Postgraduate course

Universitat de Valencia 2020

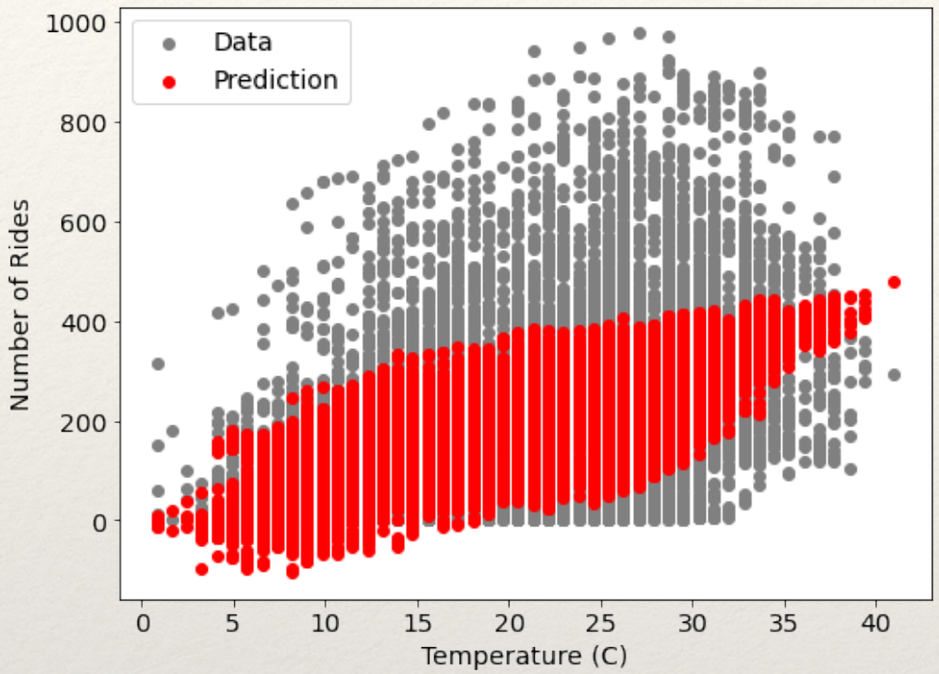
Introduction to Machine Learning for physicists

Veronica Sanz (UV/IFIC)

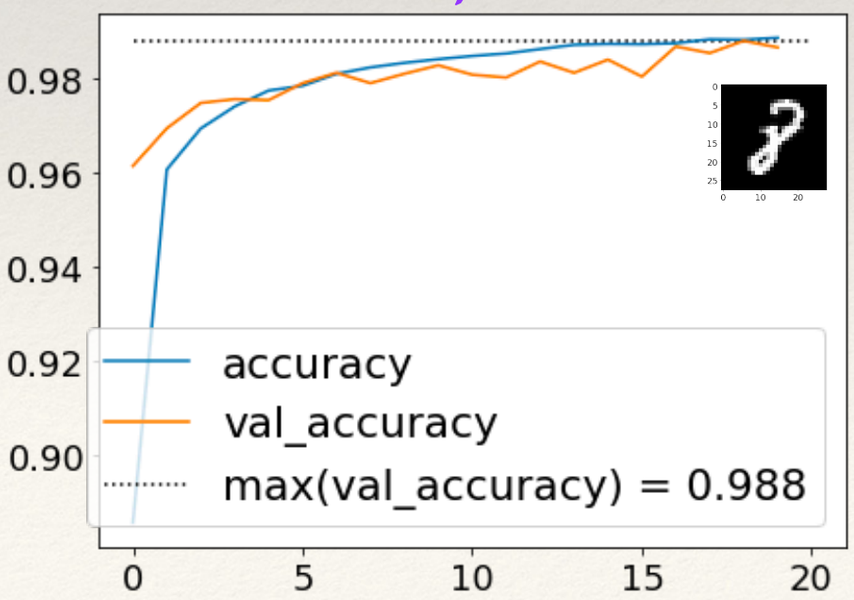
LECTURE 5
UNSUPERVISED

So far we have been getting better at predicting outputs

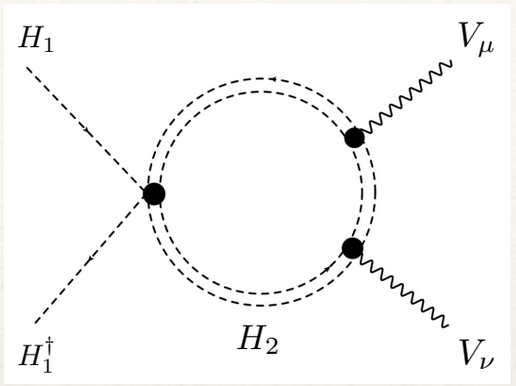
Multivariate linear regression

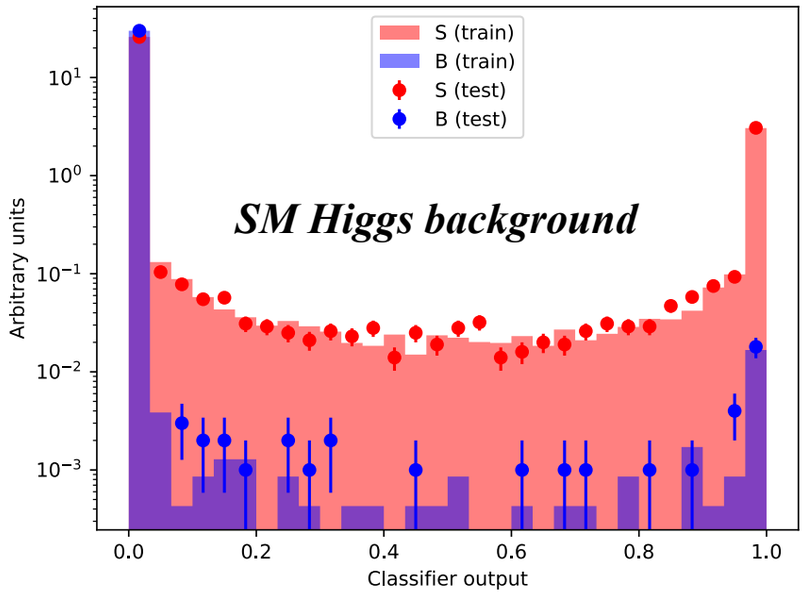


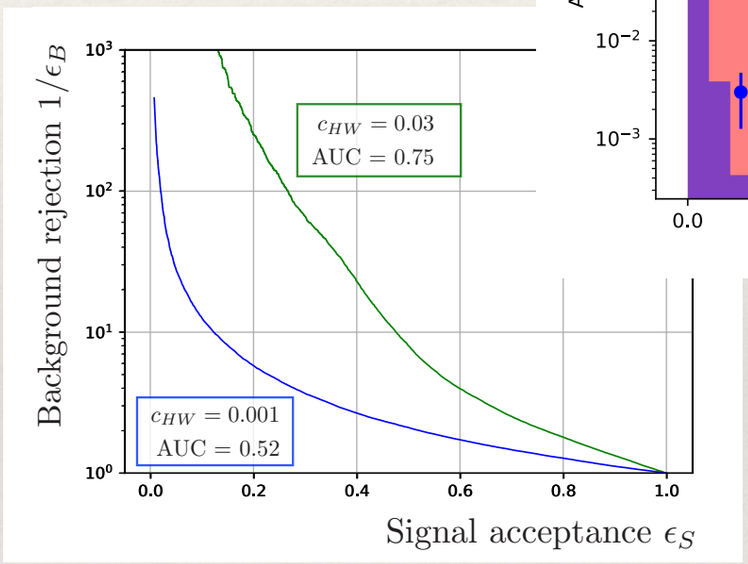
Classification task with images
DNNs, CNNs



Binary classification in particle physics







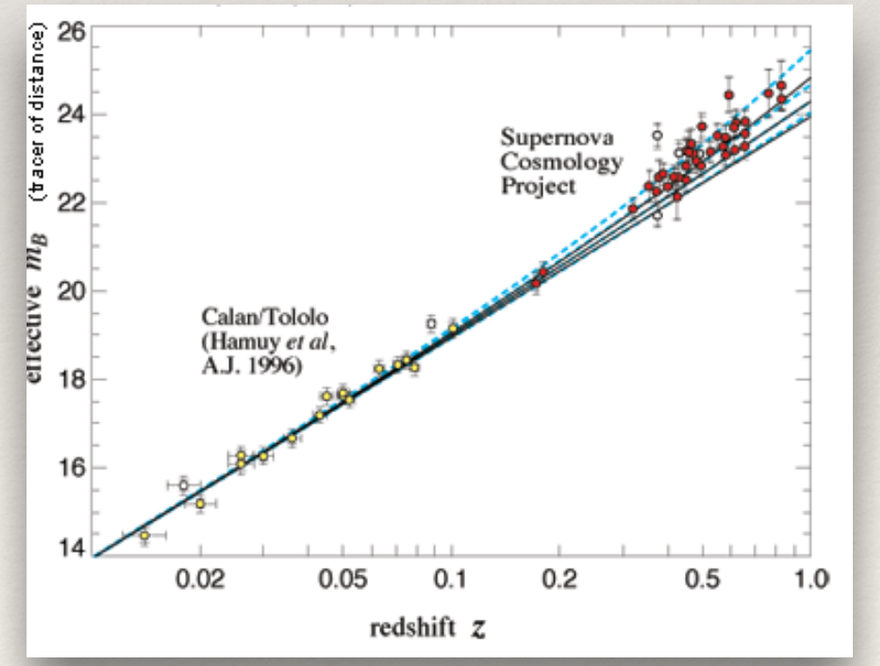
in all these cases we knew the labels,
the output of each input
we knew what we were looking for
our learning was *guided, supervised*

What if we didn't know what we were looking for?

what if the labels were not there?
because they are unknown or too costly to be obtained
or you wanted to learn something beyond these labels?

8.65126013755798E-01	-1.10967397689819E+00	1.37884104251862E+00	1.22582662105556E+00	-1.59931445121765E+00	-3.35545927286148E-01	4.52621191740036E-01	1.396528840065E+00	1.43482126295567E-02	7.61019229888916E-01	8.73991847038269E-01	6.46077632904053E-01	6.55984878540039E-01	2.75867860764265E-02	
6.79041624069214E-01	-1.16113260388374E-01	-1.44924676418304E+00	7.25930333137512E-01	8.32636177539825E-01	-1.06863963603973E+00	7.34275758266449E-01	6.16540551185608E-01	1.10223841667175E+00	-3.8318920135498E-01	6.74581229686737E-01	9.89126741886139E-01	1.30116701126099E+00	2.11356925964355E+00	
6.96710646152496E-01	1.53248345851898E+00	4.51736569404602E-01	9.53612387180328E-01	1.08532404899597E+00	-1.26148509979248E+00	2.19569746404886E-02	7.69306182861328E-01	1.76076497882605E-02	3.32875728607178E-01	6.81880474090576E-01	1.27228811383247E-01	1.65573984384537E-01	4.10077683627605E-02	
1.00609767436981E+00	-2.18872010707855E-01	1.27333605289459E+00	1.62148654460907E+00	7.60176777839661E-02	5.86132228374481E-01	5.61284124851227E-01	1.70752331614494E-01	5.829758644104E-01	4.111288189888E-01	1.04620361328125E+00	7.77394592761993E-01	6.59388244152069E-01	1.17781555652618E+00	
8.37866902351379E-01	1.13452970981598E+00	-9.11251723766327E-01	6.16625964641571E-01	1.00015830993652E+00	1.2927371263504E+00	1.27245831489563E+00	-6.76616489887238E-01	7.64919698238373E-01	1.16551697254181E+00	6.31192624568939E-01	7.86714017391205E-01	1.10604000091553E+00	8.15494537353516E-01	
1.64174771308899E+00	-1.82980680465698E+00	1.50152146816254E-01	2.70264196395874E+00	-1.2972204387188E-01	-1.38081395626068E+00	1.94475173950195E-01	-1.47864866256714E+00	5.51068745553493E-02	3.0361208319664E-01	2.63883185386658E+00	6.36116921901703E-01	2.13914230465889E-01	2.300715893507E-01	2
1.46664869785309E+00	1.15517094731331E-01	-1.03616142272949E+00	8.77246916294098E-01	6.90861403942108E-01	1.13006901741028E+00	8.9072197675705E-01	4.26515430212021E-01	1.28321182727814E+00	-5.63879549503326E-01	1.0900456905365E+00	1.32757031917572E+00	1.08076226711273E+00	1.58807563781738E+00	
1.64159500598907E+00	-1.04459571838379E+00	5.43058097362518E-01	7.44841694831848E-01	-3.0812594294548E-01	-9.56824660301208E-01	1.24903869628906E+00	-1.47708666324615E+00	7.31511473655701E-01	-8.47167015075684E-01	1.16093373298645E+00	1.60183656215668E+00	1.2244086265564E+00	1.25674676895142E+00	
1.36144161224365E+00	-5.06246566772461E-01	-6.46163880825043E-01	1.04786920547485E+00	1.13672995567322E+00	6.98664963245392E-01	1.65792429447174E+00	-6.15012310445309E-02	2.15440988540649E+00	1.67285251617432E+00	1.52531015872955E+00	1.11749887466431E+00	6.50135576725006E-01	0E+00	
1.29748213291168E+00	-2.54911869764328E-01	1.11683440208435E+00	9.79169011116028E-01	-1.35346204042435E-01	-6.9715404510498E-01	1.02749526500702E+00	-1.66322216391563E-01	1.2542005777359E+00	-4.79648977518082E-01	9.85065758228302E-01	1.2597473859787E+00	1.13483691215515E+00	7.60779619216919E-01	
3.7850496172905E-01	-1.08213400840759E+00	1.00809907913208E+00	6.11425042152405E-01	-1.28556573390961E+00	-6.55805096030235E-02	9.16272640228271E-01	-8.86185646057129E-01	1.371866106987E+00	-4.21904683113098E-01	3.91696661710739E-01	7.90428757667542E-01	1.79072606563568E+00	1.38843953609467E+00	
6.75932705402374E-01	1.26536428928375E-01	9.56136286258698E-01	6.86048328876495E-01	-1.99499741196632E-01	-1.70811021327972E+00	9.24258589744568E-01	-6.93052709102631E-01	1.38742446899414E+00	-5.49809157848358E-01	5.77164828777313E-01	1.04205667972565E+00	1.60216736793518E+00	1.92126834392548E+00	
4.63986992835999E-01	1.13328350707889E-02	3.69053810834885E-01	5.53186655044556E-01	3.89279514551163E-01	3.0157333612442E-01	1.65791296958923E+00	1.52030777931213E+00	2.4887318611145E+00	-5.59999048709869E-01	4.2525789141655E-01	1.12646055221558E+00	2.35060715675354E+00	2.46035838127136E+00	
1.16491532325745E+00	-1.21687364578247E+00	1.38738358020782E+00	1.28936243057251E+00	8.53503286838531E-01	-8.64189207553864E-01	5.52336990833282E-01	3.71582925319672E-01	8.29125940799713E-01	-1.4921934902668E-01	1.86399924755096E+00	9.7541731595993E-01	4.64367836713791E-01	1.68607497215271E+00	
8.00468981266022E-01	-9.84456762671471E-02	-7.95078039169312E-01	4.77954834699631E-01	-4.29955363273621E-01	1.08130276203156E+00	8.88447284698486E-01	-3.40310782194138E-01	9.66846108436584E-01	8.3782559633255E-01	5.76379001140594E-01	6.60021543502808E-01	1.01617026329041E+00	6.41459822654724E-01	
3.41047215461731E+00	5.90151190757751E-01	1.50781488418579E+00	2.19785332679749E+00	9.15379881858826E-01	-2.43708327412605E-01	1.00378489494324E+00	9.47994440793991E-02	8.52464973926544E-01	-5.42668044567108E-01	2.5086042881012E+00	2.03399634361267E+00	7.19505548477173E-01	4.56694543361664E-01	

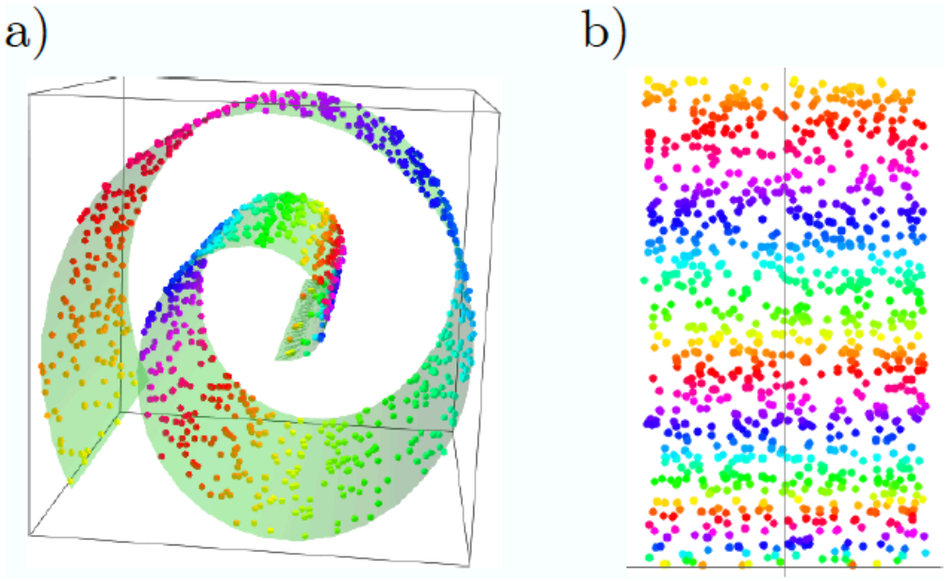
what would you do?
as a physicist, you would start thinking on
possible physical relations, plotting things,
trying to obtain the **best data representation**
the representation which **manifests** a behaviour

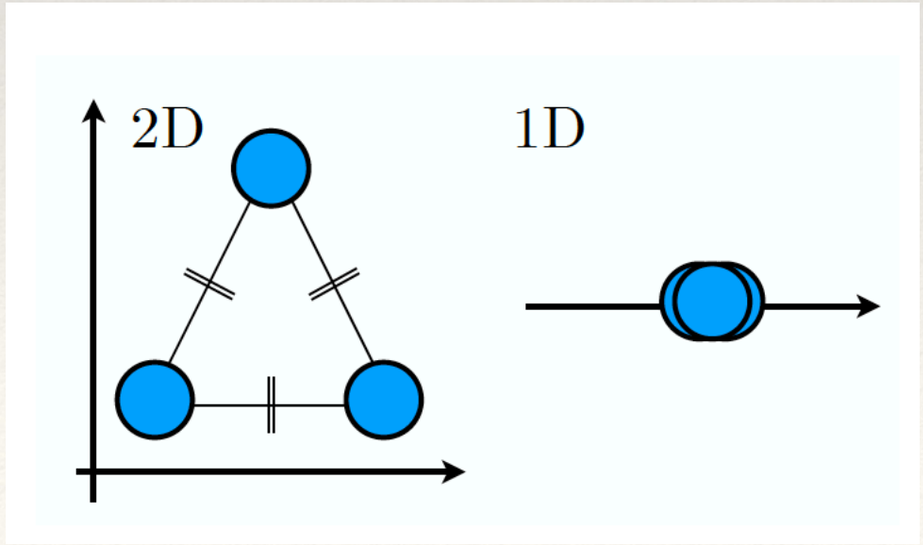


So everything starts with data visualization

But we can't visualise things in more than 3D
when most data we want to mine is high-dimensional…

So you need to do DIMENSIONAL REDUCTION
from original space to **latent space**





Reduction n-D to few-D isn't simply
projecting in a lower dimensional space
one dimension at a time
Choice: *direction* to project
to keep as much info as possible

Bad choice
end up with a *crowding problem*
the best choice to represent this data is 2D,
going to 1D does limit your ability to learn
There's hope, stat dynamics shows
micro->macro can work

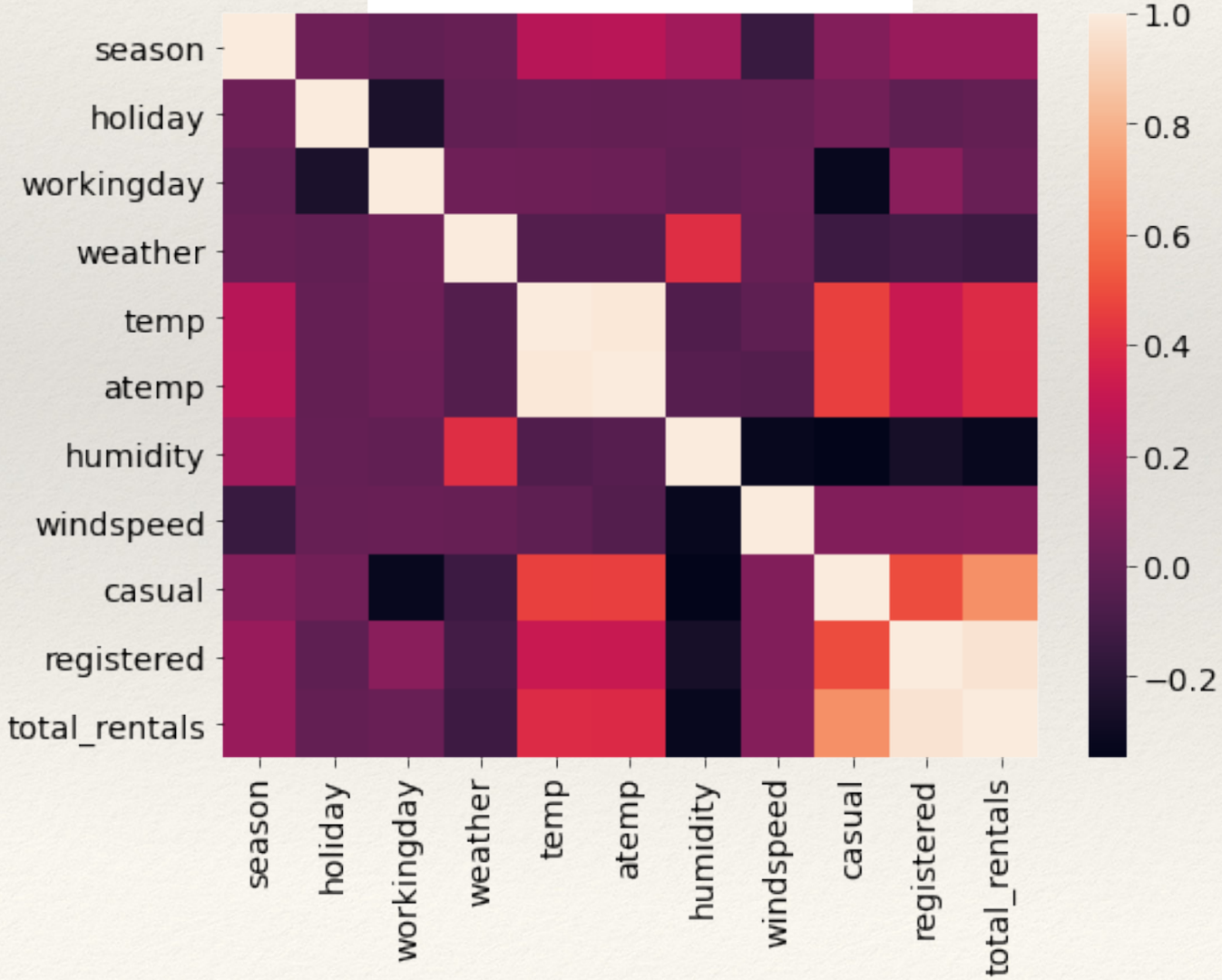
Being smart at dimensional reduction

The direction to project out dimensions is important

We need a *criteria*

Principal Component Analysis (PCA)

$$\boldsymbol{\Sigma}(\boldsymbol{X}) = \frac{1}{N-1}\boldsymbol{X}^T\boldsymbol{X}.$$



In our representation of the data there are clearly some *redundancies*

temp-atemp

registered-total rentals

weather type-all other variables

working day- casual

there could be one or more LINEAR COMBINATIONS of some of these variables which capture most of the information —> best predictor of rides

PCA

The procedure is simple enough, take the correlation matrix

$$\boldsymbol{\Sigma}(\boldsymbol{X}) = \frac{1}{N-1}\boldsymbol{X}^T\boldsymbol{X}.$$

and diagonalize it

$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$ with $\boldsymbol{S}$ diagonal

$$\begin{aligned}\boldsymbol{\Sigma}(\boldsymbol{X}) &= \frac{1}{N-1}\boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \\ &= \boldsymbol{V}\left(\frac{\boldsymbol{S}^2}{N-1}\right)\boldsymbol{V}^T \\ &\equiv \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T.\end{aligned}$$

$\boldsymbol{\Lambda}$

is a diagonal matrix with ordered eigenvalues

$\boldsymbol{V}$

contains the eigenvectors the directions of decreasing eigenvalue

We can then dimensionally reduce, but removing the directions in $\boldsymbol{V}$ with the smallest eigenvalues, the ones which carry less information in correlations

eqs. from this excellent review

t-SNE

PCA is good as a first try at visualisation but is limited by its linearity
Often we would like to preserve **local** structures in higher-dimensions,
and PCA won't do that
A good example of non-linear techniques is
t-SNE (t-stochastic neighbour embedding)

In a nutshell,
t-SNE compares local distributions in the original and latent space

ORIGINAL
$$p_{i|j} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-||x_i - x_k||^2/2\sigma_i^2)}$$

LATENT
$$q_{ij} = \frac{(1+||y_i - y_j||^2)^{-1}}{\sum_{k\neq i}(1+||y_i - y_k||^2)^{-1}}$$

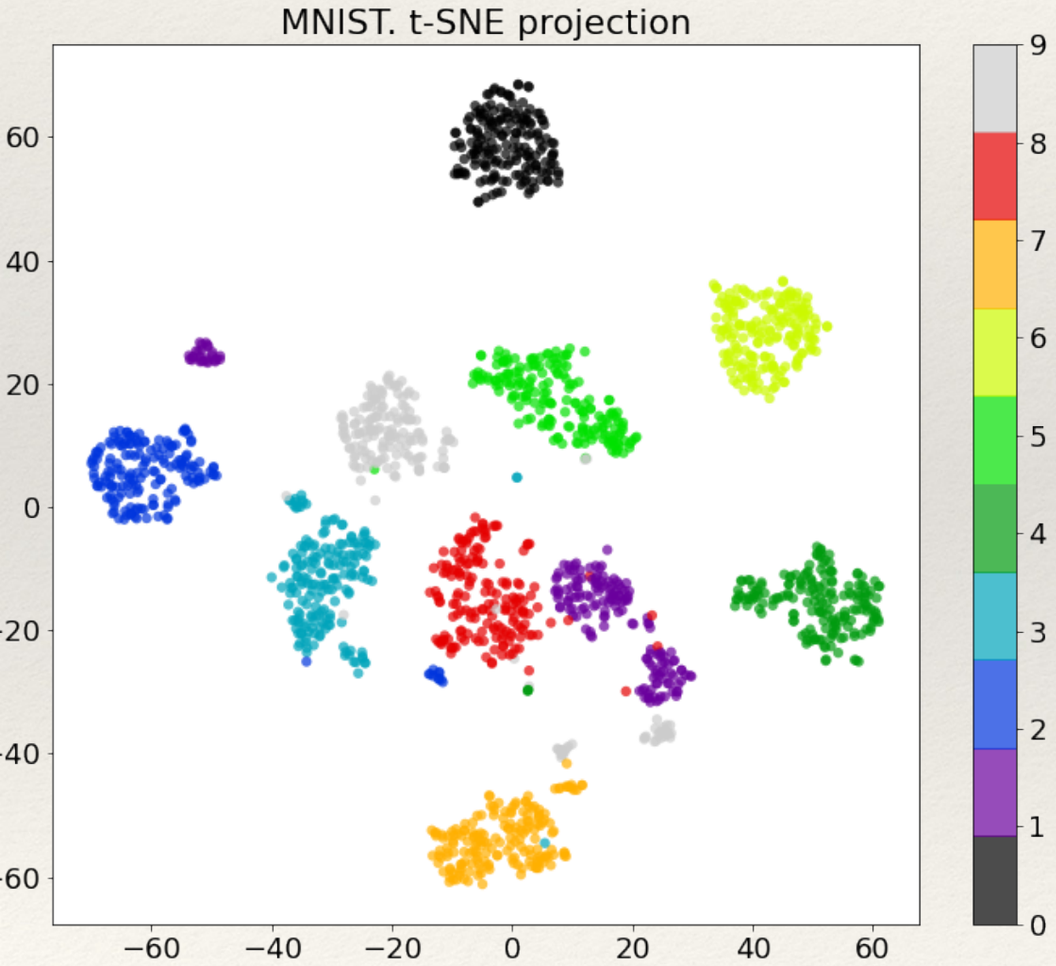
with sigma_i some parameter

and provides a criteria for minimisation

the latent space choice which achieves the minimum is then chosen as latent space

Clustering

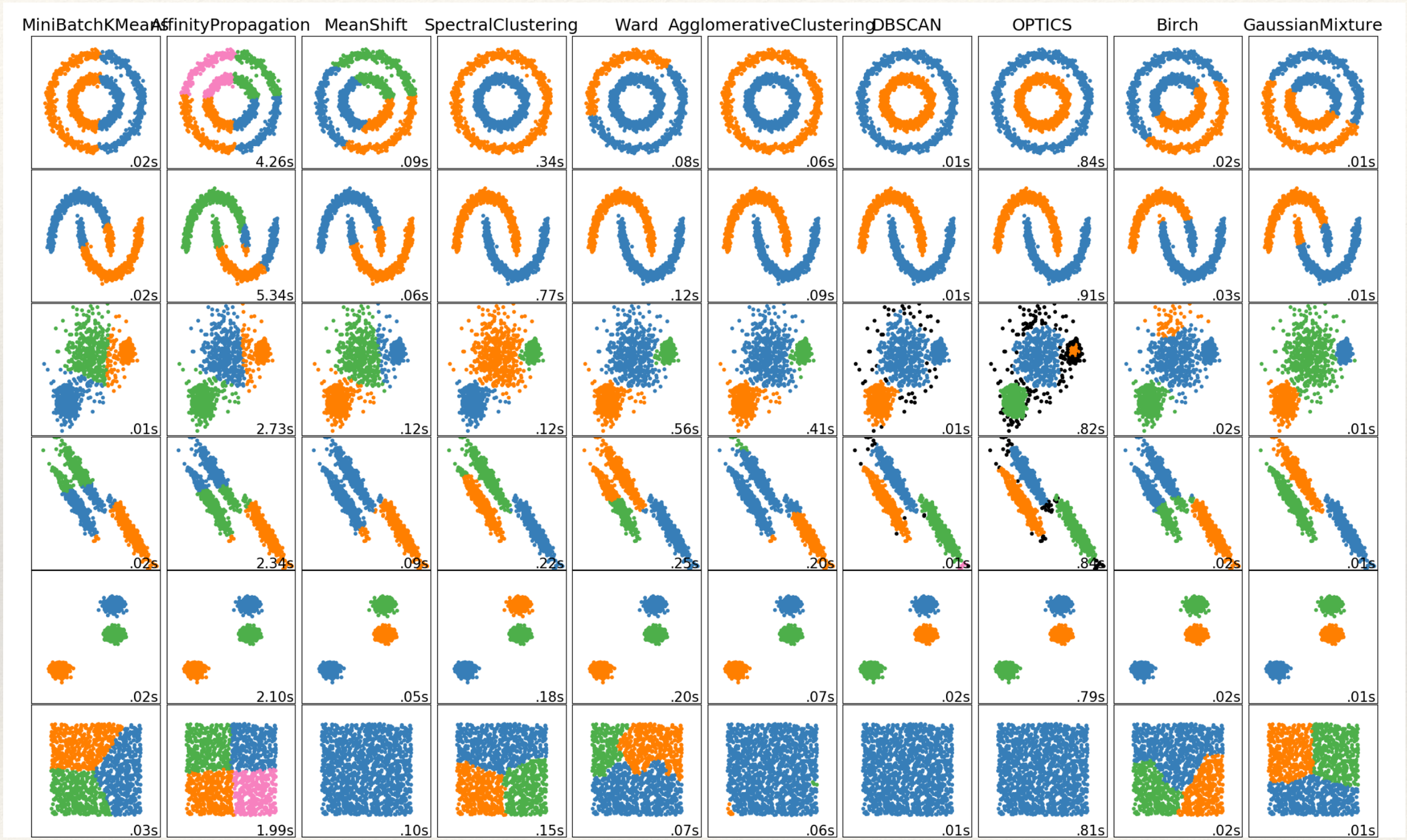
Now that we have reduced dimensionality
by PCA or t-SNE or another method
we can start thinking on finding patterns in it



Clustering is the most intuitive way to find patterns
Finding clusters of common behaviour
using some distance criteria
in the latent space

Clustering is an iterative procedure
start with some parameters like N clusters, cluster size etc
and try clustering the data using these criteria
The mathematical expressions are cumbersome (many definitions of running parameters)
but the intuitive meaning is clear

Different clustering methods



From SCIKIT webpage on clustering methods

Today

We are going to follow a suggestion made by one of you

What if we had the MNIST dataset and didn't have labels?
Would we find that there are 10 classes? and to what accuracy?

Unsupervised learning has less predictive power than supervised learning
provided you can supervise
Nevertheless with t-SNE and K-means we will get 94%

The notebook today is **short**
I want you to realise that all these visualisation and clustering methods
are available with a simple line of code
I would like you to go over it, try to understand it
and do a homework task: use your own choice of dataset
I give you a suggestion:
Samsung Galaxy and S3 data of gyroscopes from users
identify types of activities