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/

Slides have been taken and adapted from Introduction to Computer Science and Programming in Python from MIT OpenCourseWare with due credit to the generous authors of slides from: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/lecture-slides-code/

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

Outline

- Fundamentals of Machine Learning
- Precision Measures
- Decision Trees, Random Forests
- Linear Regression
- Logistic Regression
- Artificial Neural Networks and Deep Learning

Fundamentals of Learning

CBS

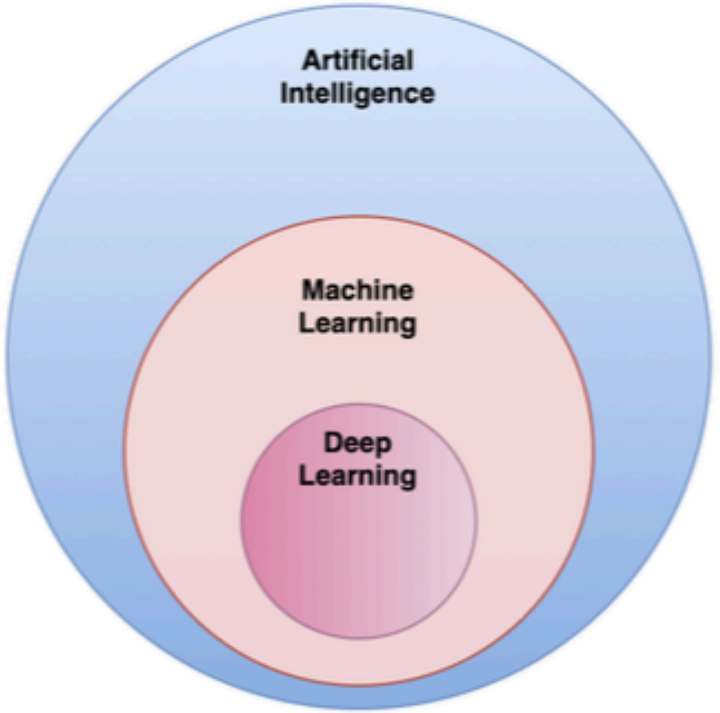
What is Machine Learning

A part of artificial intelligence (AI)

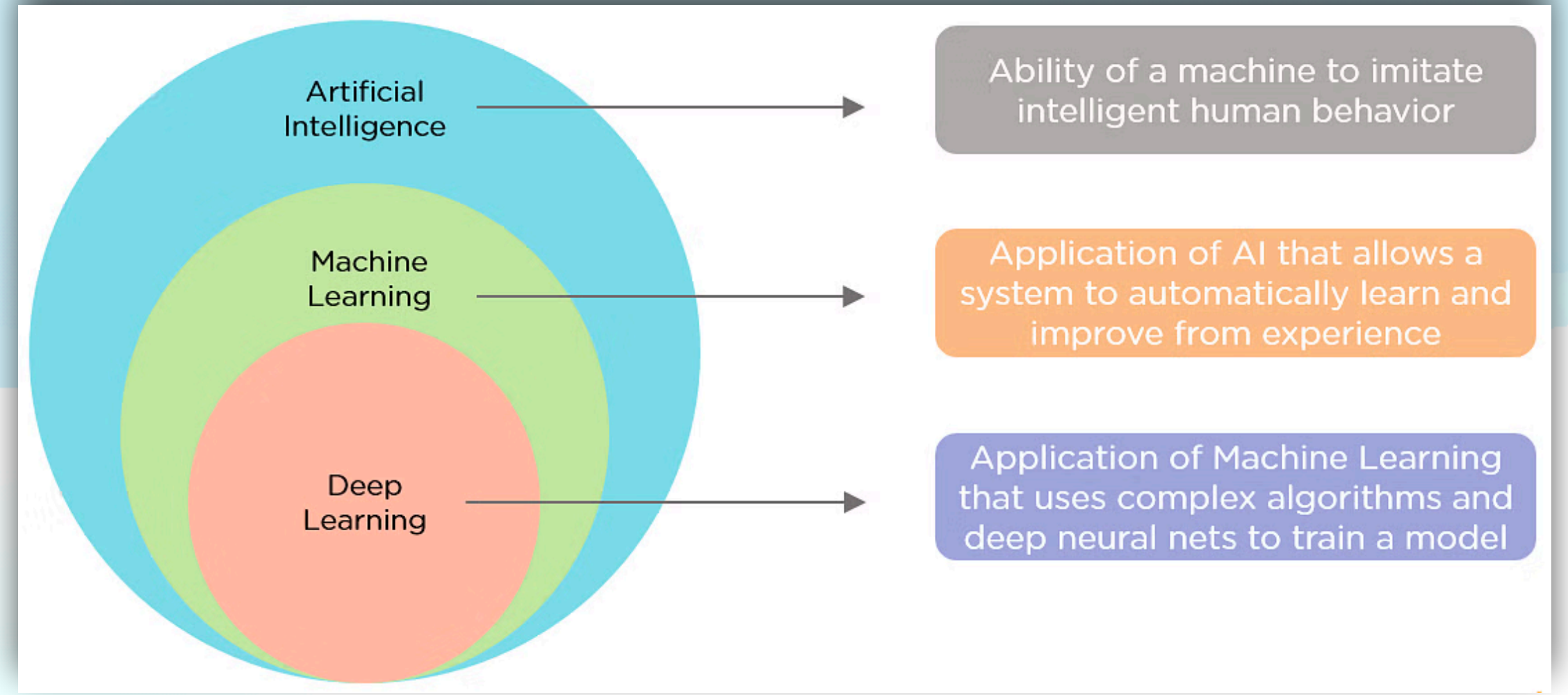
Machine learning (ML) is the study of algorithms that can

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

Deep learning (DL) is a subfield of ML

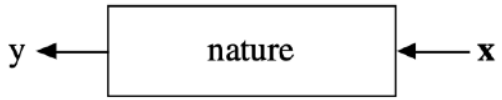


Source: Sippo Rossi, Introduction to Machine Learning



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

Statistical Modeling: The Two Cultures

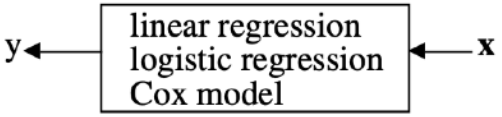


There are two goals in analyzing the data:

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:

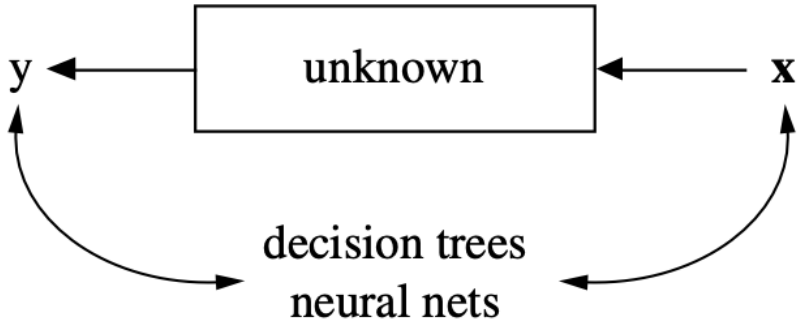


Model validation. Yes–no using goodness-of-fit tests and residual examination.
Estimated culture population. 98% of all statisticians.

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)

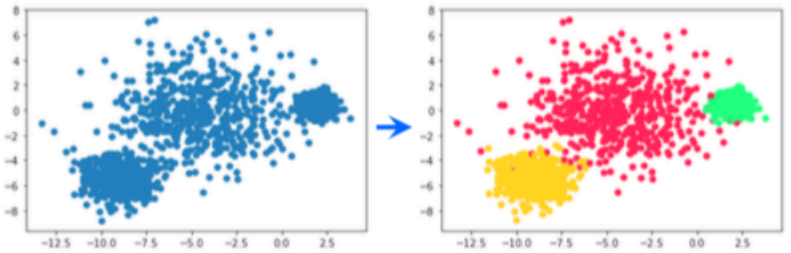
Stanford Dogs dataset
(120 breeds)

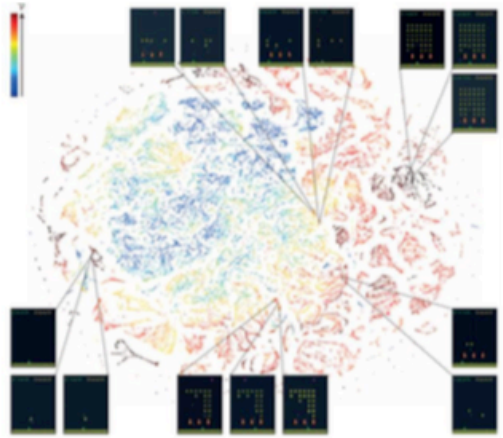
Classification and regression

- 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



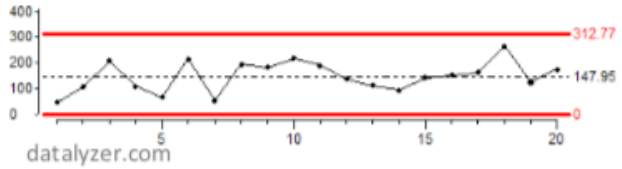


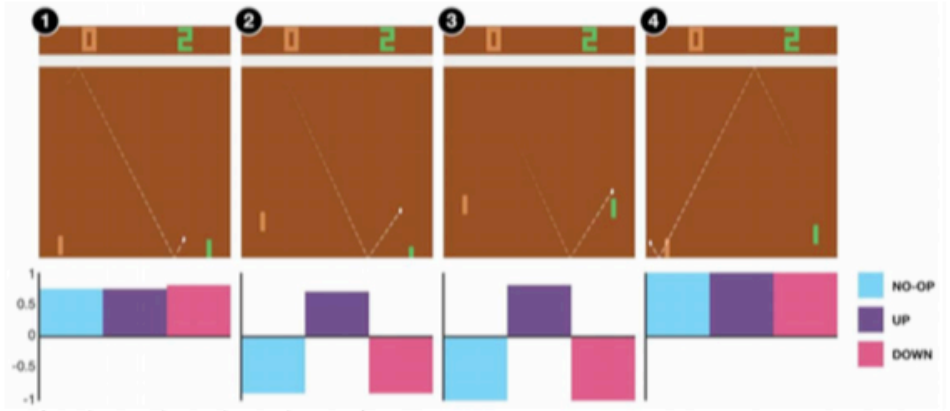
(Mnih, Kavukcuoglu et al., 2015)

Customer Segmentation,
Product Segmentation etc.

Reinforcement learning

- An agent observes the state of the environment, takes actions and gets rewards
- Agent learns by itself to maximize reward





(Mnih, Kavukcuoglu et al., 2015)

Exploratory data analysis by Nick Pogrebnyakov

Reinforcement learning

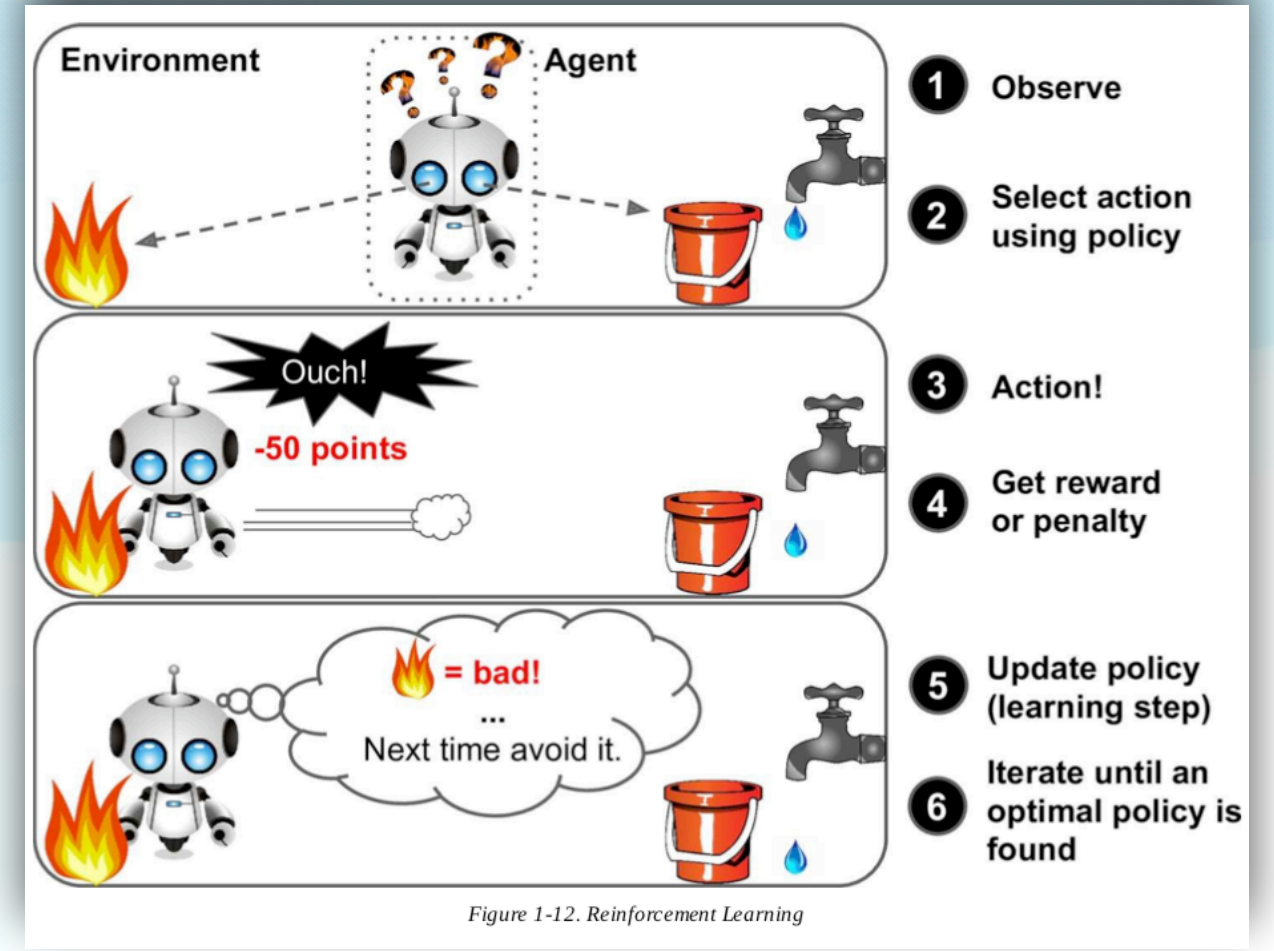
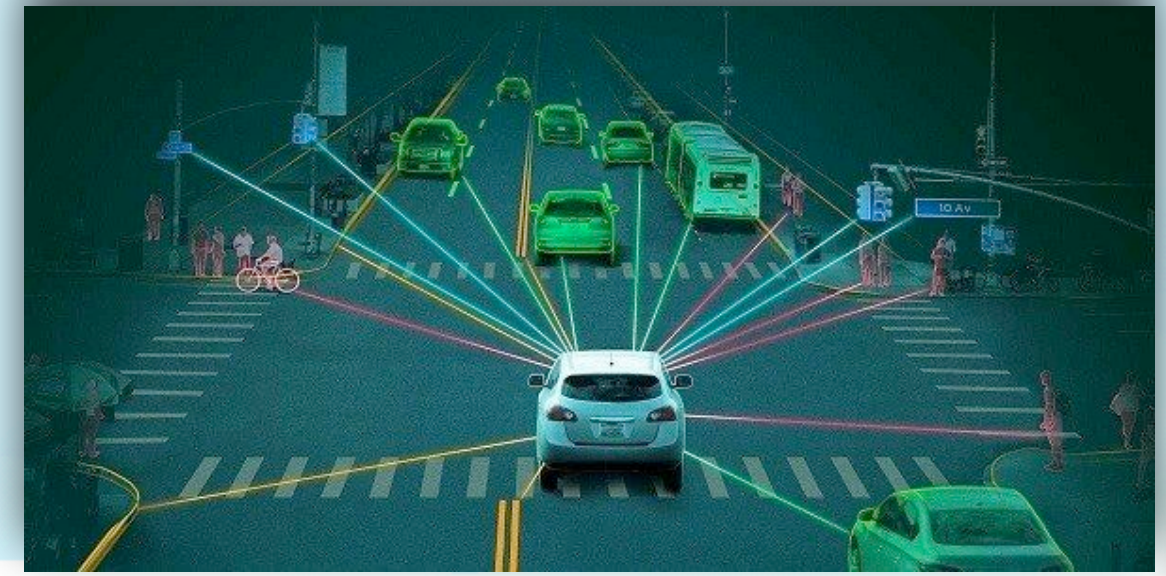
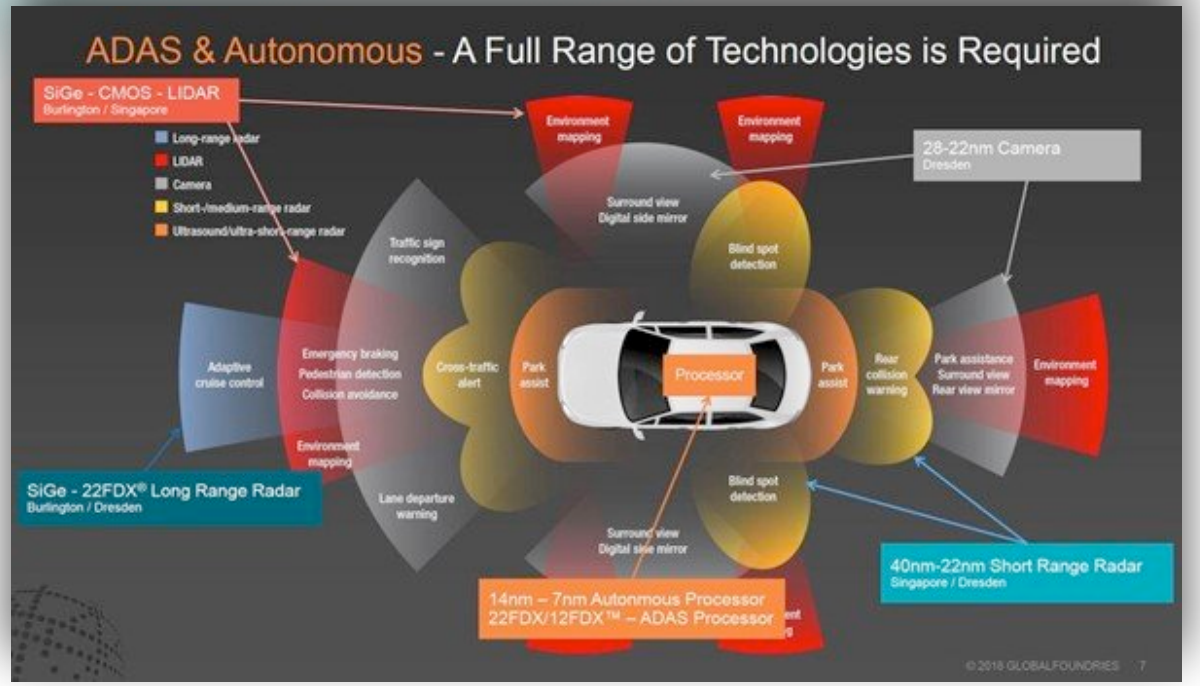


Figure 1-12. Reinforcement Learning

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

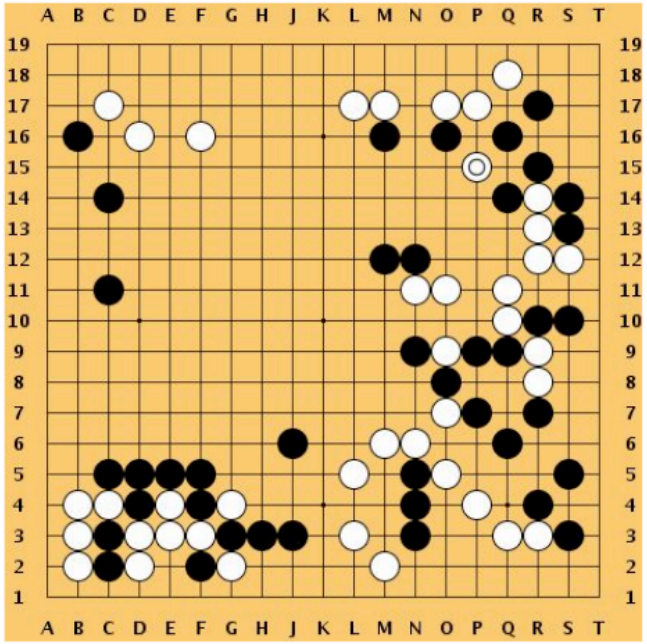
Reinforcement learning - Self Driving car





Reinforcement learning

Go



Objective: Win the game!

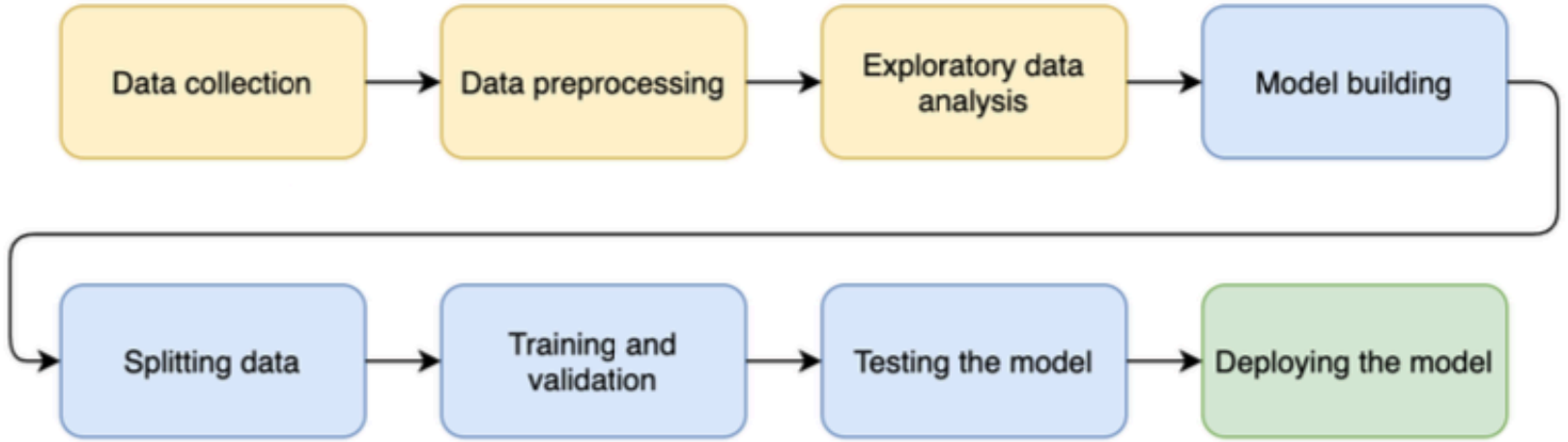
State: Position of all pieces
Action: Where to put the next piece down
Reward: 1 if win at the end of the game, 0 otherwise

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



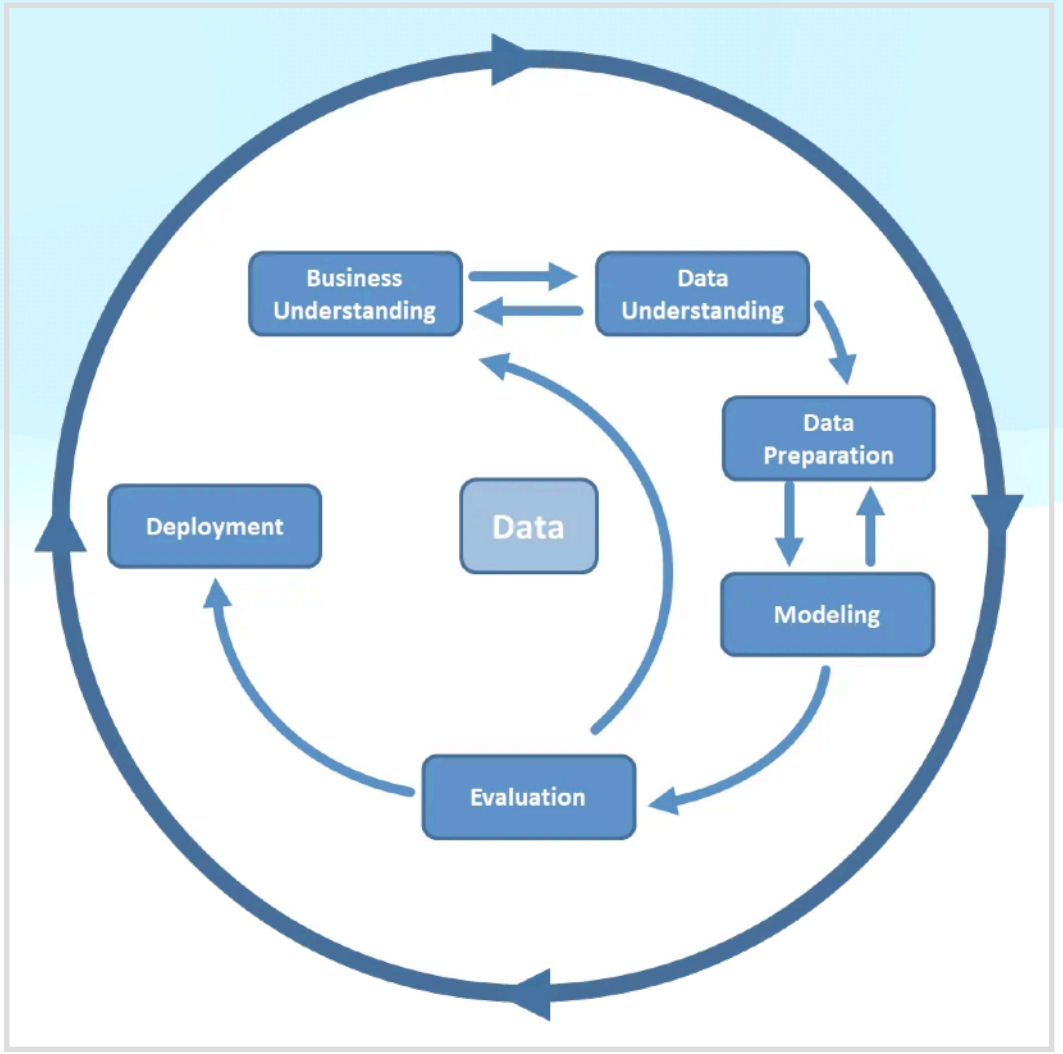
The yellow boxes represent preparation and blue boxes the development of the ML model

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

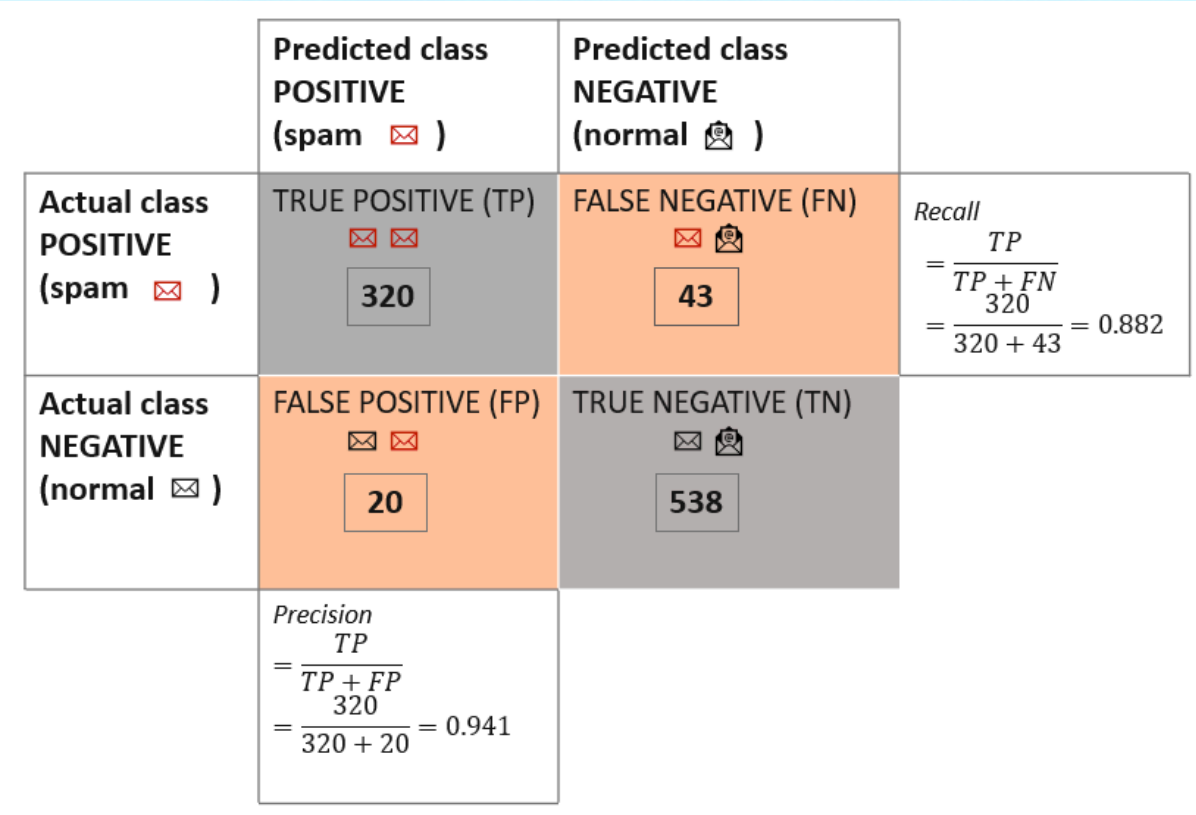
Precision Measures

Performance measures

- 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



- 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

Why Recall 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

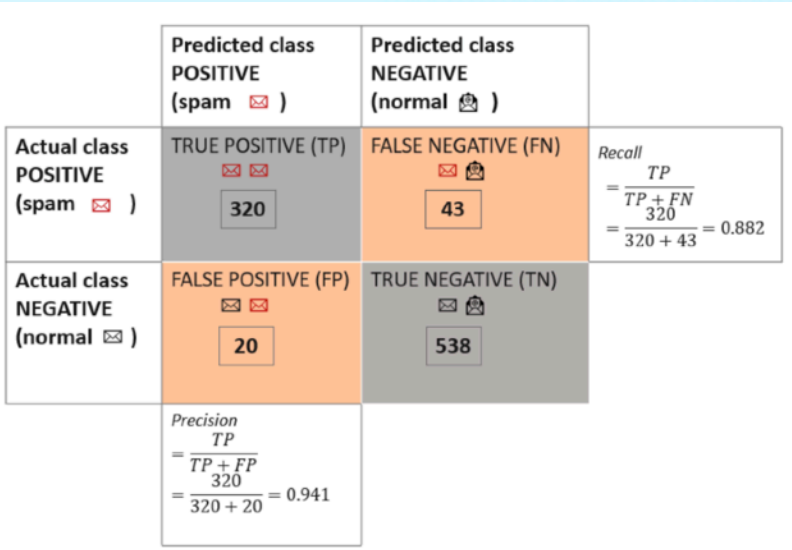
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

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)

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

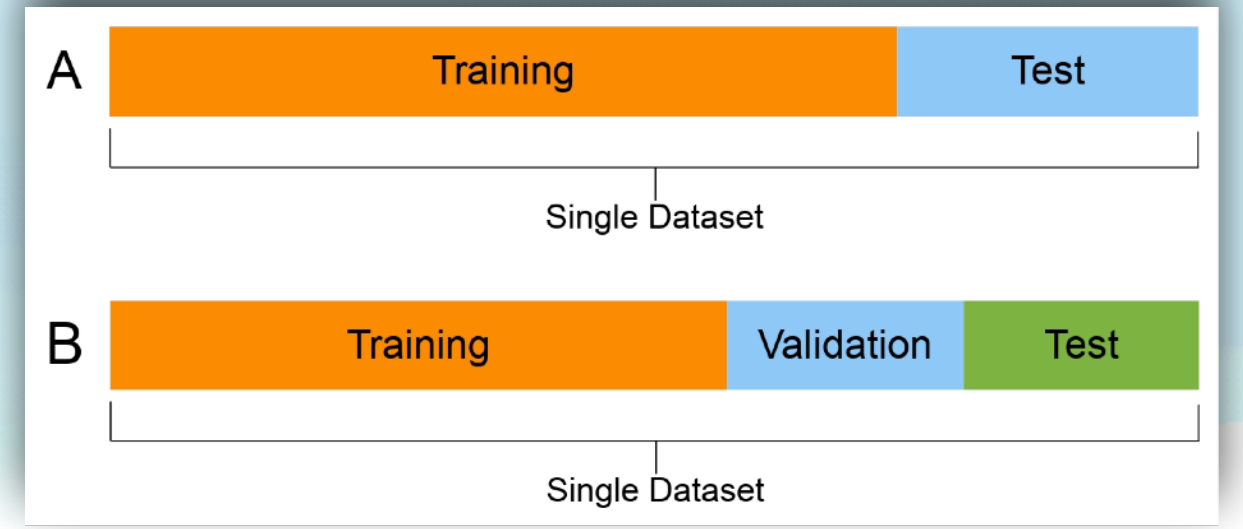
Regression metrics

- 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

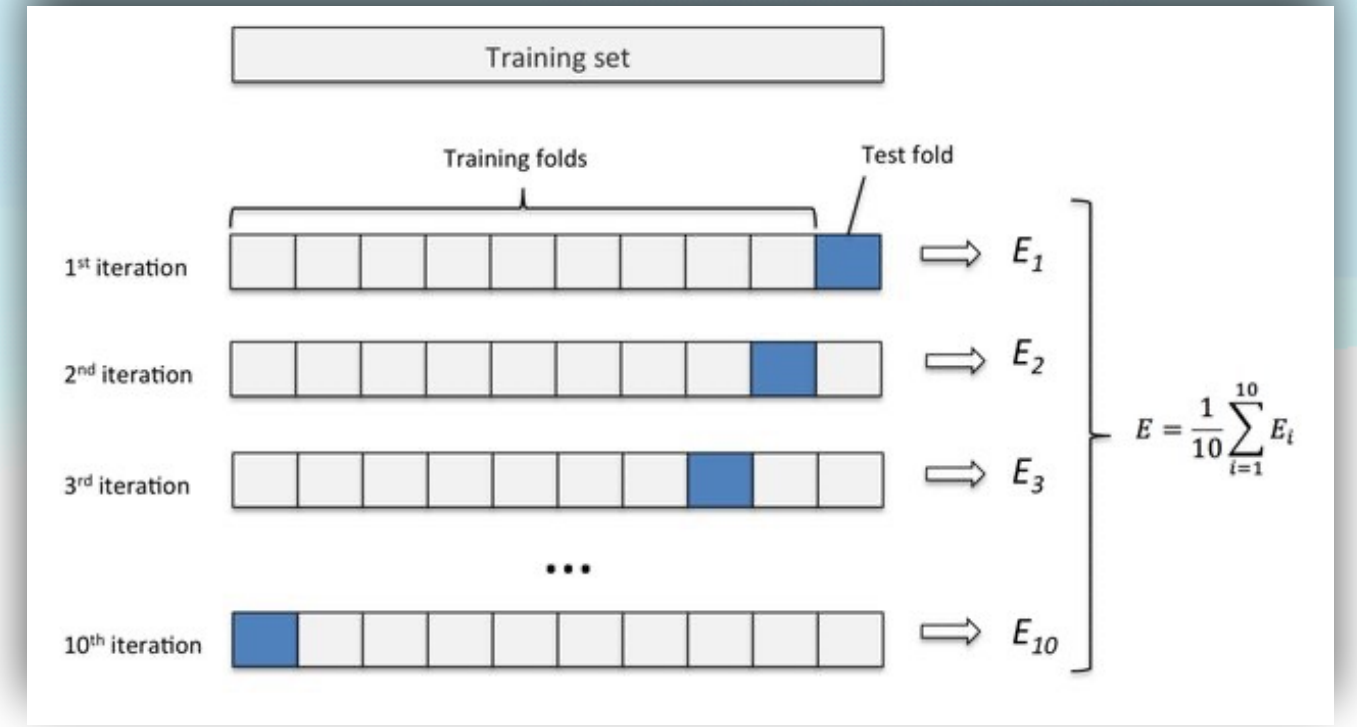
```
X_train, X_test, y_train, y_test =
        train_test_split(…)
```


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

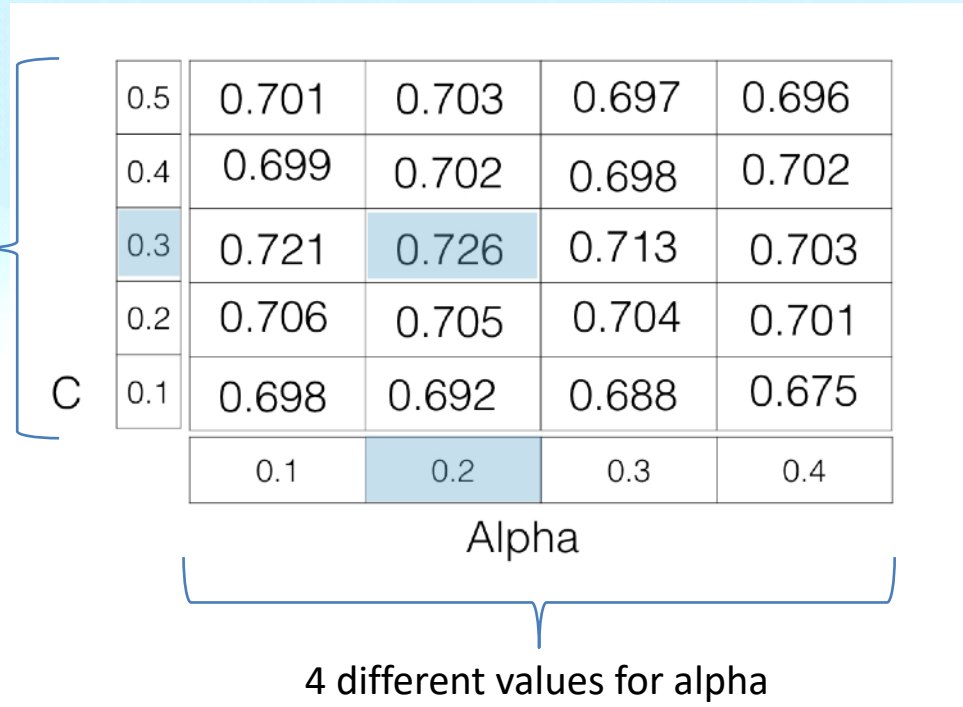


Image by Datacamp

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

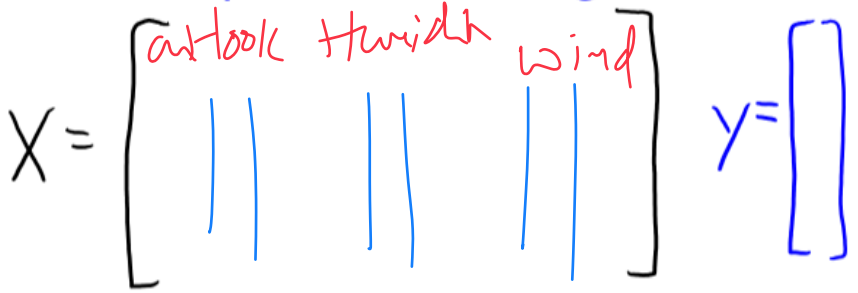
New data:

D15	Rain	High	Weak	?

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

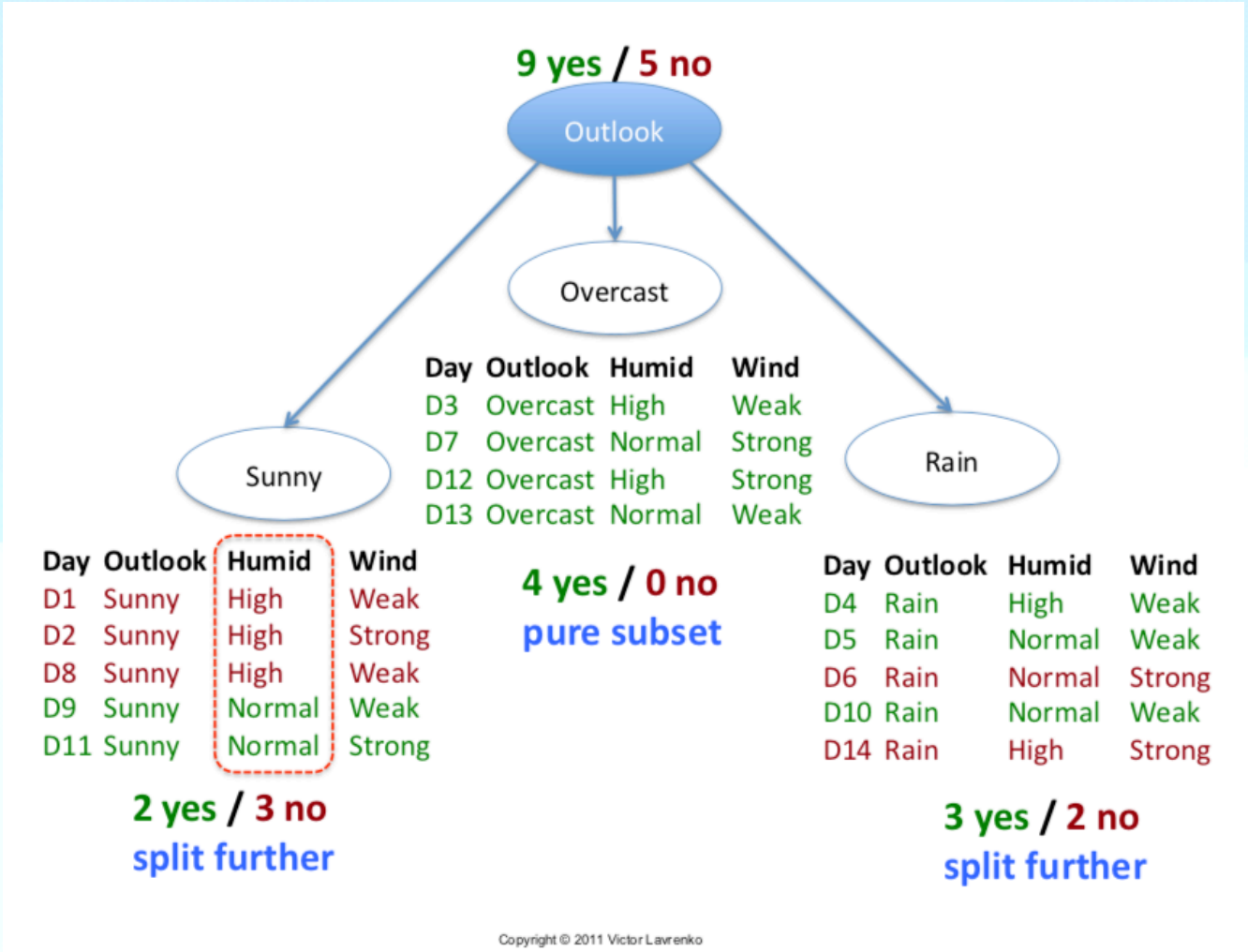
Motivating Example

- We're going to revisit supervised learning:



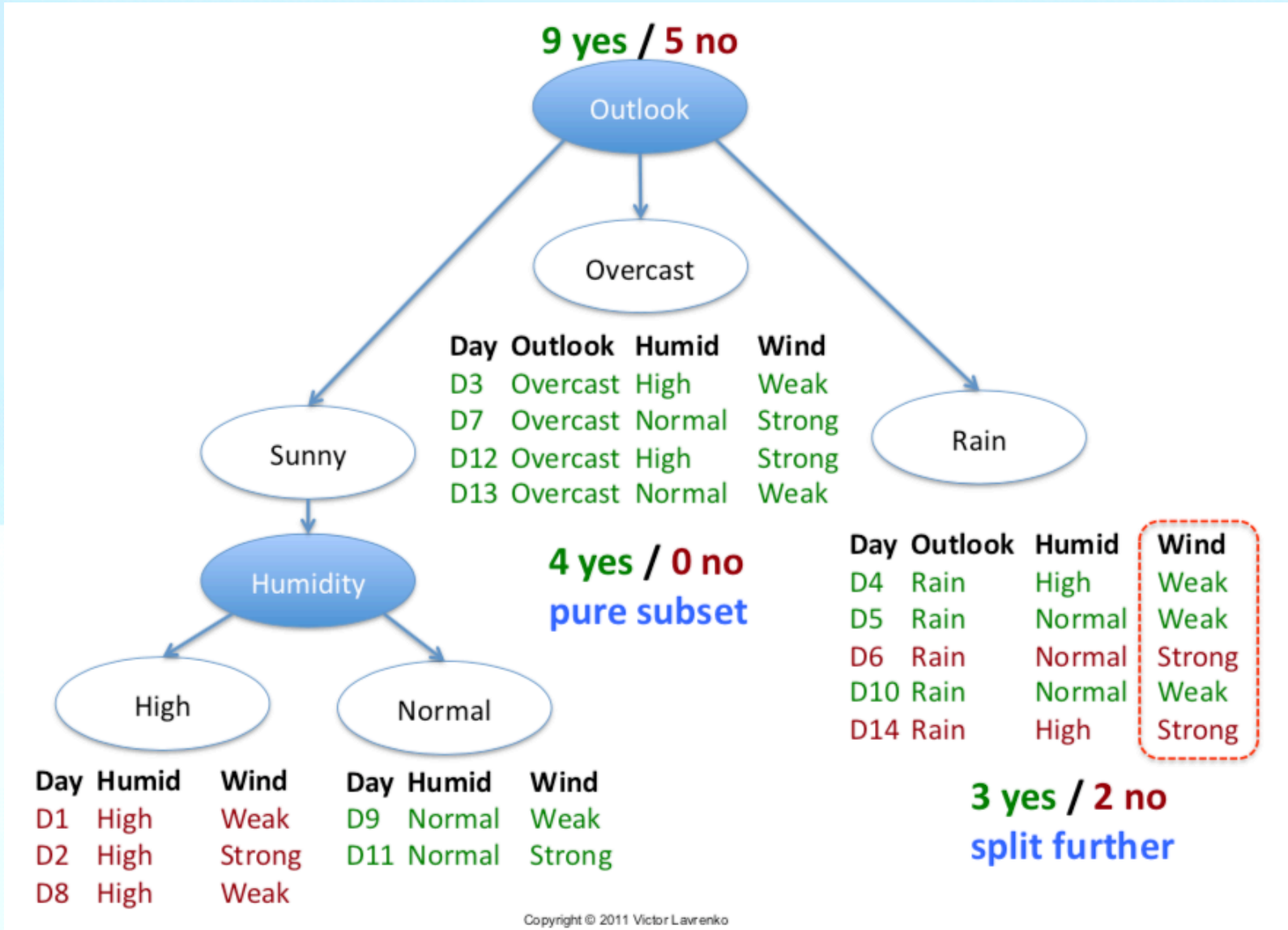
- Previously, we considered classification:
  - We assumed y_i was discrete: y_i = Play or y_i = No Play

Motivating Example - 1



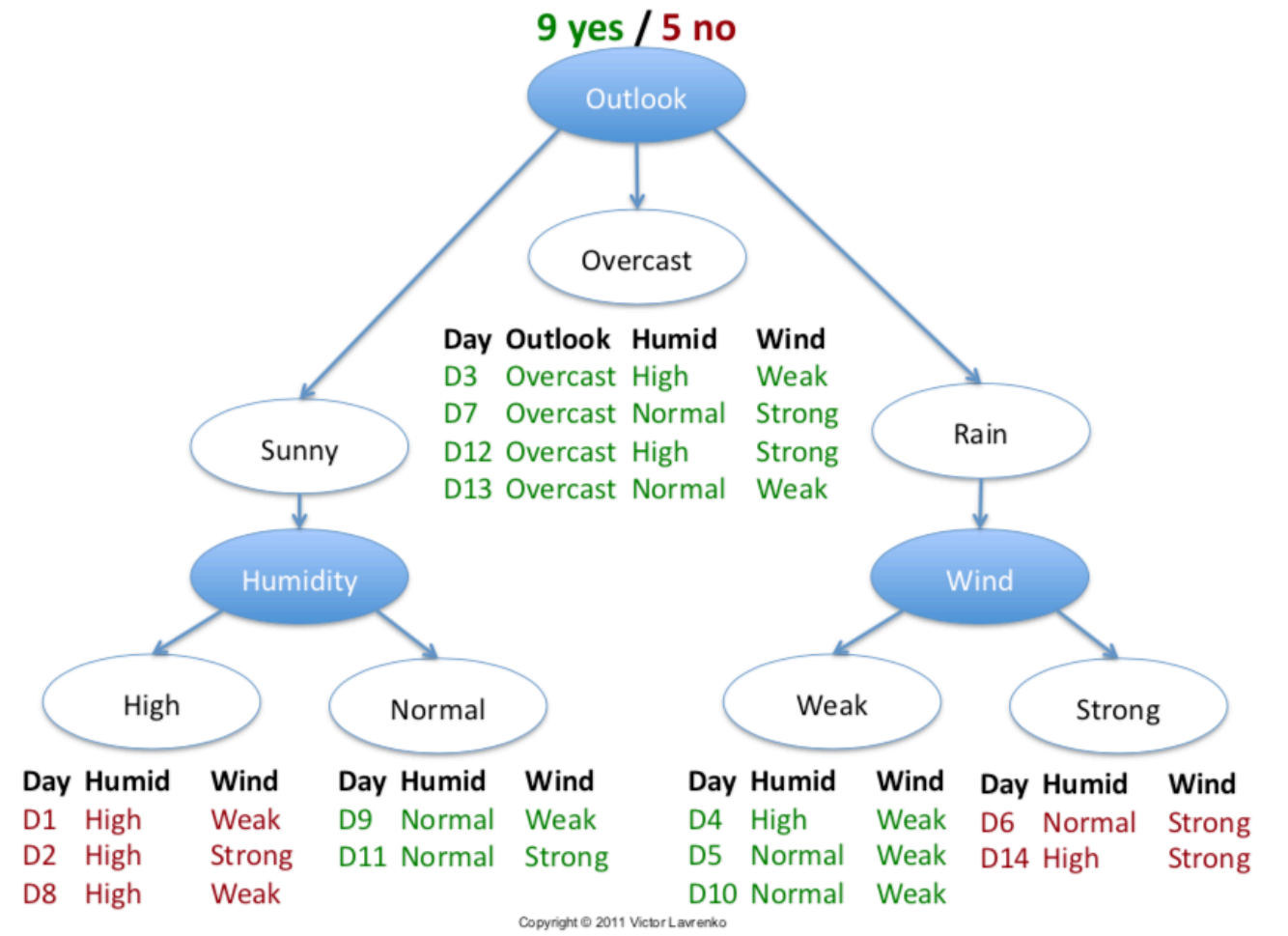
Source: Victor Lavrenko and Charles Sutton
http://www.inf.ed.ac.uk/teaching/courses/iaml/2011/slides/dt.pdf

Motivating Example - 2



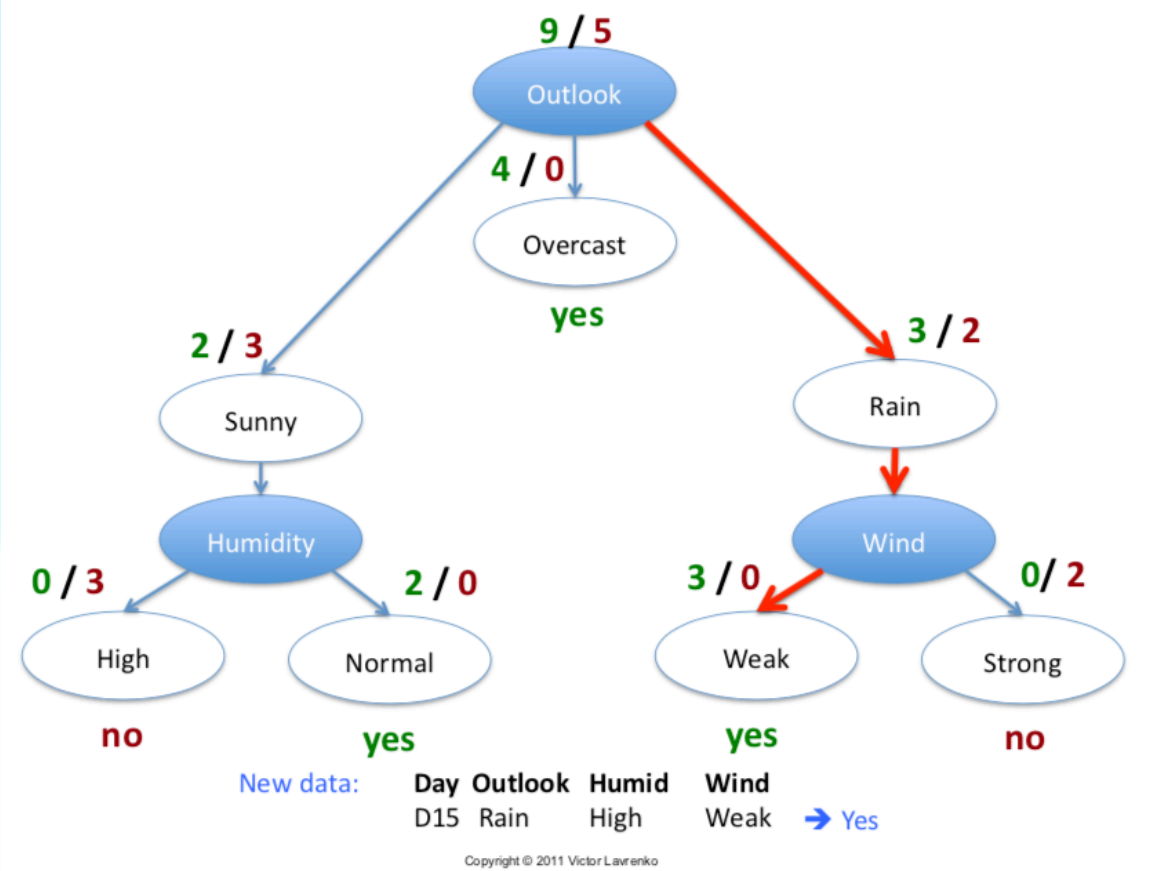
Source: Victor Lavrenko and Charles Sutton
http://www.inf.ed.ac.uk/teaching/courses/iaml/2011/slides/dt.pdf

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

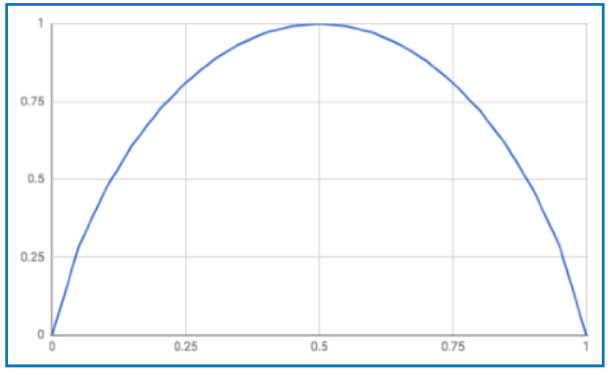
$G = 1 - [\ (3/7)^2+(1/7)^2+(2/7)^2+(1/7)^2\ ] =$ **0.69**

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

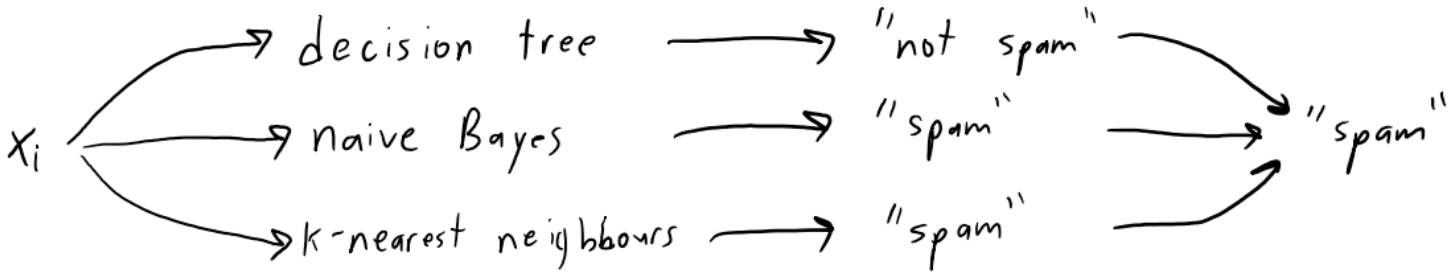
CBS

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).

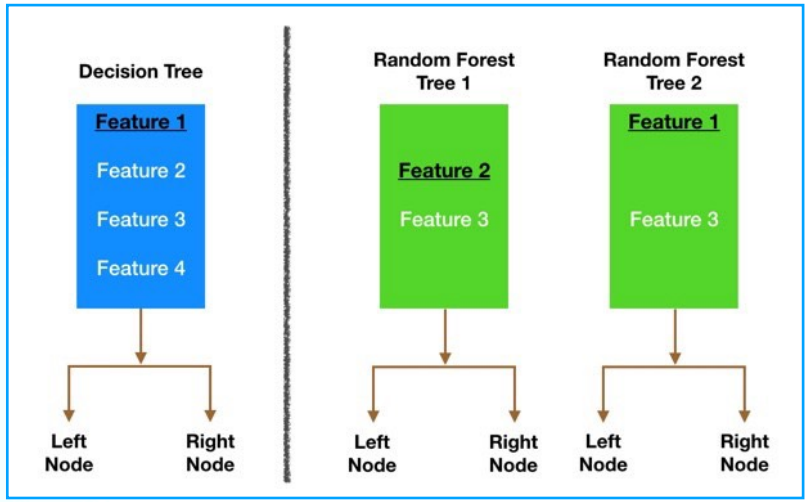


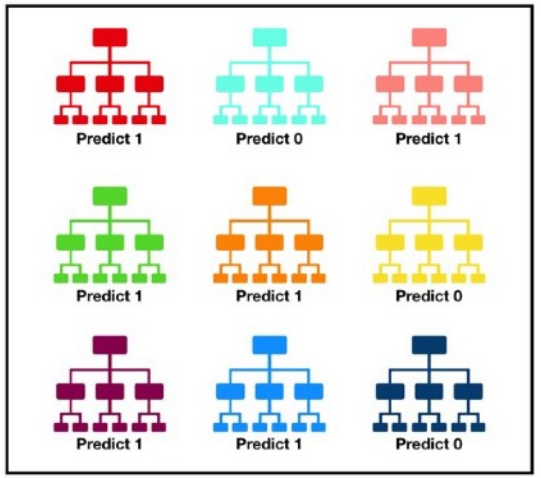
Random Forests

- 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.

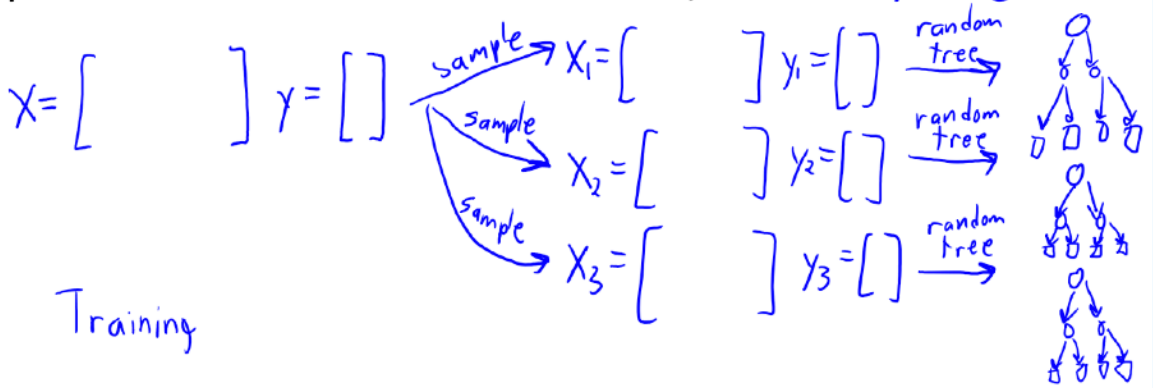




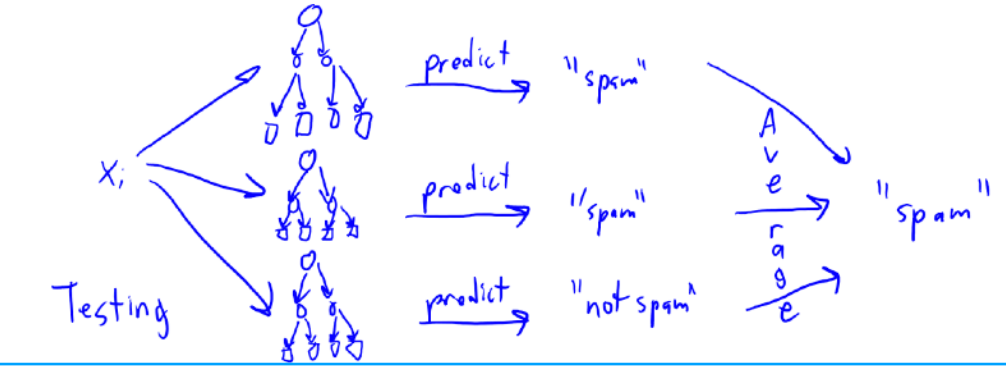
Tally: Six 1s and Three 0s
Prediction: 1

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.



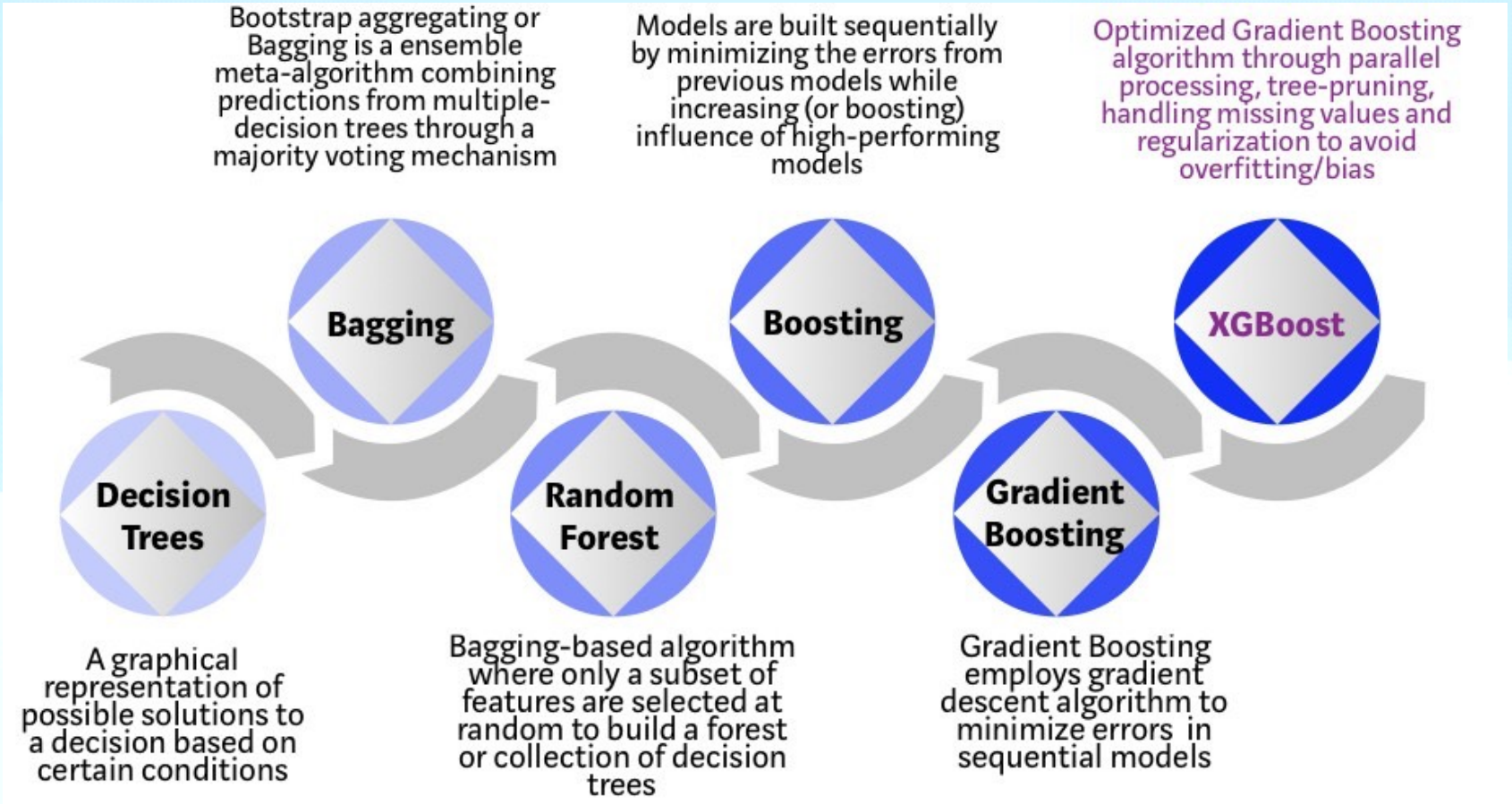
- Random forests are one of the best 'out of the box' classifiers.
- Fit deep decision trees to random bootstrap samples of data, base splits on random subsets of the features, and classify using mode.



Boosting: Key Ideas

- Basic steps:
  1. Fit a classifier on the training data.
  2. Give a higher weight to examples that the classifier got wrong.
  3. Fit a classifier on the weighted training data.
  4. Go back to 2.
- Final prediction: weighted vote of individual classifier predictions.
- Boosted decision trees are very fast/accurate classifiers.
  - "AdaBoost": classic boosting method.
  - "XGBoost": recent method that has been winning Kaggle competitions.

Random Forests



Linear Regression

CBS

Supervised Learning Round 2: Regression

- 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
- Now we're going to consider regression:
  - We allow y_i to be numerical: y_i = 10.34cm.

Regression examples

- We want to discover relationship between numerical variables:
  - Does number of lung cancer deaths change with number of cigarettes?
  - Does how UBC GPA relate to high school GPA?
  - Can I predict your credit score based on your age, occupation, and income?

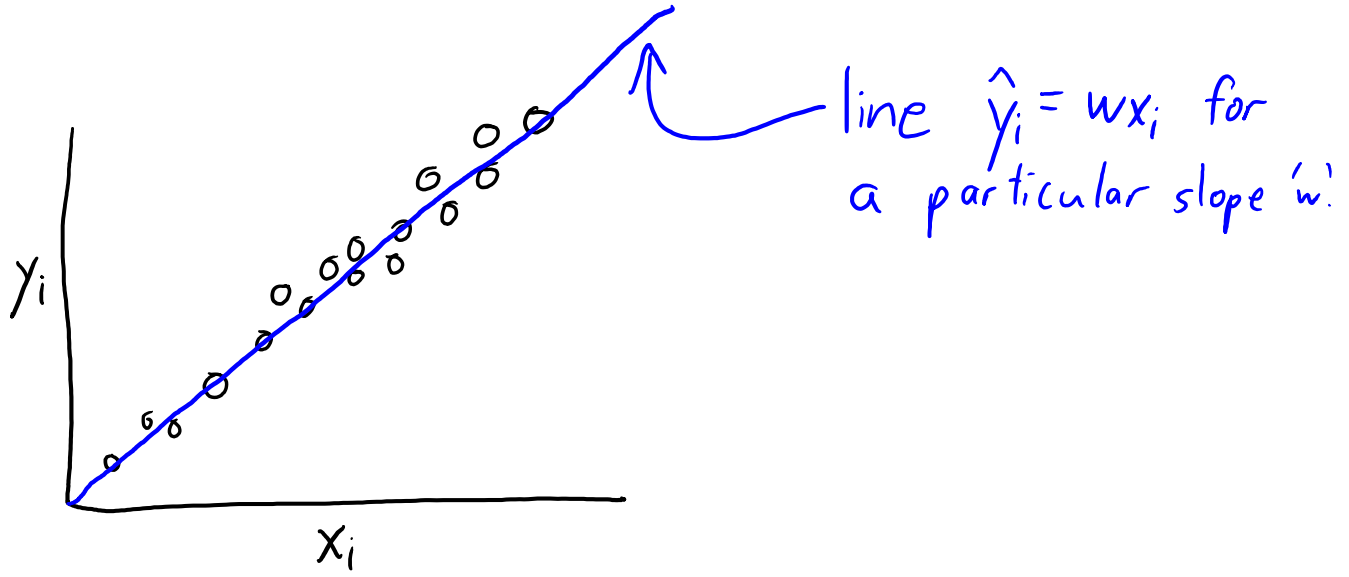
Linear Regression in 1 Dimension

- Assume we only have 1 feature (d = 1):
  - E.g., x_i is number of cigarettes and y_i is number of lung cancer deaths.
- Linear regression makes predictions $\hat{y}_i$ using a linear function of x_i:

$$\hat{y}_i = w x_i$$

- The parameter 'w' is the weight or regression coefficient of x_i.
- As x_i changes, slope 'w' affects the rate that $\hat{y}_i$ increases/decreases:
  - Positive 'w': $\hat{y}_i$ increase as x_i increases.
  - Negative 'w': $\hat{y}_i$ decreases as x_i increases.

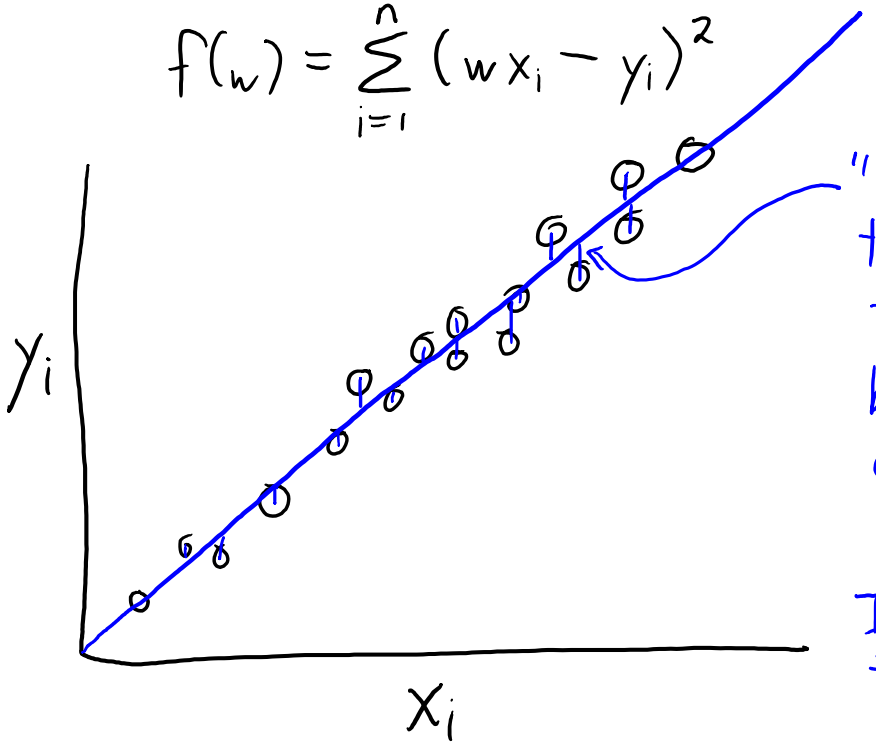
Linear Regression in 1 Dimension



Least Squares Objective

- Classic way to set slope 'w' is minimizing sum of squared errors:

$$f(w) = \sum_{i=1}^{n} (w x_i - y_i)^2$$



"Error" is the sum of the squared values of these vertical distances between the line (wx_i) and the targets (y_i)

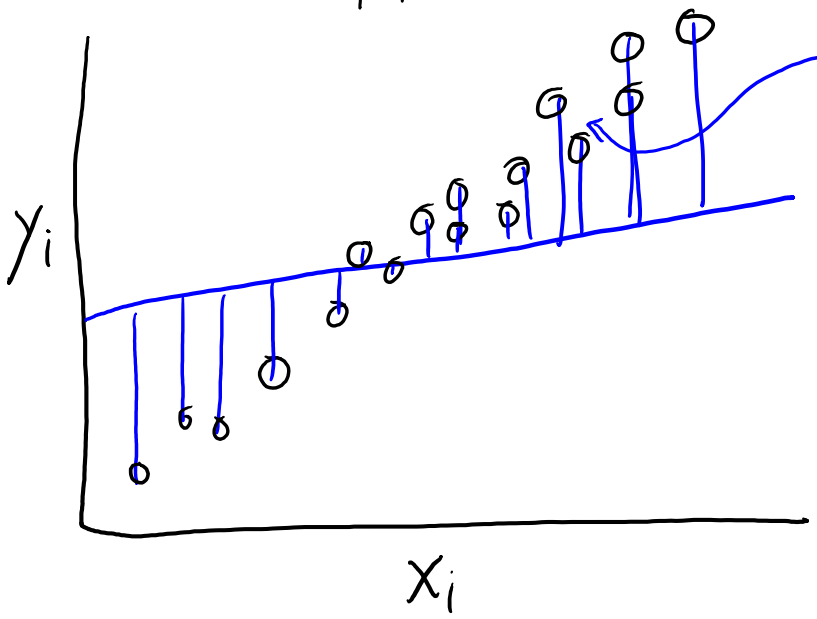
↓

If this error is small, then our predictions are close to the targets.

Least Squares Objective

- Classic way to set slope 'w' is minimizing sum of squared errors:

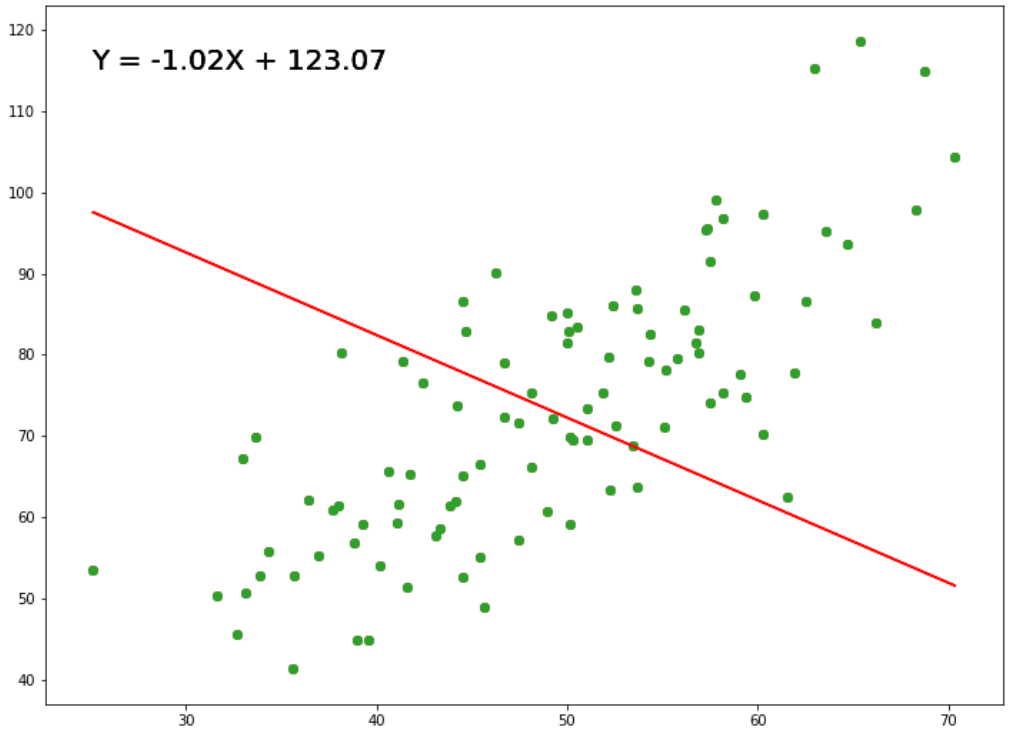
$$f(w) = \sum_{i=1}^{n} (w x_i - y_i)^2$$



"Error" is the sum of the squared values of these vertical distances between the line (wx_i) and the targets (y_i)

↓

If this error is large, then our predictions are far from the targets.



You can run the visualisation at:
https://towardsdatascience.com/linear-regression-using-least-squares-a4c3456e8570

Motivation: Combining Explanatory Variables

- Smoking is not the only contributor to lung cancer.
  - For example, environmental factors like exposure to asbestos.
- How can we model the combined effect of smoking and asbestos?
- A simple way is with a 2-dimensional linear function:

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2}$$

"weight" of feature 1

Value of feature 1 in example 'i'

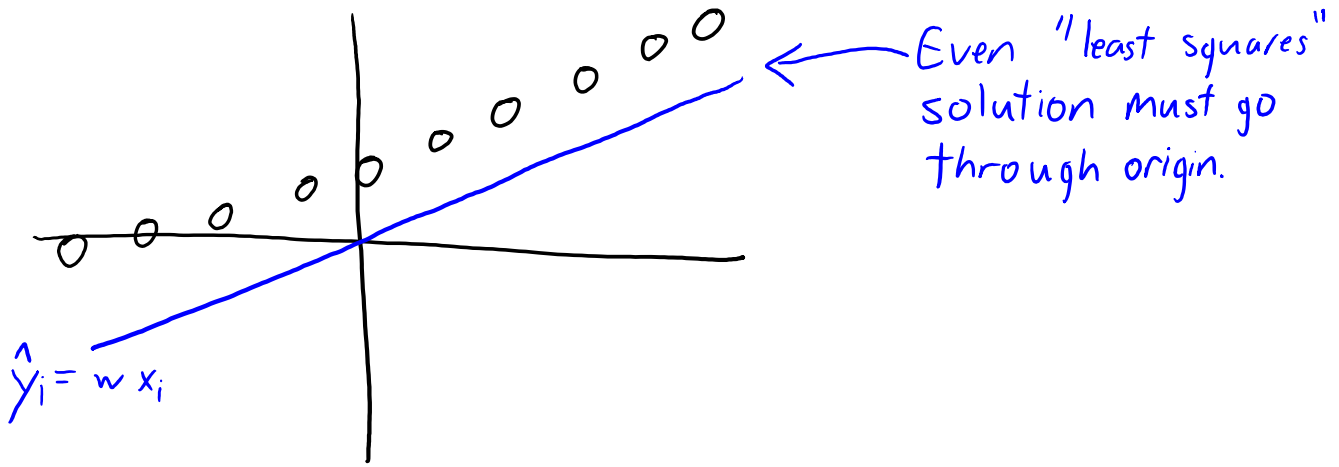
"weight" on feature 2.

Value of feature 2 in example 'i'

- We have a weight w_1 for feature '1' and w_2 for feature '2'.

Modeling a y-intercept?

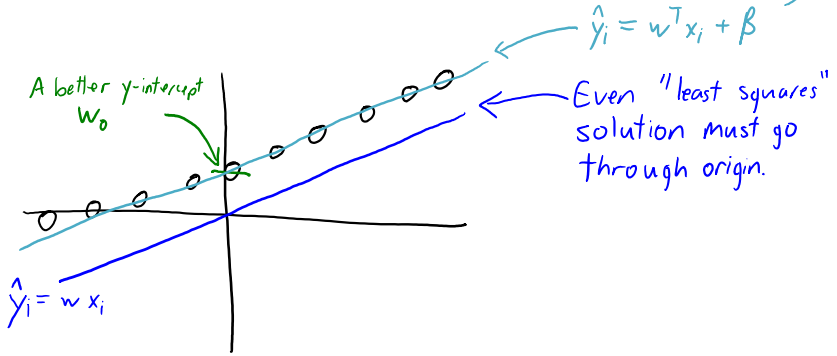
- Linear model is $\hat{y}_i = wx_i$ instead of $\hat{y}_i = wx_i + \beta$ with y-intercept β.
- Without an intercept, if $x_i = 0$ then we must predict $\hat{y}_i = 0$.



Modeling a y-intercept?

- Linear model is $\hat{y}_i = wx_i$ instead of $\hat{y}_i = wx_i + \beta$ with y-intercept β.
- Without an intercept, if $x_i = 0$ then we must predict $\hat{y}_i = 0$.

Adding y-intercept fixes this.



w_0 (basic salary) = 30 000

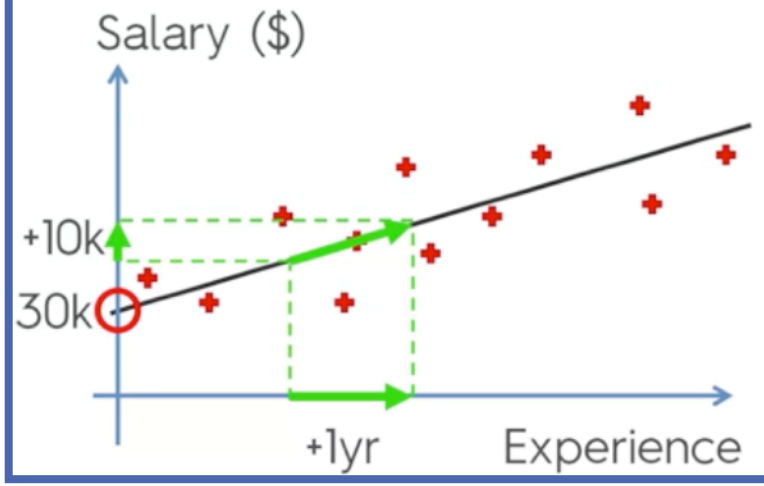
w_1 (yearly raise 10%) = 3000

x (no of years) = 3

Salary = 30000 + 3000 x 3

y = 39000 DKK

Simple Linear Regression:



$y = b_0 + b_1 * x$

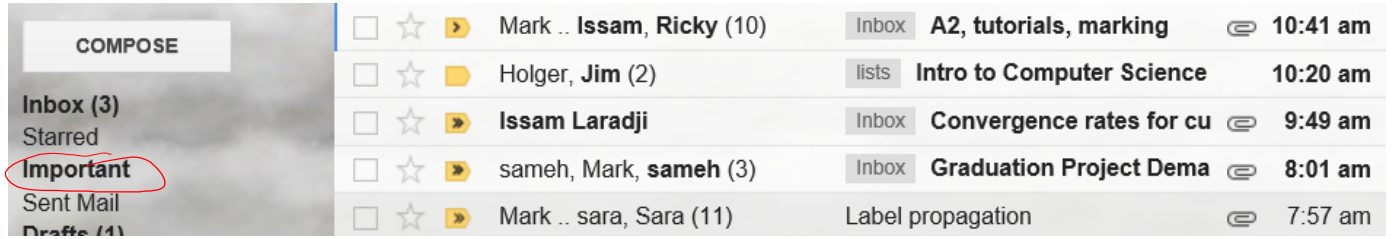
Salary = b_0 + b_1 *Experience

Logistic Regression

CBS

Motivation: Identifying Important E-mails

- How can we automatically identify ‘important’ e-mails?



- A binary classification problem (“important” vs. “not important”).
  - Labels are approximated by whether you took an “action” based on mail.
  - High-dimensional feature set (that we’ll discuss later).

- Gmail uses a linear classifier for this problem.

Binary Classification Using Regression?

- Can we apply linear models for binary classification?
  - Set $y_i = +1$ for one class ("important").
  - Set $y_i = -1$ for the other class ("not important").
- At training time, fit a linear regression model:

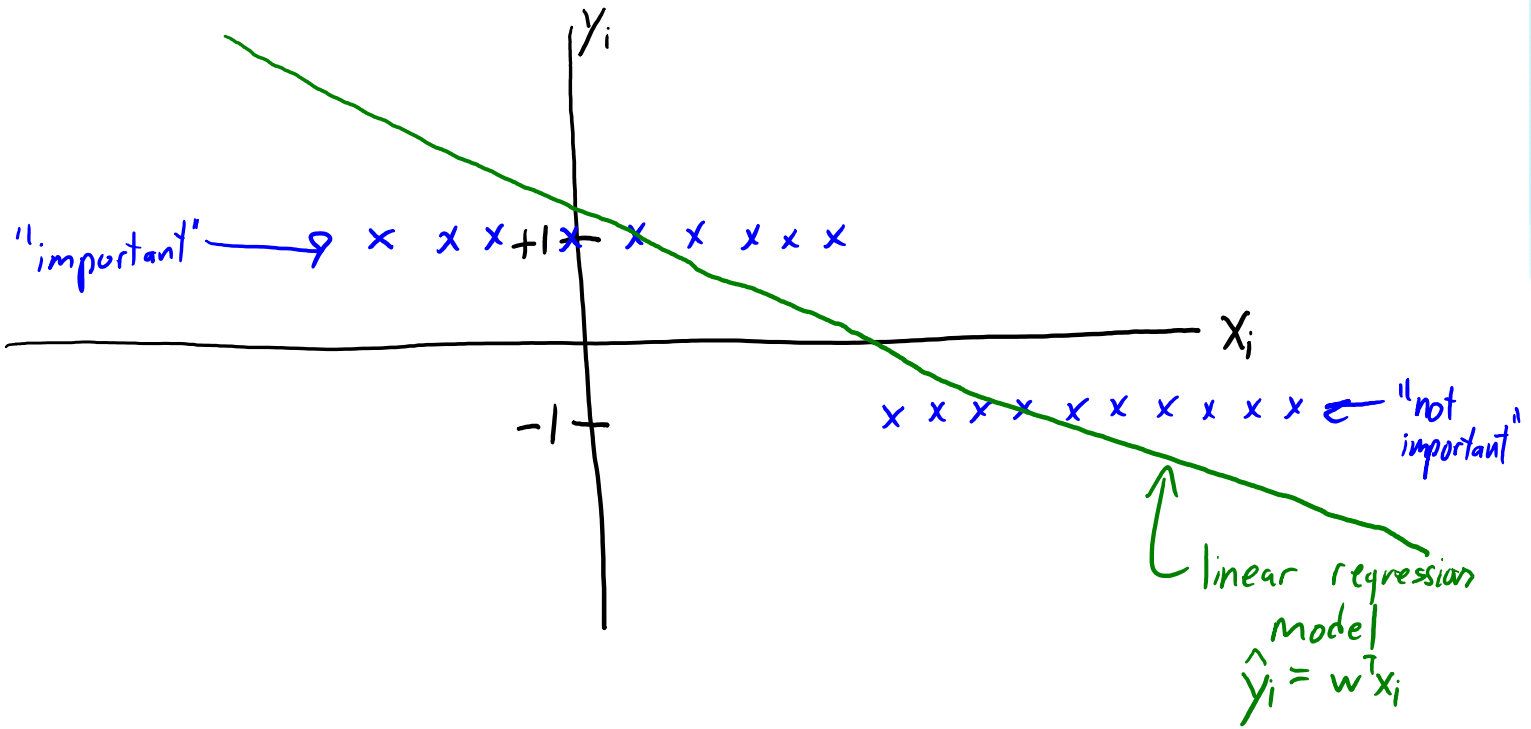
$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id}$$
$$= w^T x_i$$

- The model will try to make $w^T x_i = +1$ for "important" e-mails, and $w^T x_i = -1$ for "not important" e-mails.

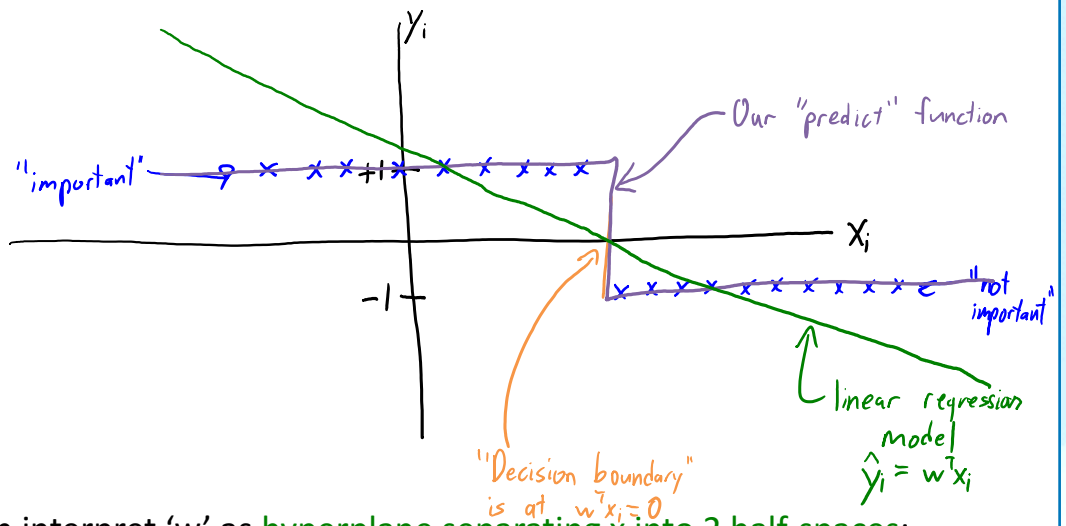
Binary Classification Using Regression?

- Can we apply linear models for binary classification?
  - Set $y_i = +1$ for one class ("important").
  - Set $y_i = -1$ for the other class ("not important").
- Linear model gives real numbers like 0.9, -1.1, and so on.
- So to predict, we look at the sign of $w^T x_i$.
  - If $w^T x_i = 0.9$, predict $\hat{y}_i = +1$.
  - If $w^T x_i = -1.1$, predict $\hat{y}_i = -1$.
  - If $w^T x_i = 0.1$, predict $\hat{y}_i = +1$.
  - If $w^T x_i = -100$, predict $\hat{y}_i = -1$.

Decision Boundary in 1D

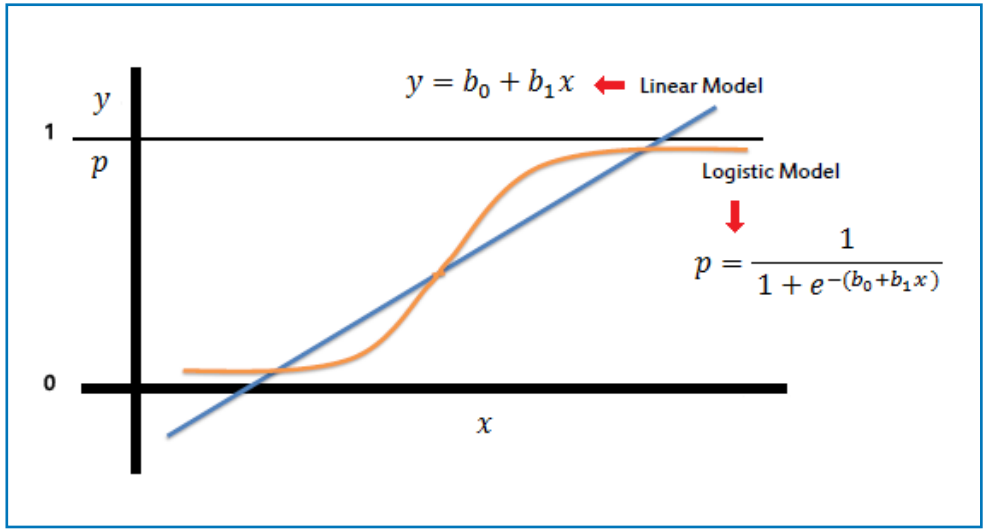


Decision Boundary in 1D



- We can interpret 'w' as hyperplane separating x into 2 half-spaces:
  - Half-space where $w^T x_i > 0$ and half-space where $w^T x_i < 0$.

Sigmoid function



Artificial Neural Networks and Deep Learning

CBS

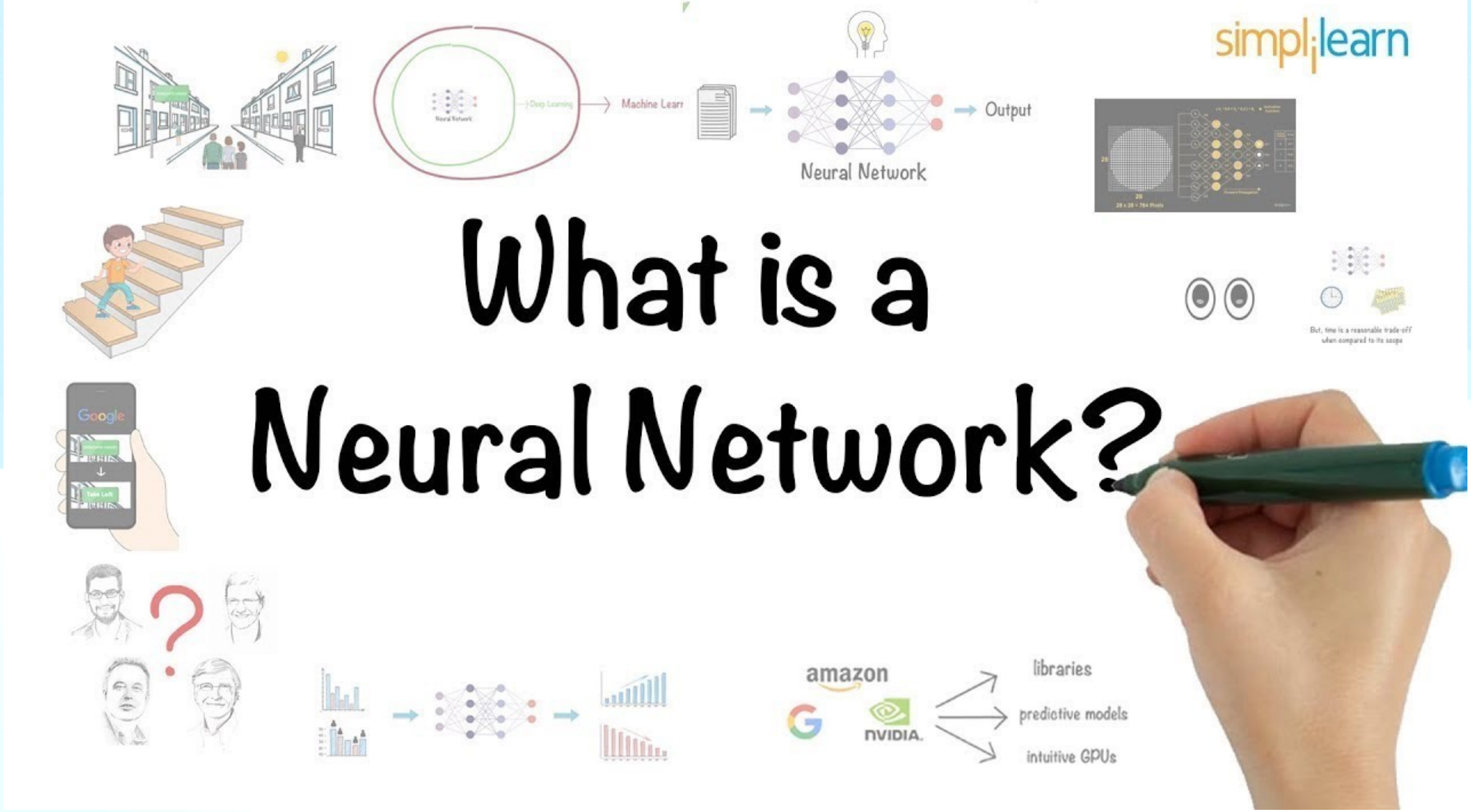
Artificial Neural Networks

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

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

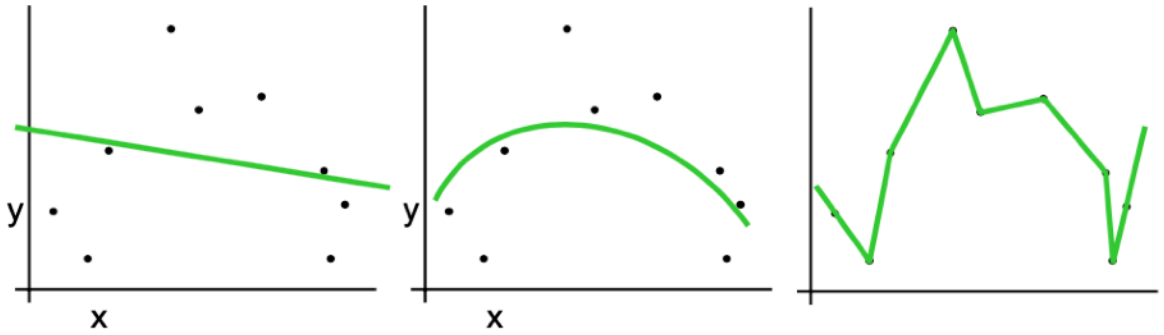
3BLUE1BROWN SERIES S3 • E1: But what is a Neural Network? | Deep learning, chapter 1

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



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

Which is best?



Why not choose the method with the best fit to the data?

Source: www.cs.cmu.edu/~awm

Introducing Non-Linearity

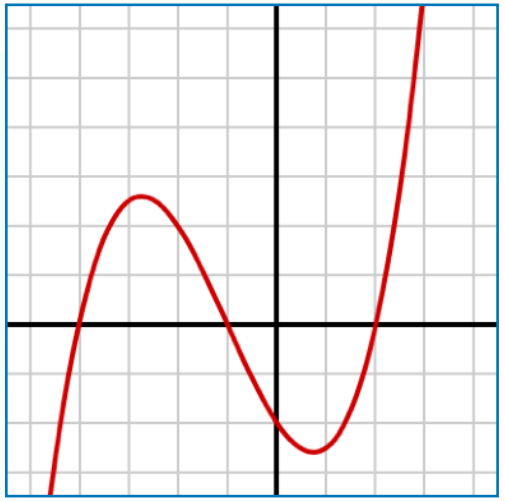
- To increase flexibility, something needs to be non-linear.
- Typical choice: transform z_i by non-linear function 'h'.

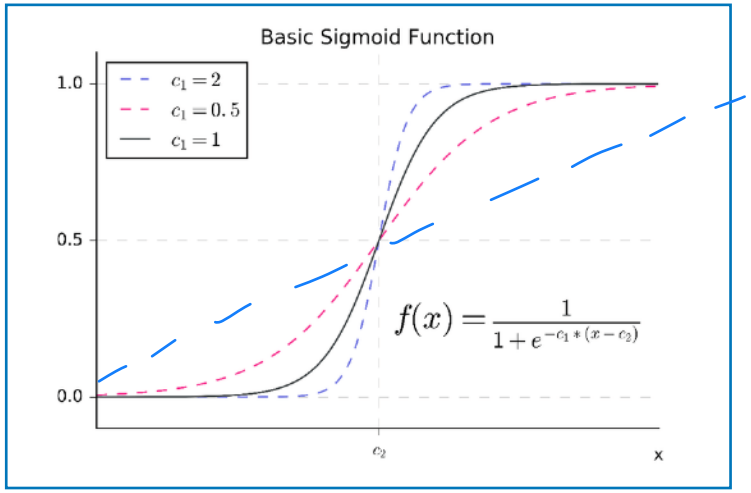
$$z_i = W x_i \qquad y_i = v^T h(z_i)$$

  - Here the function 'h' transforms 'k' inputs to 'k' outputs.
- Common choice for 'h': applying sigmoid function element-wise:

$$h(z_{ic}) = \frac{1}{1 + \exp(-z_{ic})}$$

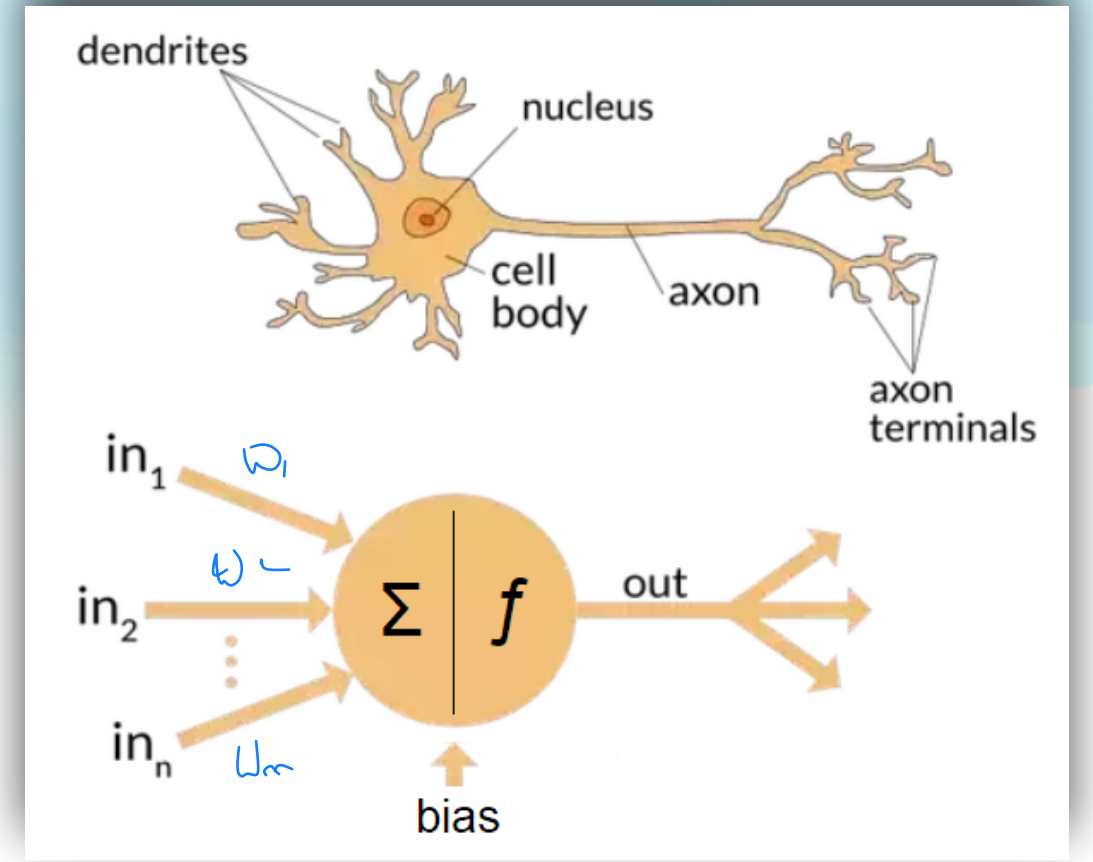
- So this takes the z_{ic} in $(-\infty,\infty)$ and maps it to (0,1).
- This is called a "multi-layer perceptron" or a "neural network".





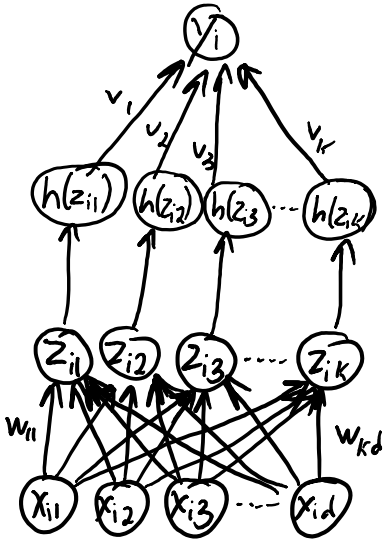
Even if machine learning can solve problems and beat humans in certain fields, it **does not mean** that these algorithms behave like humans and are just as equally able in other fields.

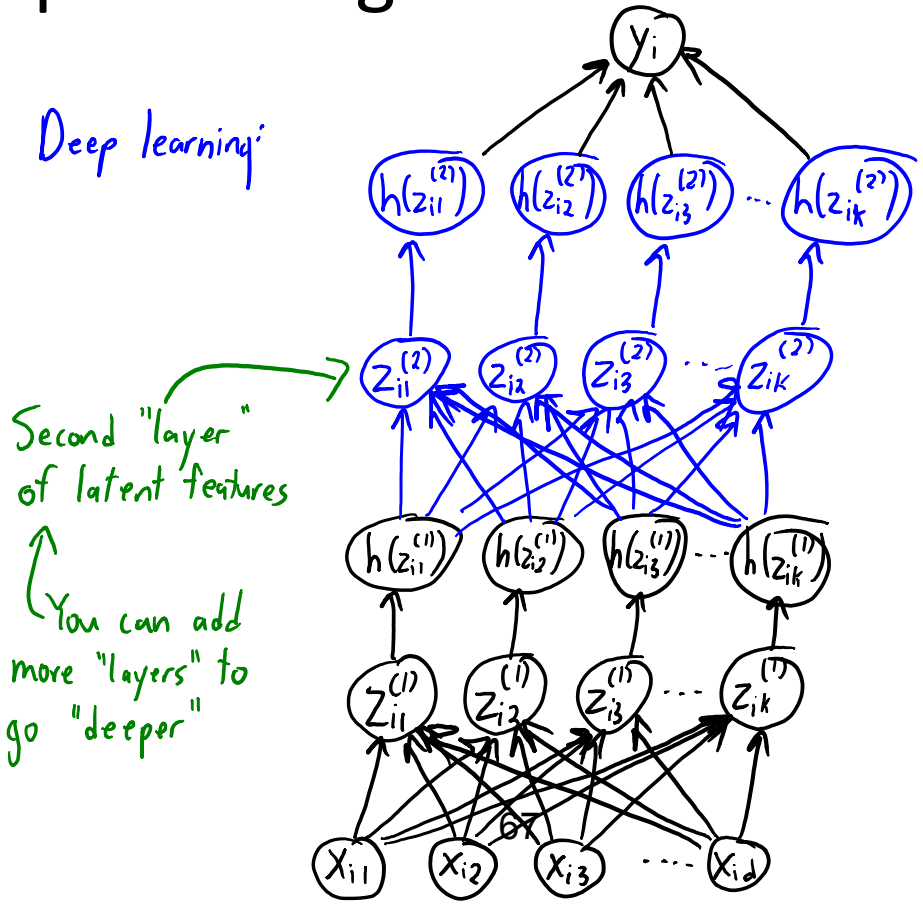
It's important to understand that AI isn't a replacement for human intelligence. It's an extension of it. It can help us to make better decisions and to be more productive, but it can't replace the value of human intuition and creativity.



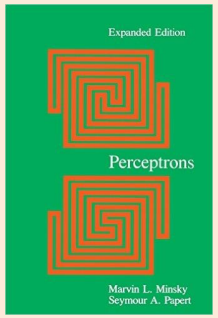
https://www.calls9.com/blogs/can-artificial-intelligence-replace-human-intelligence

Deep Learning

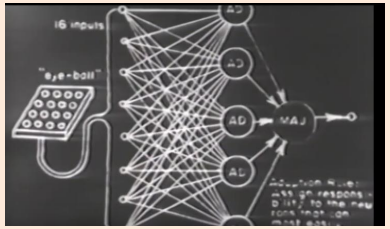




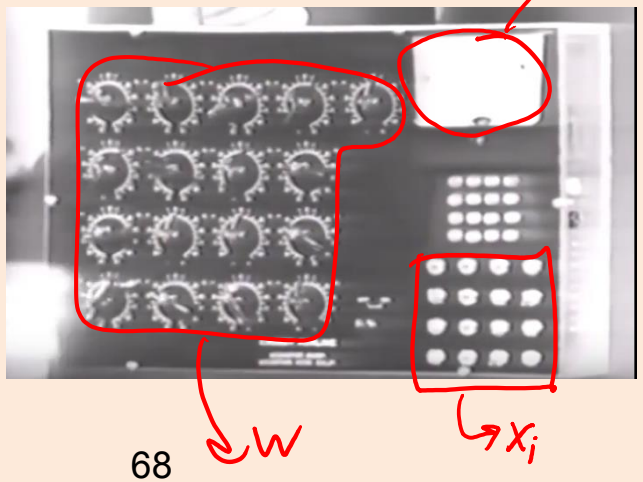
ML and Deep Learning History



- 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



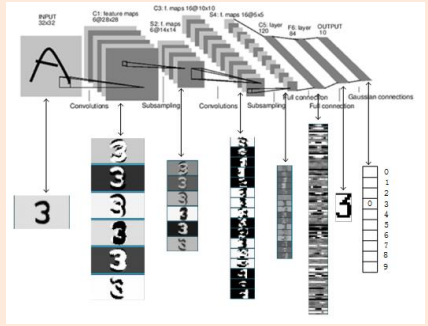
- Then drop in popularity:
  - Quickly realized limitations of linear models.

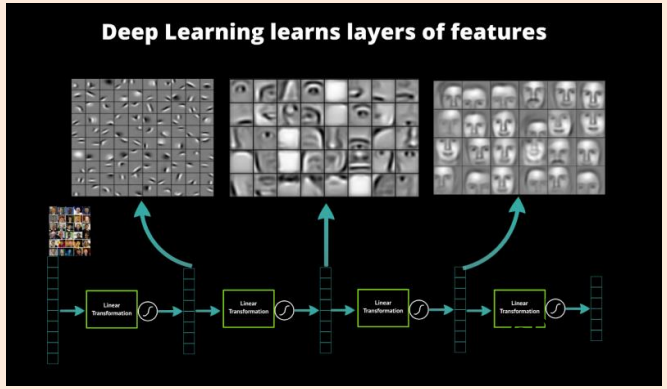


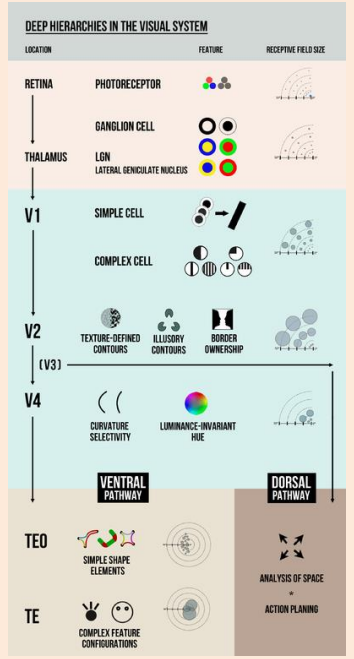
https://mitpress.mit.edu/books/perceptrons/

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.







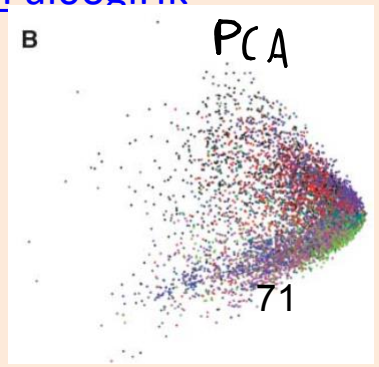
https://en.wikibooks.org/wiki/Sensory_Systems/Visual_Signal_Processing
http://www.datarobot.com/blog/a-primer-on-deep-learning/
http://blog.csdn.net/strint/article/details/44163869

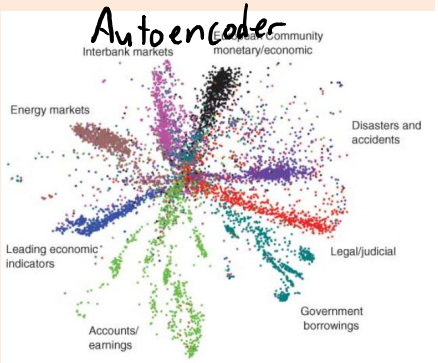
ML and Deep Learning History

- 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.

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

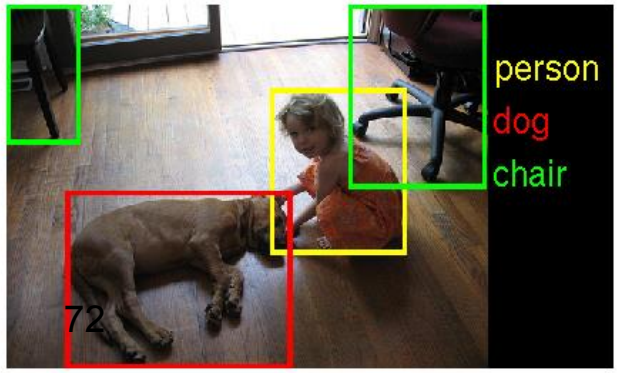




https://www.cs.toronto.edu/~hinton/science.pdf

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/

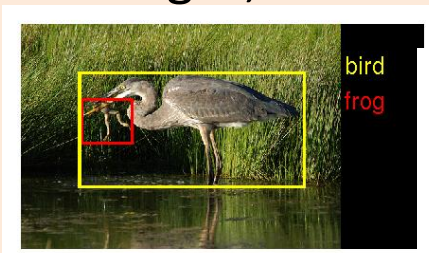
2010s: DEEP LEARNING!!!

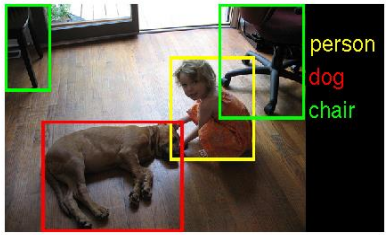
- Media hype:
  - "How many computers to identify a cat? 16,000"
    New York Times (2012).
  - "Why Facebook is teaching its machines to think like humans"
    Wired (2013).
  - "What is 'deep learning' and why should businesses care?"
    Forbes (2013).
  - "Computer eyesight gets a lot more accurate"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

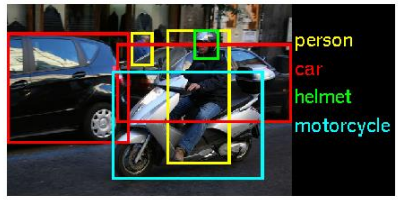
- 2015: huge improvement in language understanding.

ImageNet Challenge

- Millions of labeled images, 1000 object classes.





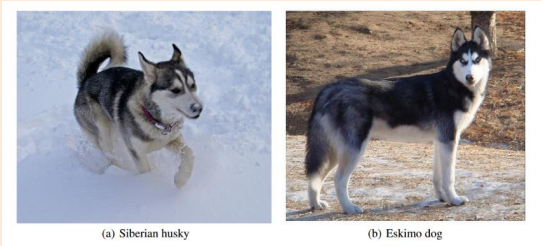


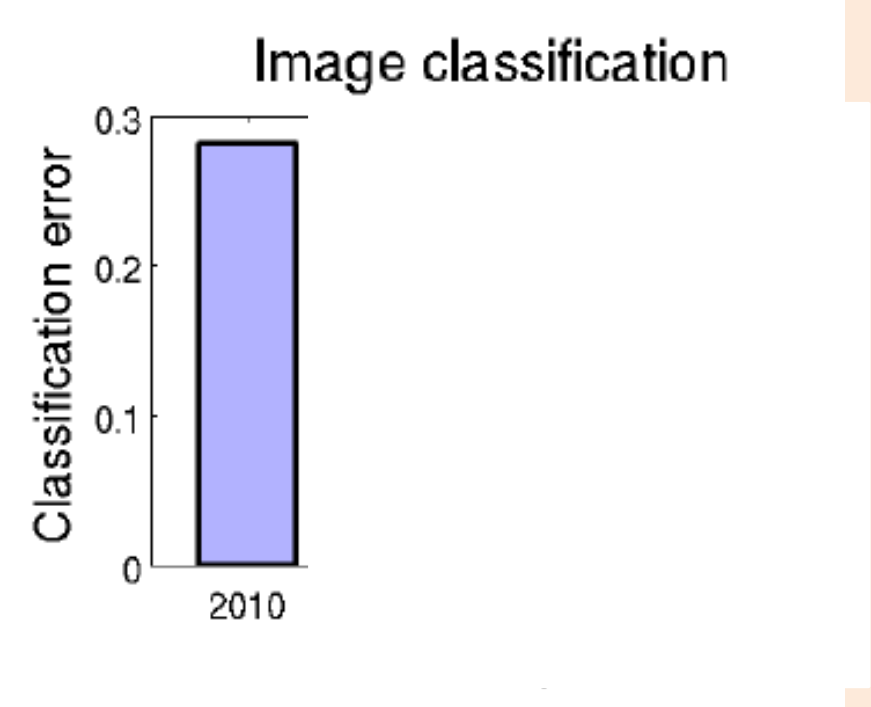
Easy for humans but hard for computers.

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

ImageNet Challenge

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

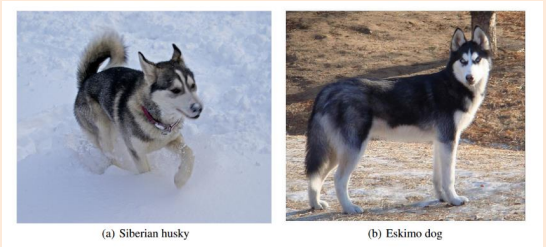


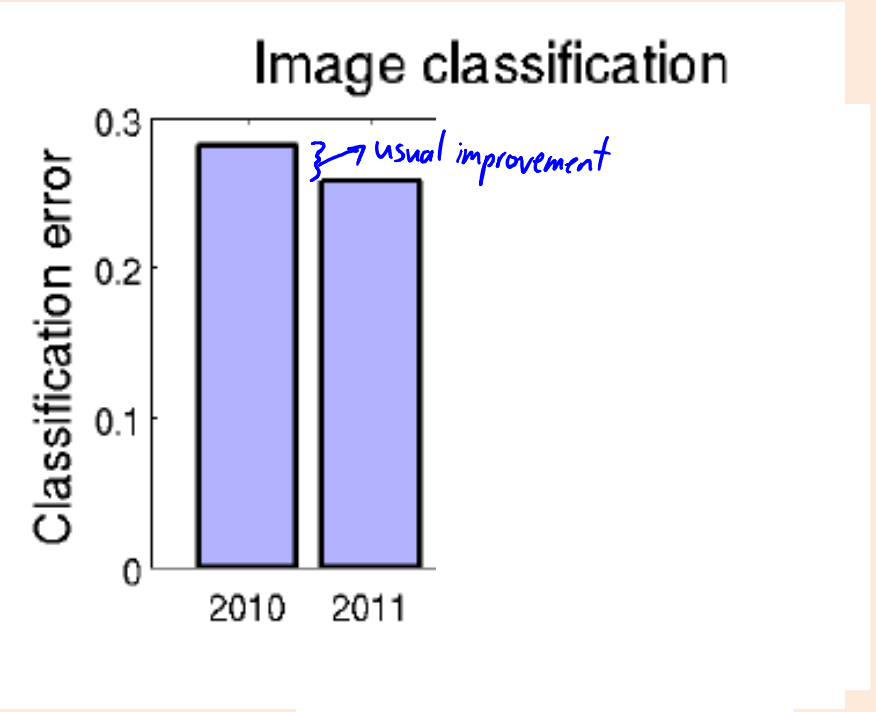


https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

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

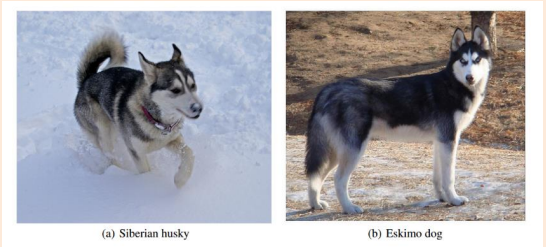


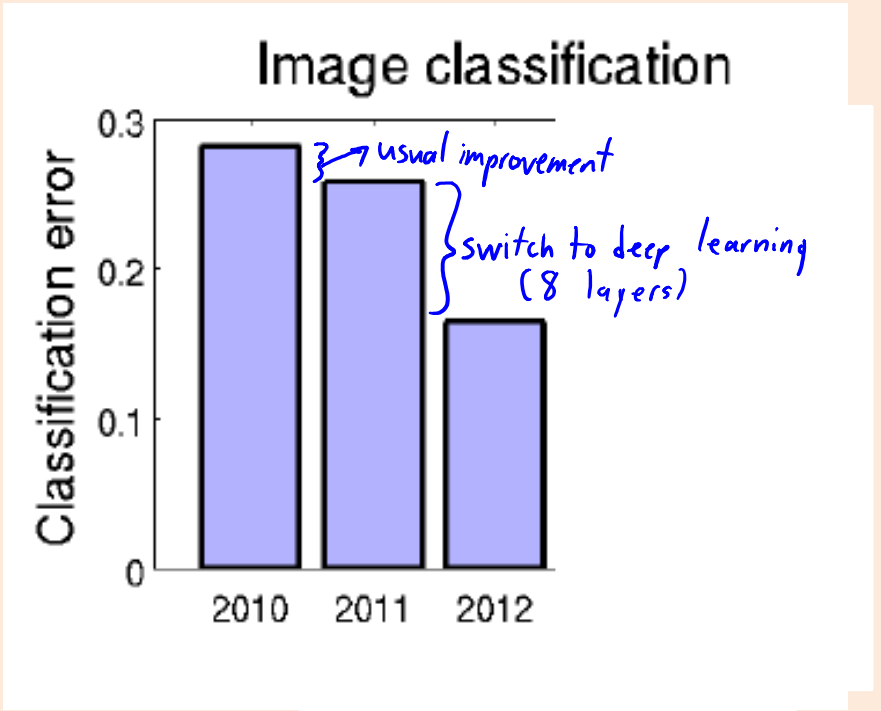


https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

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

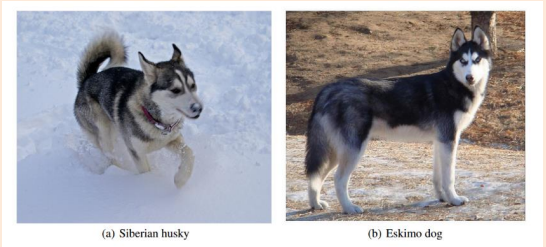


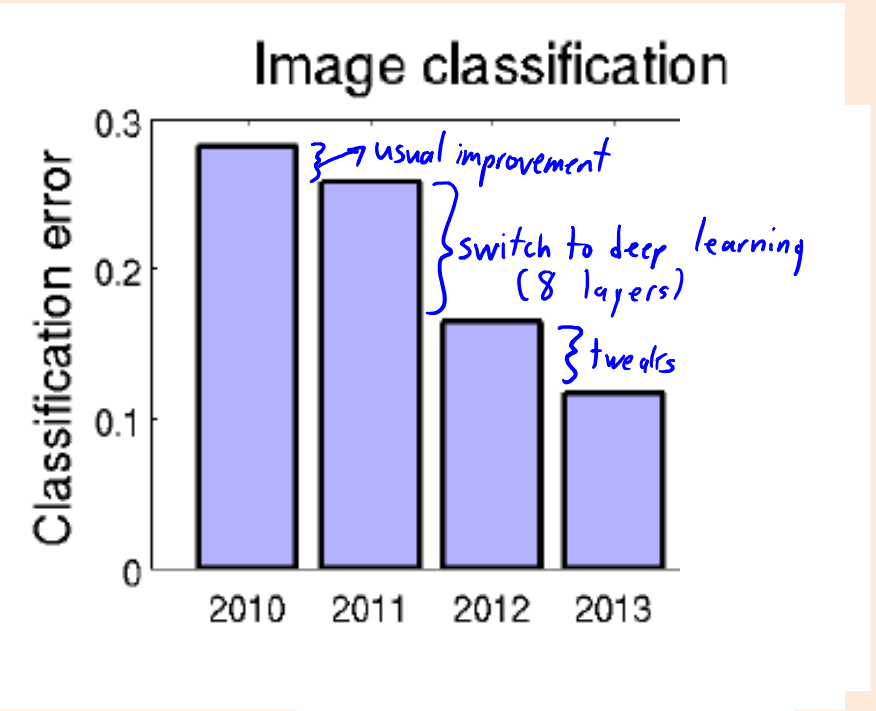


https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

ImageNet Challenge

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

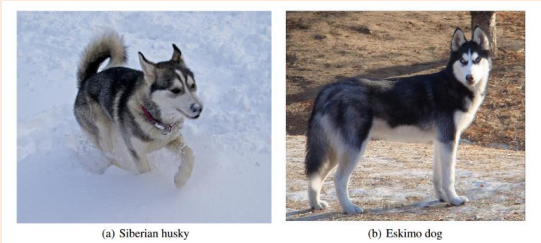


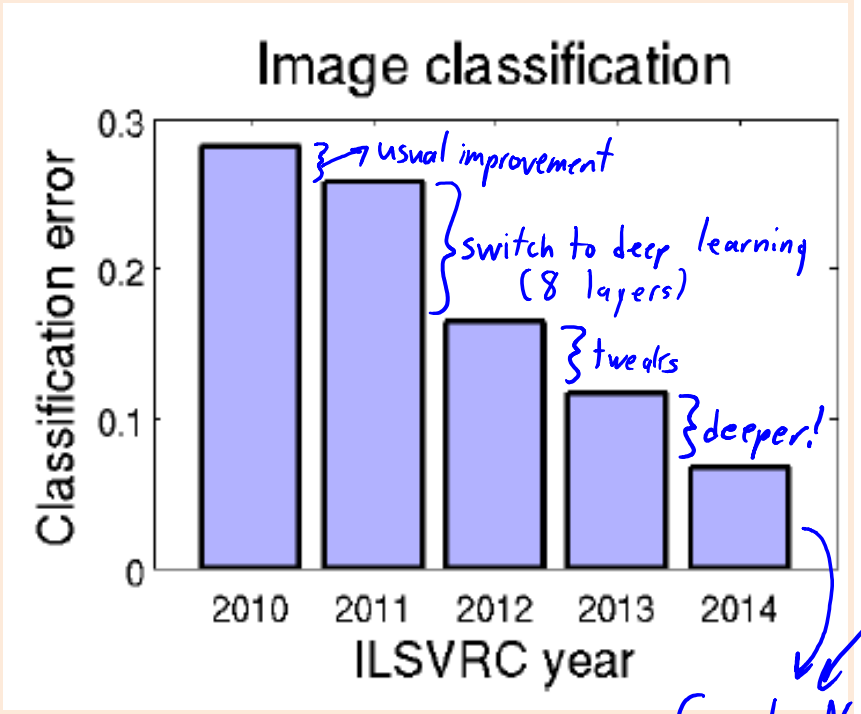


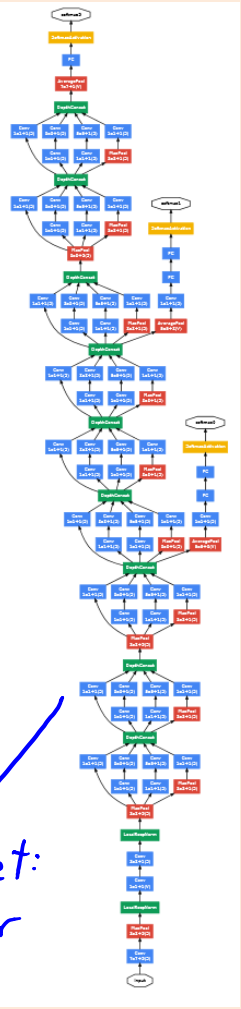
https://ischlag.github.io/2016/04/05/important-ILSVRC-achievements/
http://arxiv.org/pdf/1409.0575v3.pdf
http://arxiv.org/pdf/1409.4842v1.pdf

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.

Thank you!

Raghava Mukkamala

rrm.digi@cbs.dk

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

https://raghavamukkamala.github.io/

https://cbsbda.github.io/