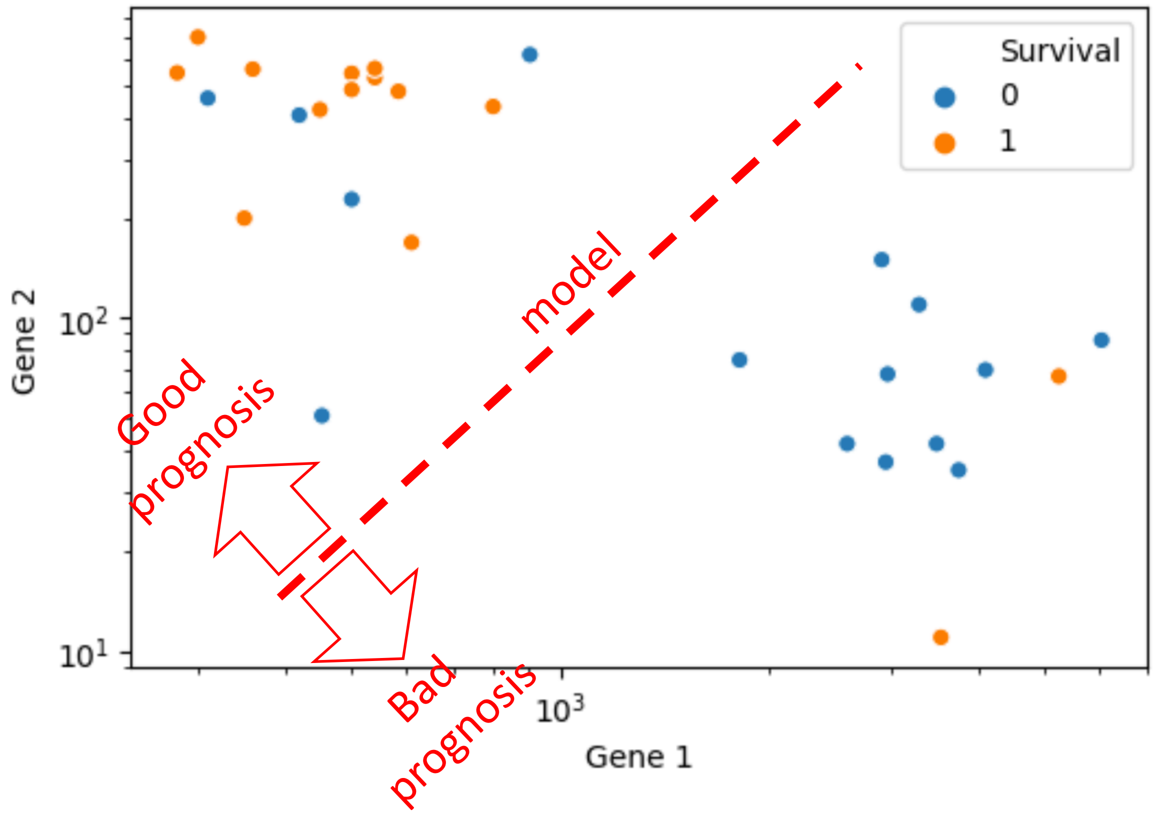
"Labels"

Dependent variables : y

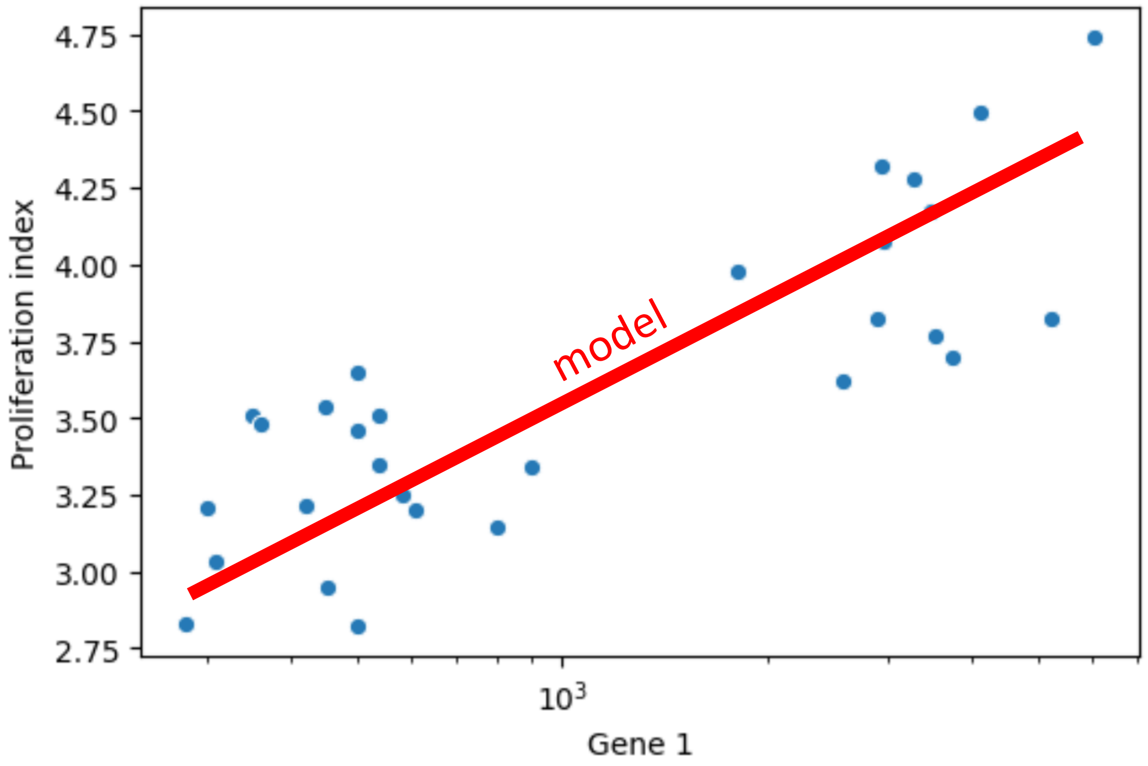
	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Gene 6	Gene 7	Gene 8	Gene 9	Gene 10	Survival	Proliferation index
Sample 1	300	700	270	38	0	0	38	0	0	0	0	0.37940
Sample 2	584	481	437	131	43	43	87	0	0	0	0	0.45072
Sample 3	350	200	114	0	114	114	0	0	0	0	0	0.63810
Sample 4	280	547	429	0	39	117	0	0	0	78	0	0.92688
Sample 5	450	424	196	98	32	196	0	0	0	0	1	0.20938
Sample 6	500	545	72	36	0	36	0	36	0	0	1	0.04551
Sample 7	610	169	169	0	0	0	169	0	0	0	1	0.33923
Sample 8	500	228	114	0	57	57	57	0	57	0	0	0.49039
Sample 9	540	529	721	48	48	0	48	48	0	48	0	0.09787
Sample 10	500	487	205	102	25	51	0	51	0	0	1	0.86256
Sample 11	800	433	166	66	66	0	99	0	0	0	1	0.91319
Sample 12	420	408	363	0	0	0	0	136	45	0	0	0.85531
Sample 13	540	564	333	102	25	25	25	25	0	0	1	0.36976
Sample 14	310	459	459	86	86	0	28	57	28	57	0	0.73904
Sample 15	360	561	280	62	31	0	62	31	0	0	1	0.69861
Sample 16	904	620	212	35	35	53	35	35	35	0	0	0.46501
Sample 17	3490	42	213	71	35	0	106	0	0	0	1	0.70675
Sample 18	453	51	647	64	129	0	129	0	0	0	0	0.82493
Sample 19	2948	37	0	0	61	61	61	0	0	0	1	0.30731
Sample 20	4105	70	274	18	54	36	73	18	18	18	0	0.87440

Supervised learning

Classification



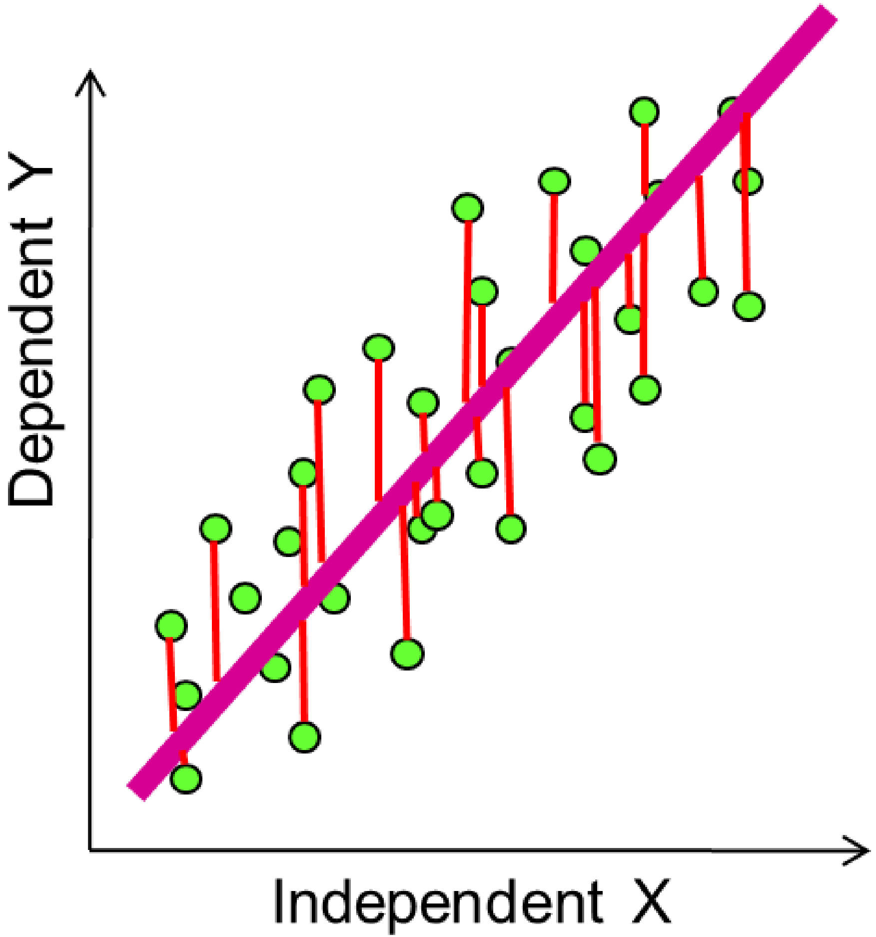
Regression



Problem of visualization

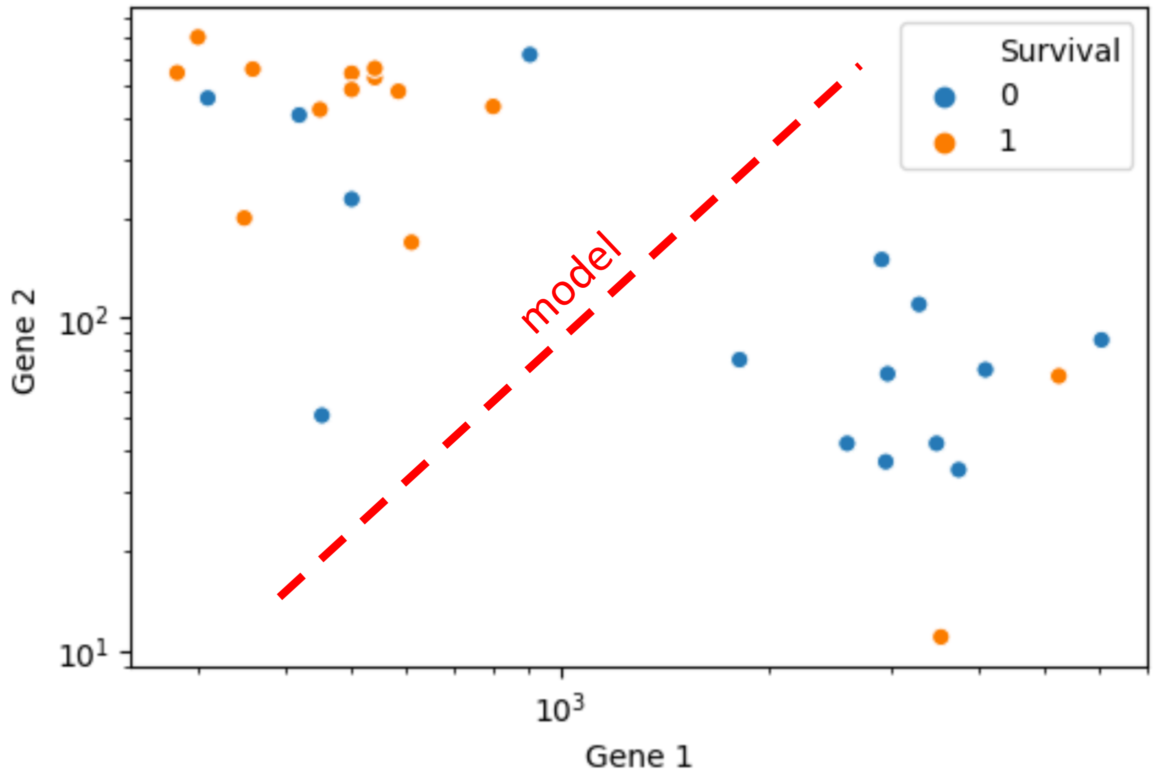
Mean Squared Error and R^2

- Linear regression minimizes the squared sum of **residuals (model errors)**
- **MSE = Mean Squared Error**



$$\sum_{i=1}^{m} \left\| \quad \right\|^2 \to \min$$

Classification error

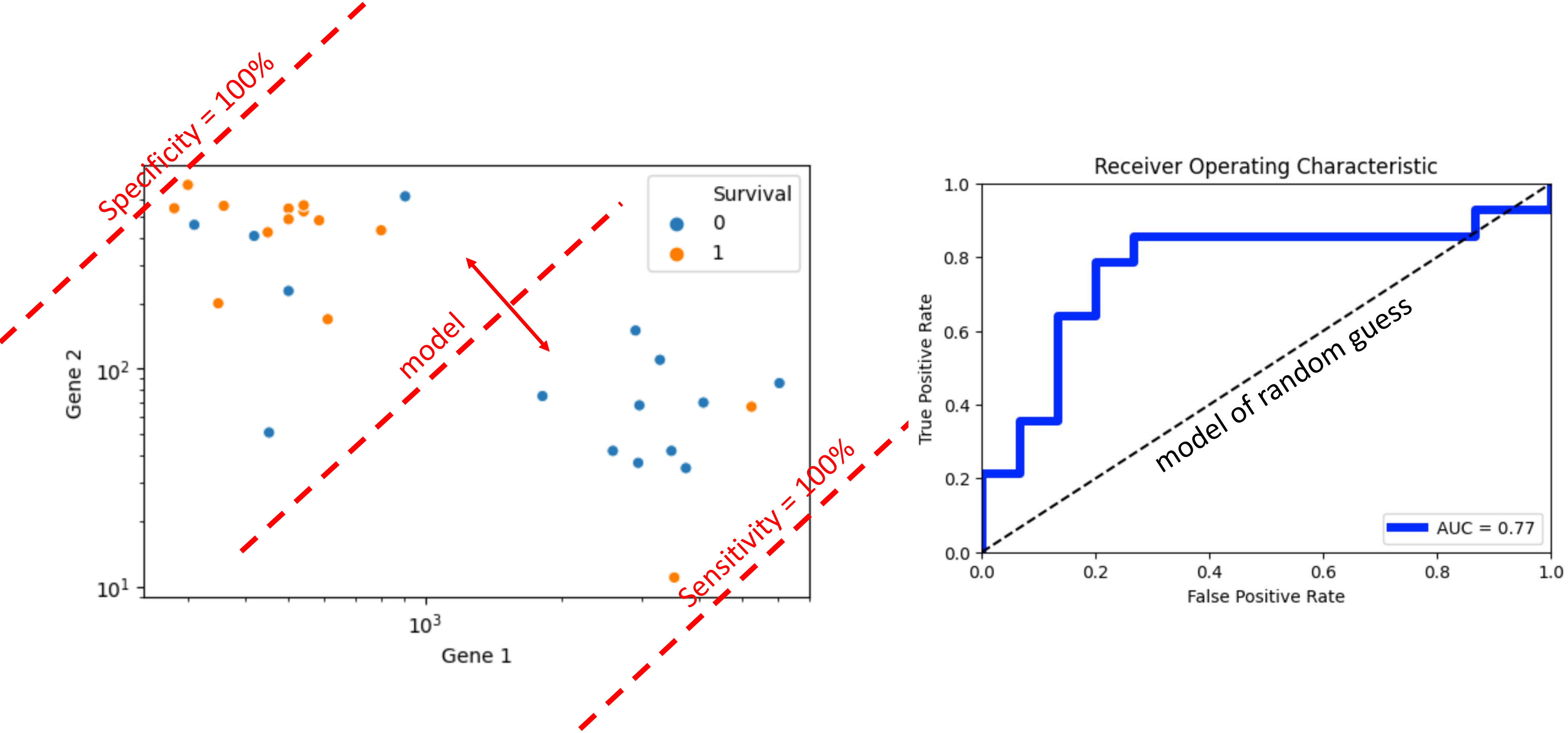


Survivial Prediction Error

True Positives (TP) = 12
True Negatives (TN) = 10
False Positives (FP) = 5
False Negatives (FN) = 2

Accuracy = (TP+TN)/all = 76%
Sensitivity = TP/(TP+FN) = 86%
Specificity = TN/(TN+FP) = 66%

Classification error: ROC curve and AUC



What is the difference between dimensionality reduction and clustering?

Unsupervised learning

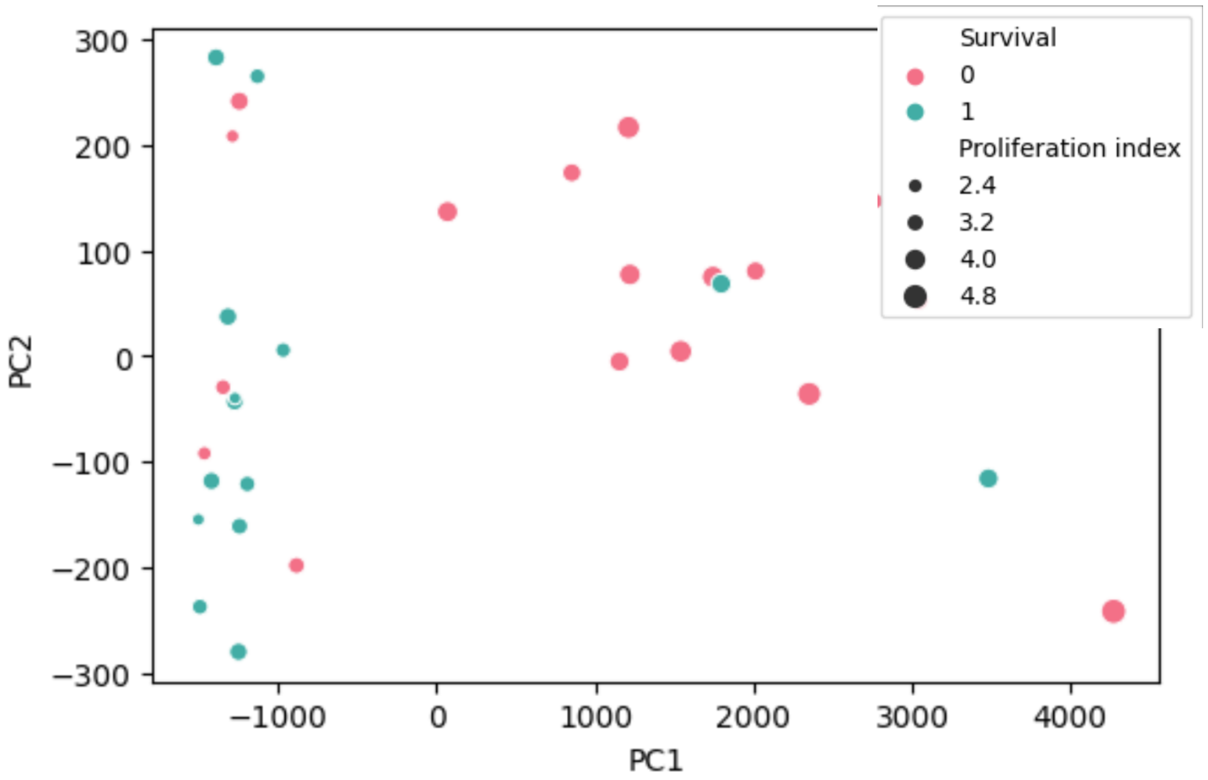
We do not use them to build the model

Independent or explanatory variables : X | **Dependent variables : y**

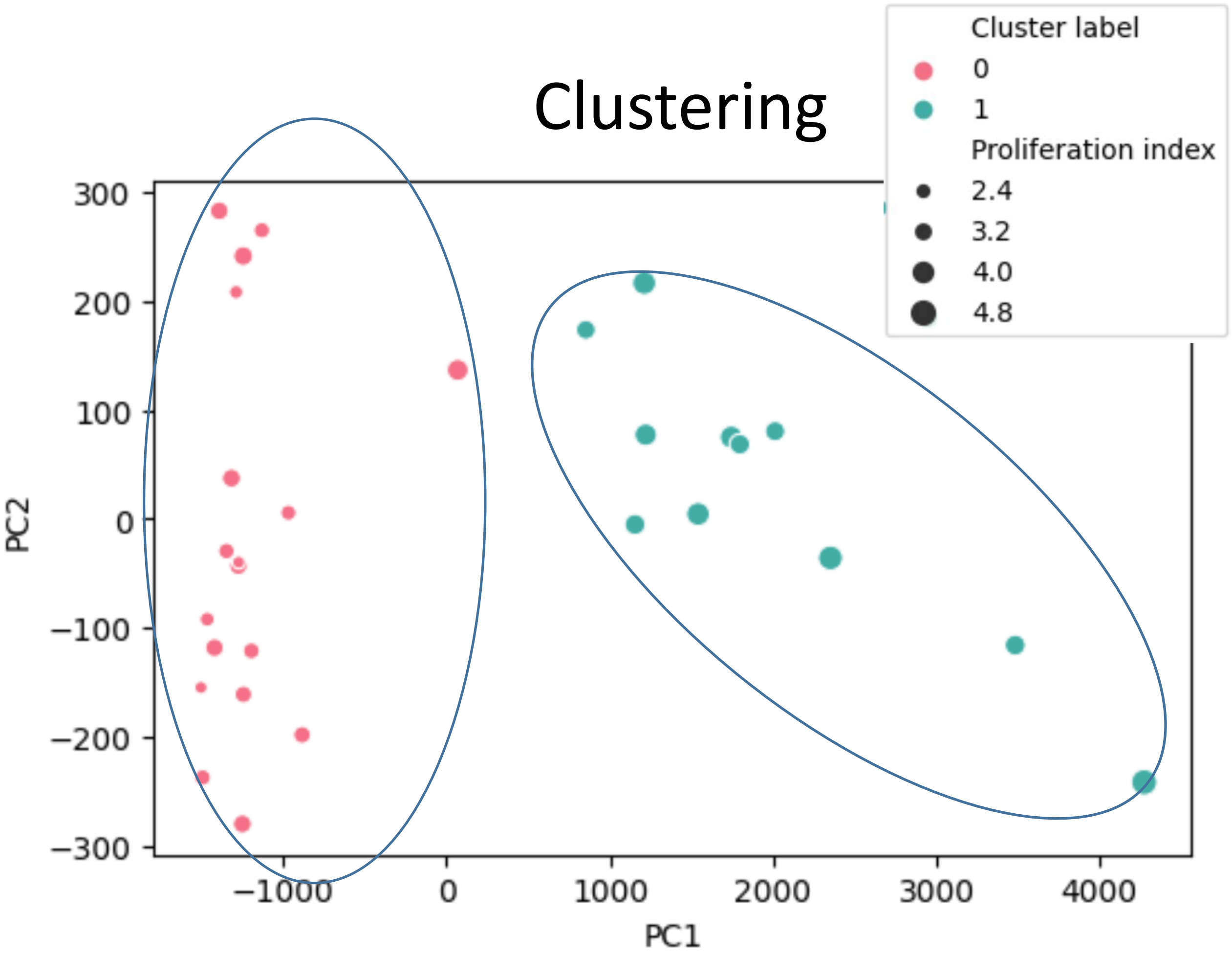
	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Gene 6	Gene 7	Gene 8	Gene 9	Gene 10	Survival	Proliferation index
Sample 1	300	700	270	38	0	0	38	0	0	0	0	0.37940
Sample 2	584	481	437	131	43	43	87	0	0	0	0	0.45072
Sample 3	350	200	114	0	114	114	0	0	0	0	0	0.63810
Sample 4	280	547	429	0	39	117	0	0	0	78	0	0.92688
Sample 5	450	424	196	98	32	196	0	0	0	0	1	0.20938
Sample 6	500	545	72	36	0	36	0	36	0	0	1	0.04551
Sample 7	610	169	169	0	0	0	169	0	0	0	1	0.33923
Sample 8	500	228	114	0	57	57	57	0	57	0	0	0.49039
Sample 9	540	529	721	48	48	0	48	48	0	48	0	0.09787
Sample 10	500	487	205	102	25	51	0	51	0	0	1	0.86256
Sample 11	800	433	166	66	66	0	99	0	0	0	1	0.91319
Sample 12	420	408	363	0	0	0	0	136	45	0	0	0.85531
Sample 13	540	564	333	102	25	25	25	25	0	0	1	0.36976
Sample 14	310	459	459	86	86	0	28	57	28	57	0	0.73904
Sample 15	360	561	280	62	31	0	62	31	0	0	1	0.69861
Sample 16	904	620	212	35	35	53	35	35	35	0	0	0.46501
Sample 17	3490	42	213	71	35	0	106	0	0	0	1	0.70675
Sample 18	453	51	647	64	129	0	129	0	0	0	0	0.82493
Sample 19	2948	37	0	0	61	61	61	0	0	0	1	0.30731
Sample 20	4105	70	274	18	54	36	73	18	18	18	0	0.87440

Unsupervised learning

Dimensionality reduction



Clustering



Should data scientist understand the underlying principles (mathematics) of machine learning methods?

Any AI (ML) method in four lines of code in *any* programming language

```
from libraryA import ModelType

model = ModelType(ModelParameter=par)

model.fit(X,Y)

model.predict(X)

model.score(X,Y)
```

The rest is either data pre-processing or presenting the results…

Any AI (ML) method in four lines of code in *any* programming language

```
from libraryA import ModelType

model = ModelType(ModelParameter=par)

model.fit(X,y)

model.predict(X)

model.score(X,y)
```

Select model type

Fit the model to data

Predict using the model

Ask if the model is good

The rest is either data pre-processing or presenting the results…

“Zoo” of machine learning

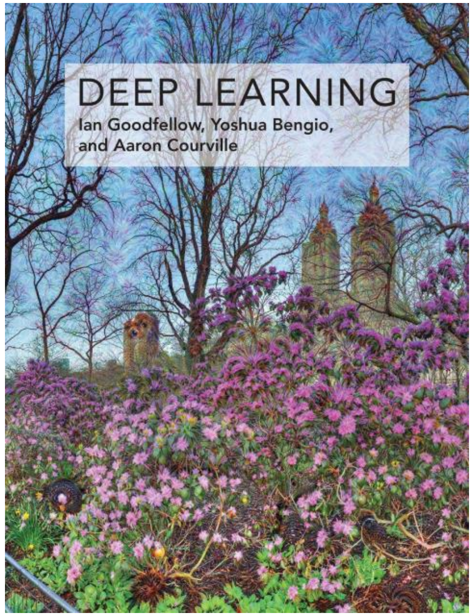
- “model.fit(),model.predict()” technological revolution makes machine learning technically accessible to almost anyone without strong background in mathematics
- This creates an illusion that this background is not needed
- This gives an impression that machine learning is a “zoo of algorithms”
- This attitude is pragmatic but VERY limited, also in applications
- Understanding mathematical principles helps in choosing learning **hyperparameters**
- Unfortunately, there is no unifying theory of machine learning created yet

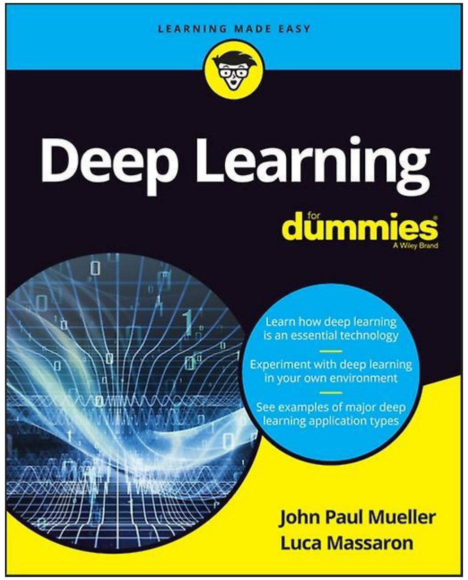
Myth of deep learning

- No need in zoo of machine learning methods
- No need to understand math behind
- One just need DEEP LEARNING

- However, despite the hype, deep learning probably accounts for less than 1% of the machine learning projects in production right now. Most of the recommendation engines and online adverts that you encounter when you browse the net are not powered by deep learning. Most models used internally by companies to manage their subscribers, for example churn analysis, are not deep learning models. The models used by credit institutions to decide who gets credit do not use deep learning

https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788992893/1/ch01lvl1sec13/some-common-myths-about-deep-learning





<u>Supervised learning:</u>
What is the linear regression model?

Linear regression model

Data

Prediction

$$F_{\beta_1,\beta_2,\ldots,\beta_k}(x_1, x_2, \ldots, x_p) = y$$

Parameters

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$

Number of parameters = number of features + 1

	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Gene 6	Gene 7	Gene 8	Gene 9	Gene 10	β_0	Prediction
Sample 1	300	700	270	38	0	0	38	0	0	0		
β	0.1	0.3	-0.9	2.3	5.3	-6	0.8	1.1	0.1	-0.2	-110	4.8

Linear regression

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\mathsf{T} \boldsymbol\beta + \varepsilon_i, \qquad i = 1, \ldots, n,$$

$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \epsilon$

where, for $i = n$ observations:

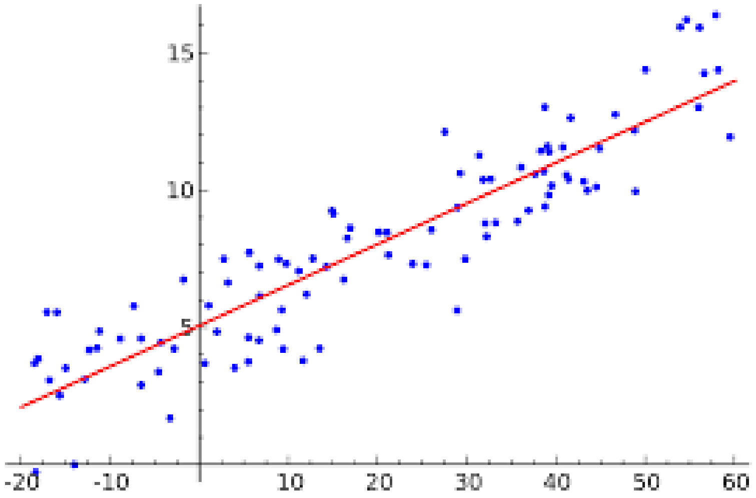
y_i = dependent variable

x_i = expanatory variables

β_0 = y-intercept (constant term)

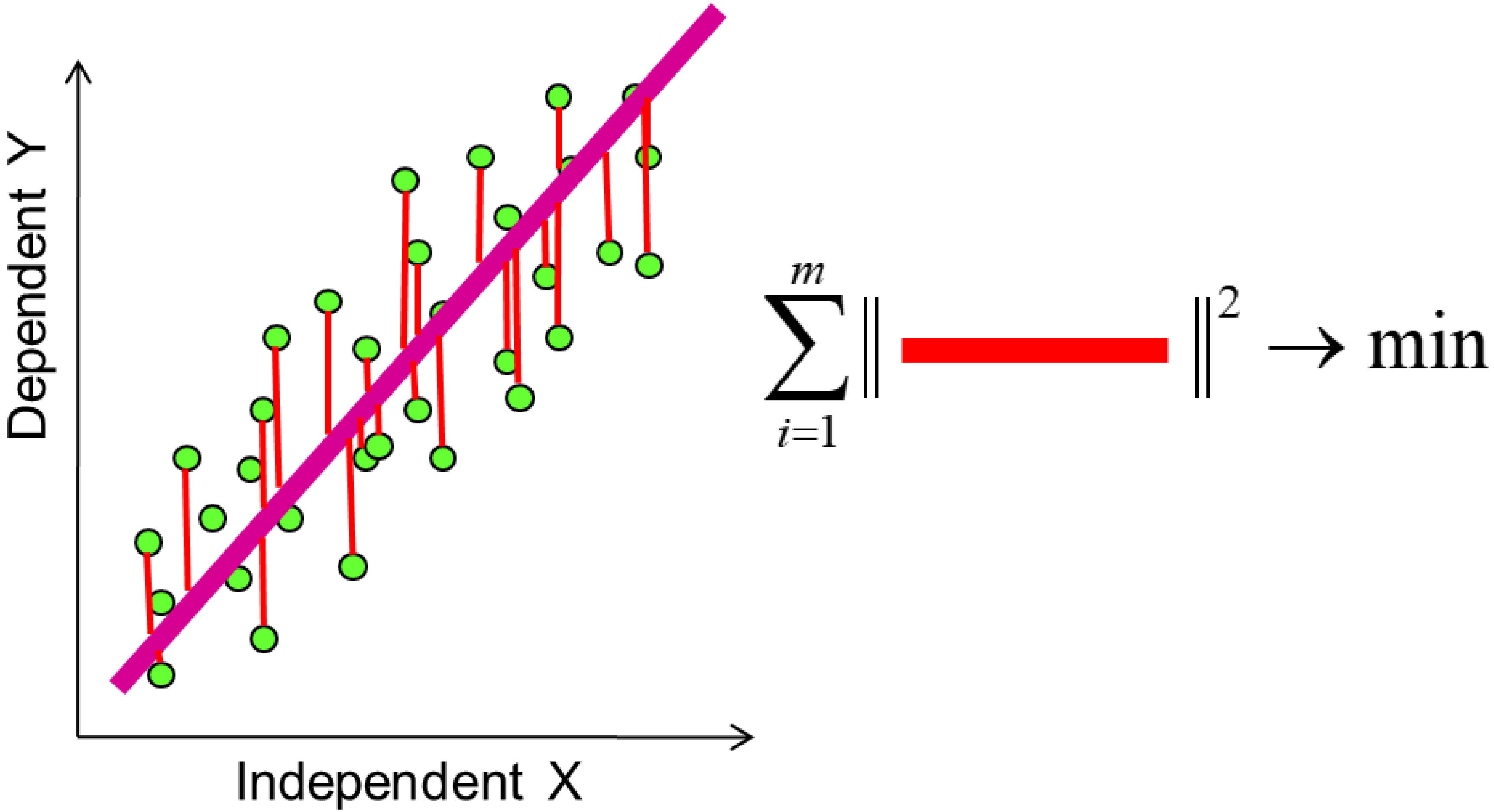
β_p = slope coefficients for each explanatory variable

ϵ = the model's error term (also known as the residuals)



How the parameters of linear regression are computed?

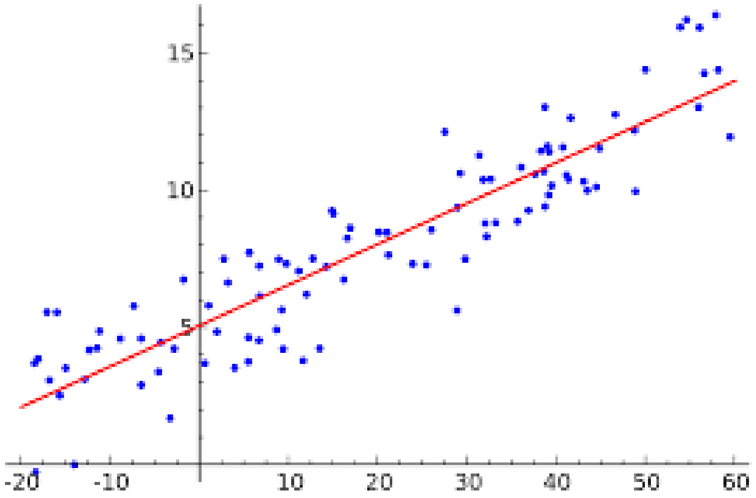
- Linear regression minimizes the squared sum of **residuals (model errors)**
- **MSE = Mean Squared Error**



Linear regression

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

- Linear regression: the father of all supervised machine learning methods (the idea comes from 1805!)
- The most used machine learning method today
- The first machine learning method to apply, and see what it gives
- Linear regression can be used to produce non-linear data models



Linear regression is explainable ML model!

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\mathsf{T} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

Coefficients β_1, β_2, … , β_p are comparable if independent variables are standardized (to z-scores) and have straingforward interpretation

It is possible to estimate statistical significance of β_i coefficients and provide p-value on the hypothesis that the coefficient is non-zero

This can help to simplify the regression

Other methods (such as regularization by lasso) for selecting important variables are readily available

Linear regression caveats

- Main problem : Large p, small n
- If p is large and intrinsic dimension of X is small: **many correlated features**! The definition of parameters becomes unstable
- If p is large and intrinsic dimension of X is high : many features are non-relevant, problem of **overfitting**
- Well developed methodology for dealing with these problems: **regularization**

Regularized linear regression

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

Ridge regularization : make the sum $\sum \beta_i^2$ as small as possible among all closely accurate regression models

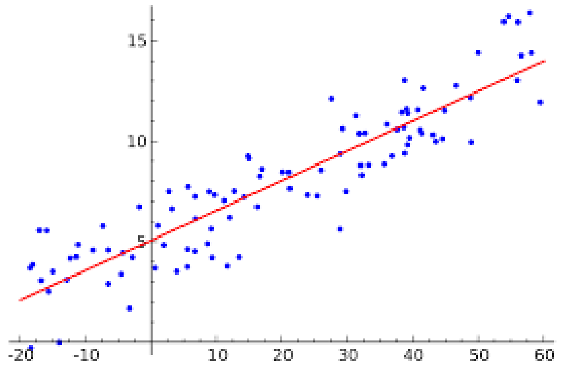
Lasso regularization : make the sum $\sum |\beta_i|$ as small as possible among all closely accurate regression models

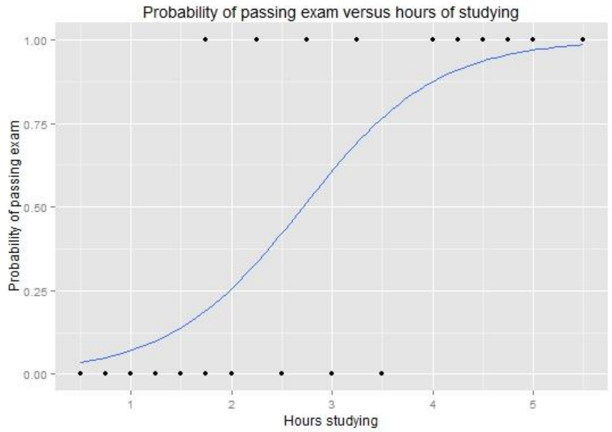
Linear regression and its close relatives

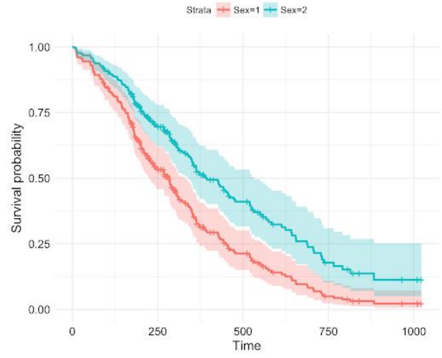
- **Ordinary Least Square** regression : when the dependent variable is continuous

- **Logistic regression** (logit): when the dependent variable is discrete (for example, binary)

- **Survival Cox** linear regression : when the target variable is a pair (follow up time + event)

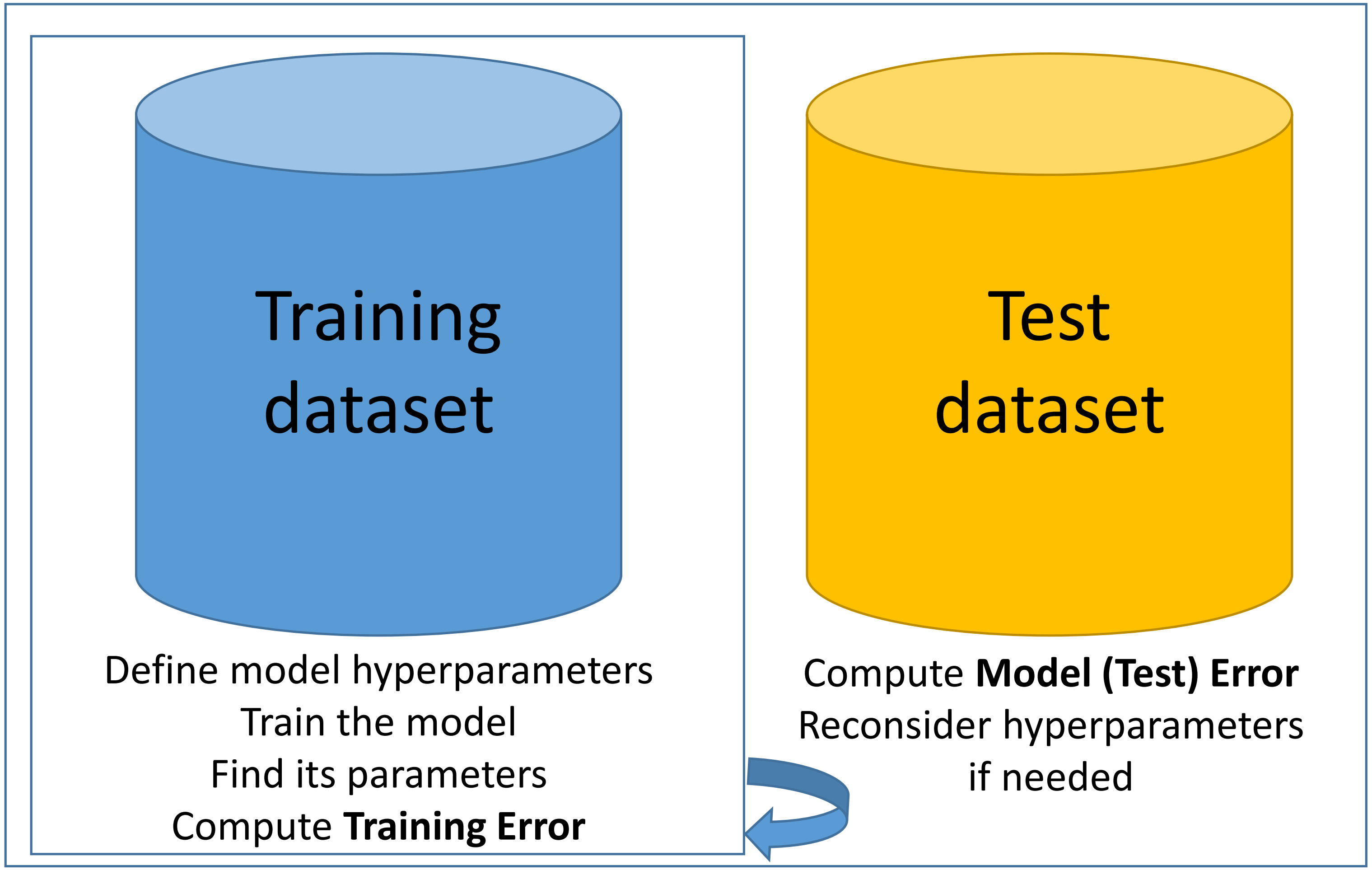






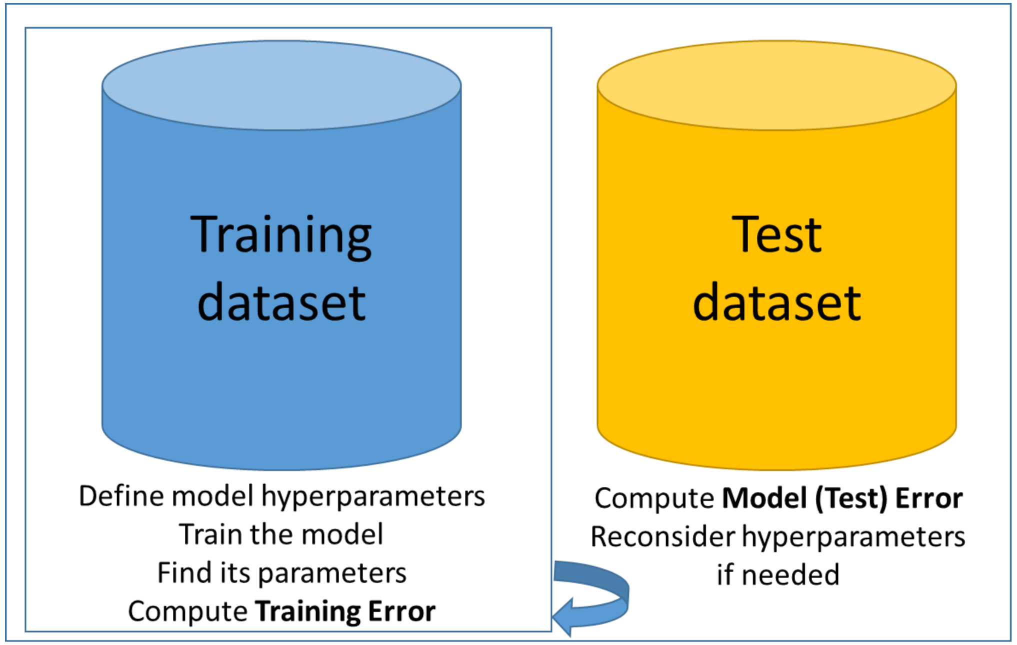
Supervised learning:
Validating machine learning models

To produce useful machine learning model, *three* types of datasets are needed (ideally)



Confusion in terminology ‘Test’ and ‘Validation’

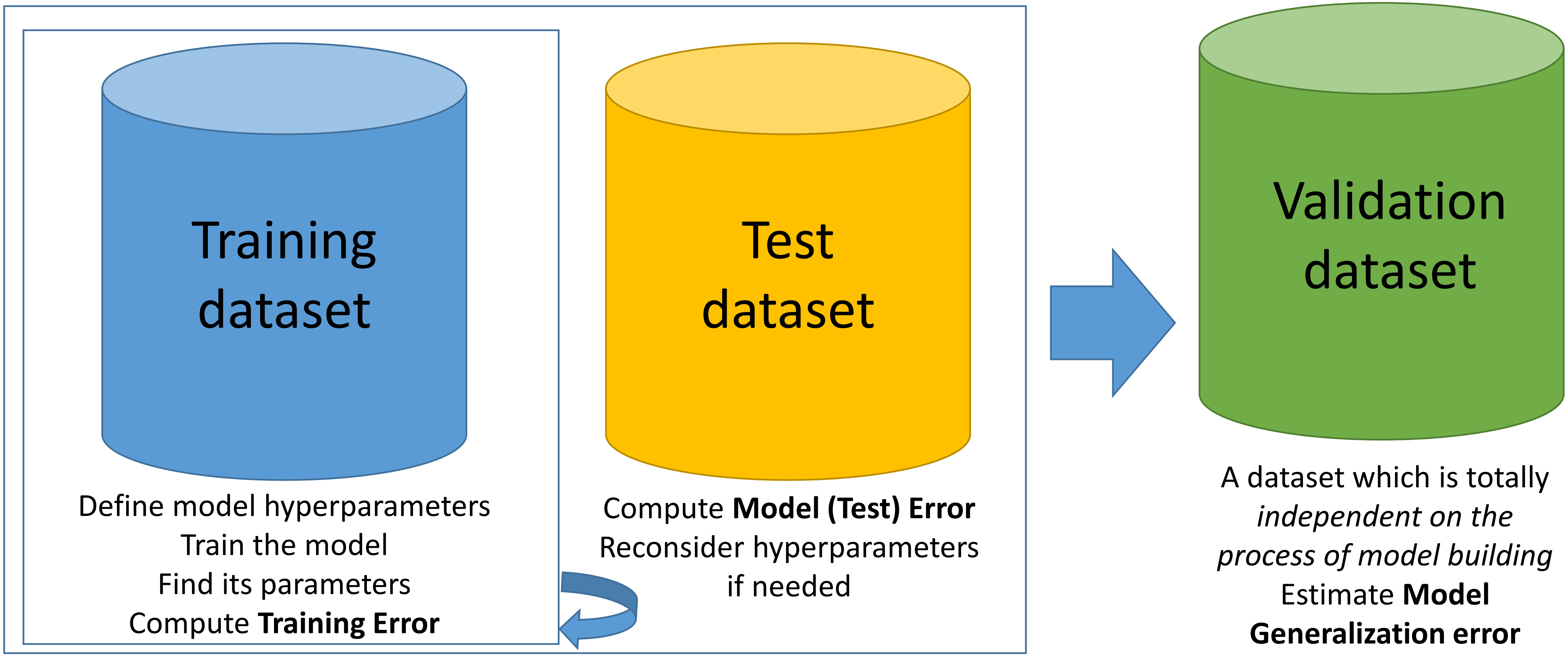
To produce useful machine learning model, *three* types of datasets are needed (ideally)



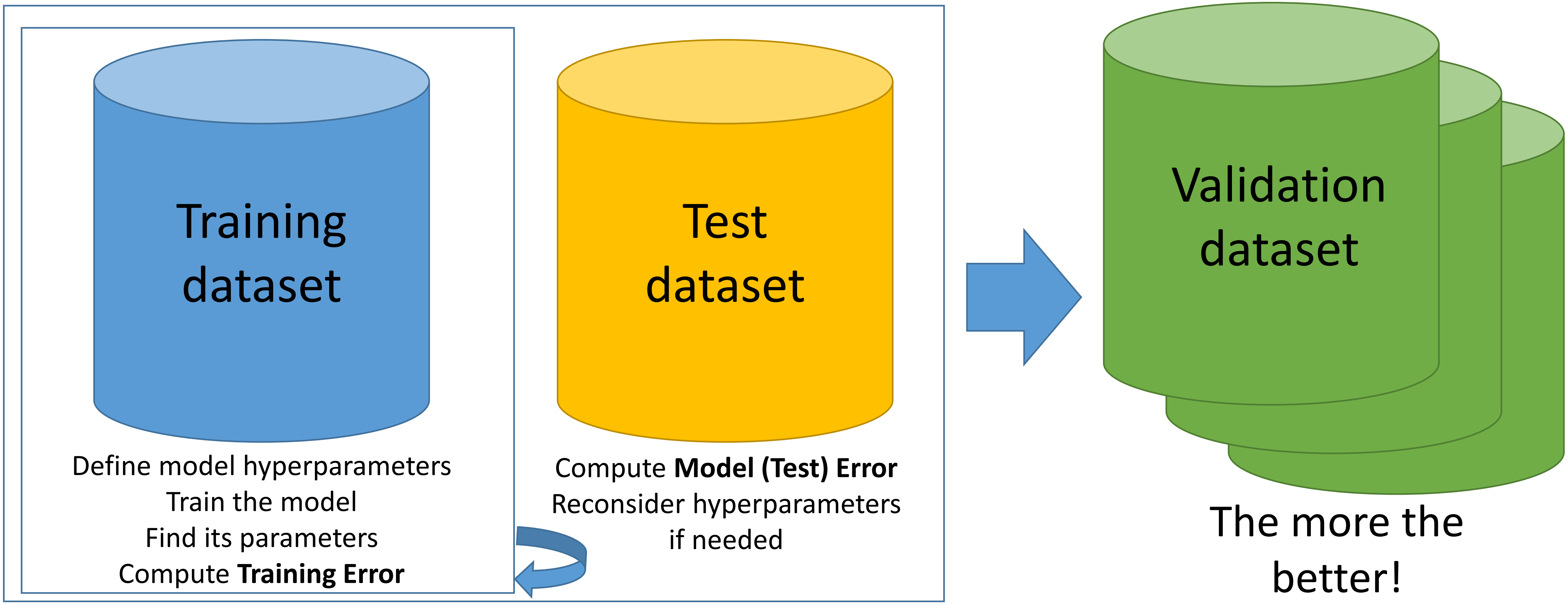
If there is no better choice…

	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5	Gene 6	Gene 7	Gene 8	Gene 9	Gene 10	Survival	Proliferation Index
Sample 1	300	700	270	38	0	0	38	0	0	0	0	0.37940
Sample 2	584	481	437	131	43	43	87	0	0	0	0	0.45072
Sample 3	350	200	114	0	114	114	0	0	0	0	0	0.63810
Sample 4	280	547	429	0	39	117	0	0	0	78	0	0.92688
Sample 5	450	424	196	98	32	196	0	0	0	0	1	0.20938
Sample 6	500	545	72	36	0	36	0	36	0	0	1	0.04551
Sample 7	610	169	169	0	0	0	169	0	0	0	1	0.33923
Sample 8	500	228	114	0	57	57	57	0	57	0	0	0.49039
Sample 9	540	529	721	48	48	0	48	48	0	48	0	0.09787
Sample 10	500	487	205	102	25	51	0	51	0	0	1	0.86256
Sample 11	800	433	166	66	66	0	99	0	0	0	1	0.91319
Sample 12	420	408	363	0	0	0	0	136	45	0	0	0.85531
Sample 13	540	564	333	102	25	25	25	25	0	0	1	0.36976
Sample 14	310	459	459	86	86	0	28	57	28	57	0	0.73904
Sample 15	360	561	280	62	31	0	62	31	0	0	1	0.69861
Sample 16	904	620	212	35	35	53	35	35	35	0	0	0.46501
Sample 17	3490	42	213	71	35	0	106	0	0	0	1	0.70675
Sample 18	453	51	647	64	129	0	129	0	0	0	0	0.82493
Sample 19	2948	37	0	0	61	61	61	0	0	0	1	0.30731
Sample 20	4105	70	274	18	54	36	73	18	18	18	0	0.87440

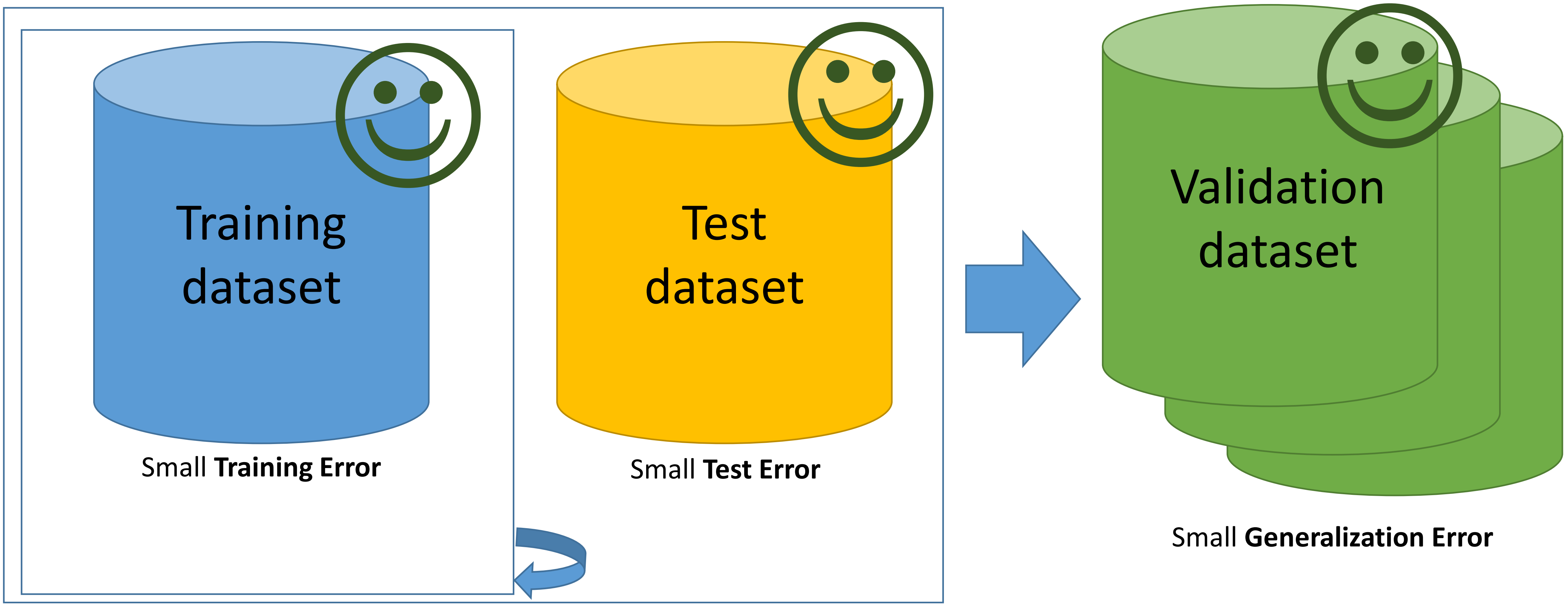
To produce useful machine learning model, *three* types of datasets are needed (ideally)



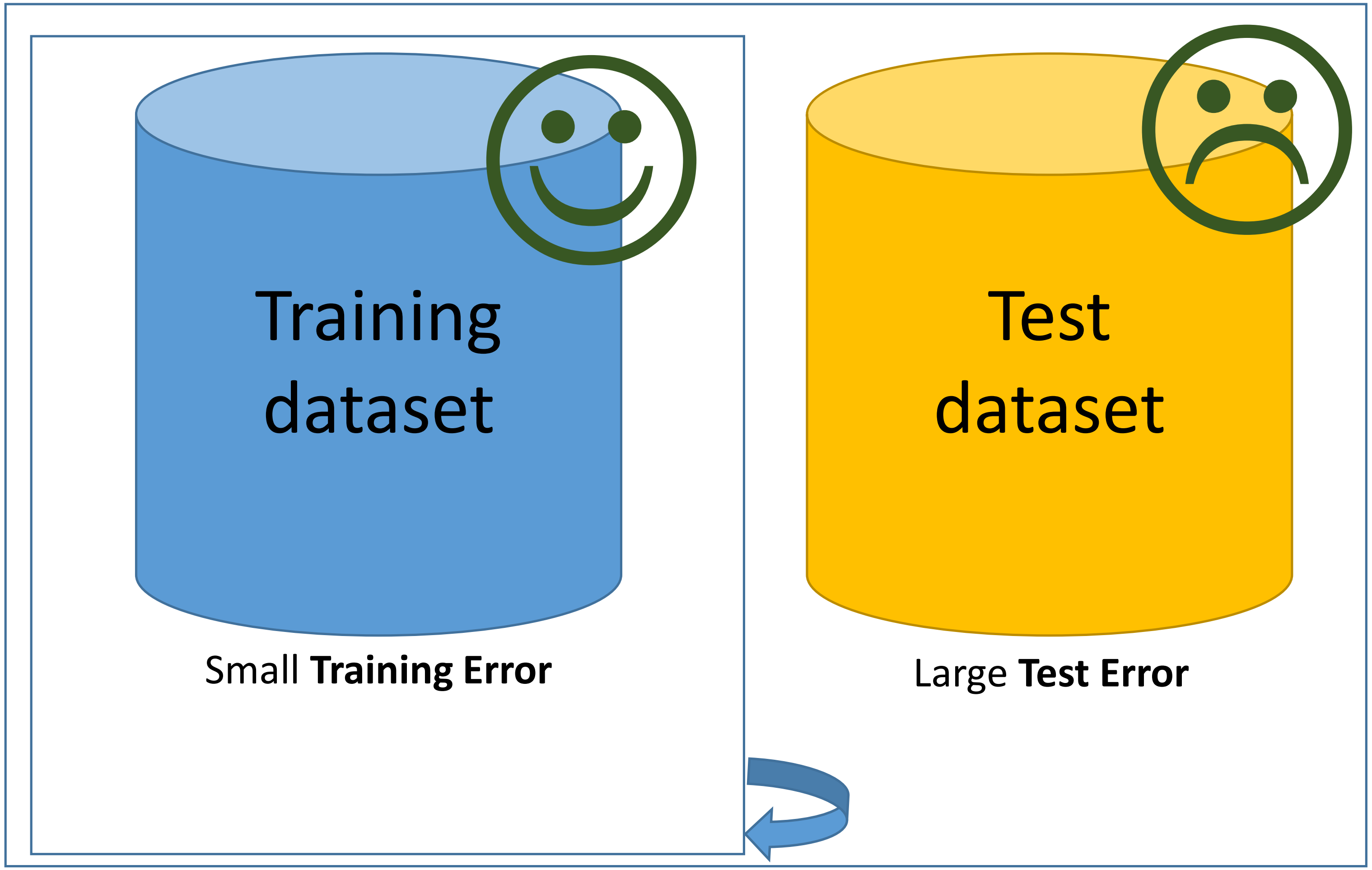
To produce useful machine learning model, *three* types of datasets are needed (ideally)



Great ML model!



Overfitting



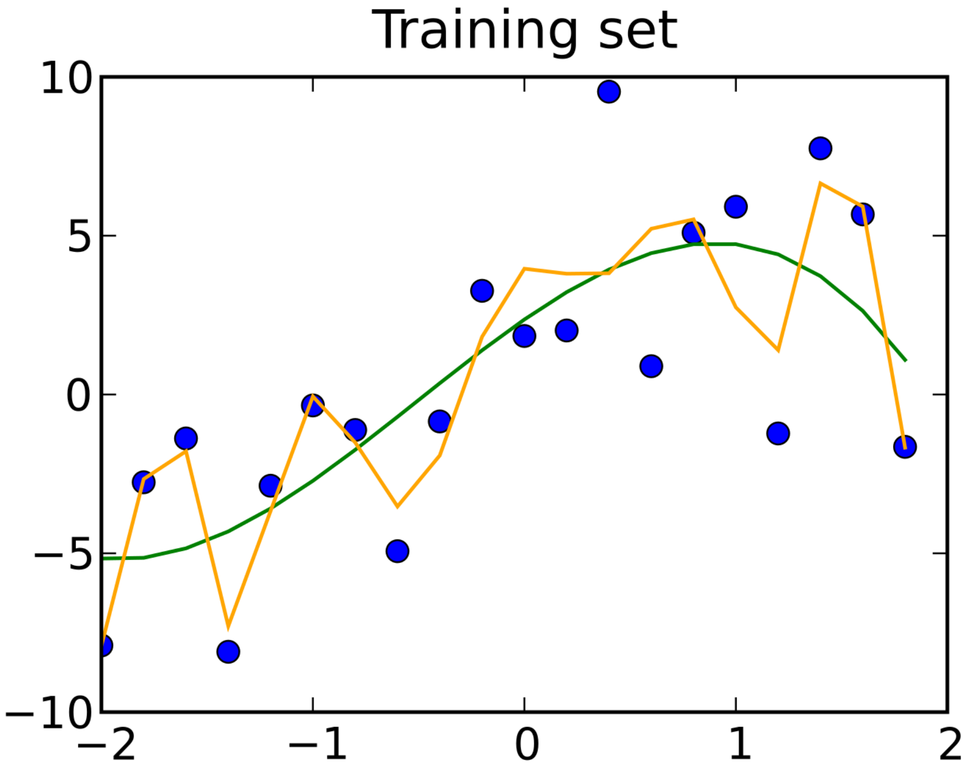
Causes for overfitting:

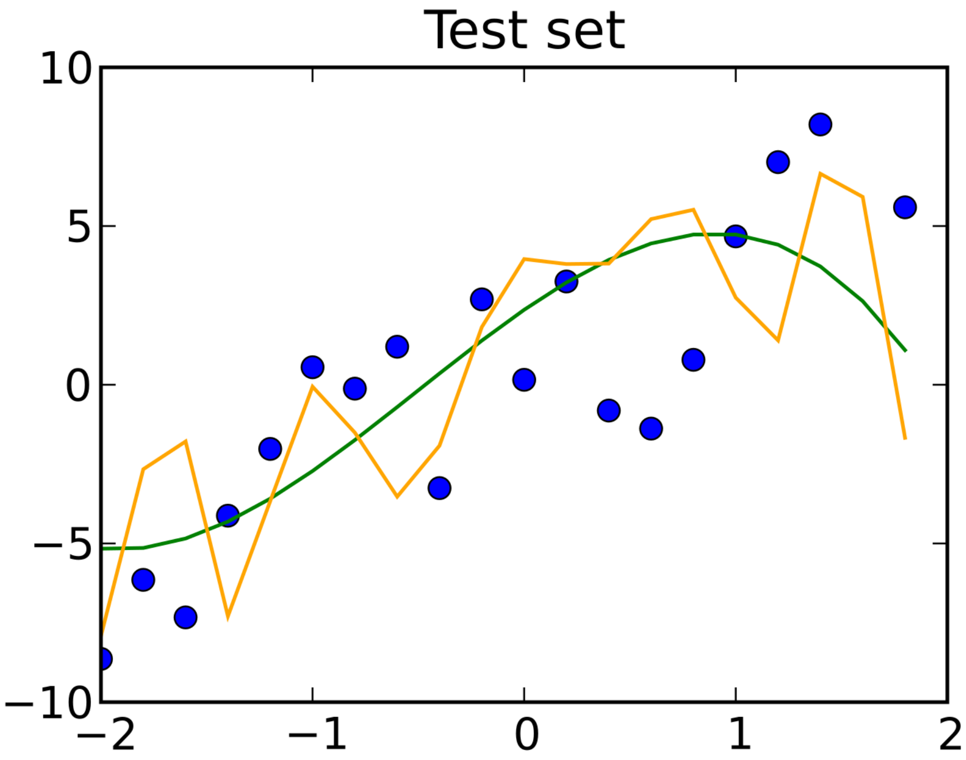
1) Model is too complex, contains too many parameters

2) Strong outliers in the data

3) Too small training dataset

Overfitting in regression





MSE = 4

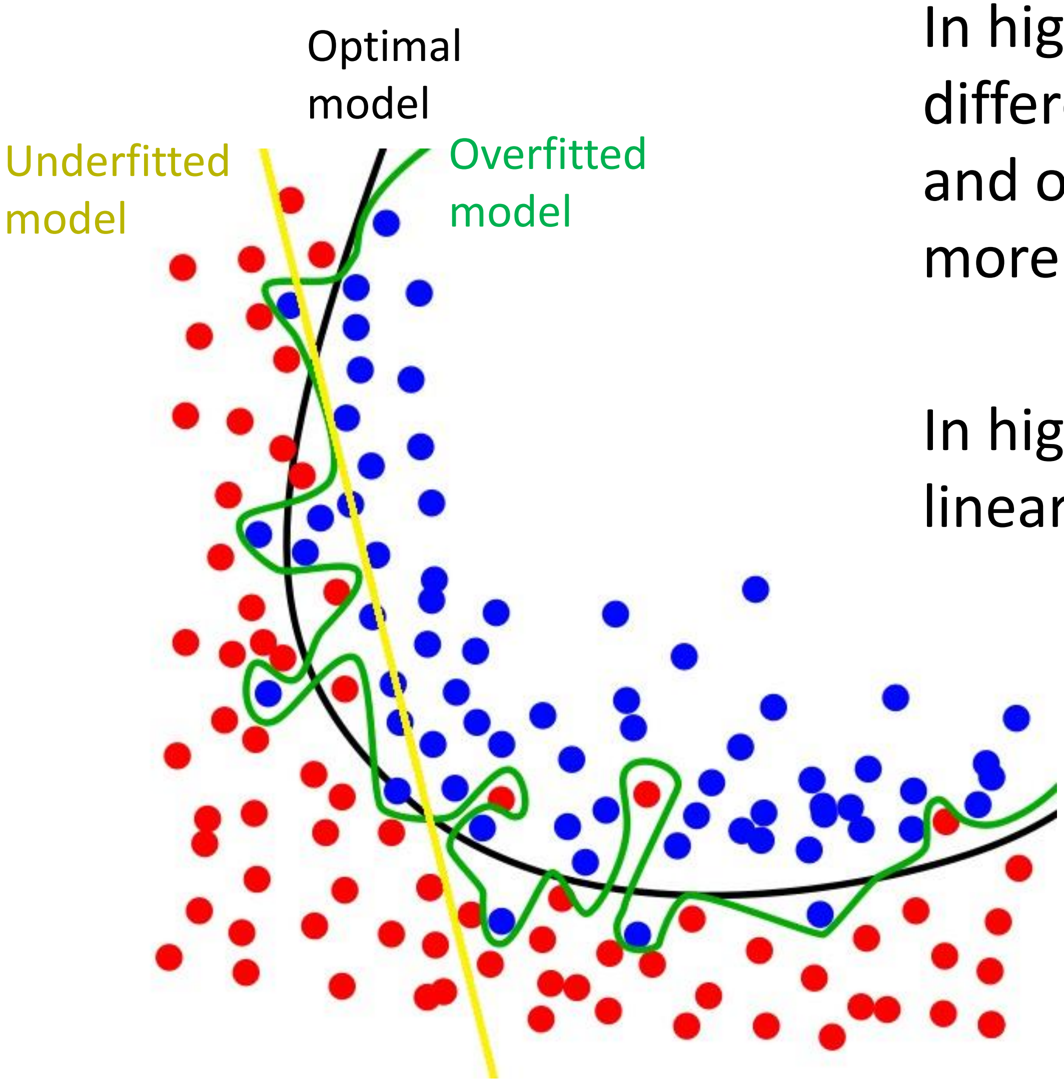
MSE = 9

MSE = 15

MSE = 13

Green model is better!

Overfitting in classification



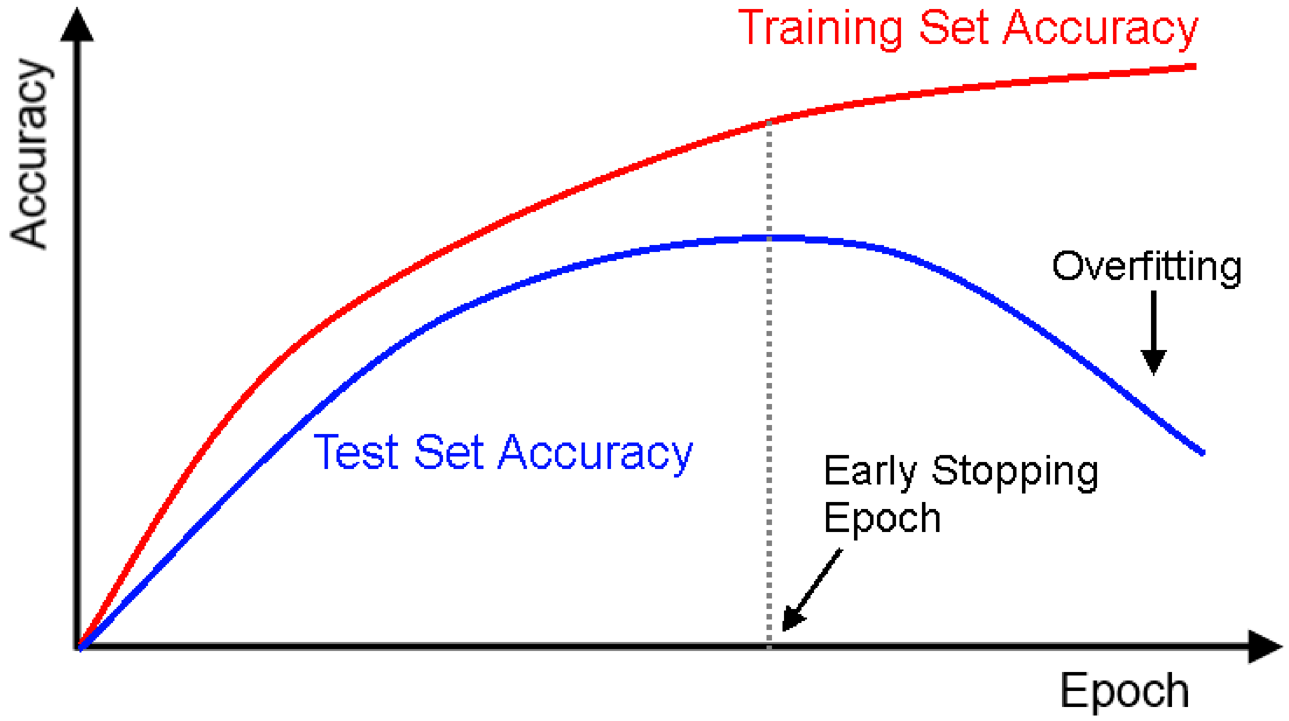
In higher dimensions the difference between underfitted and overfitted models become more important

In high dimensions even the linear model can overfit!

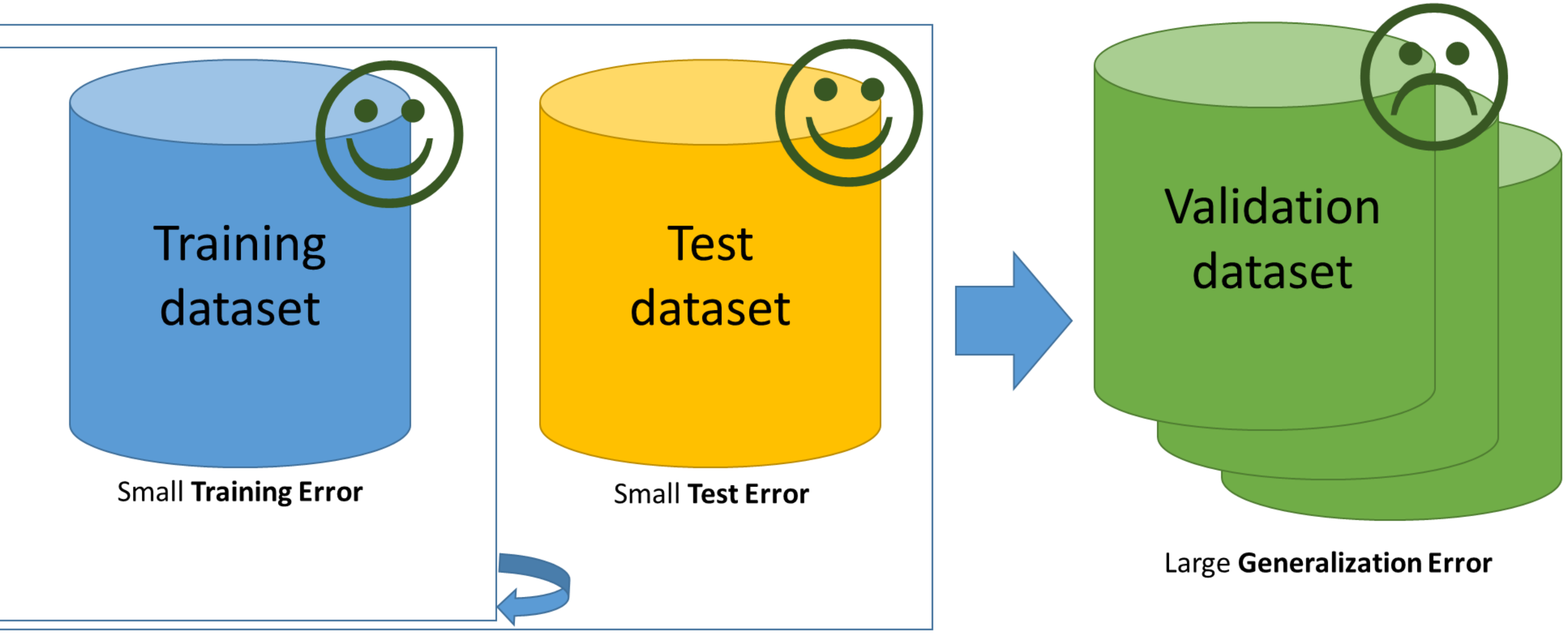
https://ezako.com/fr/les-concepts-doverfitting-et-underfitting-en-machine-learning/

Overfitting in neural networks

Learning curve



Lack of generalization



Causes for bad generalization:

Spurious correlations

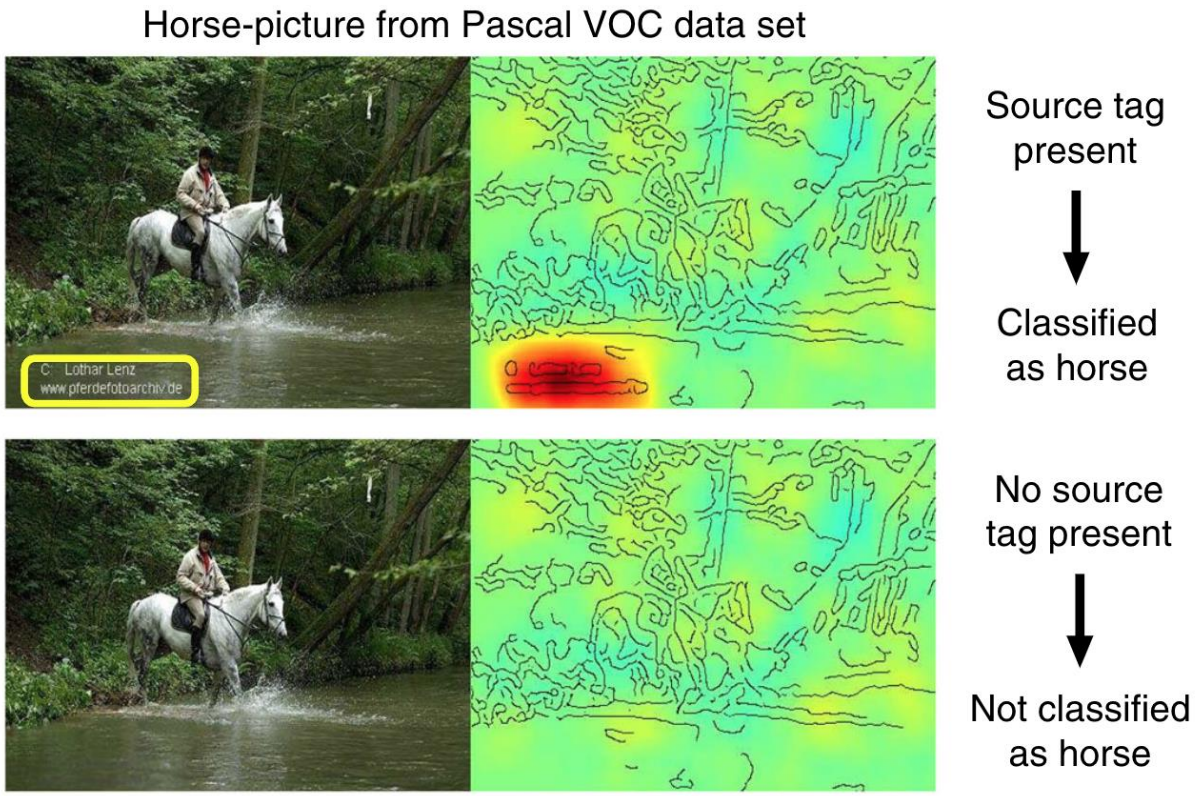
Unrepresentative data

Train test leakage

Data or concept drift

“Clever Hans” effect in supervised machine learning

https://en.wikipedia.org/wiki/Clever_Hans

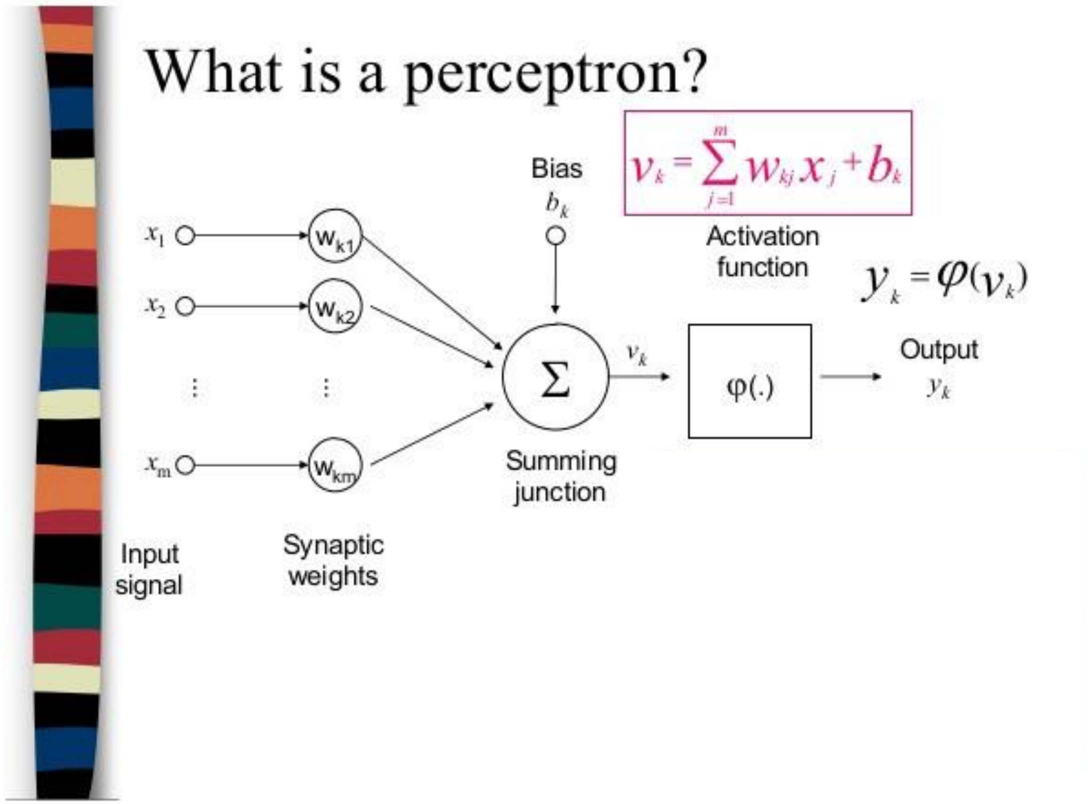


https://www.nature.com/articles/s41467-019-08987-4

Supervised learning:
From linear regression to deep learning

Linear regression and a simple perceptron (formal neuron)

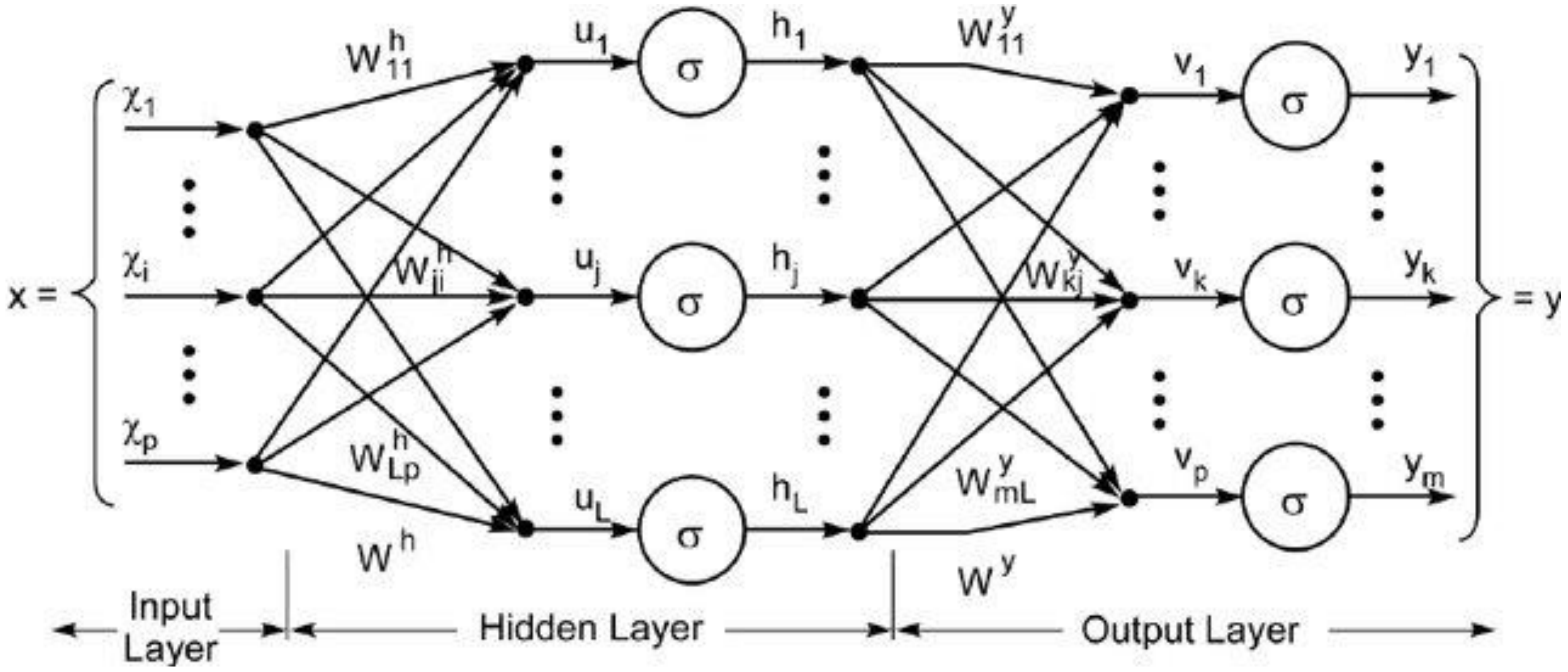
- Invented by Frank Rosenblatt in 1950s
- Elementary unit of any complex and deep neural network today



If φ(x) = x, then it is simple linear regression model

If φ(x) is a step-wise or sigmoidal function then it is a
binary classifier just as logistic regression (even though they are trained with different algorithms!)

Multilayered perceptron



https://ailephant.com/glossary/multilayer-perceptron/

Supervised learning:
Classification models

Zoo of supervised machine learning models

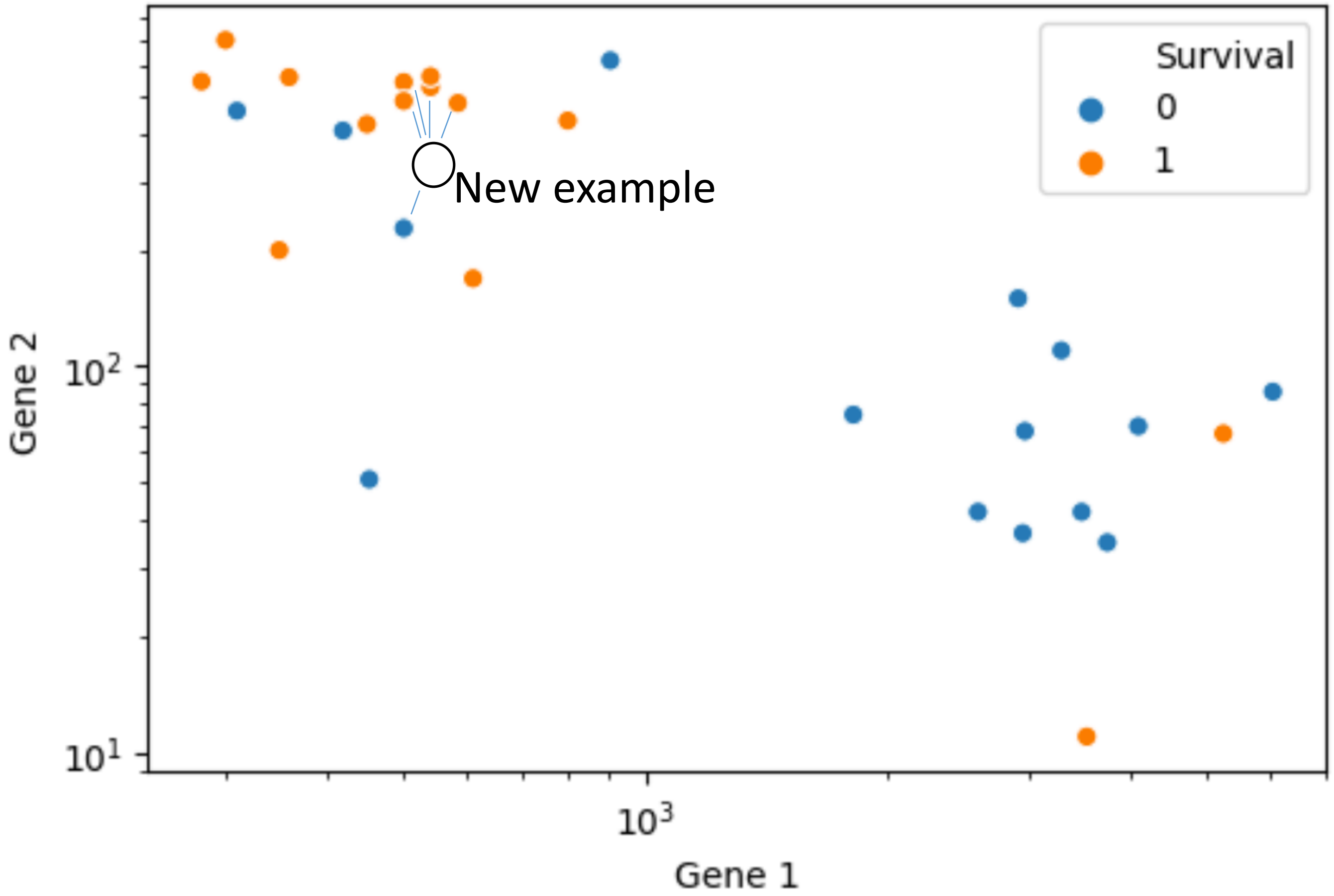
- k-Nearest Neighbour classifier
- Random forests
- Discriminant analysis
  - Fisher Discriminant Analysis
  - Support Vector Machines
- Probabilistic methods based on modeling joint probability distribution:

  Naïve Bayes

  Bayesian networks

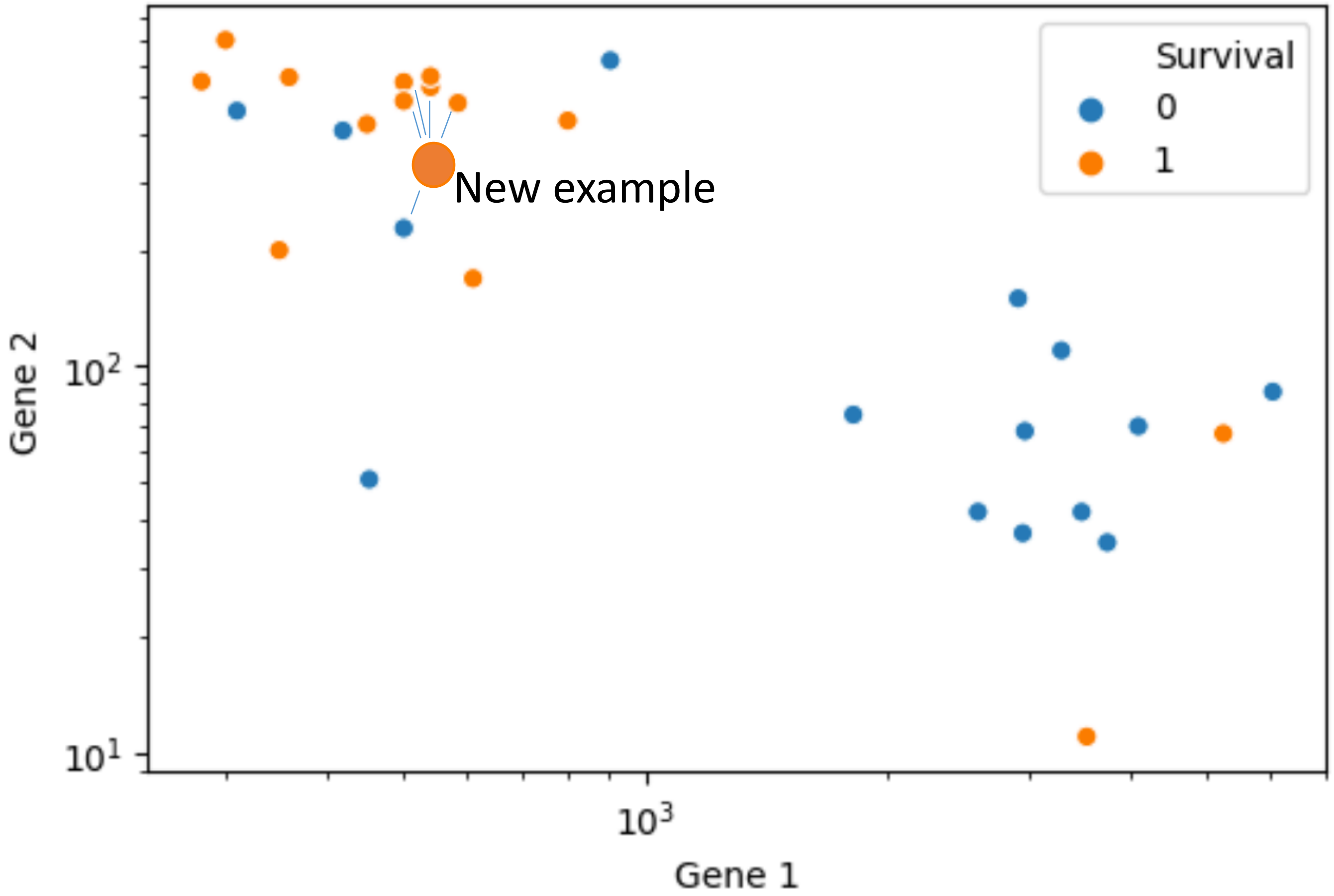
Zoo of supervised machine learning models

- k-Nearest Neighbour classifier



Zoo of supervised machine learning models

- k-Nearest Neighbour classifier



Zoo of supervised machine learning models ☺

- k-Nearest Neighbour classifier **– simple to implement, one parameter!**
- Random forests **– works out of the box, generalize well!**
- Discriminant analysis
  - Fisher Discriminant Analysis
  - Support Vector Machines **- works with relatively few samples!**
- Probabilistic methods based on modeling joint probability distribution:

  Naïve Bayes **– no overfitting!**

  Bayesian networks **– creates generative data model!**

Zoo of supervised machine learning models ☹

- k-Nearest Neighbour classifier **– poor in performance!**
- Random forests **– parameters are too complex!**
- Discriminant analysis:
  - Fisher Discriminant Analysis
  - Support Vector Machines **- does not scale well!**
- Probabilistic methods based on modeling joint probability distribution:

  Naïve Bayes **– might create huge bias!**

  Bayesian networks **– requires a lot of data!**

Unsupervised learning:
What is it? Why it is needed?

Unsupervised learning

“**Unsupervised learning** (**UL**) is a type of algorithm that learns patterns from untagged data.” (c) Wikipedia

“Learning of intrinsic connections and interdependencies between features and objects”

Learning of a human being is essentially unsupervised (self-supervised) and observational

Unsupervised learning

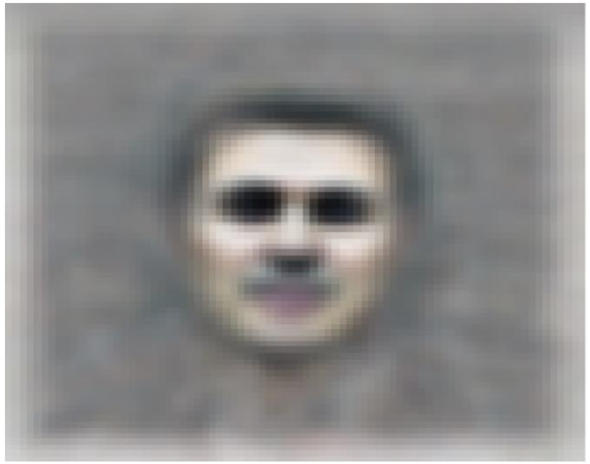
Two main tasks:

1) Clustering : **de-novo labeling** of the data points, based on their mutual similarity

2) Dimensionality reduction: presenting high-dimensional data point cloud in low dimensional space, such that some important features are preserved

Google cat: example of massive unsupervised learning

“Google cat”



Le et al. Building High-level Features Using Large Scale Unsupervised Learning. ICML-2012

Unsupervised learning: Distance matrix and neighbourhood graph

Distance matrix

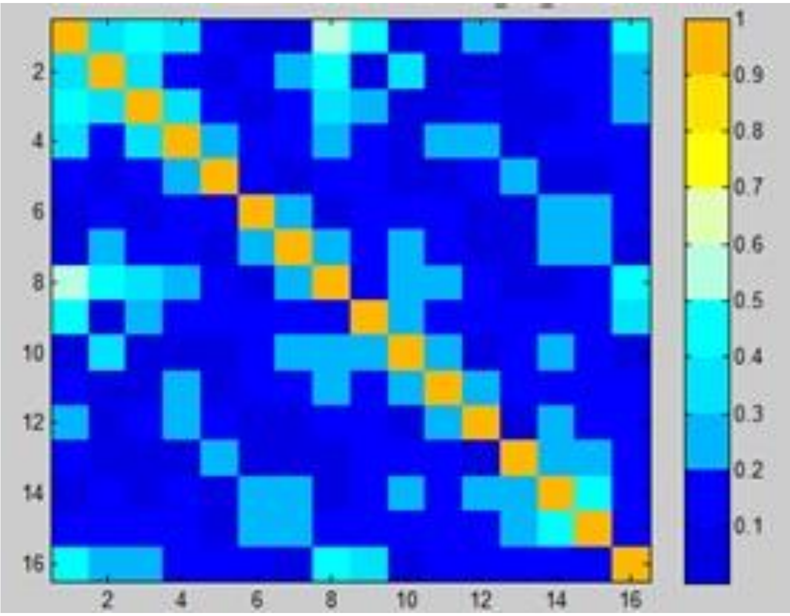
- Non-negative, symmetric

- Various distance functions: Euclidean, correlation-based, angular, Manhattan, etc.

- Convenient for searching close and distant neighbours

- Inconvenient to store cause the number of elements grows quadratically:

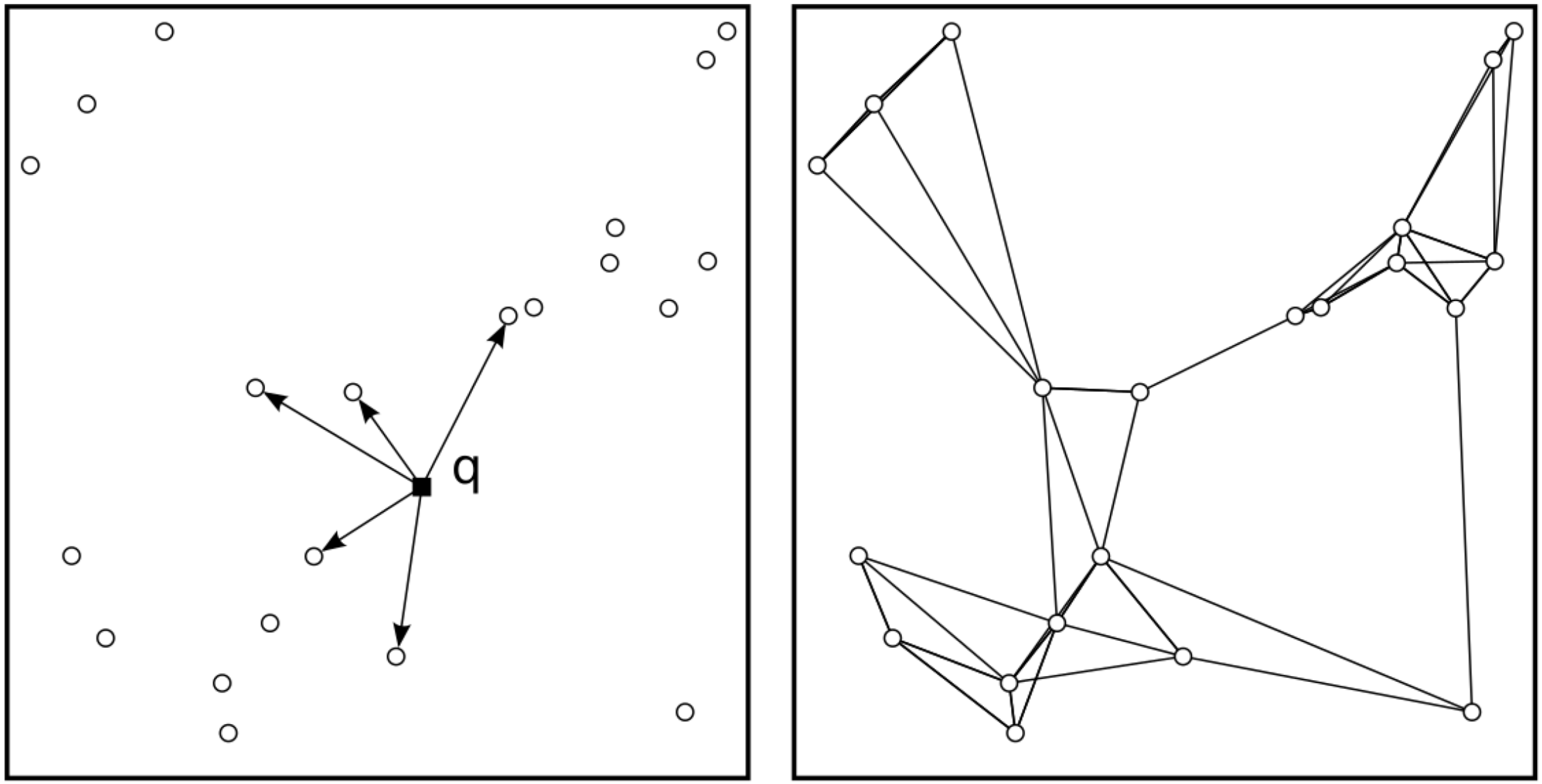
100000 * 100000 * 2 **bytes** (float16 **size**) = 20 Gb of **RAM**



	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0

k Nearest Neighbor (kNN) graph

k-nearest neighbors, $k = 5$ $\qquad$ k nearest neighbors graph ($k = 3$)



Requires N*k integer numbers: for 100000 objects - **2-3 Mb** of memory!

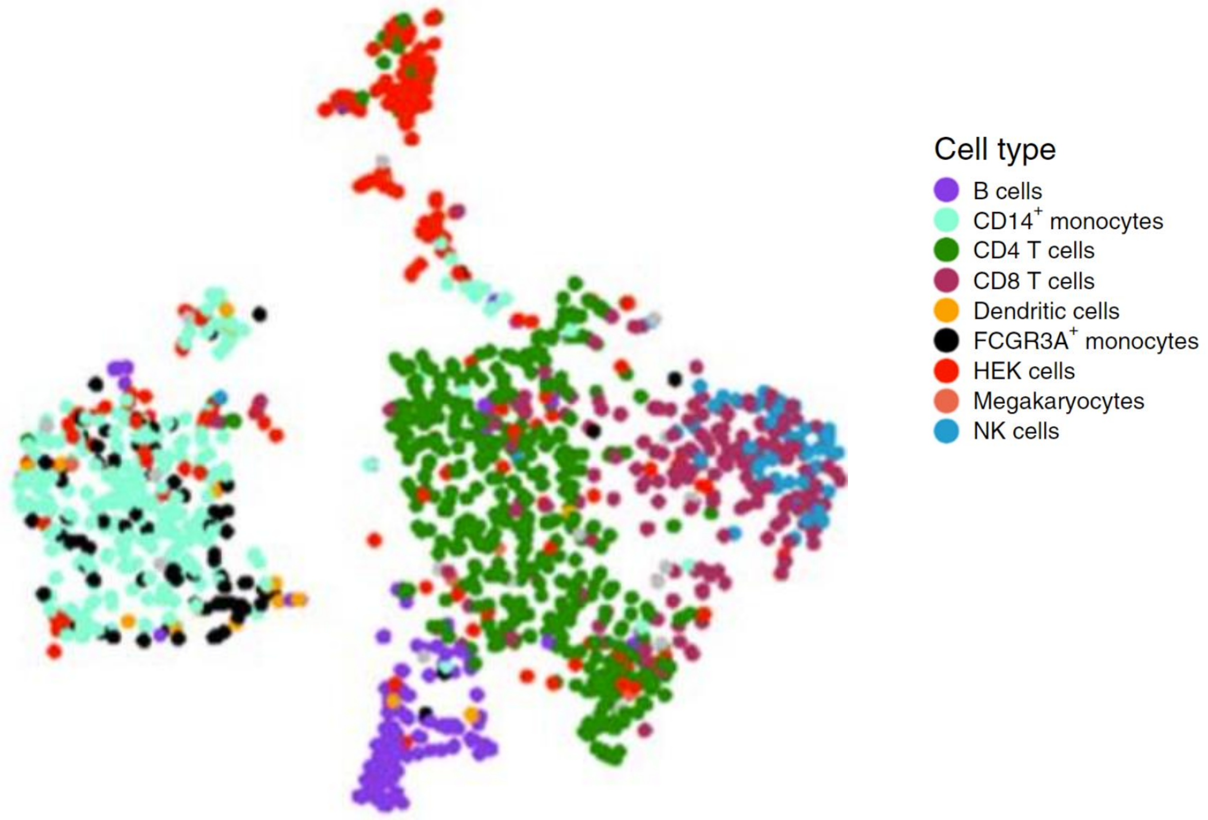
Unsupervised learning:
Clustering methods

Clustering problem in machine learning

The goal of clustering is to separate a finite, unlabeled data set into a finite and discrete set of “natural”, “hidden” data structures

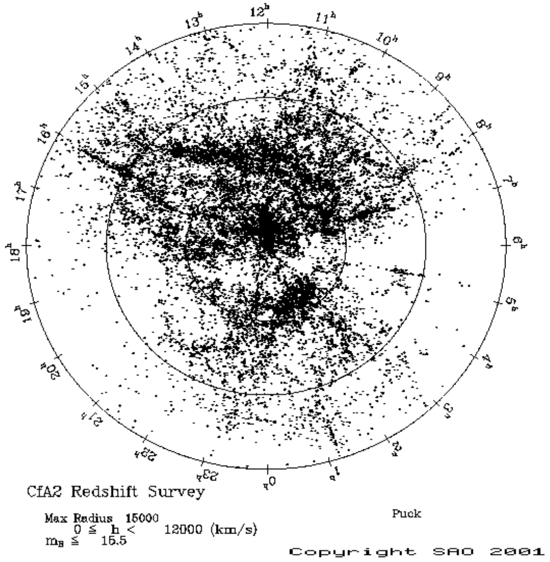
Distinguish *classes* and *clusters*!!!

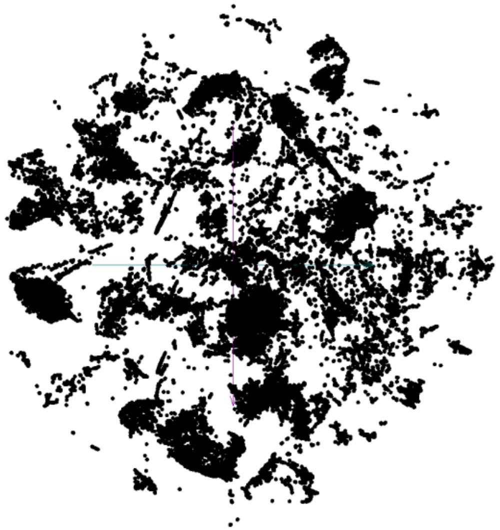
- **Class** = set of data points with the same pre-defined label
- **Cluster** = result of solving a clustering problem



From Mereu et al, Nature Biotech , 2020

Real-life datasets can have complex clusters



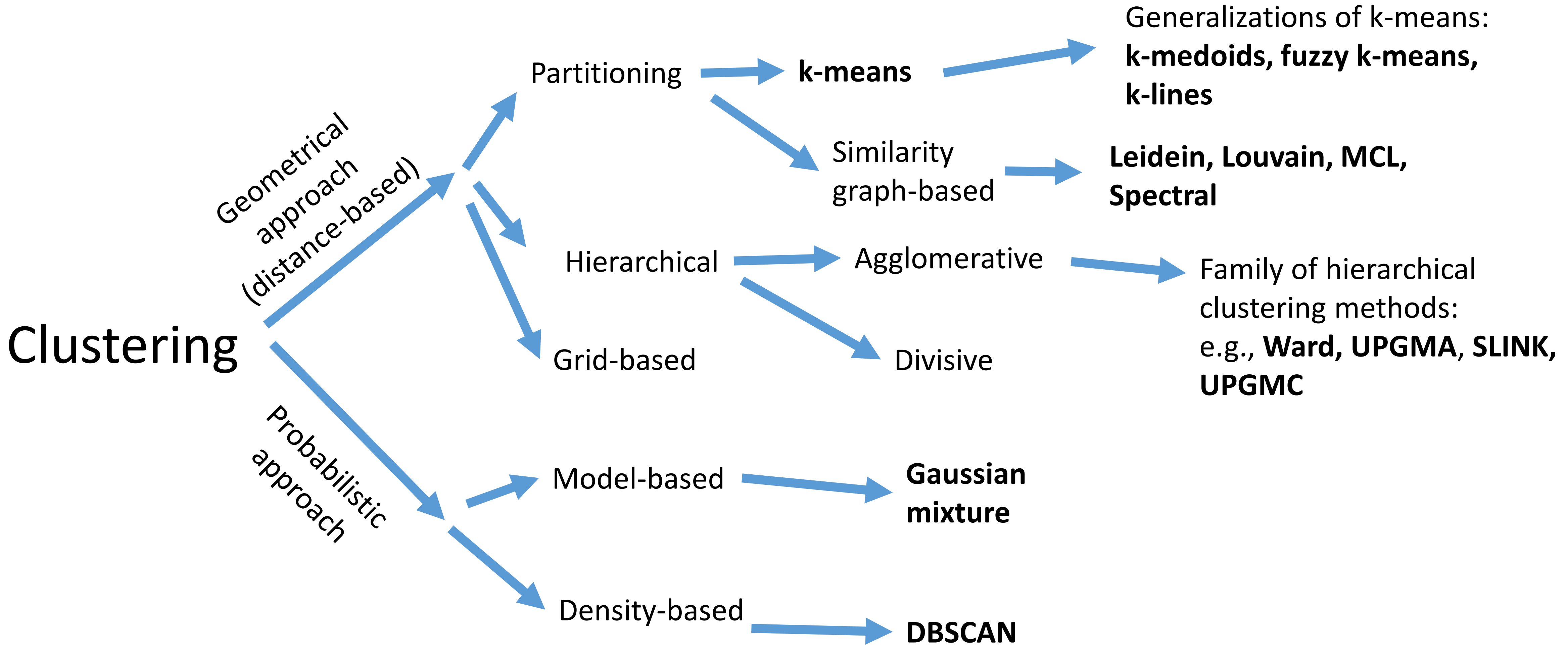


https://amp.pharm.mssm.edu/archs4/data.html

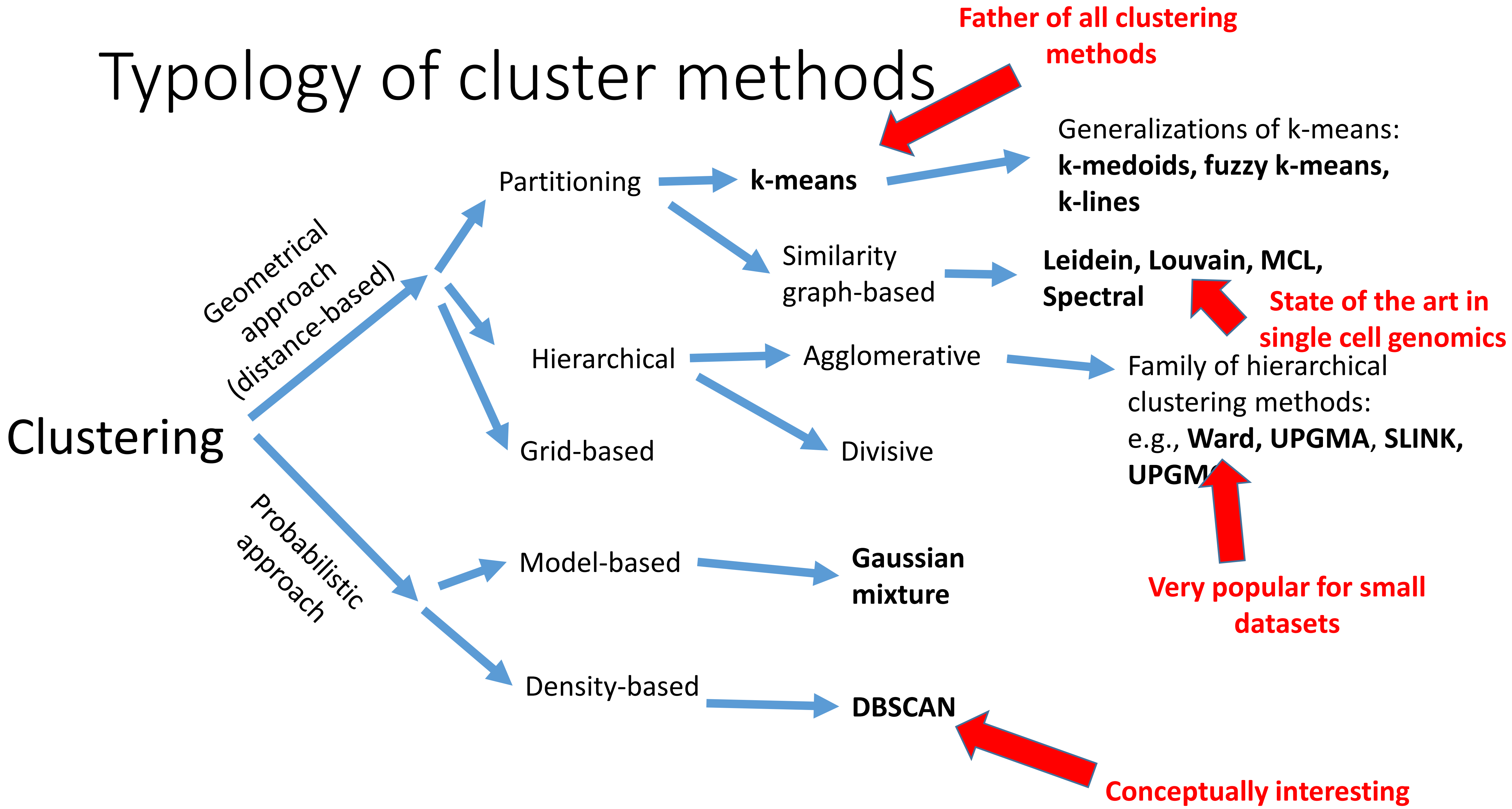
ANY clustering method requires specifying the number of clusters as a parameter

- Sometimes it is done explicitly
- Sometimes it is done through some kind of ‘scale’ or ‘resolution’ parameter

Typology of cluster methods



Typology of cluster methods

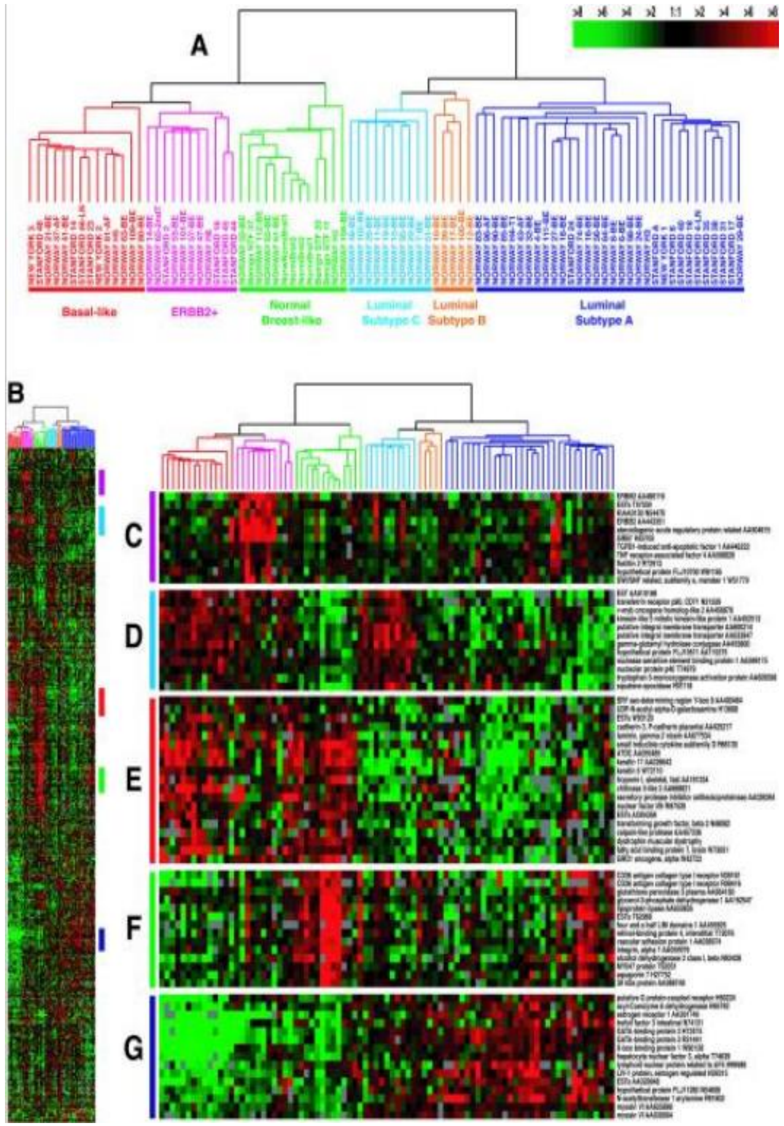


Unsupervised learning:
Some clustering examples

Hierarchical clustering for studying cancer

Dendrogram

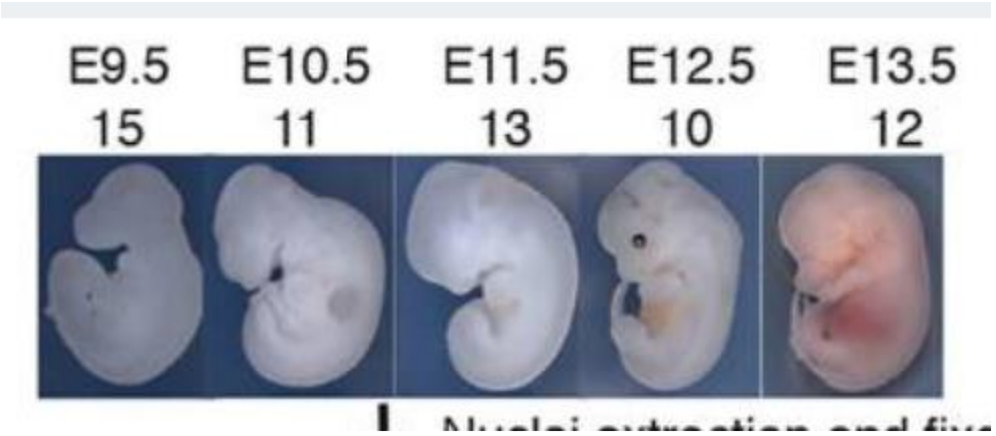
Heatmap



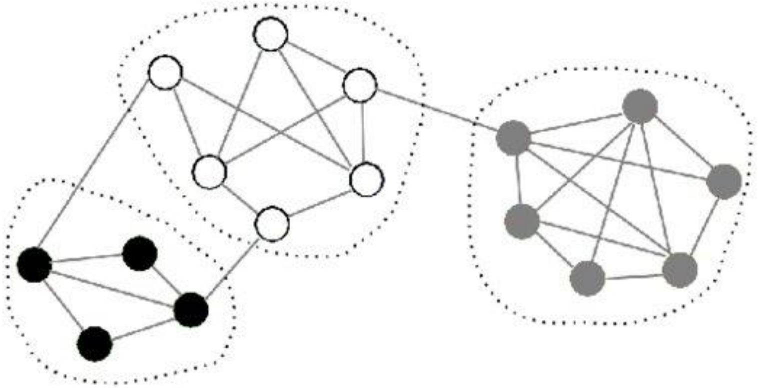
Sorlie, PNAS 2001

Clusters and visualizes the data!

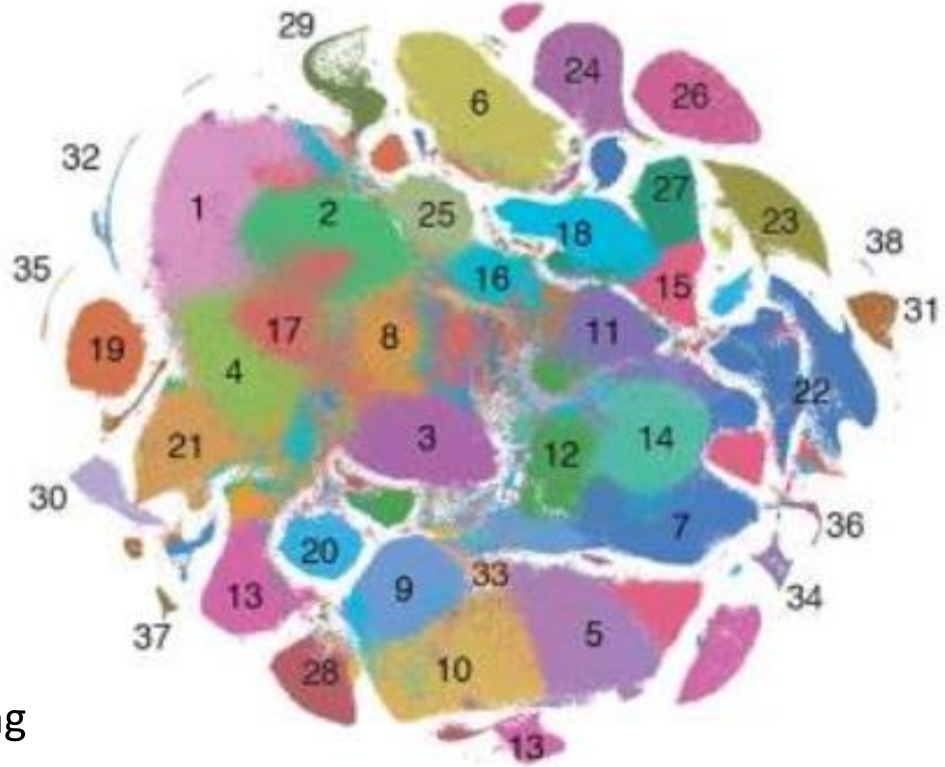
Graph-based clustering became new killer application in life sciences



2 million data points – individual cells from mouse embryo



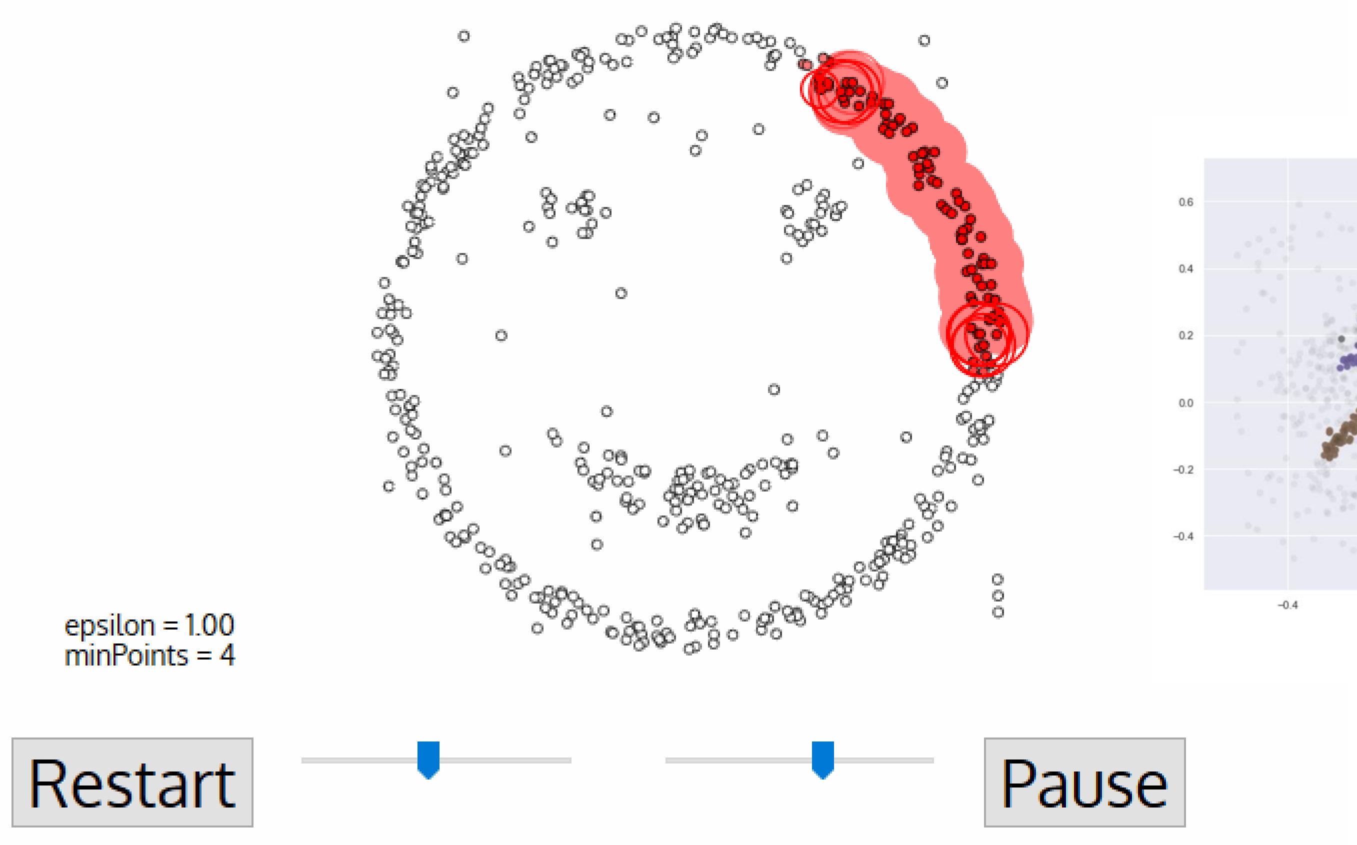
Finding communities in neighbourhood graphs



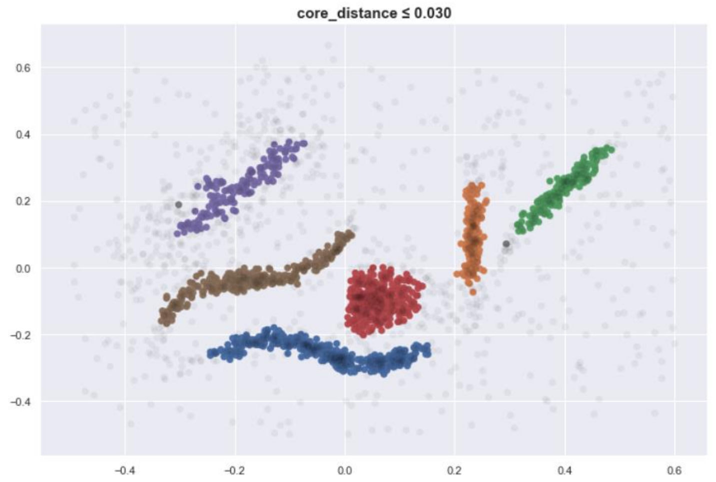
Louvain clustering

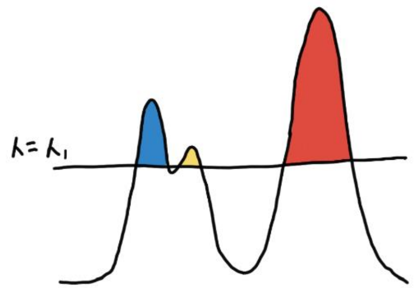
(from Cao et al, Nature, 2019)

Density-based clustering: cluster as an area of density concentration

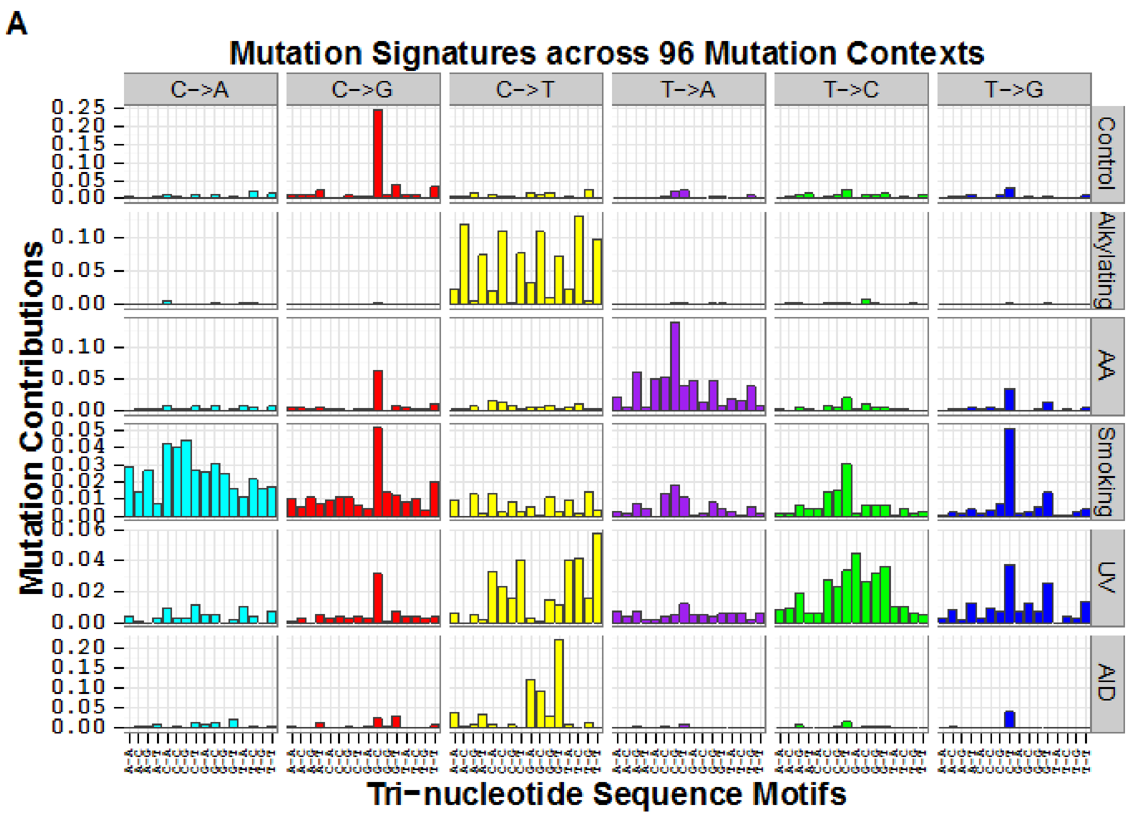


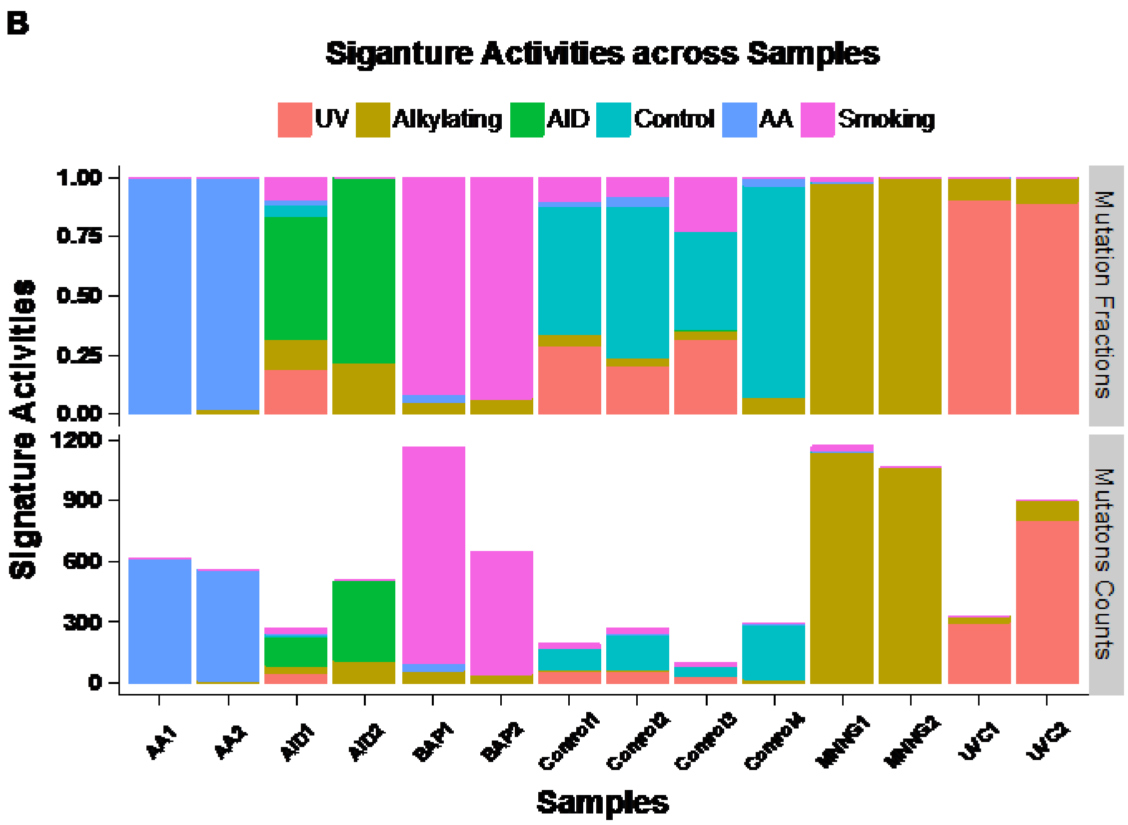
DBSCAN





Non-negative matrix factorization (NMF): cluster as a factor, having activity





https://software.broadinstitute.org/cancer/cga/msp

Unsupervised learning:
What is dimensionality reduction?

Dimensionality reduction formula

Encoder or ***Projector*** operator

$$\boxed{R^p \rightarrow R^m,\ m \ll p}$$

Data space

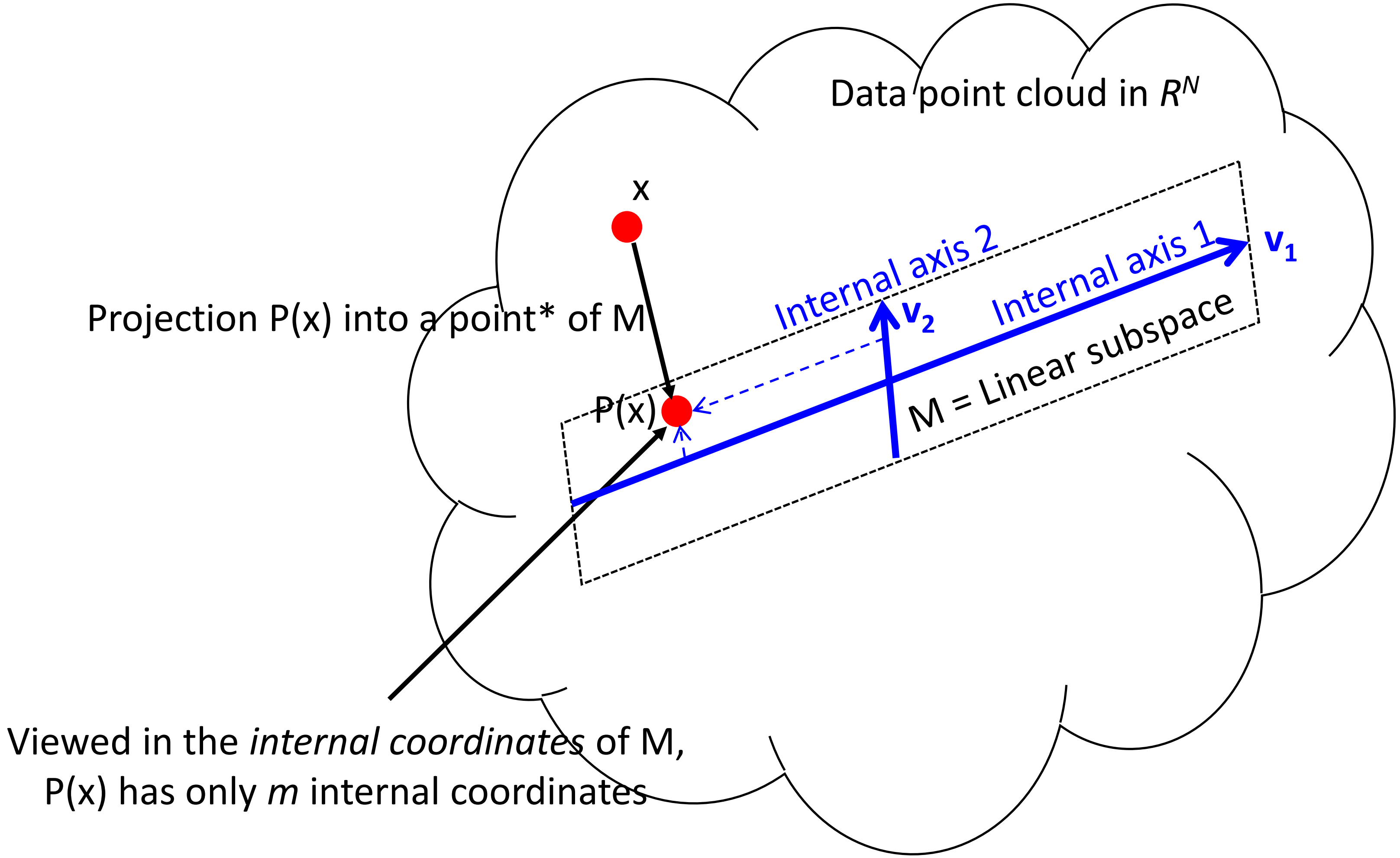
Some space*,**,***

*can have complex geometry and topology

**does not have to be part of R^p

***simplest case: m-dimensional linear subspace in R^p

Simplest geometrical image



*for example, into the closest point, P(x) = arg min || y - x ||

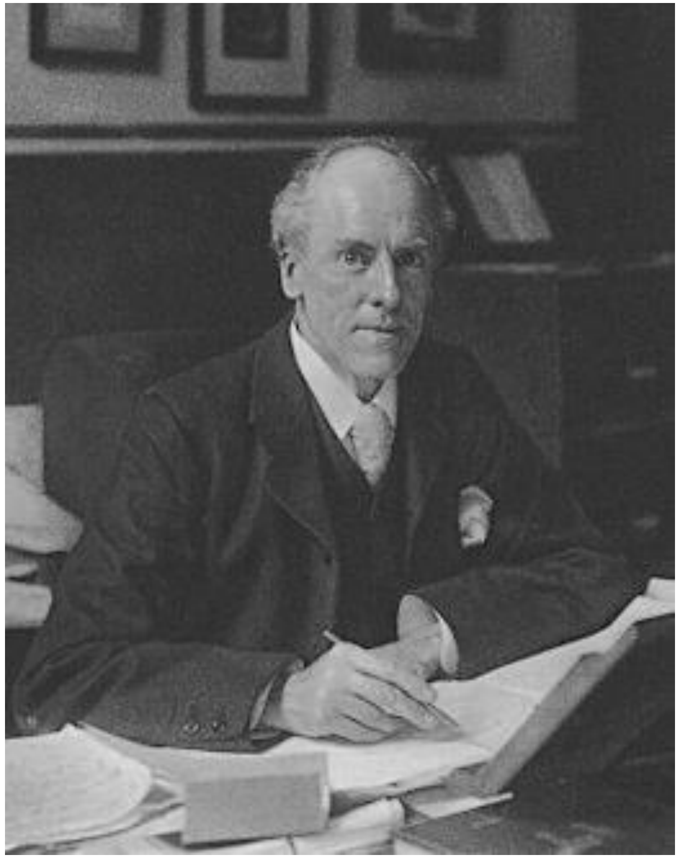
Why do we need to reduce dimension?

- **Converting wide data to the classical case N>>p**
- Improving signal/noise ratio for many other supervised or unsupervised methods
- Fighting with the curse of dimensionality
- Computational and memory tractability of data mining methods
- Visualizing the data
- Feature construction

Unsupervised learning:
What is Principal Component Analysis (PCA)?

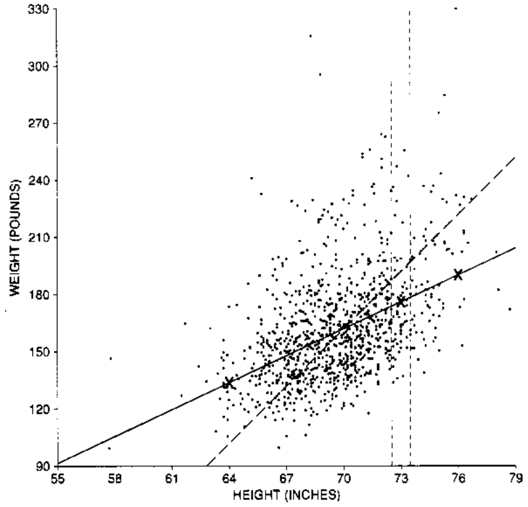
Principal Component Analysis (PCA):

(really) central method for unsupervised machine learning which is 120 years old!



Karl Pearson, 1857 –1936

Pearson (1901): problem of choice of dependent and independent variables



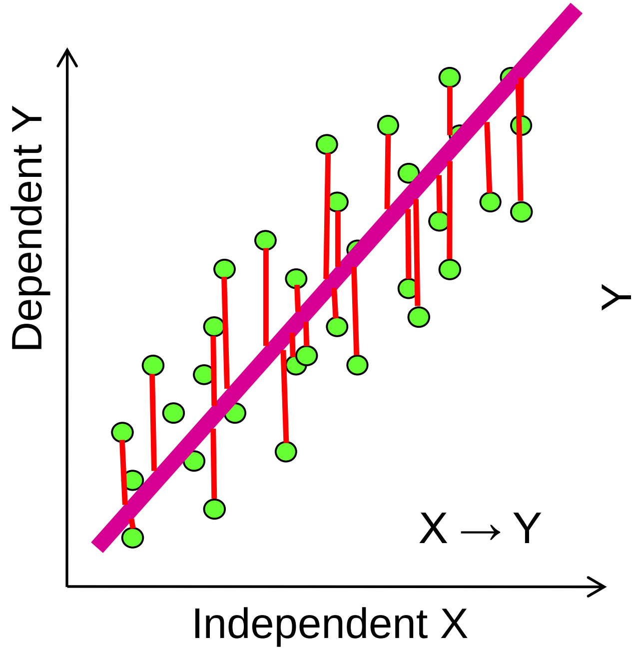
Weight = $a_0 + a_1$ Height

Height = $b_0 + b_1$ Weight

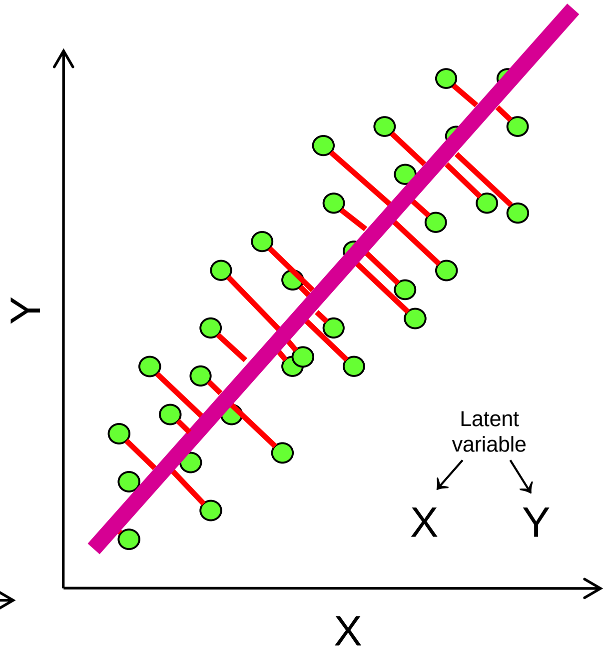
$a_1 \neq 1/b_1$!!!

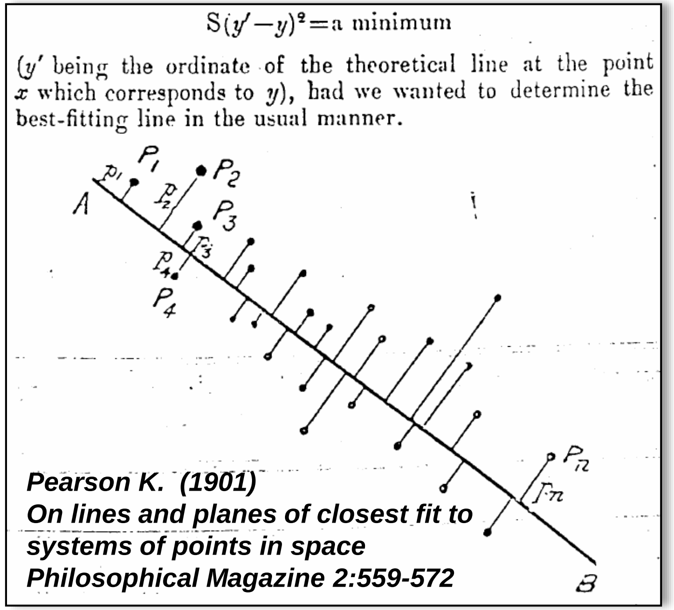
$$\sum_{i=1}^{m} \| \quad \|^2 \to \min$$

Linear regression



Principal component (best fit line)



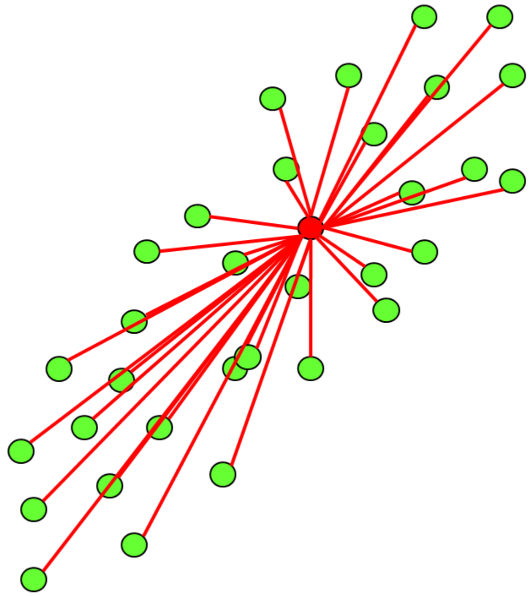

S$(y'-y)^2$=a minimum

(y' being the ordinate of the theoretical line at the point x which corresponds to y), had we wanted to determine the best-fitting line in the usual manner.

Pearson K. (1901)
On lines and planes of closest fit to systems of points in space
Philosophical Magazine 2:559-572

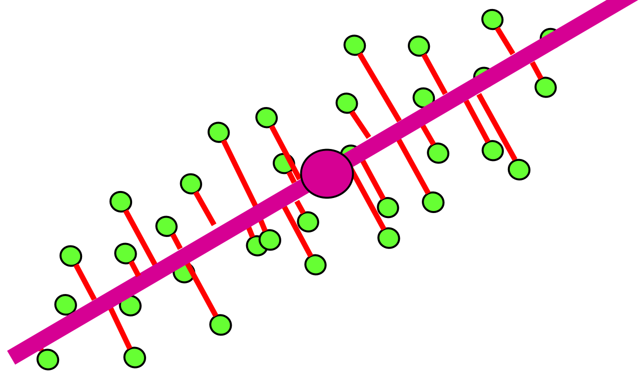
Principal line and principal plane

Mean point

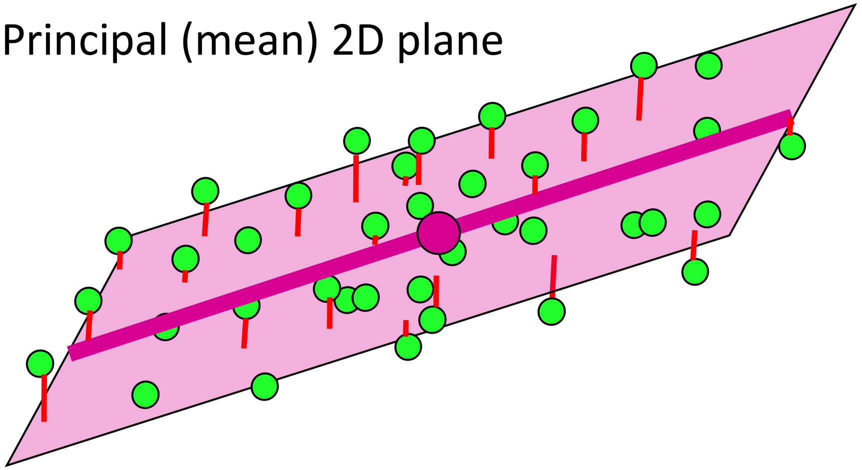


$$\sum_{i=1}^{m} \| \quad \|^2 \to \min$$

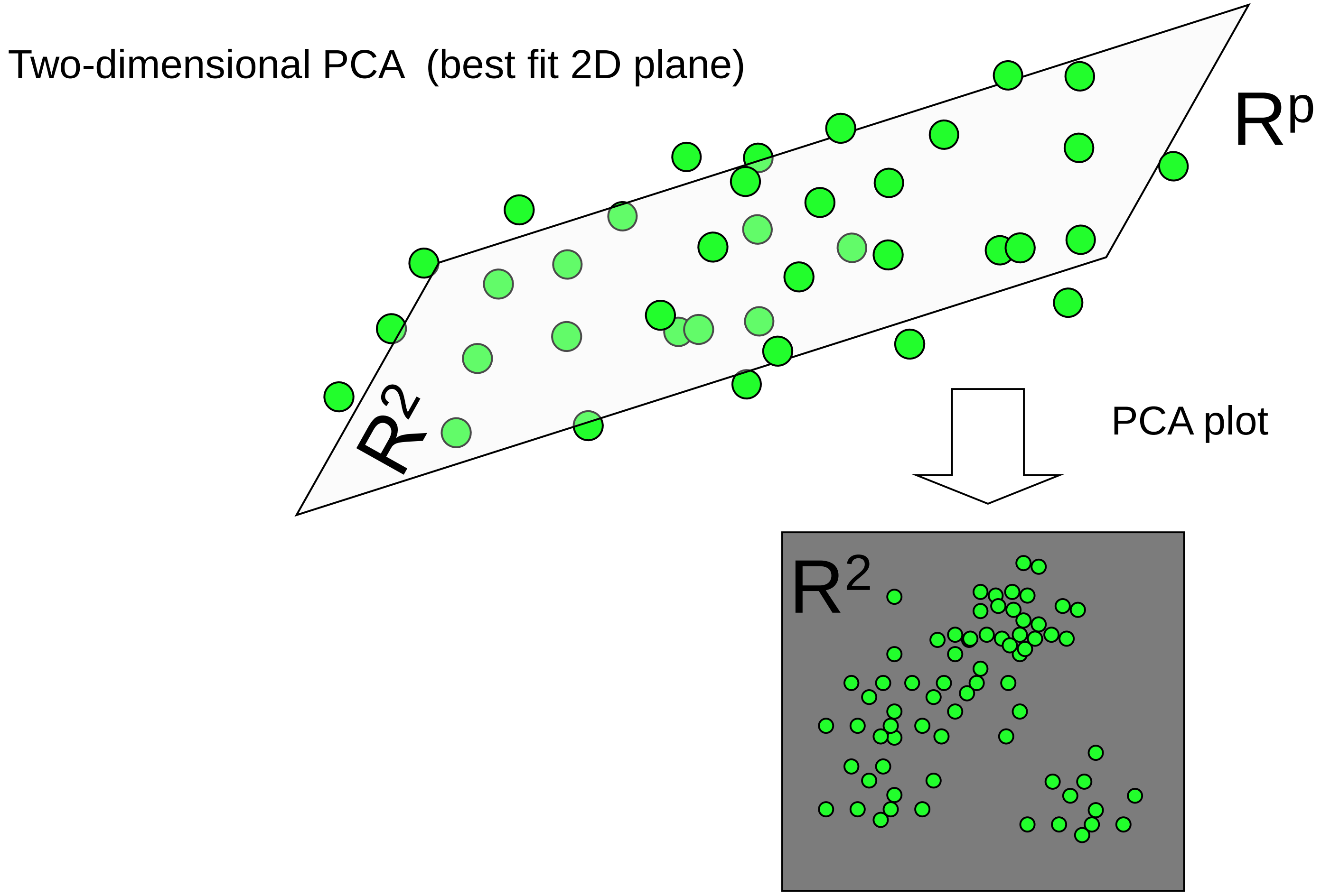
Principal (mean) line



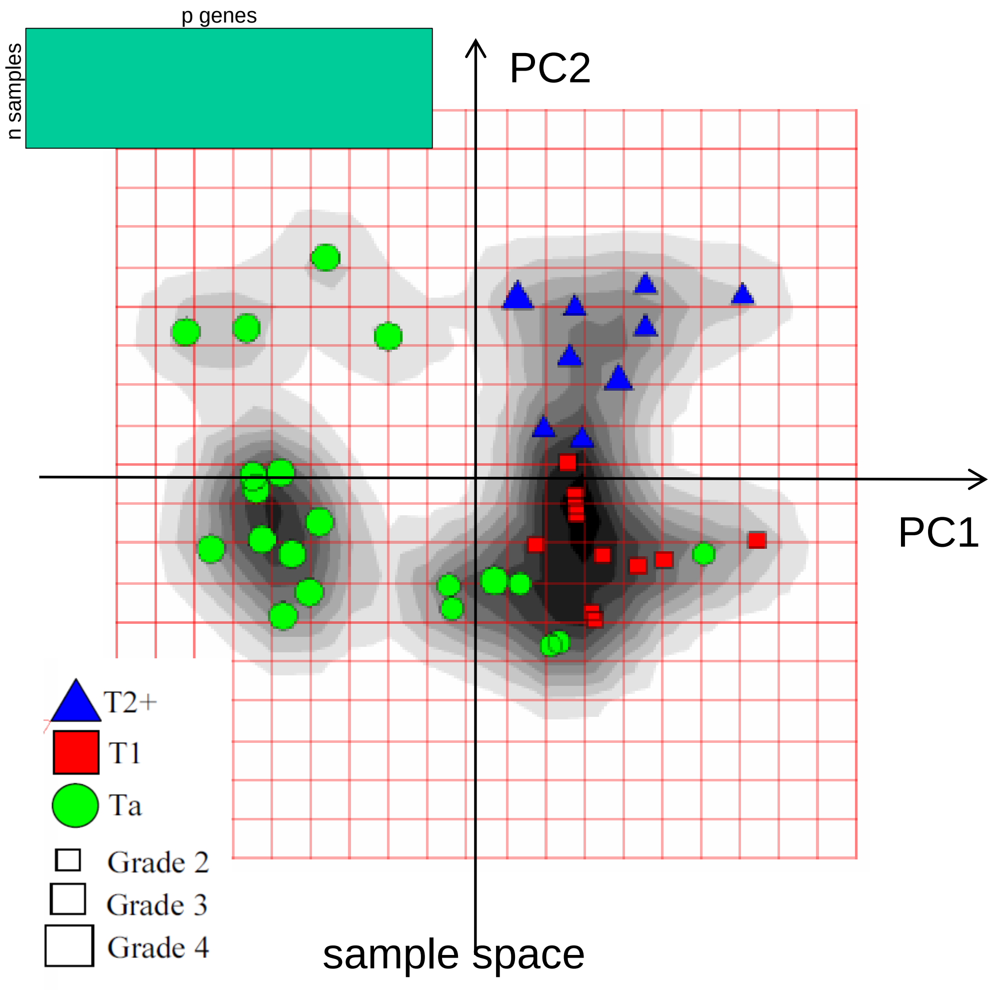
Principal (mean) 2D plane

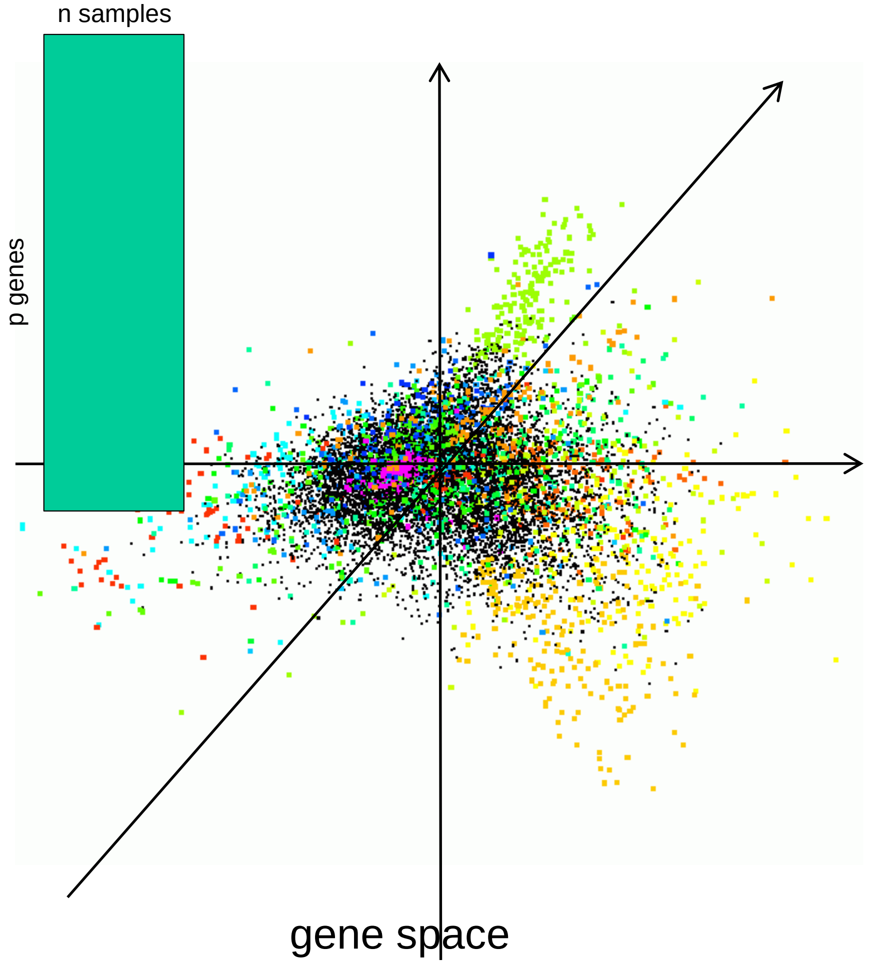


PCA as data visualization method, based on dimension reduction



PCA plots of transcriptomic datasets





Classification, diagnosis, prognosis

Identification of molecular mechanisms, Interpretation

Unsupervised learning:
What is matrix factorization?

Low rank matrix factorization $X = UV$

$$N \left\{ \overbrace{\begin{bmatrix} & & \\ & X & \\ & & \end{bmatrix}}^{p} \right. \approx N \left\{ \overbrace{\begin{bmatrix} \\ U \\ \\ \end{bmatrix}}^{m} \right. \left. \overbrace{\begin{bmatrix} & V & \end{bmatrix}}^{p} \right\} m$$

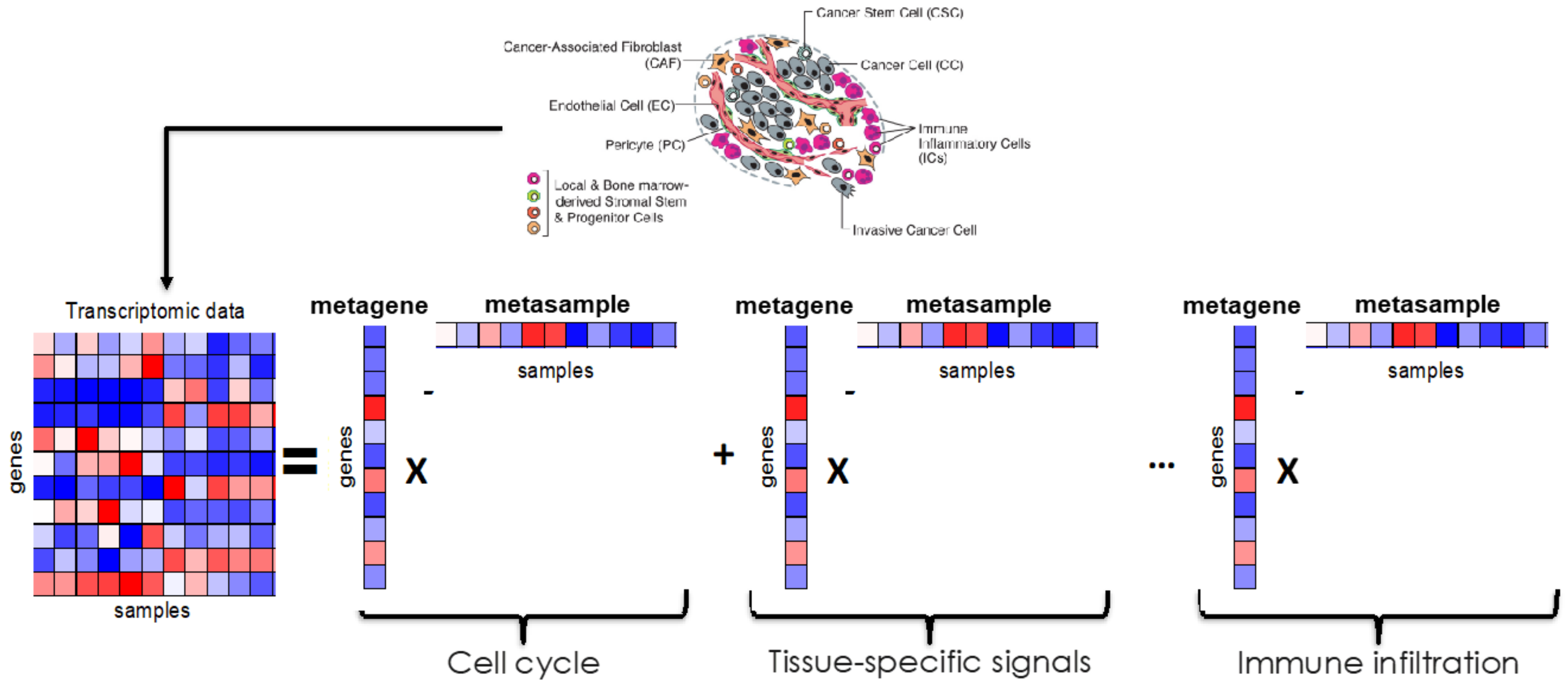
Each column in *U* and row in *V* (together) are called a *component*

Elements of *U* can be used for further analysis as a new data matrix

Elements of *V* can be used for *explaining components*

Low rank matrix factorization $X = UV$

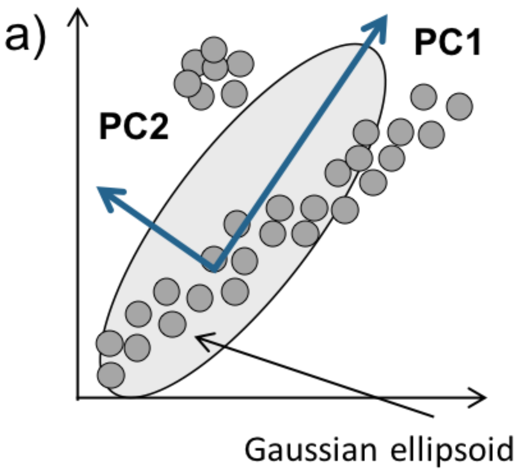
Moving from thousands of genes to few biological factors through MF



Brunet JP. et al., PNAS (2004).
Stein-O'Brien, G.L. et al. Trends in Genetics (2018).

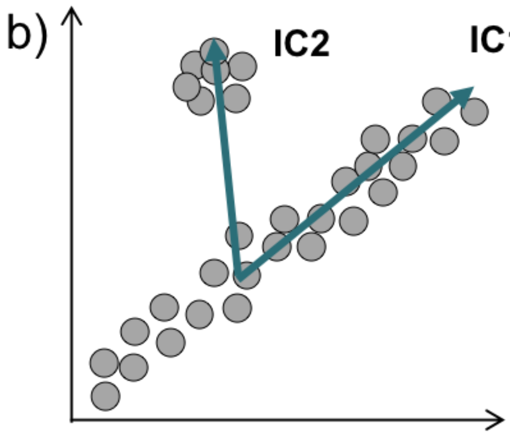
Three most popular matrix factorization methods

Principal Component Analysis	Independent Component Analysis	Non-negative Matrix Factorization
PCA	ICA	NMF



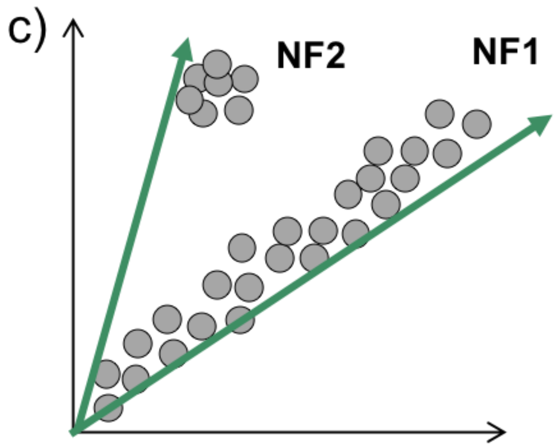
Pros: Deterministic, Fast

Contras: too ‘myopic’



Pros: elegant data model as superposition of independent latent factors

Cons: factors in practice have always negative part



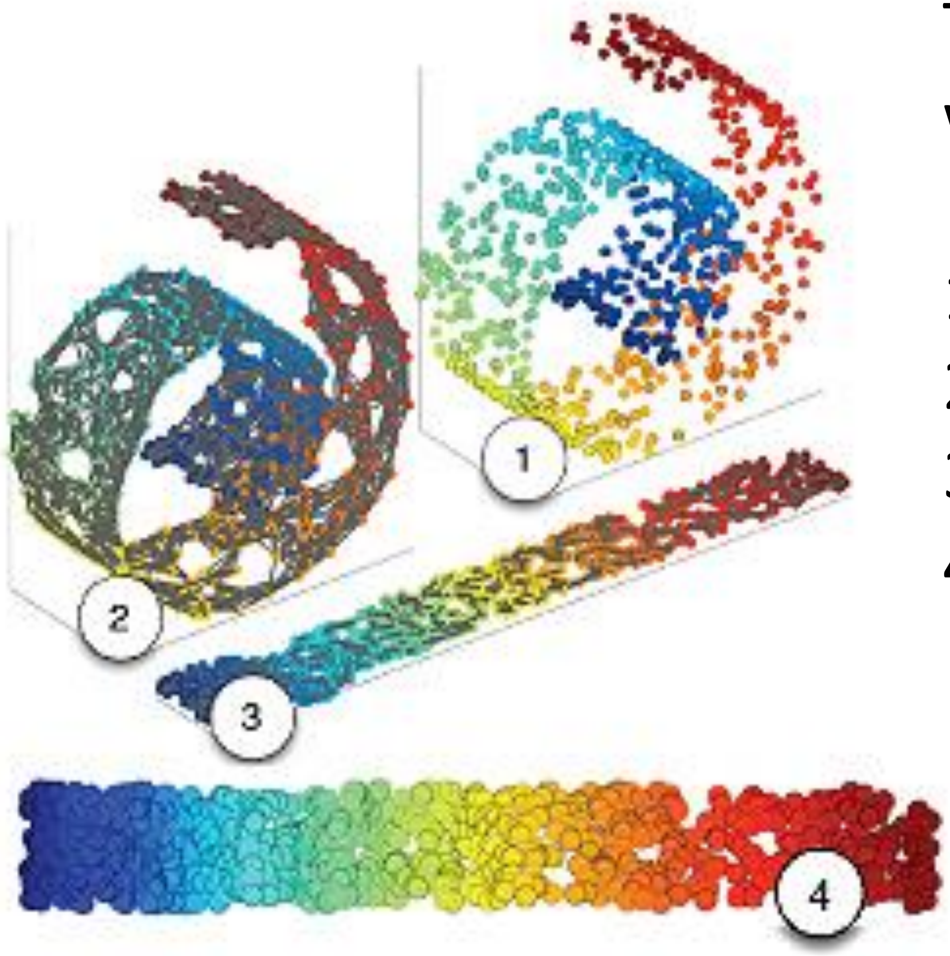
Pros: non-negativity matches biological reality

Cons: not a consistent method, correlation to average

Unsupervised learning:

Non-linear dimensionality reduction methods (aka manifold learning)

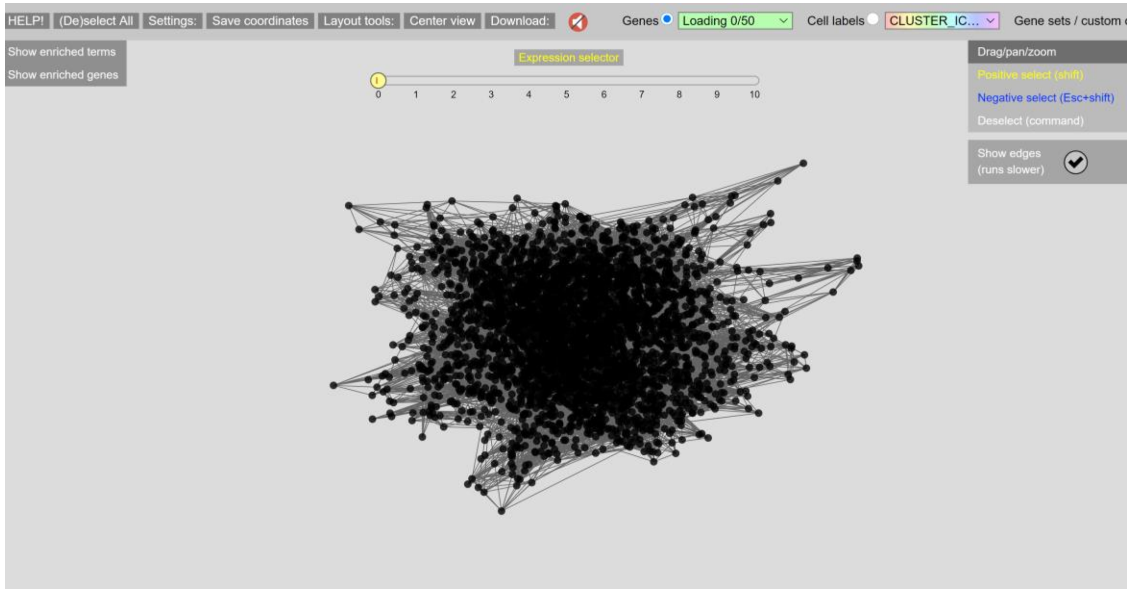
Manifold learning



Typical steps in learning and working with data manifold

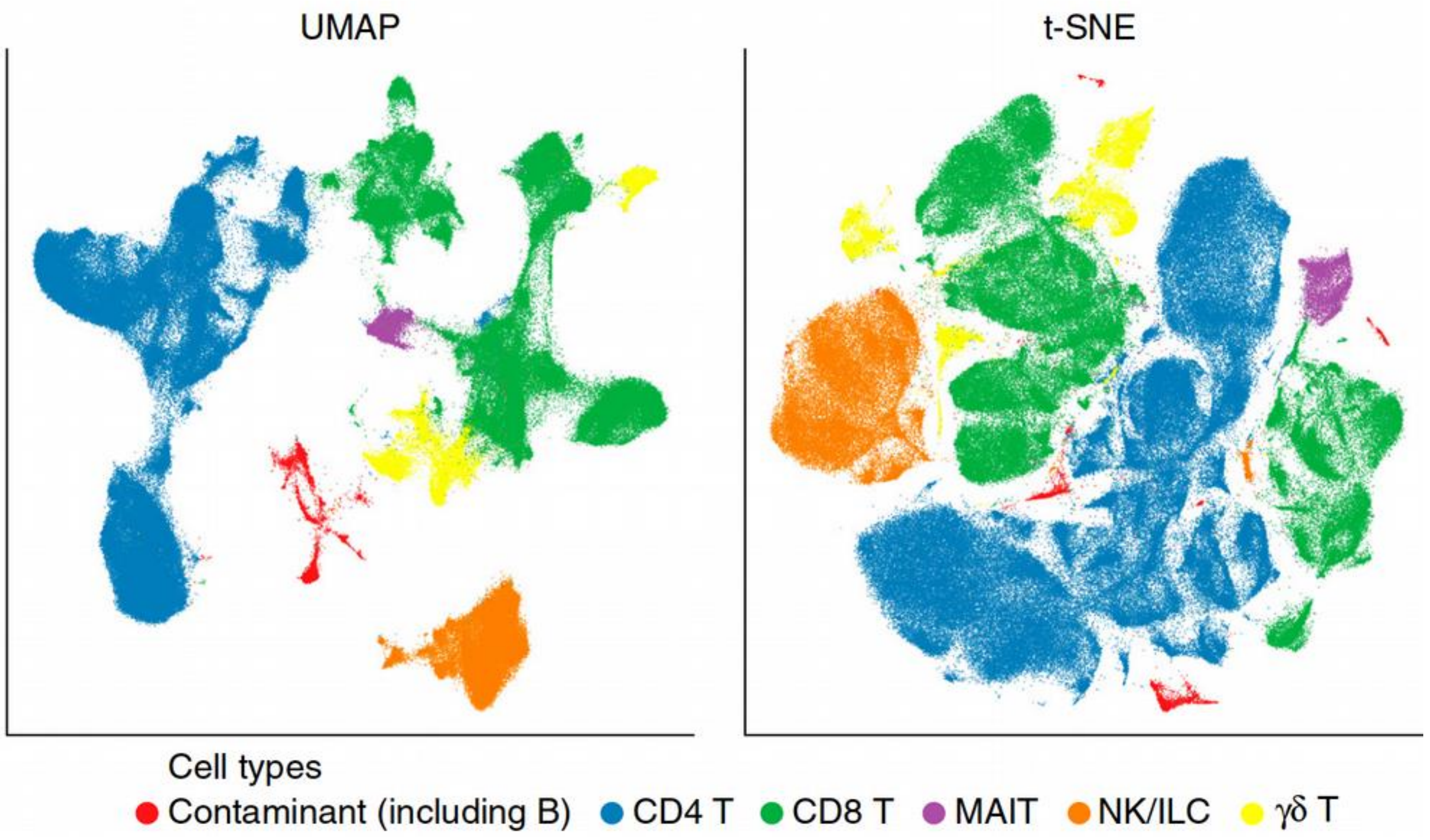
1) Data point cloud
2) Neigbourhood graph
3) Unfolding (layouting) the graph in 2D
4) Presenting the data points projections

Example of applying graph layouting to reduce data dimensionality (here, simple kNN graph)



https://www.ihes.fr/~zinovyev/mosaic/SPRING/springViewer.html?datasets/CHLA9_nufp

T-SNE and UMAP: two killer applications in single cell field



Both are good in representing local relations

Differ in the exact way to construct the neighbourhood graph: e.g., UMAP tries to compensate for the effects of high-dimensional data

https://towardsdatascience.com/reduce-dimensions-for-single-cell-4224778a2d67

Comparing tSNE and UMAP

- UMAP better represents the global structure of the dataset
- UMAP is way faster than t-SNE
- UMAP is more stable to subsampling than t-SNE
- UMAP can work directly in very

high ambient dimensionalities ($>10^6$)

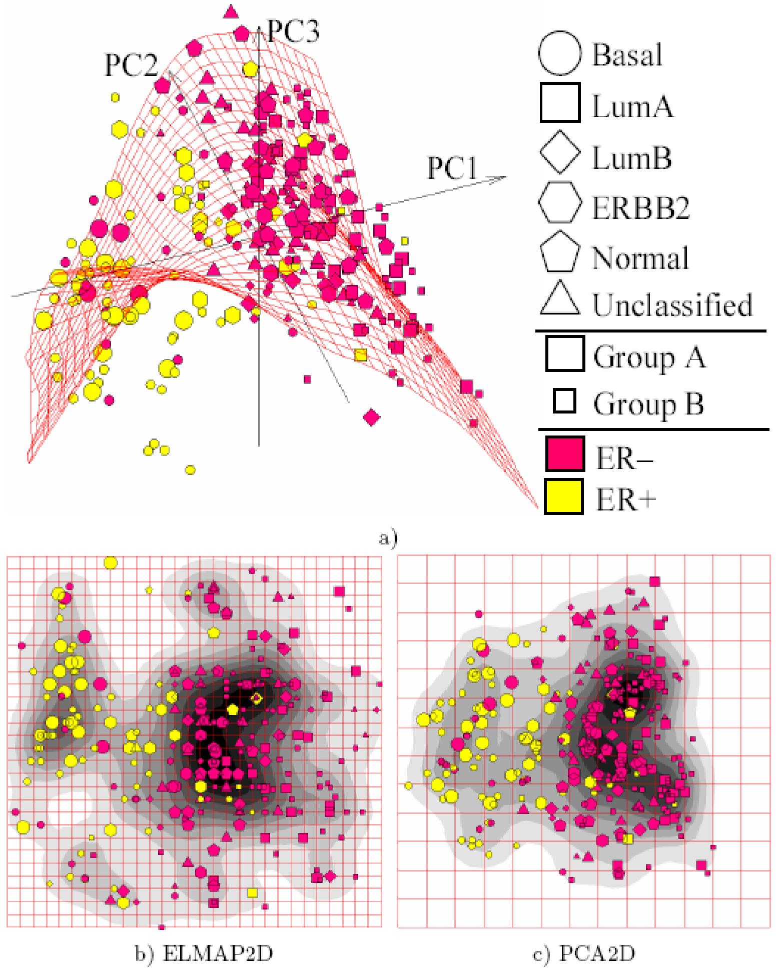
from https://arxiv.org/pdf/1802.03426.pdf

Comments to both t-SNE and UMAP methods

- Parameters really matter
- Cluster sizes in a UMAP plot mean nothing
- Distances between clusters might not mean anything
- Random noise doesn’t always look random
- You may need more than one plot
- For large ‘neighbourhood’ parameters, both methods give results similar to Multi-dimensional scaling or PCA
- Both can work with non-Euclidean metrics in R^p

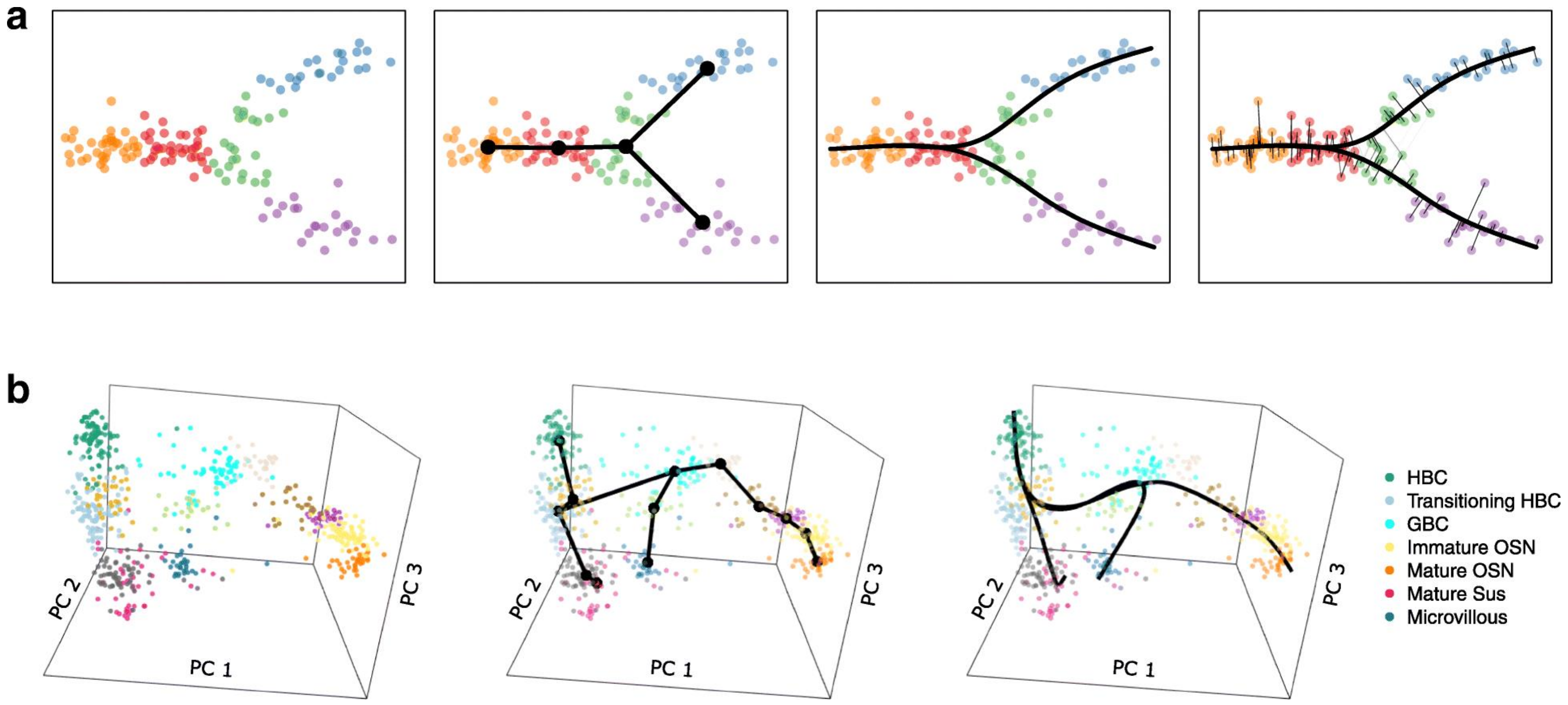
Methods not based on neighbourhood graphs

- Principal manifolds (e.g., elastic maps)
- Self-organizing maps (SOMs)
- Neural network-based autoencoders and variational autoencoders (VAEs)

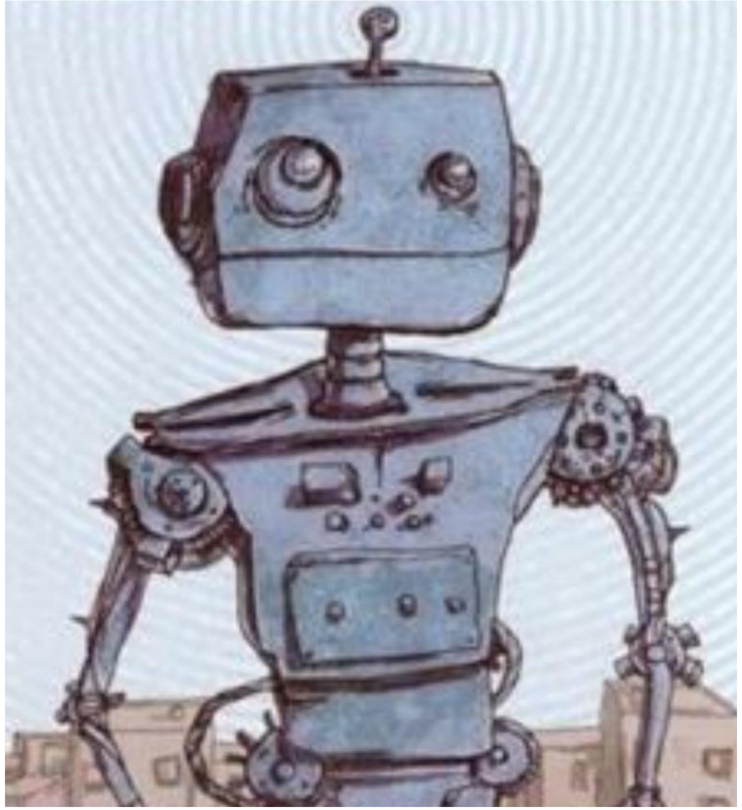


https://en.wikipedia.org/wiki/Elastic_map

Trajectory inference as a special type of manifold learning/clustering



https://en.wikipedia.org/wiki/Trajectory_inference



Good bye!