CBS COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

Business Analytics and Data-Driven Decision Making

Foundations of Predictive Analytics: Machine Learning Algorithms

Raghava Mukkamala

Associate Professor & Director,
Centre for Business Data Science
Copenhagen Business School, Denmark
Email: rrm.digi@cbs.dk, Centre: https://cbsbda.github.io/

Some slides have been taken and adapted from the course
CPSC 340: Machine Learning and Data Mining
https://www.cs.ubc.ca/~schmidtm/Courses/340-F22/L30.pdf

EQUIS ACCREDITED · AACSB ACCREDITED · ASSOCIATION AMBA ACCREDITED · CEMS · PIM

Outline

- Fundamentals of Machine Learning
- Precision Measures
- Supervised: Decision Trees, Random Forests
- Unsupervised: Clustering
- Unsupervised: Density-based Based Clustering
- Supervised: Artificial Neural Networks and Deep Learning

Fundamentals of Learning

CBS

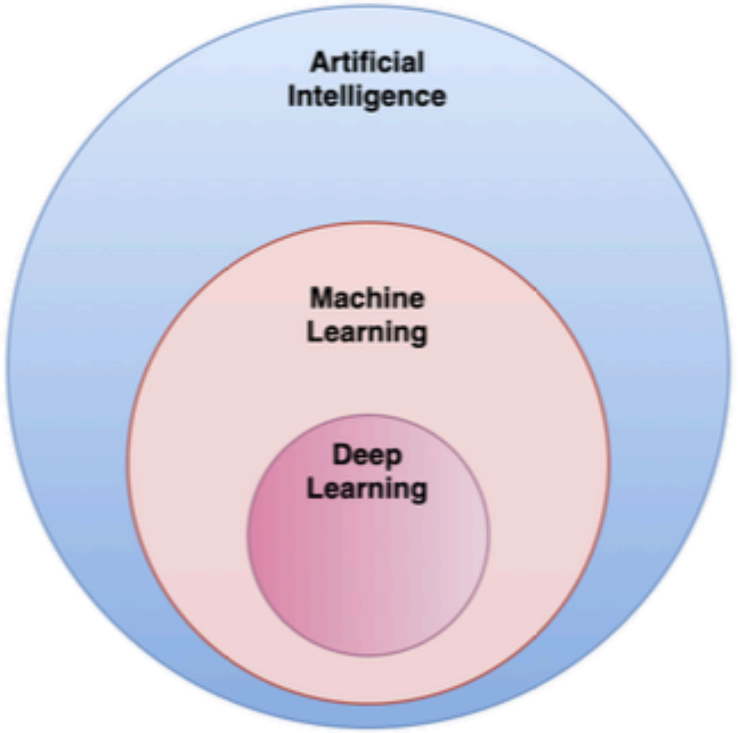
What is Machine Learning

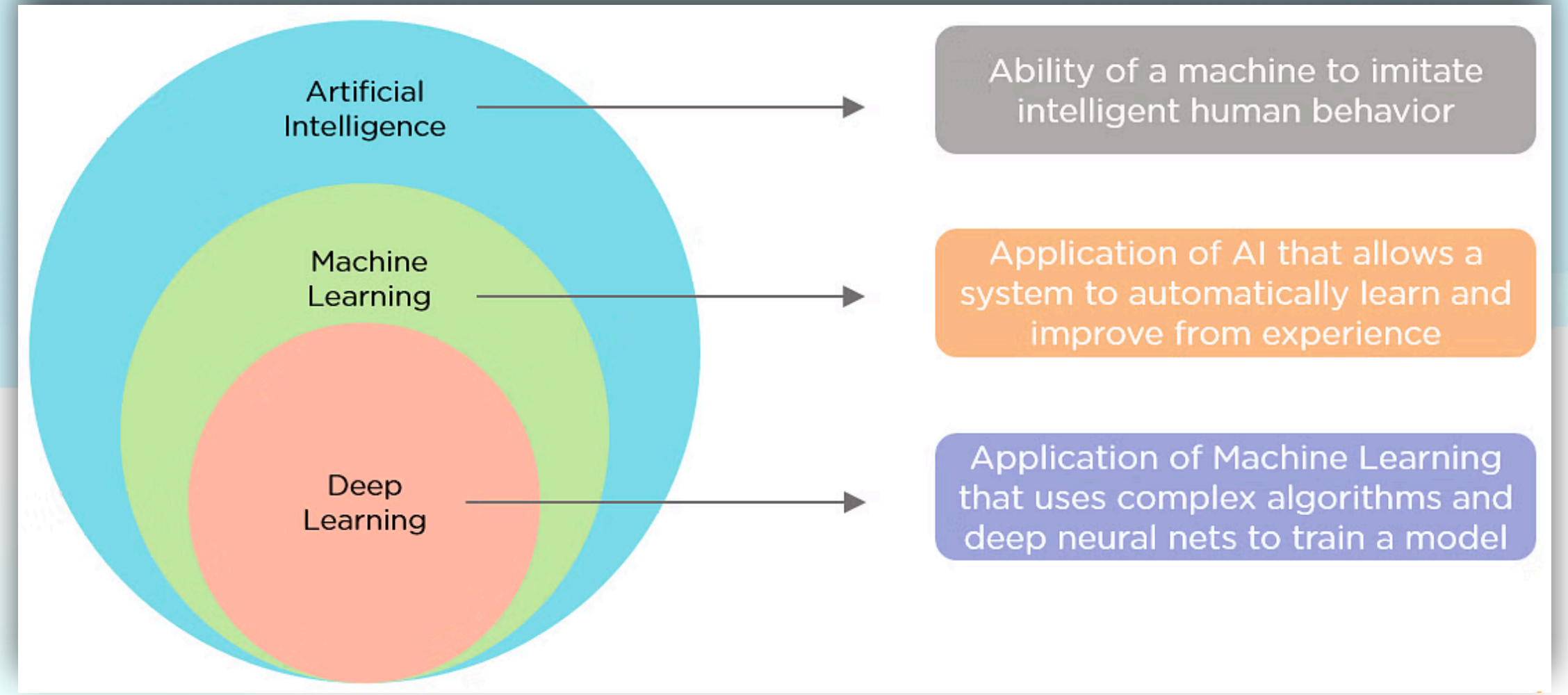
A part of artificial intelligence (AI)

Machine learning (ML) is the study of algorithms that can

1) learn from data
2) make predictions on new data

Deep learning (DL) is a subfield of ML

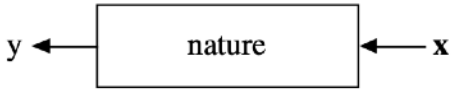




Source: https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/ai-vs-machine-learning-vs-deep-learning

Statistical Modeling: The Two Cultures

What is the difference between statistics and machine learning?

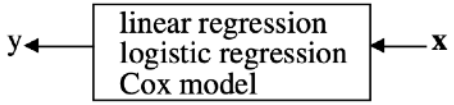


There are two goals in analyzing the data:

Prediction. To be able to predict what the responses are going to be to future input variables;
Information. To extract some information about how nature is associating the response variables to the input variables.

There are two different approaches toward these goals:

The values of the parameters are estimated from the data and the model then used for information and/or prediction. Thus the black box is filled in like this:

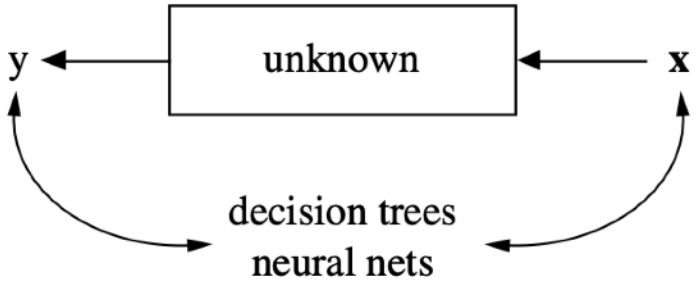


Model validation. Yes–no using goodness-of-fit tests and residual examination.
Estimated culture population. 98% of all statisticians.

The Data Modeling Culture

The Algorithmic Modeling Culture

The analysis in this culture considers the inside of the box complex and unknown. Their approach is to find a function $f(\mathbf{x})$—an algorithm that operates on $\mathbf{x}$ to predict the responses $\mathbf{y}$. Their black box looks like this:



Model validation. Measured by predictive accuracy.
Estimated culture population. 2% of statisticians, many in other fields.

The Algorithmic Modeling Culture

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, *16*(3), 199-231.
https://projecteuclid.org/journals/statistical-science/volume-16/issue-3/Statistical-Modeling--The-Two-Cultures-with-comments-and-a/10.1214/ss/1009213726.full

Fundamentals of Learning

- Machine learning: building models of data that helps to understand the data and its underlying hidden patterns.

  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning

Supervised learning

- Training data includes labels
- Example algorithms:
  - Linear regression, logistic regression
  - Decision tree, random forest
  - Support vector machines (SVM)
  - Neural network; generative models (GANs)

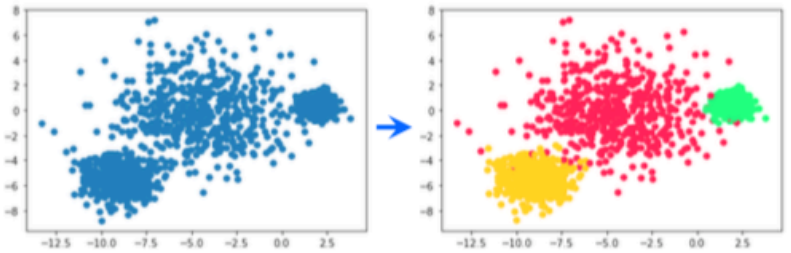
Stanford Dogs dataset
(120 breeds)

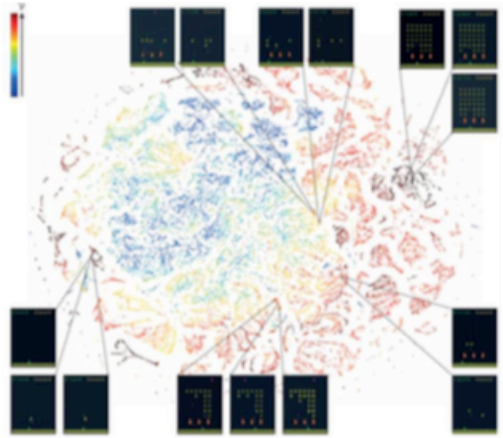
Classification and regression

- Differ in the types of labels (dependent variables)
- Classification
  - Discrete/categorical labels
  - Image labeling, sentiment prediction
- Regression
  - Continuous labels
  - Stock prices, temperature, sales volume

Unsupervised learning

- Labels are not provided
- Example algorithms
  - Clustering: k-means, hierarchical
  - Dimensionality reduction: principal components analysis (PCA), t-SNE

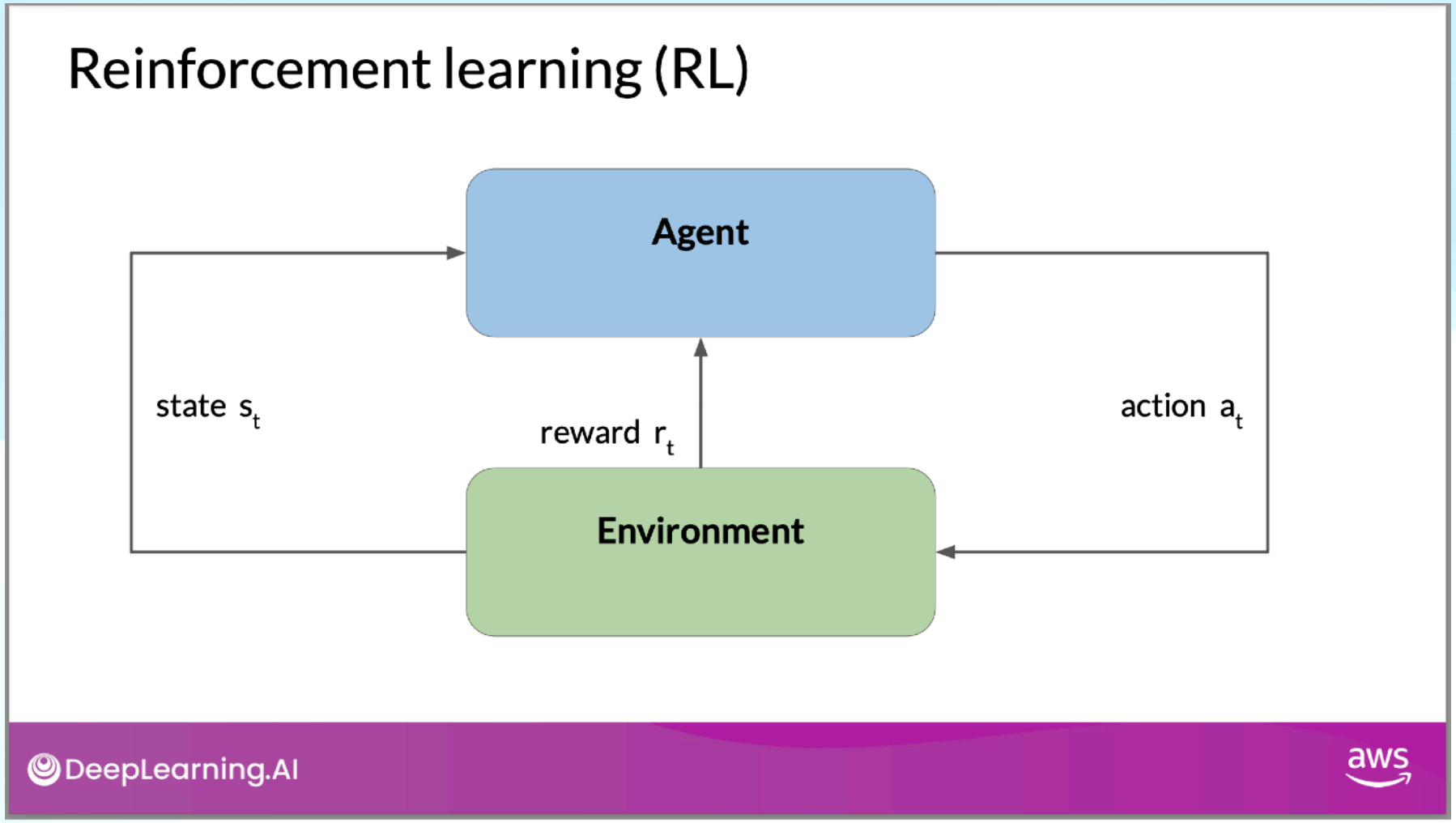




(Mnih, Kavukcuoglu et al., 2015)

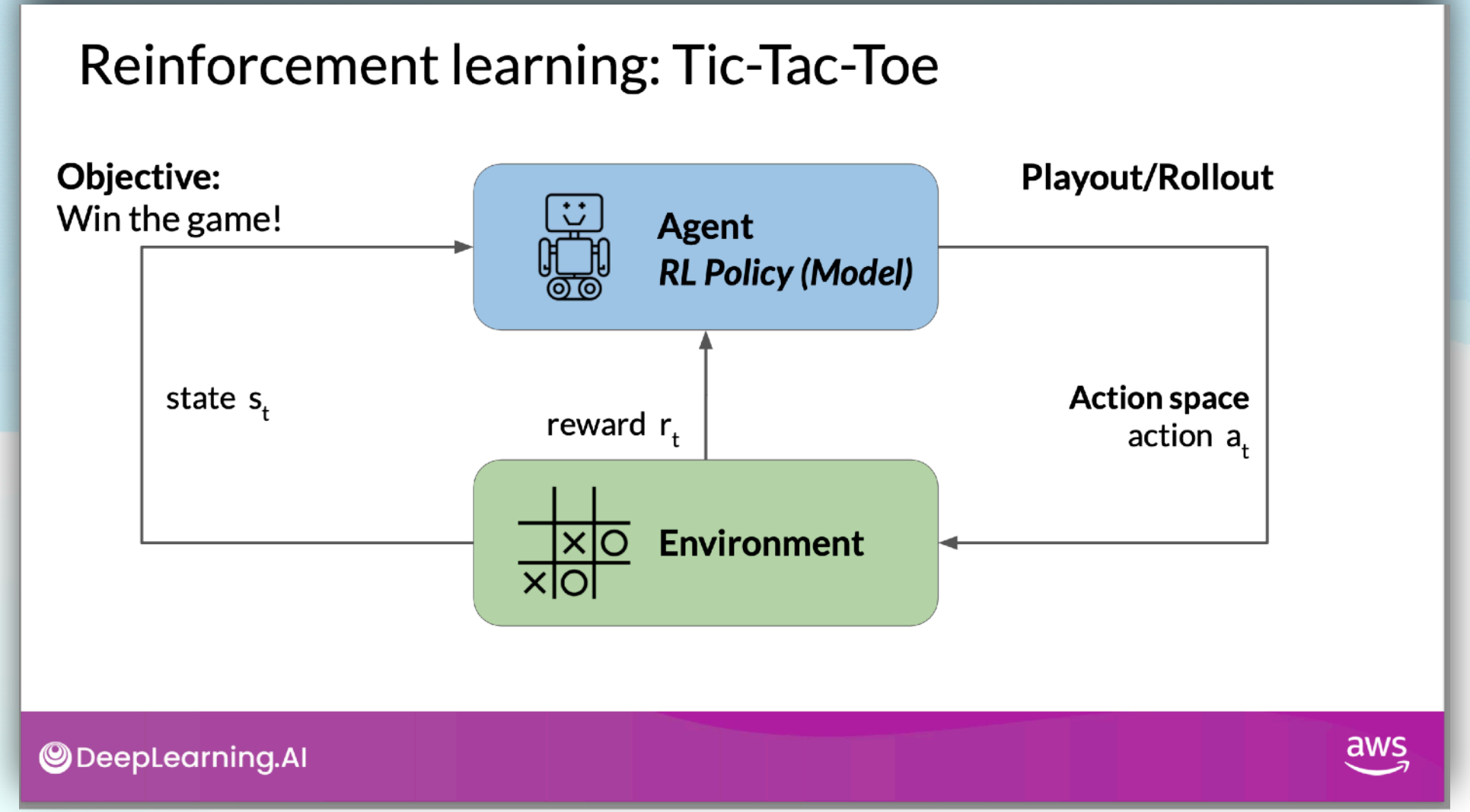
Customer Segmentation,
Product Segmentation etc.

Reinforcement learning



Source: DeepLearning.ai

Reinforcement learning



Source: DeepLearning.ai

Reinforcement learning

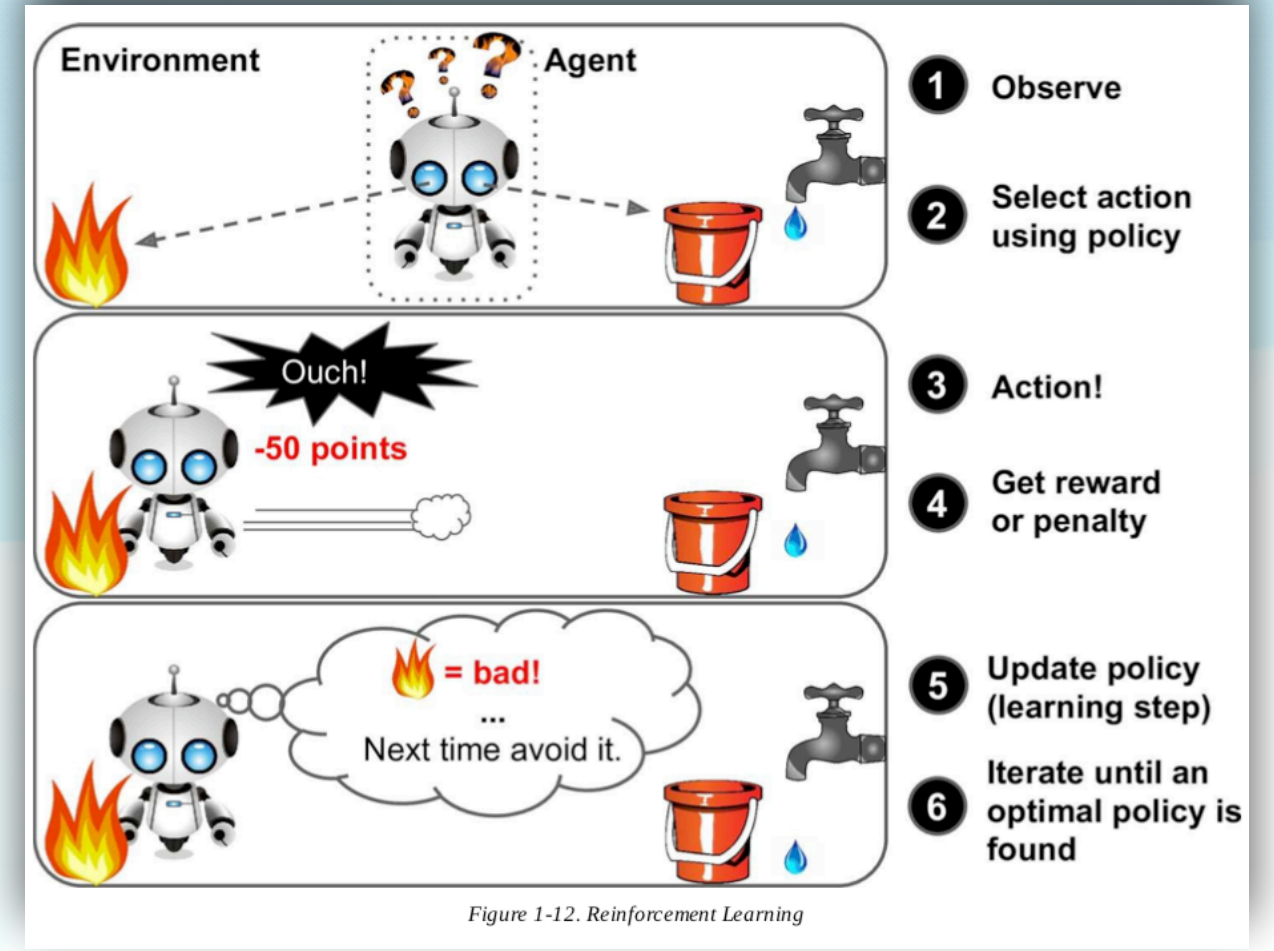
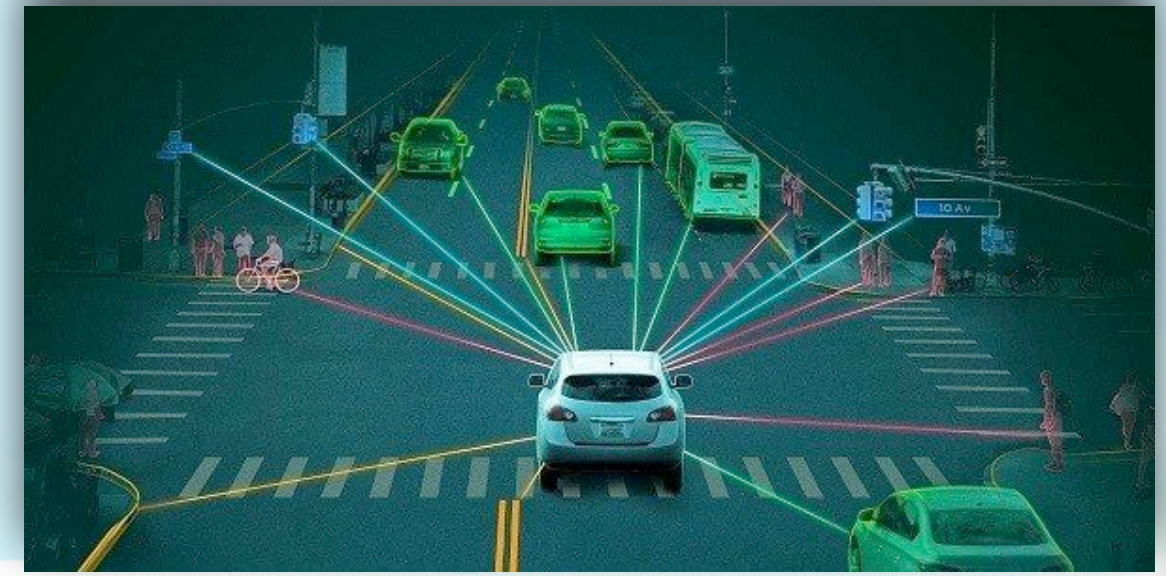
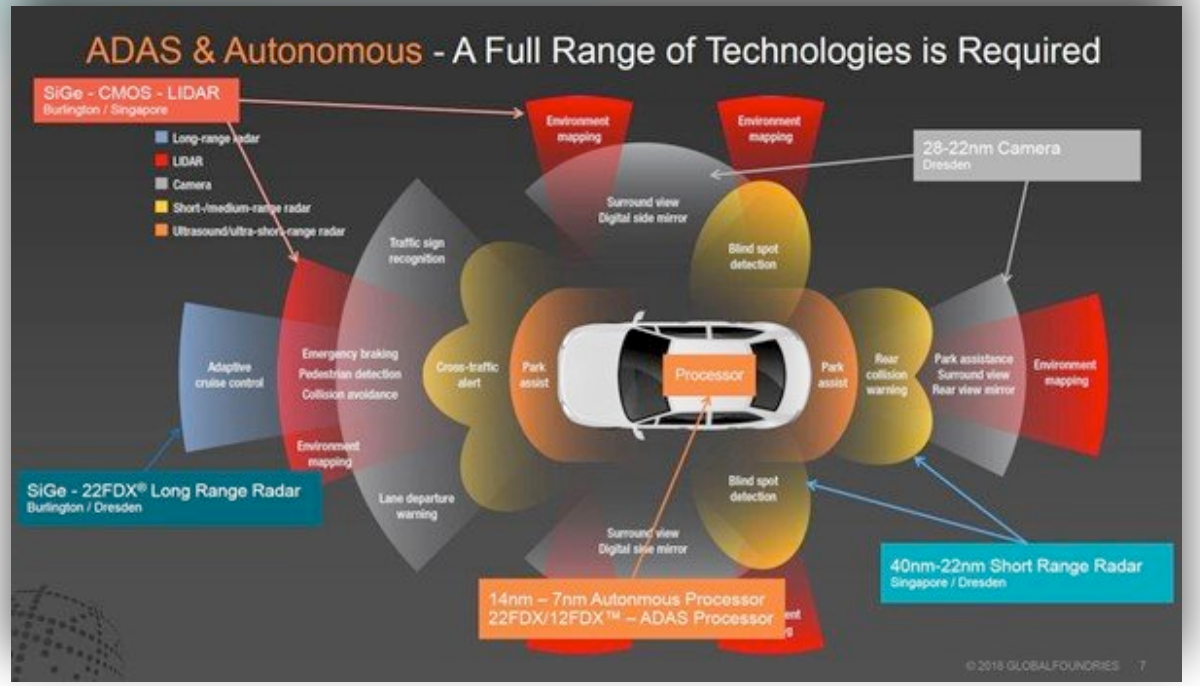


Figure 1-12. Reinforcement Learning

Source: Aurélien Géron "Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems"

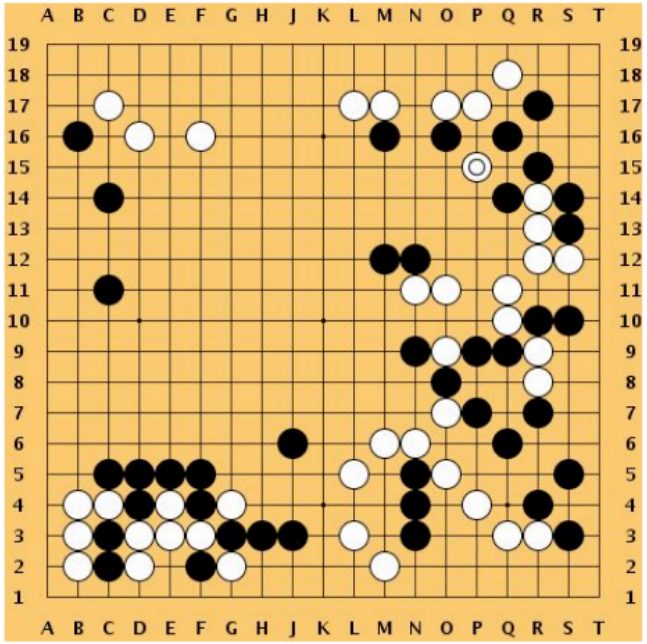
Reinforcement learning - Self Driving car





Reinforcement learning

Go



Objective: Win the game!

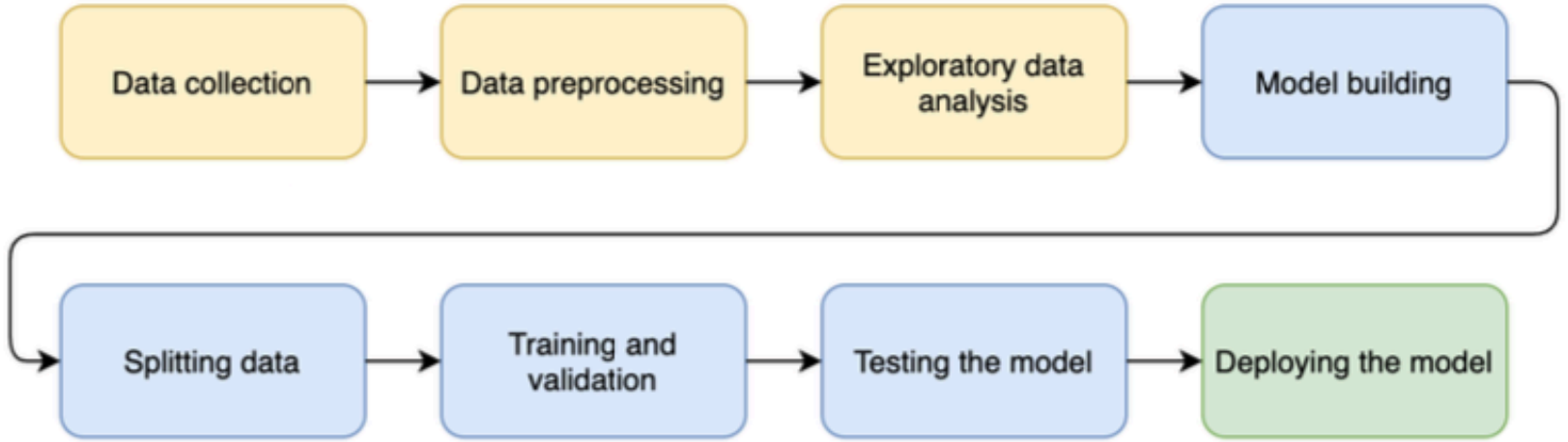
State: Position of all pieces
Action: Where to put the next piece down
Reward: 1 if win at the end of the game, 0 otherwise

Source : http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture14.pdf

https://www.nytimes.com/2017/05/23/business/google-deepmind-alphago-go-champion-defeat.html

https://en.wikipedia.org/wiki/Go_and_mathematics

An example of a workflow in a ML project



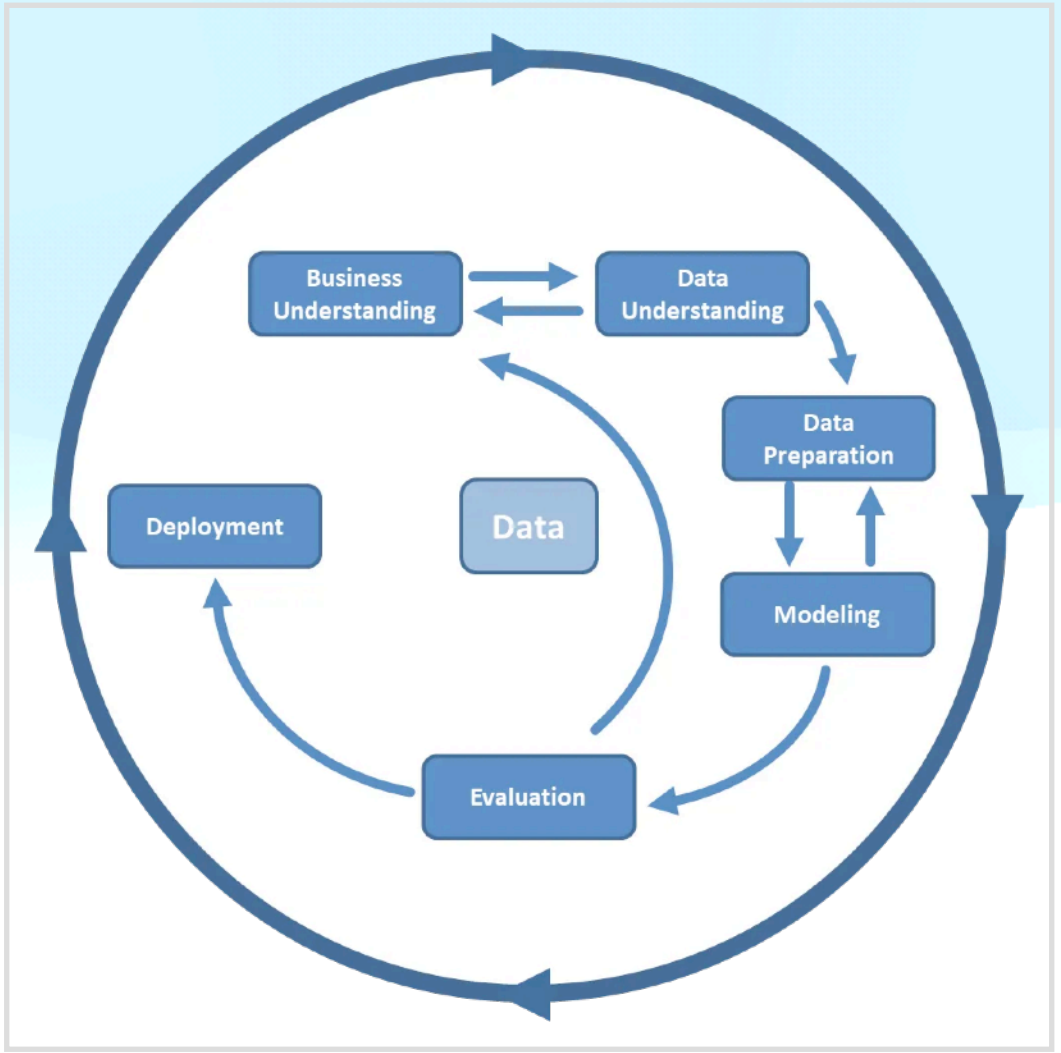
The yellow boxes represent preparation and blue boxes the development of the ML model

In reality the workflow will be more iterative and you will go back and forth between the steps

Source: Sippo Rossi, Introduction to Machine Learning

CRISP-DM Methodology

- One of the most popular academic frameworks for the Data Science Projects
- Several iterations between
  - Business Understanding and Data Understanding
  - Data Preparation and Modeling
- Based on the evaluation, you might choose to repeat whole process



https://towardsdatascience.com/using-crisp-dm-to-grow-as-data-scientist-a07ce3fd9d56

Precision Measures

Metrics to Measure Performance

- How do we know whether the algorithm did a good job or not?
- Classification metrics
  - Confusion matrix, precision, recall
  - F1 score and accuracy

- Regression metrics
  - Mean squared error
  - Root mean squared error

Precision and Recall

	Predicted class POSITIVE (spam ✉)	Predicted class NEGATIVE (normal 🕵)	
Actual class POSITIVE (spam ✉)	TRUE POSITIVE (TP) ✉ ✉ 320	FALSE NEGATIVE (FN) ✉ 🕵 43	*Recall* $= \frac{TP}{TP + FN} = \frac{320}{320 + 43} = 0.882$
Actual class NEGATIVE (normal ✉)	FALSE POSITIVE (FP) ✉ ✉ 20	TRUE NEGATIVE (TN) ✉ 🕵 538	
	Precision $= \frac{TP}{TP + FP} = \frac{320}{320 + 20} = 0.941$		

- We have 921 emails containing spam and normal with actual labels by human
- We ask the algorithm to predict the label for each email: -> predicted class/ label: spam or normal
- Actual: 363 spam + 558 normal
- Predicted: 340 spam + 581 normal
- Each email has
  - an actual label (spam/normal)
  - a predicted label (spam/normal)
- Based on we can put each email into one of the 4 buckets (left side)

https://www.knime.com/blog/from-modeling-to-scoring-confusion-matrix-and-class-statistics

Which One Is Important?

Positive: having COVID

Negative: Healthy

COVID Rapid Test

Reality	Positive	Negative
Positive	**True Positives** Person with COVID Predicted as COVID	**False Negatives** Person with COVID Predicted as Healthy
Negative	**False Positives** Healthy Person Predicted as COVID	**True Negatives** Healthy Person Predicted as Healthy

Recall

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Precision

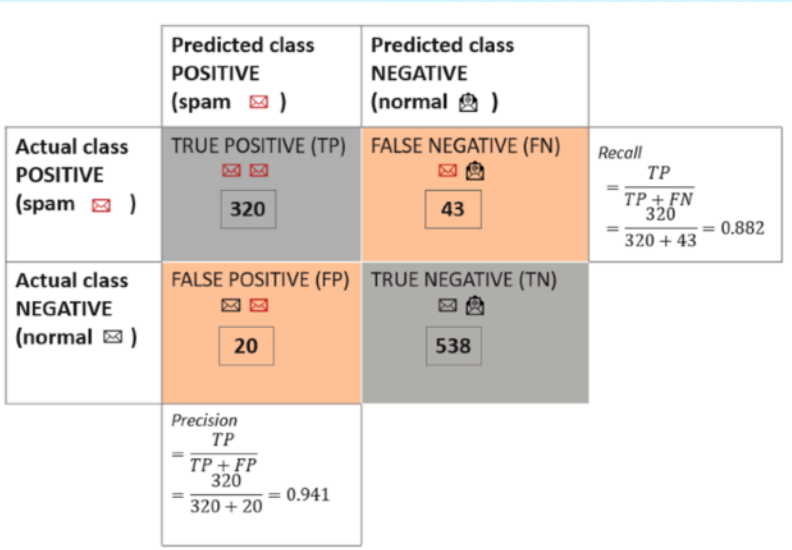
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Which is preferable?

High precision or
High recall?

F_1 score and accuracy

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$



Accuracy: all correctly classified examples (TN & TP)

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}}$$

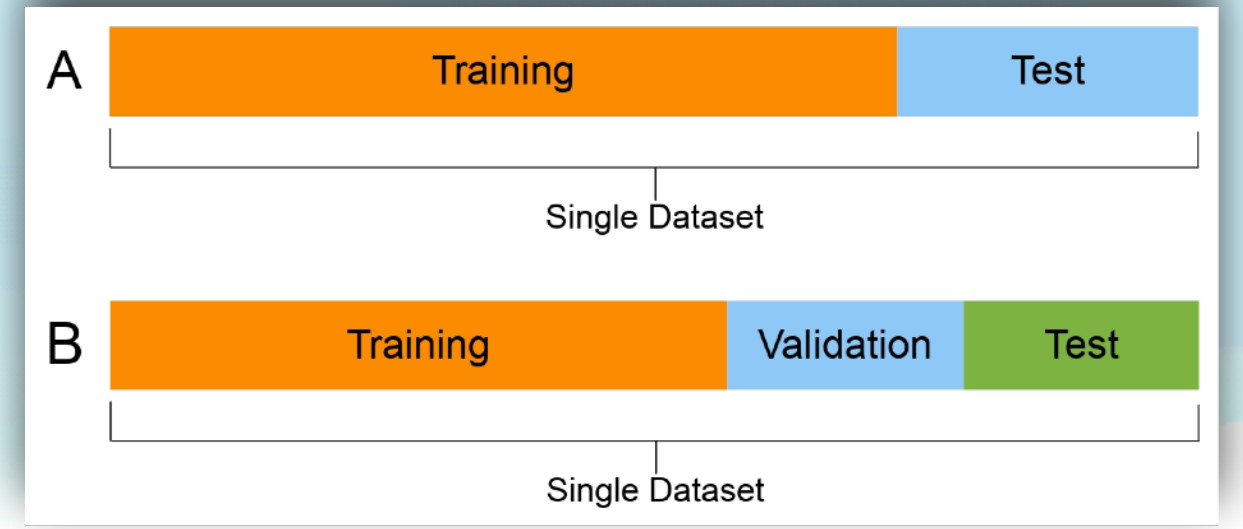
Regression metrics

- Mean squared error: $\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}(y_{\mathrm{pred}} - y_{\mathrm{true}})^2$
- Root mean squared error: $\mathrm{RMSE} = \sqrt{\mathrm{MSE}}$
  - MSE/RMSE works very well
- Mean absolute error: $\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}|y_{\mathrm{pred}} - y_{\mathrm{true}}|$
  - Less sensitive to outliers than MSE/RMSE

Splitting data

Dataset is divided into 2 or 3 subsets:

- Training set, to train the model
- Validation set, to tune the hyperparameters
- Test set, to confirm the results



Never train and test a model with the same data

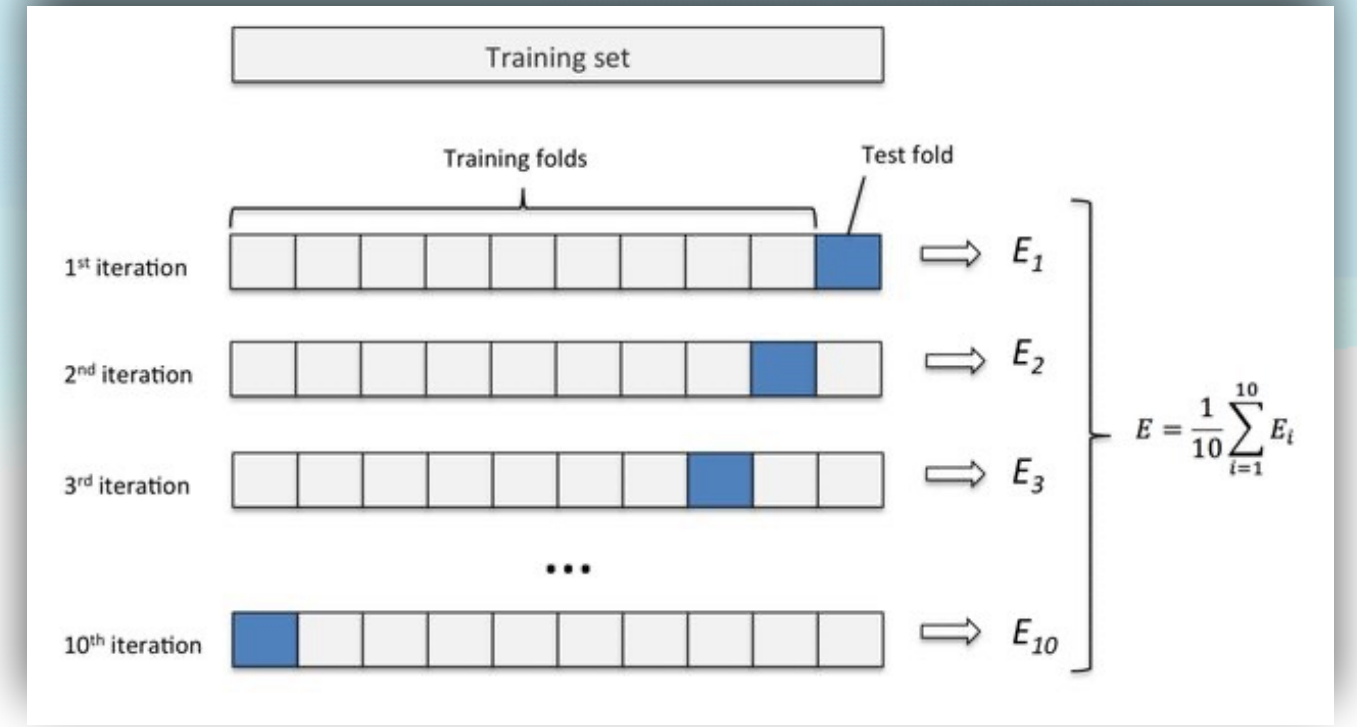
```
X_train, X_test, y_train, y_test =
        train_test_split(…)
```


Image from https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets#/media/File:ML_dataset_training_validation_test_sets.png

Cross-validation

- A less biased or less optimistic estimate of the model than the train/test split

- k-folds cross-validation, where k is the number of splits (e.g., 10)




```
KFold(n_splits=10, random_state=None, shuffle=False)
```


Image from Introduction to Support Vector Machines and Kernel Methods, Ashfaque and Iqbal, 2019

Parameters

There are two types of parameters:

1) Parameters
2) Hyperparameters

The model parameters are adjusted automatically as you fit (train) a ML model

Hyperparameters are parameters that control how a ML model learns and need to be adjusted by the user

Grid search

Grid search is a way to systematically tune hyperparameters

Grid search illustrated for 2 hyperparameters for Logistic Regression: Alpha and C

The values in the matrix are the accuracy of the model with each hyperparameter combination

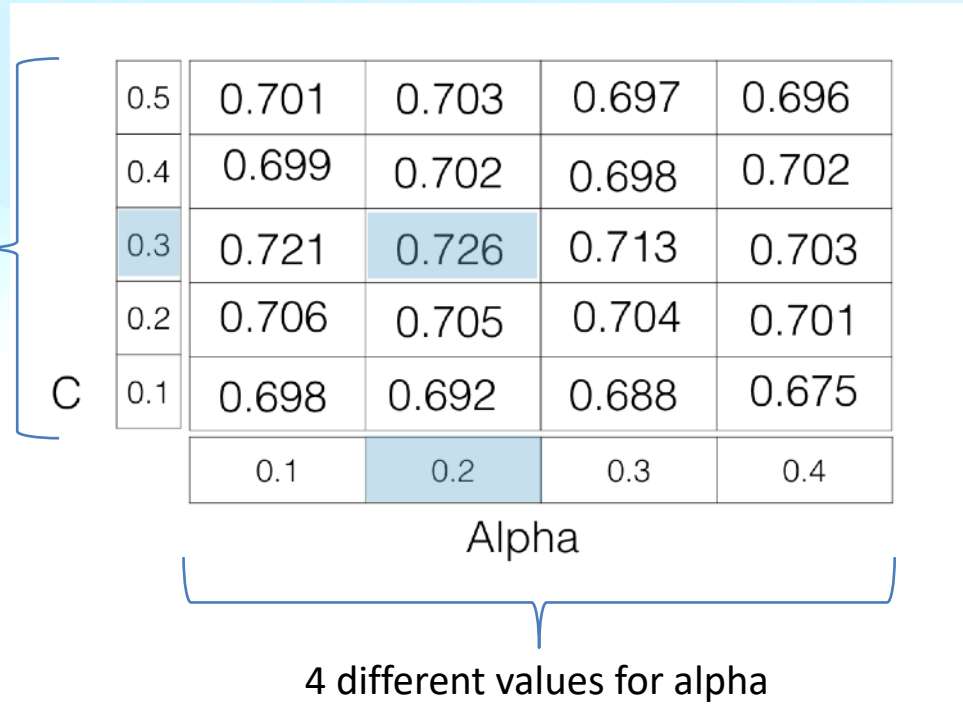


Image by Datacamp

Decision Trees

CBS

Motivating Example

Predict if John will play tennis

- Hard to guess
- Divide & conquer:
  - split into subsets
  - are they pure? (all yes or all no)
  - if yes: stop
  - if not: repeat
- See which subset new data falls into

Training examples: **9 yes / 5 no**

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No

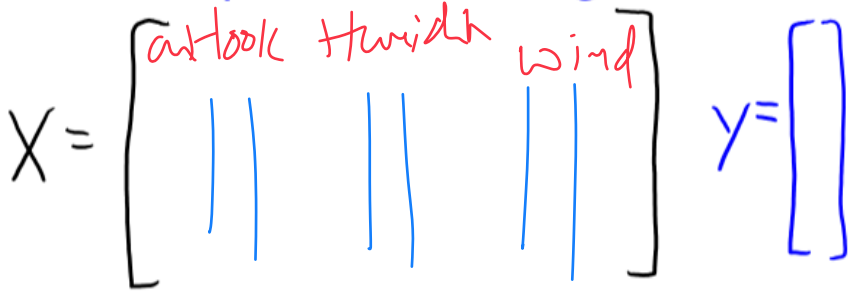
New data:

D15	Rain	High	Weak	?

Source: Victor Lavrenko and Charles Sutton
http://www.inf.ed.ac.uk/teaching/courses/iaml/2011/slides/dt.pdf

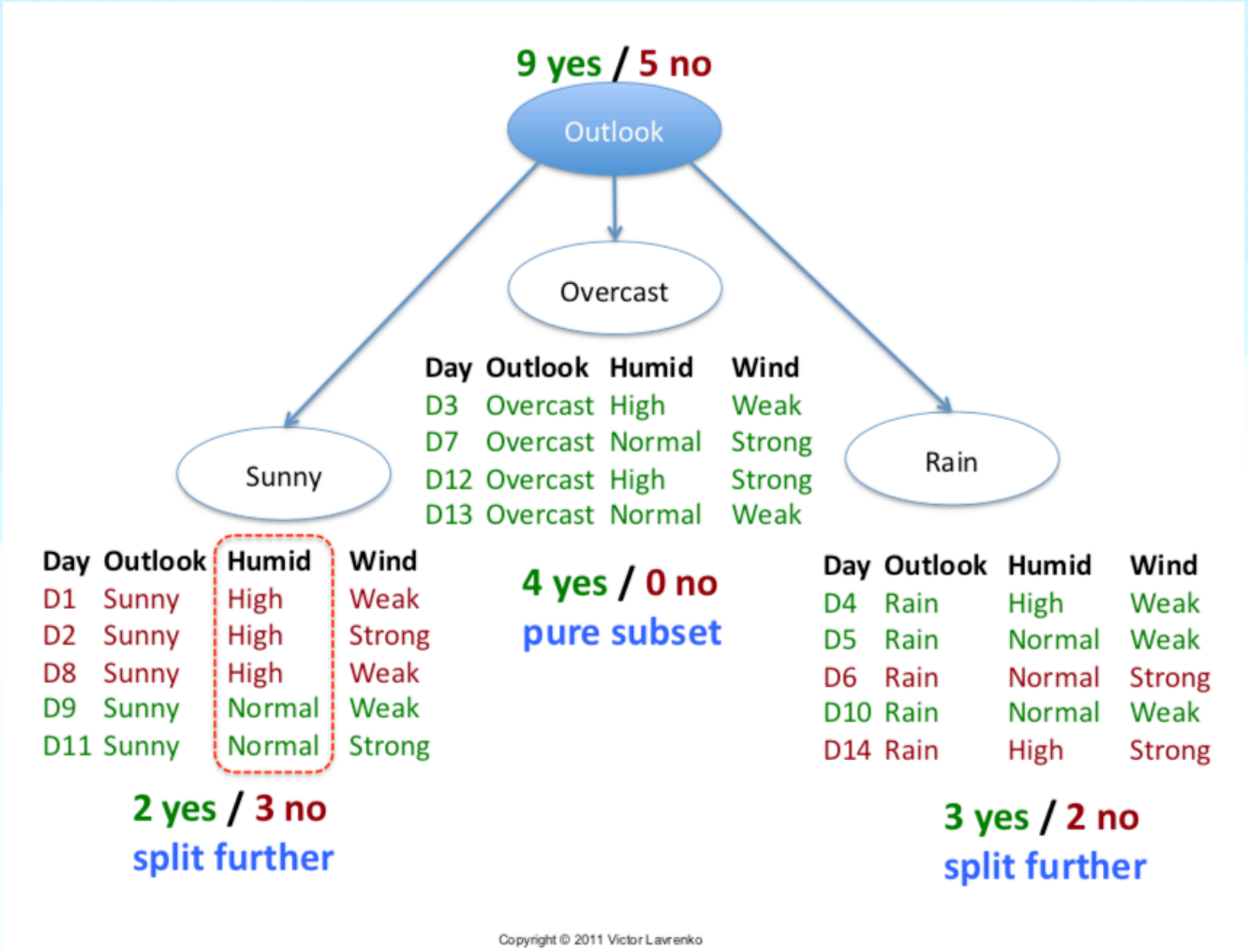
Motivating Example

- We're going to revisit supervised learning:



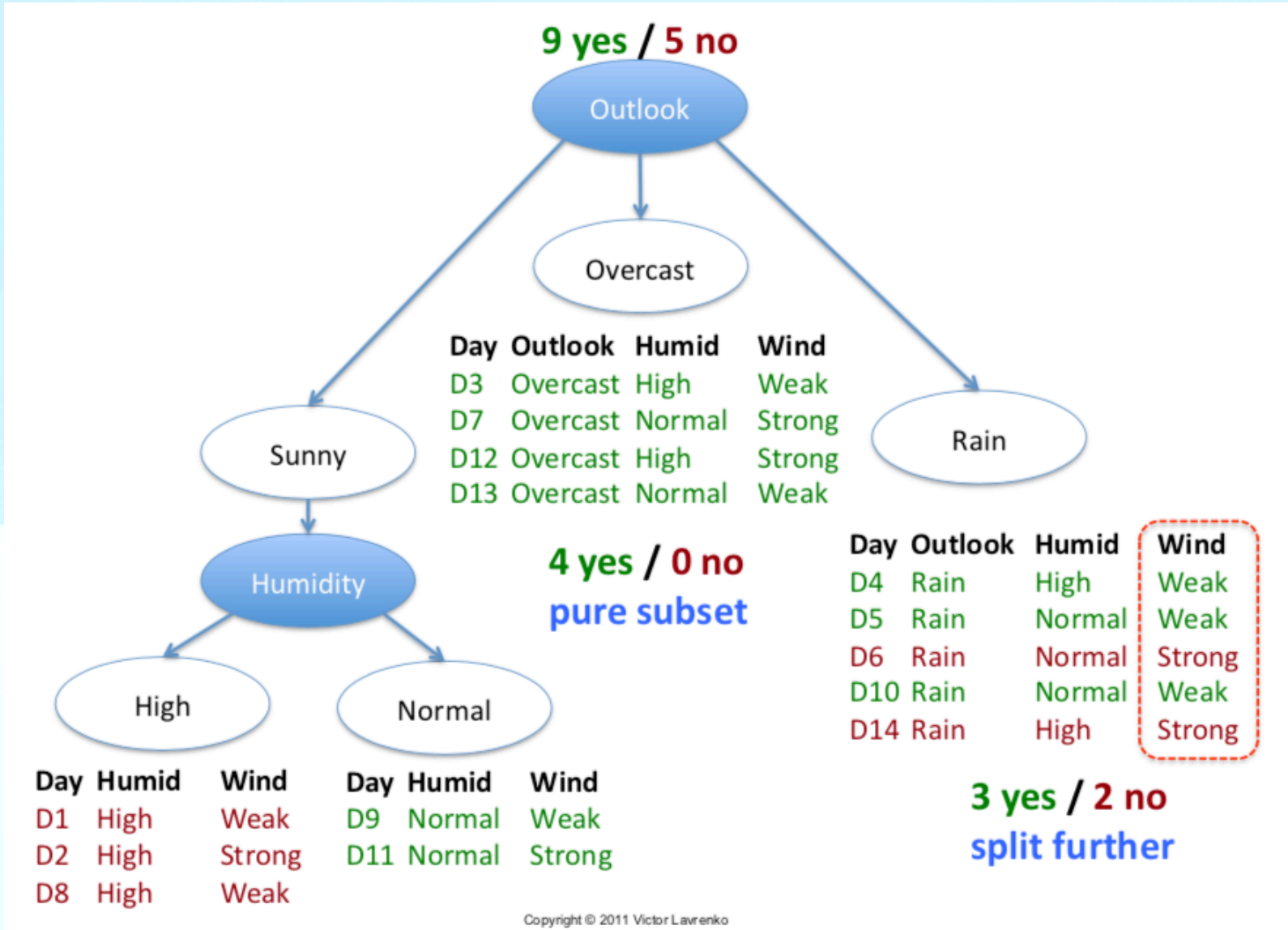
- Previously, we considered classification:
  - We assumed y_i was discrete: y_i = Play or y_i = No Play

Motivating Example - 1



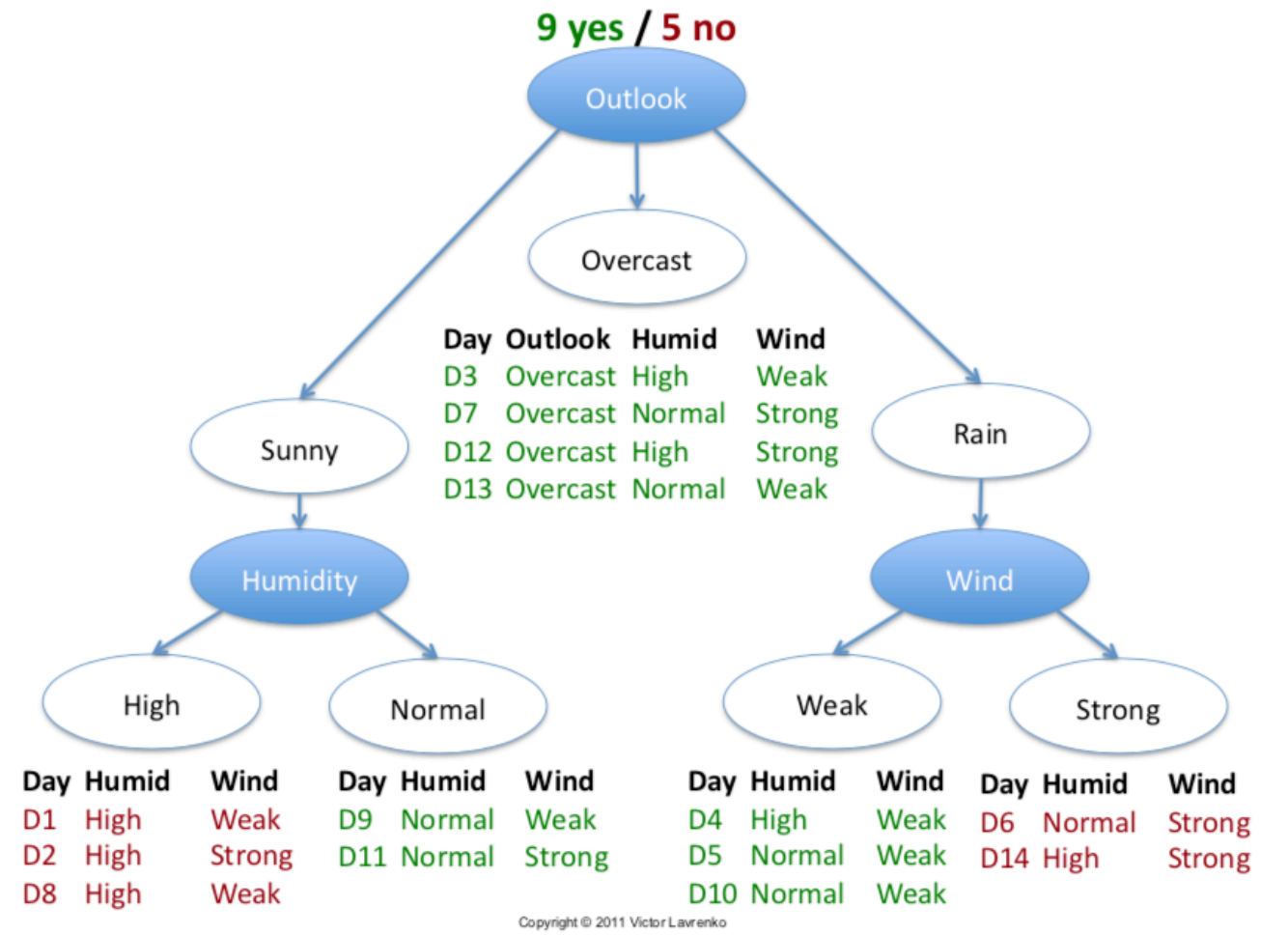
Source: Victor Lavrenko and Charles Sutton
http://www.inf.ed.ac.uk/teaching/courses/iaml/2011/slides/dt.pdf

Motivating Example - 2



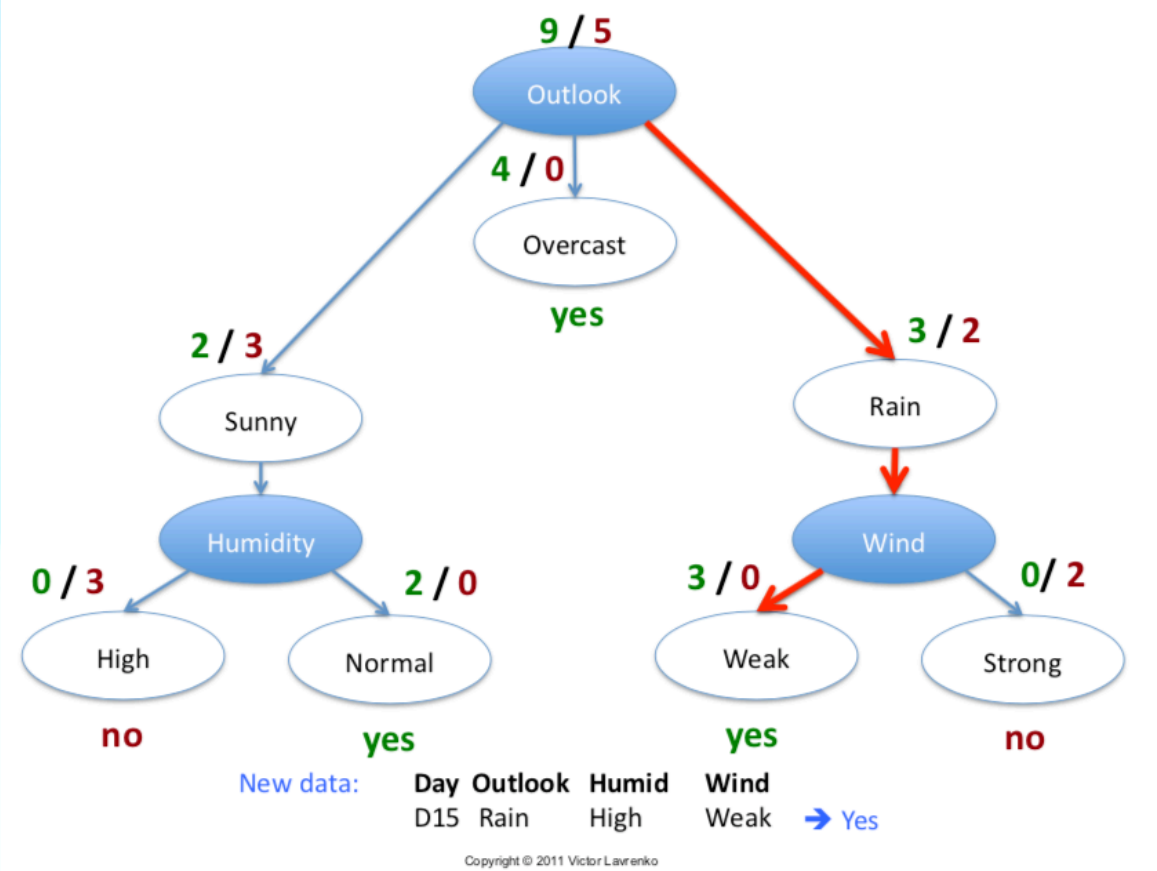
Source: Victor Lavrenko and Charles Sutton
http://www.inf.ed.ac.uk/teaching/courses/iaml/2011/slides/dt.pdf

Motivating Example -3



Source: Victor Lavrenko and Charles Sutton
http://www.inf.ed.ac.uk/teaching/courses/iaml/2011/slides/dt.pdf

Motivating Example - 4



Source: Victor Lavrenko and Charles Sutton
http://www.inf.ed.ac.uk/teaching/courses/iaml/2011/slides/dt.pdf

Algorithm

- Split the training set based on:
  - Feature k
  - Threshold t_k
- …that maximize purity of the resulting subsets measured by cost $J(k, t_k)$
- Repeat until a stopping condition is met

Cost

$$J(k, t_k) = \frac{m_{\text{left}}}{m} G_{\text{left}} + \frac{m_{\text{right}}}{m} G_{\text{right}}$$

- m: total instances for current split
- $m_{\text{left/right}}$: instances in left/right subset after the split
- $G_{\text{left/right}}$: impurity of the left/right subset

Impurity: Gini and entropy

- Gini: $G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2$
  - $p_{i,k}$: ratio of instances of class k in node i

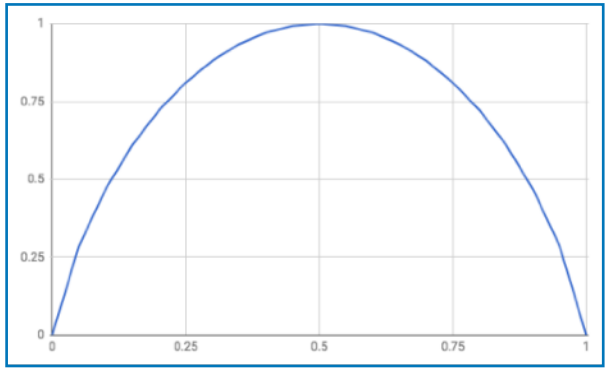
$G = 1 - [\ (3/7)^2+(1/7)^2+(2/7)^2+(1/7)^2\ ] =$ **0.69**

$G = 1 - [\ (6/7)^2+(1/7)^2\ ] =$ **0.24**

- Entropy: $H_i = -\sum_{\substack{k=1 \\ p_{i,k} \neq 0}} p_{i,k} \log p_{i,k}$

What Is the Gini Index?

The Gini index or Gini coefficient is a statistical measure of distribution developed by the Italian statistician Corrado Gini in 1912. It is often used as a gauge of economic inequality, measuring income distribution or, less commonly, wealth distribution among a population. The coefficient ranges from 0 (or 0%) to 1 (or 100%), with 0 representing perfect equality and 1 representing perfect inequality. Values over 1 are theoretically possible due to negative income or wealth.



Random Forests and XGBoost

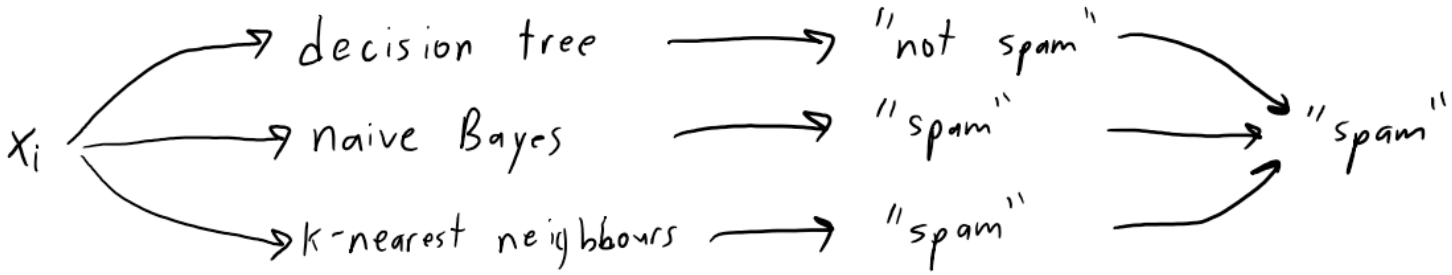
CBS

Ensemble Methods

- Ensemble methods are classifiers that have classifiers as input.
  - Also called “meta-learning”.
- They have the best names:
  - Averaging.
  - Boosting.
  - Bootstrapping.
  - Bagging.
  - Cascading.
  - Random Forests.
  - Stacking.
- Ensemble methods often have higher accuracy than input classifiers.

Averaging

- Input to averaging is the predictions of a set of models:
  - Decision trees make one prediction.
  - Naïve Bayes makes another prediction.
  - KNN makes another prediction.
- Simple model averaging:
  - Take the mode of the predictions (or average if probabilistic).

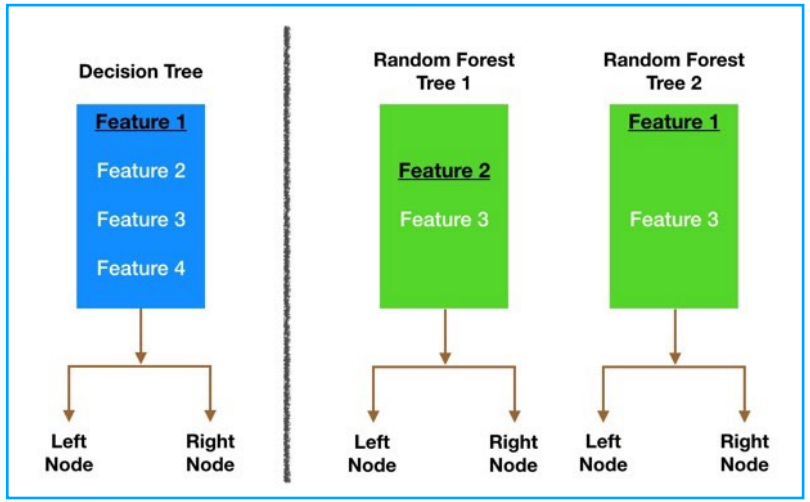


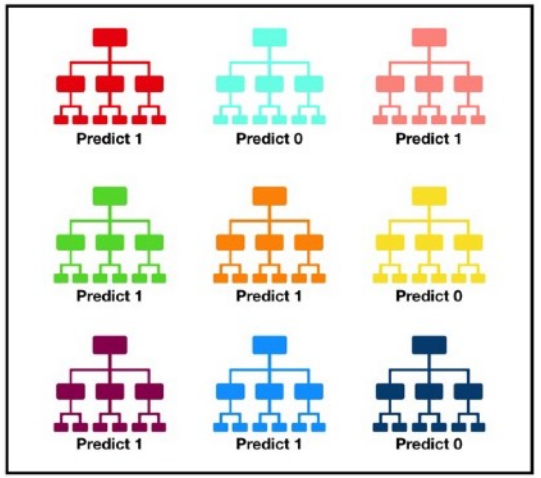
Random Forests

- Random forests average a set of deep decision trees.
  - Tend to be one of the best "out of the box" classifiers.
    - Often close to the best performance of any method on the first run.
  - And predictions are very fast.

- Do deep decision trees make independent errors?
  - No: with the same training data you'll get the same decision tree.

- Two key ingredients in random forests:
  - Bootstrapping.
  - Random trees.

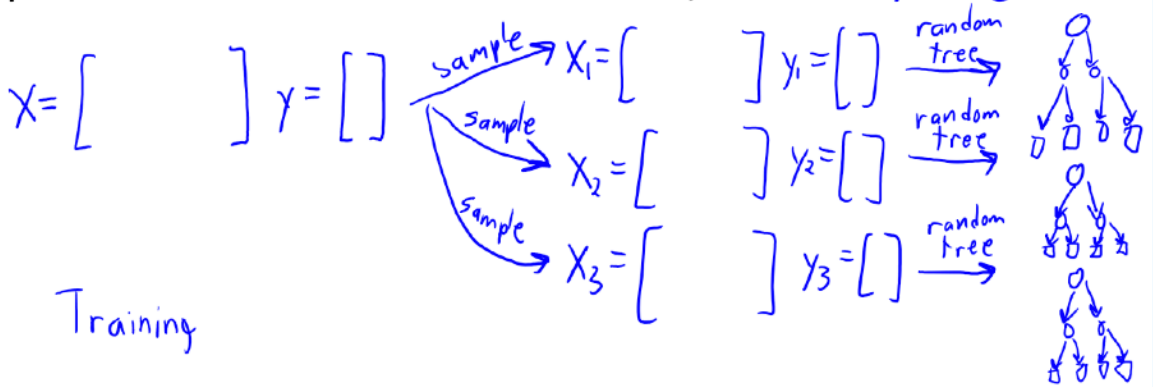




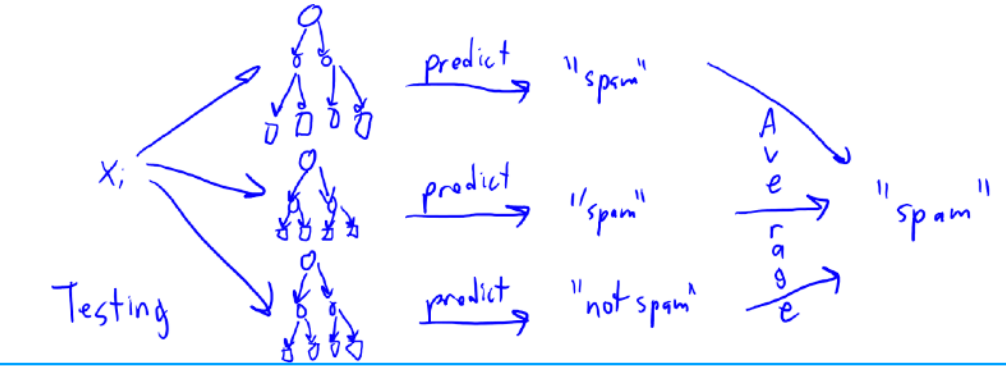
Tally: Six 1s and Three 0s
Prediction: 1

Random Forests

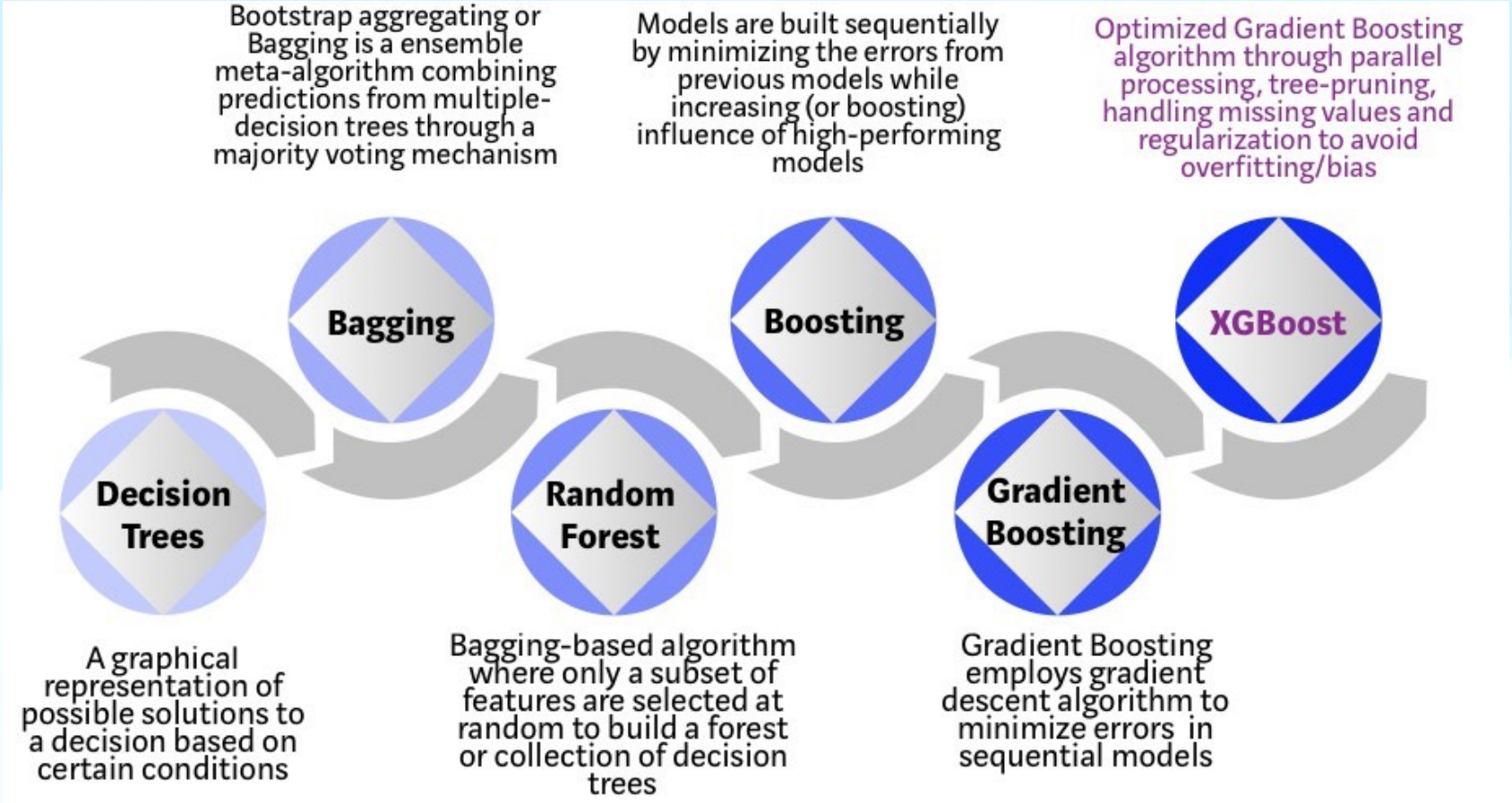
- Random forests are one of the best 'out of the box' classifiers.
- Fit deep decision trees to random bootstrap samples of data, base splits on random subsets of the features, and classify using mode.



- Random forests are one of the best 'out of the box' classifiers.
- Fit deep decision trees to random bootstrap samples of data, base splits on random subsets of the features, and classify using mode.

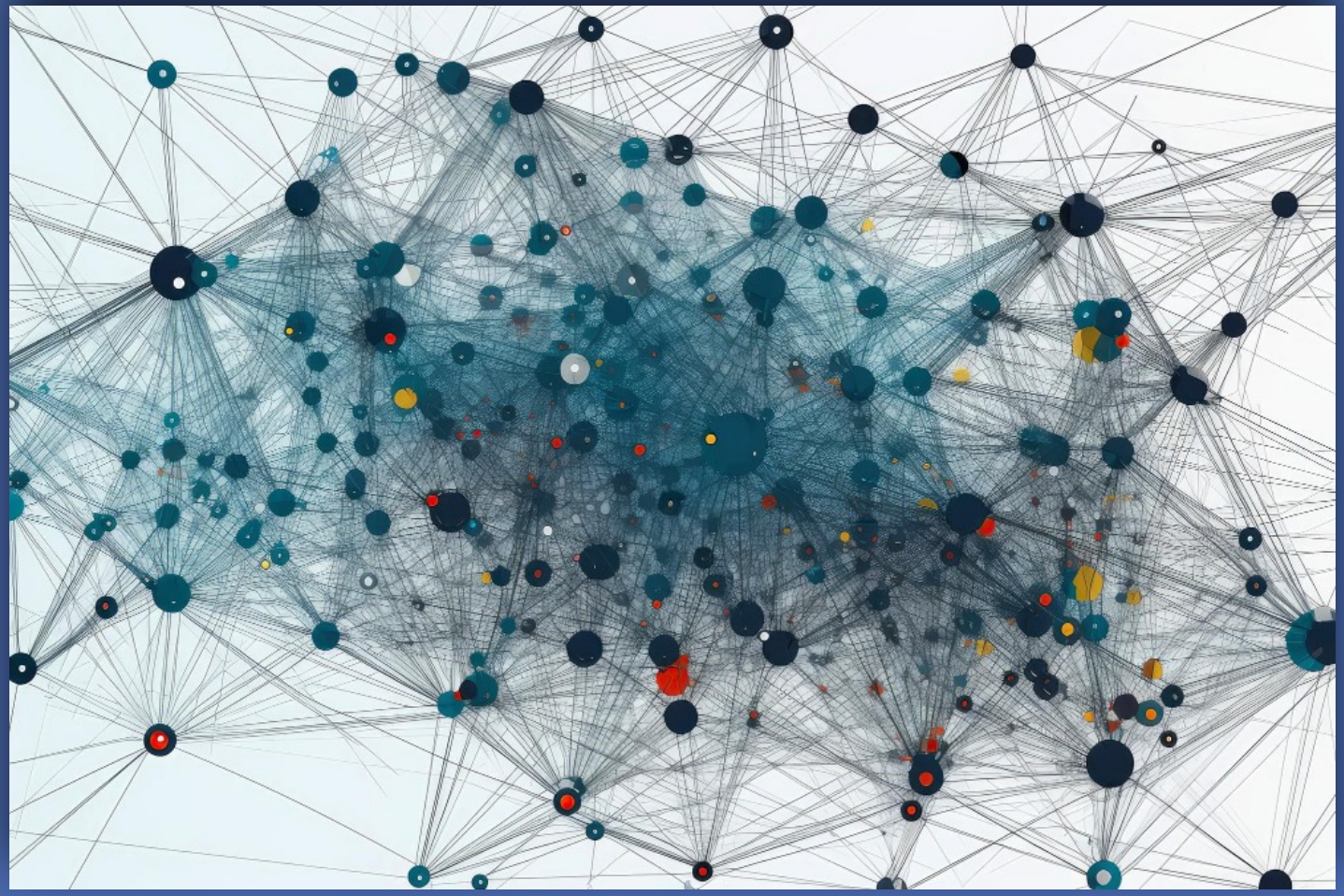


Random Forests



https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

Clustering



CBS

Unsupervised Learning

- Supervised learning:
  - We have features x_i and class labels y_i.
  - Write a program that produces y_i from x_i.
- Unsupervised learning:
  - We only have x_i values, but no explicit target labels.
  - You want to do "something" with them.
- Some unsupervised learning tasks:
  - Outlier detection: Is this a 'normal' x_i?
  - Similarity search: Which examples look like this x_i?
  - Association rules: Which x_j occur together?
  - Latent-factors: What 'parts' are the x_i made from?
  - Data visualization: What does the high-dimensional X look like?
  - Ranking: Which are the most important x_i?
  - Clustering: What types of x_i are there?

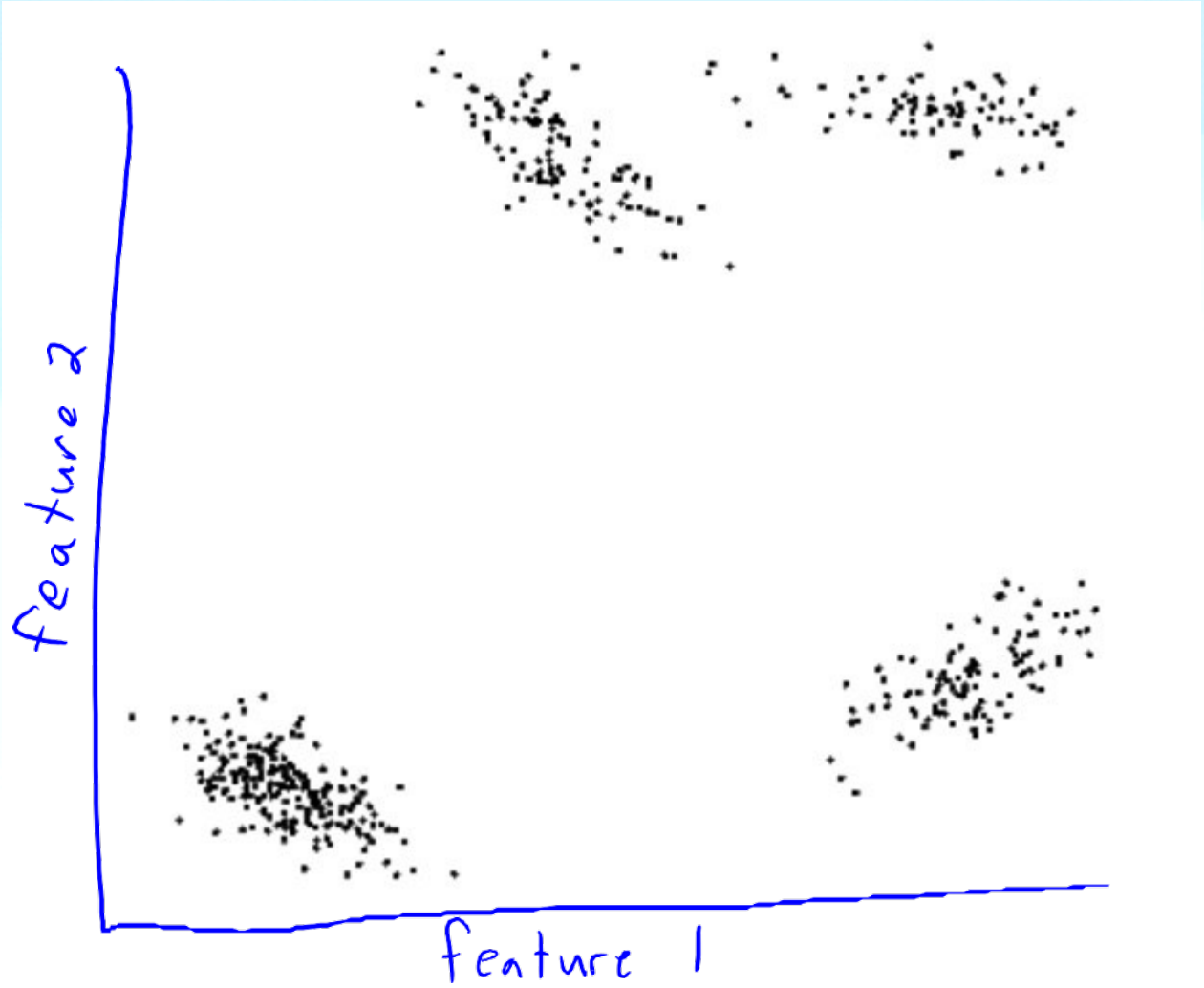
Clustering

- Clustering:
  - Input: set of objects described by features xi.
  - Output: an assignment of objects to ‘groups’.
- Unlike classification, we are not given the ‘groups’.
  - Algorithm must discover groups
- Customer Segmentation e.g. R F M Model
  - Recency - days since last customer transaction
  - Frequency - number of transactions in the last 12 months
  - Monetary Value - total spend in the last 12 months

Clustering Example

Input: data matrix 'X'.

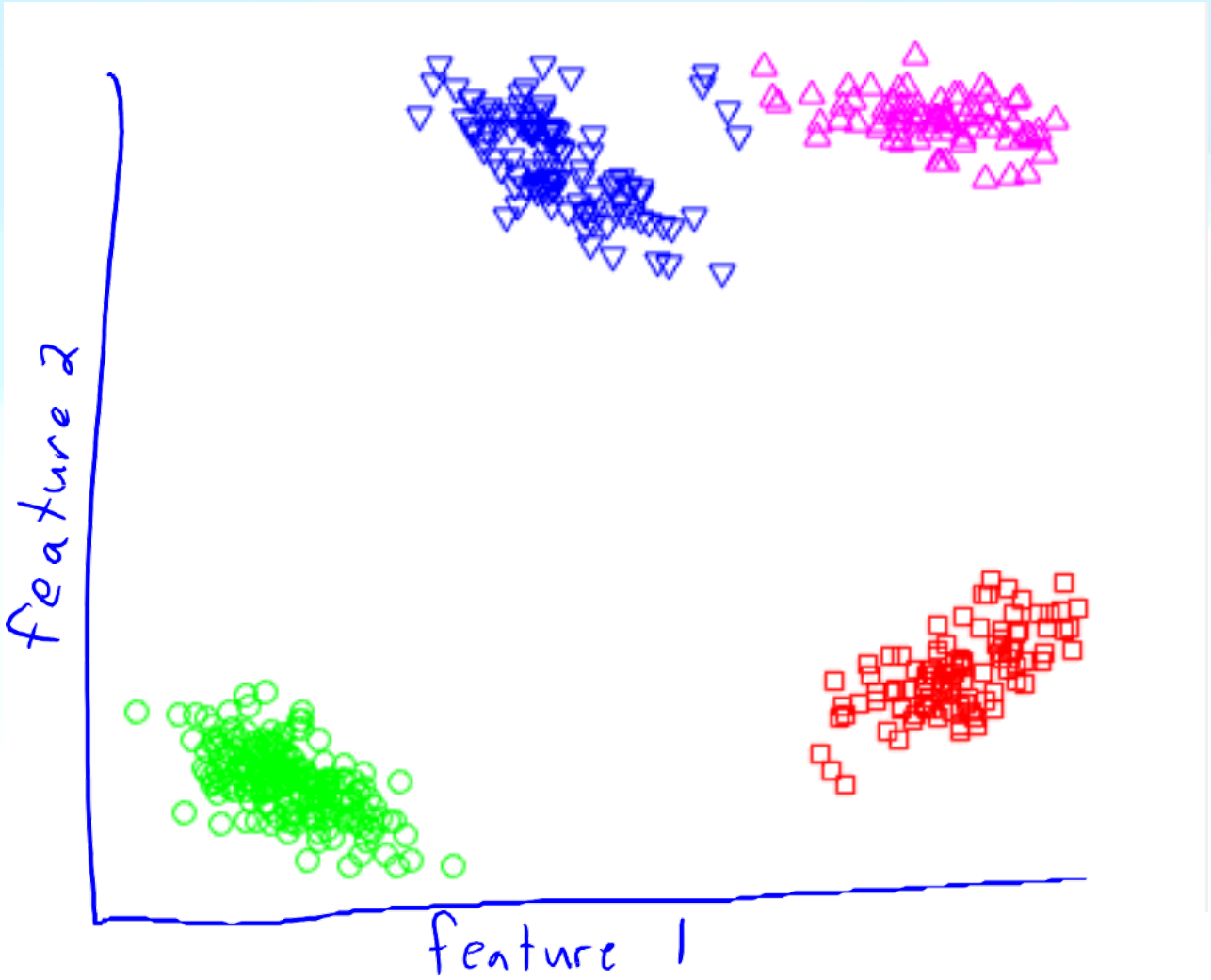
$$X = \begin{bmatrix} -9.0 & -7.3 \\ -10.9 & -9.0 \\ 13.7 & 19.3 \\ 13.8 & 20.4 \\ 12.8 & 20.6 \\ \vdots & \vdots \end{bmatrix}$$



Clustering Example

Input: data matrix 'X'.

$$X = \begin{bmatrix} -9.0 & -7.3 \\ -10.9 & -9.0 \\ 13.7 & 19.3 \\ 13.8 & 20.4 \\ 12.8 & 20.6 \\ \vdots & \vdots \end{bmatrix}$$



Output: clusters '$\hat{y}$'.

$$\hat{y} = \begin{bmatrix} 2 \\ 2 \\ 3 \\ 3 \\ 1 \\ \vdots \end{bmatrix}$$

Other Clustering Applications

- NASA: what types of stars are there?
- Biology: are there sub-species?
- Documents: what kinds of articles are there on Wikipedia?

- Recommendation systems
  - Product recommendation (e.g. Amazon)
  - User Recommendations (e.g. Netflix)

K-Means

- Most popular clustering method is k-means.
- Input:
  - The number of clusters ‘k’ (hyperparameter).
  - Initial guess of the center (the “mean”) of each cluster.
- Algorithm:
  1. Assign each x_i to its closest mean.
  2. Update the means based on the cluster assignments.
  3. Repeat steps 1-2 until convergence.

Jupyter notebok demo (part 1)

https://colab.research.google.com/drive/1KtbuJtysBtV7hqSIqvA5BSI5PNm_5HLA#scrollTo=pS8t2s37m8VF

https://colab.research.google.com/drive/1KraiThBZFC4VxO9BdkeBpvW2dwQ5TLgS?authuser=1&pli=1&usp=drive_fs

K-Means Issues

- Guaranteed to converge when using Euclidean distance.
- Given a new test object: assign it to the nearest mean to cluster it.
- Assumes you know number of clusters 'k'.
  - Lots of heuristics to pick 'k', none satisfying:
    - https://en.wikipedia.org/wiki/Determining_the_number_of_clusters_in_a_data_set
- Each object is assigned to one (and only one) cluster:
  - No possibility for overlapping clusters or leaving objects unassigned.
- It may converge to sub-optimal solution…
  - Classic approach to dealing with sensitivity to initialization: random restarts.
    - Try several different random starting points, choose the "best".
  - A more clever approach called k-means++.
    - This is what scikit-learn's KMeans does by default.

Drawbacks of k-means

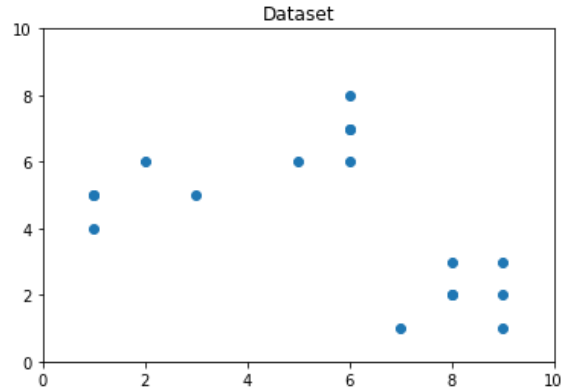
- Final cluster assignments depend on initialization of centers, meaning that:
  - Cluster assignments may be different on different runs
  - K-means may not achieve the global optimum
- Number of clusters need to be specified beforehand → Elbow method, Silhouette analysis
- Poor results on non-linear cluster shapes → Other clustering methods (spectral)
- Slow on large number of samples → Minibatch k-means

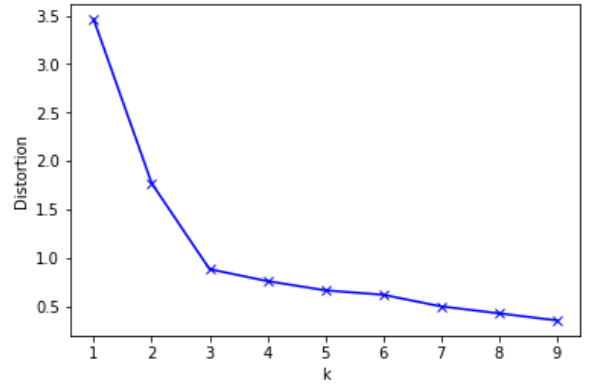
(VanderPlas 2016)

Elbow method

- Run k-means for several *k*
- Distortion: sum of distances of each point to the center of the closest cluster
- Look for *k* where the curve stops decreasing rapidly

https://pythonprogramminglanguage.com/kmeans-elbow-method/



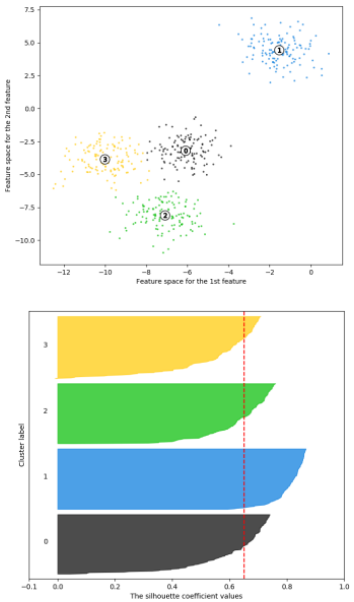


Silhouette analysis

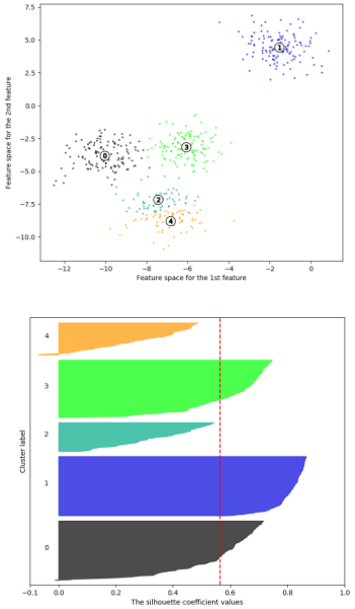
- Silhouette coefficient shows how far each sample is from other cluster centers
  - +1: far from others; 0: at the boundary; -1: sample assigned to the wrong cluster
- Thickness shows cluster size
- Look for:
  - Average silhouette score
  - Scores for clusters < average
  - Individual scores < 0

https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

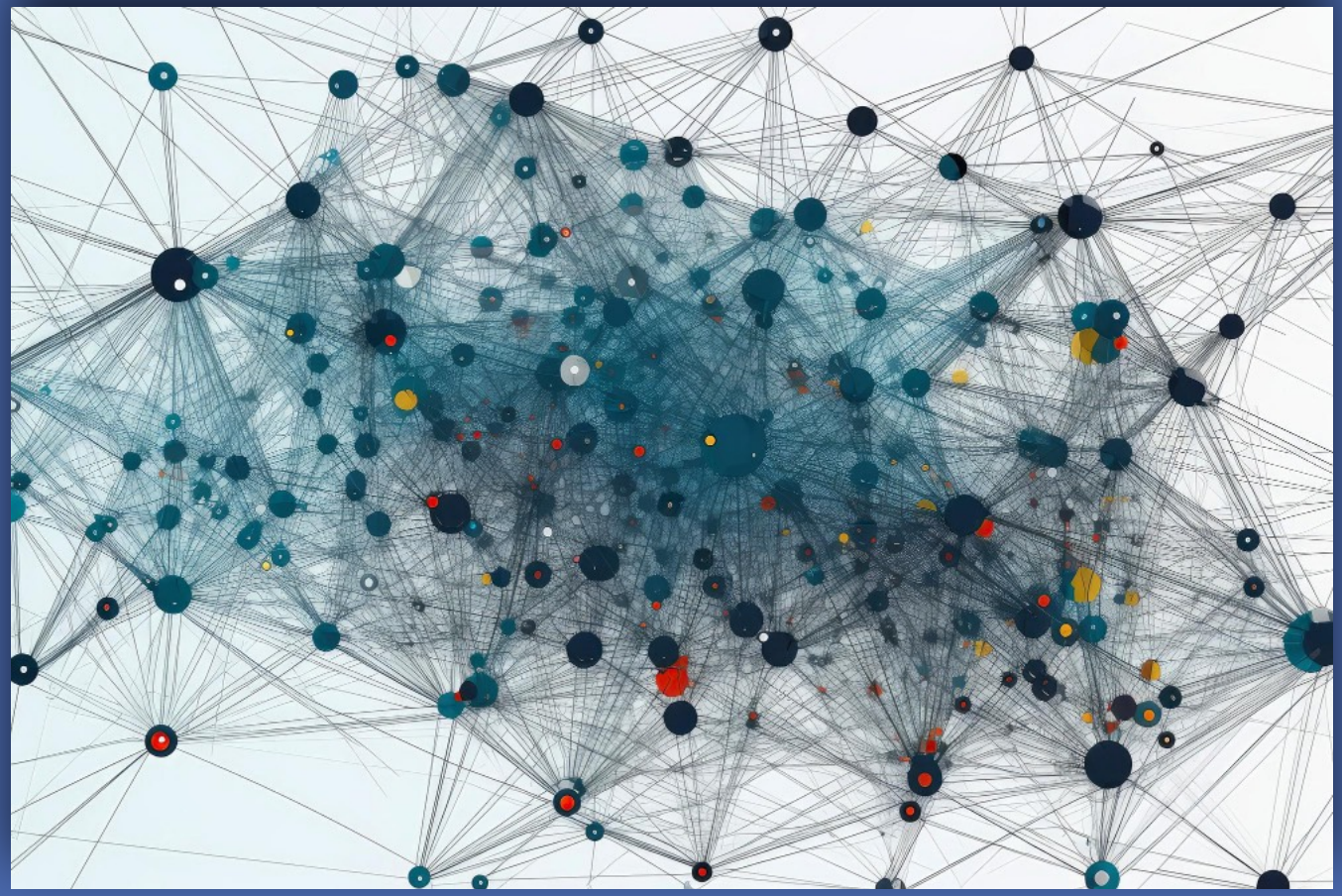
4 clusters



5 clusters



Density Based Clustering



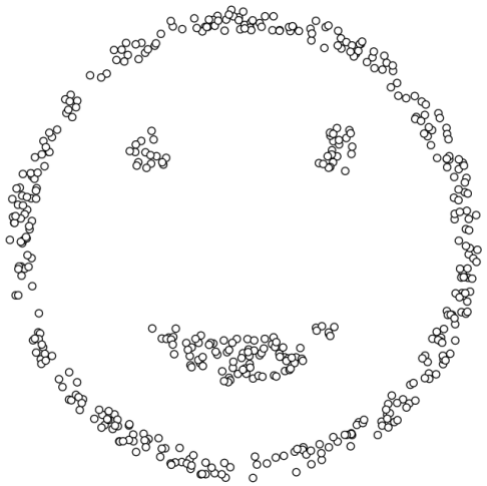
CBS

Motivation for Density-Based Clustering

- Density-based clustering:
  - Clusters are defined by "dense" regions.
  - Objects in non-dense regions don't get clustered.
    - Not trying to "partition" the space.
- Clusters can be non-convex
- It's a non-parametric clustering method:
  - No fixed number of clusters 'k'.
  - Clusters can become more complicated with more data.

Density-based clustering

- Assumes clusters are dense regions of samples, separated by sparse density
  - Finds clusters of any shape
  - Unlike e.g. k-means, which assumes clusters of convex shapes



naftaliharris.com

- Core samples: samples in areas of high density

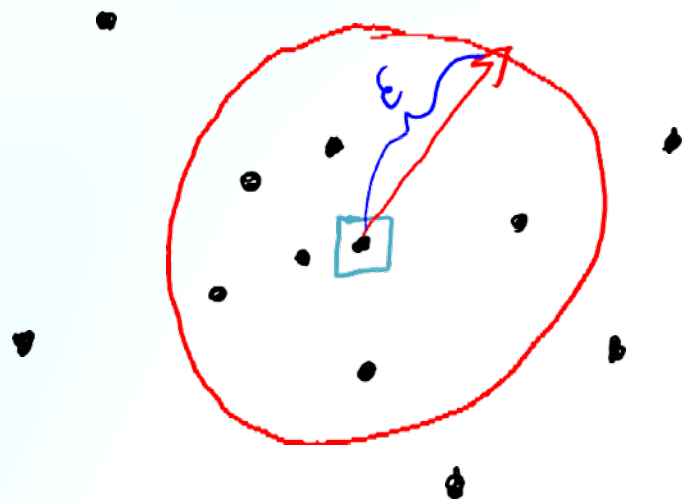
(Han, Kamber & Pei, 2001; https://scikit-learn.org/stable/modules/clustering.html)

(Han, Kamber & Pei, 2001; https://scikit-learn.org/stable/modules/clustering.html

Jupyter notebook demo (part 2)

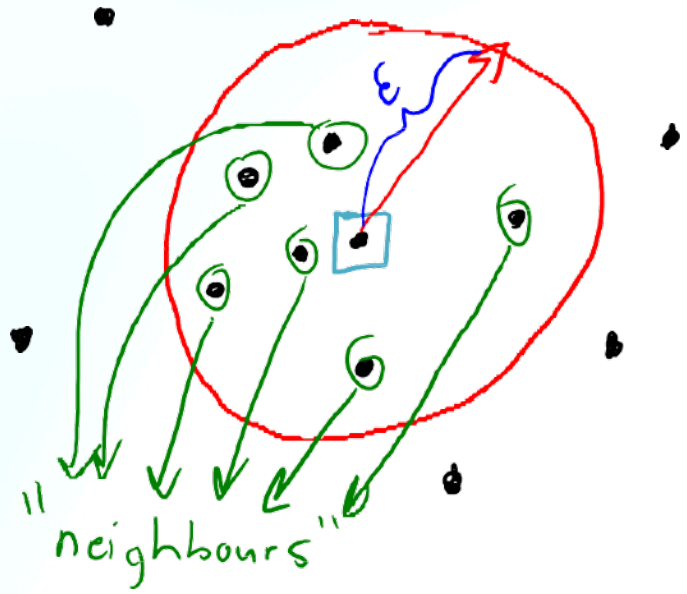
Density-Based Clustering

- Density-based clustering algorithm (DBSCAN) has two hyperparameters:
  - Epsilon (ε): distance we use to decide if another point is a "neighbour".



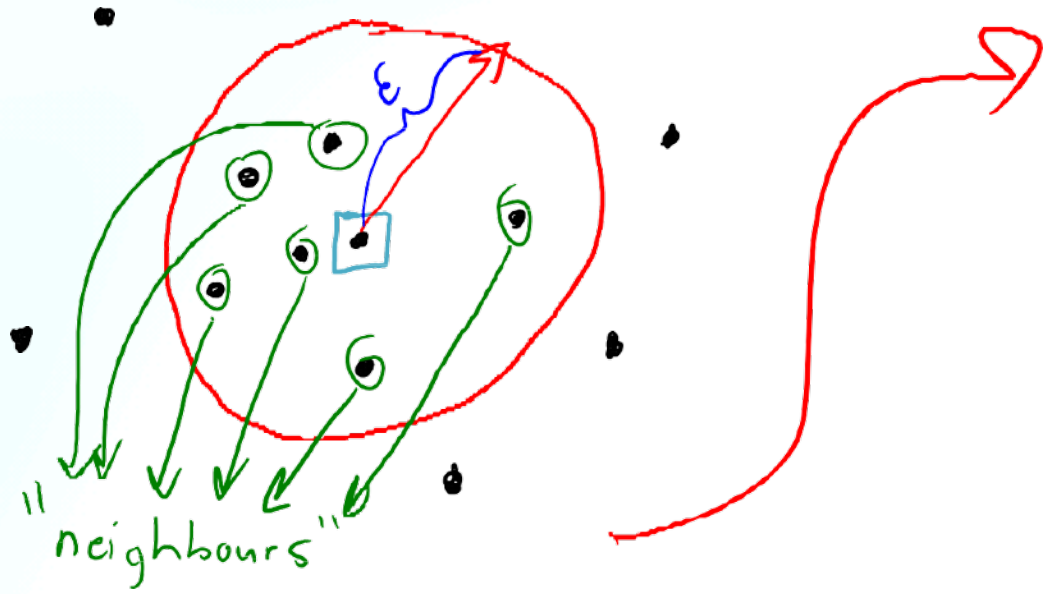
Density-Based Clustering

- Density-based clustering algorithm (DBSCAN) has two hyperparameters:
  - Epsilon (ε): distance we use to decide if another point is a "neighbour".



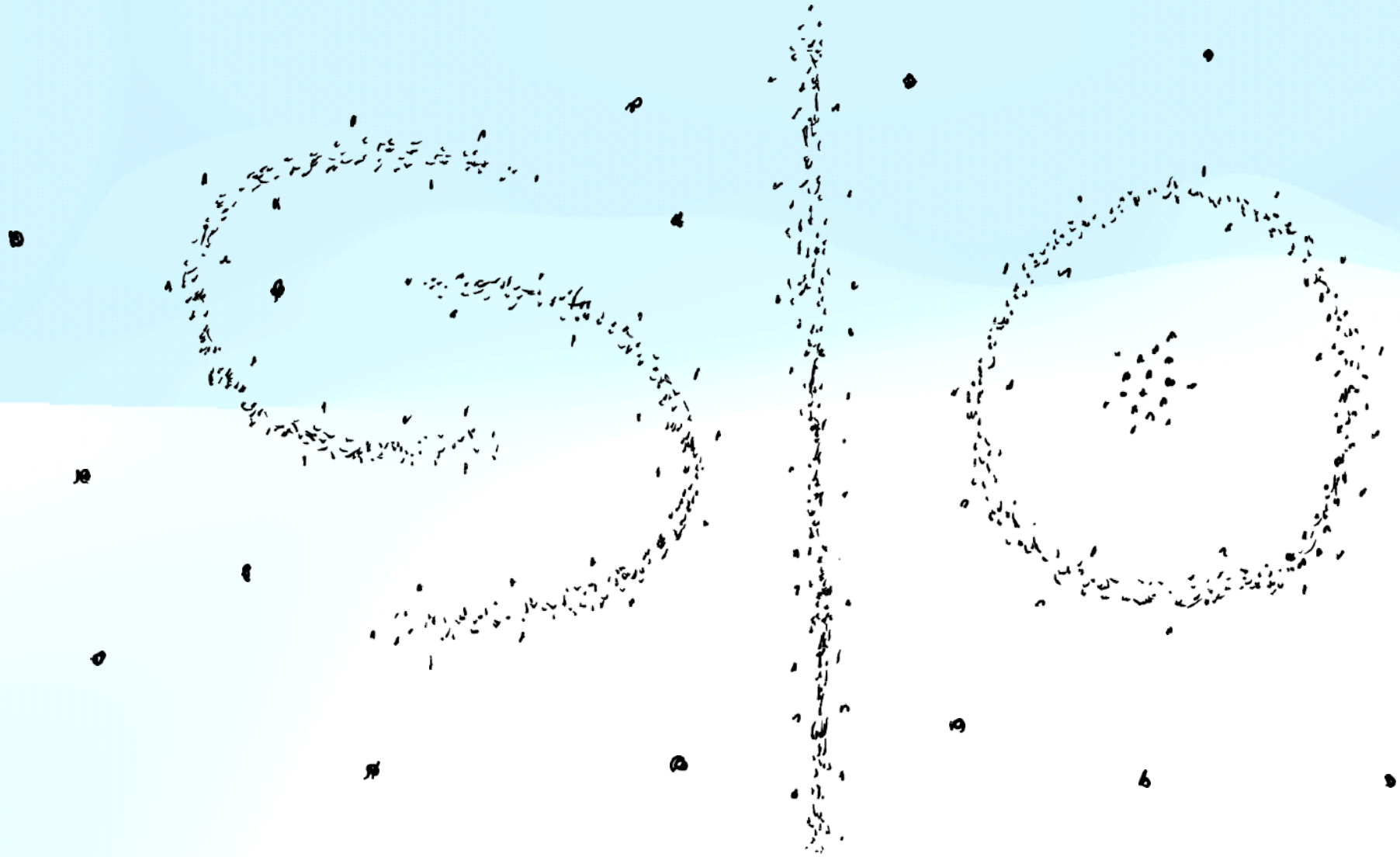
Density-Based Clustering

- Density-based clustering algorithm (DBSCAN) has two hyperparameters:
  - Epsilon (ε): distance we use to decide if another point is a "neighbour".
  - MinNeighbours: number of neighbours needed to say a region is "dense".
    - If you have at least minNeighbours "neighbours", you are called a "core" point.

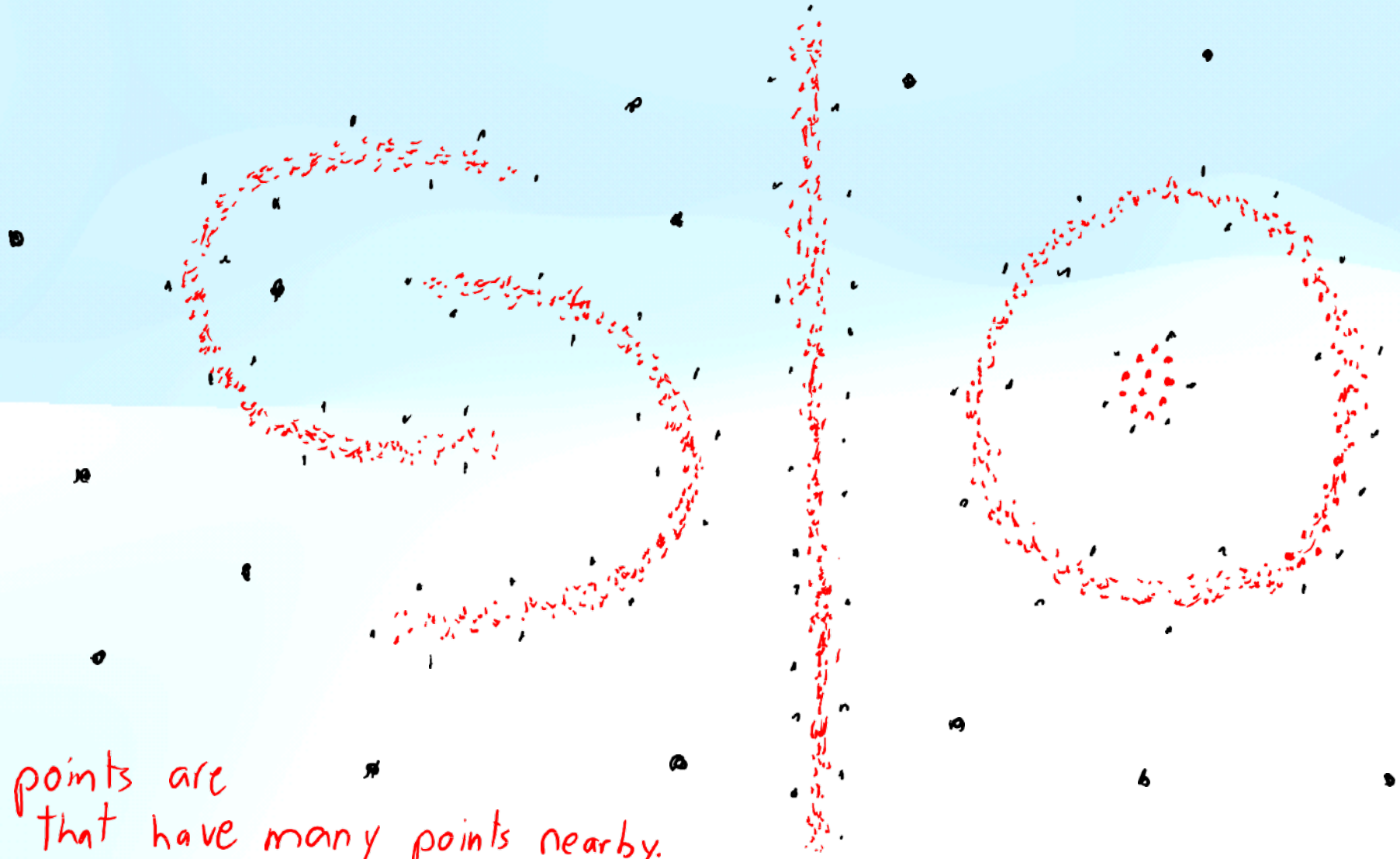


E.g., if minNeighbours = 3 then this is a "core" point since 6 points are "neighbours"

Density-Based Clustering

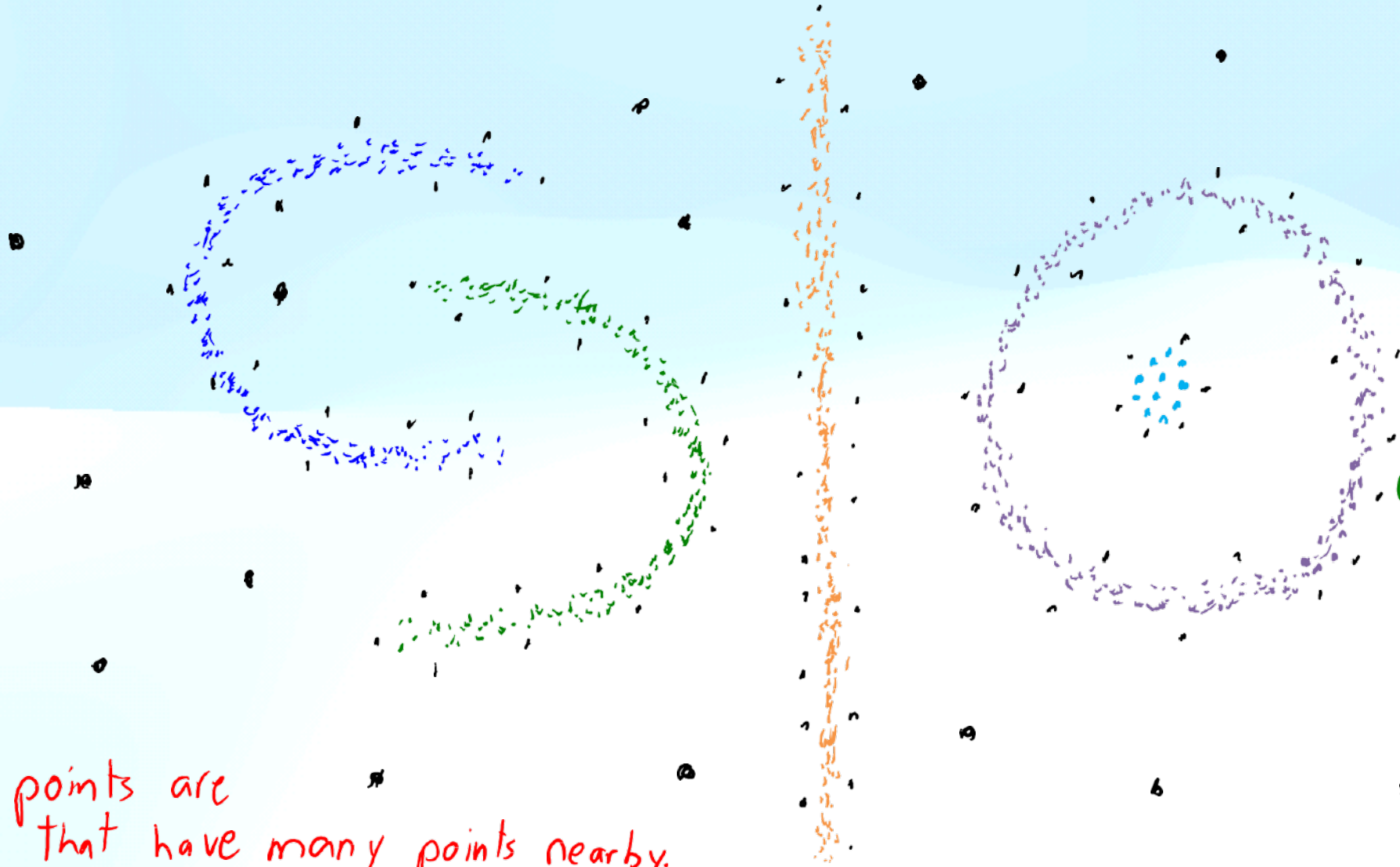


Density-Based Clustering



"Core" points are points that have many points nearby.

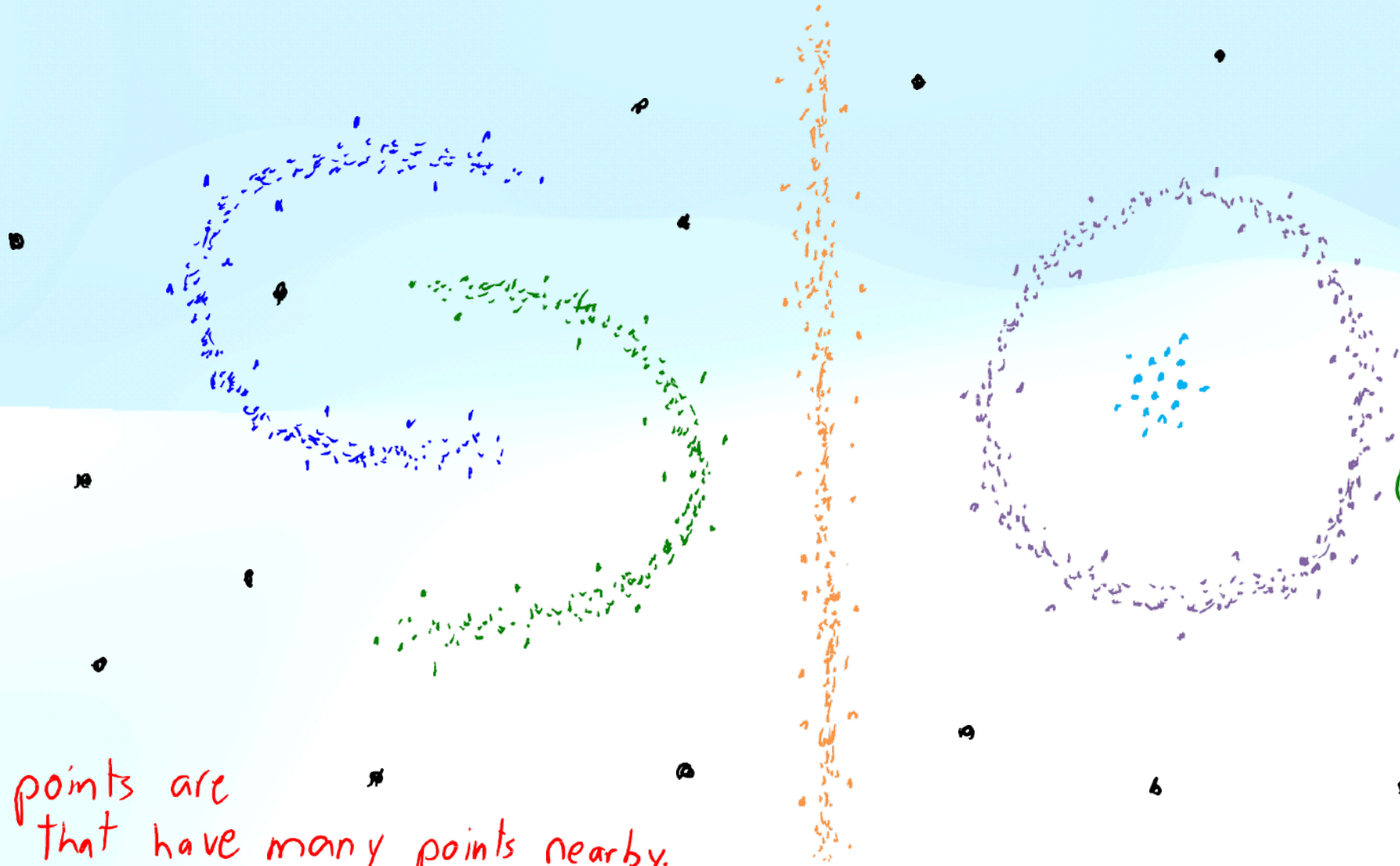
Density-Based Clustering



"Core" points are points that have many points nearby.

Clusters contain:
1. All "core" points that can be reached by following a sequence of core points.

Density-Based Clustering



"Core" points are points that have many points nearby.

Clusters contain:

1. All "core" points that can be reached by following a sequence of core points.
2. All non-core neighbours of core points (boundary points)

Density-Based Clustering Pseudo-Code

- For each example x_i:
  - If x_i is already assigned to a cluster, do nothing.
  - Test whether x_i is a 'core' point (≥ minNeighbours examples within 'ε').
    - If x_i is not core point, do nothing (this could be an outlier).
    - If x_i is a core point, "expand" cluster.

- "Expand" cluster function:
  - Assign all x_j within distance 'ε' of core point x_i to cluster.
  - For each newly-assigned neighbour x_j that is a core point, "expand" cluster.

Density-Based Clustering Issues

- Some points are not assigned to a cluster.
  - Good or bad, depending on the application.
- Ambiguity of "non-core" (boundary) points:

- Sensitive to the choice of ε and minNeighbours.
  - Otherwise, not sensitive to initialization (except for boundary points).

- If you get a new example, finding cluster is expensive.
  - Need to compute distances to training points.
- In high-dimensions, need a lot of points to 'fill' the space.

Artificial Neural Networks and Deep Learning

Some slides have been taken and adapted from the course
CPSC 340: Machine Learning and Data Mining
https://www.cs.ubc.ca/~schmidtm/Courses/340-F22/L30.pdf

CBS

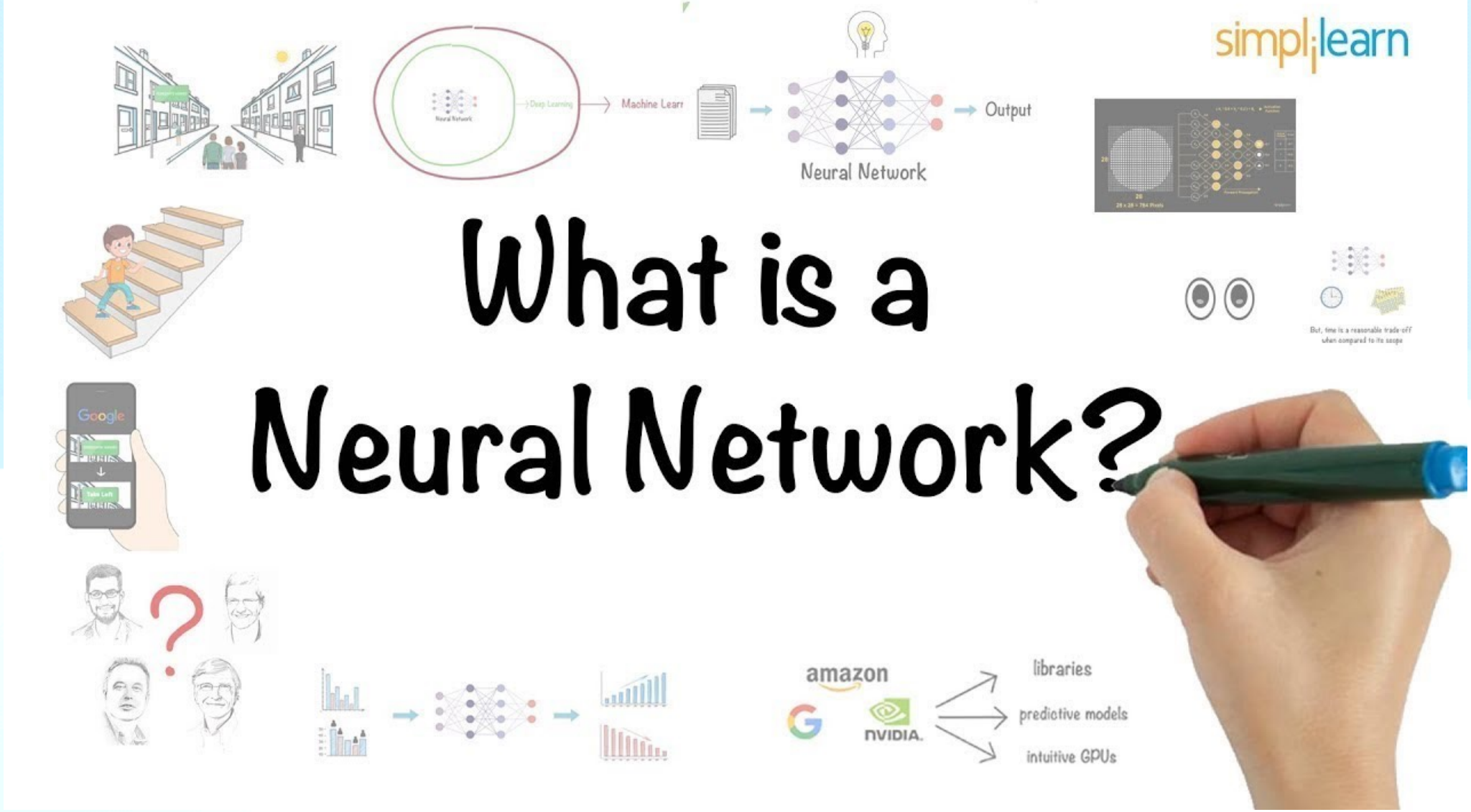
Artificial Neural Networks

Neural Network In 5 Minutes | What Is A Neural Network? | How Neural Networks Work | Simplilearn

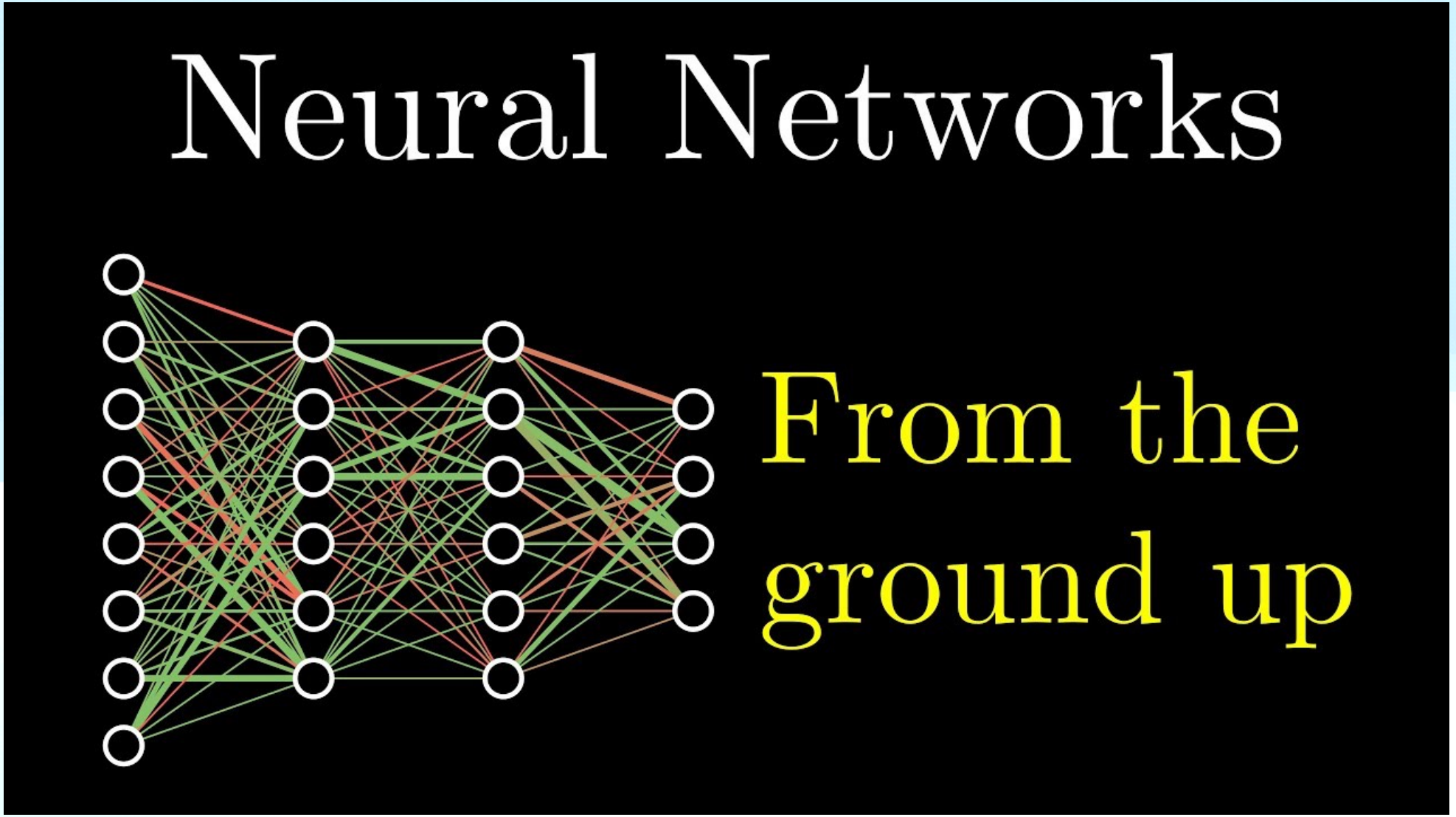
https://www.youtube.com/watch?v=bfmFfD2RIcg

3BLUE1BROWN SERIES S3 • E1: But what is a Neural Network? | Deep learning, chapter 1

https://www.youtube.com/watch?v=aircAruvnKk

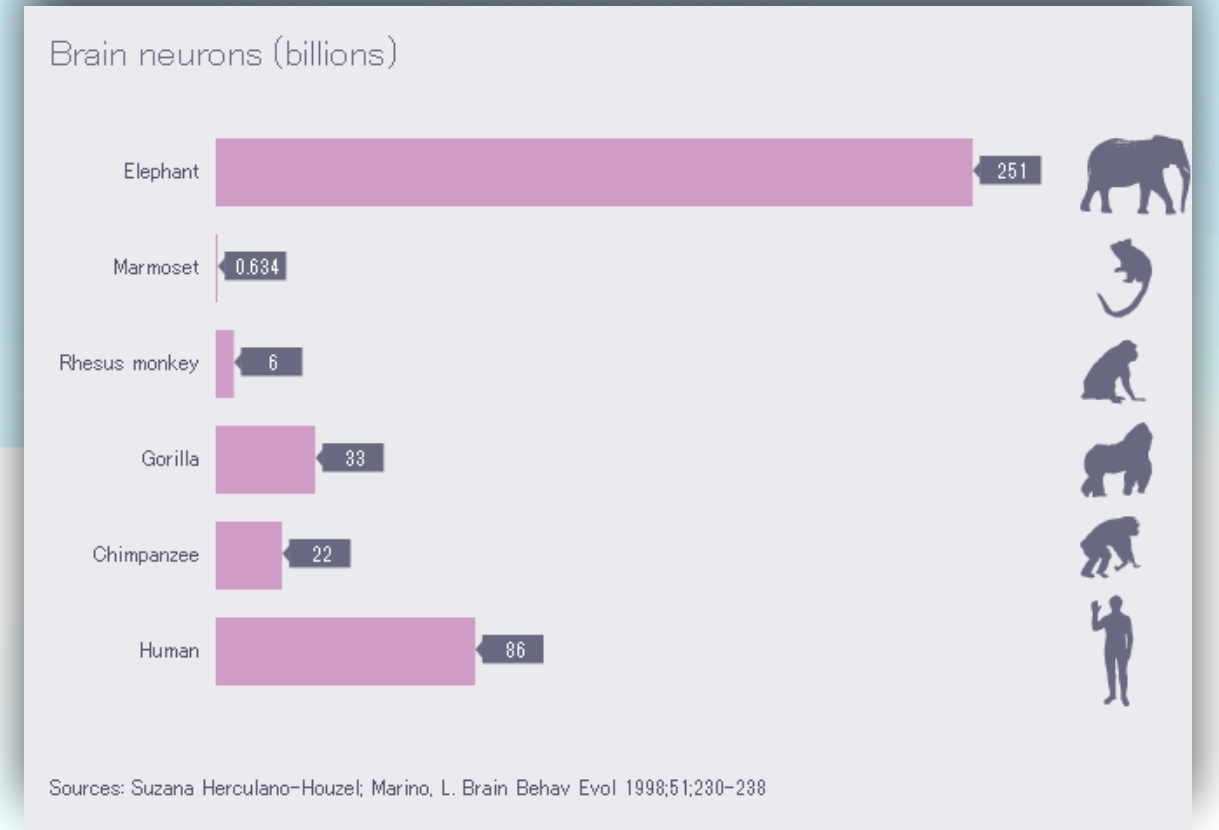


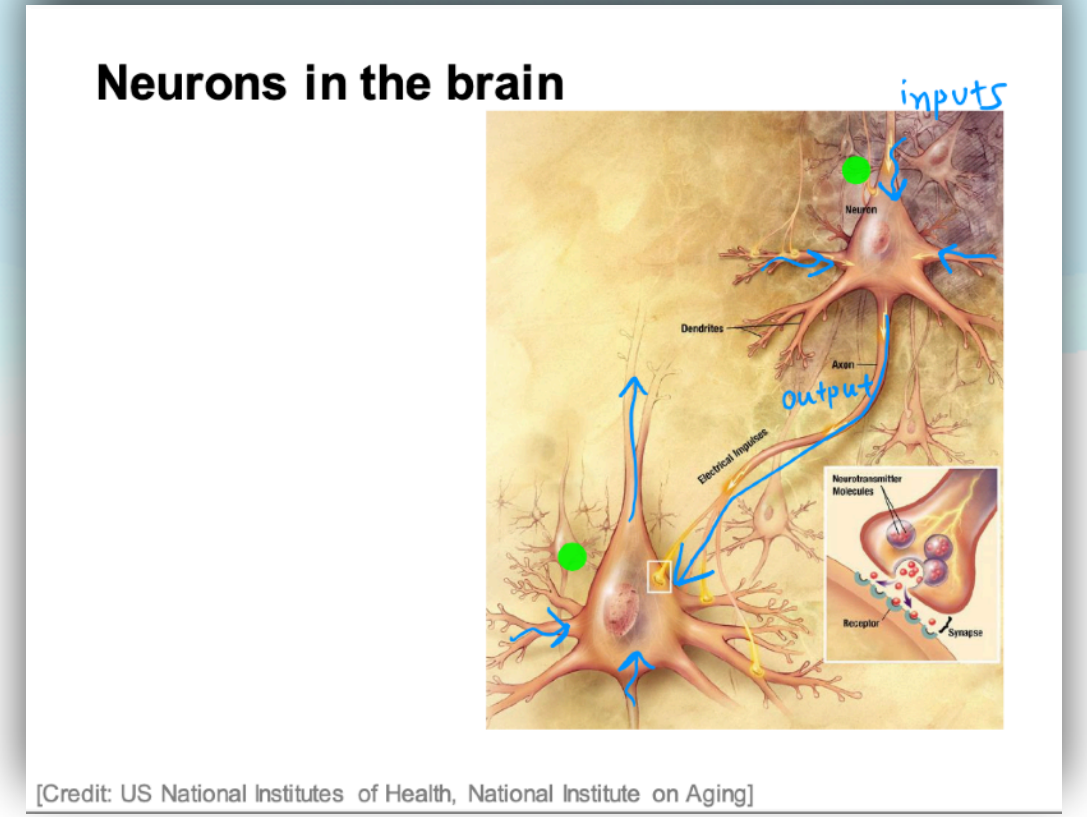
Simple Learn Neural Network In 5 Minutes
https://www.youtube.com/watch?v=bfmFfD2RIcg



But what is a neural network? | Chapter 1, Deep learning
https://www.youtube.com/watch?v=aircAruvnKk

Artificial Neural Networks



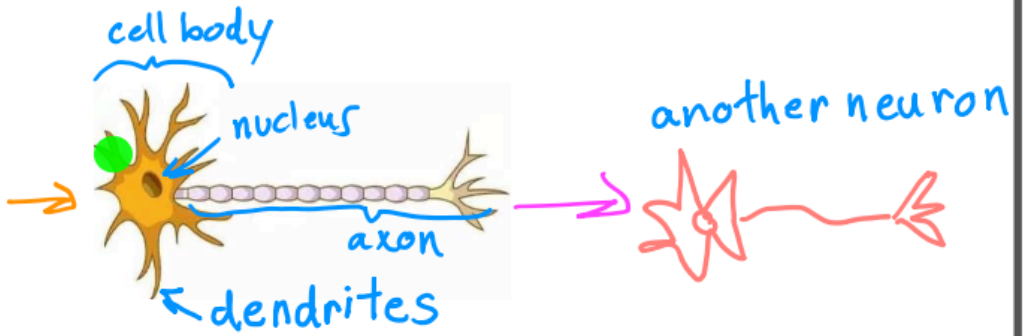


Source: https://www.deeplearning.ai/courses/machine-learning-specialization/

Artificial Neural Networks

Biological neuron

inputs outputs



Simplified mathematical model of a neuron

inputs outputs

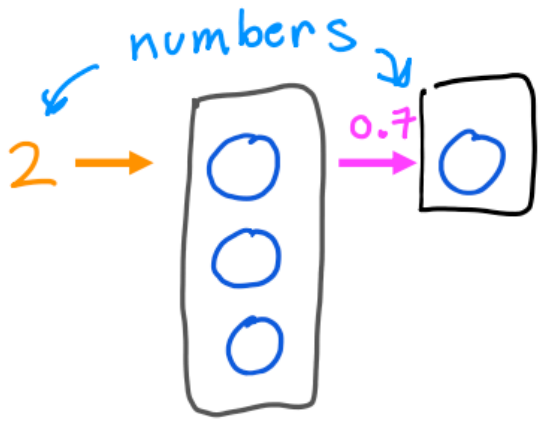
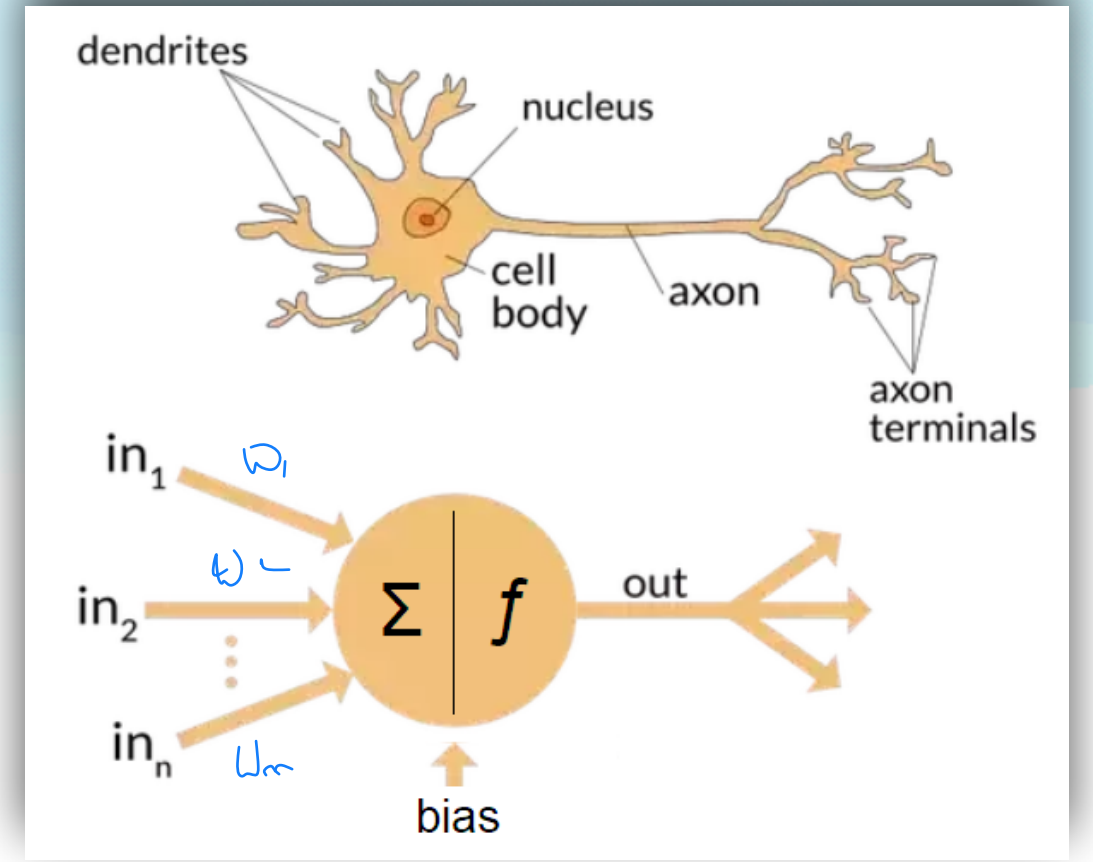


image source: https://biologydictionary.net/sensory-neuron/

Source: https://www.deeplearning.ai/courses/machine-learning-specialization/

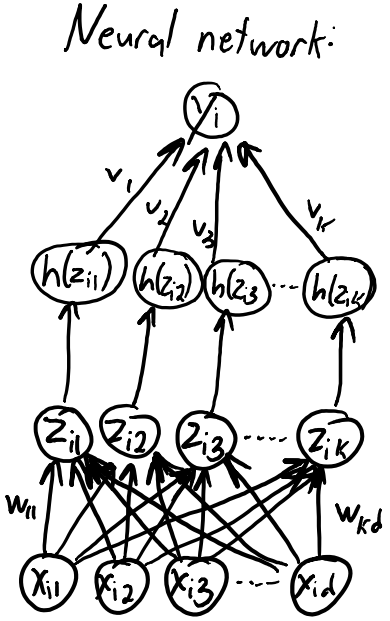
Even if machine learning can solve problems and beat humans in certain fields, it **does not mean** that these algorithms behave like humans and are just as equally able in other fields.

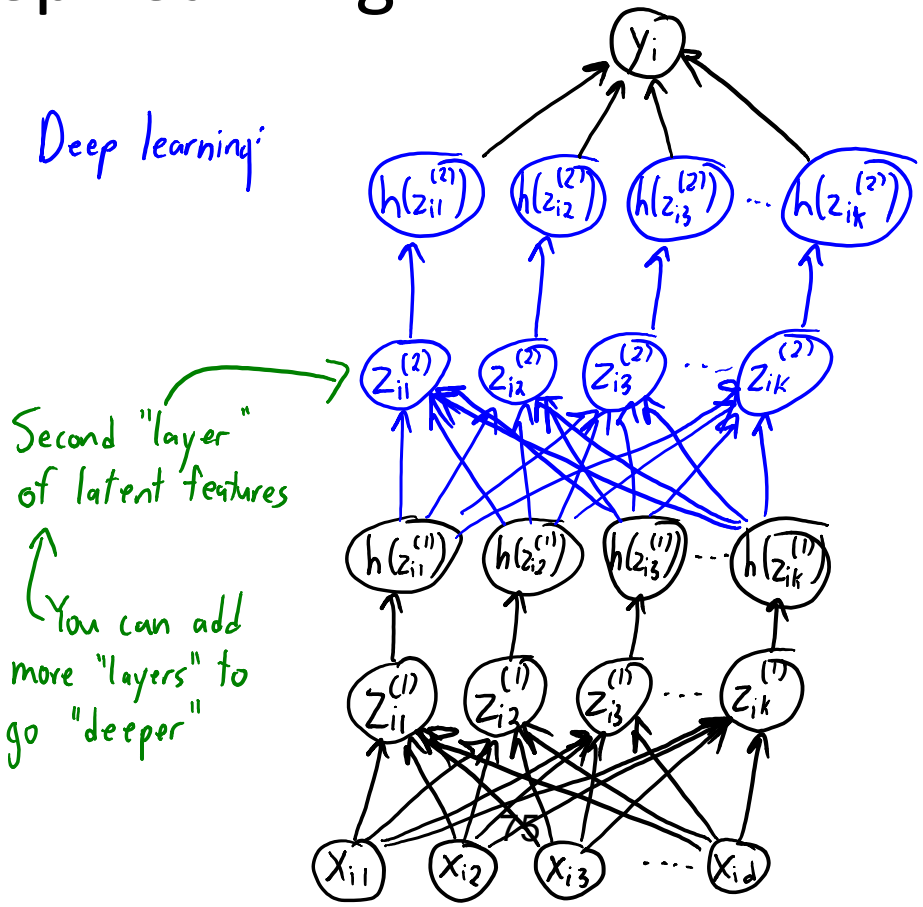
It's important to understand that AI isn't a replacement for human intelligence. It's an extension of it. It can help us to make better decisions and to be more productive, but it can't replace the value of human intuition and creativity.



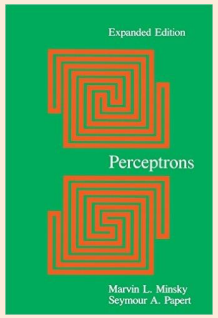
https://www.calls9.com/blogs/can-artificial-intelligence-replace-human-intelligence

Deep Learning

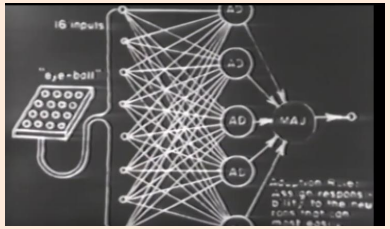




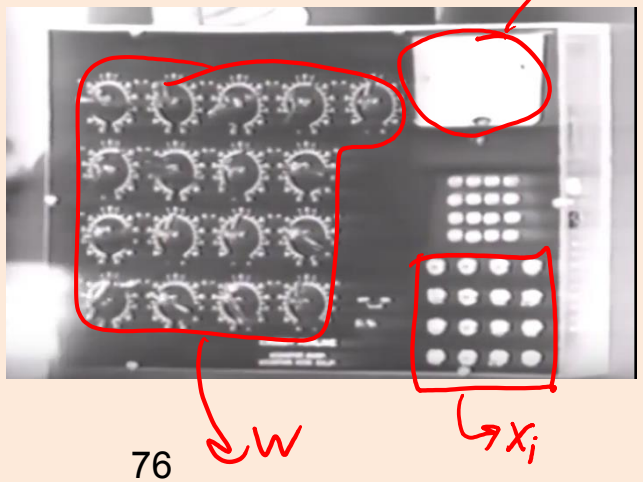
ML and Deep Learning History



- 1950 and 1960s: Initial excitement.
  - Perceptron: linear classifier and stochastic gradient (roughly).
  - "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence." New York Times (1958).
    - https://www.youtube.com/watch?v=IEFRtz68m-8
  - Marvin Minsky assigns object recognition to his students as a summer project



- Then drop in popularity:
  - Quickly realized limitations of linear models.

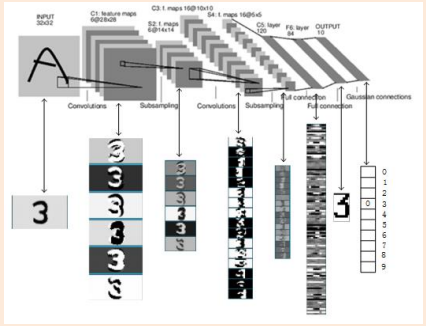


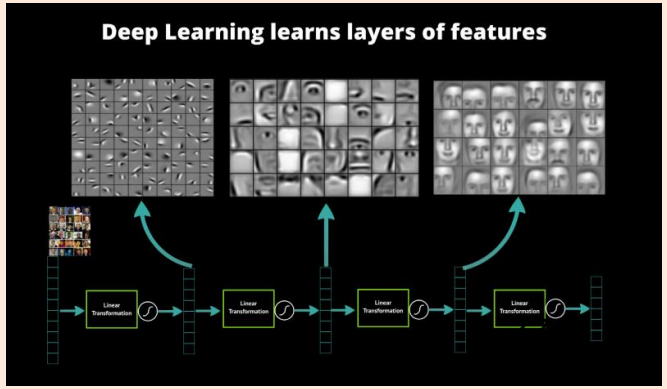
https://mitpress.mit.edu/books/perceptrons/

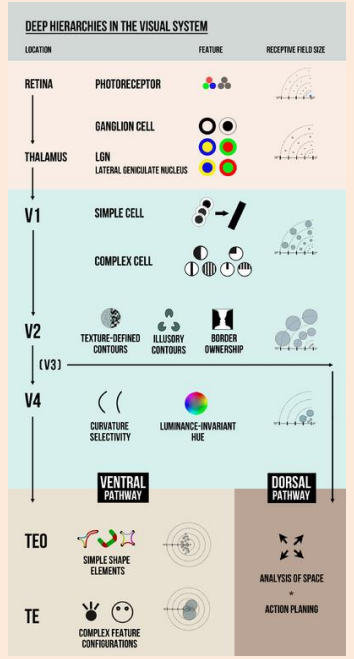
https://www.youtube.com/watch?v=IEFRtz68m-8

ML and Deep Learning History

- 1970 and 1980s: Connectionism (brain-inspired ML)
  - Want "connected networks of simple units".
    - Use parallel computation and distributed representations.
  - Adding hidden layers z_i increases expressive power.
    - With 1 layer and enough sigmoid units, a universal approximator.
  - Success in optical character recognition.





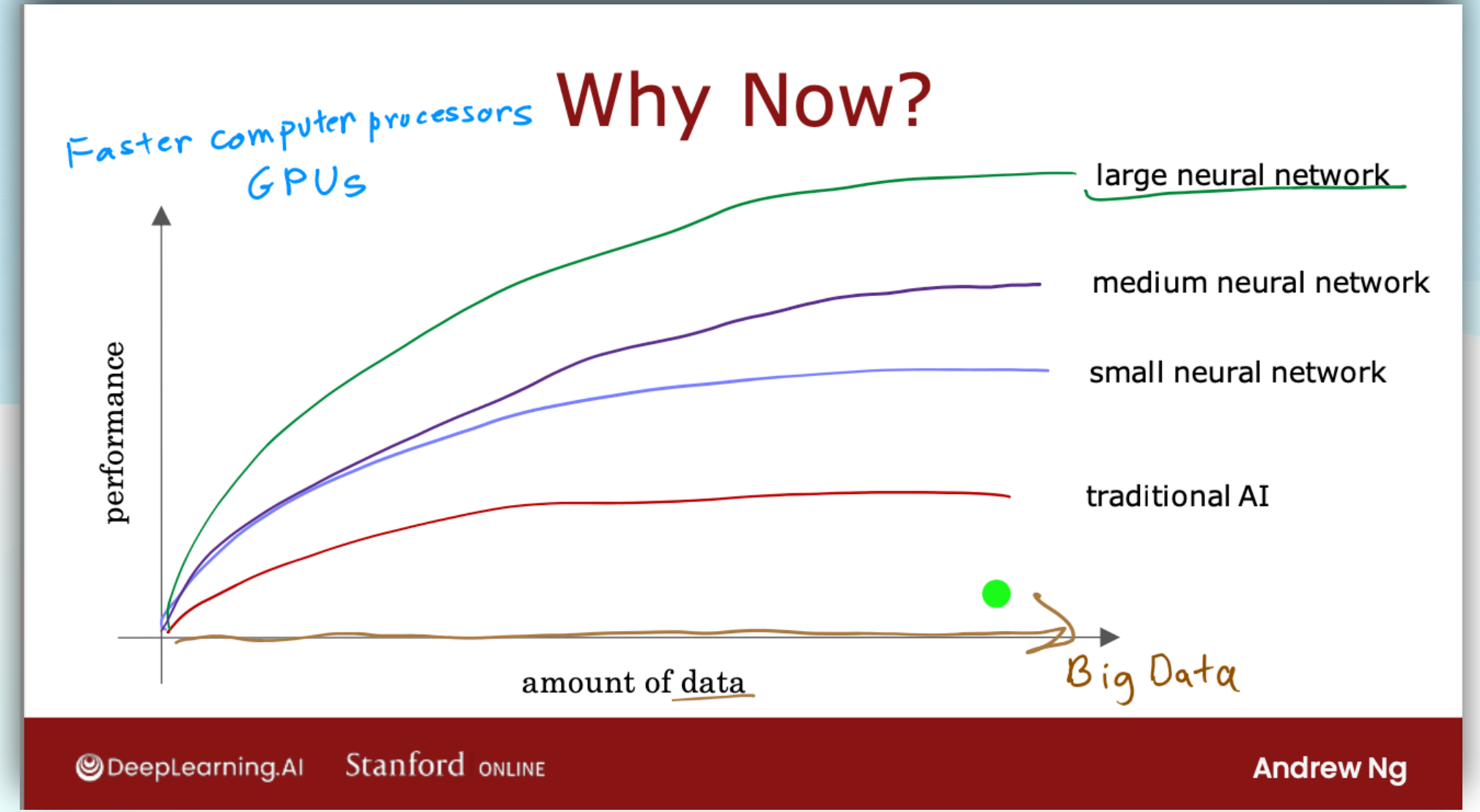


https://en.wikibooks.org/wiki/Sensory_Systems/Visual_Signal_Processing
http://www.datarobot.com/blog/a-primer-on-deep-learning/
http://blog.csdn.net/strint/article/details/44163869

ML and Deep Learning History

- 1990s and early-2000s: drop in popularity.
  - It proved really difficult to get multi-layer models working robustly.
  - We obtained similar performance with simpler models:
    - Rise in popularity of logistic regression and SVMs with regularization and kernels.
  - ML moved closer to other fields (CPSC 540):
    - Numerical optimization.
    - Probabilistic graphical models.
    - Bayesian methods.

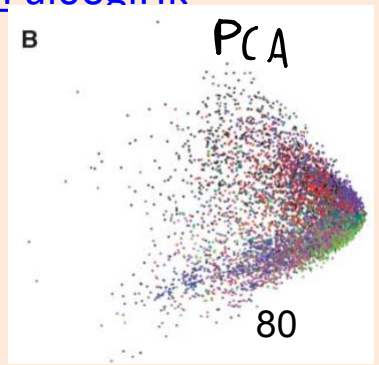
Artificial Neural Networks

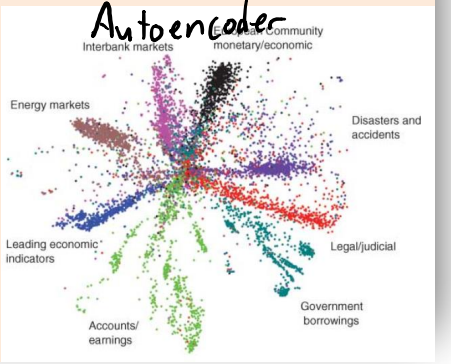


Source: https://www.deeplearning.ai/courses/machine-learning-specialization/

ML and Deep Learning History

- Late 2000s: push to revive connectionism as "deep learning".
  - Canadian Institute For Advanced Research (CIFAR) NCAP program:
    - "Neural Computation and Adaptive Perception".
    - Led by Geoff Hinton, Yann LeCun, and Yoshua Bengio ("Canadian mafia").
  - Unsupervised successes: "deep belief networks" and "autoencoders".
    - Could be used to initialize deep neural networks.
    - https://www.youtube.com/watch?v=KuPai0ogiHk

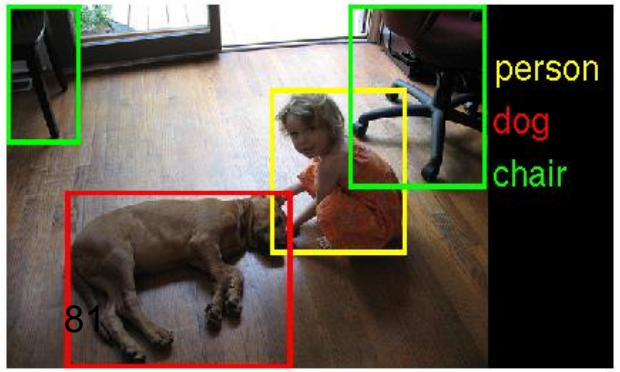




https://www.cs.toronto.edu/~hinton/science.pdf

2010s: DEEP LEARNING!!!

- Bigger datasets, bigger models, parallel computing (GPUs/clusters).
  - And some tweaks to the models from the 1980s.
- Huge improvements in automatic speech recognition (2009).
  - All phones now have deep learning.
- Huge improvements in computer vision (2012).
  - Changed computer vision field almost instantly.
  - This is now finding its way into products.



http://www.image-net.org/challenges/LSVRC/2014/

2010s: DEEP LEARNING!!!

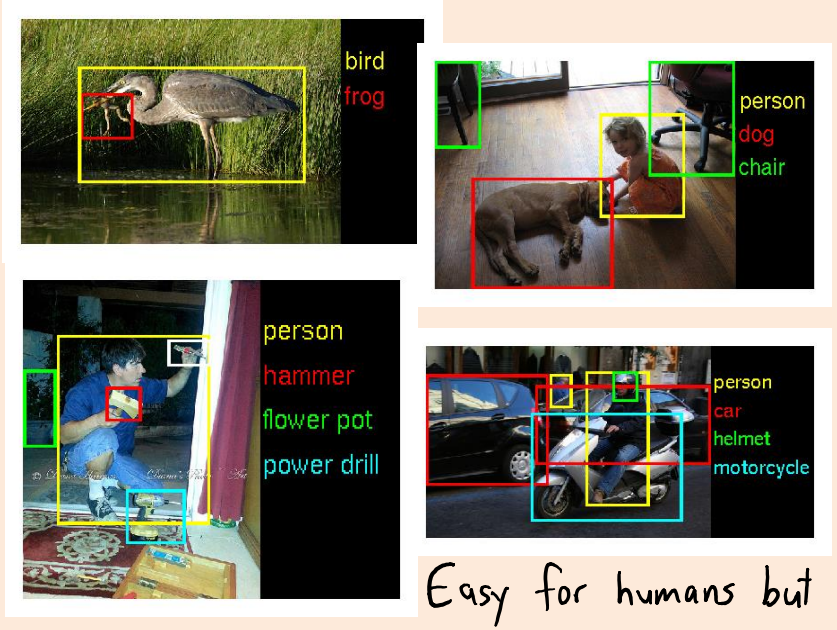
- Media hype:
  - "How many computers to identify a cat? 16,000"
    New York Times (2012).
  - "Why Facebook is teaching its machines to think like humans"
    Wired (2013).
  - "What is 'deep learning' and why should businesses care?"
    Forbes (2013).
  - "Computer eyesight gets a lot more accurate"
    New York Times (2014).

- 2015: huge improvement in language understanding.

Start of Large Language Models

ImageNet Challenge

- Millions of labeled images, 1000 object classes.

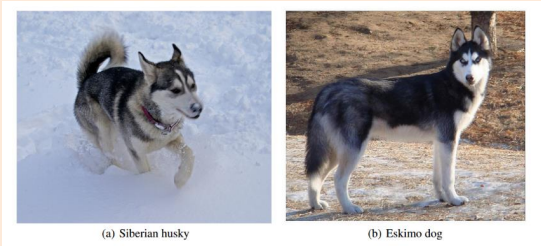


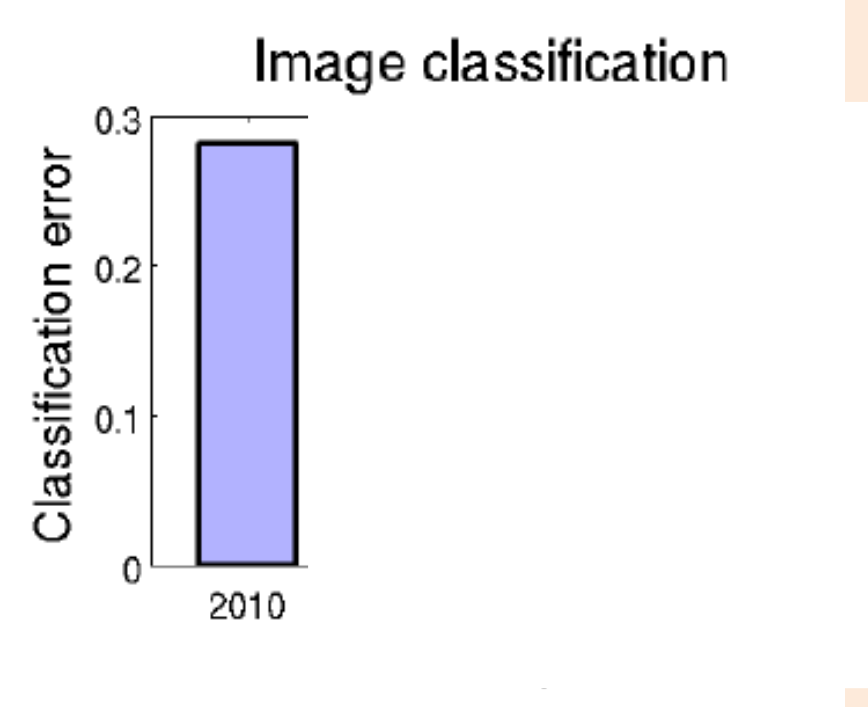
Easy for humans but hard for computers.

http://www.image-net.org/challenges/LSVRC/2014/

ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.

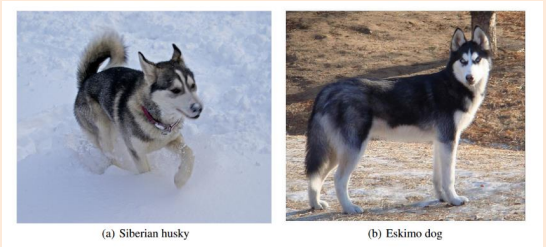




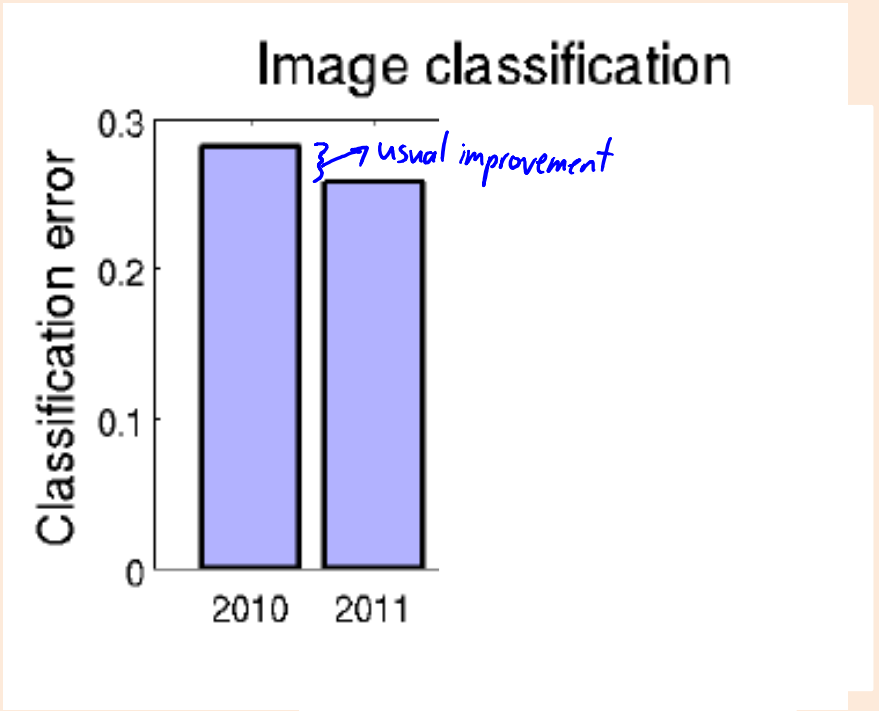
https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.



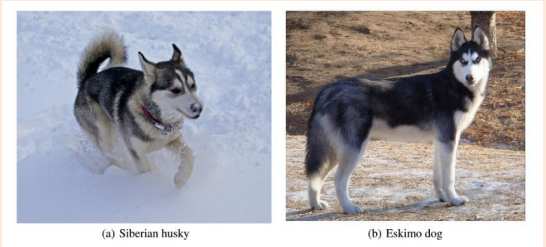
(a) Siberian husky (b) Eskimo dog

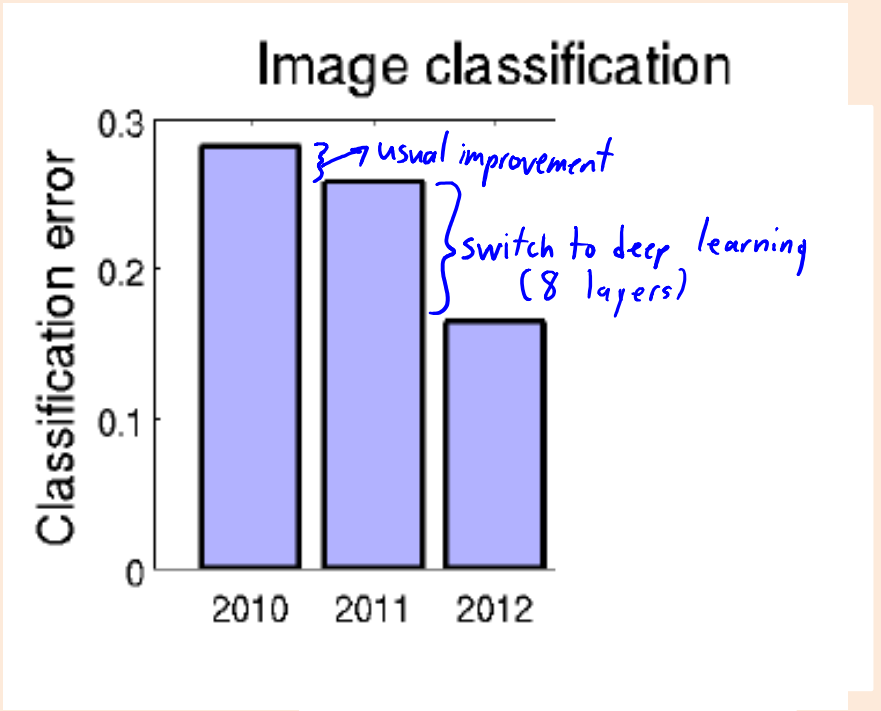


https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.

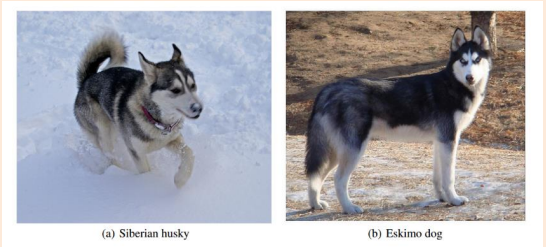


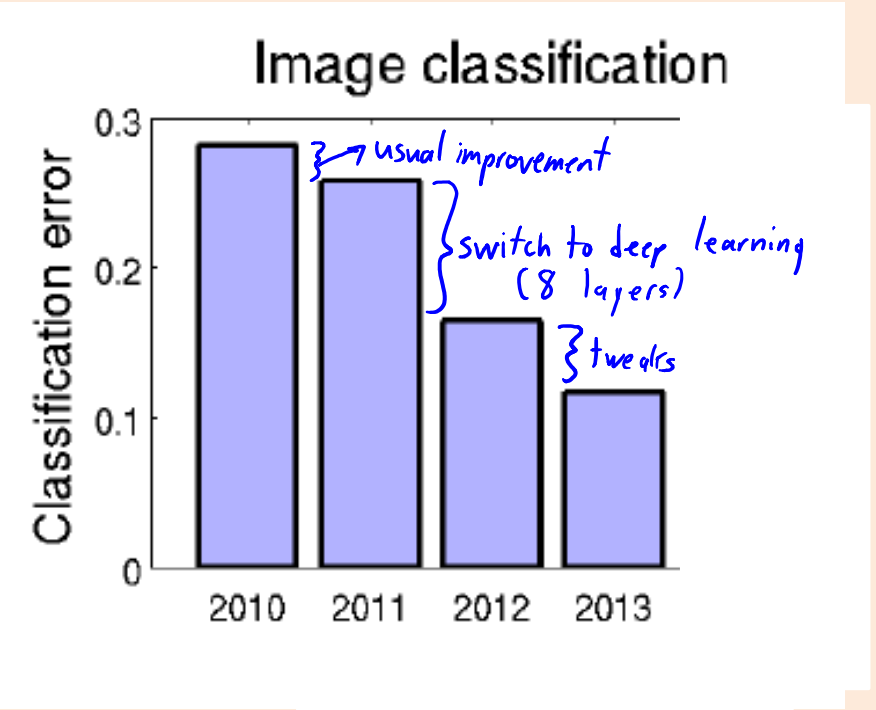


https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.

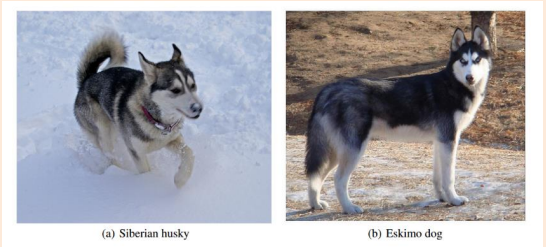


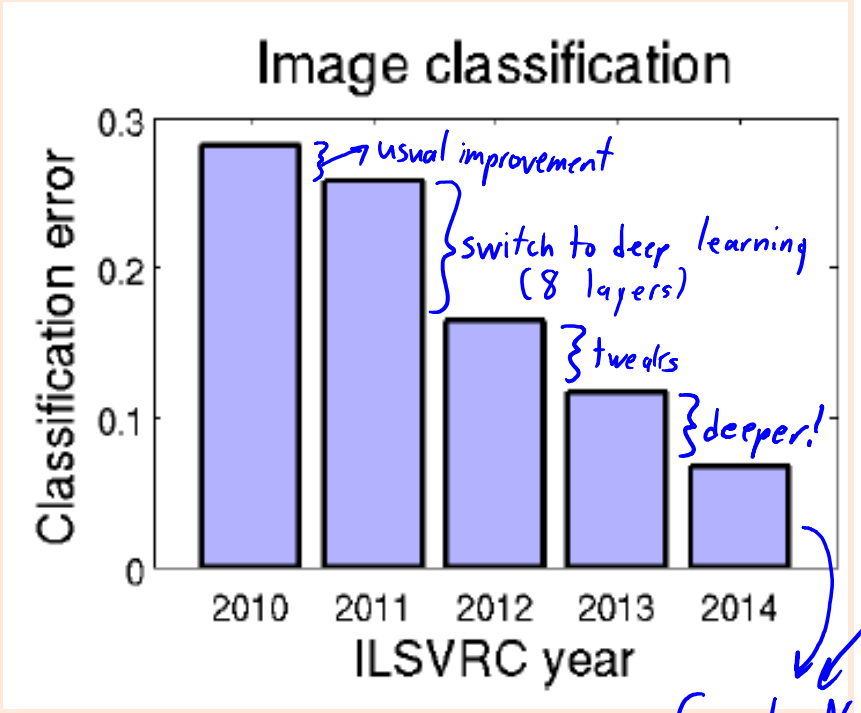


https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.





https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

- Object detection task:
  - Single label per image.
  - Humans: ~5% error.
- 2015: Won by Microsoft Research Asia
  - 3.6% error.
  - 152 layers.
- 2016: Chinese University of Hong Kong:
  - Ensembles of existing methods.
- 2017: fewer entries, organizers decided this would be last year.

and then came Generative AI and Large Language Models

References

- https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Thank you!

Raghava Mukkamala

rrm.digi@cbs.dk

https://www.cbs.dk/staff/rrmdigi

https://raghavamukkamala.github.io/

https://cbsbda.github.io/