Machine Learning

CARTs

Jeff Abrahamson

août 2019

Data Set for Quiz

2019-03-16	81682.0
2019-03-18	81720.0
2019-03-20	81760.0
2019-03-24	81826.0
2019-03-25	81844.0
2019-03-26	81864.0
2019-03-27	81881.0
2019-03-28	81900.0
2019-03-30	81933.0
2019-04-03	82003.0

Review

What is Machine Learning?

Learning is what we do when we can't explain how.

- Supervised
- Unsupervised
- Reinforcement

Lots of maths

We'll try to ignore it, but it's there...

- Vector spaces and linear algebra
- Probability
- Statistics
- Optimisation theory
- Differential calculus

The curse of dimensionality.

Data Science

1. Define the question of interest
2. Get the data
3. Clean the data
4. Explore the data
5. Fit statistical models
6. Communicate the results
7. Make your analysis reproducible

Observational vs experimental

Data

Anecdote: it doesn’t accumulate to be data.

Data

High bias, low variance

Low bias, high variance

High bias, high variance

Low bias, low variance

Features

Feature Engineering

Data

One of K = one-hot encoding

Outliers: don’t ignore them!

Feature Engineering

1. Brainstorm
2. Pick some
3. Make them
4. Evaluate
5. Repeat

Easy Features

Text

bag of words

Easy Features

Images

corners, edges, point matching

Easy Features

We'll see more

Linear Regression

Problem: $\{(x_i, y_i)\}$.

Given x, predict $\hat{y}$.

Here y is continuous.

Linear Regression

x: **explanatory** or **predictor** variable.

y: **response** variable.

For some reason, we believe a linear model is a good idea.

Residuals

What's left over.

$$\text{data} = \text{fit} + \text{residual}$$

Residuals

What's left over.

$$y_i = \hat{y}_i + e_i$$

Residuals

What's left over. Goal: small residuals.

$$\sum e_i^2$$

Logistic regression

- Binary output
- Classification

Logistic regression

- Have: continuous and discrete inputs
- Want: class (0 or 1)

Logistic regression

Logistic (sigmoid, logit) function

$$g(z) = \frac{1}{1 + e^{-z}}$$

One vs Rest, One vs One

What I described yesterday:

- OvR (OvA): compute k classifiers
- OvO: compute $k(k-1)/2$ classifiers

The missing point: the classifiers give scores, not just in/out answers.

One vs Rest, One vs One

One vs Rest:

Accept the judgement of the classifier with the highest score.

One vs Rest, One vs One

One vs One:

Classifiers vote. Accept the class that gets the most votes. Advantage: Reduces multi-class classification to single-class classification.

Disadvantage: Classifier scores aren't necessarily comparable. For example, classes may have very different numbers of members.

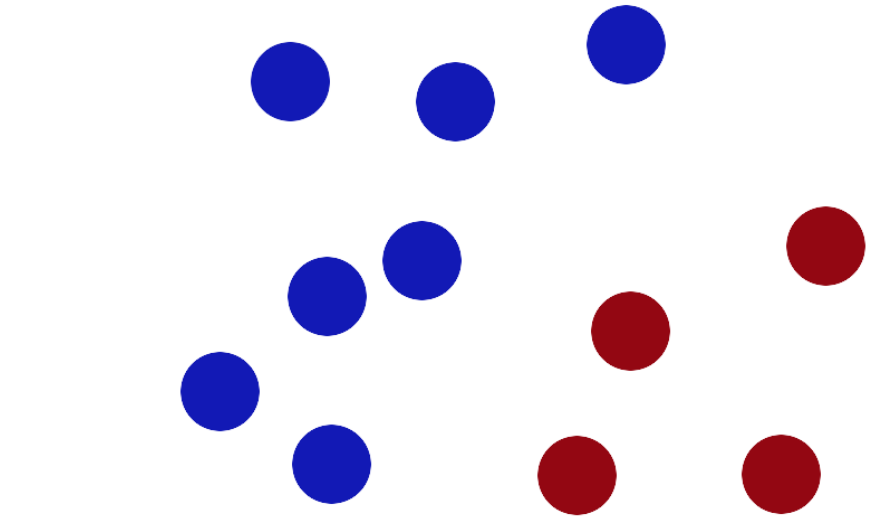
Hyperparameters

- The word hyperparameter is not well-defined.
- In most contexts, it is the parameters of the underlying distribution
- In training, we learn the parameters of the model
- We choose the hyperparameters to govern the training
- So we may want to experiment to learn the distribution parameters that best optimise our learned model's performance

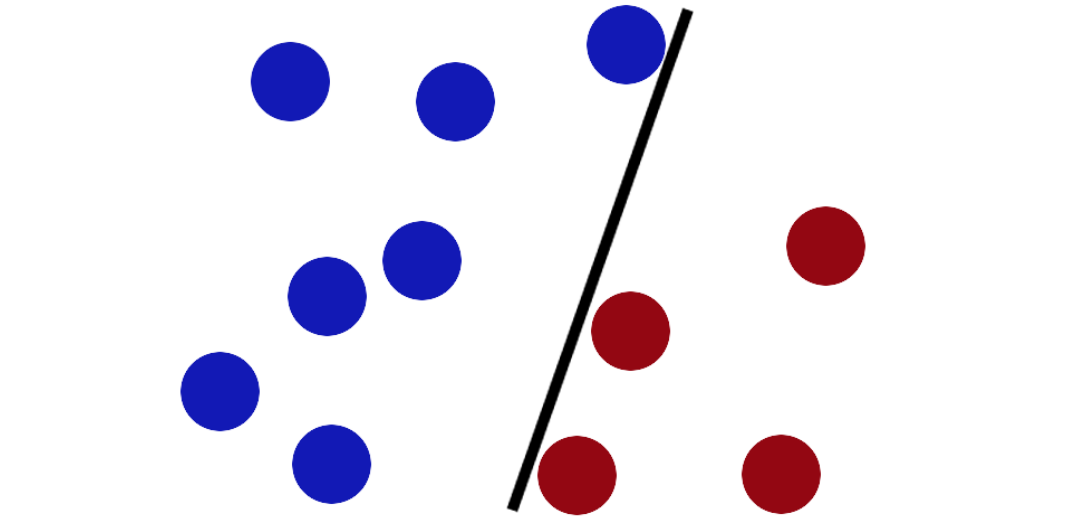
Testing

- Set aside (partition) data for testing (e.g., 70% / 30%)
- Learn on training set, test on testing set
- When searching hyperparameters, set aside again (e.g., 60% / 20% / 20%)

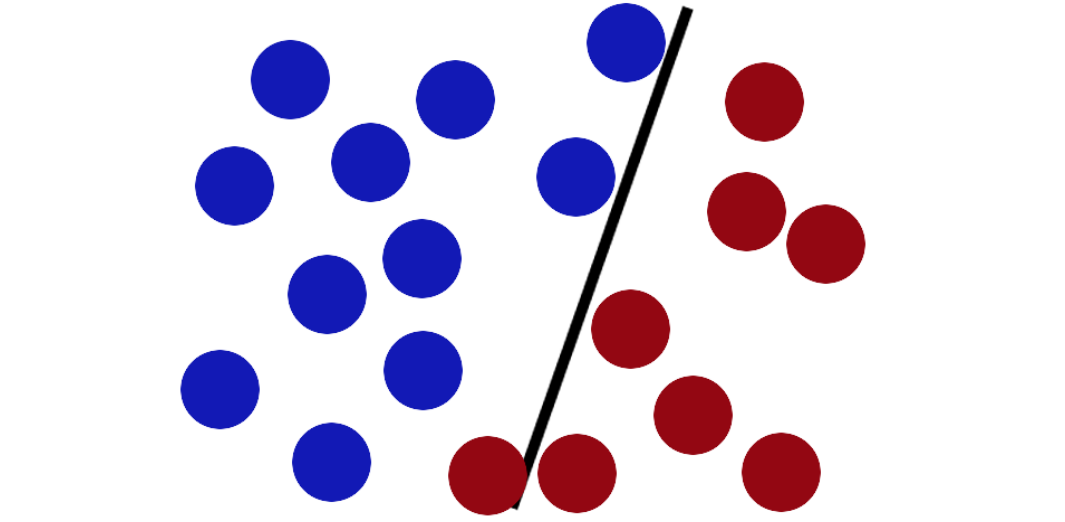
The simple explanation



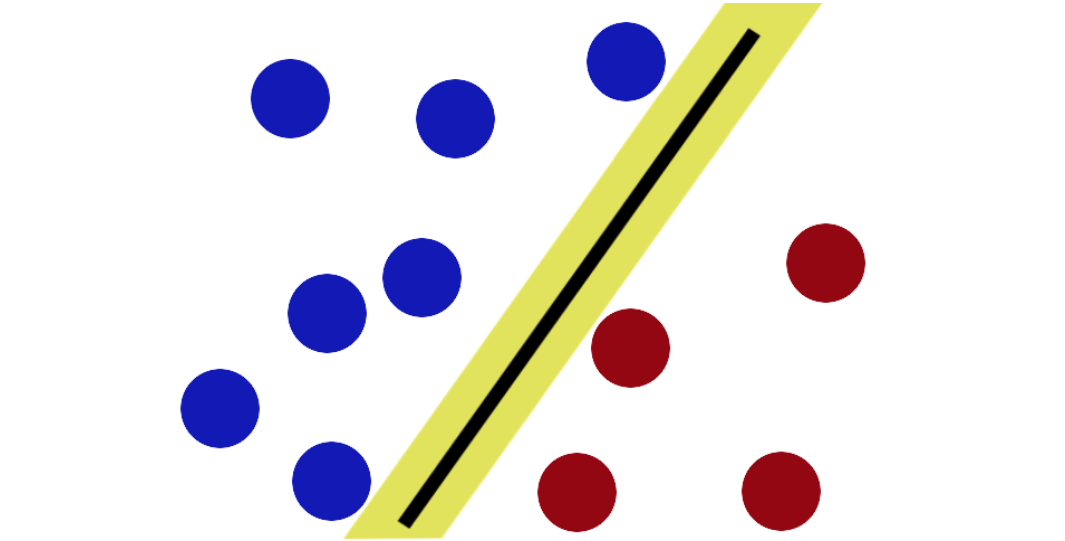
The simple explanation



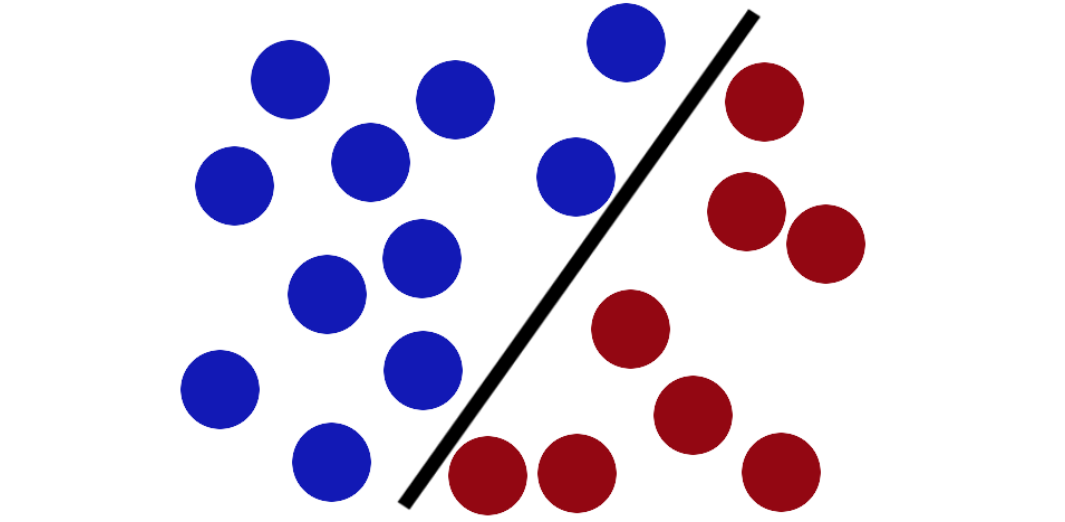
The simple explanation



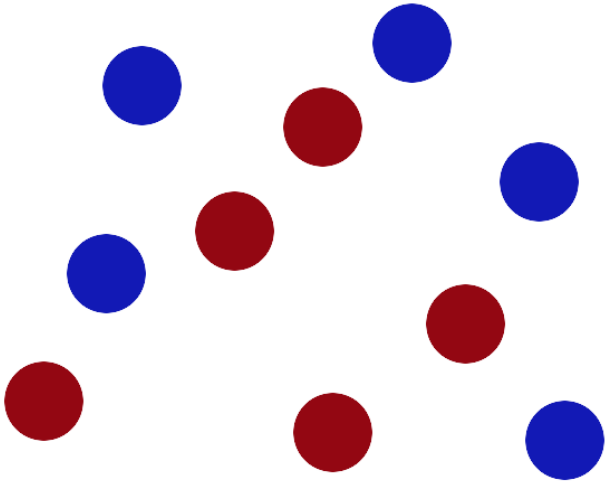
The simple explanation



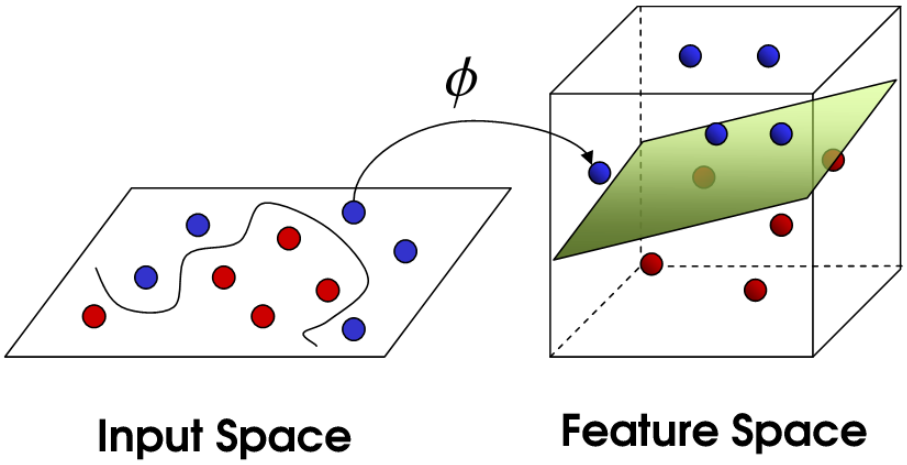
The simple explanation



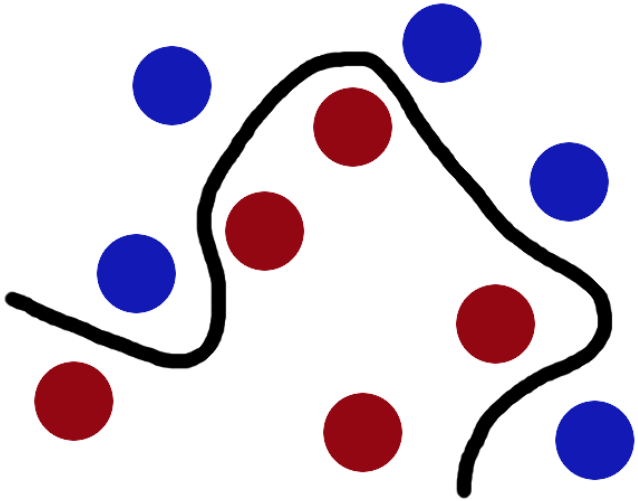
The simple explanation



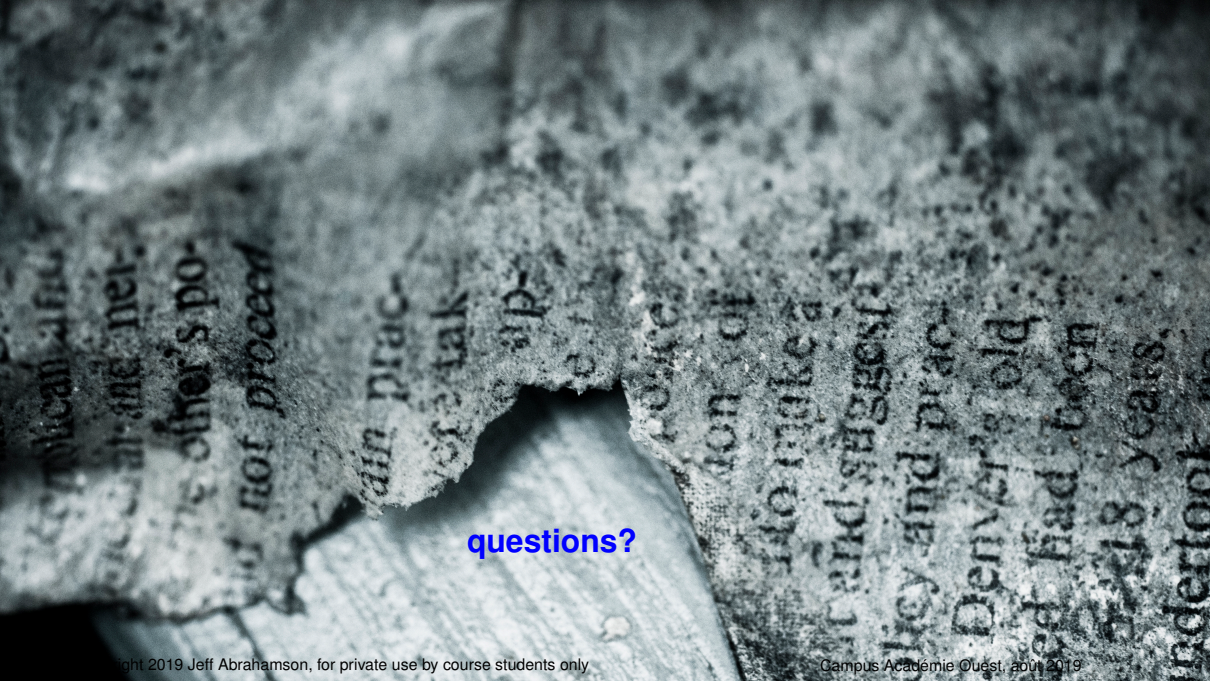
The simple explanation



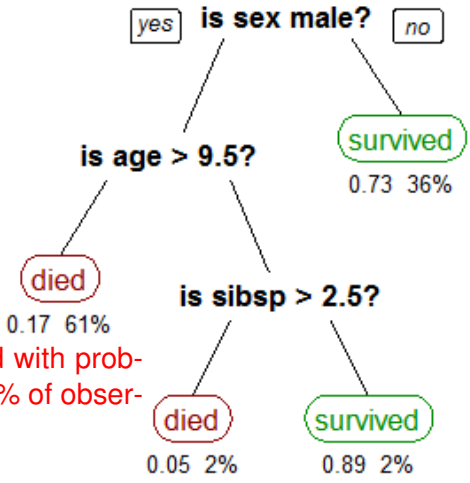
The simple explanation



video time



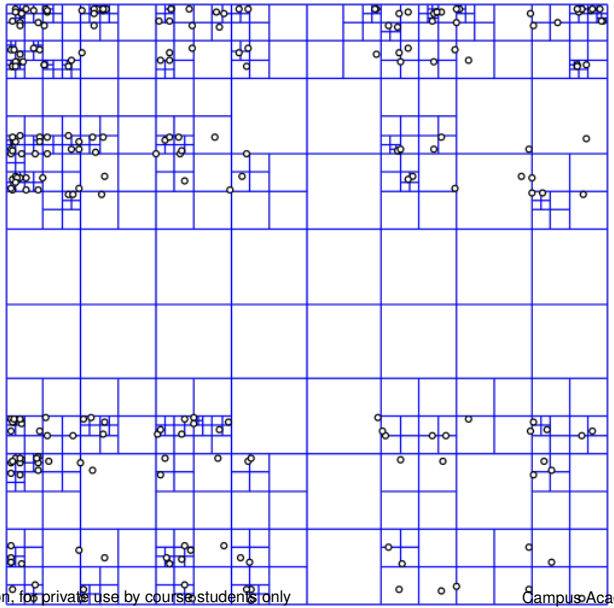
Decision Trees



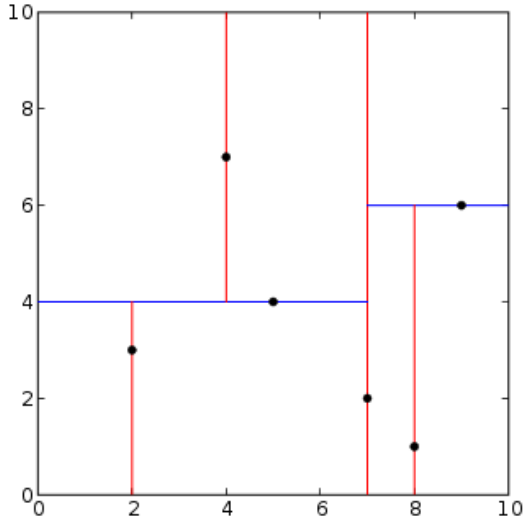
E.g., passengers died with probability .17 which is 61% of observations

Stephen Milborrow

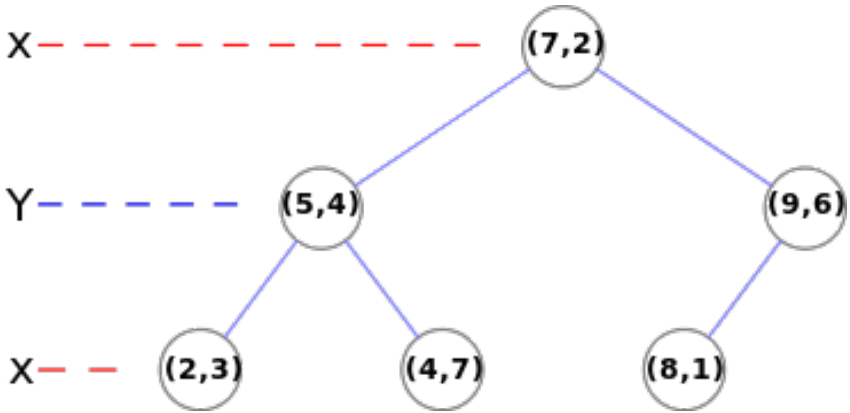
Quadtree



kd-tree



kd-tree



kd-tree animation

Decision Trees

Variations

- Classification tree
- Regression tree

CART = classification and regression trees

Decision Trees

Ensemble methods

- Bagging
- Random forest
- Boosted trees *(gradient boosted trees)*
- Rotation forest

Boostrap aggregating = bagging

Boostrap

A family of statistical methods using sampling with replacement.

(Also: an example of an ensemble method.)

Boostrap aggregating = bagging

- Increase stability
- Increase accuracy
- Reduce variance
- Avoid overfitting

A type of model averaging (ensemble method).

Boostrap aggregating = bagging

- Training set D of size n
- Sample D *with replacement* to create $D_1, \ldots, D_k$ of size n'
- If $n = n'$, expect $1 - 1/e \approx 63.2\%$ repeats

Boostrap aggregating = bagging

- Training set D of size n
- Sample D *with replacement* to create $D_1, \ldots, D_k$ of size n'
- If $n = n'$, expect $1 - 1/e \approx 63.2\%$ repeats

- Train k models
- Average (regression) or vote (classification)

Boostrap aggregating = bagging

Do not confuse with

- Boosting (and AdaBoost)
- Boostrap (statistics)
- Cross validation

Random subspace method

attribute bagging = feature bagging

Random subspace method

Bagging (bootstrap aggregation) = resampling to create more data sets, train models on different samples

Attribute bagging = project to create more data sets, train models on different samples

Random forests

Combine bagging with random subspace method

scikit-learn

```
from sklearn.ensemble import RandomForestClassifier

X = [[0, 0], [1, 1]]
Y = [0, 1]
clf = RandomForestClassifier(n_estimators=10)
clf = clf.fit(X, Y)
clf.predict([[.6, .6]])
```

scikit-learn

```
from sklearn.ensemble import RandomForestClassifier

X = [[0, 0], [1, 1]]
Y = [0, 1]
clf = RandomForestClassifier(n_estimators=10)
clf = clf.fit(X, Y)
clf.predict([[.6, .6]])
```

`X` has size `[n_samples, n_features]`

`Y` has size `[n_samples]`

scikit-learn

```
from sklearn.ensemble import RandomForestRegressor

X = [[0, 0], [1, 1]]
Y = [0, 1]
clf = RandomForestRegressor(n_estimators=10)
clf = clf.fit(X, Y)
clf.predict([[.6, .6]])
```

scikit-learn

```
from sklearn.ensemble import RandomForestRegressor

X = [[0, 0], [1, 1]]
Y = [0, 1]
clf = RandomForestRegressor(n_estimators=10)
clf = clf.fit(X, Y)
clf.predict([[.6, .6]])
```

`X` has size `[n_samples, n_features]`

`Y` has size `[n_samples]`