AI504: Programming for Artificial Intelligence

Week 2: Basic Machine Learning

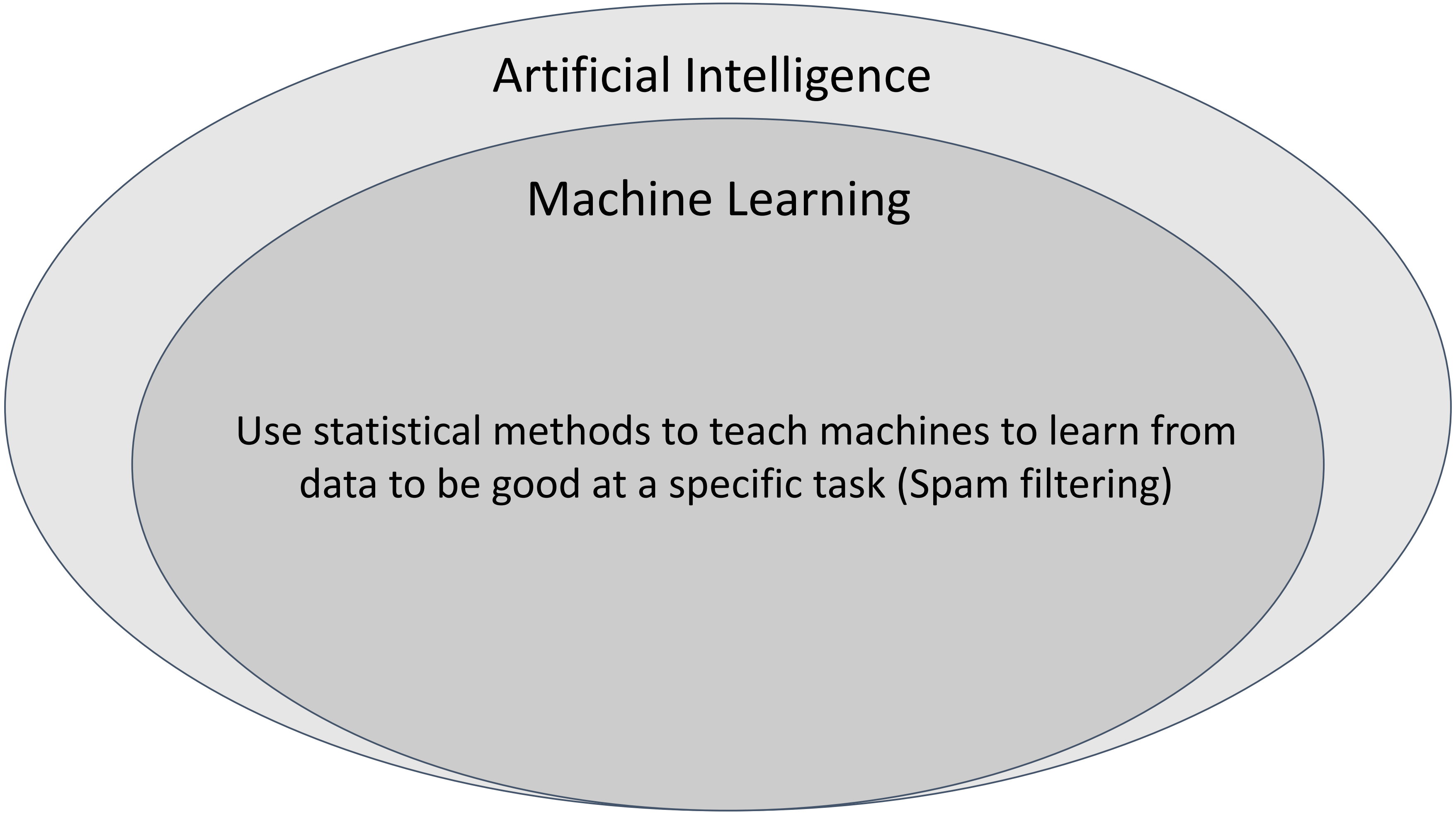
Edward Choi
Grad School of AI
edwardchoi@kaist.ac.kr

What is AI?

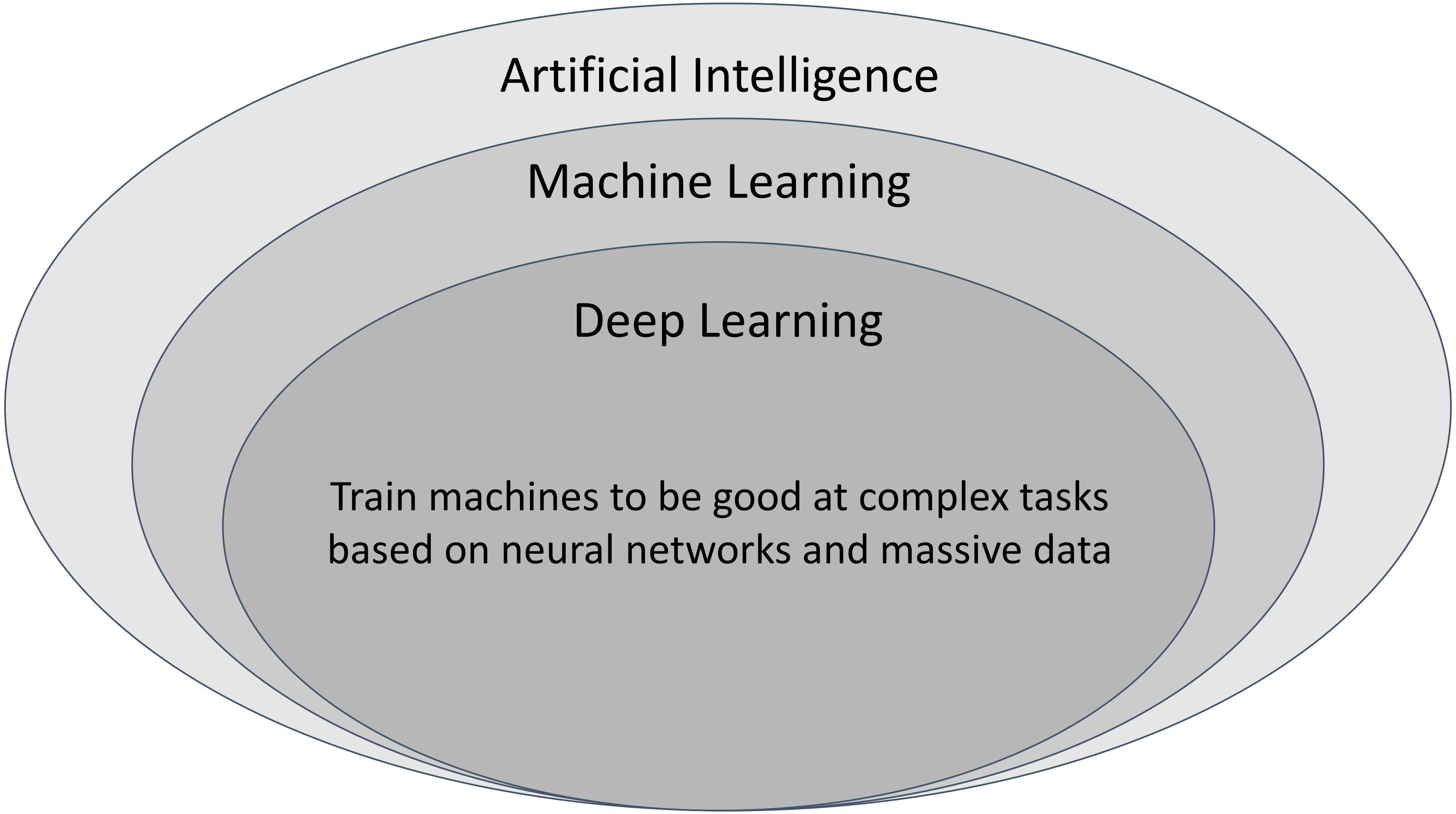
Artificial Intelligence

Make machines/computers mimic human intelligence
Concept as old as the computer (Chess program by Alan Turing)

What is Machine Learning?



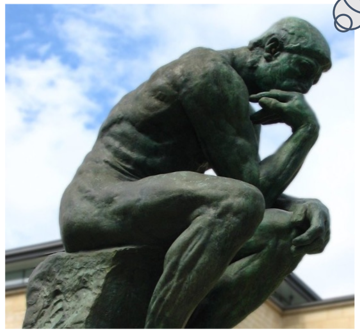
What is Deep Learning?



Why Deep Learning?

Less feature enginnering

Eyes? Color?
Nose? Fur?
Ears? Tongue?



Dog

Cat

Input

Feature Extraction

Classification

Classical machine learning process

Why Deep Learning?

Less feature engineering

Dog

Cat

Input

Feature Extraction + Classification

Deep learning process

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Machine Learning Categories

- Supervised Learning
  - Learn a function that maps an input **x** to an output **y**
  - Examples?
- Unsupervised Learning
- Reinforcement Learning

Machine Learning Categories

- Supervised Learning
  - Learn a function that maps an input **x** to an output **y**
  - Examples
    - Image classification
    - French-English translation
    - Image captioning
- Unsupervised Learning
- Reinforcement Learning

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
  - Learn a distribution/manifold function of data **X** (no label **y**)
  - Examples?
- Reinforcement Learning

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
  - Learn a distribution/manifold function of data **X** (no label **y**)
  - Examples
    - Clustering
    - Low-rank matrix factorization
    - Kernel density estimation
- Reinforcement Learning

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
  - Given an environment **E** and a set of actions **A**, learn a function that maximizes the long-term reward R.
  - Examples?

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
  - Given an environment **E** and a set of actions **A**, learn a function that maximizes the long-term reward R.
  - Examples
    - Go
    - Atari
    - Self-driving car

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Lines are a bit blurry with generative models and self-supervised learning

Machine Learning Categories

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

All practices in this course are either SL or UL

Optimization

- SL, UL, RL → You need to **train your model** (i.e. function)
  - Your machine needs to learn from data
  - That’s why it’s called statistical machine learning!

Optimization

- SL, UL, RL ➔ You need to **train your model** (i.e. function)
  - Your machine needs to learn from data
  - That's why it's called statistical machine learning!
- How to train your model $f(x; \theta)$
  - You need to have a goal
    - Otherwise how are you going to train your model?
  - We call this goal "objective function"

Optimization

- SL, UL, RL ➔ You need to **train your model** (i.e. function)
  - Your machine needs to learn from data
  - That’s why it’s called statistical machine learning!
- How to train your model f(x; θ)
  - You need to have a goal
    - Otherwise how are you going to train your model?
  - We call this goal “objective function”
    - e.g. Dog classification

➔ Objective: Correctly classify Dog/No Dog

Optimization

- SL, UL, RL ➔ You need to **train your model** (i.e. function)
  - Your machine needs to learn from data
  - That's why it's called statistical machine learning!
- How to train your model f(x; θ)
  - You need to have a goal
    - Otherwise how are you going to train your model?
  - We call this goal "objective function"
    - e.g. Dog classification

➔ Objective: Minimize *loss*(y, y')

y:true class, y':predicted class (y' = f(x; θ))

Optimization

- SL, UL, RL ➔ You need to **train your model** (i.e. function)
  - Your machine needs to learn from data
  - That's why it's called statistical machine learning!
- How to train your model $f(x; \theta)$
  - You need to have a goal
    - Otherwise how are you going to train your model?
  - We call this goal "objective function"
    - e.g. Dog classification

Objective: Minimize $-(y*\log(y') + (1-y)*\log(1-y'))$

y:true class, y':predicted class ($y' = f(x; \theta)$)

Optimization

- SL, UL, RL ➔ You need to **train your model** (i.e. function)
  - Your machine needs to learn from data
  - That's why it's called statistical machine learning!
- How to train your model $f(x; \theta)$
  - You need to have a goal
    - Otherwise how are you going to train your model?
  - We call this goal "objective function"
- You need to find θ that minimizes *loss*$(y, f(x; \theta))$
  - That's why it's called optimization!

Optimization

- SL, UL, RL ➔ You need to **train your model** (i.e. function)
  - Your machine needs to learn from data
  - That's why it's called statistical machine learning!
- How to train your model $f(x; \theta)$
  - You need to have a goal
    - Otherwise how are you going to train your model?
  - We call this goal "objective function"
- You need to find θ that minimizes *loss*$(y, f(x; \theta))$
  - There are other loss functions
  - Regression: Mean Squared Error (MSE) or Mean Absolute Error (MAE)

How to Optimize Your Model

- Find x that minimizes $f(x) = x^2 - 2x + 1$
  - $x = 1$ ➔ One global minimum
  - Why one global minimum?

How to Optimize Your Model

- Find x that minimizes $f(x) = x^2 - 2x + 1$
  - $x = 1$ ➔ One global minimum
  - Why one global minimum?
- Find θ that minimizes *loss*(y, f(x; θ))
  - *loss*(y, f(x; θ)) is a complex (not meaning imaginary), non-convex function
  - Many local minima, no analytical solution

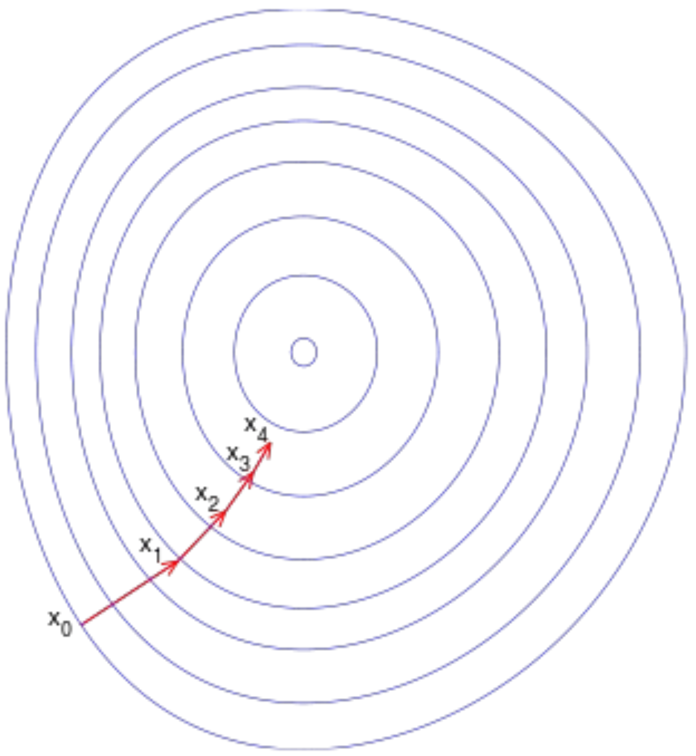
How to Optimize Your Model

- Find x that minimizes $f(x) = x^2 - 2x + 1$
  - $x = 1$ ➔ One global minimum
  - Why one global minimum?
- Find θ that minimizes *loss*(y, f(x; θ))
  - *loss*(y, f(x; θ)) is a complex (not meaning imaginary), non-convex function
  - Many local minima, no analytical solution
- Numerical method
  - Iteratively find a better θ until you are satisfied.

Gradient Descent

- Updating your parameters θ based on your training data X, Y

$$\theta_{k+1} = \theta_k - \eta \nabla_\theta \mathcal{L}(Y, f(X; \theta_k))$$



Stochastic Gradient Descent

- Updating your parameters based on a subset of the training data

$$\theta_{k+1} = \theta_k - \eta \nabla_\theta \mathcal{L}(\boxed{Y}, f(\boxed{X}; \theta_k))$$

Not your entire data.

A subset (i.e. minibatch) of the entire data.

- Why use SGD?
  - Your training data is too big
    - Cannot fit in your memory, takes too long to calculate the gradients
  - Sometimes smaller batch-size helps avoid suboptimal minimum
  - If your minibatches are I.I.D, SGD ➔ GD
- Modern deep learning is founded on SGD

Evaluation

- How do you know when to stop SGD?
  - You can’t optimize your model forever
- Using the value of your loss function
  - It’s not very intuitive
  - Hard to tell when to stop? (1e-3? 1e-4?)

Evaluation

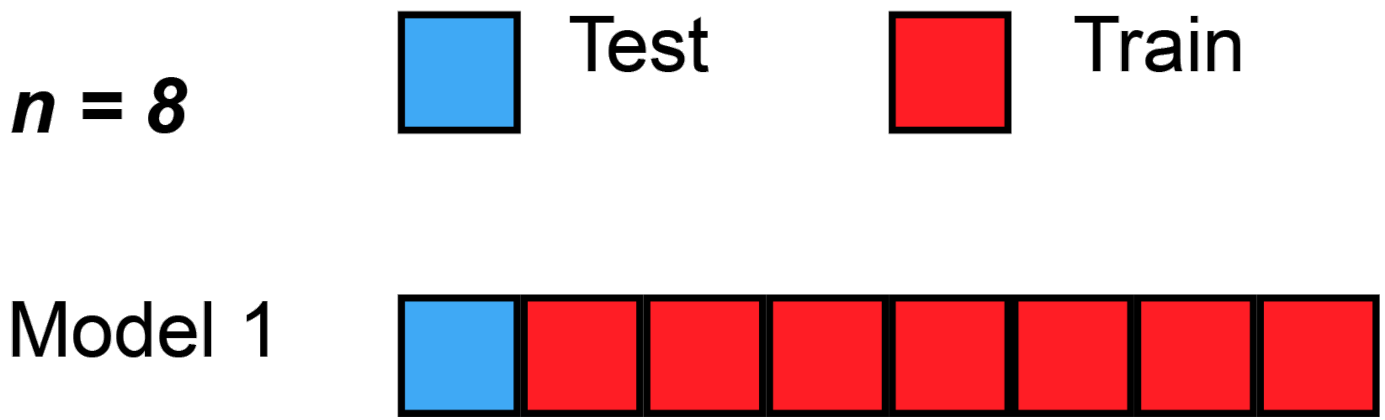
- Popular Evaluation Metrics
  - Accuracy
    - Used for multi-class classification
  - Area under the ROC (AUROC)
    - Used for binary classification
  - Precision & Recall
    - Used for information retrieval
  - BLEU Score
    - Used for machine translation
  - Perplexity
    - Used for language modeling
  - FID Score
    - Use for image generation

Train & Validation & Test

- Split the entire dataset into three sets
- Training set
  - Use this set to train your model
- Validation set (i.e. development set)
  - Use this set to evaluate your model, and decide when to stop training
- Test set
  - Use this “unseen” set to evaluate the final model
  - Don’t use this set during the training phase!!

N-fold Cross Validation

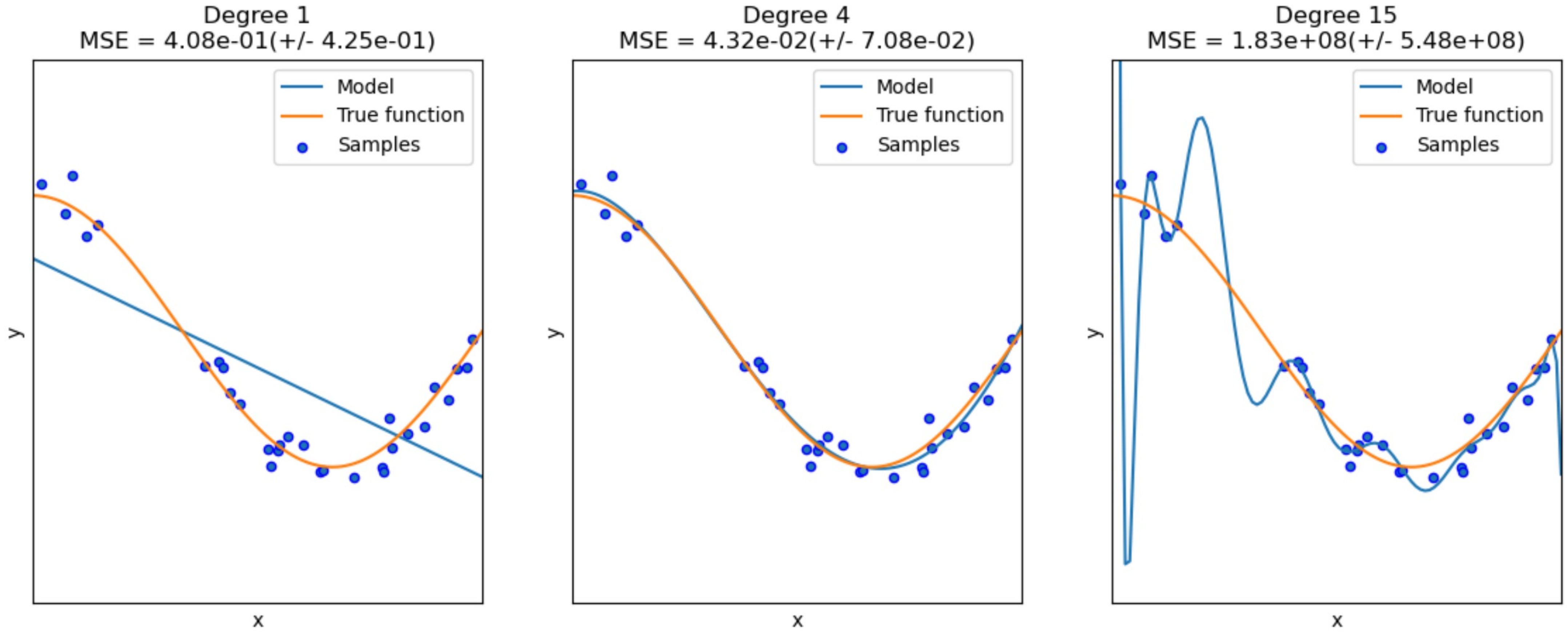
- Test your model's performance in diverse train/validation/test splits
  - Maybe you got lucky with an easy split, so test N times



- Not often used in modern deep learning
  - Dataset is too large ➔ Time-consuming & statistically unnecessary

Overfitting & Underfitting

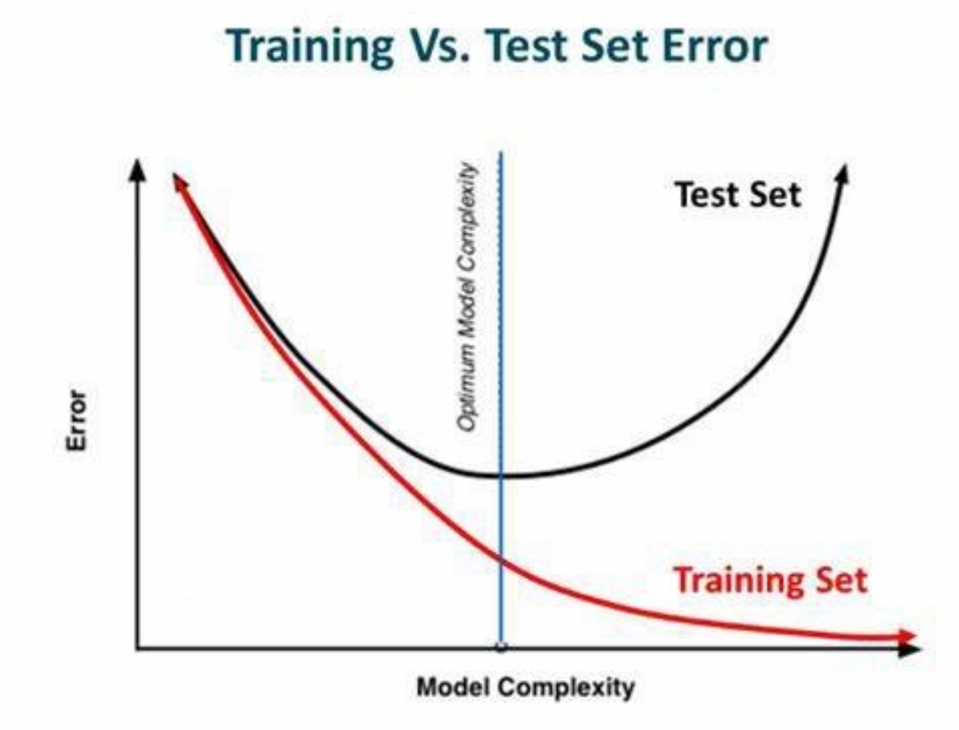
- Data complexity VS model capacity



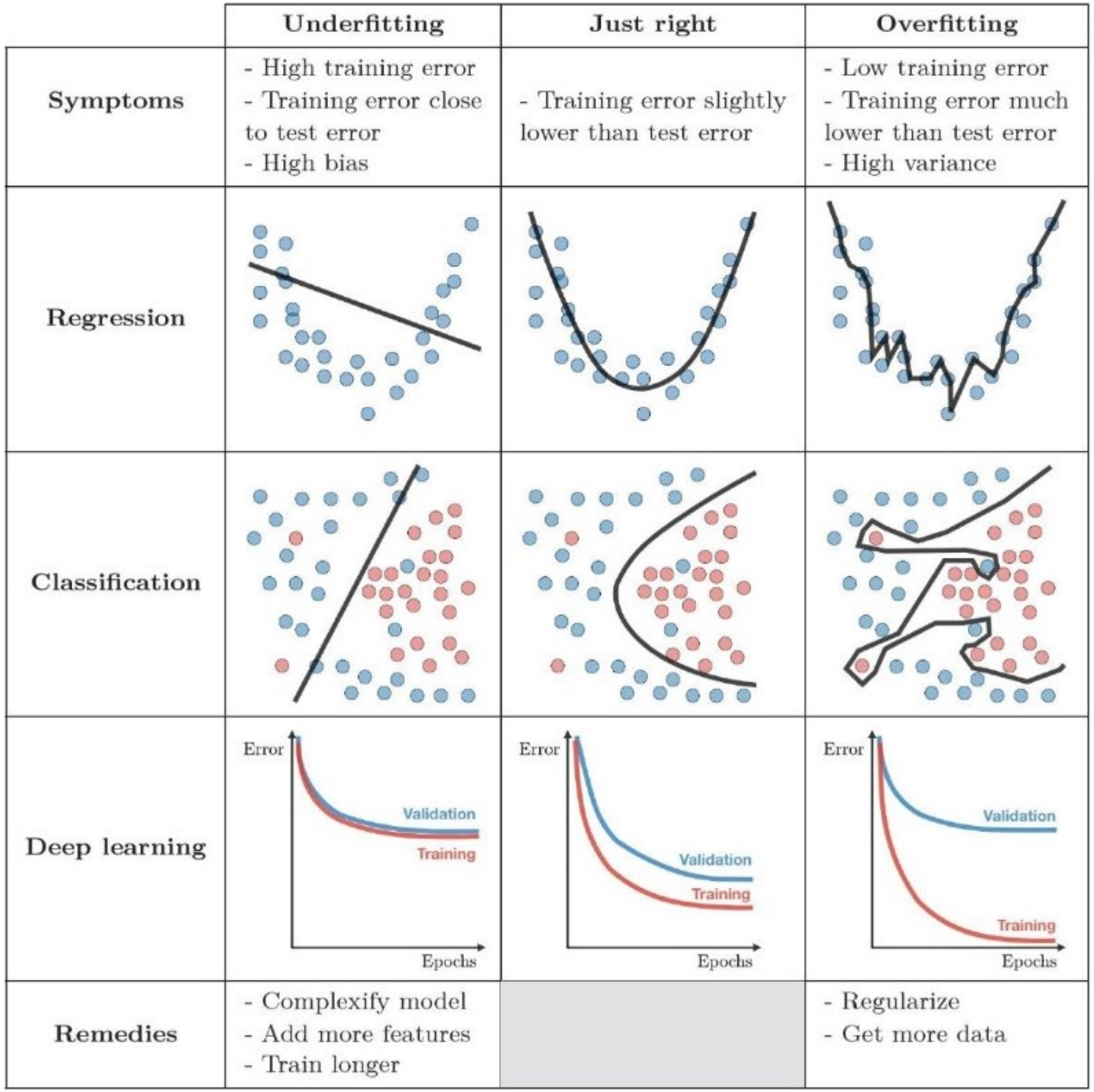
https://datascience.foundation/sciencewhitepaper/underfitting-and-overfitting-in-machine-learning

Overfitting to Training Data

- Doing too well on training data doesn't always lead to good test performance
  - Does not "generalize" well to unseen data



Overfitting & Underfitting

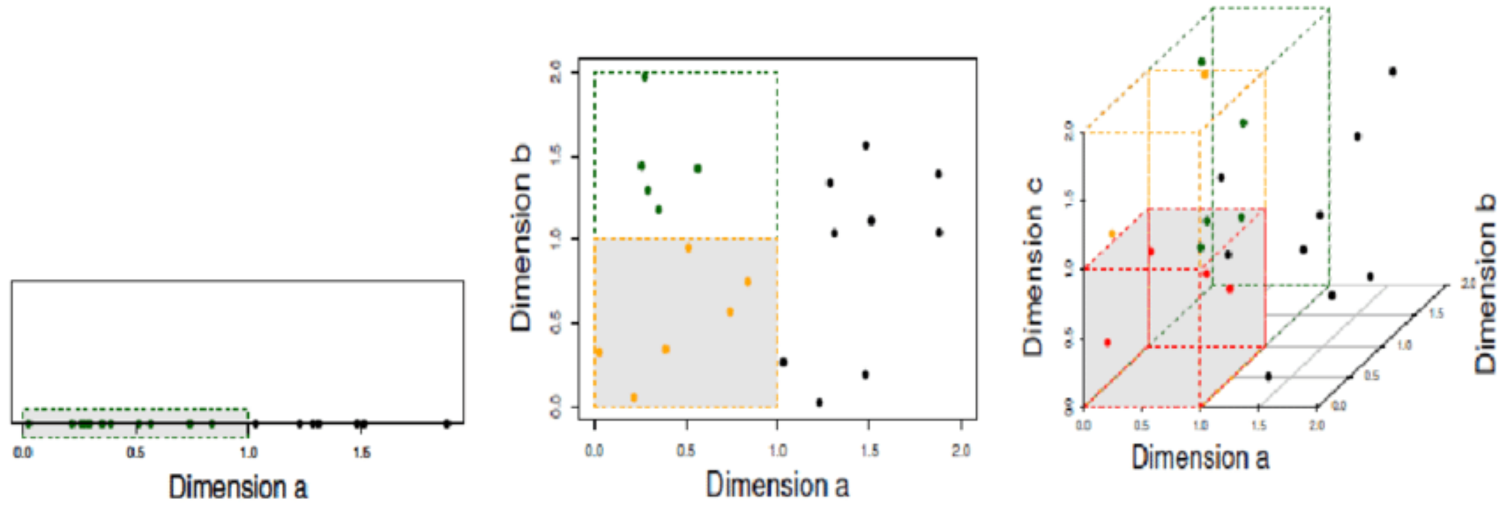


	Underfitting	Just right	Overfitting
Symptoms	- High training error - Training error close to test error - High bias	- Training error slightly lower than test error	- Low training error - Training error much lower than test error - High variance
Regression			
Classification			
Deep learning			
Remedies	- Complexify model - Add more features - Train longer		- Regularize - Get more data

https://www.kaggle.com/getting-started/183376

Curse of Dimensionality

- When underfitting
  - Add more features
- linear ↑ in # feature ➔ exponential ↑ in # data to fill the space

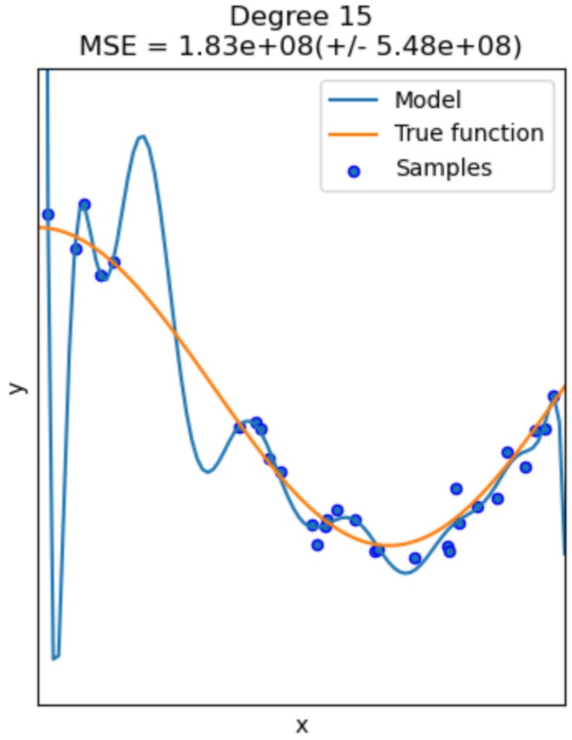


- Still plays some role in modern deep learning

Rajput, D.S., Singh, P.K. and Bhattacharya, M., 2012. IQRAM: a high dimensional data clustering technique. *International Journal of Knowledge Engineering and Data Mining*, 2(2-3), pp.117-136.

Regularization

- Restrict the freedom of your model
  - Downsizing the hypothesis space



$$w_{15}x^{15} + w_{14}x^{14} + \ldots + w_1x^1 + b = \mathbf{w}^T\mathbf{x} + b$$

$w_{15} = 191.14$

$w_{14} = -89.32$

$w_{13} = 239.81$

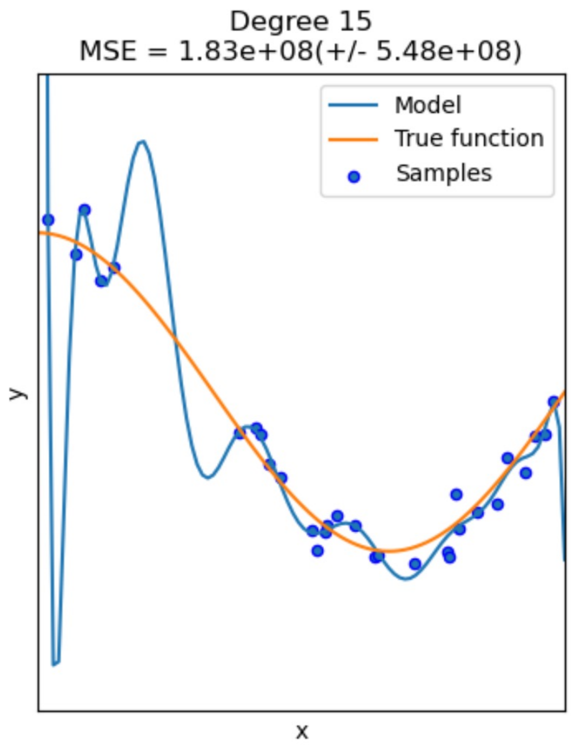
…

Objective function with ***L_2* regularization**:

$$\mathcal{L}(y, y') + \frac{\lambda}{2}\|\boldsymbol{w}\|^2$$

Regularization

- Restrict the freedom of your model
  - Downsizing the hypothesis space



$$w_{15}x^{15} + w_{14}x^{14} + \ldots + w_1x^1 + b = \mathbf{w}^T\mathbf{x} + b$$

$w_{15} = 191.14$

$w_{14} = -89.32$

$w_{13} = 239.81$

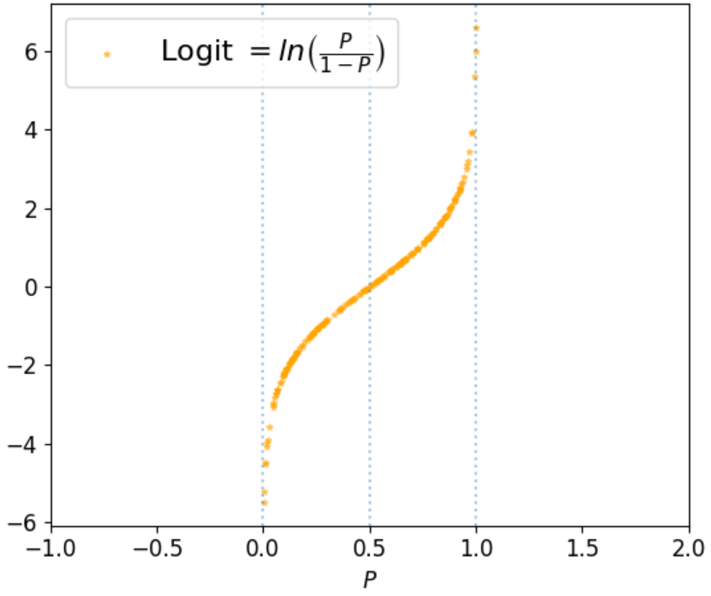
…

Objective function with $\boldsymbol{L_1}$ **regularization**:

$$\mathcal{L}(y, y') + \lambda(|w_{15}| + |w_{14}| + \cdots)$$

Popular Classifiers

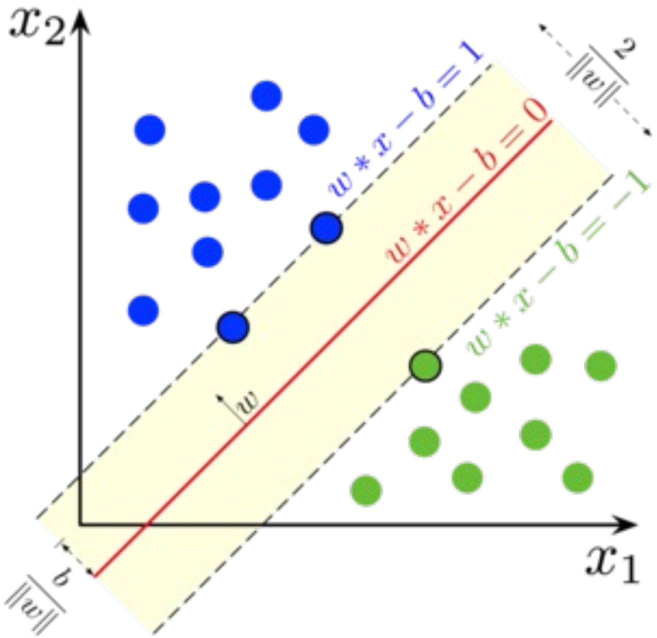
- Logistic Regression
  - Probability ➔ Odds ➔ Log of odds (Logit)
  - Assume the logit can be modeled linearly
  - Trained via gradient descent



https://medium.com/ro-data-team-blog/interpret-the-impact-size-with-logistic-regression-coefficients-5eec21baaac8

Popular Classifiers

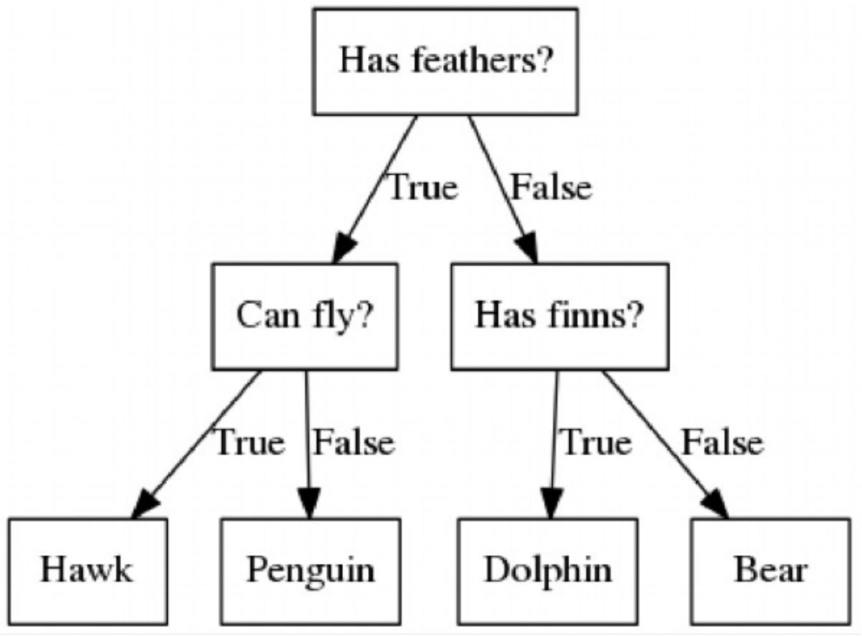
- Support Vector Machine
  - Maximize the margin between two classes
  - Trained via constrained optimization (Lagrange Multiplier)
    - Or via gradient descent with hinge loss



https://en.wikipedia.org/wiki/Support-vector_machine

Popular Classifers

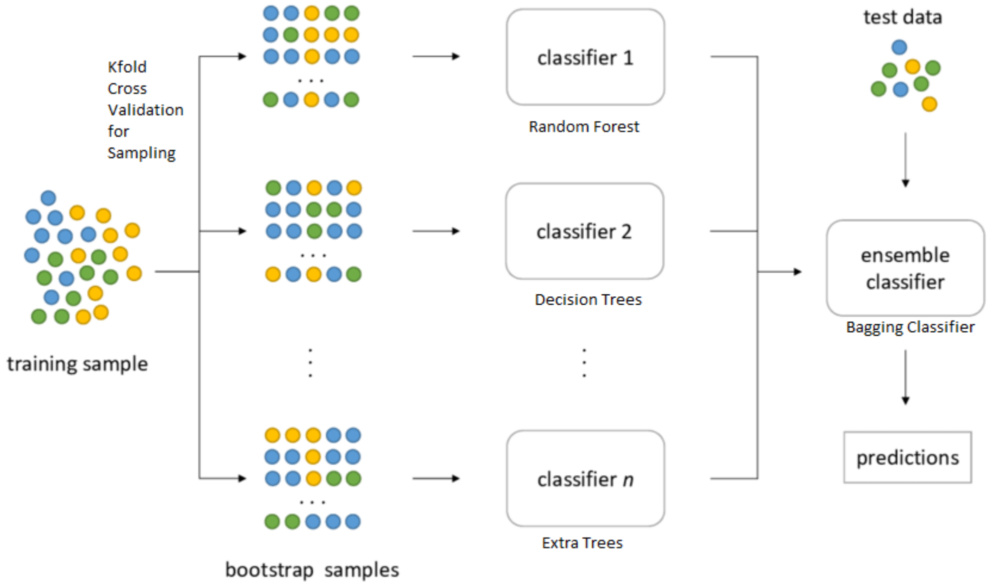
- Decision Tree
  - Build a tree based on features
  - Trained via the CART algorithm



https://medium.com/towards-artificial-intelligence/decision-trees-explained-with-a-practical-example-fe47872d3b53

Ensembles

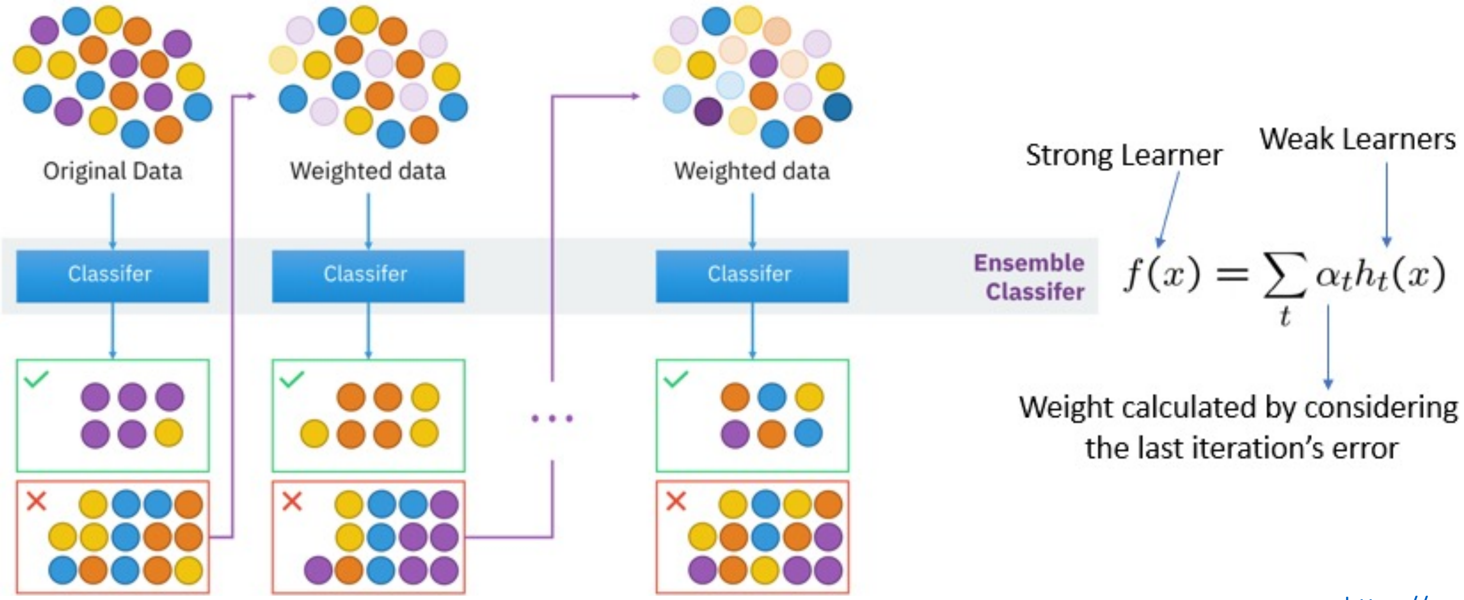
- Use multiple classifiers (or regressors) to improve performance
- Bagging
  - Train multiple classifiers on different subsets of data
  - Train multiple classifiers on different subsets of features



https://medium.com/ml-research-lab/bagging-ensemble-meta-algorithm-for-reducing-variance-c98fffa5489f

Ensembles

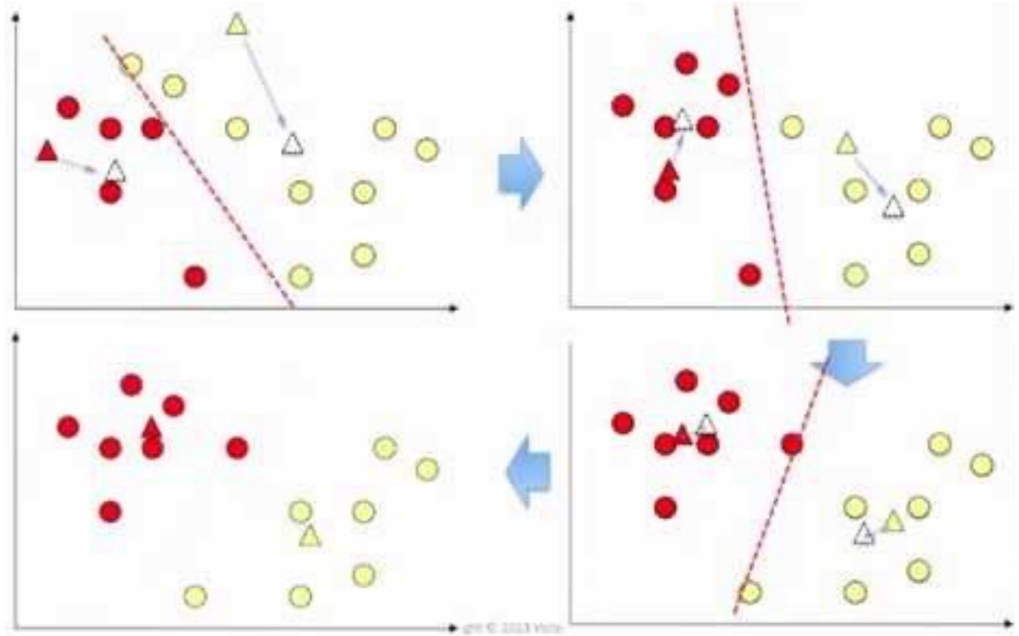
- Use multiple classifiers (or regressors) to improve performance
- Bagging
- Boosting
  - (k+1)-th classifier tries to correct the k-th classifiers errors.



https://en.wikipedia.org/wiki/Boosting_(machine_learning)

Popular Clustering

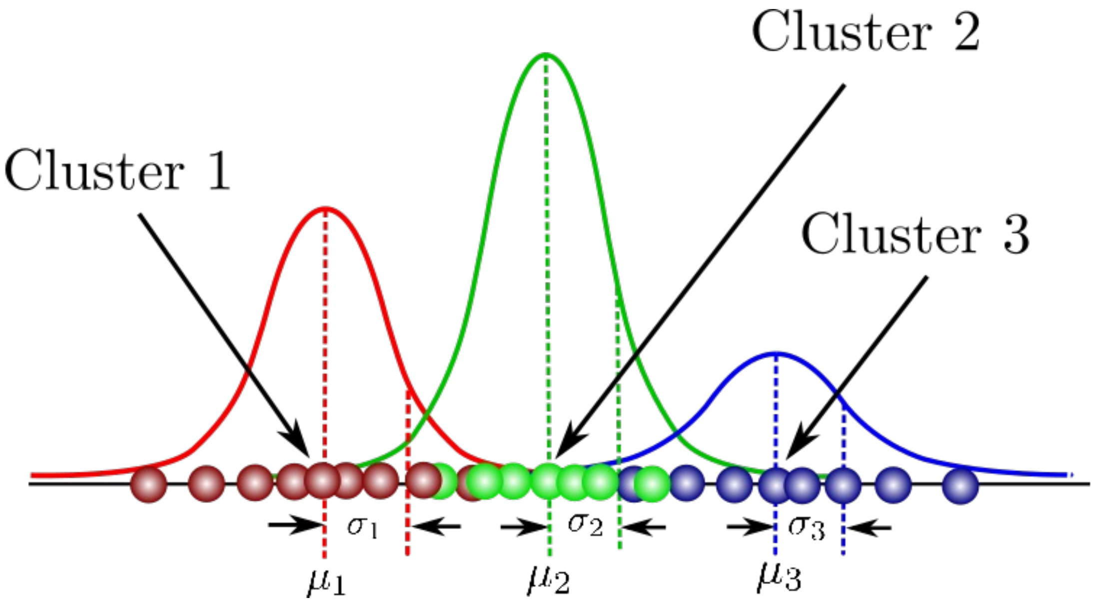
- K-means
  - Update membership of each sample using the closest centroid.
  - Update the centroid value using all the member samples.
  - Repeat the above steps



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

Popular Clustering

- Mixture of Gaussian
  - Generalization of K-means clustering
  - A sample has a probabilistic membership to each cluster
  - Trained via Expectation-Maximization (EM)



https://towardsdatascience.com/gaussian-mixture-models-explained-6986aaf5a95

AI504: Programming for Artificial Intelligence

Week 2: Basic Machine Learning

Edward Choi
Grad School of AI
edwardchoi@kaist.ac.kr