Unsupervised Learning

What if we don’t have any labels?

Up until now we have dealt with the case where each data point has a feature vector and a corresponding output label: ($\mathbf{x}$, y).

But what if we don’t have access to y at training time?

Unsupervised Learning

Often our data contains some structure.

e.g.

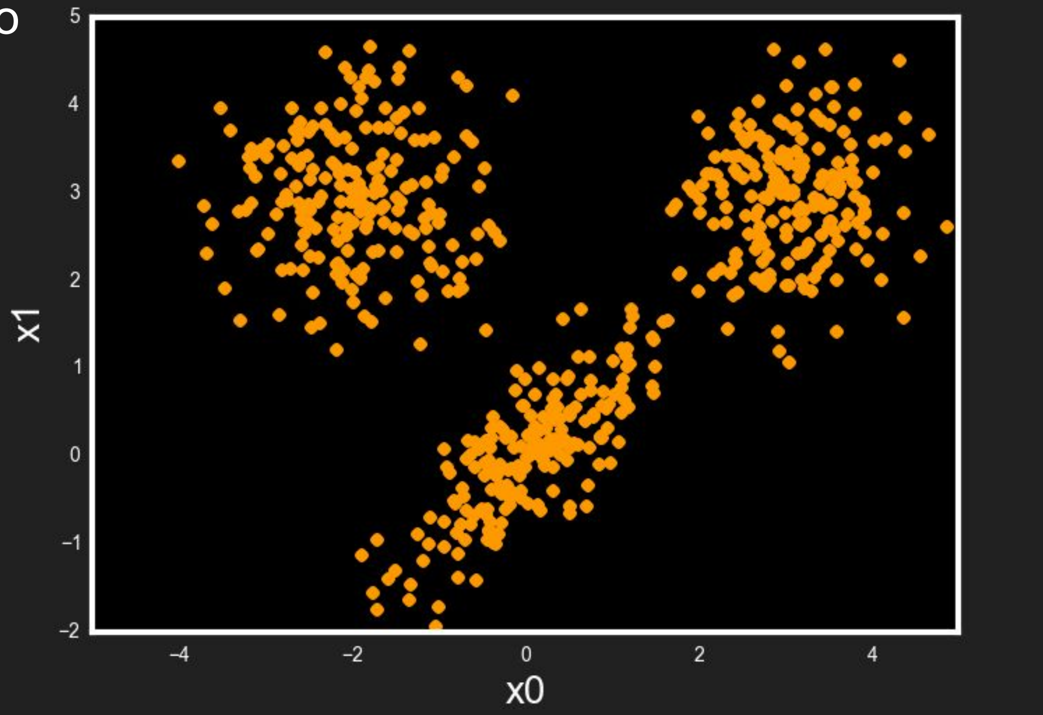
When classifying with linear models, we are assuming separability between classes.

In nearest neighbour classification we assume that datapoints close to each other have the same label. e.g. smoothness

Clustering

What if we want to assign our data to discrete classes but we have no labels?

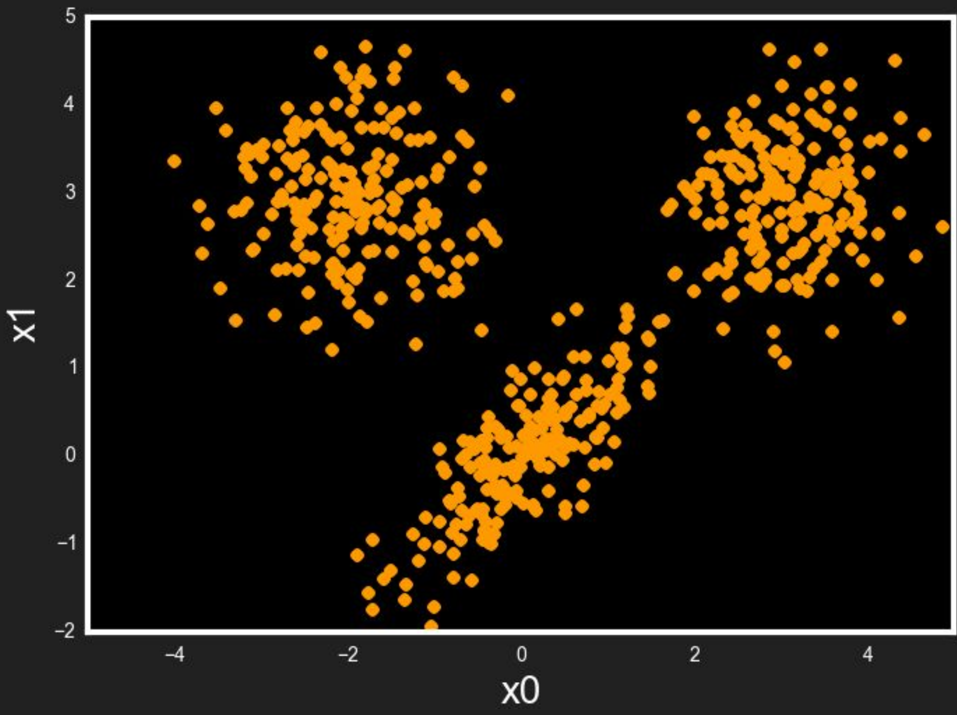
Can we use the structure in the data?



Why Cluster?

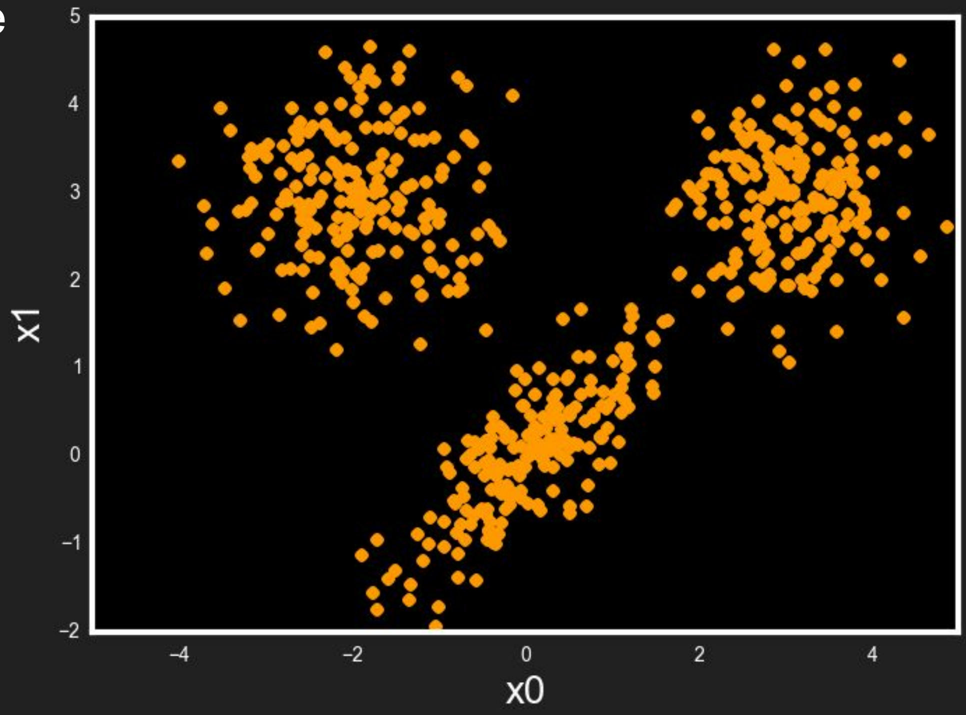
Often we wish find groupings or patterns in our data.

Datapoints in the same cluster are deemed to be similar under some measure.



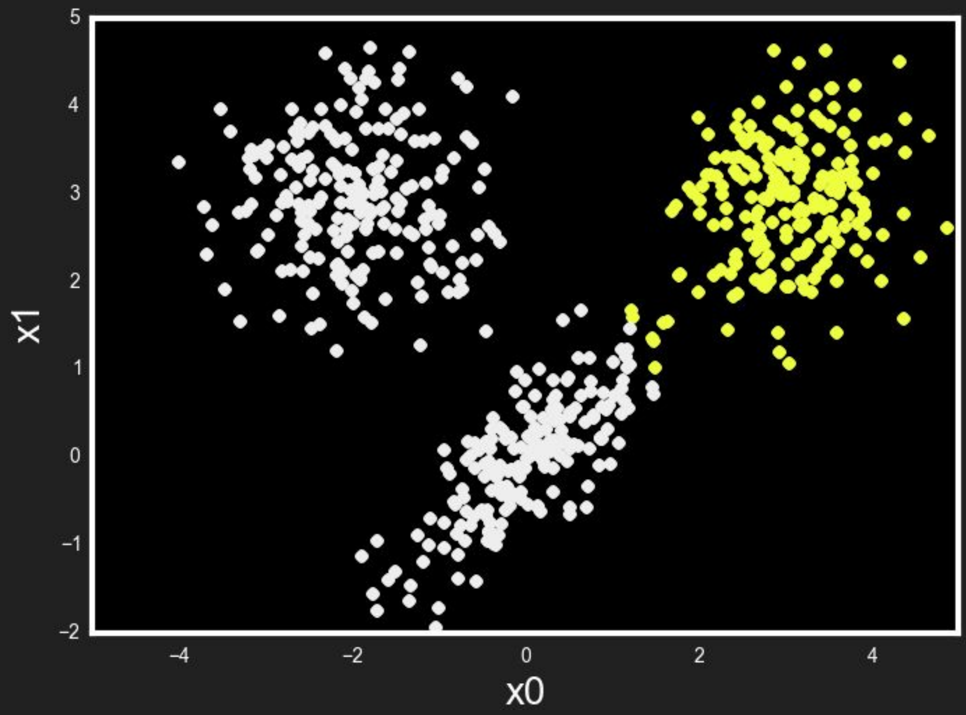
K-Means Clustering

In k-means clustering we specify the number of clusters (K) that we wish to cluster the data into.



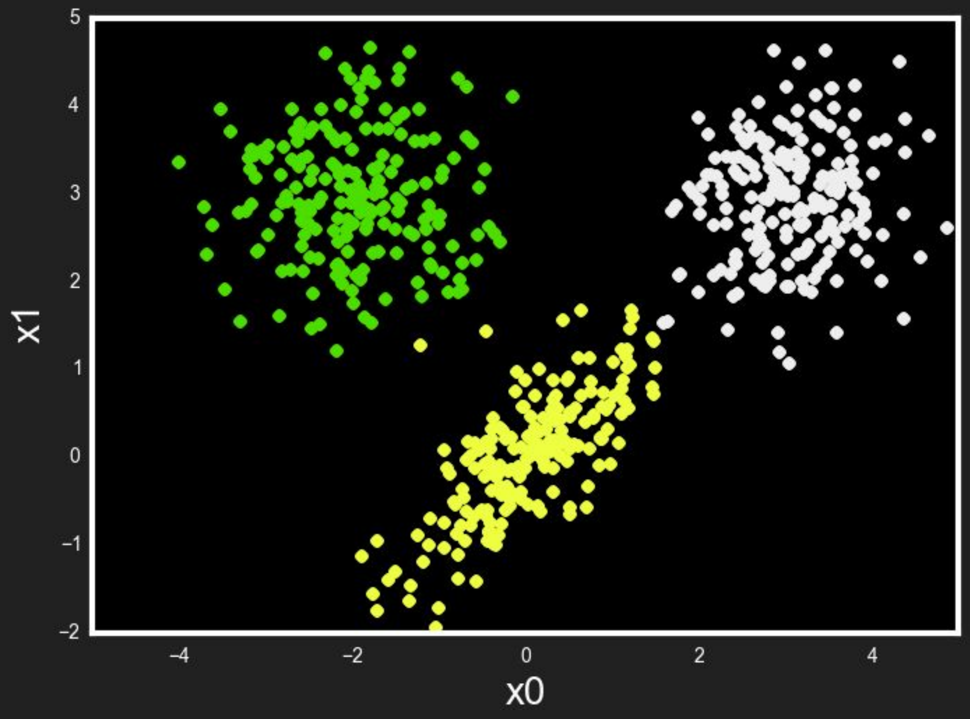
K-Means Clustering

K = 2



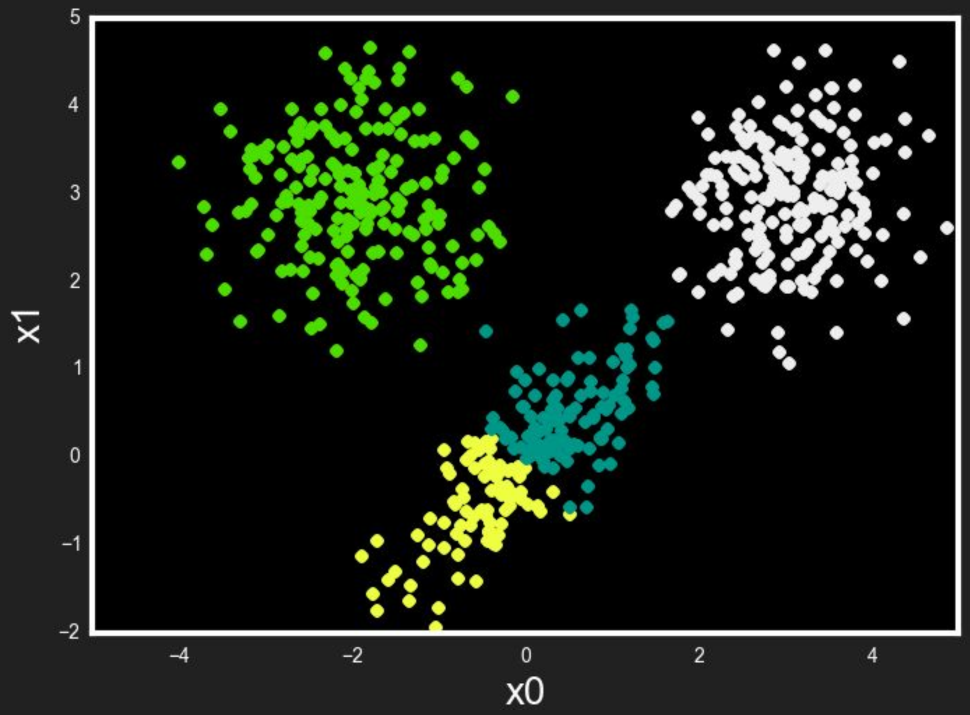
K-Means Clustering

K = 3



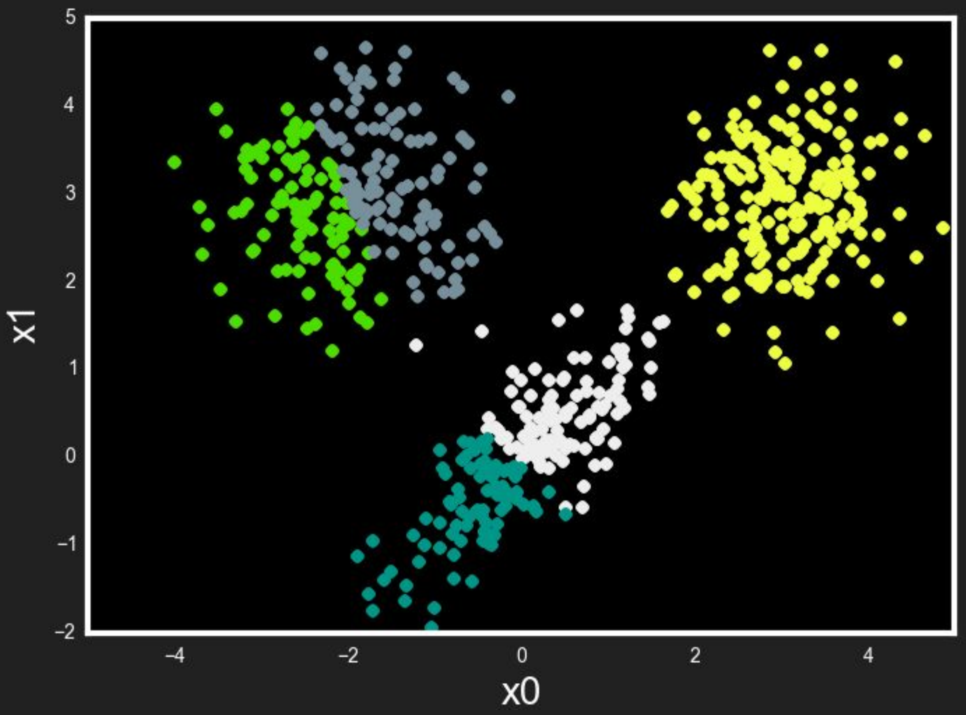
K-Means Clustering

K = 4



K-Means Clustering

K = 5



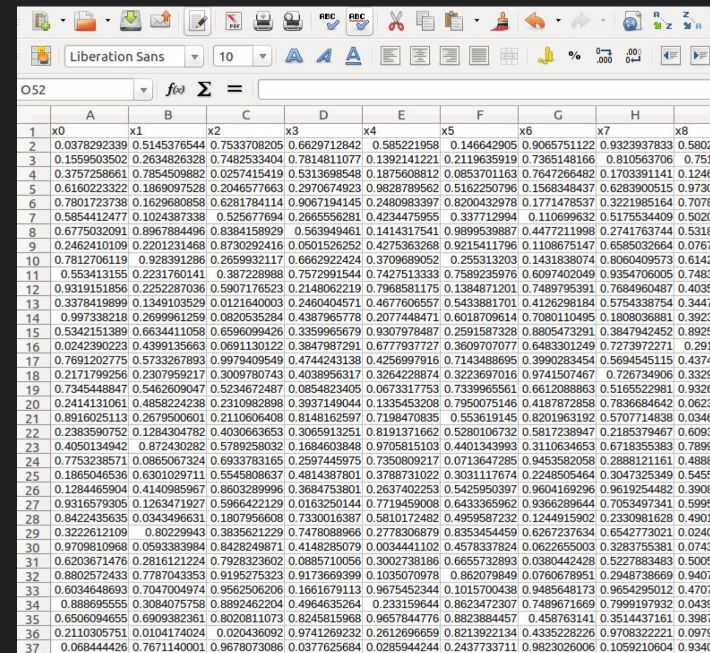
Difficult to know what the correct value of K should be.

Dimensionality Reduction

Often the data we collect can be very high dimensional. e.g. D > 1000

This poses a problem as it is difficult to visualize anything greater than 3 dimensions.

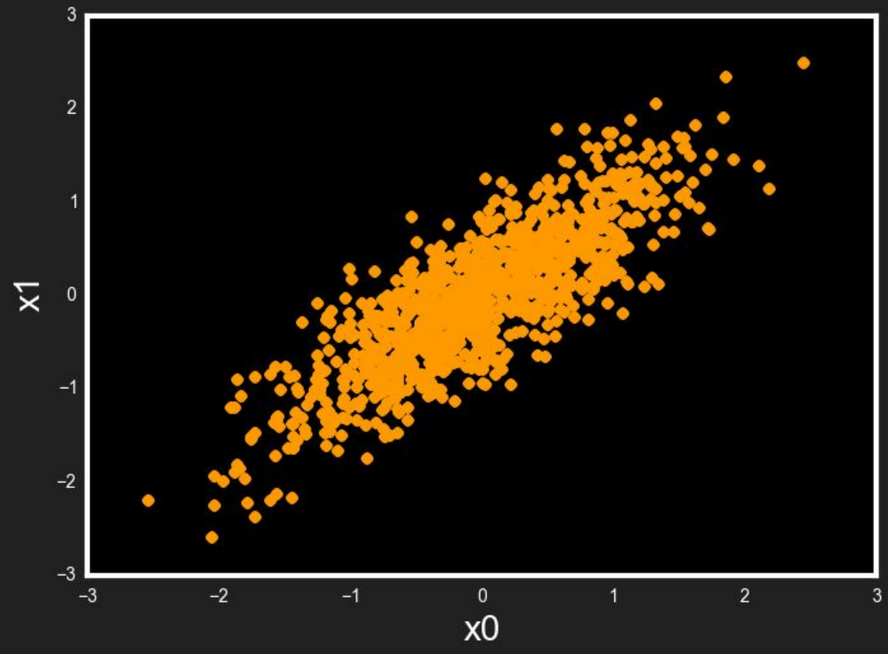
We can project this data down to a lower dimension. P << D, where P is typically 2 or 3



	A	B	C	D	E	F	G	H	I
1	x0	x1	x2	x3	x4	x5	x6	x7	x8
2	0.0378292339	0.5145376544	0.7533708205	0.6629712842	0.585221958	0.146642905	0.9065751122	0.9323937833	0.580203
3	0.1559503502	0.2634826328	0.7482533404	0.7814811077	0.1392141221	0.2119635919	0.7365148166	0.810563706	0.75185
4	0.3757258661	0.7854509882	0.0257415419	0.5313698548	0.1875608812	0.0853701163	0.7647266482	0.1703391141	0.124684
5	0.6160223322	0.1869097528	0.2046577663	0.2970674923	0.9828789562	0.5162250796	0.1568348437	0.6283900515	0.973052
6	0.7801723738	0.1629680858	0.6281784114	0.9067194145	0.2480983397	0.8200432978	0.1771478537	0.3221985164	0.707889
7	0.5854412477	0.1024387338	0.525677694	0.2665556281	0.4234475955	0.337712994	0.110699632	0.5175534409	0.502012
8	0.6775032091	0.8967884496	0.8384158929	0.563949461	0.1414317541	0.9899539887	0.4477211998	0.2741763744	0.531828
9	0.2462410109	0.2201231468	0.8730292416	0.0501526252	0.4275363268	0.9215411796	0.1108675147	0.6585032664	0.076768
10	0.7812706119	0.928391286	0.2659932117	0.6662922424	0.3709689052	0.255313203	0.1431838074	0.8060409573	0.614200
11	0.553413155	0.2231760141	0.387228988	0.7572991544	0.7427513333	0.7589235976	0.6097402049	0.9354706005	0.748303
12	0.9319151856	0.2252287036	0.5907176523	0.2148062219	0.7968581175	0.1384871201	0.7489795391	0.7684960487	0.403583
13	0.3378419899	0.1349103529	0.0121640003	0.2460404571	0.4677606557	0.5433881701	0.4126298184	0.5754338754	0.344714
14	0.997338218	0.2699961259	0.0820535284	0.4387965778	0.2077448471	0.6018709614	0.7080110495	0.1808036881	0.392361
15	0.5342151389	0.6634411058	0.6596099426	0.3359965679	0.9307978487	0.2591587328	0.8805473291	0.3847942452	0.892554
16	0.0242390223	0.4399135663	0.0691130122	0.3847987291	0.6777937727	0.3609707077	0.6483301249	0.7273972271	0.29116
17	0.7691202775	0.5733267893	0.9979409549	0.4744243138	0.4256997916	0.7143488695	0.3990283454	0.5694545115	0.437423
18	0.217179925 6	0.2307959217	0.3009780743	0.4038956317	0.3264228874	0.3223697016	0.9741507467	0.726734906	0.332929
19	0.7345448847	0.5462609047	0.5234672487	0.0854823405	0.0673317753	0.7339965561	0.6612088863	0.516552 2981	0.932639
20	0.2414131061	0.4858224238	0.2310982898	0.3937149044	0.1335453208	0.7950075146	0.4187872858	0.7836684642	0.062306
21	0.8916025113	0.2679500601	0.2110606408	0.8148162597	0.7198470835	0.553619145	0.8201963192	0.5707714838	0.034617
22	0.2383590752	0.1284304782	0.4030663653	0.3065913251	0.8191371662	0.5280106732	0.5817238947	0.2185379467	0.609332
23	0.4050134942	0.872430282	0.5789258032	0.1684603848	0.9705815103	0.4401343993	0.3110634653	0.6718355383	0.789982
24	0.7753238571	0.0865067324	0.6933783165	0.2597445975	0.7350809217	0.0713647285	0.9453582058	0.2888121161	0.488896
25	0.1865046536	0.6301029711	0.5545808637	0.4814387801	0.3788731022	0.3031117674	0.2248505464	0.3047325349	0.545563
26	0.1284465904	0.4140985967	0.8603289996	0.3684753801	0.2637402253	0.5425950397	0.9604169296	0.9619254482	0.390888
27	0.9316579305	0.1263471927	0.5966422129	0.0163250144	0.7719459008	0.6433365962	0.9366289644	0.7053497341	0.599572
28	0.8422435635	0.0343496631	0.1807956608	0.7330016387	0.5810172482	0.4959587232	0.1244915902	0.2330981628	0.490166
29	0.3222612109	0.80229943	0.3835621229	0.7478088966	0.2778306879	0.8353454459	0.6267237634	0.6542773021	0.024091
30	0.9709810968	0.0593383984	0.8428249871	0.4148285079	0.0034441102	0.4578337824	0.0622655003	0.3283755381	0.074316
31	0.6203671476	0.2816121224	0.7928323602	0.0885710056	0.3002738186	0.6655732893	0.0380442428	0.5227883483	0.500547
32	0.8802572433	0.7787043353	0.9195275323	0.9173669399	0.1035070978	0.862079849	0.0760678951	0.2948738669	0.940763
33	0.6034648693	0.7047004974	0.9562506206	0.1661679113	0.9675452344	0.1015700438	0.9485648173	0.9654295012	0.470791
34	0.888695555	0.3084075758	0.8892462204	0.4964635264	0.233159644	0.8623472307	0.7489671669	0.7999197932	0.043965
35	0.6506094655	0.6909382361	0.8020811073	0.8245815968	0.9657844776	0.8823884457	0.458763141	0.3514437161	0.398729
36	0.2110305751	0.0104174024	0.020436092	0.9741269232	0.2612696659	0.8213922134	0.4335228226	0.9708322221	0.097997
37	0.068444426	0.7671140001	0.9678073086	0.0377625684	0.0285944244	0.2437733711	0.9823026006	0.1059210604	0.934008

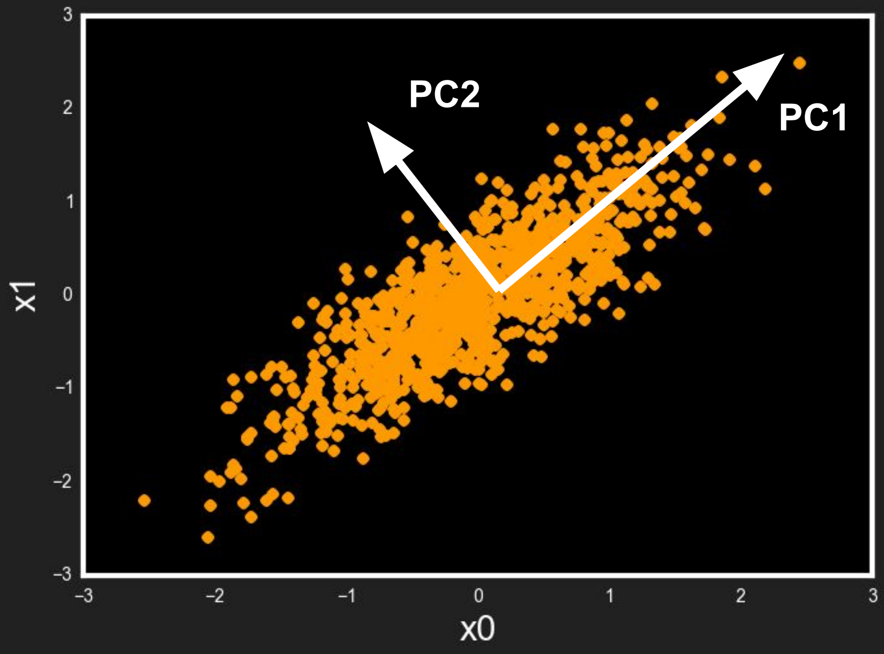
Dimensionality Reduction - PCA

Here we have a 2D dataset.

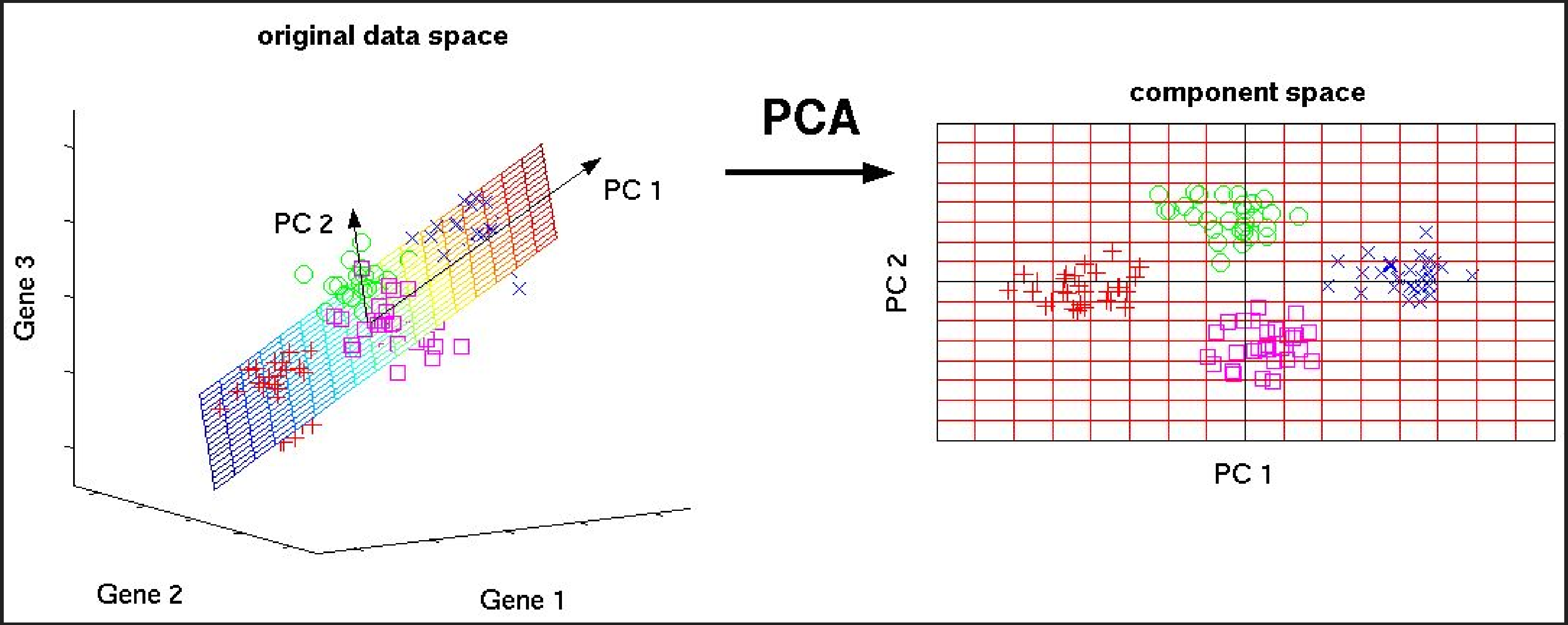


Dimensionality Reduction - PCA

In PCA we perform a linear coordinate transform, rotating the coordinate system so that it aligns with the direction of highest variance.



Dimensionality Reduction - PCA

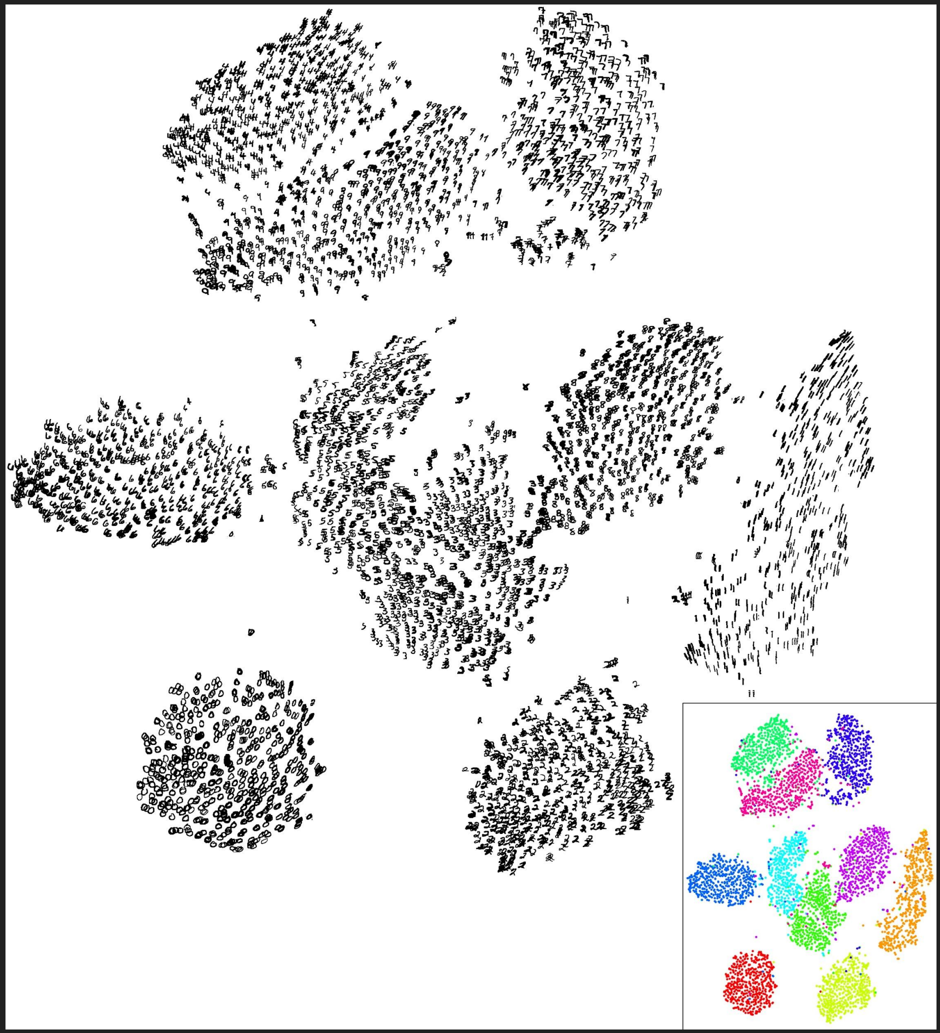


http://www.nlpca.org/pca_principal_component_analysis.html

Dimensionality Reduction - Visualization

This image depicts a dataset of handwritten digits that have been projected to a 2D feature space.

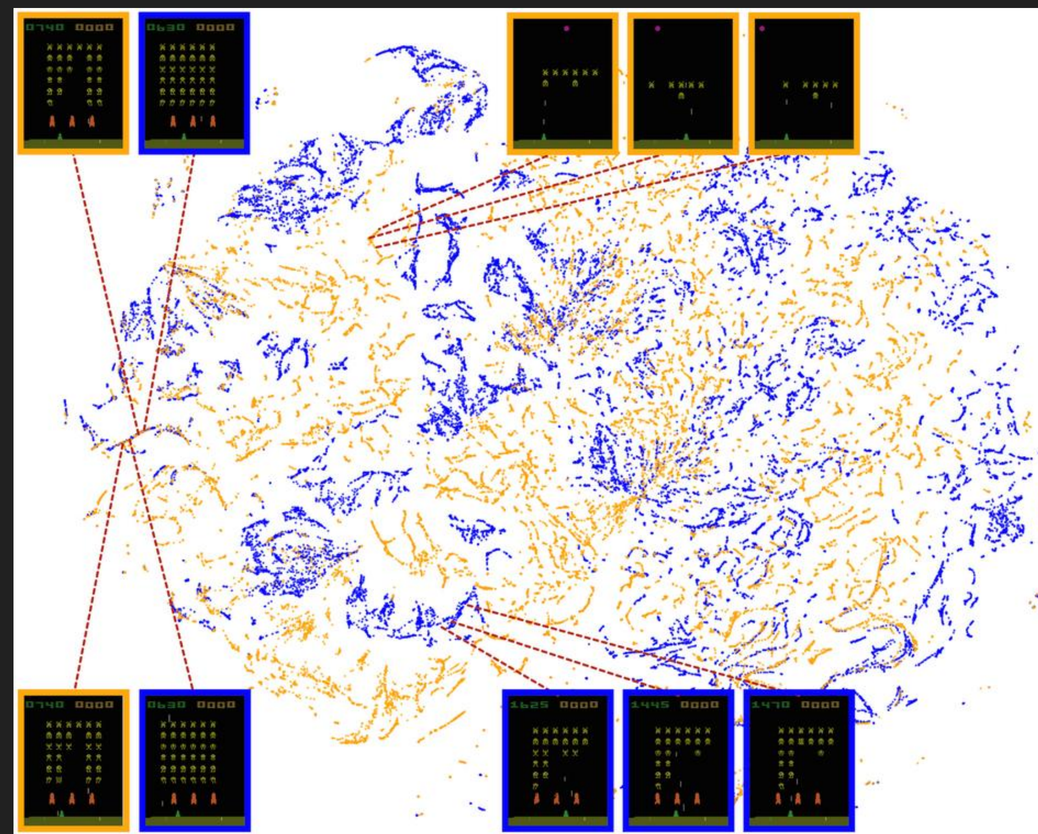
Each point in the 2D space represents an image of hand drawn digit.



https://lvdmaaten.github.io/tsne/

Dimensionality Reduction - Visualization

Here we see a visualization of the features used by an algorithm that has learned how to play video games.

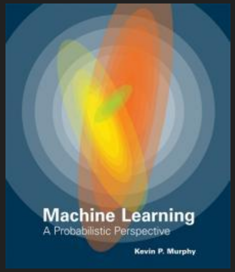


http://www.nature.com/nature/journal/v518/n7540/fig_tab/nature14236_SF1.html

Resources - Books

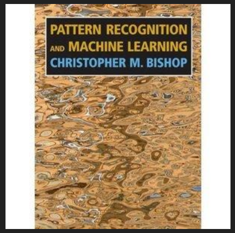
Machine Learning: A Probabilistic Perspective - Kevin P. Murphy

https://mitpress.mit.edu/books/machine-learning-0



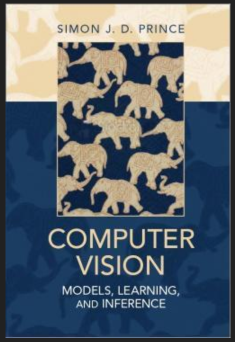
Pattern Recognition and Machine Learning - Chris Bishop

http://research.microsoft.com/en-us/um/people/cmbishop/prml/



Computer Vision: Models, Learning, and Inference - Simon J.D. Prince

http://www.computervisionmodels.com/



Resources - Online Courses

Machine Learning - Andrew Ng (Stanford)

https://www.coursera.org/learn/machine-learning

Machine Learning - mathematicalmonk (Khan Academy Style)

https://www.youtube.com/playlist?list=PLD0F06AA0D2E8FFBA

Resources - Code

Python: Scikit-Learn

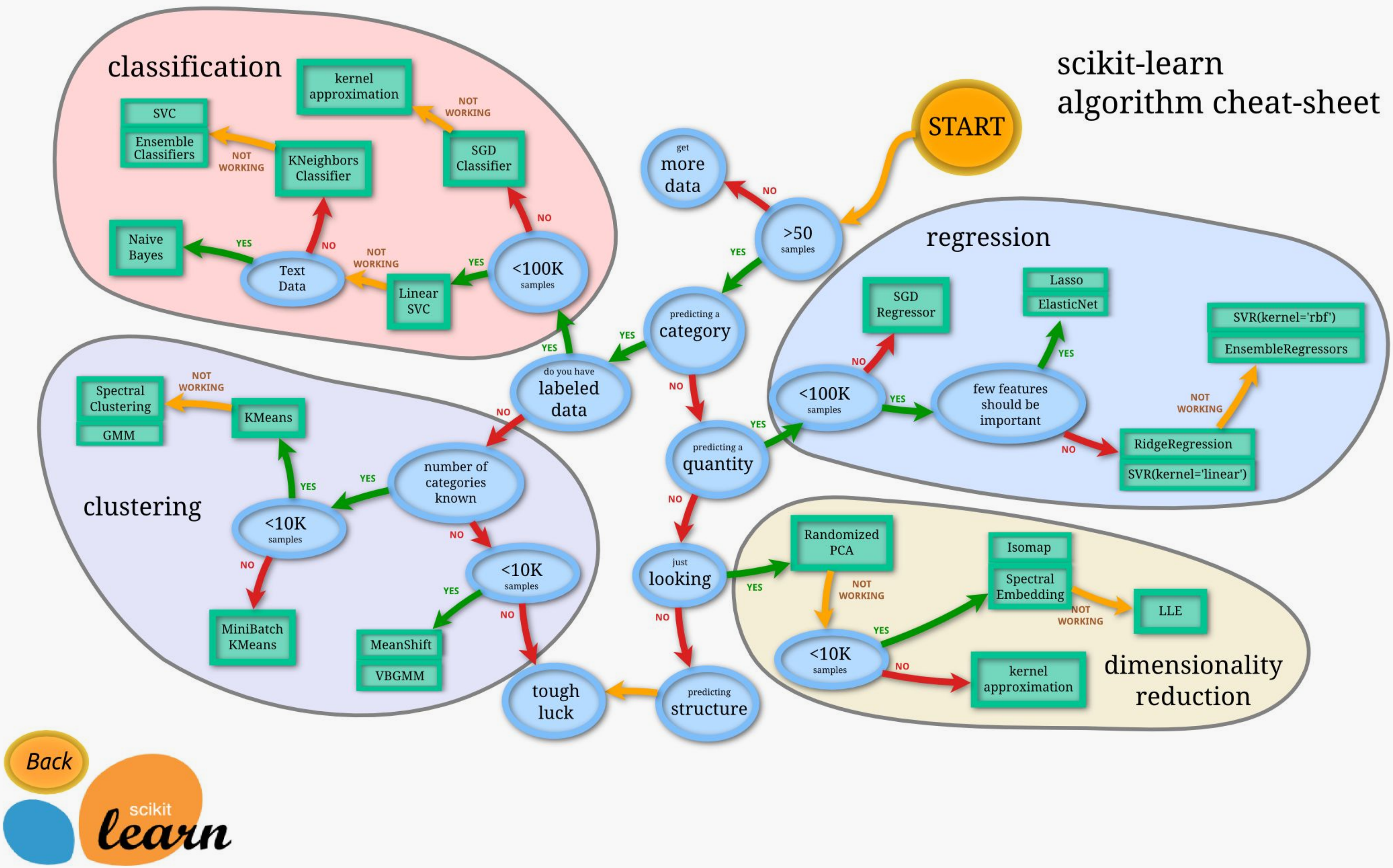
Uniform interface, supervised & unsupervised, lots of examples & tutorials, ...

http://scikit-learn.org/

R: Caret

Uniform interface, multiple classification & regression algorithms, data splitting, ...

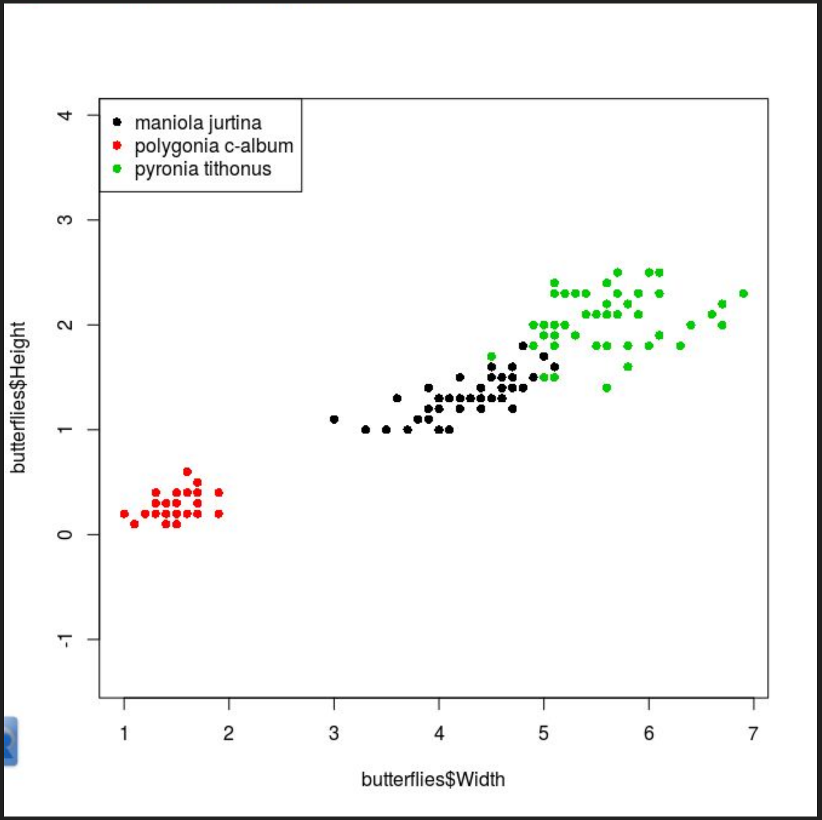
http://topepo.github.io/caret/index.html



http://scikit-learn.org/stable/_static/ml_map.png

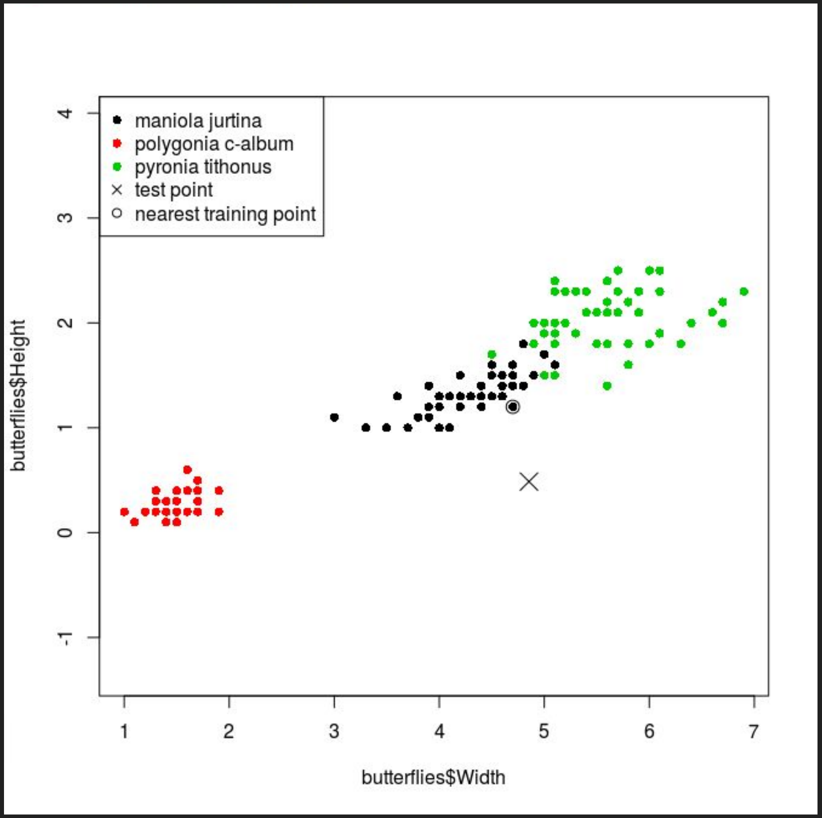
Review - Plotting and Feature Spaces

We saw examples in 1D and 2D.



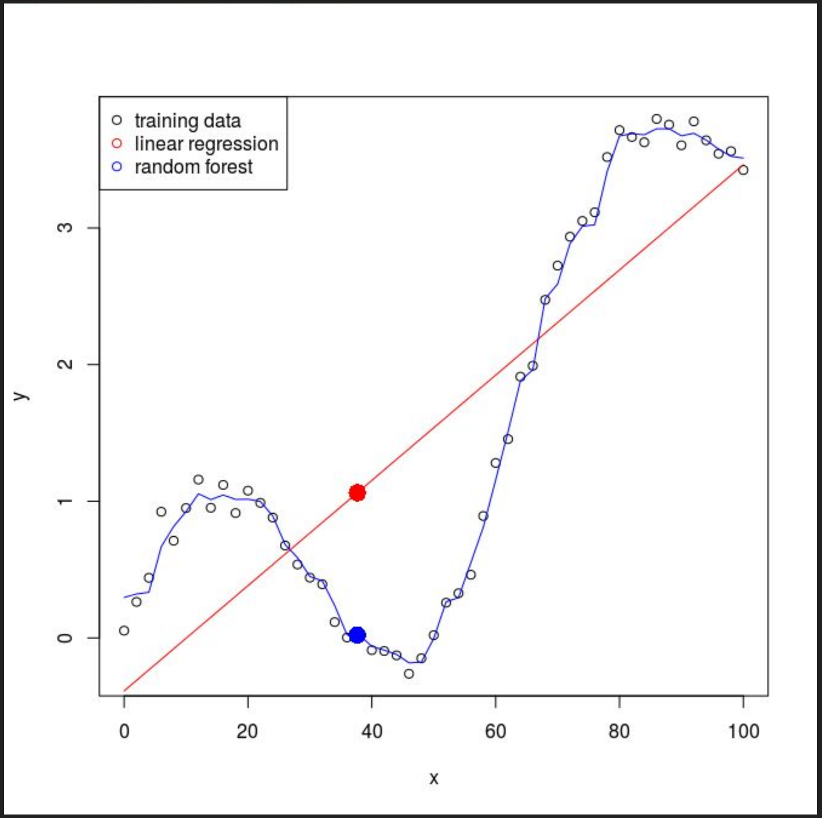
Review - Classification

Given a feature vector **x** as input we predicted its discrete class label y.



Review - Regression

Given a feature vector **x** as input we predicted a corresponding continuous value y.



Conclusion

The slides and code are available online here:

www.cs.ucl.ac.uk/staff/O.MacAodha/ml_intro