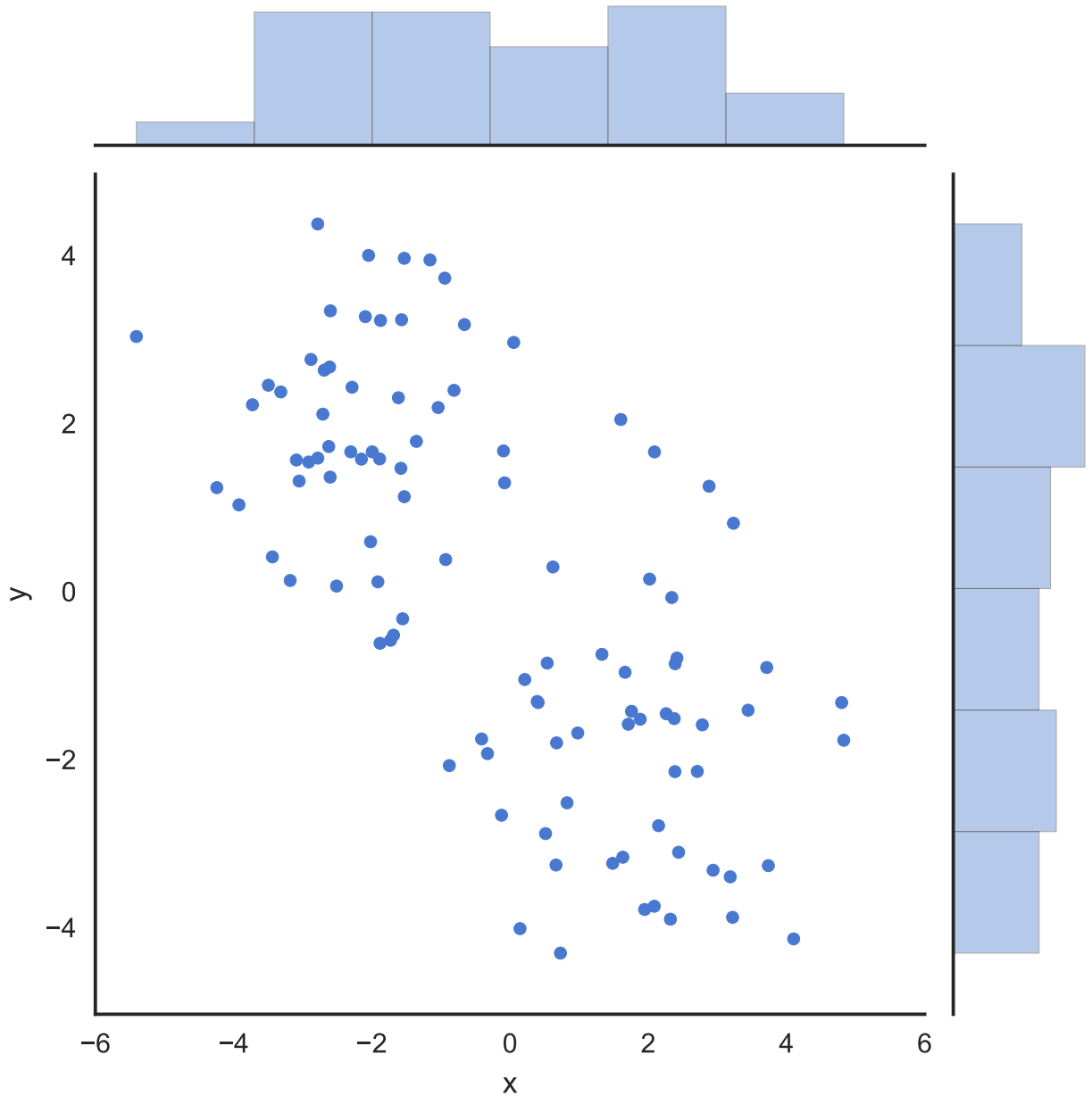
An introduction to multivariate pattern analysis

Phillip M. Alday

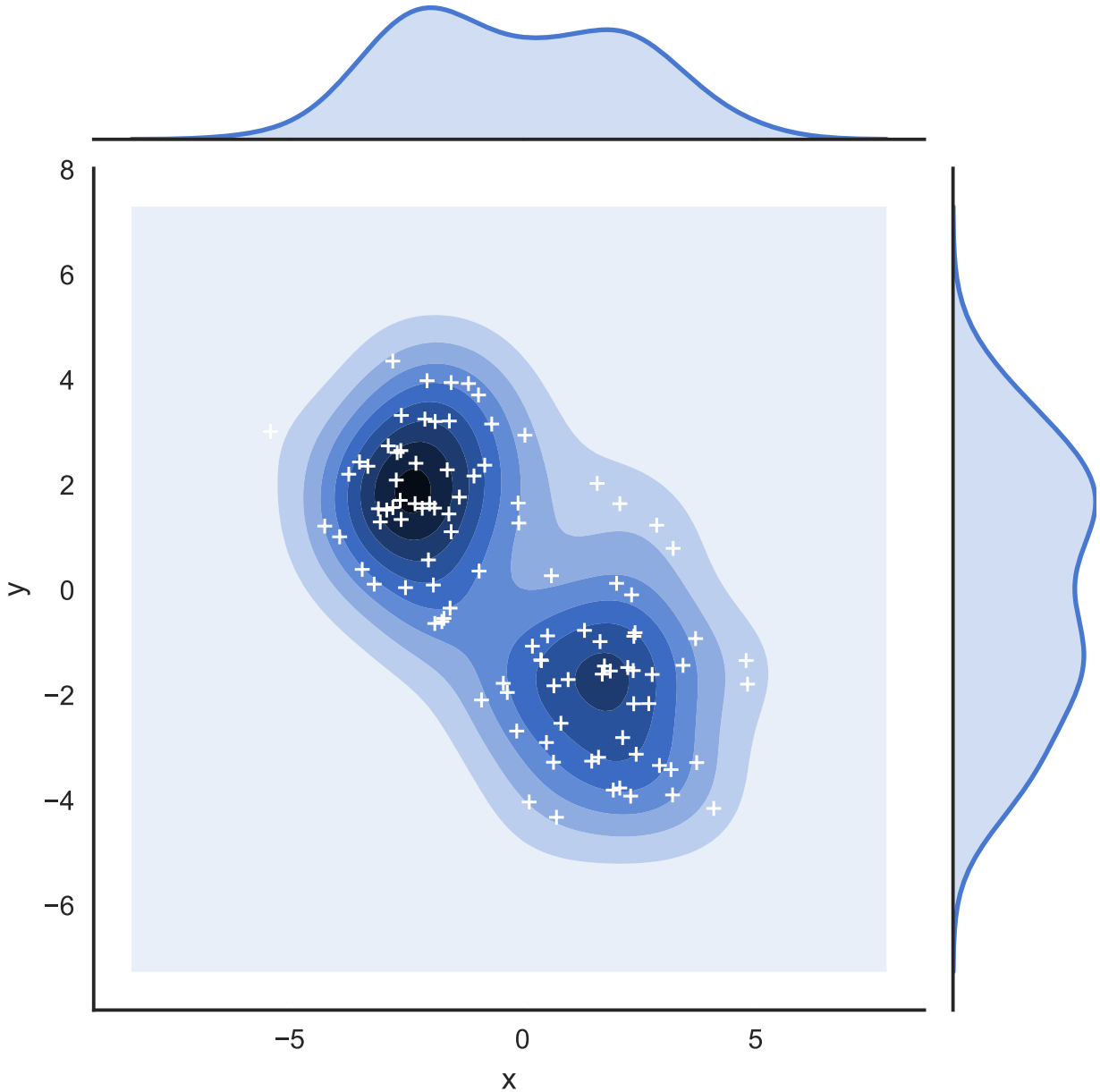
3 March 2015

Why MVPA?

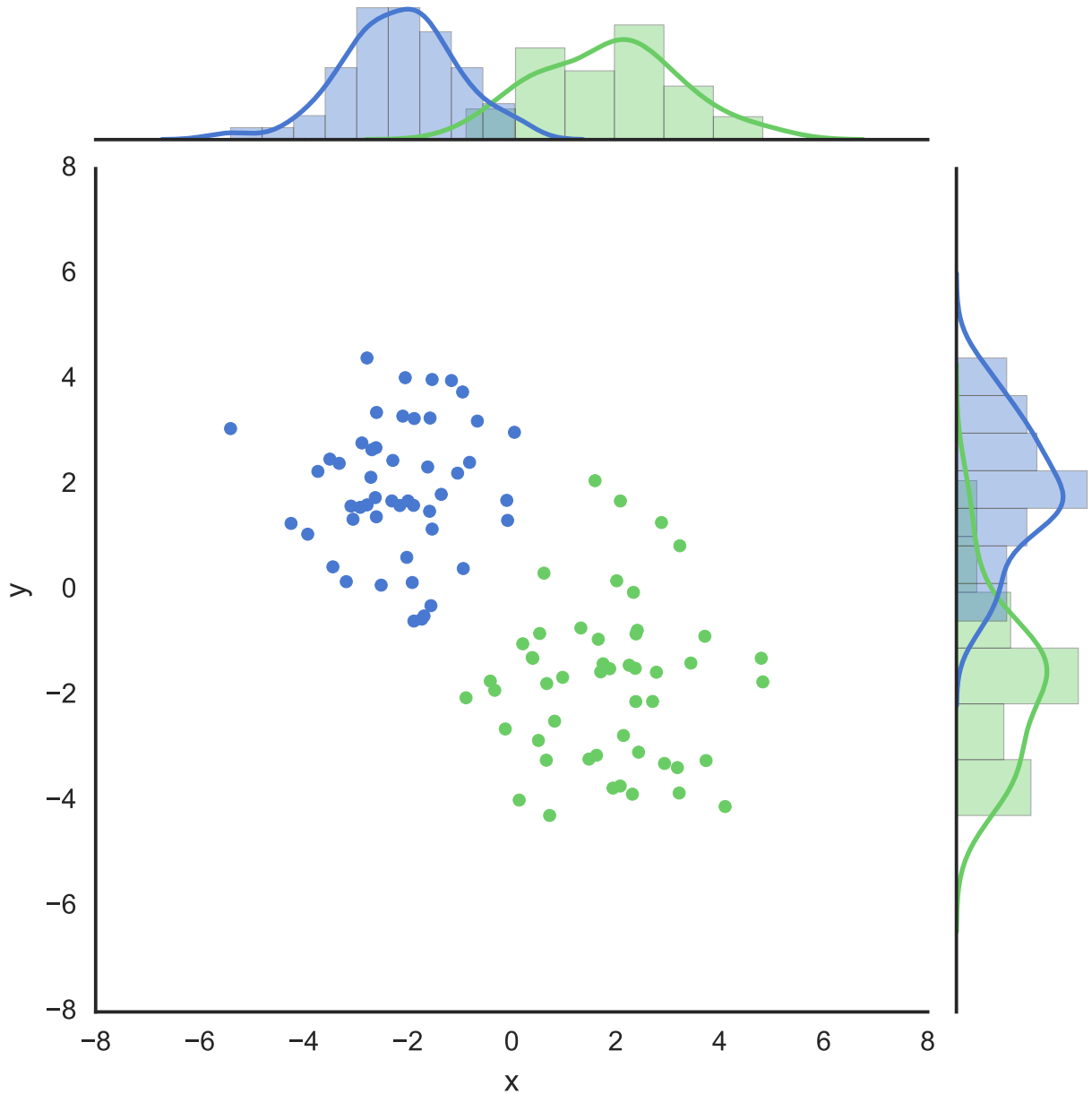
How many clusters are there?



Two, right?



Two, because I made it that way



MultiVariate Pattern Analysis

- multivariate life is hard, so we often go univariate on our dependent measure, e.g. mean ERP voltage in a given time window
- it's often possible to 'fake' multivariate analysis by including additional independent measures, e.g. ROI
- this doesn't imply anything about causality, but some of the mathematical details differ
- e.g. how do we test ERP topographies against each other?
- MVPA is a modern approach drawing from computational advances of the last few decades

The Two Cultures (Breiman 2001)

Data modelling (*statistics*)

- assumes that the structure of the statistical model somehow reflects the structure of reality
- focus on estimation
- interpretability of model parameters important
- often expressed as "Is x a significant predictor of y?"

Algorithmic modelling (*machine learning*)

- assumes that the structure of the statistical model is irrelevant because the structure of reality is not known
- focus on prediction
- interpretability of model parameters not important
- often expressed as "How accurate is my model (when applied to new data)"?

Multivariate "statistics"

- PCA and its reincarnation factor analysis
- MANOVA
- Linear discriminant analysis
- simultaneous equation modles (e.g. $(Y_1, Y_2) = \beta X + \beta_0 + \varepsilon$)

Multivariate "machine learning"

- classification
  - neural networks
  - support vector machines (SVM)
  - naive Bayes
- clustering (automatic grouping of similar objects into sets)
  - *k*-means
  - and many others

Machine learning

Machine learning

Supervised learning

The computer is "taught" via a training set where we already know the answers, i.e. what's right and wrong.

Unsupervised learning

The computer must "explore" the training set to find the solution best matching our assumptions (often expressed via algorithmic choice).

Machine learning

Supervised learning

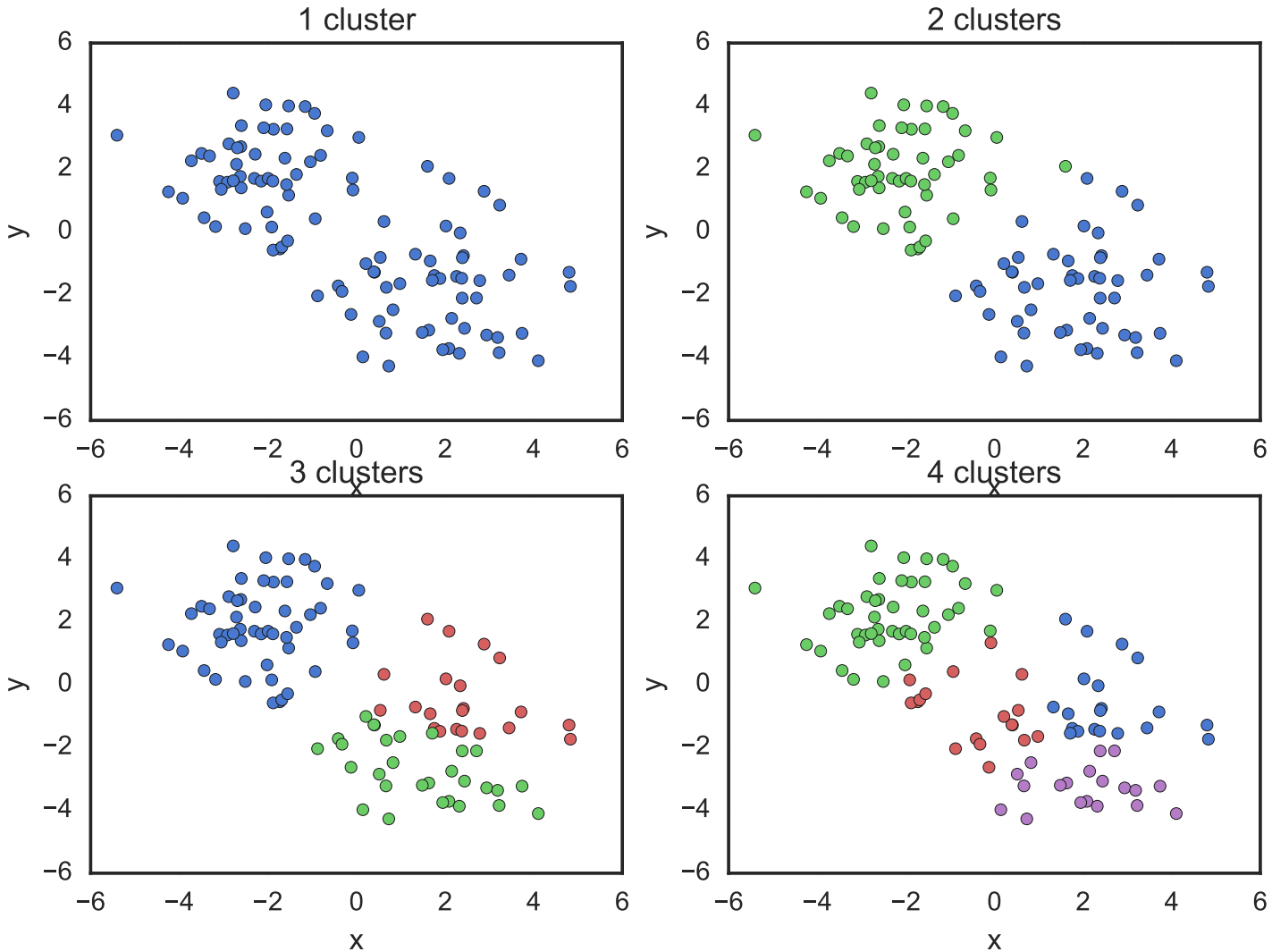
The computer is "taught" via a training set where we already know the answers, i.e. what's right and wrong.

Unsupervised learning

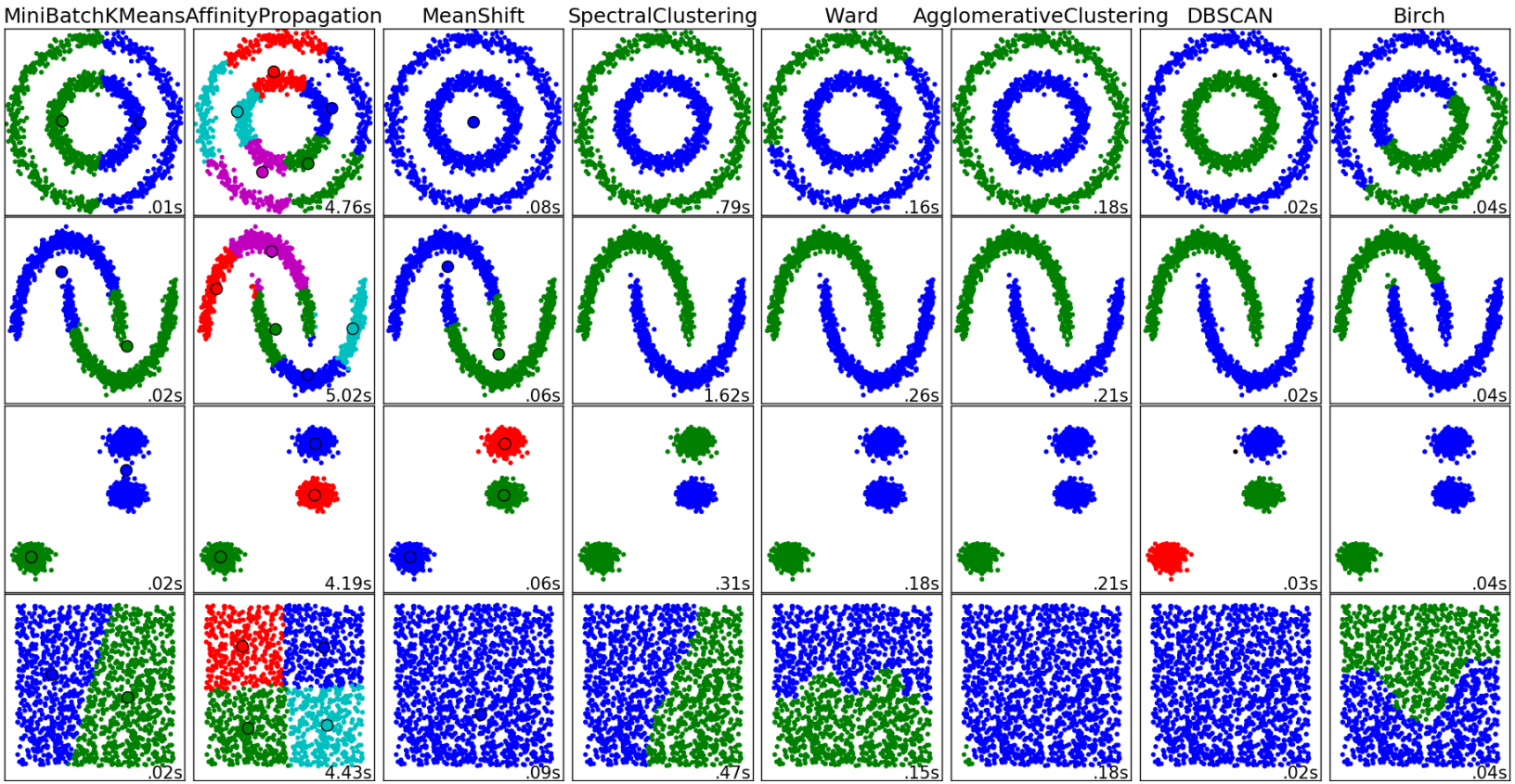
The computer must "explore" the training set to find the solution best matching our assumptions (often expressed via algorithmic choice).

Both types of machine learning involve a number of assumptions. While they are arguably more "data-driven" than traditional statistical modelling in the sense that the model structure is selected algorithmically, the ultimate shape of the model is dependent on both the data and the assumptions!

Clustering with *k*-means



There is no ideal clustering method



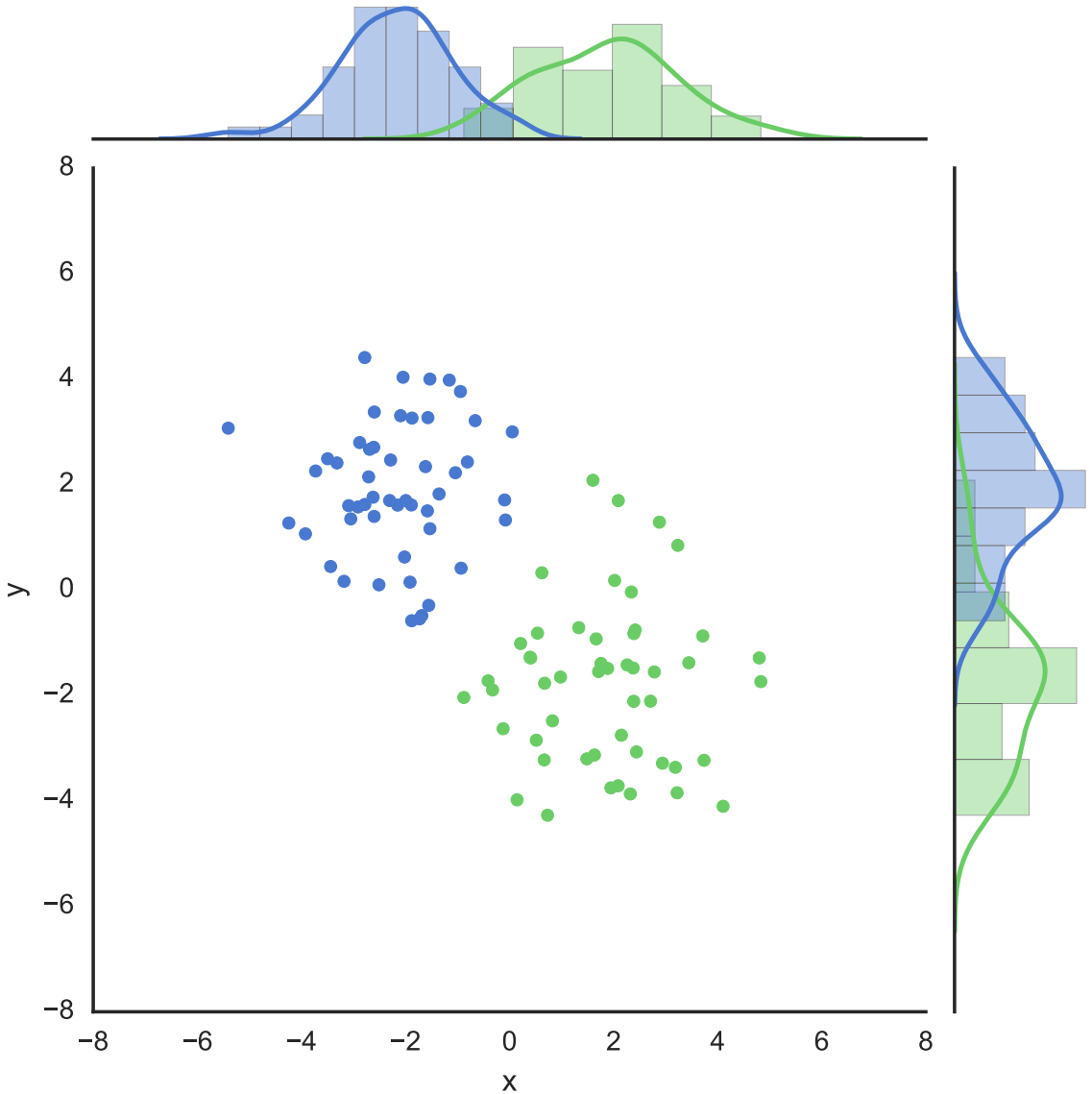
http://scikit-learn.org/stable/modules/clustering.html

Support vector machines

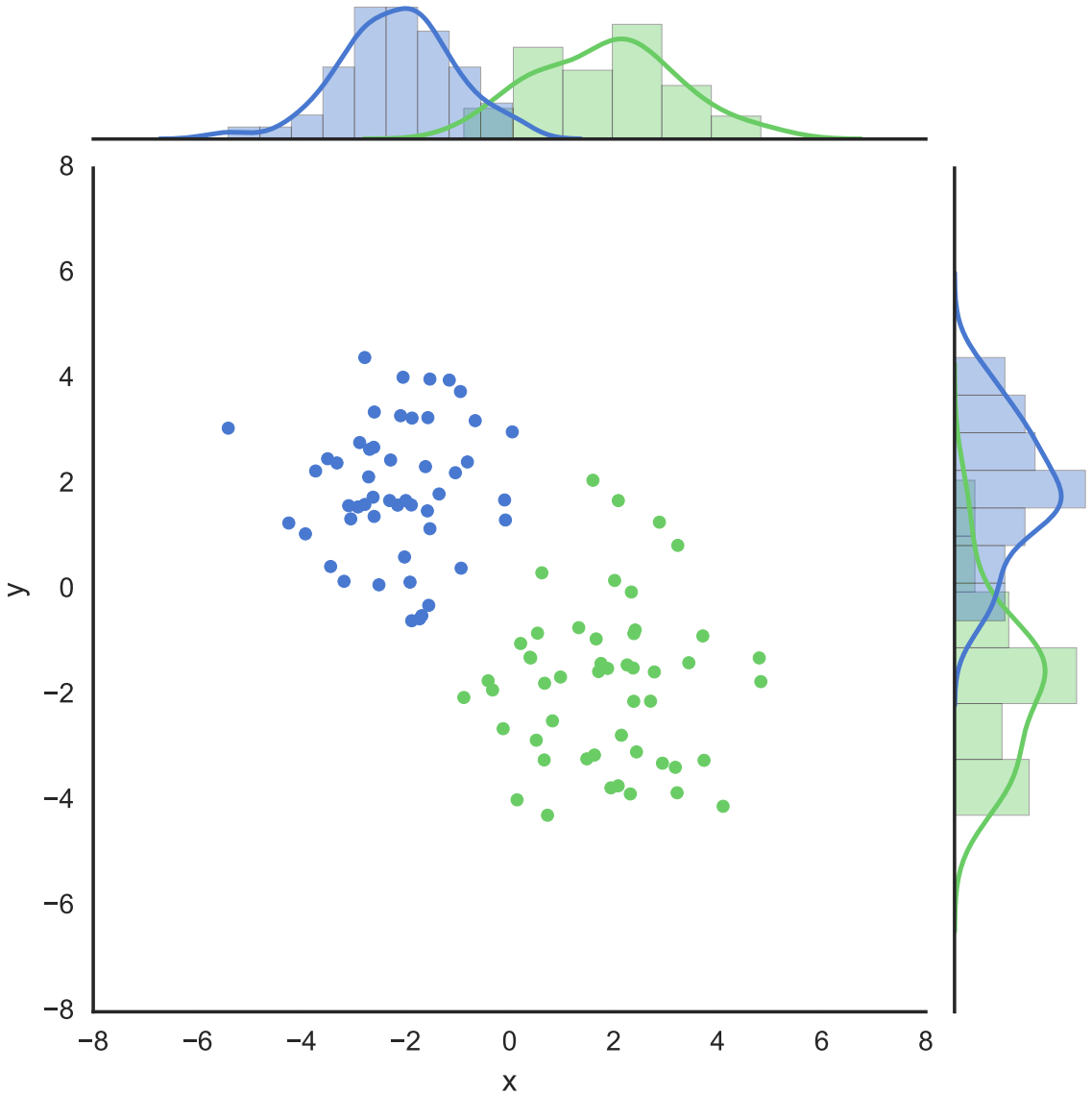
Classifying with Support Vector Machines

- in our experiments, we typically know what the categories are (i.e. conditions), but we need to find the prototypical geometry or "geography" (topography in the case of EEG)
- supervised-learning with support vector machines (SVM)

Partition the space up

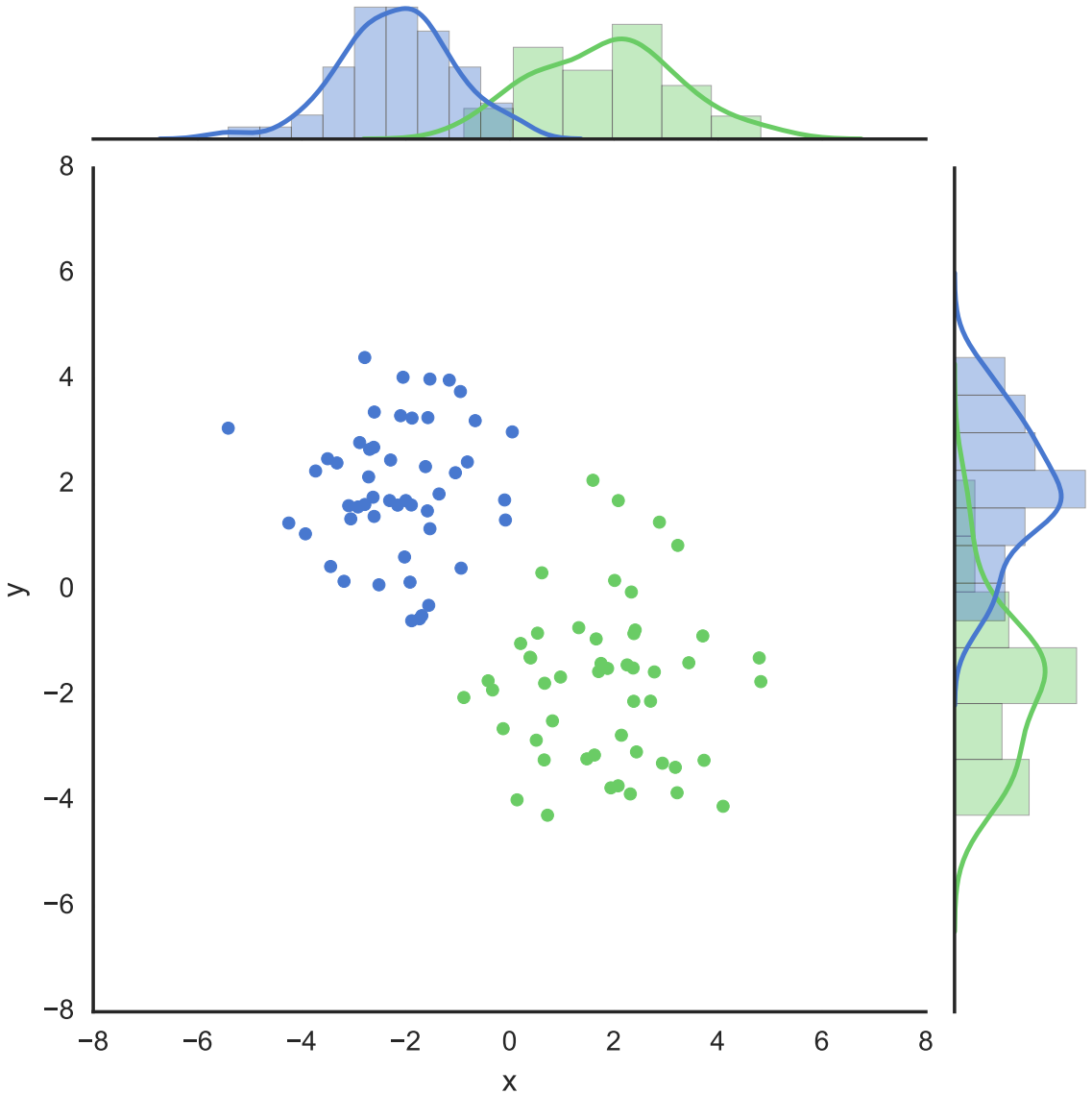


Partition the space up



- find a dividing line between groups

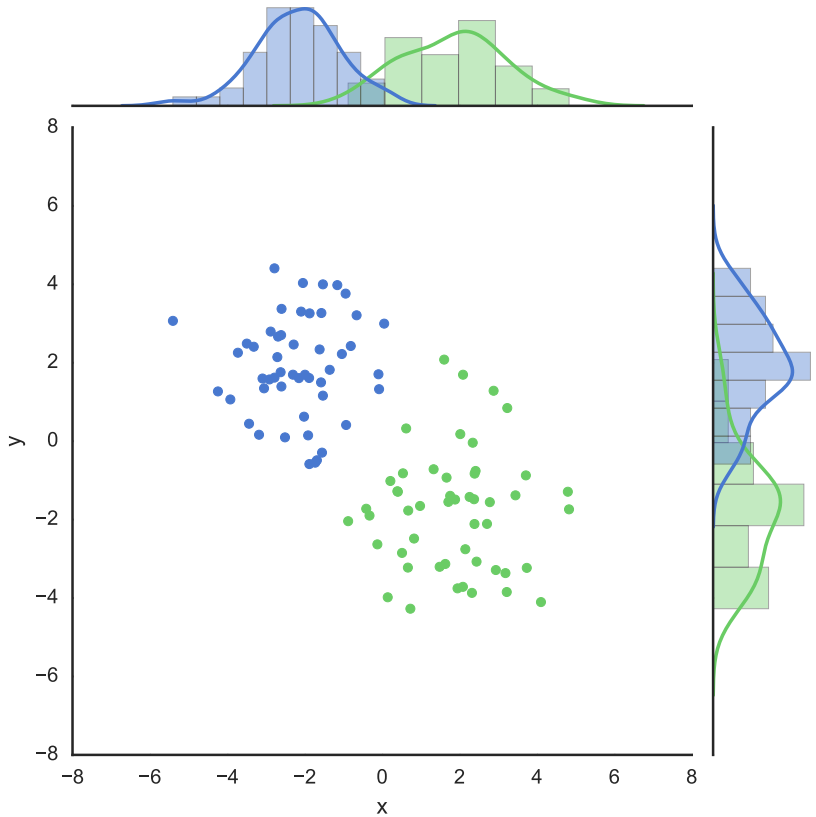
Partition the space up



- find a dividing line between groups
- (this example is somewhat trivial as the cluster centers are mirrored across $y = x$)

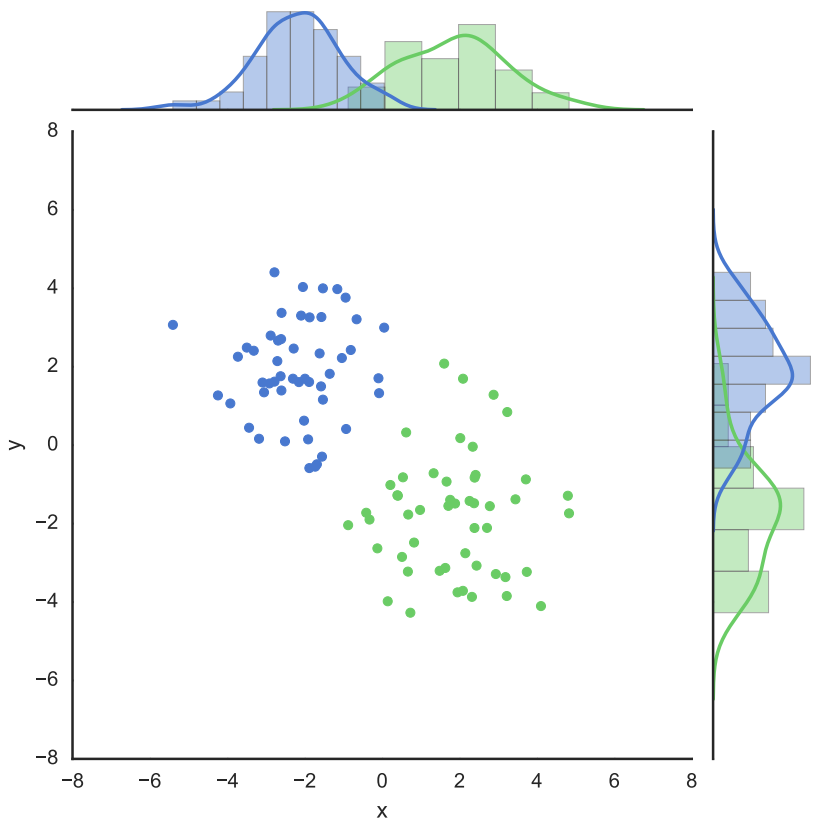
How could you do this with PCA?

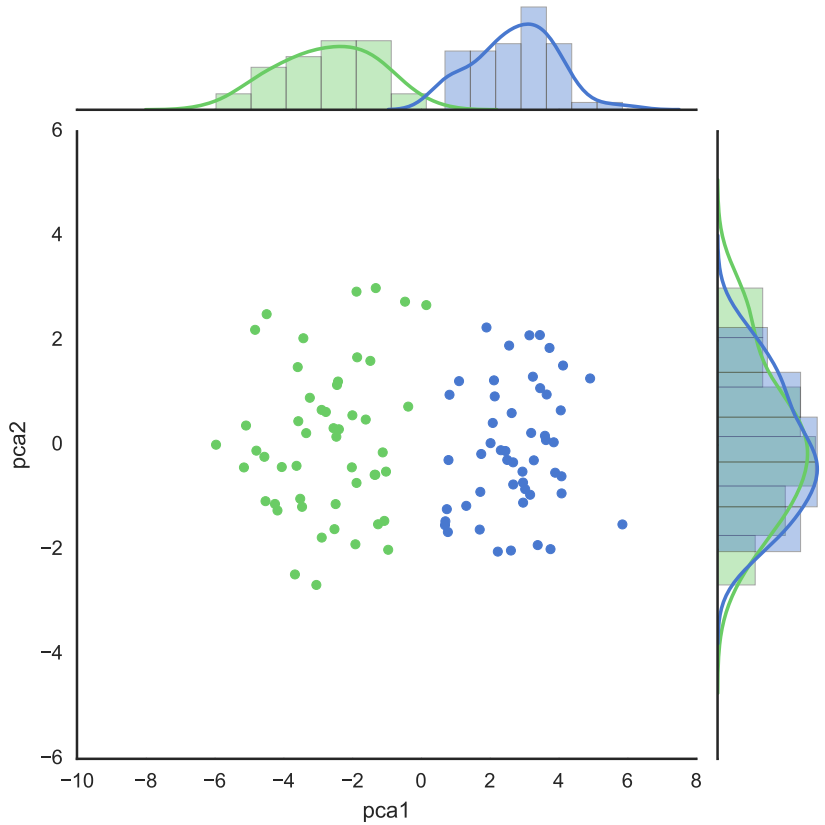
What will the components be?



How could you do this with PCA?

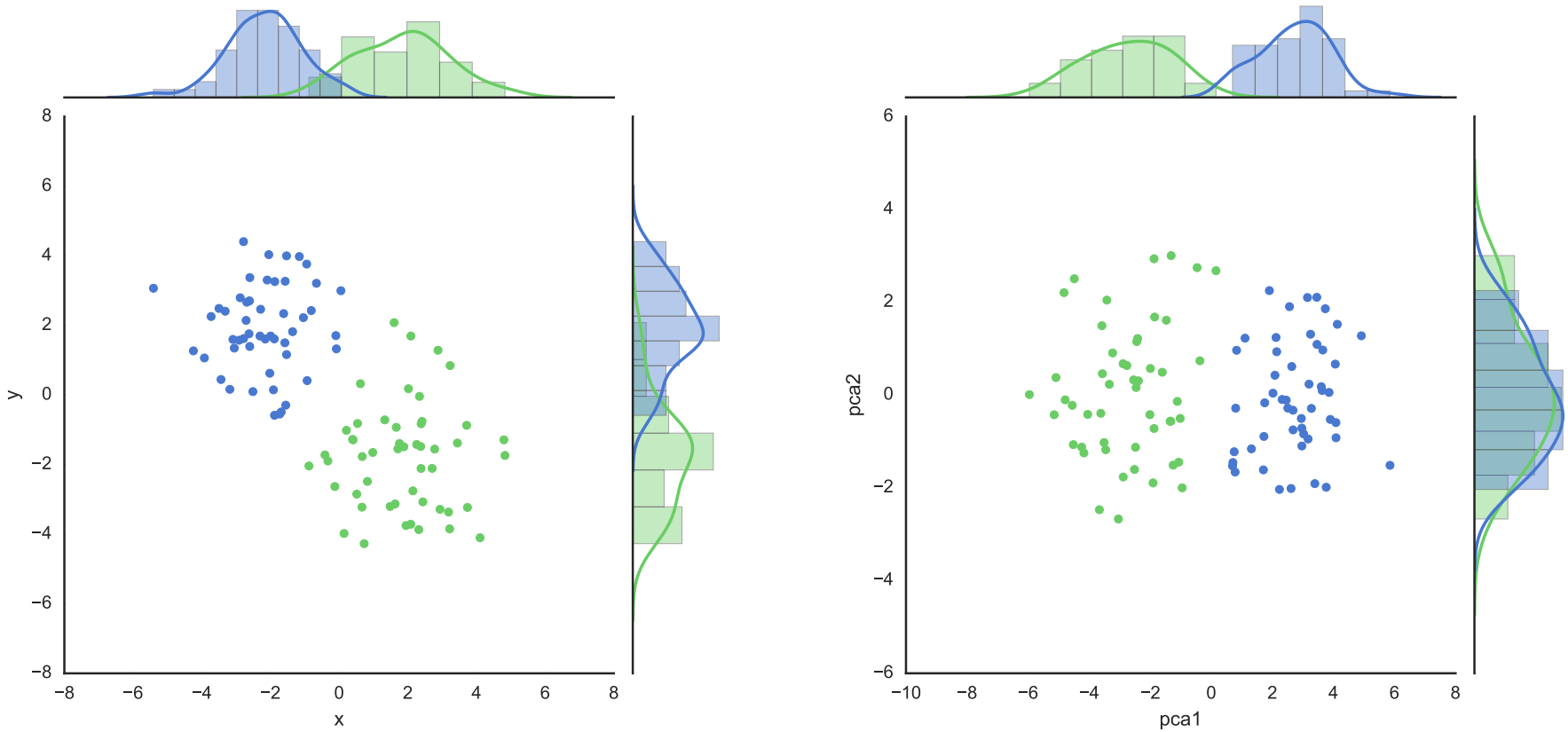
What will the components be?





How could you do this with PCA?

What will the components be?



This only works because we have two categories in two dimensions and thus the border is one-dimensional, i.e. the same size as a single component.

Support Vector Machines

- SVM works by constructing a separating hyperplane between groups
- SVM-based classification is inherently **binary**
- but it is possible to combine binary decisions to make complex decisions
  - one-vs-one: construct an SVM for every pairwise combination of group
  - one-vs-all: construct an SVM for every group
- The further away the separating hyperplane is from the nearest elements of each group, the better this works.

The theory behind this (or check your assumptions)

Theorem (Hahn-Banach)

There exists a hyperplane between any two disjoint, convex sets in Euclidean space.

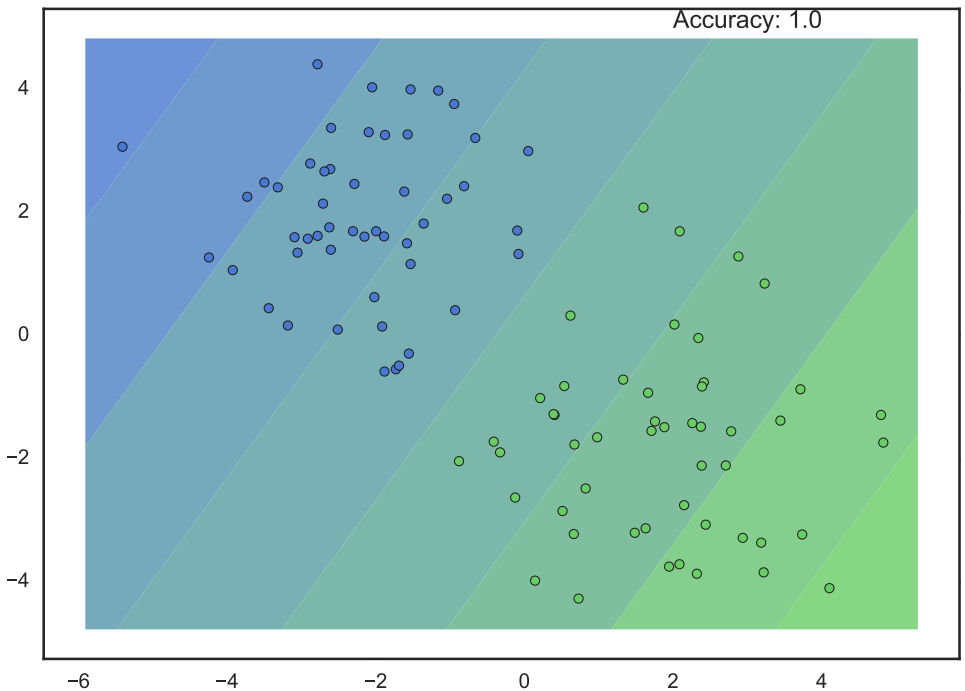
In other words, if the two groups do not overlap and are not intertwined / wrapped around each other, then we can construct a linear border between them.

Theorem

Every permutation can be written as the product of transpositions.

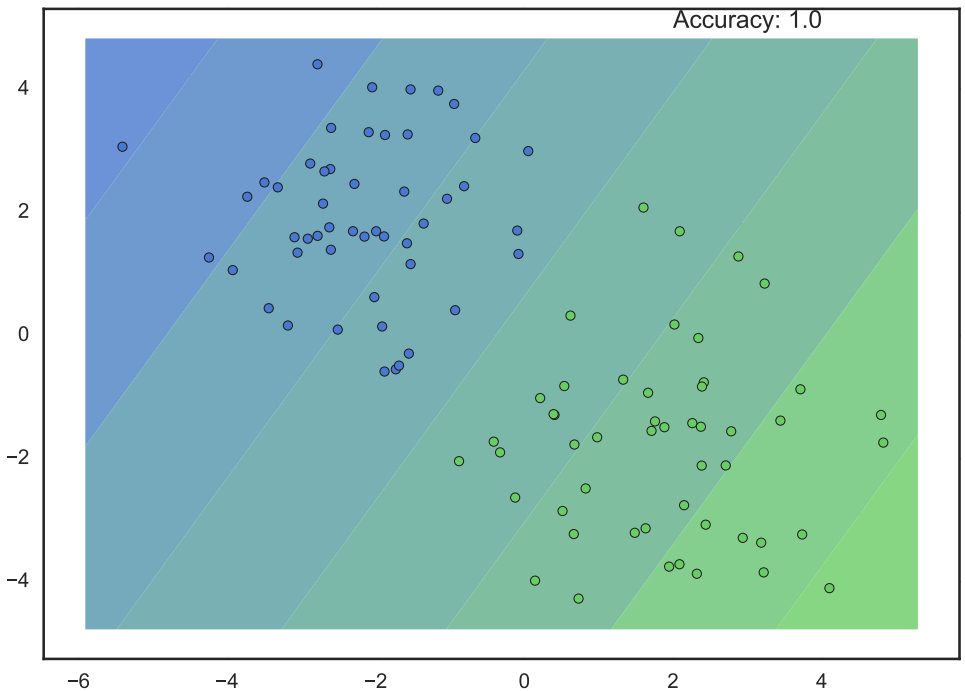
It is possible to express arbitrarily complex multiple choice questions as a series of yes-no questions.

Our easy example



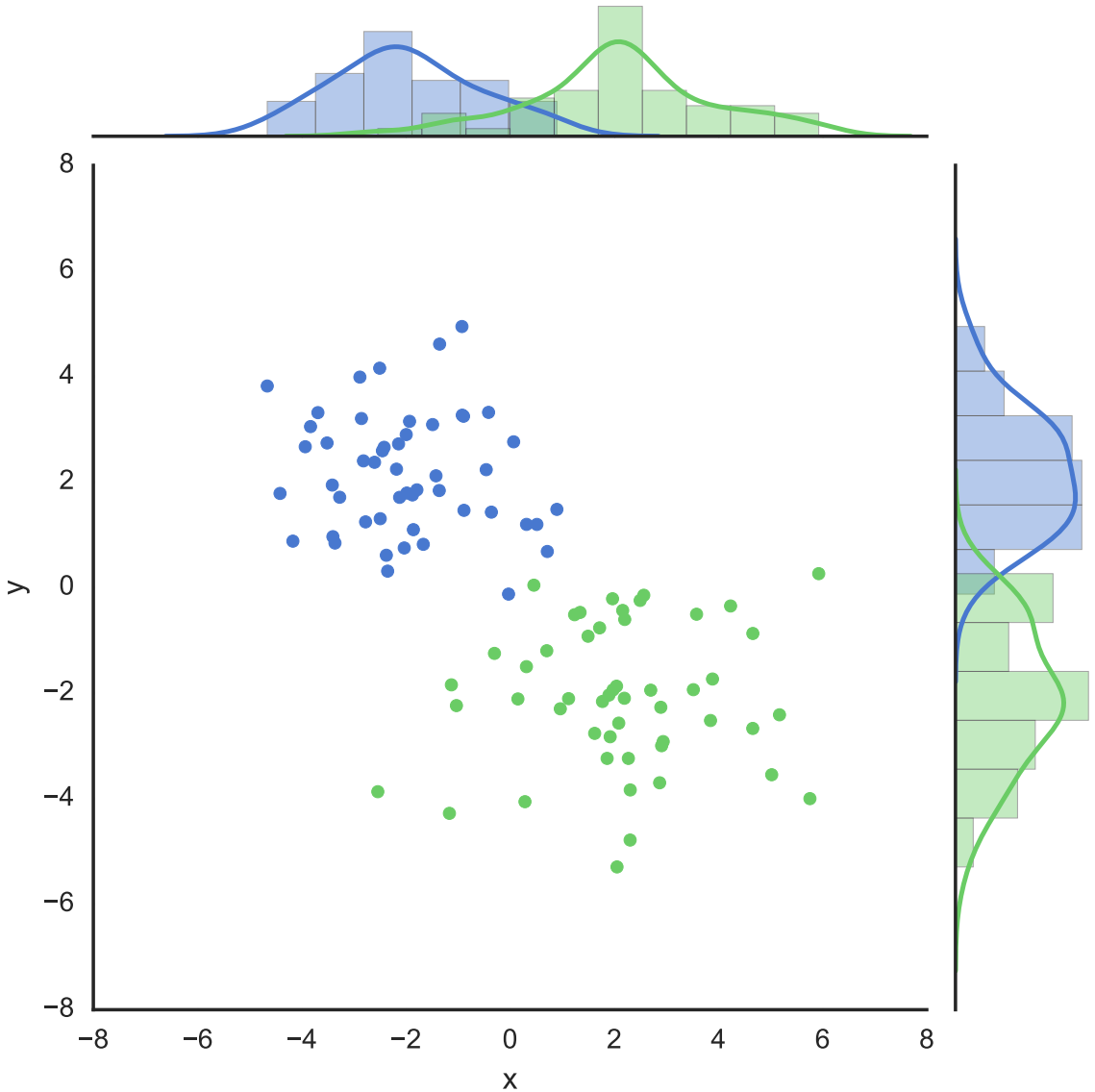
- no overlap
- circular blobs are convex

Our easy example

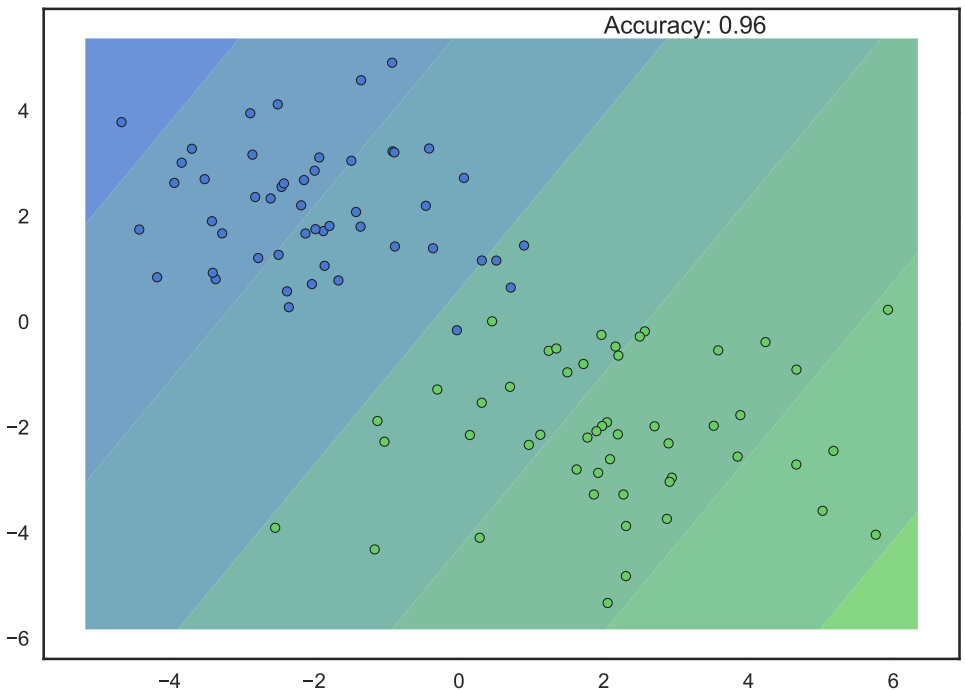


- no overlap
- circular blobs are convex
- perfect accuracy when applied back to itself

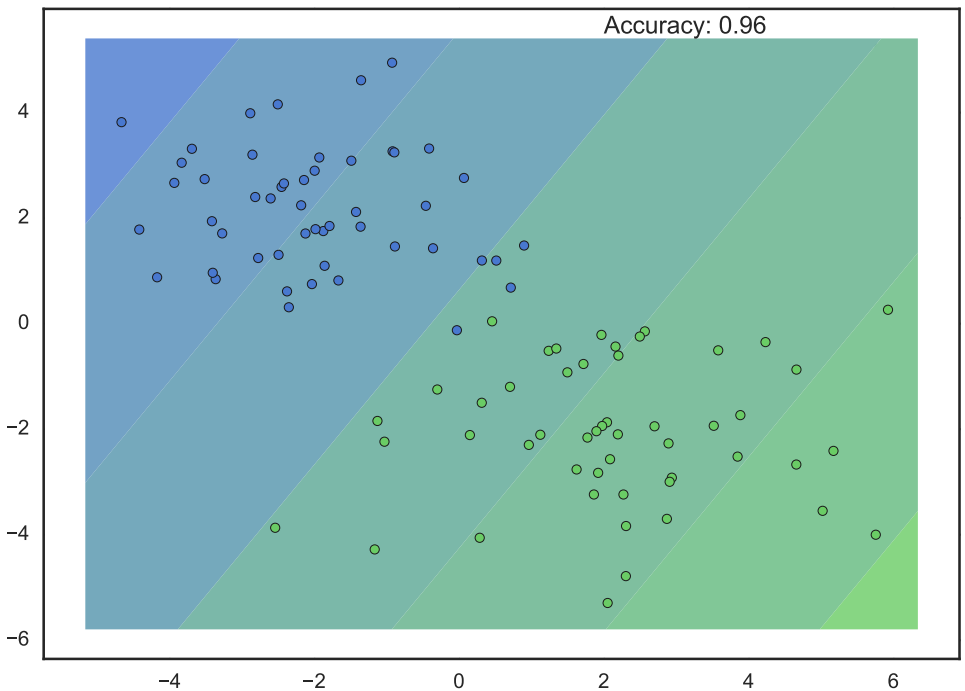
But does it have predictive power?



But does it have predictive power?



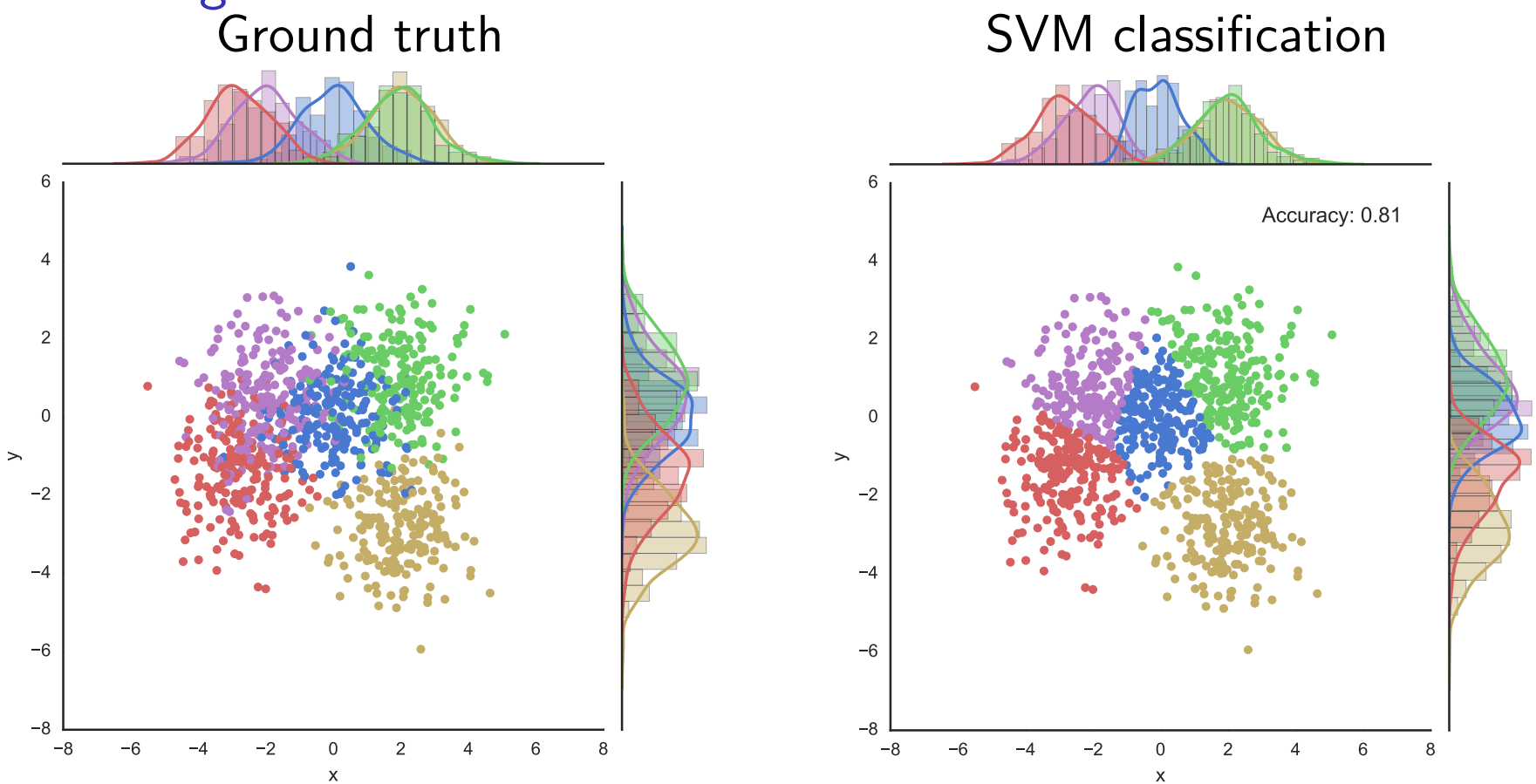
But does it have predictive power?



- small amount of overlap
- new data – "memorization" not possible
- high predictive power

A more complex example

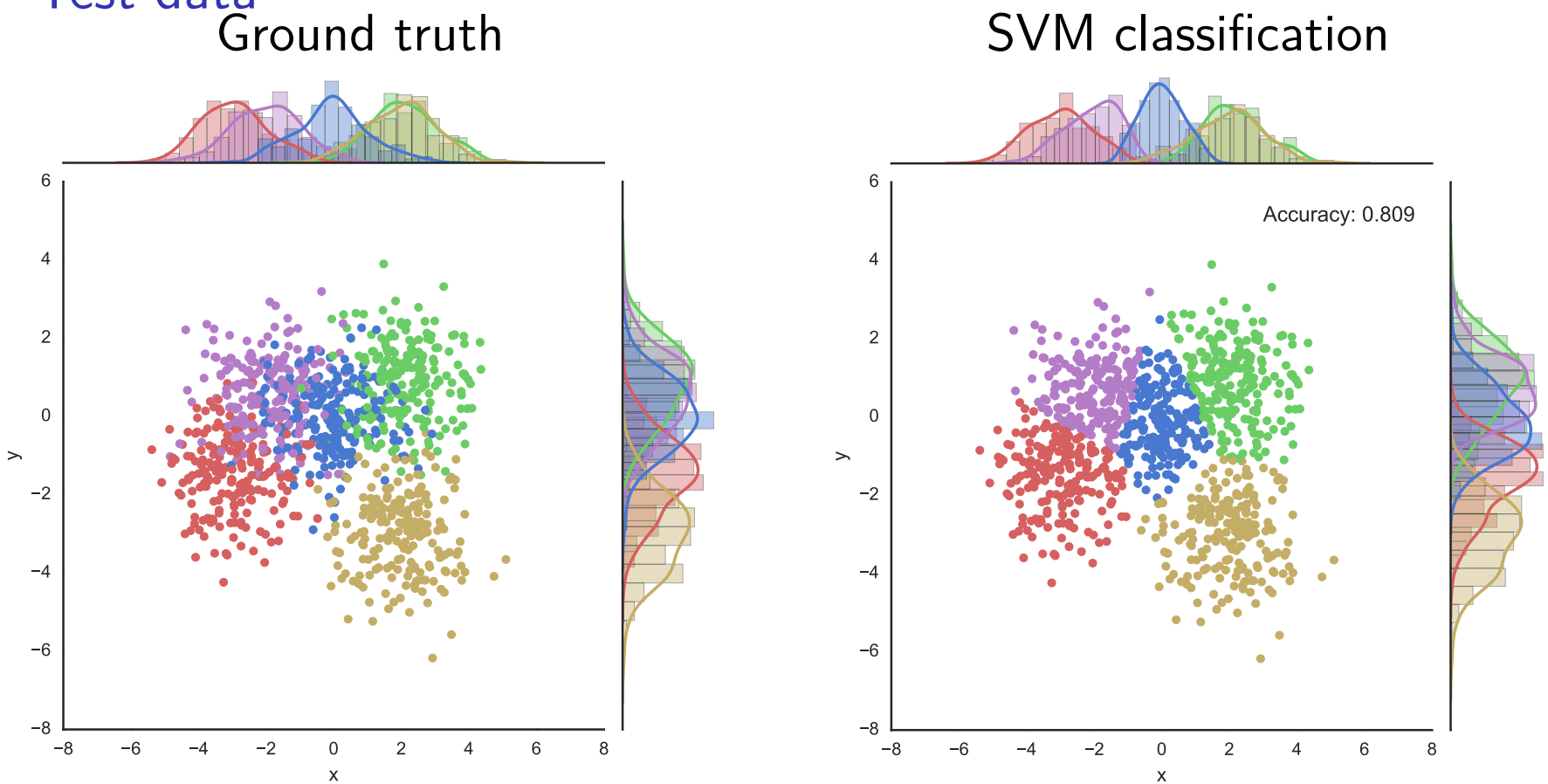
Training data



SVM is pretty good for such ideal data. Linear regression doesn't need the points to lie exactly on a line, and SVM doesn't need exactly separable data, but both do better the closer you are to meeting these assumptions.

A more complex example

Test data

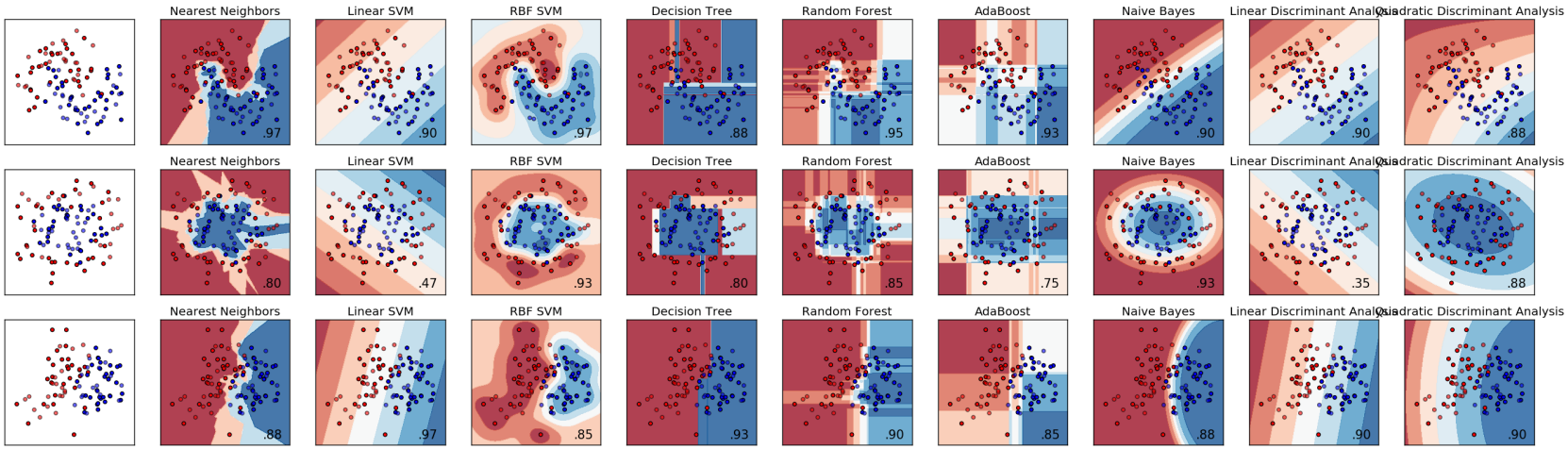


It's particularly important that we look at the predictive power and not just the model fit when dealing with high-parameter models. These data were constructed to be quite similar, so the predictive power is good, but the real-world is rarely so clean.

Cross-validation

- Training and testing on the same data is cheating – it hides "memorizing" the exam without understanding the material.
- Cross-validation allows you get around this with one dataset.
- Many different ways to do this, related to the jacknife in non-parametric statistics, such as
  - Leave-one-out (LOO)
  - *k*-fold validation

There is no ideal classifier



http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

References I

Breiman, Leo. 2001. "Statistical Modeling: The Two Cultures." *Statistical Science* 16 (3): 199–231.