Curtin University

THE CURTIN INSTITUTE FOR COMPUTATION

Introduction to Machine learning

27 August 2018 workshop

Kevin Chai
Rebecca Lange

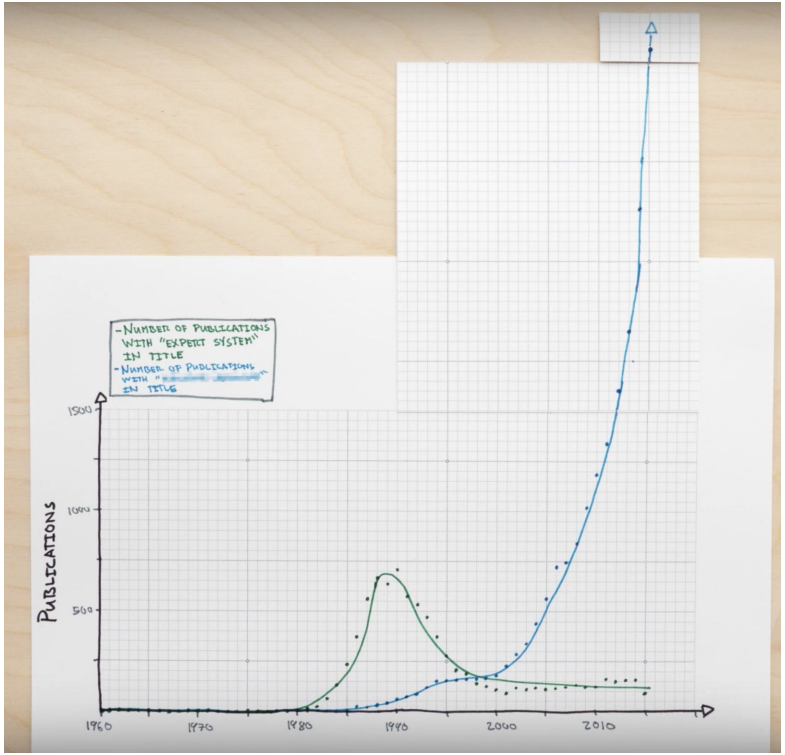
Agenda

- Introduction
- Workshop
  - Classification
  - Regression
  - Clustering
  - Dimensionality reduction
- Reference sheets
- Deep learning
- Q&A

Machine Learning

- Algorithms that learn from examples and experiences (data)

- Performs better than hard coded rules for complex problems



Source: Welch Labs, Learning to See - Part 3, video series 2016

arXiv.org

5,500+ new papers in AI and computer vision in 2017

GitHub

machine learning

90,842 repository results

Examples

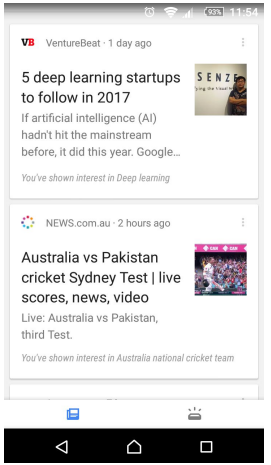

Google
Australia

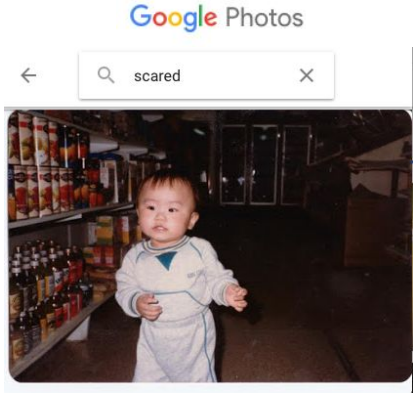

"Hey Siri, what's the best sushi place in town?"


NETFLIX

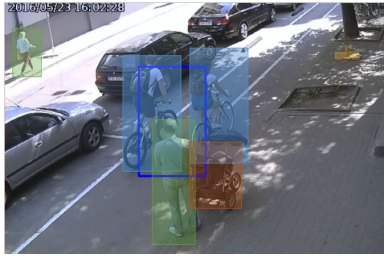

Other Movies You Might Enjoy
2001: A Space Odyssey has been added to your Queue at position 339.
This movie is available now.
Move To Top Of My Queue
Continue Browsing
Visit your Queue


Google
Translate


VentureBeat · 1 day ago
5 deep learning startups to follow in 2017
If artificial intelligence (AI) hadn't hit the mainstream before, it did this year. Google...
You've shown interest in Deep learning
NEWS.com.au · 2 hours ago
Australia vs Pakistan cricket Sydney Test | live scores, news, video
Live: Australia vs Pakistan, third Test.
You've shown interest in Australia national cricket team


Google Photos
scared

Examples



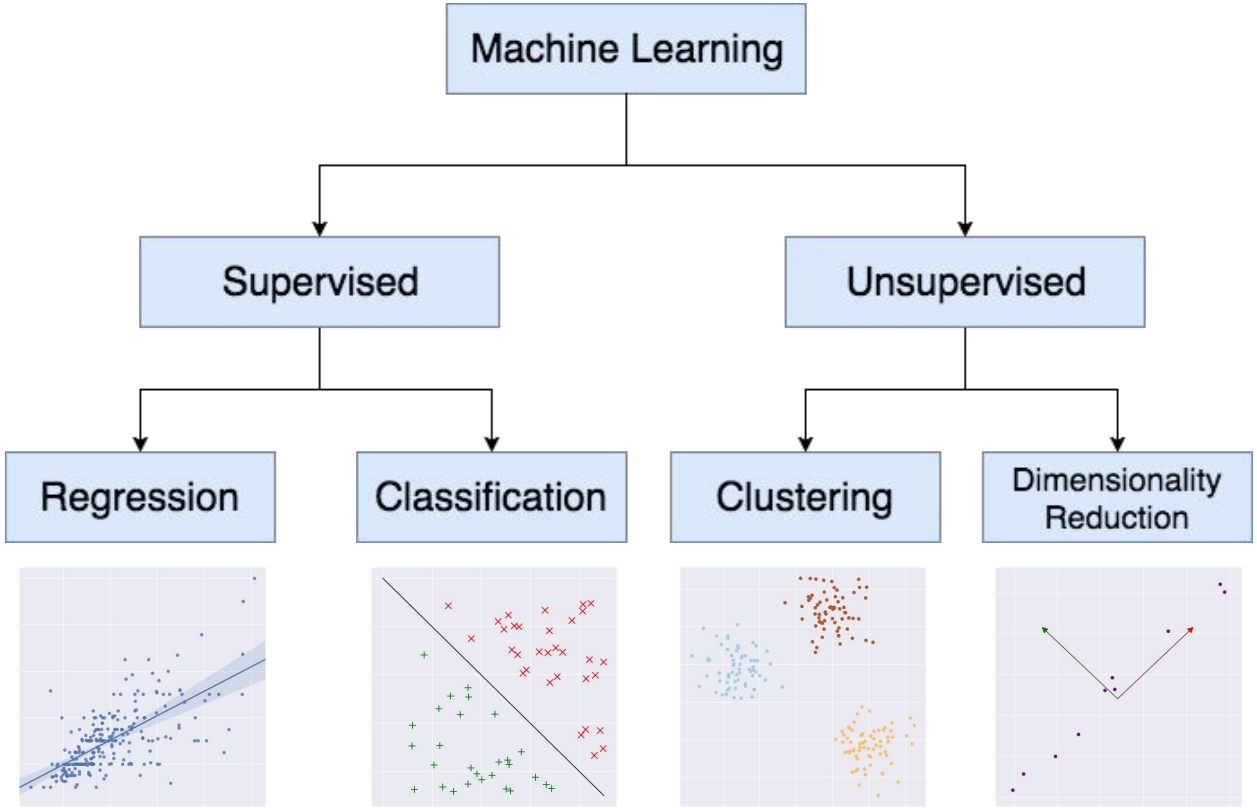

Safety belt
not fastened


Phone
calling


Long Dress
Polo-Shirt
Robe
Short Dress
T-Shirt
Blouse

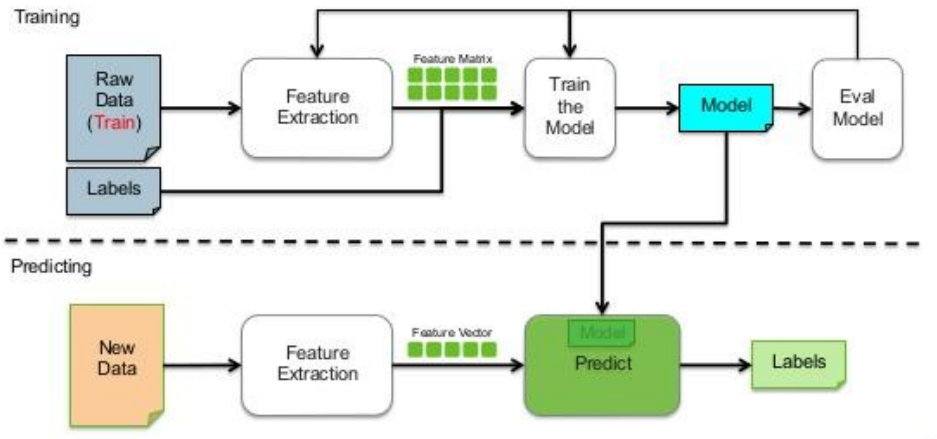

Challenge Match
8 - 15 March 2016
AlphaGo
Lee Sedol

Algorithms


Machine Learning
Supervised
Unsupervised
Regression
Classification
Clustering
Dimensionality Reduction

Supervised learning

- Trained with labelled data
- Algorithms learn in a supervised manner



Decision trees

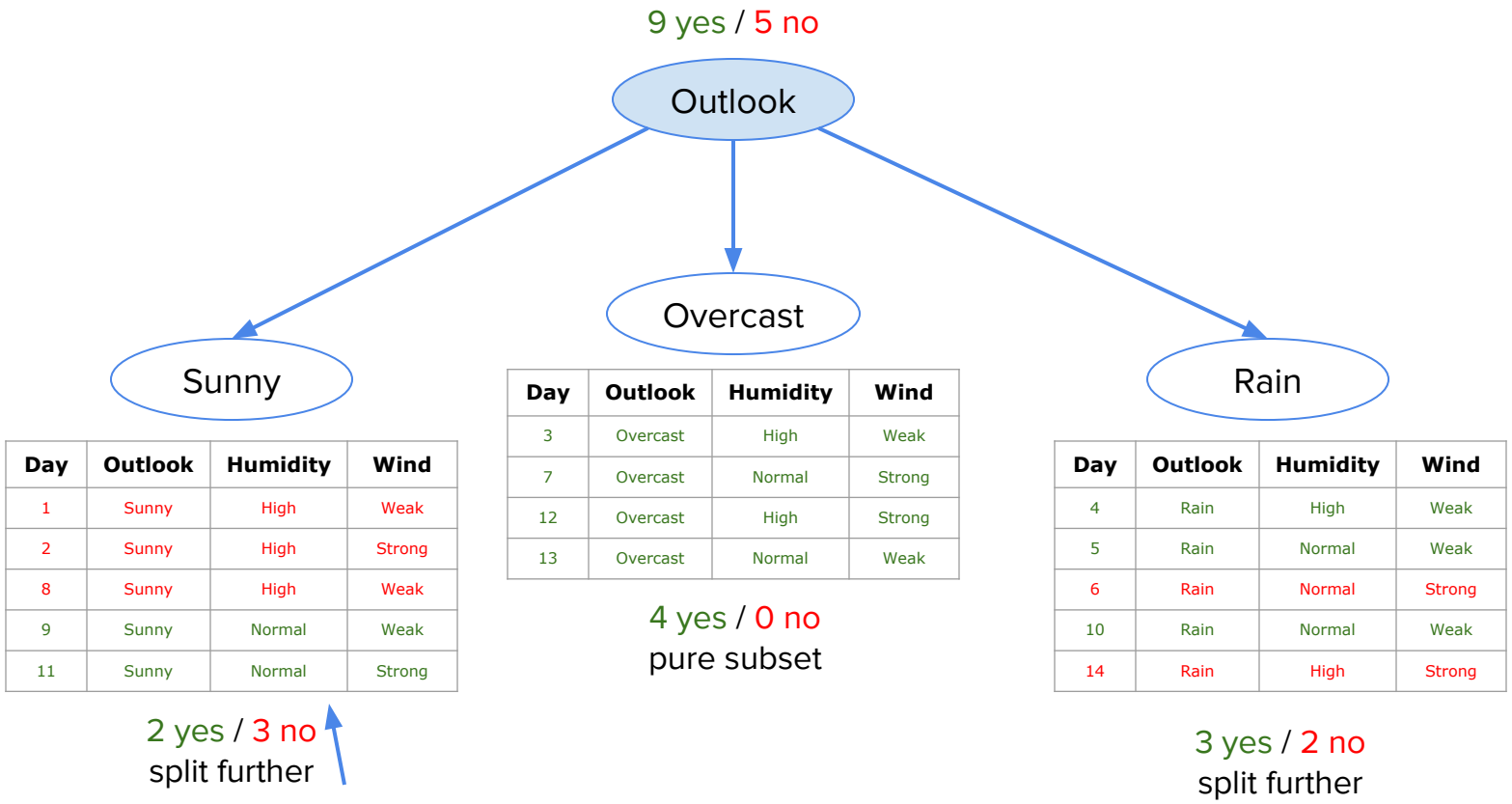
- Classification algorithm
  - Predict discrete (category) outputs
  - Will Shiv play cricket?
- Training
  - Recursively split data into subsets based on a single attribute
  - Stop when all subsets are pure (all yes / no)
- Prediction
  - Based on subset where new data is placed

Training data: 9 yes / 5 no

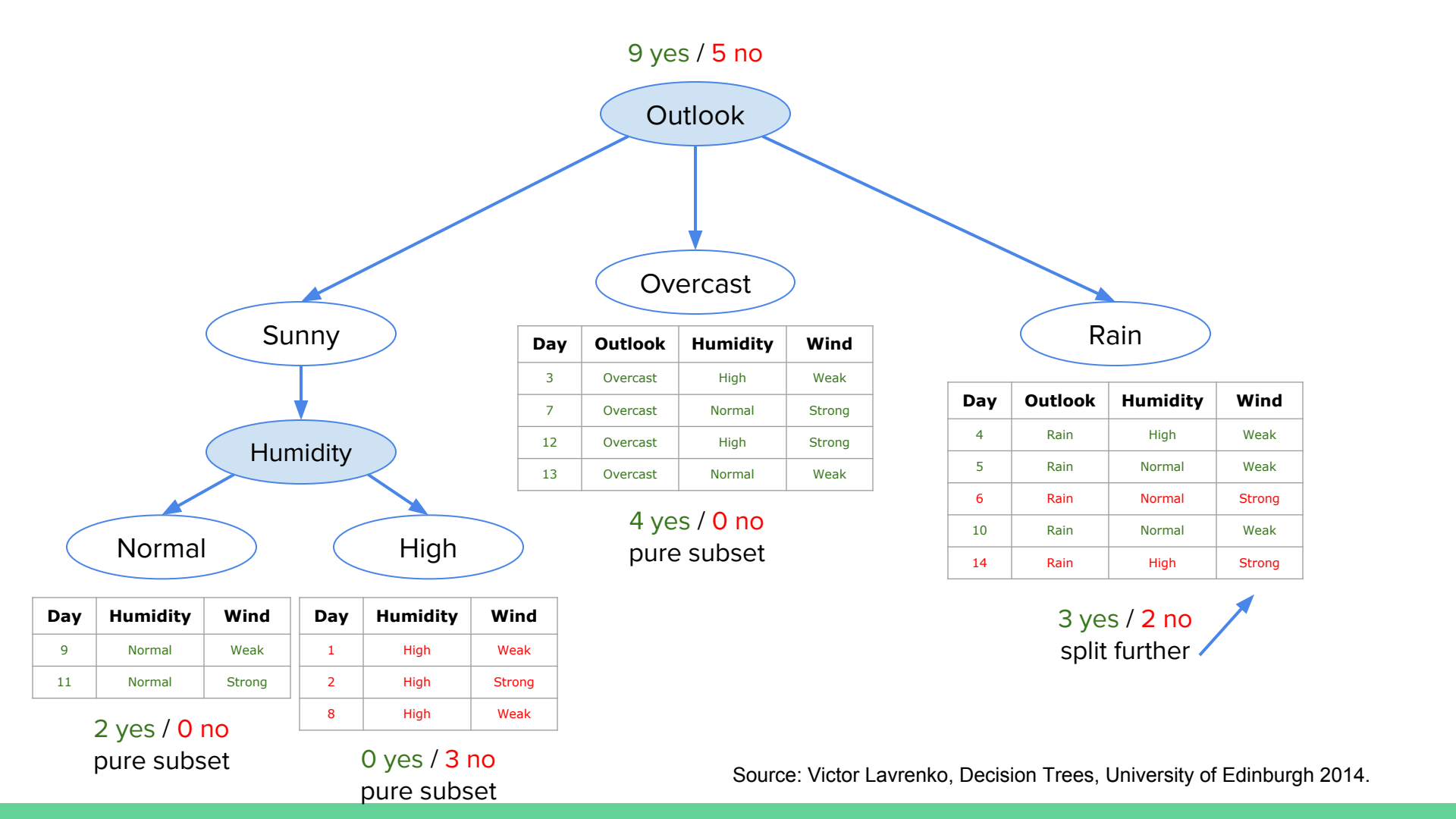
Day	Outlook	Humidity	Wind	Cricket
1	Sunny	High	Weak	No
2	Sunny	High	Strong	No
3	Overcast	High	Weak	Yes
4	Rain	High	Weak	Yes
5	Rain	Normal	Weak	Yes
6	Rain	Normal	Strong	No
7	Overcast	Normal	Strong	Yes
8	Sunny	High	Weak	No
9	Sunny	Normal	Weak	Yes
10	Rain	Normal	Weak	Yes
11	Sunny	Normal	Strong	Yes
12	Overcast	High	Strong	Yes
13	Overcast	Normal	Weak	Yes
14	Rain	High	Strong	No

New data:

15	Rain	High	Weak	?



Source: Victor Lavrenko, Decision Trees, University of Edinburgh 2014.



Day	Outlook	Humidity	Wind
3	Overcast	High	Weak
7	Overcast	Normal	Strong
12	Overcast	High	Strong
13	Overcast	Normal	Weak

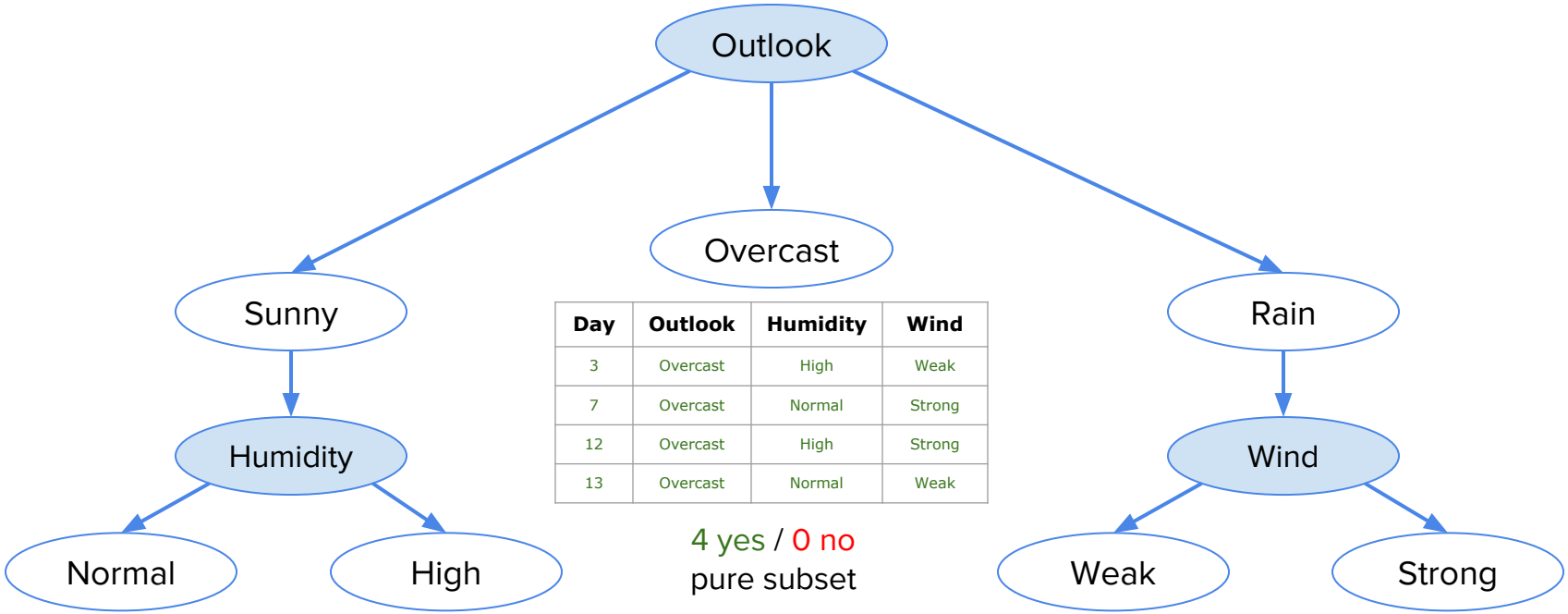
Day	Outlook	Humidity	Wind
4	Rain	High	Weak
5	Rain	Normal	Weak
6	Rain	Normal	Strong
10	Rain	Normal	Weak
14	Rain	High	Strong

Day	Humidity	Wind
9	Normal	Weak
11	Normal	Strong

Day	Humidity	Wind
1	High	Weak
2	High	Strong
8	High	Weak

Source: Victor Lavrenko, Decision Trees, University of Edinburgh 2014.

9 yes / 5 no



Day	Outlook	Humidity	Wind
3	Overcast	High	Weak
7	Overcast	Normal	Strong
12	Overcast	High	Strong
13	Overcast	Normal	Weak

4 yes / 0 no
pure subset

Day	Humidity	Wind
9	Normal	Weak
11	Normal	Strong

2 yes / 0 no
pure subset

Day	Humidity	Wind
1	High	Weak
2	High	Strong
8	High	Weak

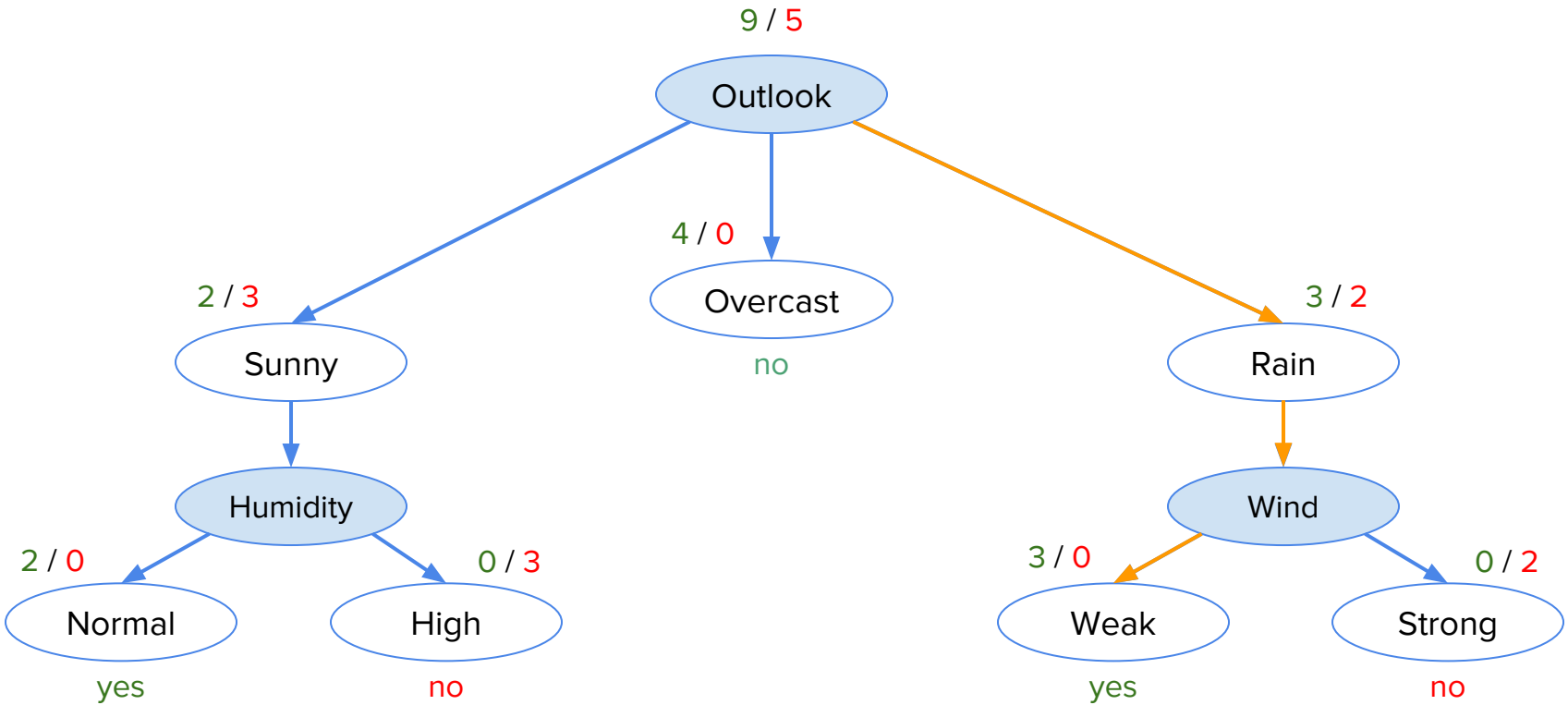
0 yes / 3 no
pure subset

Day	Outlook	Humidity
4	Rain	High
5	Rain	Normal
10	Rain	Normal

3 yes / 0 no
pure subset

Day	Outlook	Humidity
6	Rain	Normal
14	Rain	High

0 yes / 2 no
pure subset



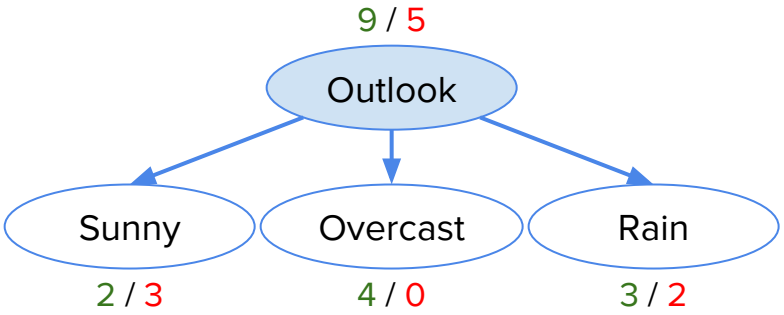
New data:

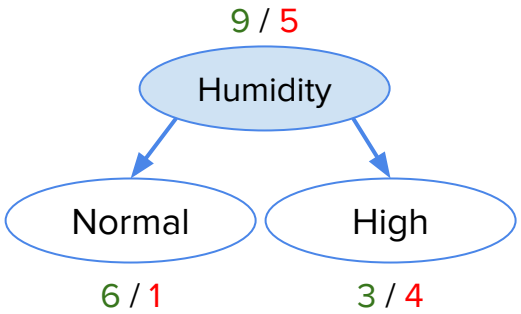
Day	Outlook	Humidity	Wind	Cricket
15	Rain	High	Weak	?

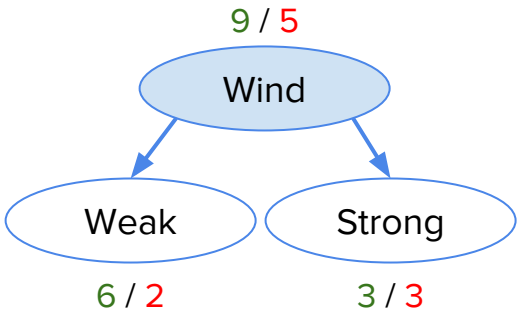
→ yes

Decision trees

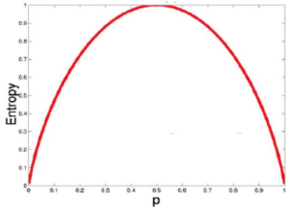
- Which attribute provides the best split?







- Split attributes based on a purity measure (e.g. entropy, gini)
- Information gain = average weighted entropy of each subset
- Example: http://www.saedsayad.com/decision_tree.htm



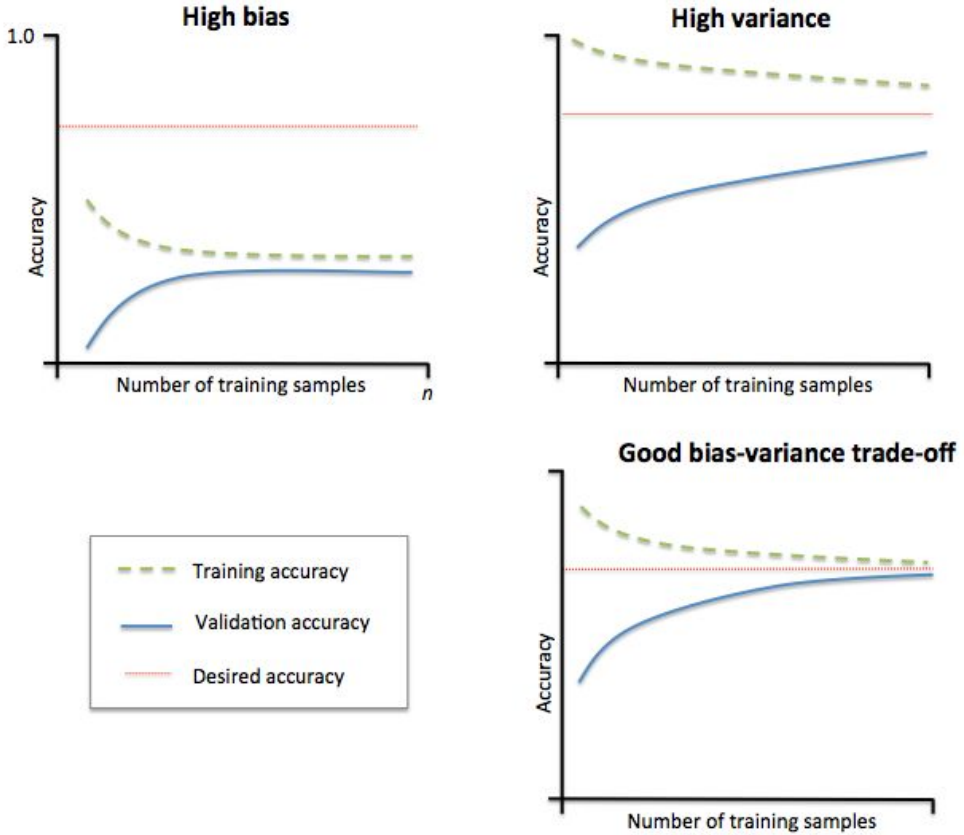
Learning curves

Underfit (high bias)

- Try more features
- Decrease regularisation

Overfit (high variance)

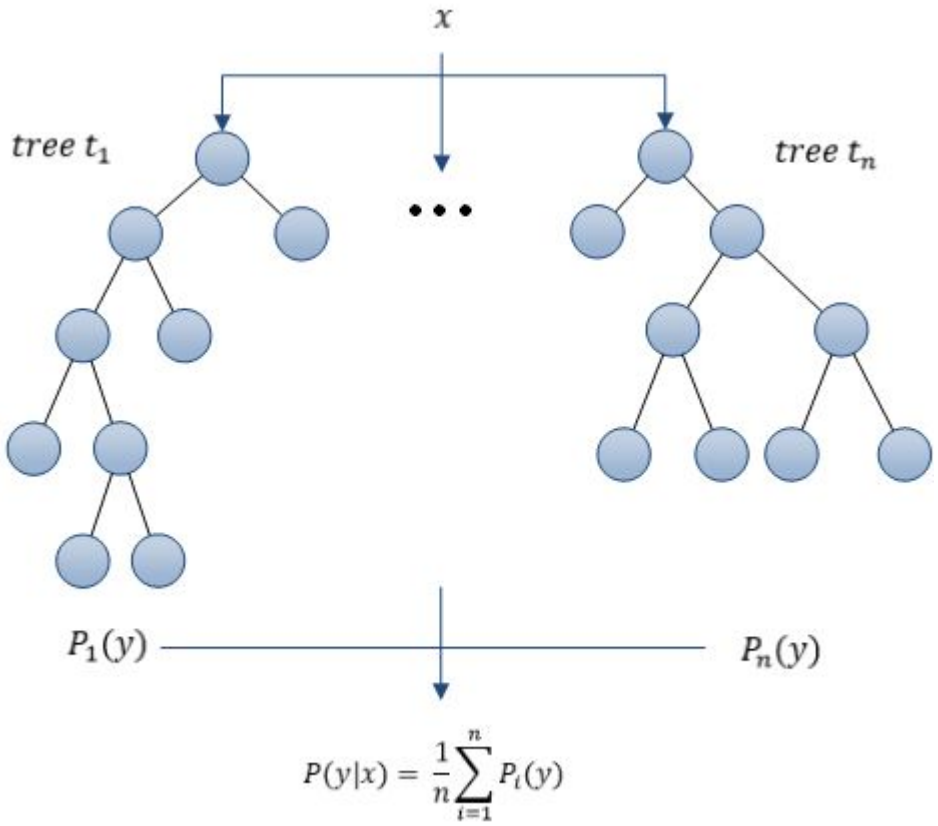
- Get more data
- Use less features
- Increase regularisation



Source: Andrew Ng, Coursera - Machine Learning. 2012

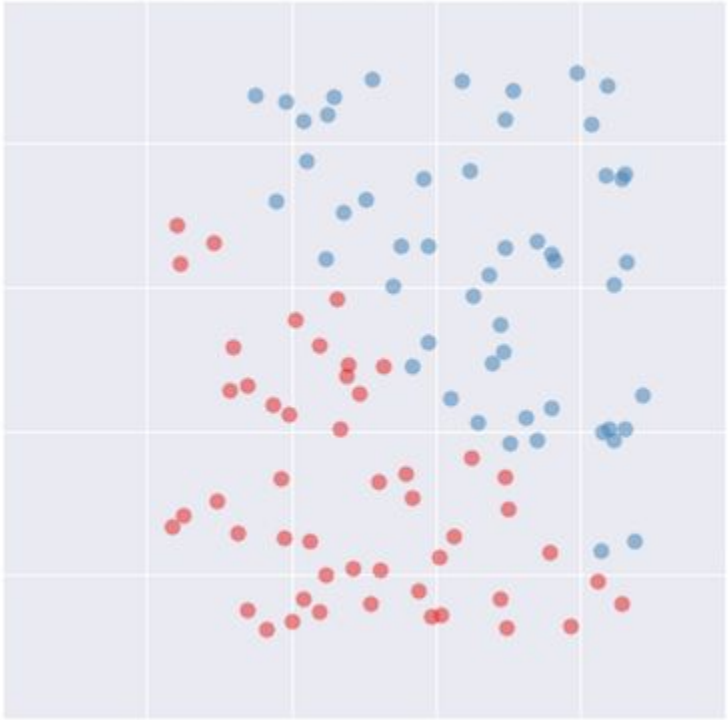
Random forests

- Train multiple decision trees (forest) based on random subsets of:
  - training examples
  - features
- Final prediction is based on the average of individual trees
- Avoids overfitting problems with single decision trees



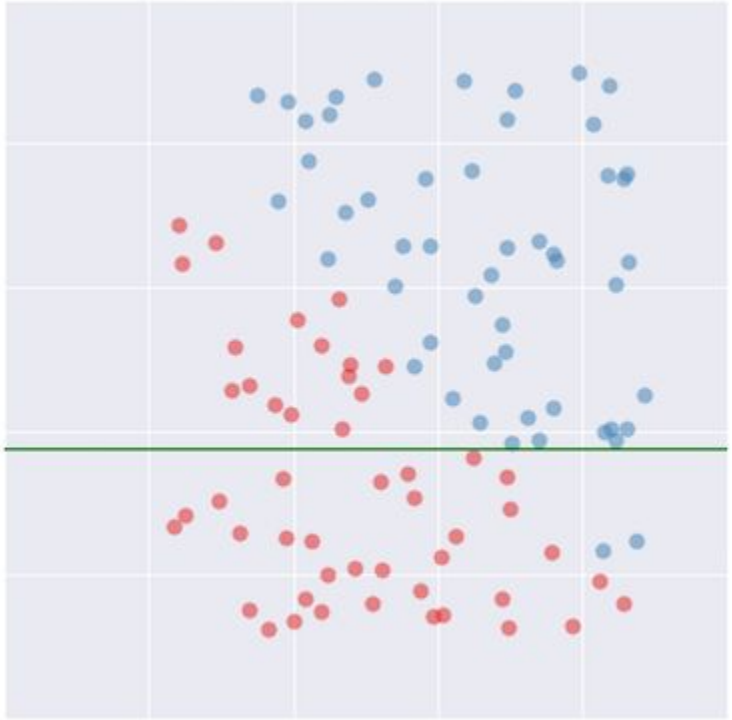
Disadvantages

- Decision trees split one attribute at a time
- Greedy algorithm
- Can result in an overly complex model



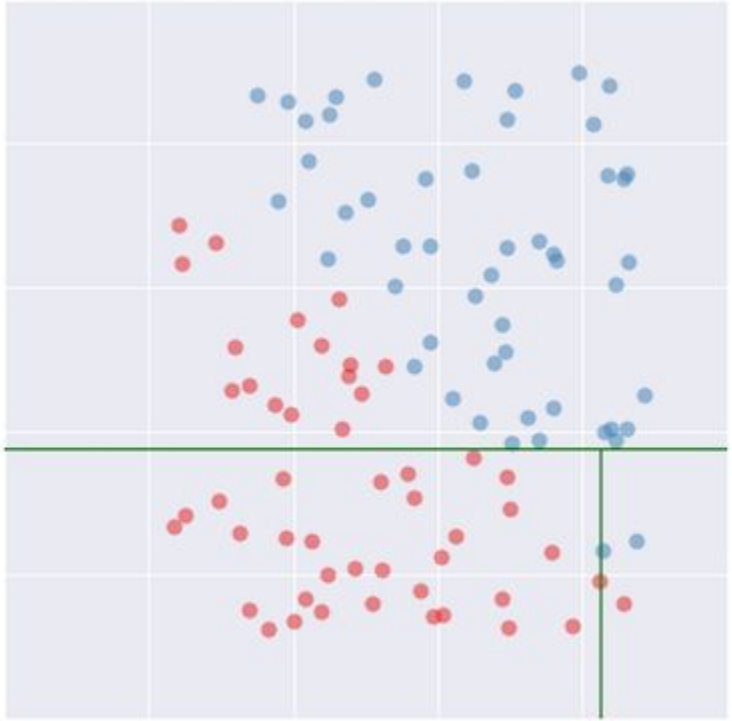
Disadvantages

- Decision trees split one attribute at a time
- Greedy algorithm
- Can result in an overly complex model



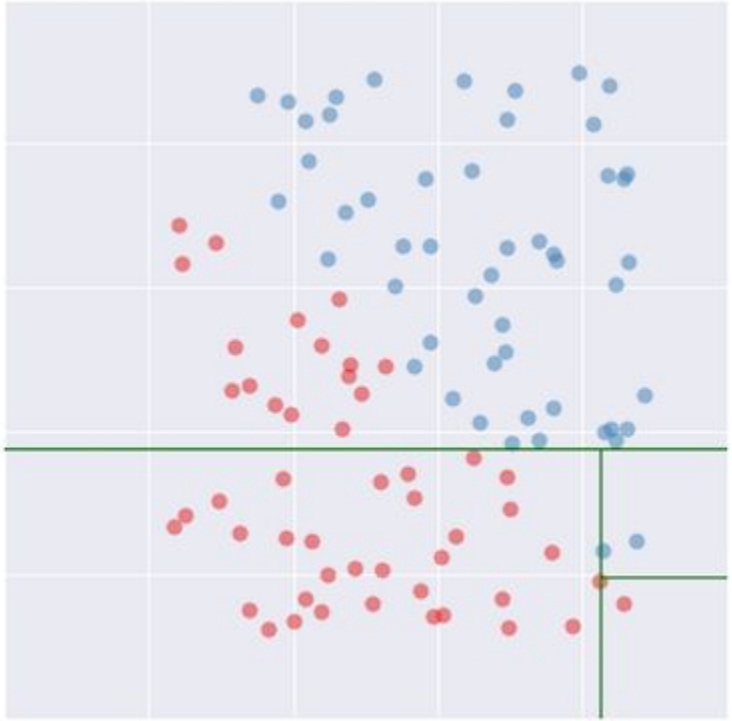
Disadvantages

- Decision trees split one attribute at a time
- Greedy algorithm
- Can result in an overly complex model



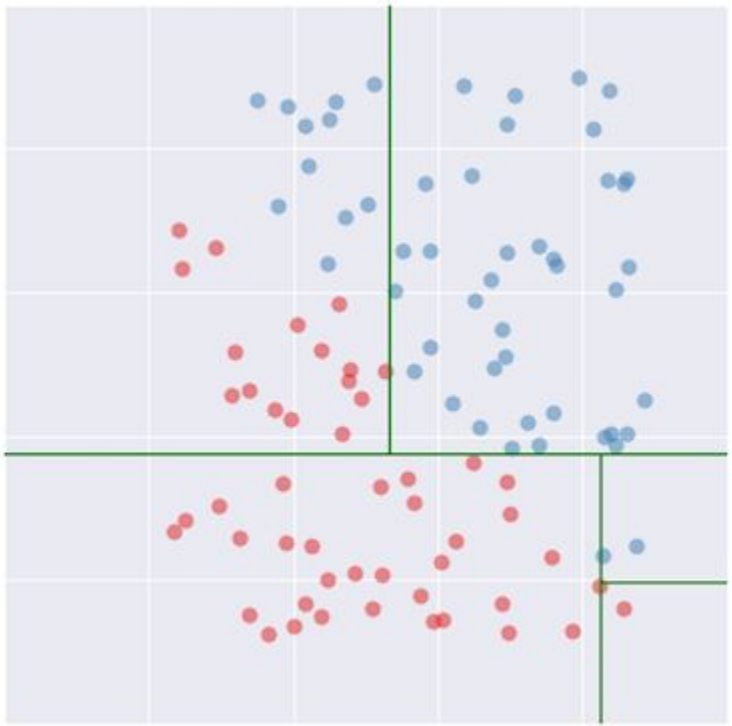
Disadvantages

- Decision trees split one attribute at a time
- Greedy algorithm
- Can result in an overly complex model



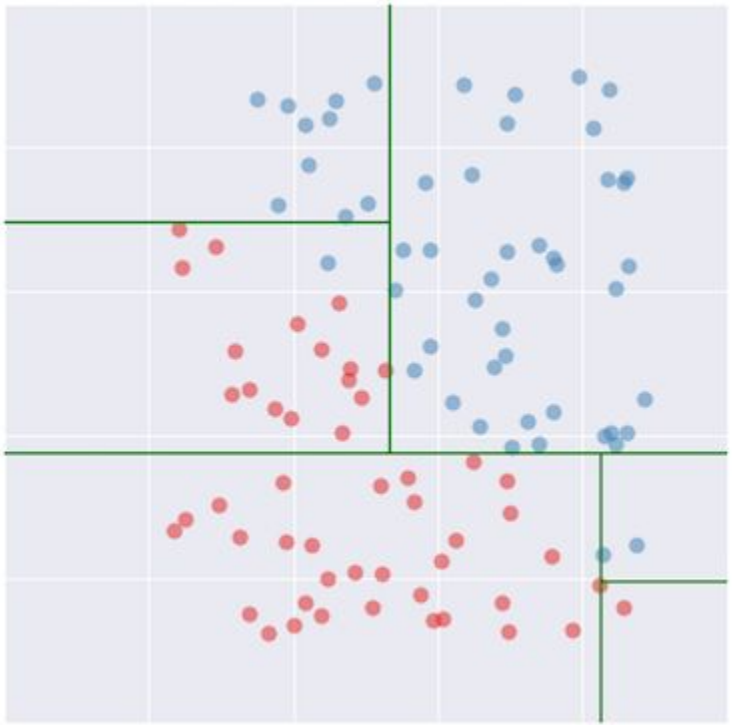
Disadvantages

- Decision trees split one attribute at a time
- Greedy algorithm
- Can result in an overly complex model



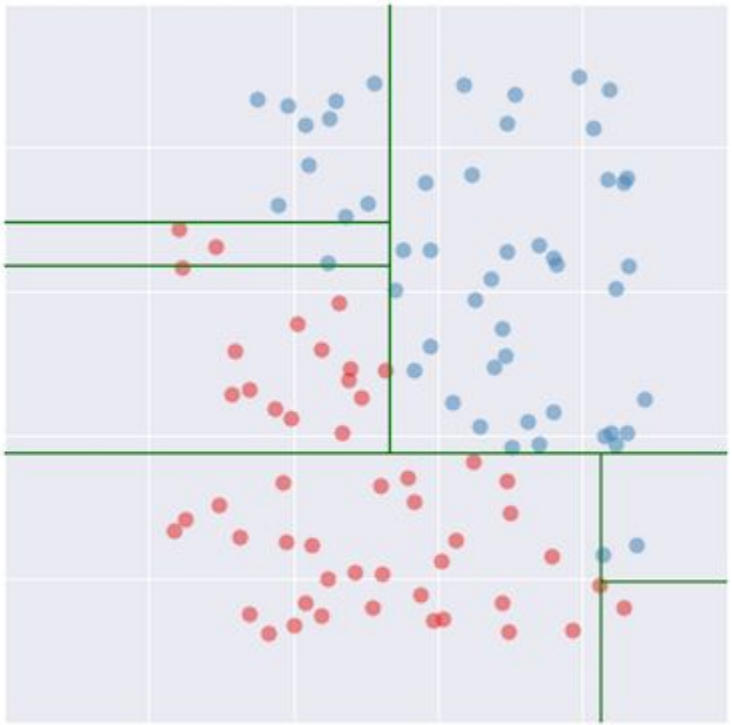
Disadvantages

- Decision trees split one attribute at a time
- Greedy algorithm
- Can result in an overly complex model



Disadvantages

- Decision trees split one attribute at a time
- Greedy algorithm
- Can result in an overly complex model



Disadvantages

- Decision trees split one attribute at a time
- Greedy algorithm
- Can result in an overly complex model

Disadvantages

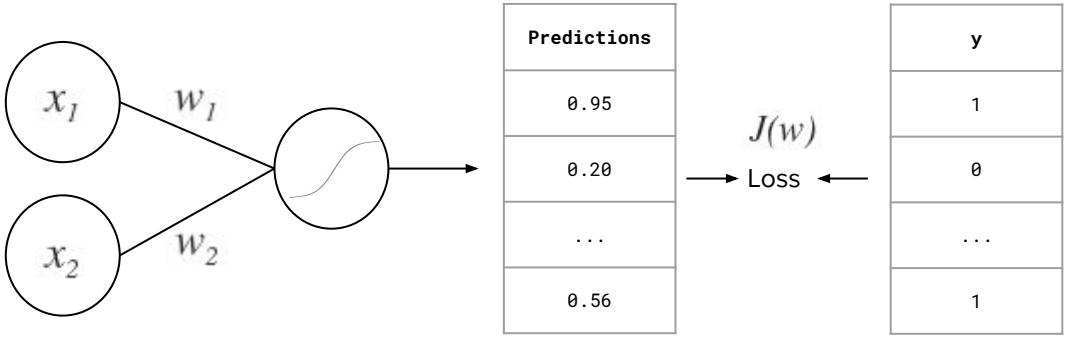
Better model

Logistic Regression

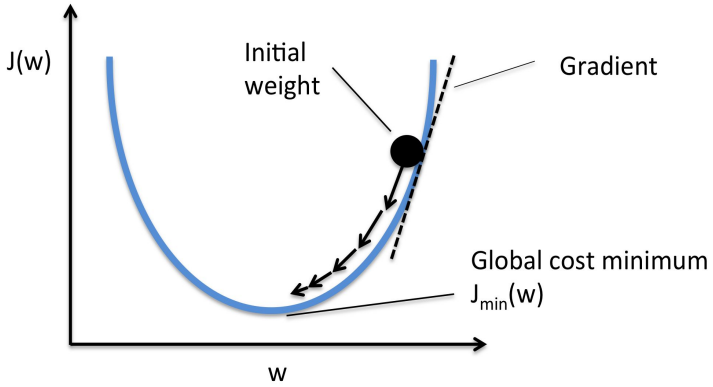
- Linear model

$$z = w_1 x_1 + w_2 x_2 + b$$

3 parameters



Optimisation



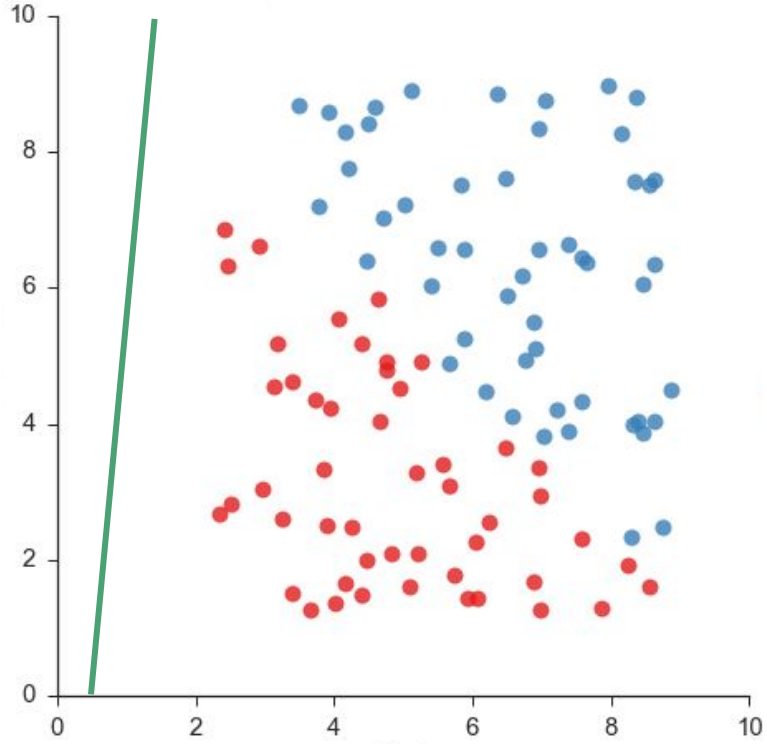
Logistic Regression

Iteration: 1

Model: $0.09x_1 - 0.01x_2 - 0.024$

Loss: 0.641

Accuracy: 0.51



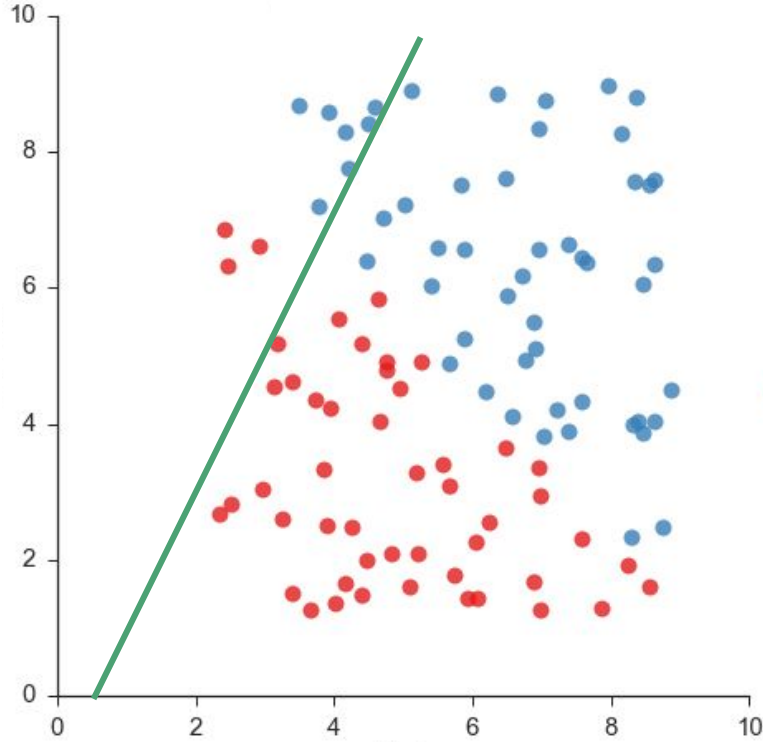
Logistic Regression

Iteration: 2

Model: $0.17x_1 - 0.08x_2 - 0.07$

Loss: 0.623

Accuracy: 0.66



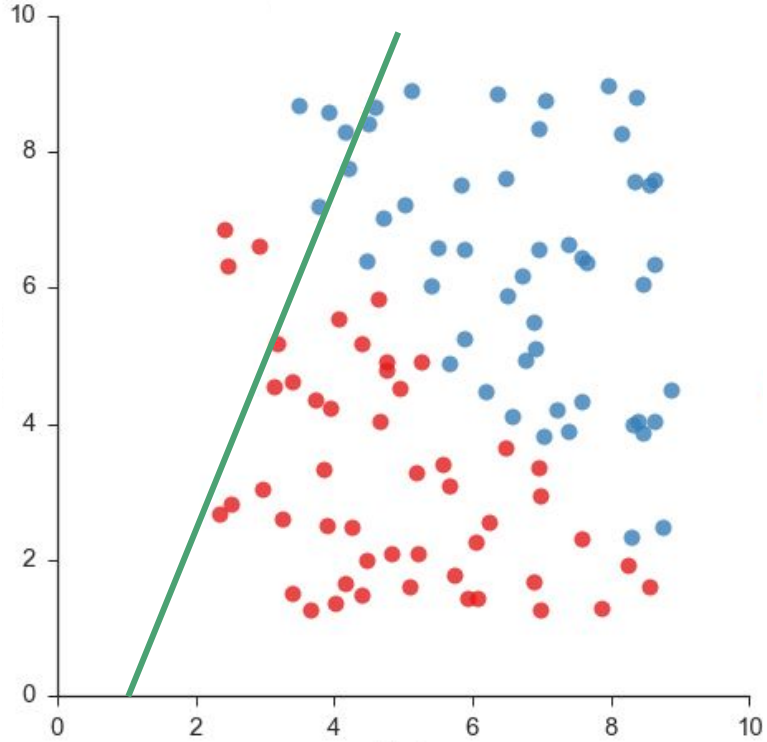
Logistic Regression

Iteration: 5

Model: $0.42x_1 - 0.16x_2 - 0.46$

Loss: 0.583

Accuracy: 0.68



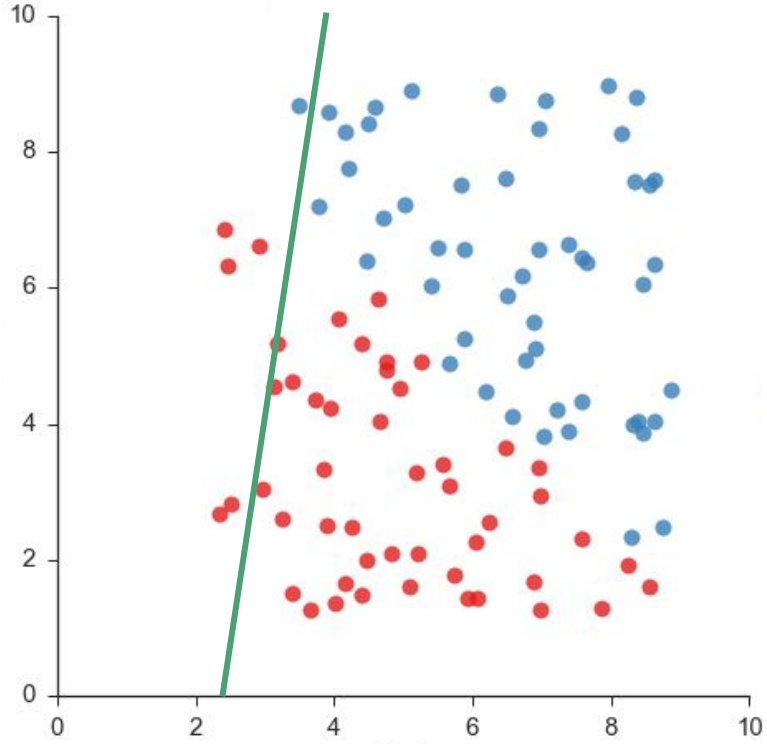
Logistic Regression

Iteration: 7

Model: $0.64x_1 - 0.08x_2 - 1.65$

Loss: 0.527

Accuracy: 0.77



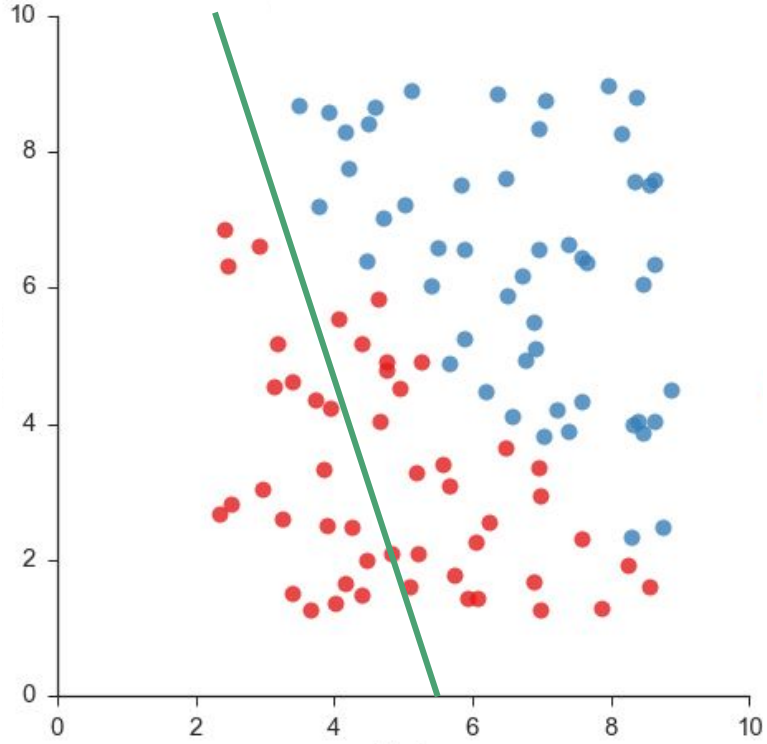
Logistic Regression

Iteration: 9

Model: $1.15x_1 - 0.37x_2 - 6.39$

Loss: 0.351

Accuracy: 0.82



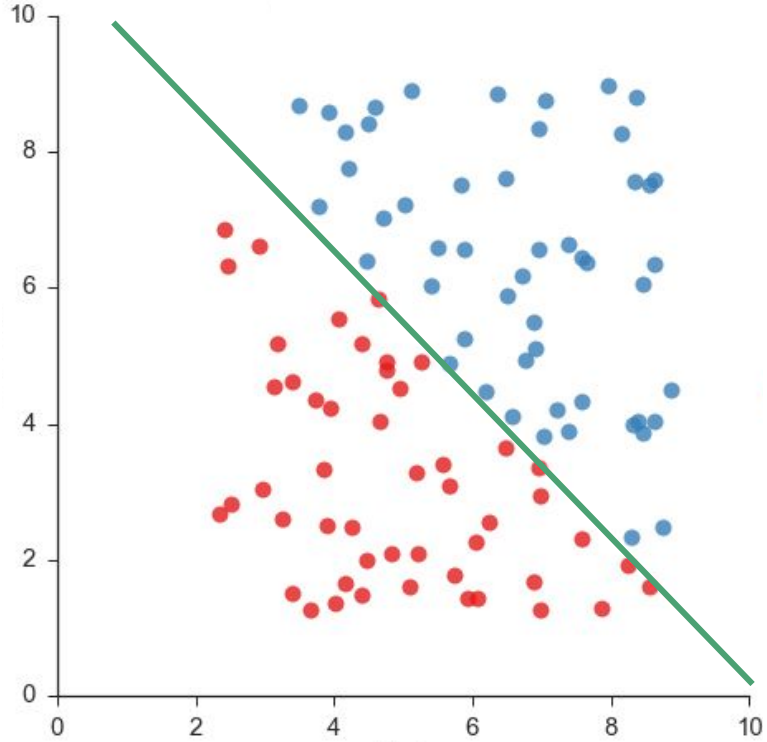
Logistic Regression

Iteration:18

Model: $2.13x_1 - 2.05x_2 - 21.88$

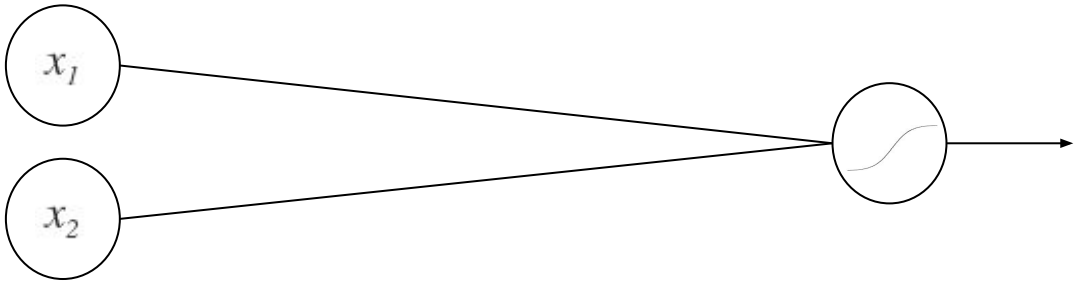
Loss: 0.091

Accuracy: 0.99



Artificial Neural Networks

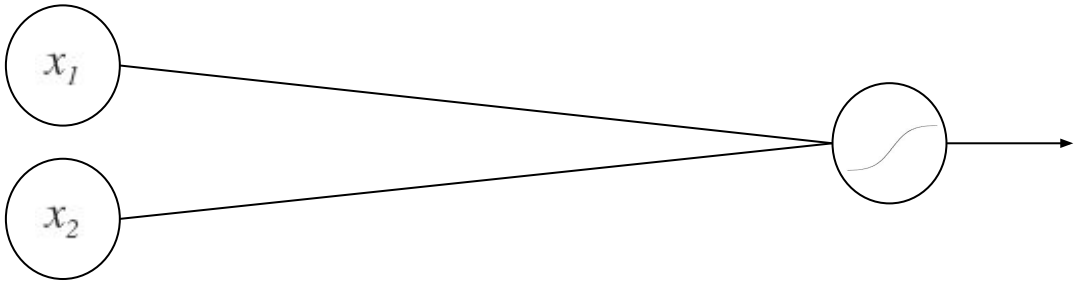
Logistic regression



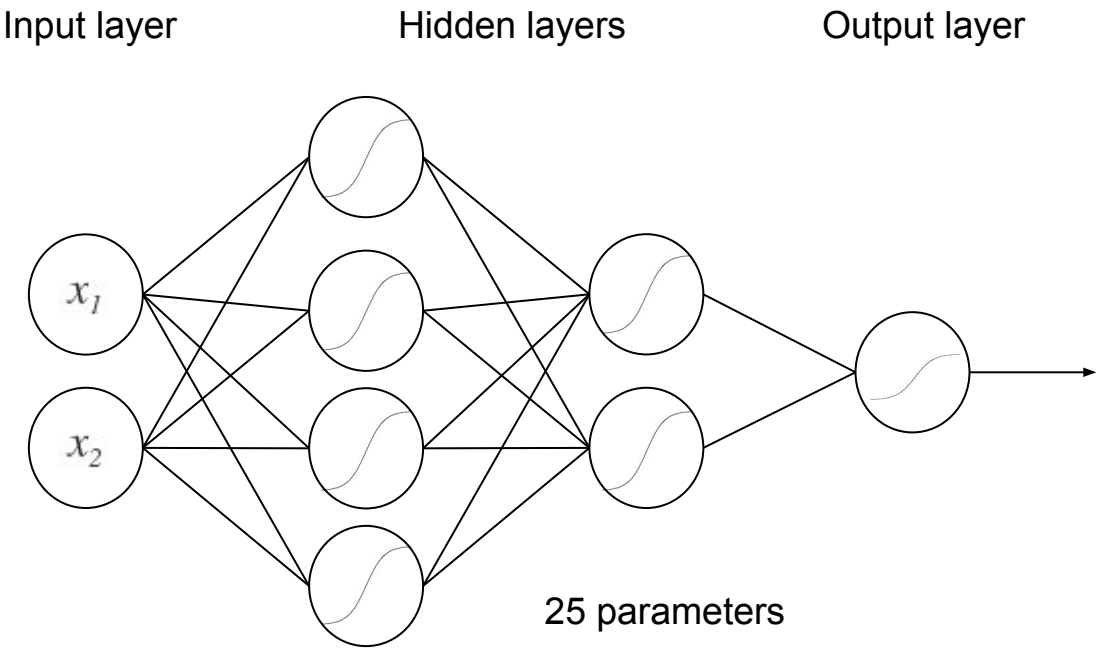
Artificial Neural Networks

Input layer

Output layer



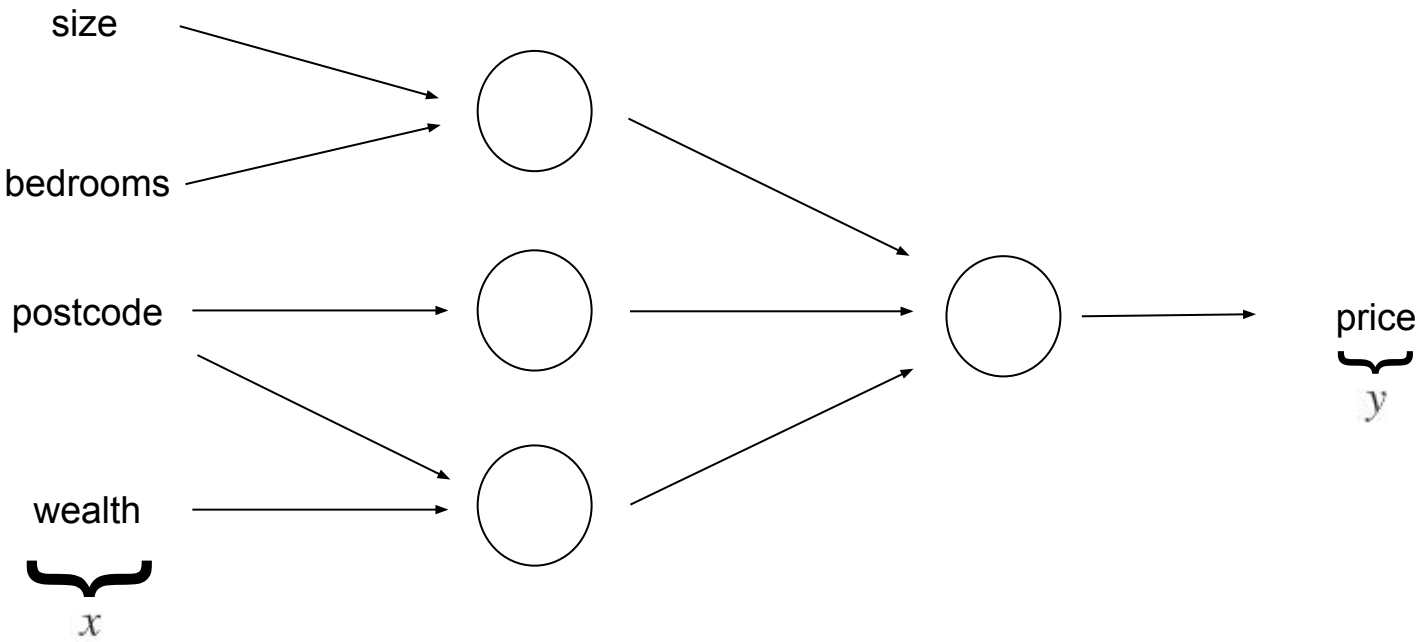
Artificial Neural Networks



Demo: http://playground.tensorflow.org/

Artificial Neural Networks

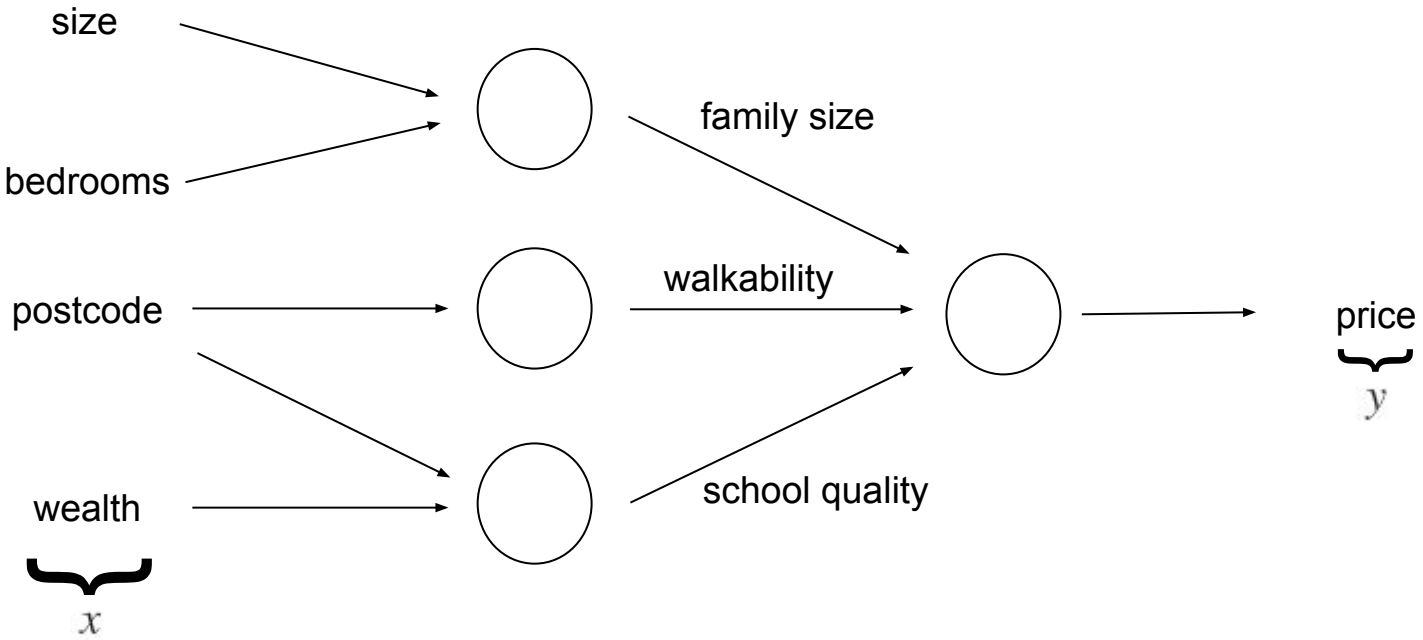
Housing price prediction



Source: Andrew Ng, Coursera - Neural Networks and Deep Learning. 2017

Artificial Neural Networks

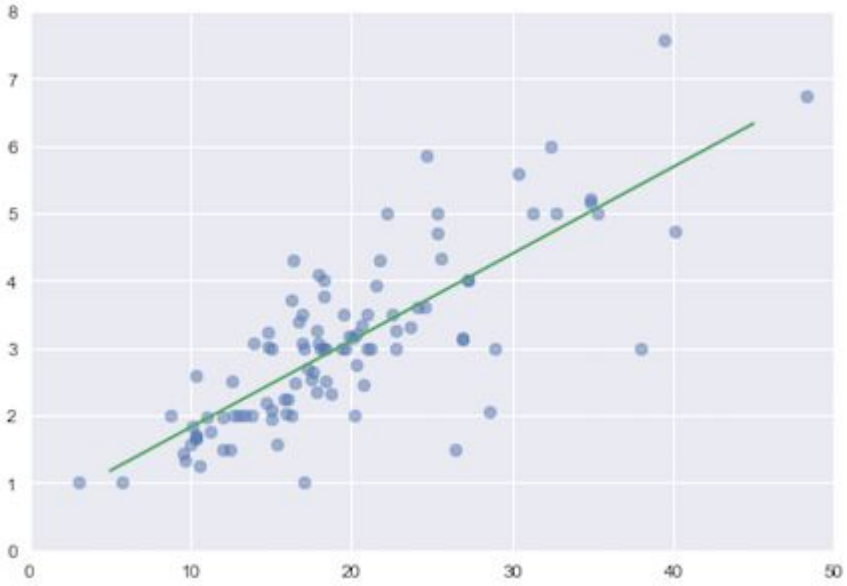
Housing price prediction



Source: Andrew Ng, Coursera - Neural Networks and Deep Learning. 2017

Linear regression

- Predict a continuous value
- Find the line of best fit
- Training process similar to logistic regression
  - different loss function
  - mean squared error

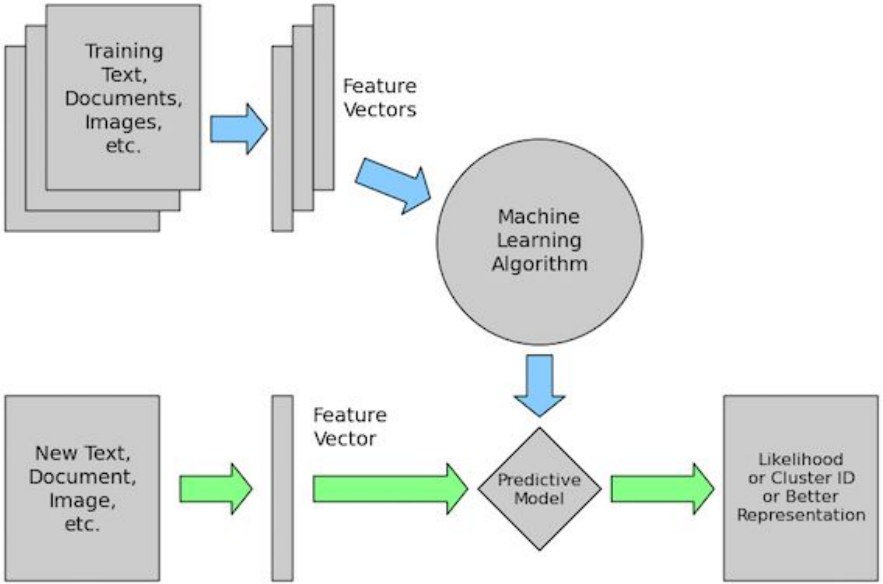


- Linear model

$$y = 0.12x + 0.53$$

Unsupervised learning

- Learn underlying structure of the data
- Dimensionality reduction

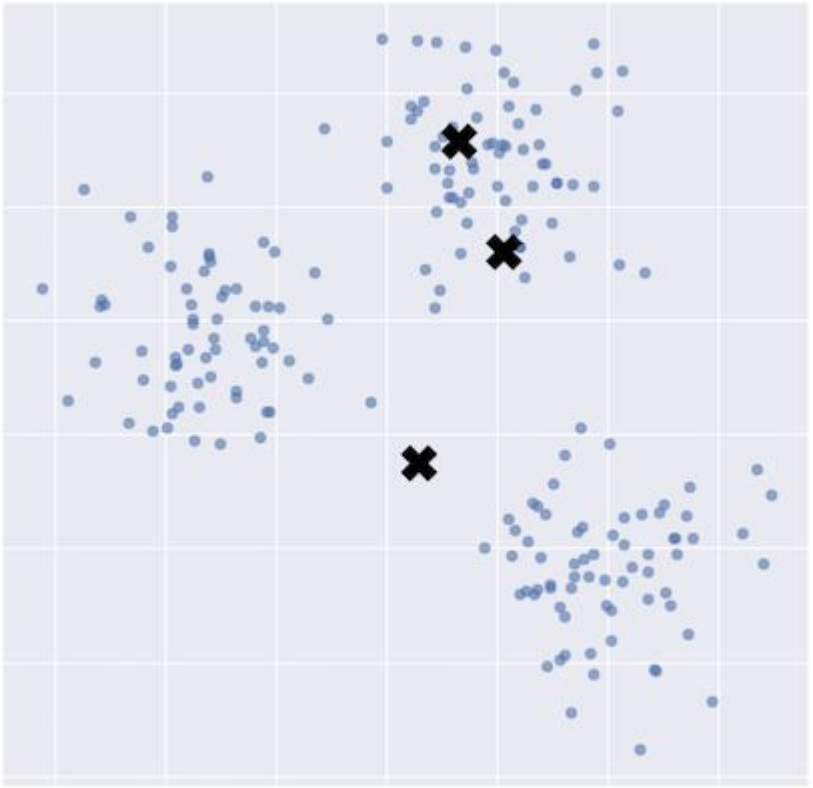


K-means

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don’t change
- New examples
  - Assign to cluster with closest centroid

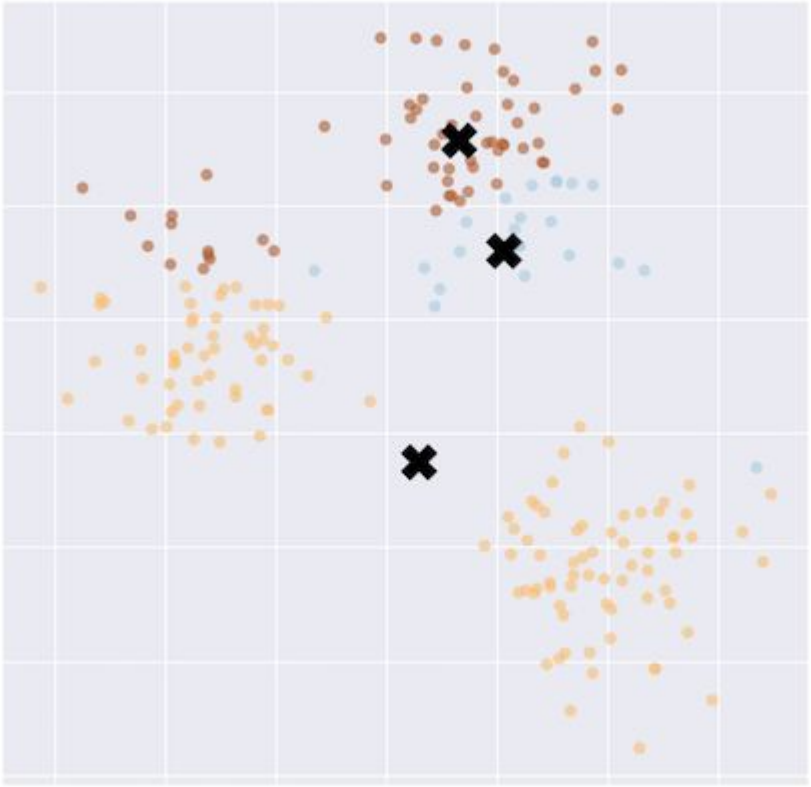
K-means

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don’t change
- New examples
  - Assign to cluster with closest centroid



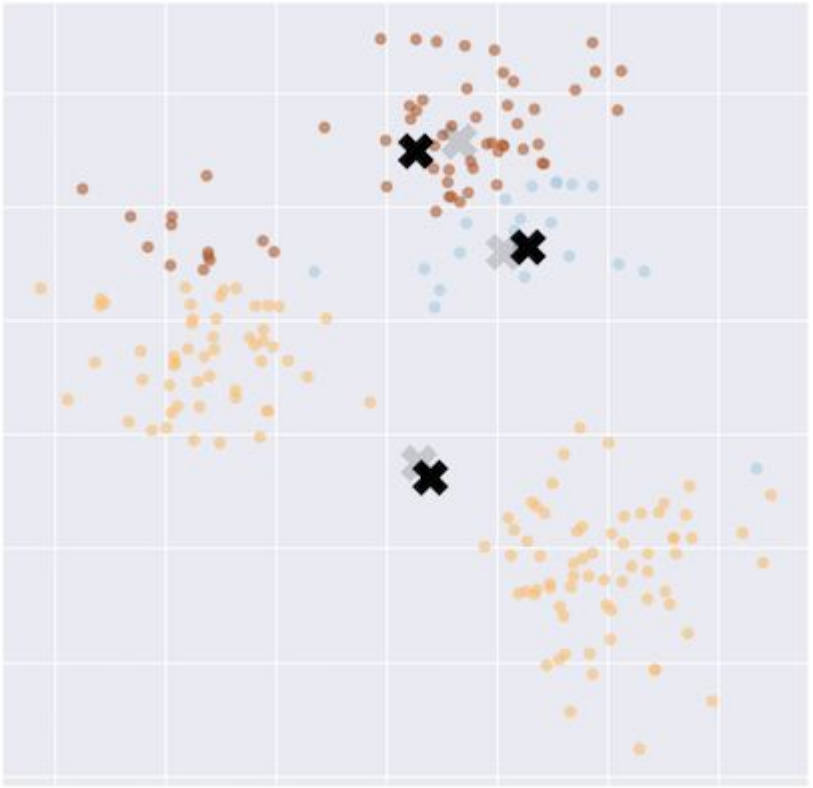
K-means

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don’t change
- New examples
  - Assign to cluster with closest centroid



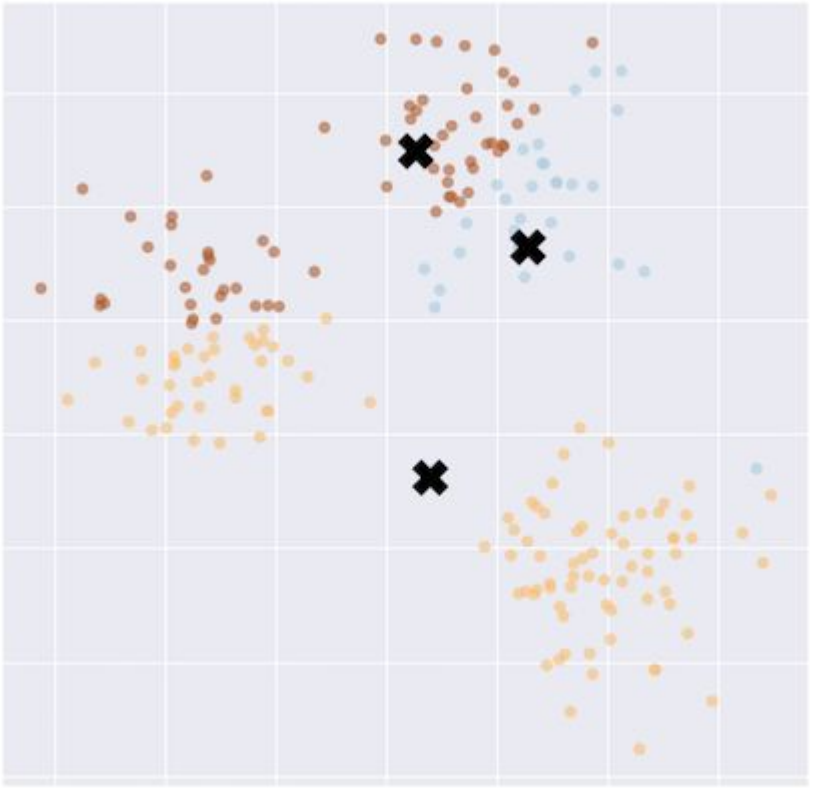
K-means

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid



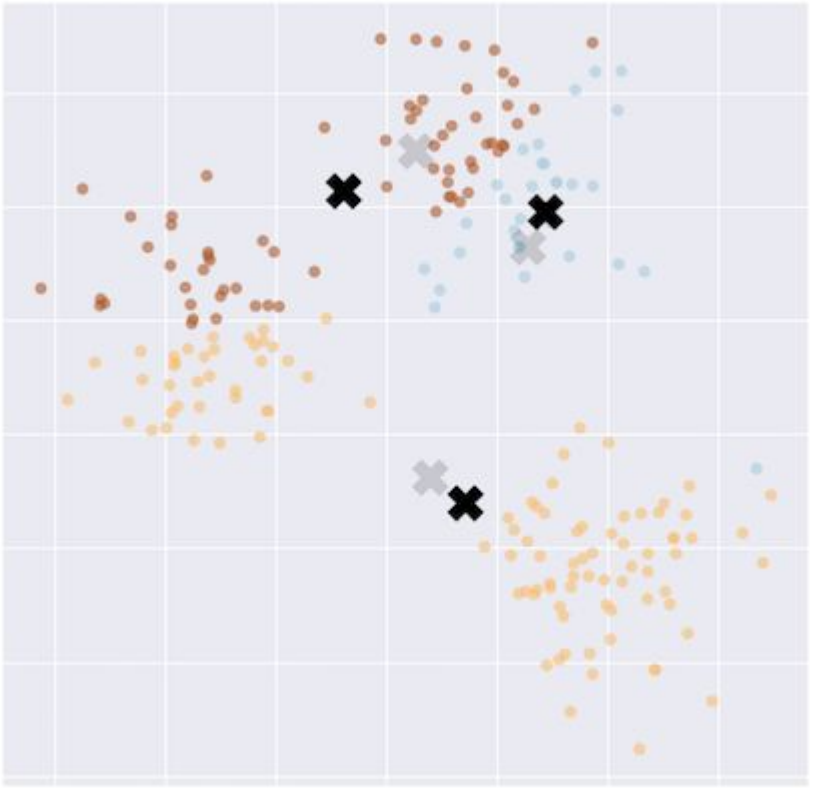
K-means

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don’t change
- New examples
  - Assign to cluster with closest centroid



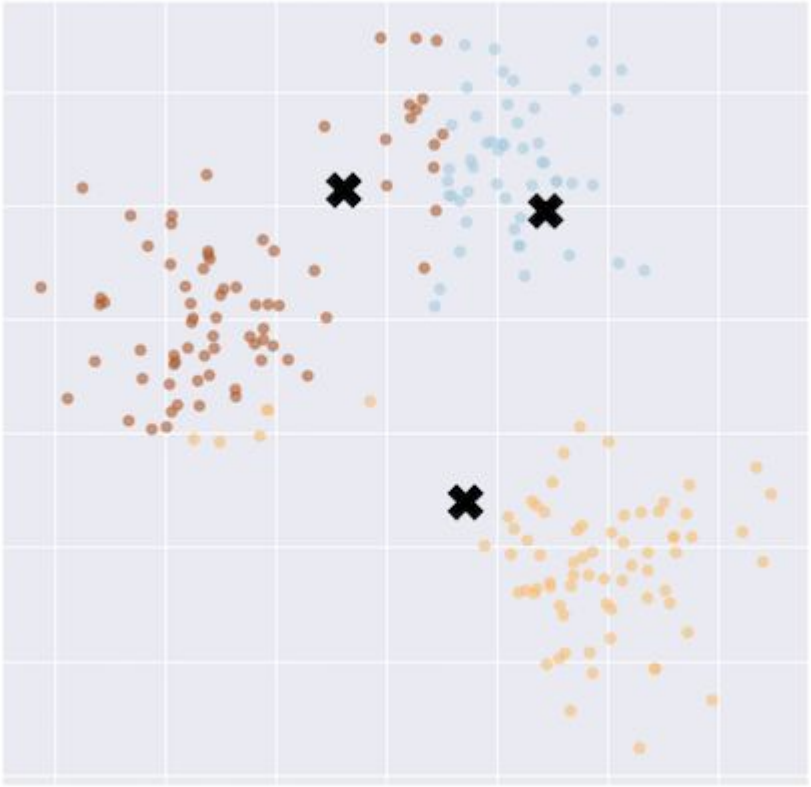
K-means

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don’t change
- New examples
  - Assign to cluster with closest centroid



K-means

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don’t change
- New examples
  - Assign to cluster with closest centroid



K-means

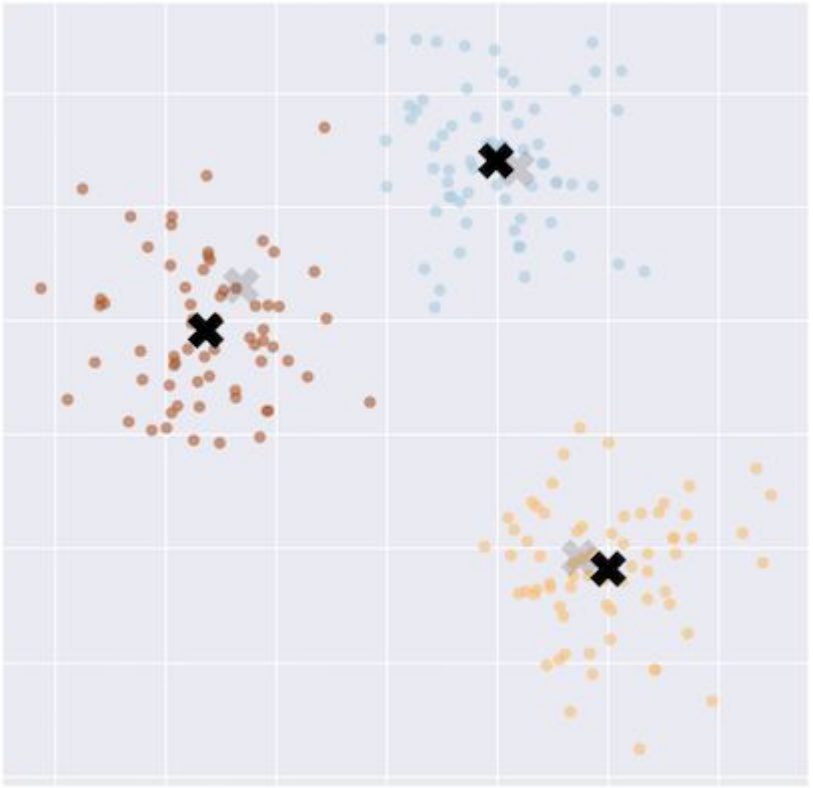
- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid

K-means

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don’t change
- New examples
  - Assign to cluster with closest centroid

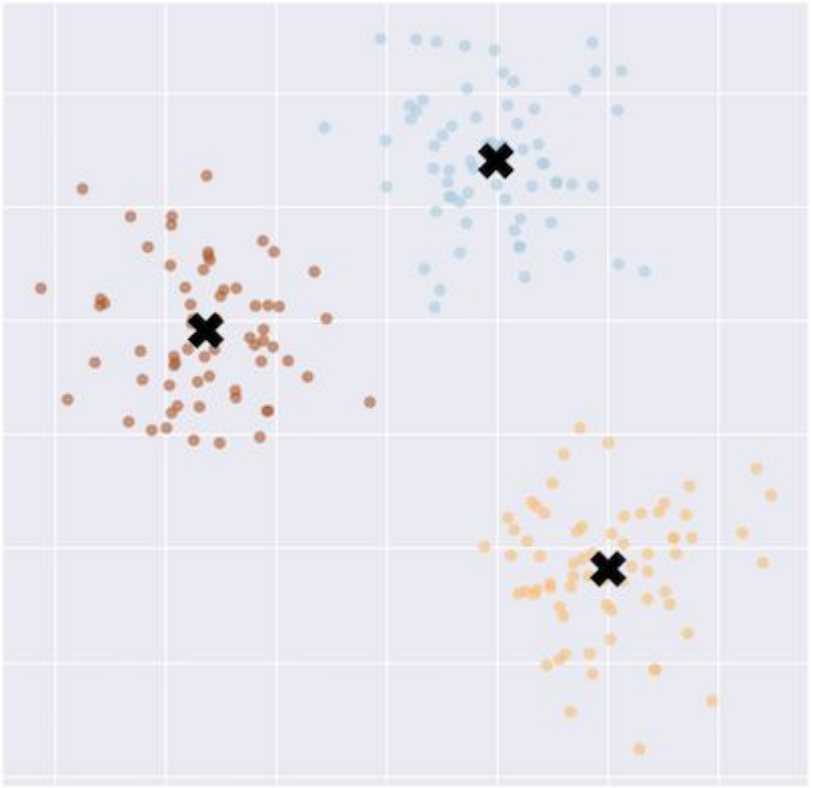
K-means

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don’t change
- New examples
  - Assign to cluster with closest centroid



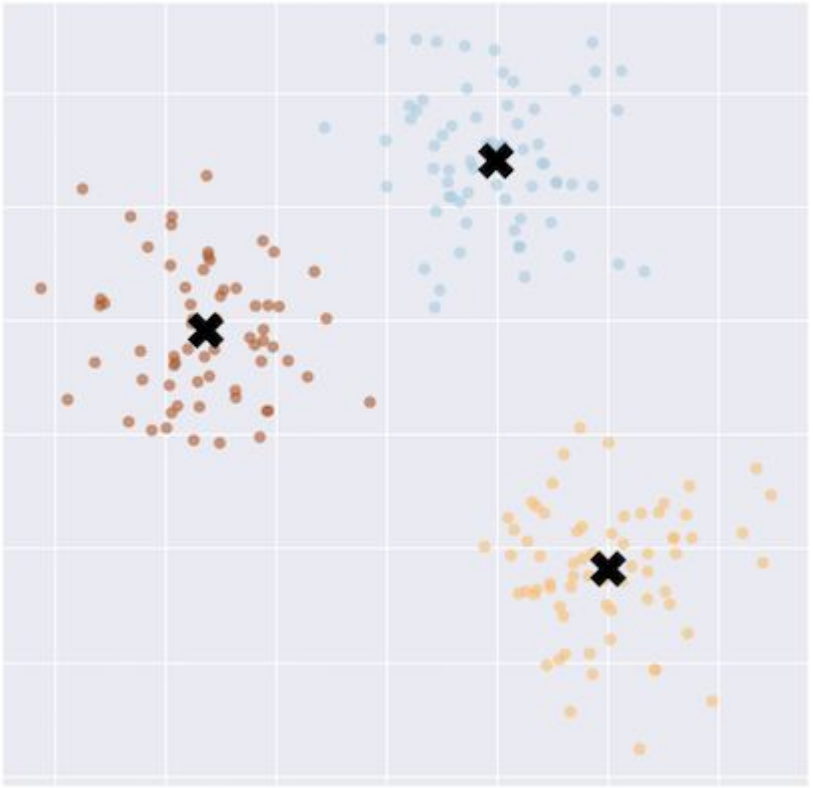
K-means

- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don’t change
- New examples
  - Assign to cluster with closest centroid



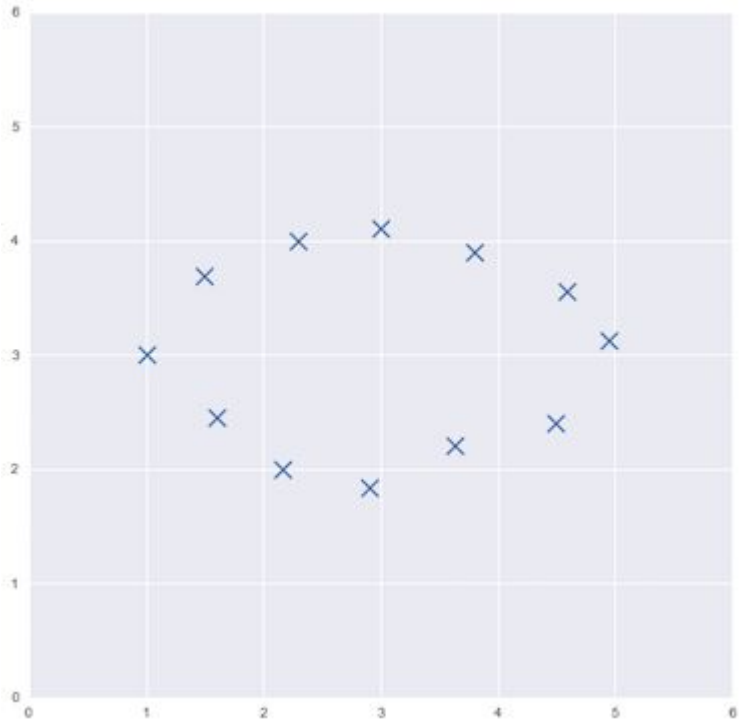
K-means

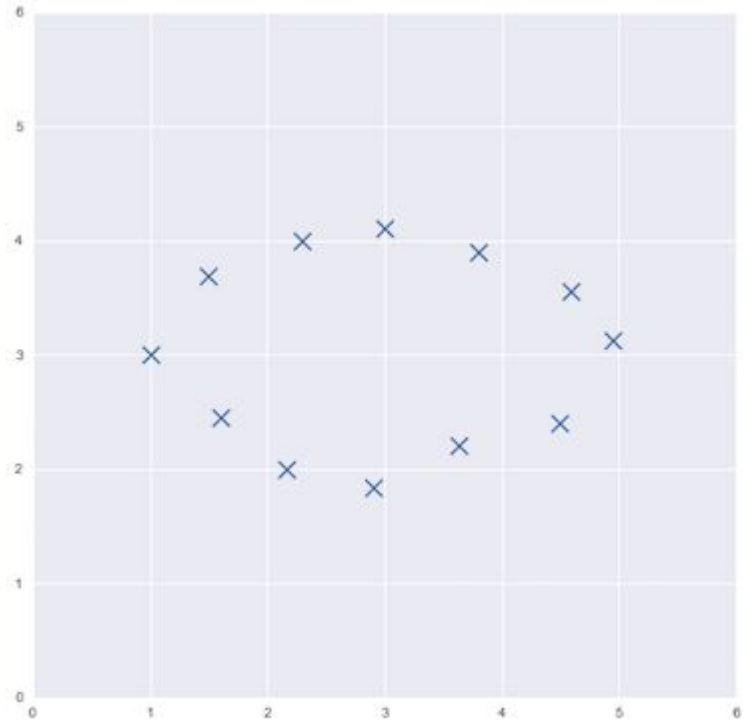
- Clustering algorithm
  - K = # of clusters
- Training
  - Randomly place K initial centroids
  - Assign each point to closest centroid
  - Update centroid position based on mean of points in assigned cluster
  - Stop when centroids don't change
- New examples
  - Assign to cluster with closest centroid



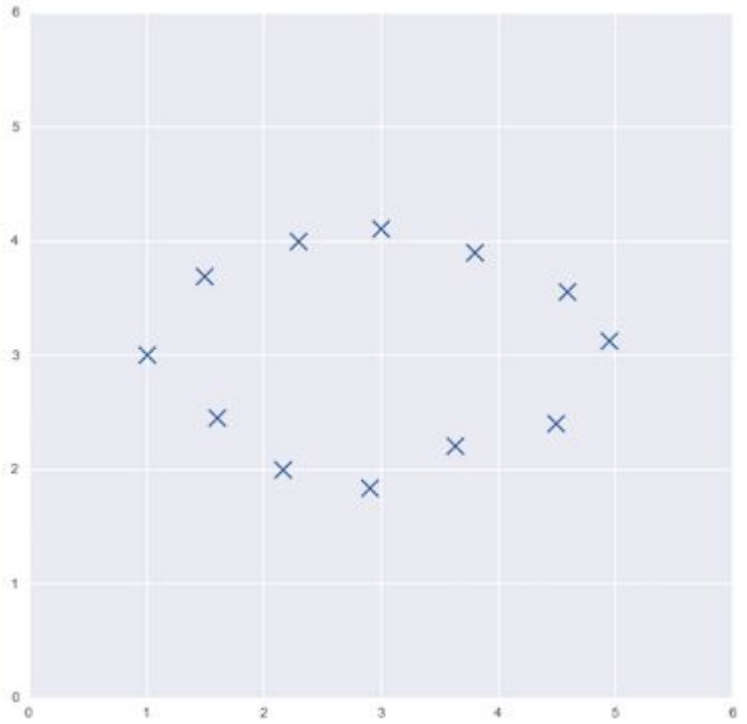
Principal Components Analysis (PCA)

- Dimensionality reduction method
- Represent data in fewer dimensions
  - Preserve as much structure as possible
- New dimensions = principal components
  - Directions where data is most spread (variance)
  - Combination of original features
- Example
  - Reduce a dataset from 2D to 1D
  - Manually try two components

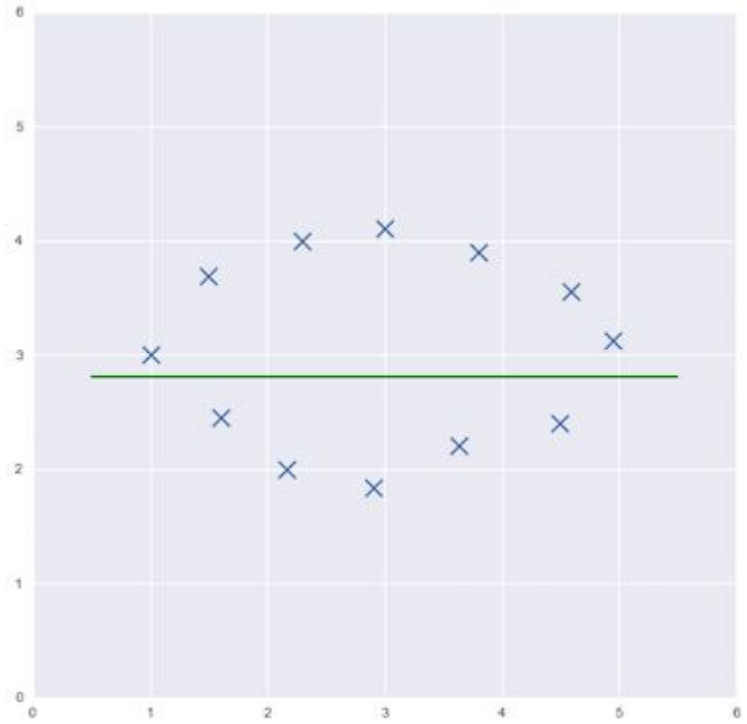




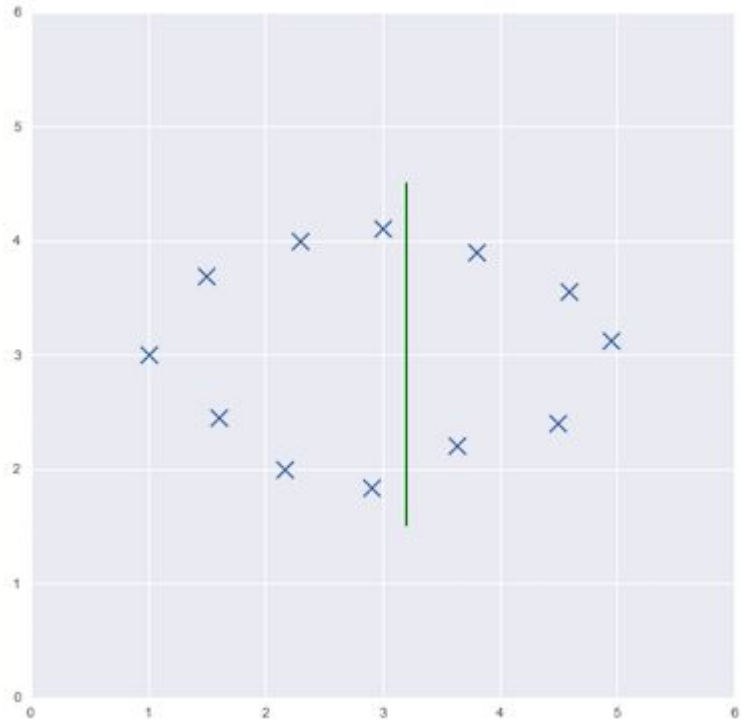
Component A



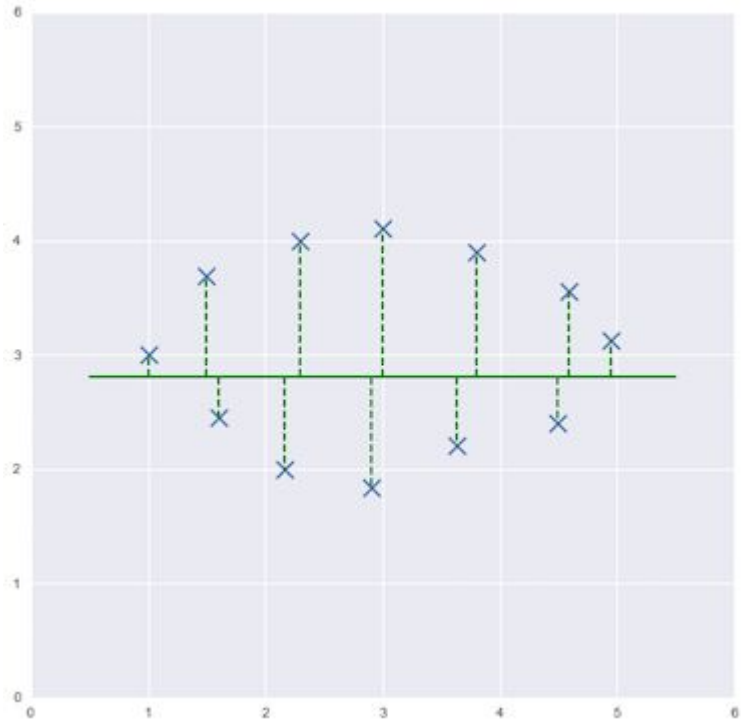
Component B



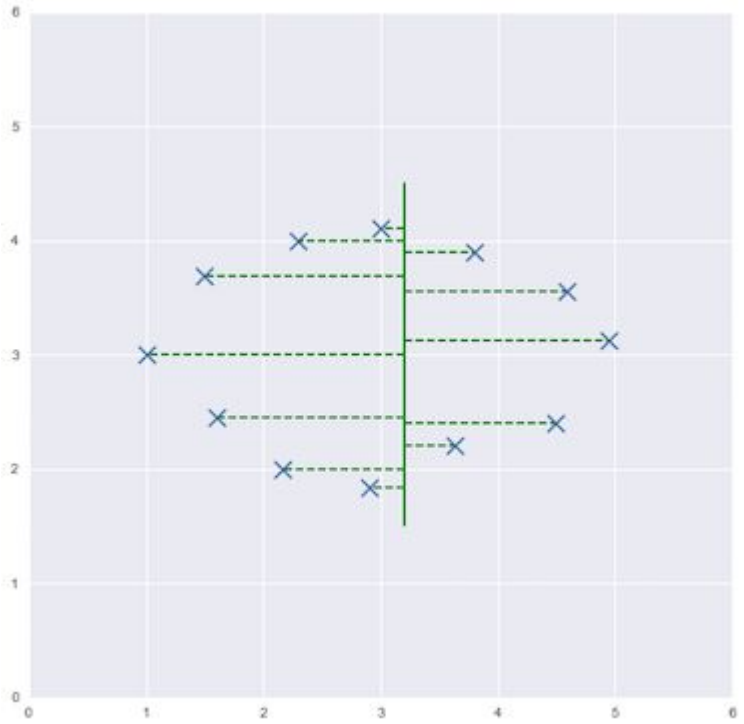
Component A



Component B



Component A

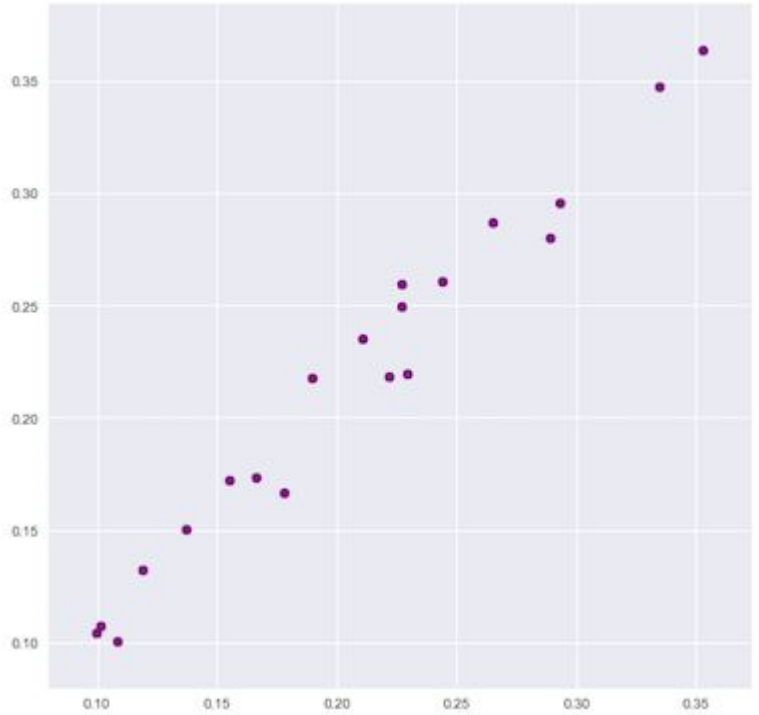


Component B

Which component represents a larger spread of the data (variance)? Component A

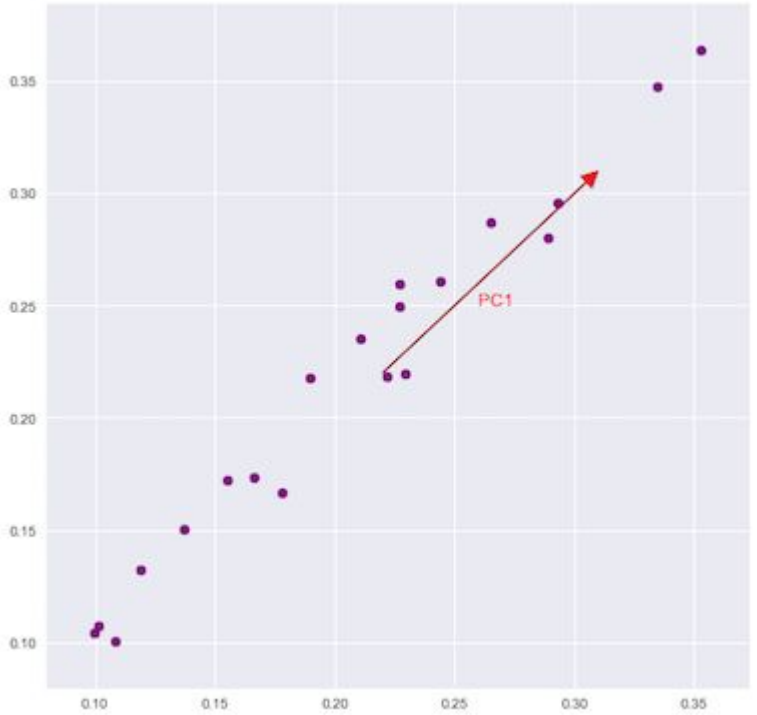
PCA

- Find a set of principal components
  - 1st: represents most variance
  - 2nd: perpendicular to 1st, most variance of remainder
  - ... until last dimension of dataset



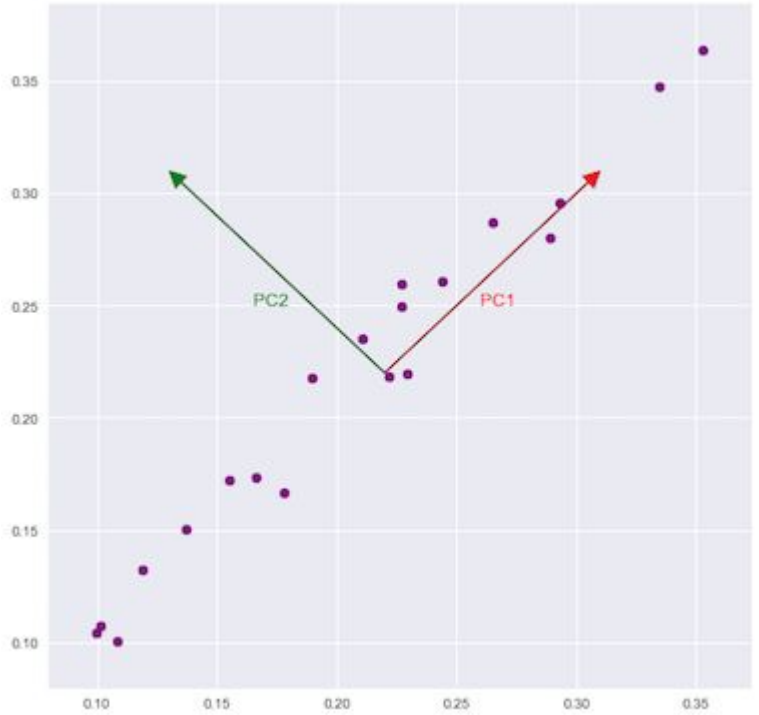
PCA

- Find a set of principal components
  - 1st: represents most variance
  - 2nd: perpendicular to 1st, most variance of remainder
  - ... until last dimension of dataset



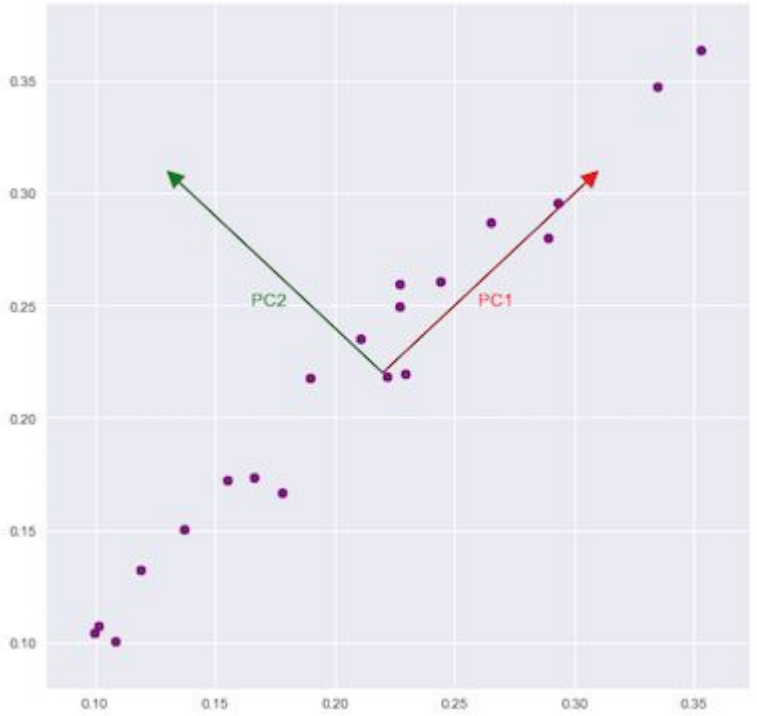
PCA

- Find a set of principal components
  - 1st: represents most variance
  - 2nd: perpendicular to 1st, most variance of remainder
  - ... until last dimension of dataset



PCA

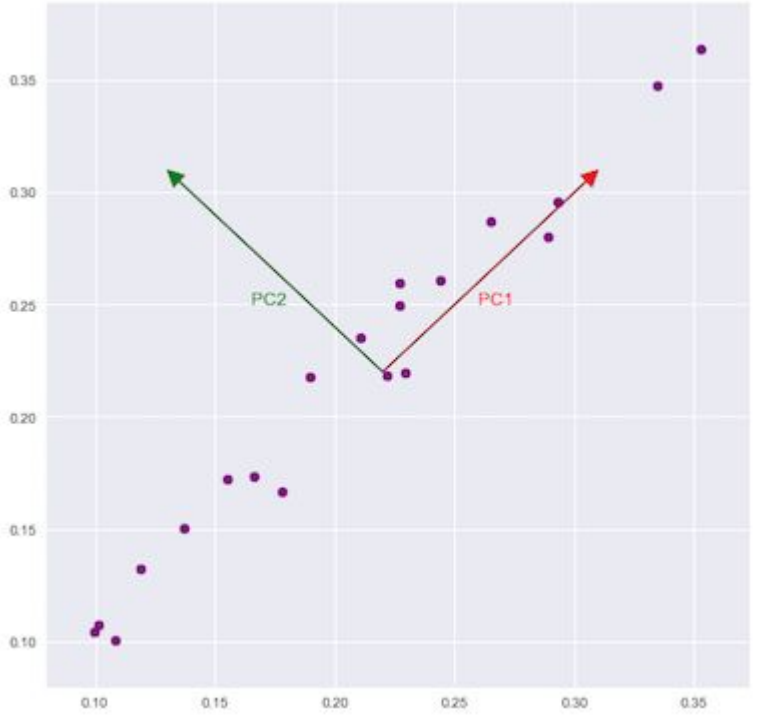
- Find a set of principal components
  - 1st: represents most variance
  - 2nd: perpendicular to 1st, most variance of remainder
  - ... until last dimension of dataset
- Reduce dataset to *k* components
  - E.g. keep top *k* components that represent at least 90% of variance in the dataset
  - Transform original data to the new dimensions



PC1 represents 99.6% of variance

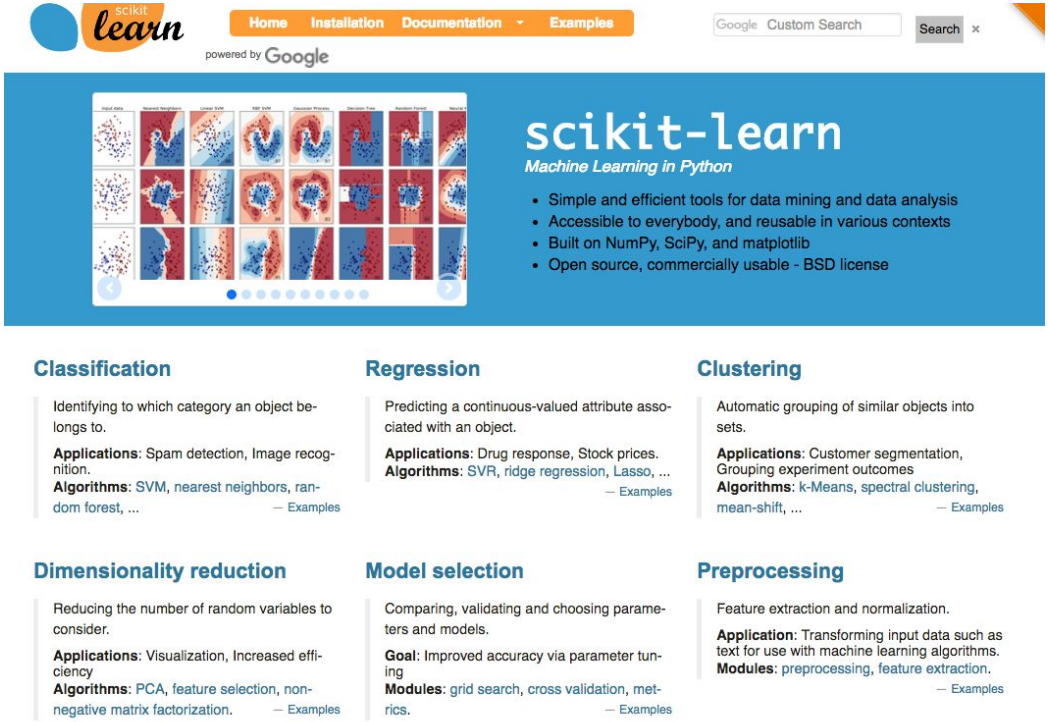
PCA

- Find a set of principal components
  - 1st: represents most variance
  - 2nd: perpendicular to 1st, most variance of remainder
  - ... until last dimension of dataset
- Reduce dataset to *k* components
  - E.g. keep top *k* components that represent at least 90% of variance in the dataset
  - Transform original data to the new dimensions
- Algorithm
  - Normalise data with zero mean
  - Calculate covariance matrix
  - Calculate:
    - Eigenvectors = principal component
    - Eigenvalues = amount of variance represented

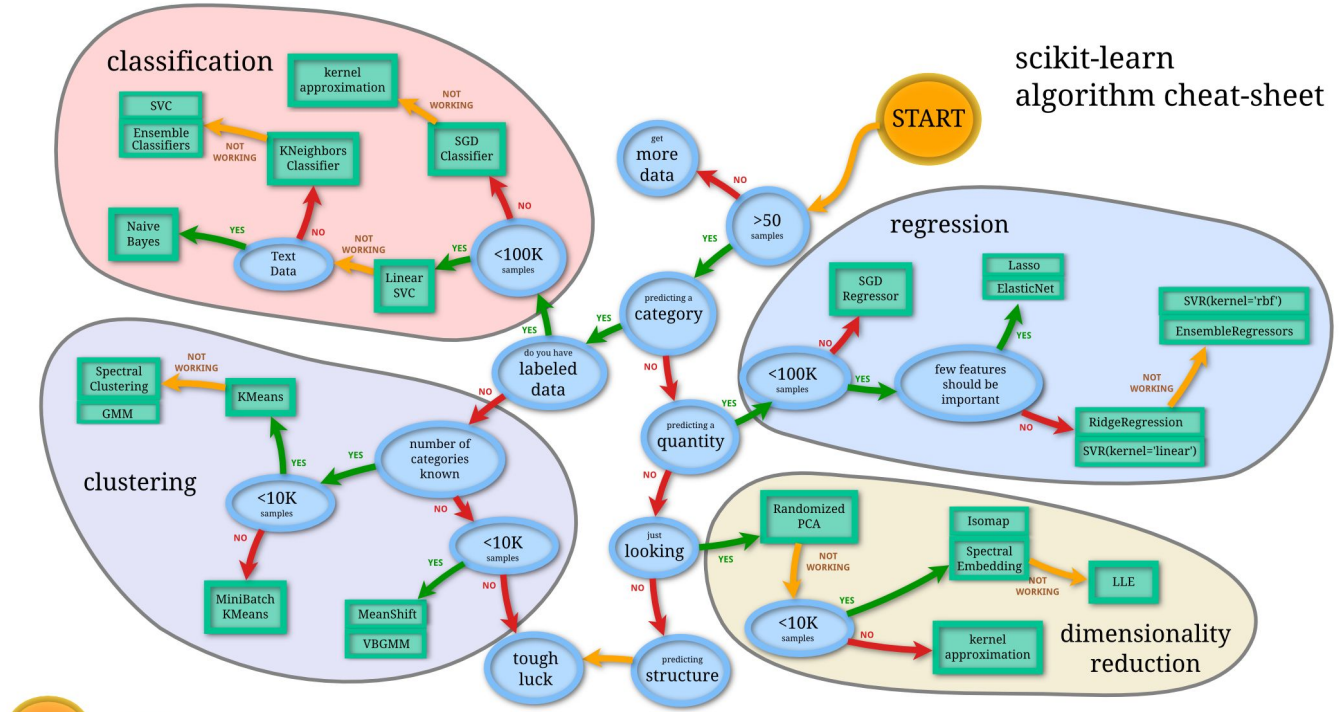


PC1 represents 99.6% of variance

Reference sheets



Reference sheets



Source: http://scikit-learn.org/stable/tutorial/machine_learning_map/

Reference sheets

Comparison of supervised machine learning techniques

Factors	Decision Trees	Neural Networks	Naïve Bayes	kNN	SVM	Rule Learners
General accuracy	**	***	*	**	****	**
Learning speed	***	*	****	****	*	**
Classification speed	****	****	****	*	****	****
Tolerance to missing values	***	*	****	*	**	**
Tolerance to irrelevant features	***	*	**	**	****	**
Tolerance to redundant features	**	**	*	**	***	**
Tolerance to highly related features	**	***	*	*	***	**
Dealing with discrete, binary and continuous features	****	***	***	***	**	***
Tolerance to noise	**	**	***	*	**	*
Dealing with model overfitting	**	*	***	***	**	**
Attempts for incremental learning	**	***	****	****	**	*
Explanation of classification	****	*	****	**	*	****
Model parameter handling	***	*	****	***	*	***

* kNN = k-Nearest Neighbours
* SVM = Support Vector Machines

Source: Kotsiantis et al., Supervised machine learning: A review of classification techniques, *Conference on Emerging Artificial Intelligence Applications* 2007.

Reference sheets

- Supervised learning
  - http://scikit-learn.org/stable/supervised_learning.html#supervised-learning
- Performance metrics
  - http://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics
- Feature selection
  - http://scikit-learn.org/stable/modules/feature_selection.html
- Model selection (cross validation, learning curves, hyperparameter tuning)
  - http://scikit-learn.org/stable/modules/classes.html#module-sklearn.model_selection
- Clustering
  - http://scikit-learn.org/stable/modules/clustering.html#clustering
- Dimensionality reduction
  - http://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition

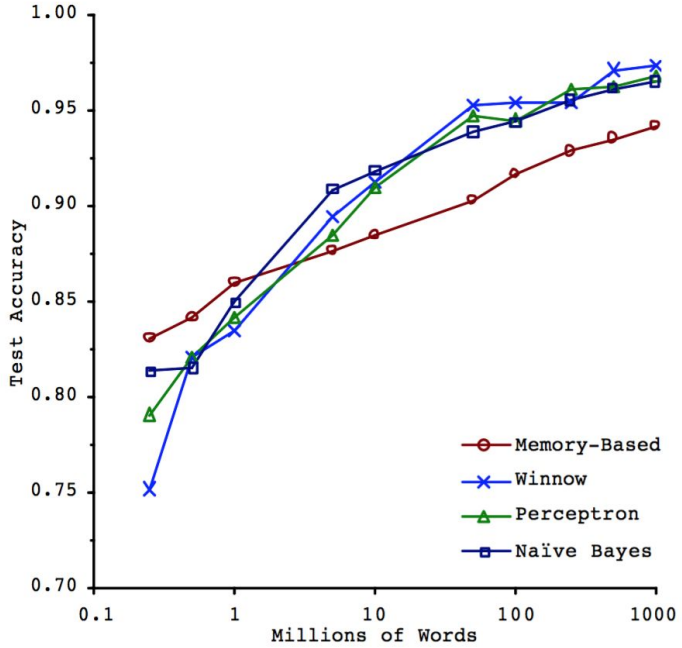
Reference sheets

- Anomaly detection
  - http://scikit-learn.org/stable/modules/outlier_detection.html
- Time series
  - http://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
  - https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/
  - http://www.statsmodels.org/dev/generated/statsmodels.tsa.arima_model.ARIMA.html
- ML blogs:
  - https://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/
  - http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Reference sheets

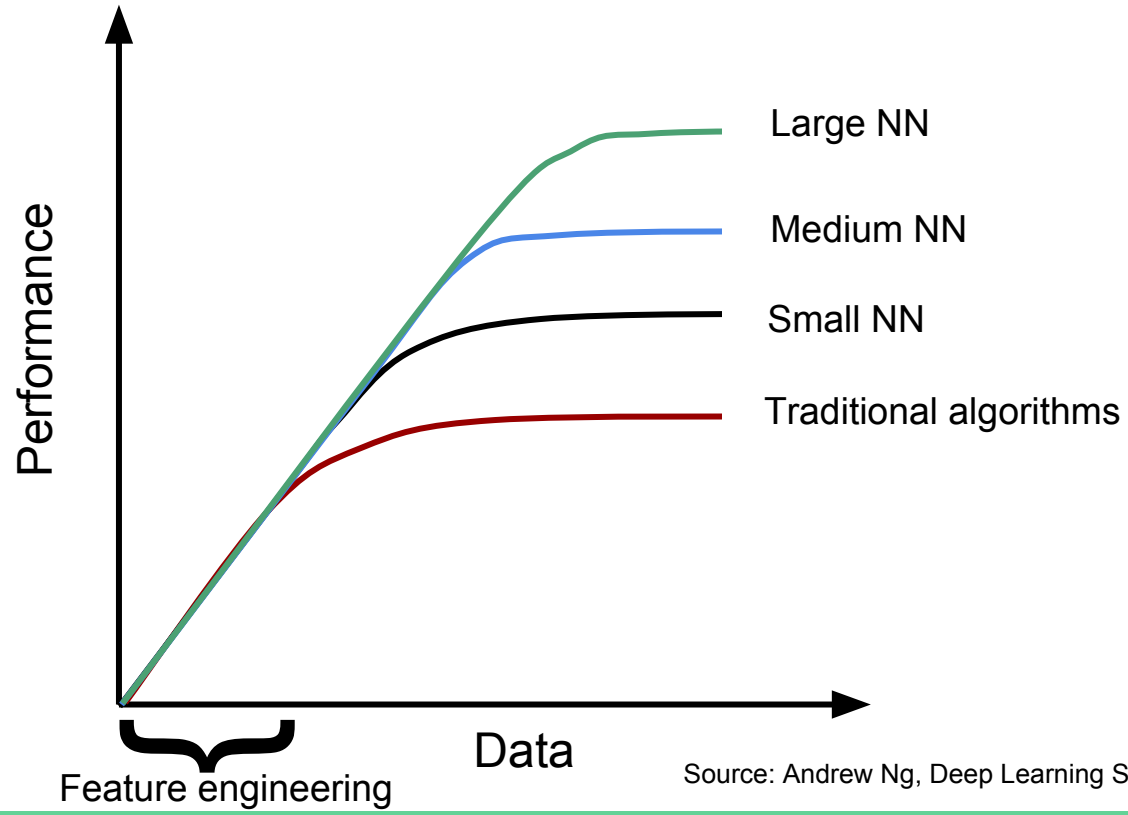
- Online courses
  - Machine learning
    - https://www.coursera.org/learn/machine-learning
    - https://www.udacity.com/course/intro-to-machine-learning--ud120
  - Deep learning
    - https://www.coursera.org/specializations/deep-learning
    - https://www.udacity.com/course/deep-learning--ud730
    - http://course.fast.ai/

Deep learning



Source: Banko & Brill. Scaling to Very Very Large Corpora for Natural Language Disambiguation. *Proceedings for the Association for Computational Linguistics (ACL)* 2001.

Deep learning



Source: Andrew Ng, Deep Learning School, Stanford University 2016

Deep learning

GALAXY ZOO
sponsored by

Completed • $16,000 • 326 teams

Galaxy Zoo - The Galaxy Challenge

Fri 20 Dec 2013 – Fri 4 Apr 2014 (2 years ago)

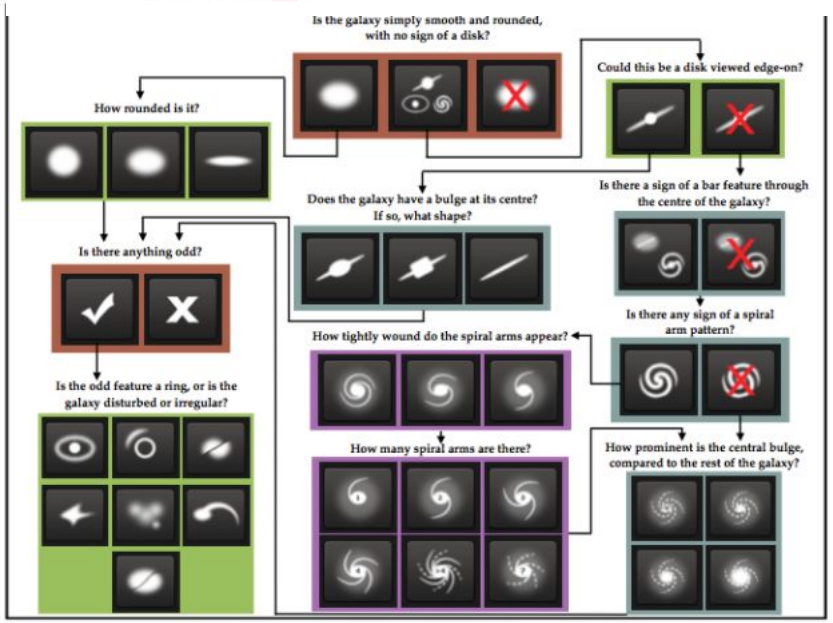
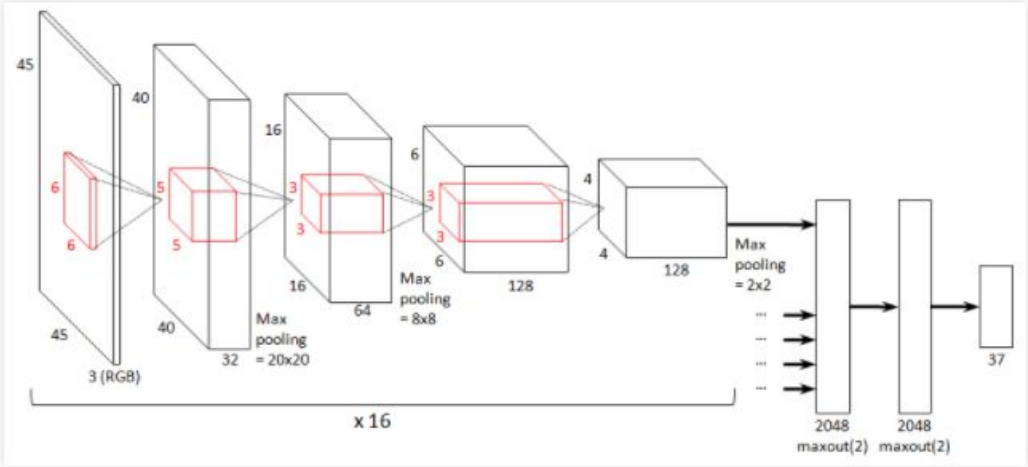
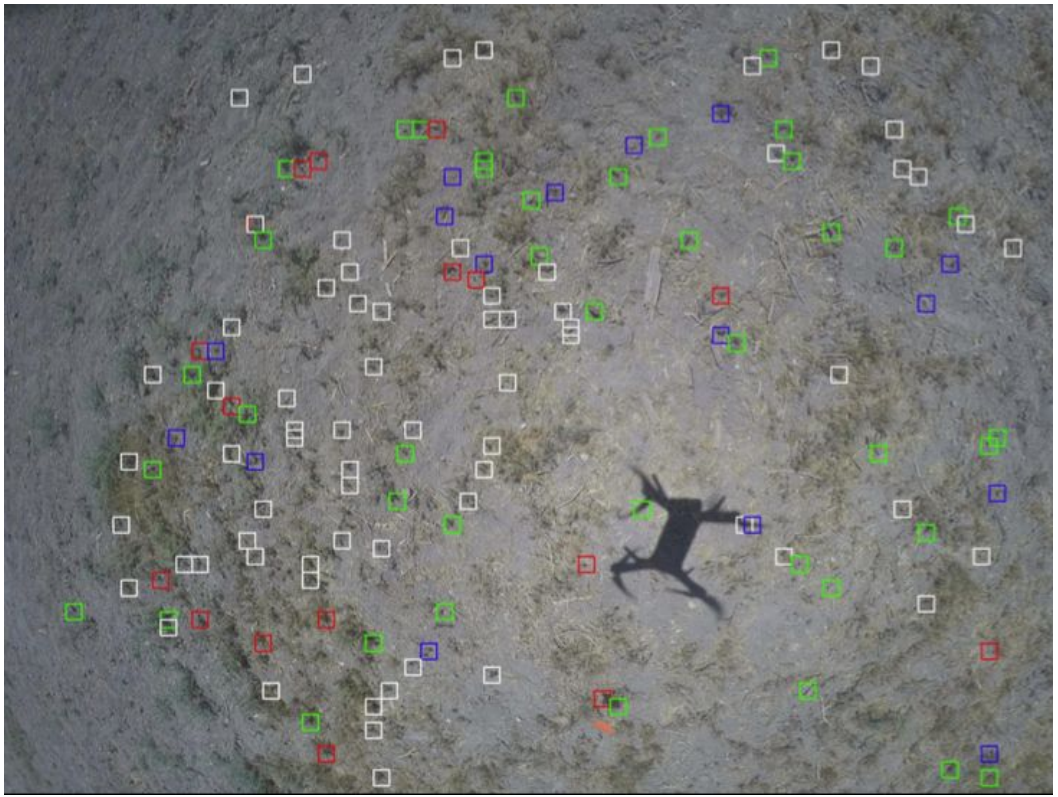


Figure 1. Flowchart of the classification tasks for GZ2, beginning at the top centre. Tasks are colour-coded by their relative depths in the decision tree. Tasks outlined in brown are asked of every galaxy. Tasks outlined in green, blue, and purple are (respectively) one, two or three steps below branching points in the decision tree. Table 2 describes the responses that correspond to the icons in this diagram.



Source: Sander Dieleman 2014 - http://benanne.github.io/2014/04/05/galaxy-zoo.html

Deep learning



Source: Sravanthi Sinha, Deep Learning Summit, Singapore 2016

DL

ICRAR International Centre for Radio Astronomy Research

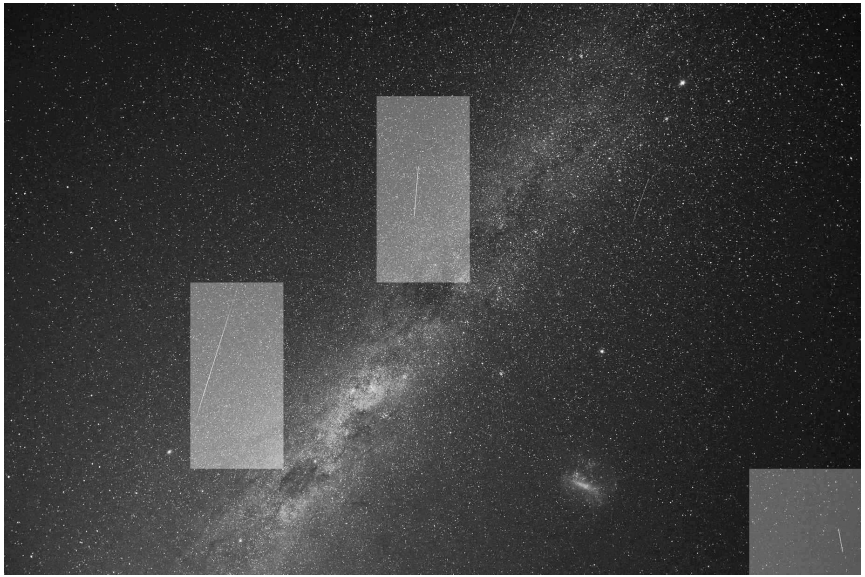
Curtin University

FIREBALLS in the sky

Curtin University
THE CURTIN INSTITUTE FOR COMPUTATION

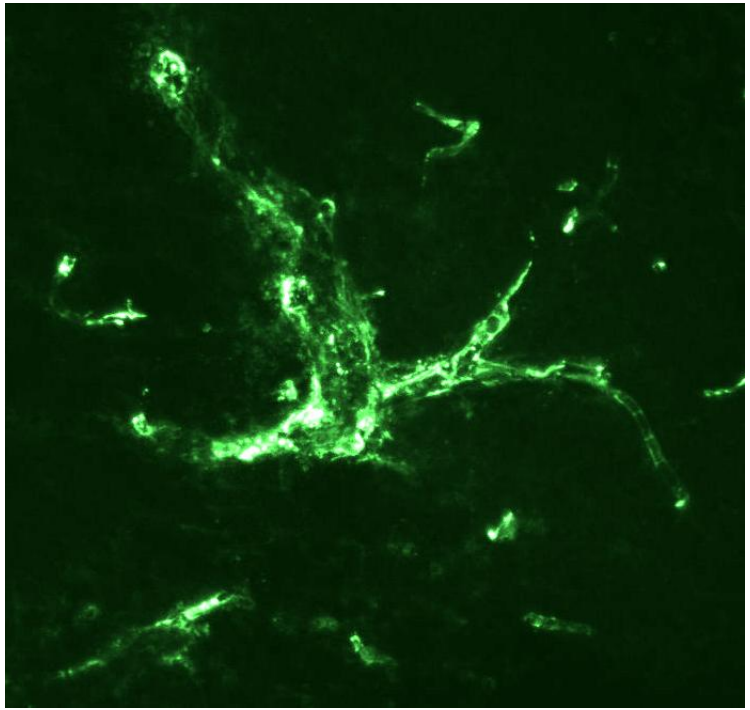
Detection of transient optical events from a remote autonomous camera in the West Australian Desert

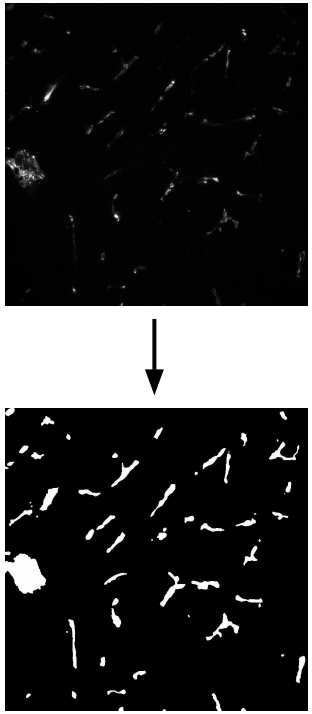
Curtin Institute for Radio Astronomy (CIRA), Desert Fireball Network (DFN)



Deep learning

Quantitation of cerebral capillary immunofluorescence images
School of Public Health, Curtin Health Innovation Research Institute (CHIRI)





Expert researcher
1 month to measure 1,000 images manually

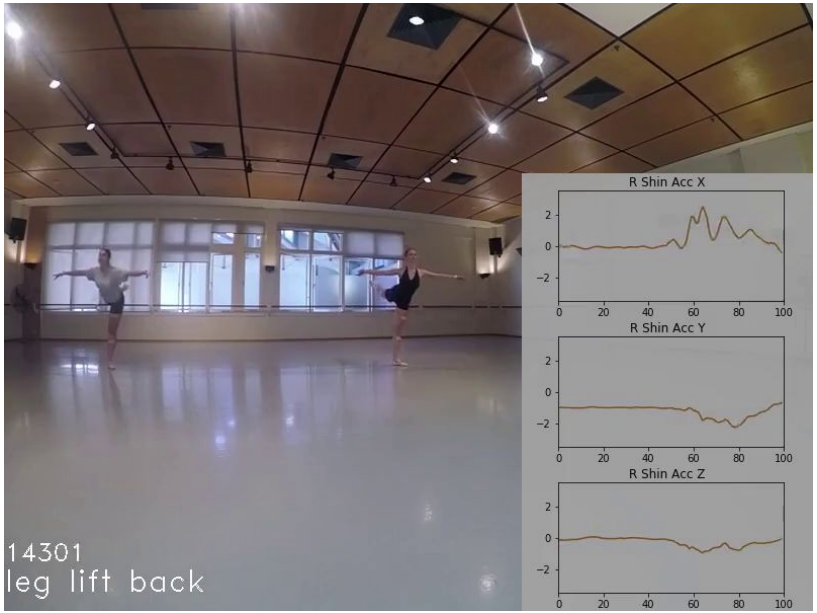
Trained model
1 ½ minutes to measure 1,000 images automatically

Deep learning

Activity recognition of ballet dancers using wearable sensors

School of Physiotherapy and Exercise Science, WA Performing Arts Academy

Curtin University
THE CURTIN INSTITUTE FOR COMPUTATION



Questions