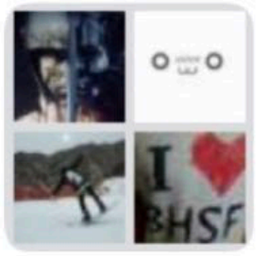
Introduction to Data Science

Lecturer: 朱占星

Teaching Assistants: 郭涵韬、朱垣金、张君钊

School of Mathematical Sciences &
Center for Data Science
Peking University



2019秋季数据科学导引

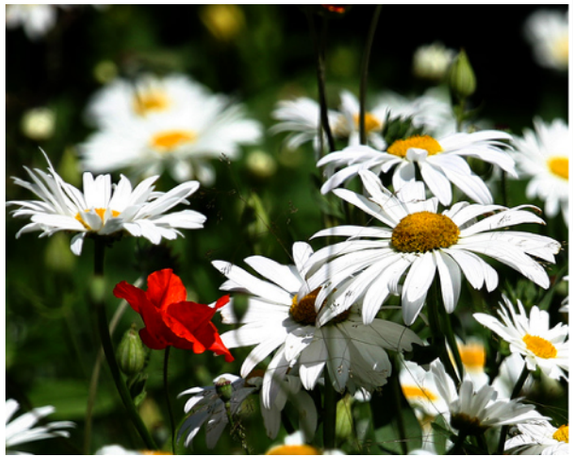
该二维码7天内(9月16日前)有效，重新进入将更新

General overview

- A preliminary course for **learning from data**, **artificial intelligence** and **other related application areas**
- Requirements
  - Calculus, linear algebra, basic statistics, numerical optimization, signal processing
  - Programming and algorithms, e.g. Python

An Classic Example: image classification/recognition

sunflowers



Daisy

tulip

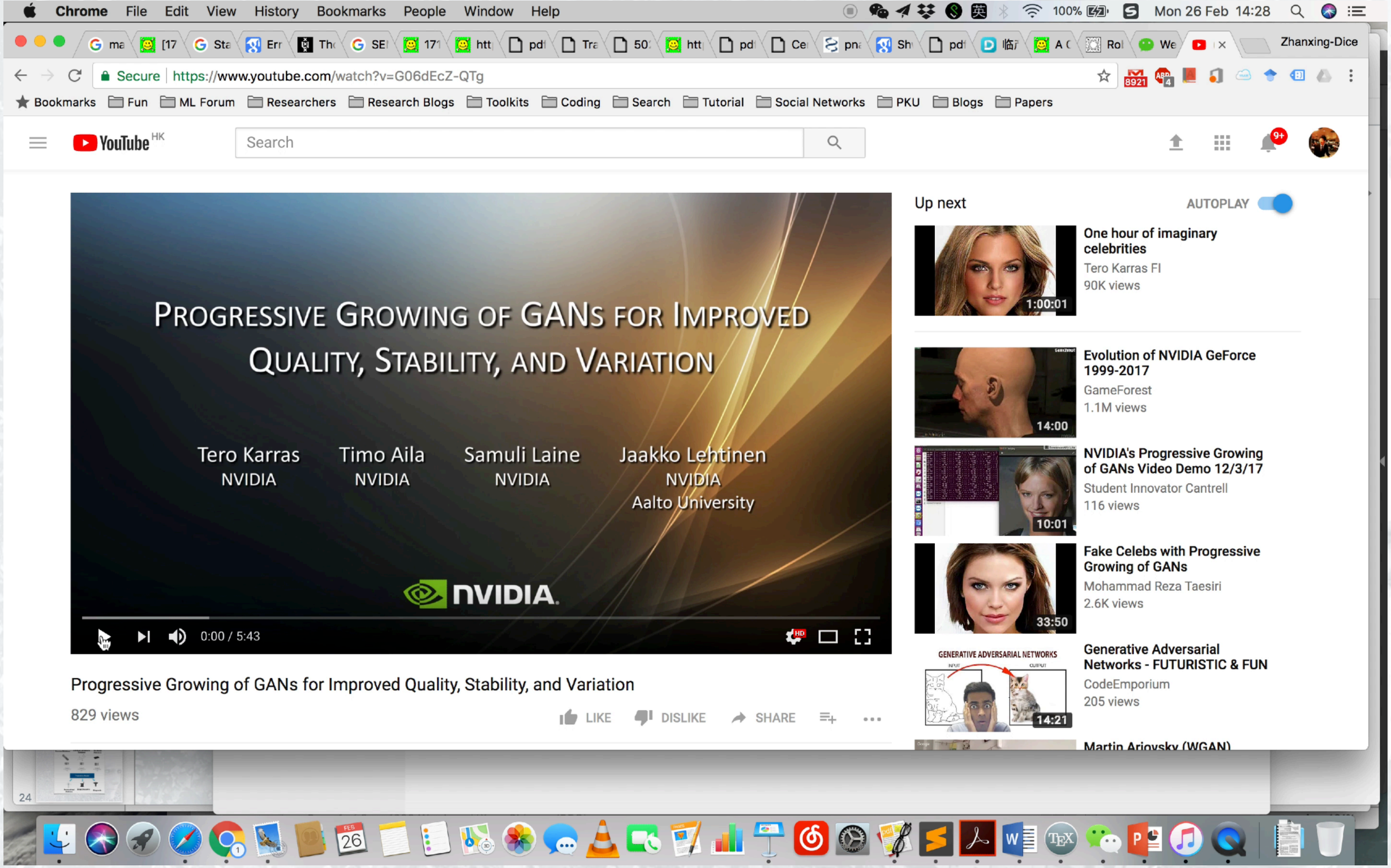
rose

?

?

Can you generate faces?

Karras T, Aila T, Laine S, Lehtinen J. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196. 2017 Oct 27.



Progressive growing of GANs from NVIDIA for generative high-quality images

submitted to ICLR 2018

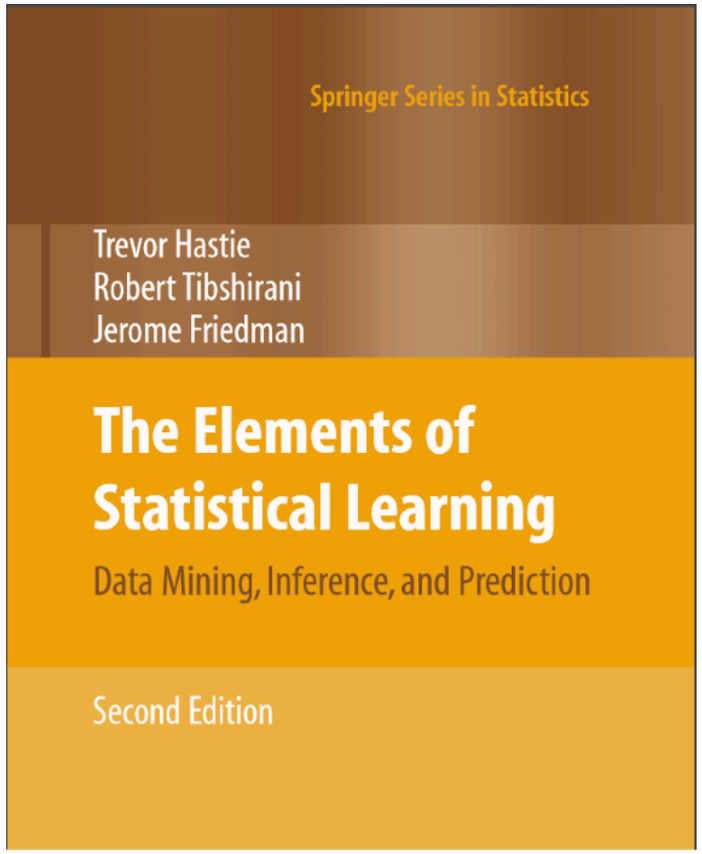
https://arxiv.org/pdf/1710.10196.pdf

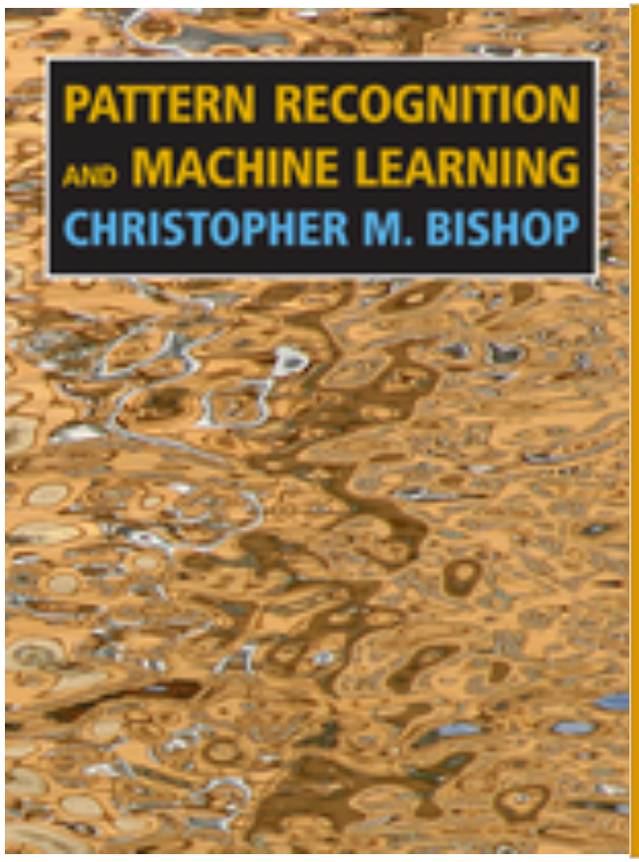
https://www.youtube.com/watch?v=G06dEcZ-QTg

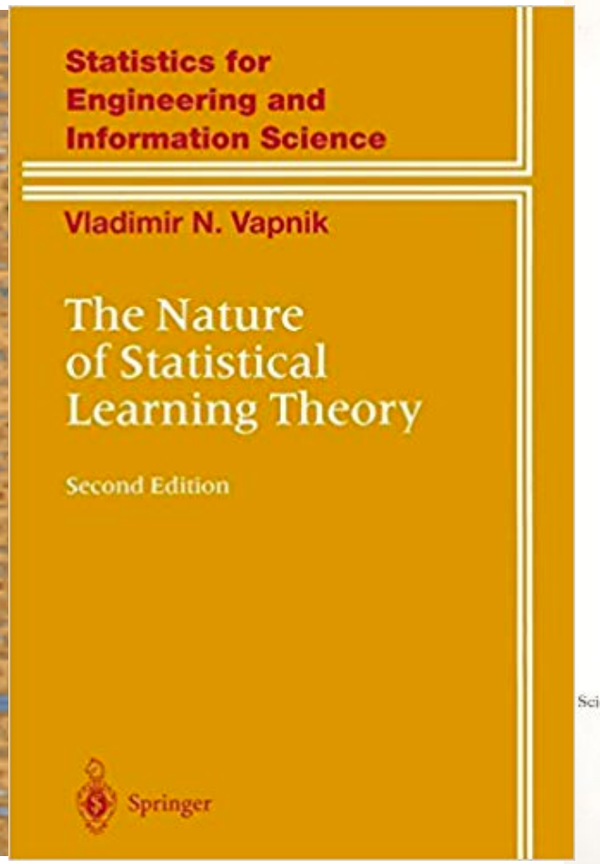
Goals of this course

- Understanding data science comprehensively from various aspects:
  - target tasks, models, algorithms, implementations and existing issues
- Good sense of data-driven perspective for solving problems
- Mathematically analyzing models and algorithms
- Implementation of solving tasks
  - Familiar with entire pipeline of data analysis
  - Debug, analysis, and improve
  - Capable of finishing some competitions, e.g. Kaggle

- Reference books
  - Elements of statistical learning
  - Pattern recognition and machine learning
  - Vapnik. The nature of statistical learning theory
  - 欧高炎等。《数据科学导引》







Arrangements and Evaluation

- Teaching assistants: 郭涵韬、朱垣金、张君钊
- Mid-term exam (Nov 25): writing exam (40% of final grade)
- Exercises
- Final projects: (60% of final grade)

kaggle.com

  - Kaggle-in-Class competition, select 1 out of 3
  - including submission to the Kaggle platform and report writing
  - Deadline: Jan 19, 2020 (strict)
  - At most 2 students as a team

Final Projects

- **Project 1: Traffic speed prediction**
  - Website: TBD
  - Consider the traffic speed prediction problem for 228 sensor stations in a region.
  - We split one day into 288 time periods (5 minutes per time period), and count an average speed in each time period for every sensor station.
  - The problem: given the speed of the previous hour (12 time periods) of these 228 stations
  - Note that we also give the distance relationship of these 288 sensor stations, you may be able to consider the interaction of their traffic conditions through these distance information.

Structured multivariate time series prediction

- **Project 2: Flower classification**

  - Website: TBD

  - Image classification is a classic task in the computer vision domain.
  - In this flower classification task, you need to train your model based on a total of 3,899 images of 5 species of flowers, and give the type of 424 flowers in the test set to get as much accuracy as possible.

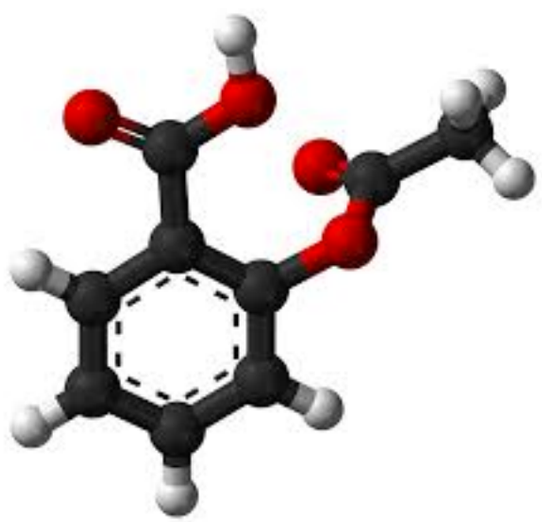
Daisy

tulip

rose

- **Project 3: Inter-atomic potential energy surface (PES) estimation**

  - Website: TBD
  - The goal of the challenge is to model the inter-atomic potential energy surface (PES) for seven small molecules: aspirin, ethanol, malonaldehyde, naphthalene, sallcylic acid, toluene, and uracil.
  - Denote r = {r1, r2, ..., rN} the positions of atoms in a molecule and denote by E(r) the PES. One should define a unified model for all the molecules.
  - This amounts to a high-dimensional regression problem.



What data science is…

- A whole pipeline related to data analysis (particularly in Big Data era)
  - Data collection
    - data types: images, videos, speech, texts, webs, tables…
    - sources: Internet, industry, various sensing devices
  - Data storage and management / computation infrastructure: databases, distributed systems and cloud
  - **Learning from data in a principled way (machine learning)**

Core of Data Science: Machine Learning

- Models learn from data
- To make predictions or learn patterns
- Models are ``trained'' by data and algorithms to obtain ability to perform the **specific** tasks

Output = Model (input; parameters)

The perspective of learning from data provides a new way to understand the world

Learning from data by scientific principles
(Statistical models and algorithms, theoretical analysis based on various math and physical knowledge)

Studying scientific/application problems by data-driven methods
(Applications in various domains)

A nice example: Kepler's Law

The Third Law of Kepler: The square of the orbital period of a planet is directly proportional to the cube of the semi-major axis of its orbit.

第古：观测与数据收集

开普勒：分析数据产生价值

行星	周期（年）	平均距离	周期^2/距离^3
水星	0.241	0.39	0.98
金星	0.615	0.72	1.01
地球	1.00	1.00	1.00
火星	1.88	1.52	1.01
木星	11.8	5.20	0.99
土星	29.5	9.54	1.00
天王星	84.0	19.18	1.00
海王星	165	30.06	1.00

表 1: 太阳系八大行星绕太阳运动的数据

Kepler discovered the law by analysing data.

Newton focused the fundamental principles of the law.

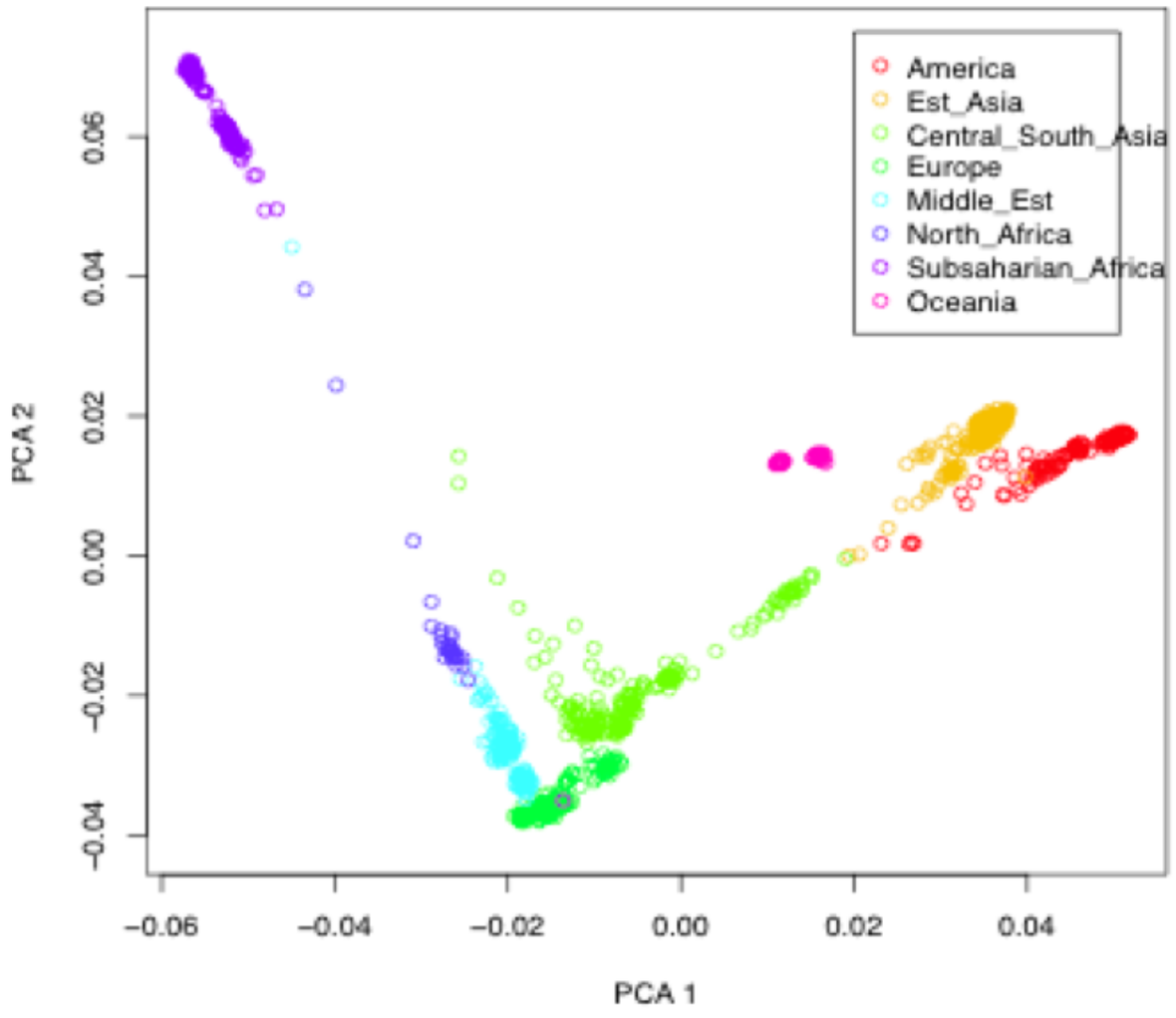
- Newton-type
  - Based on the physical laws
  - Still dominating in the research of science and engineering domains: physics, chemistry, biology etc.
  - But, it’s too complicated in most time, and less effective.
- Kepler-type
  - Data-driven, learning from data
  - More effective in complicated cases, though not from fundamental principles

- One complicated case:
  SNP (Single Nucleotide Polymorphism) in human genome research

- 1024 subjects

- PCA (principal component analysis) to see the human evolution

	SNP_1	SNP_2	$\cdots$	SNP_m
志愿者 1	0	1	$\cdots$	0
志愿者 2	0	2	$\cdots$	1
志愿者 3				
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
志愿者 n	1	9	$\cdots$	1

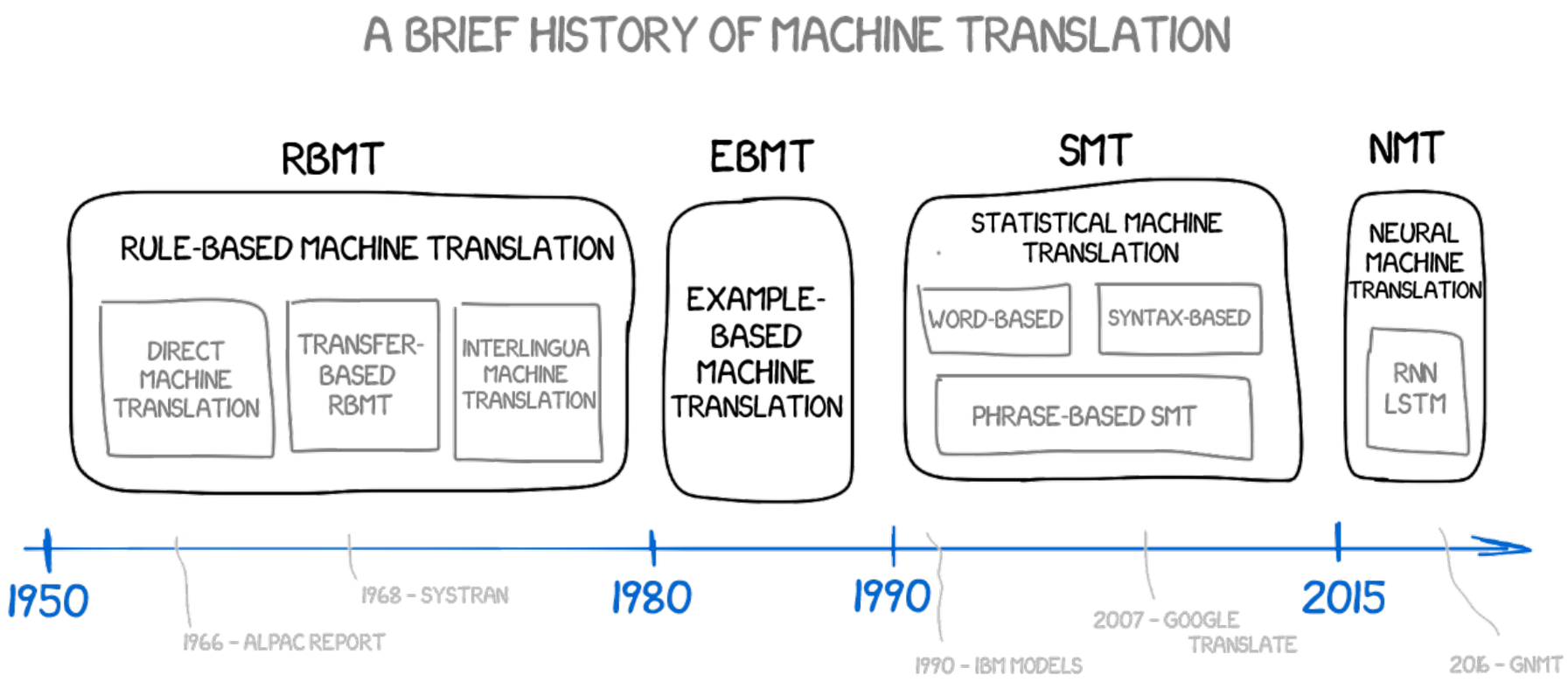


Data science / machine learning impacts other fields dramatically…

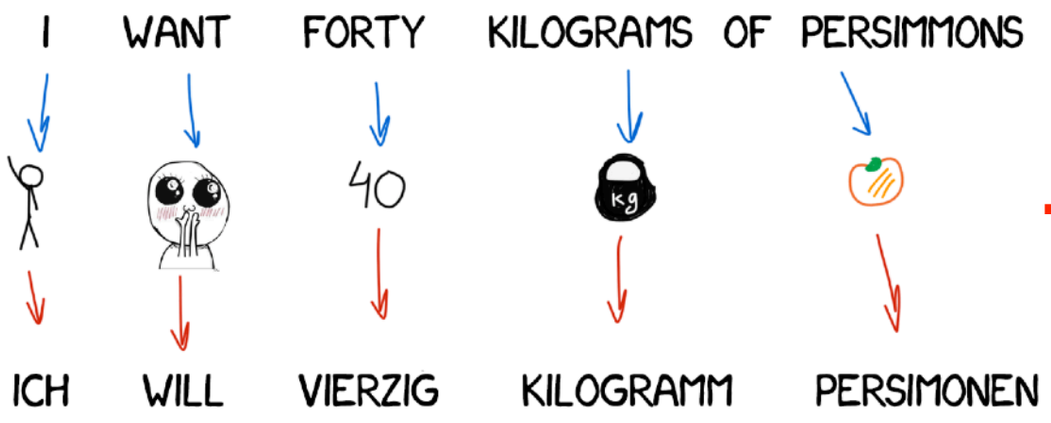
- Fuels traditional scientific domains
  - Linguistics, chemistry, physics…
- Boosts new application areas
  - General applications where data is easy to obtain
  - Solving more challenging AI tasks

Several Examples

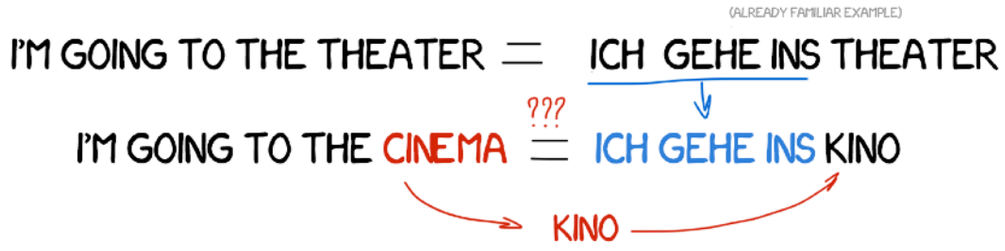
- **Linguistics: machine translation**
  - Old fashion: translate based on linguistic rules across languages, extremely complicated rules…
  - Recent: statistical machine learning (graphical models) relying on learning from data
  - More recent: deep learning (neural networks) relying on learning from data



https://medium.freecodecamp.org/a-history-of-machine-translation-from-the-cold-war-to-deep-learning-f1d335ce8b5

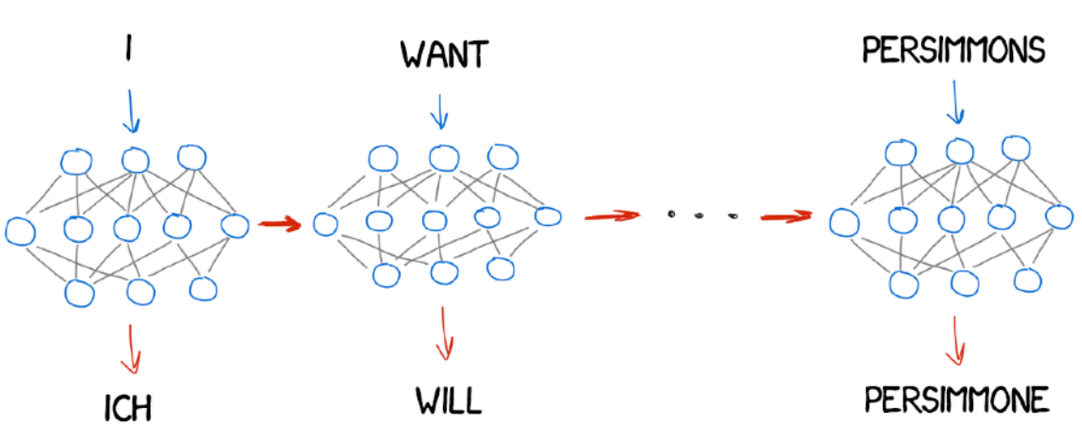


rule-based MT



example-based MT

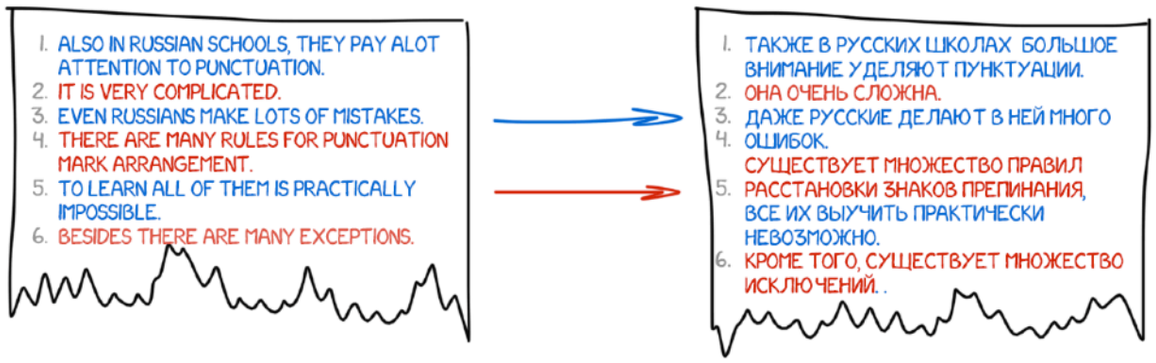
Learning a mapping f from X to Y

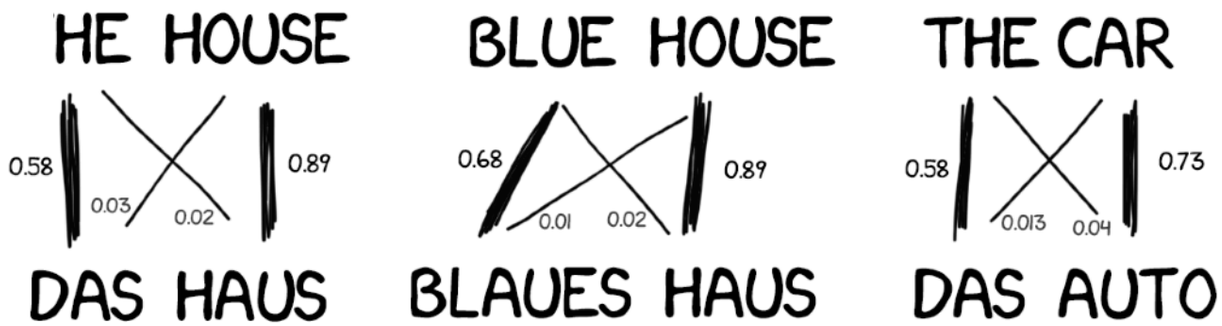


Neural MT

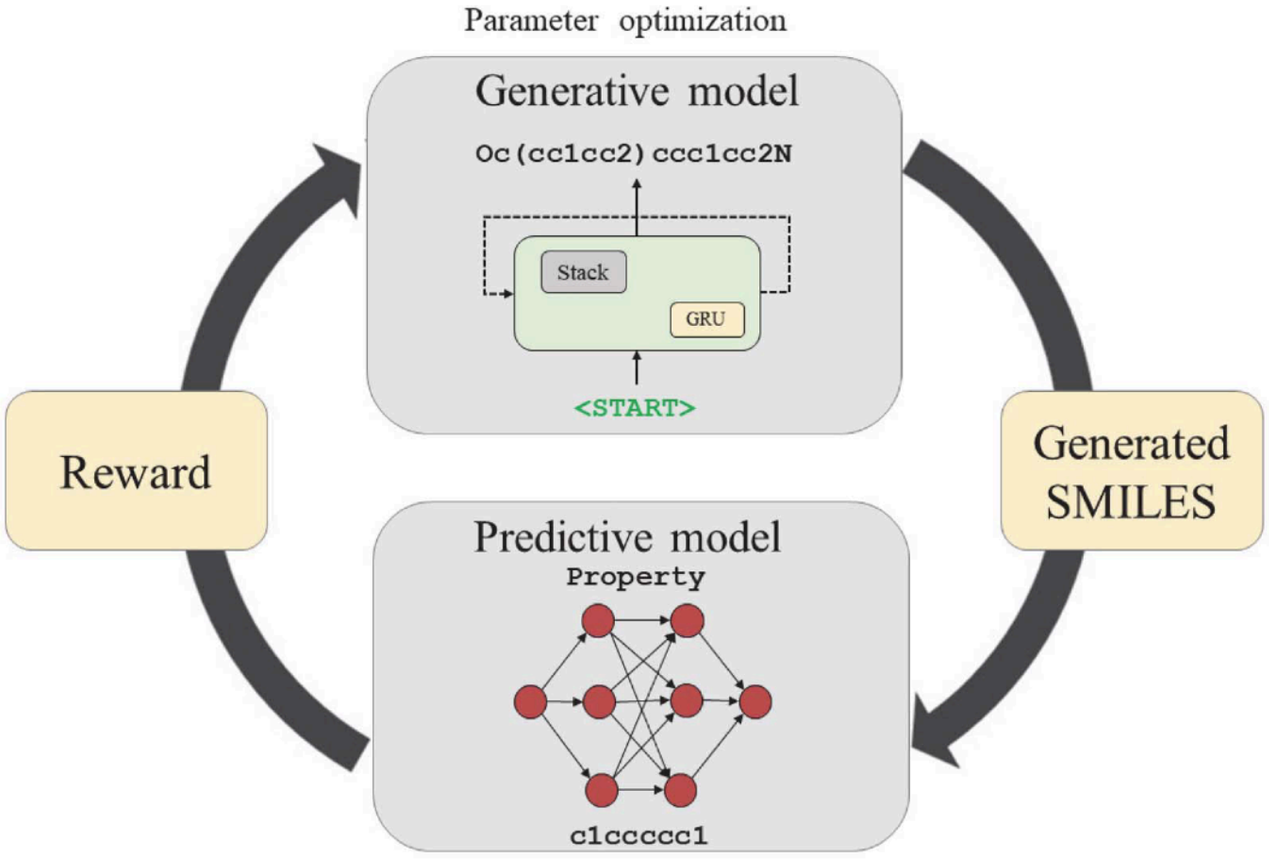
Statistical MT

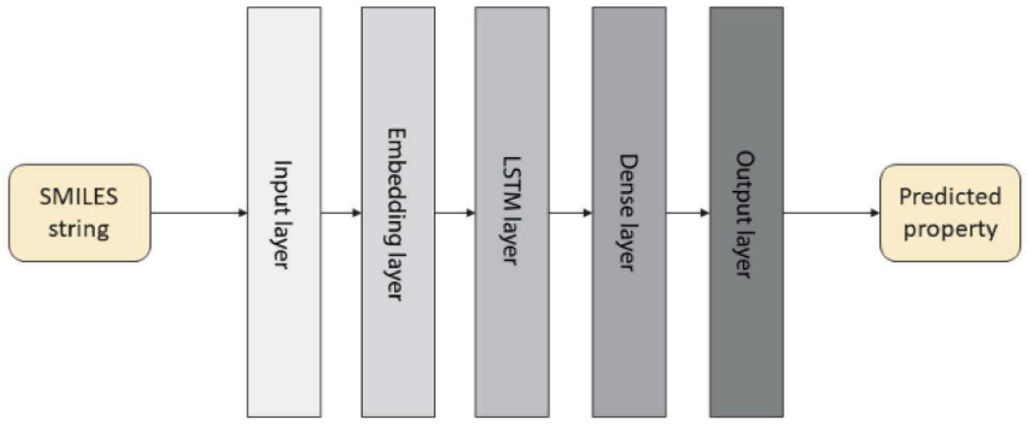
PARALLEL CORPUS



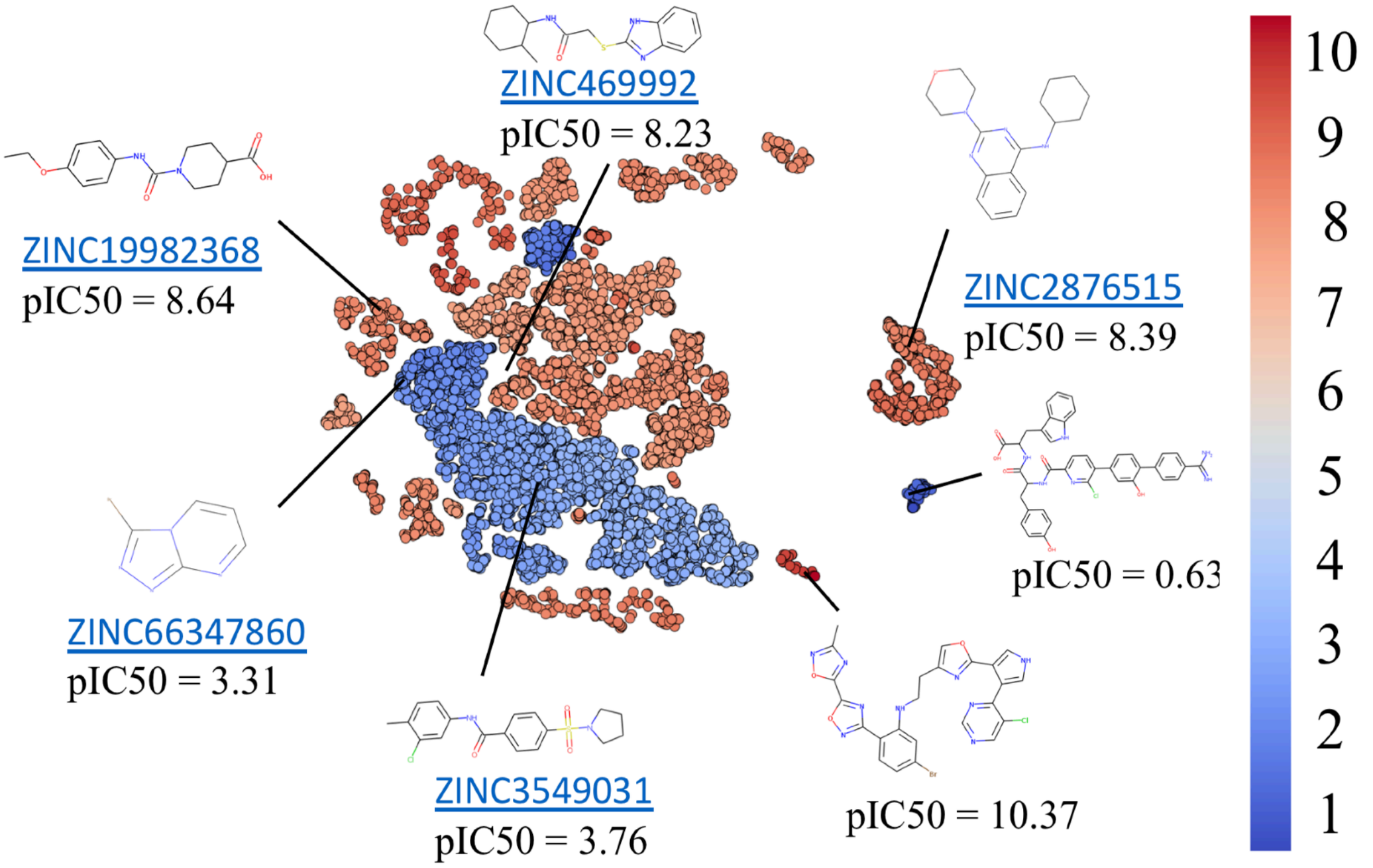


- **Drug design based on deep reinforcement learning**
  - De novo design of molecules with desired properties
  - Based on deep and reinforcement learning
  - Automatically generating simplified molecular-input line-entry system (SMILES) strings for representing molecules
  - Predictive models are derived to forecast the desired properties of the de novo generated compounds





Popova M, Isayev O, Tropsha A. Deep reinforcement learning for de novo drug design. Science advances. 2018 Jul 1;4(7):eaap7885.



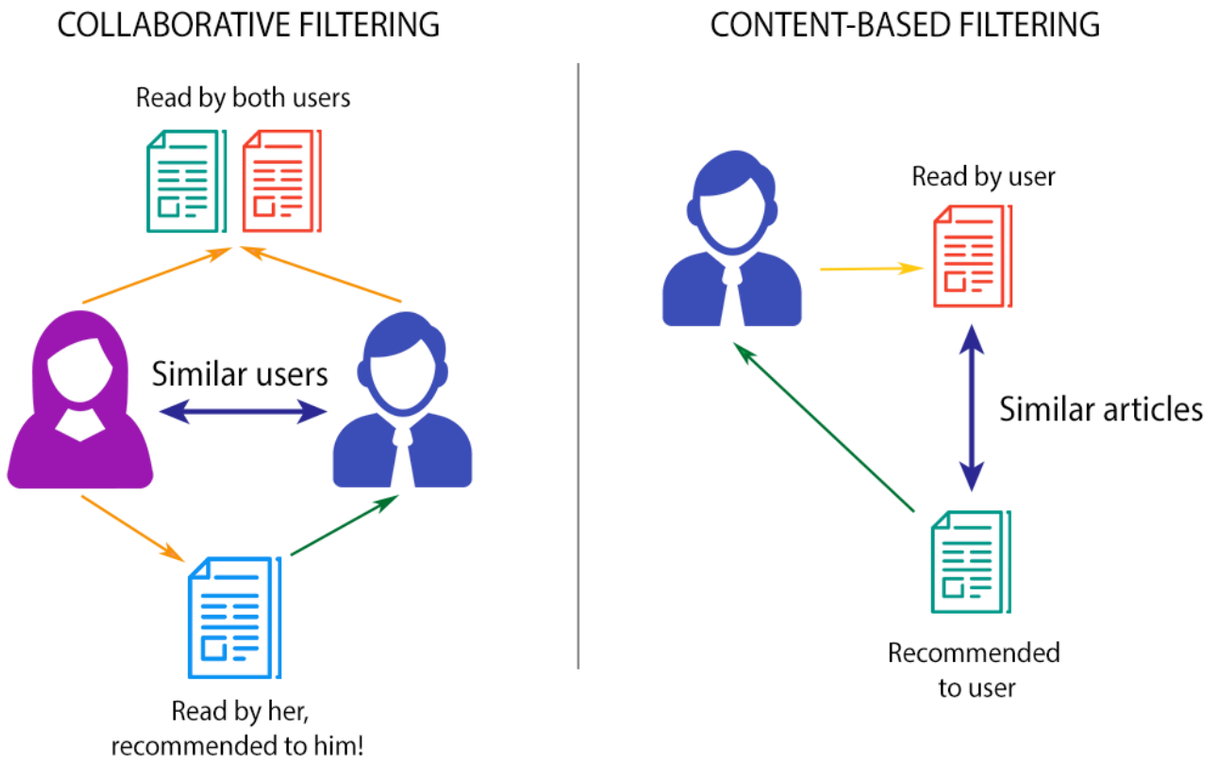
Clustering of generated molecules by t-SNE

Popova M, Isayev O, Tropsha A. Deep reinforcement learning for de novo drug design. Science advances. 2018 Jul 1;4(7):eaap7885.

- **Recommendation systems**

  - Item recommendation: widely used in electronic commence: Taobao, Jingdong, Amazon, Ebay, NetFlix…

  - Collaborative filtering



4	3			5	
5		4		4	
4		5	3	4	
	3				5
	4				4
		2	4		5

Introduction to Recommender Systems
Machine learning Paradigms
Social Network-based Recommender Systems
Learning Spark
Recommender Systems Handbook
Recommender Systems and the Social Web

Collaborative filtering based on matrix factorization

User \ Item	W	X	Y	Z
A		4.5	2.0	
B	4.0		3.5	
C		5.0		2.0
D		3.5	4.0	1.0

Rating Matrix

=

A	*1.2*	*0.8*
B	*1.4*	*0.9*
C	*1.5*	*1.0*
D	*1.2*	*0.8*

User Matrix

X

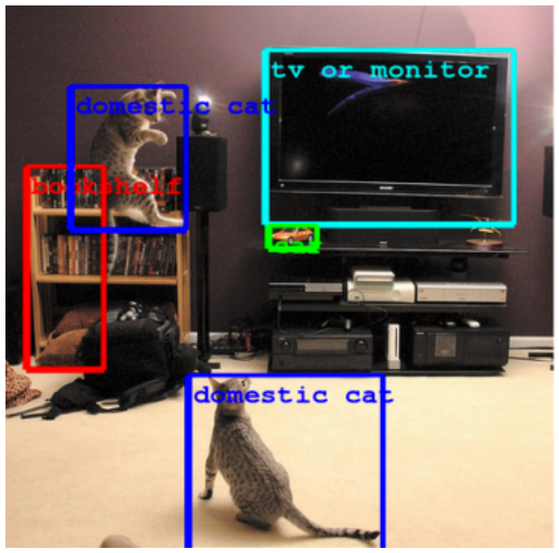
W	X	Y	Z
1.5	*1.2*	*1.0*	*0.8*
1.7	*0.6*	*1.1*	*0.4*

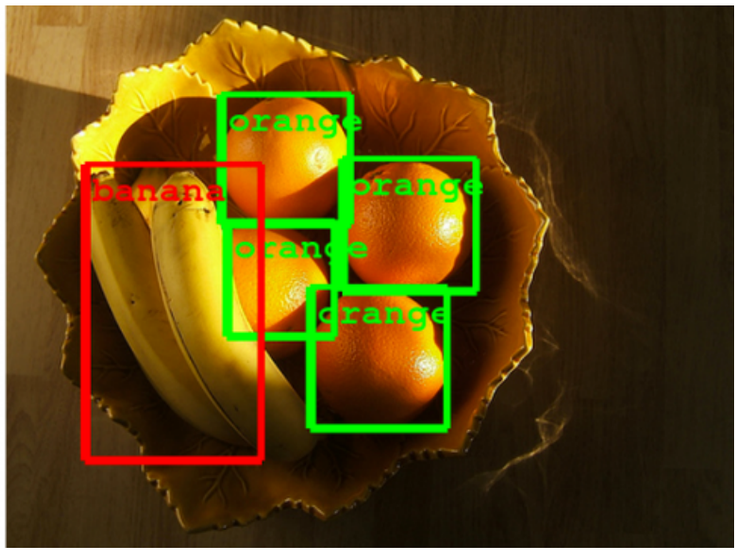
Item Matrix

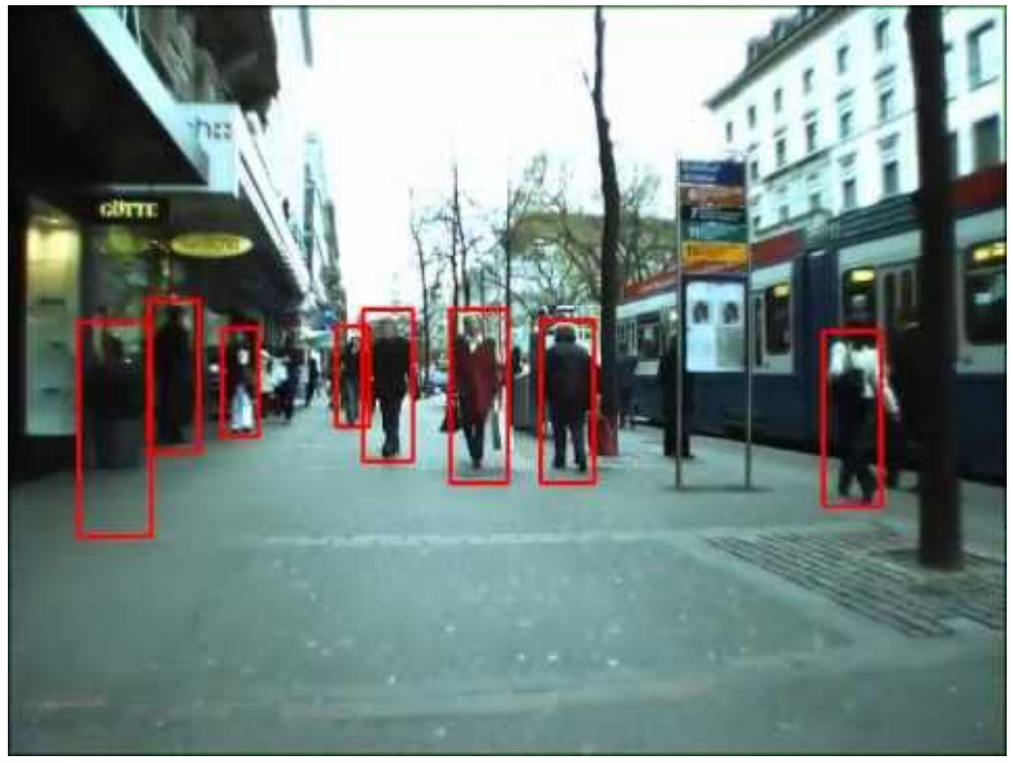
$$\min_{U,V,\Sigma} \sum_{ij \in A} \left(A_{ij} - [U\Sigma V^{\mathrm{T}}]_{ij}\right)^2$$

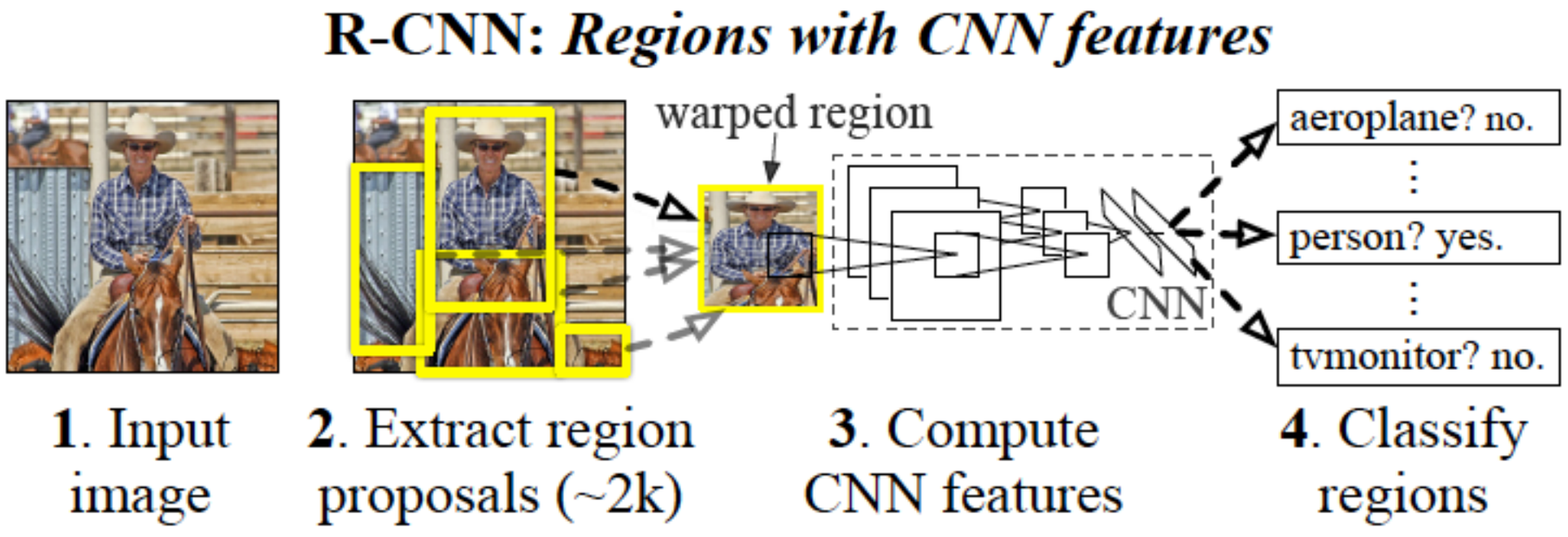
Computer Vision

- **Object recognition and detection**









Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems 2015 (pp. 91-99).

- **Semantic segmentation**

Self-driving

Medical image analysis

- **Style transfer**

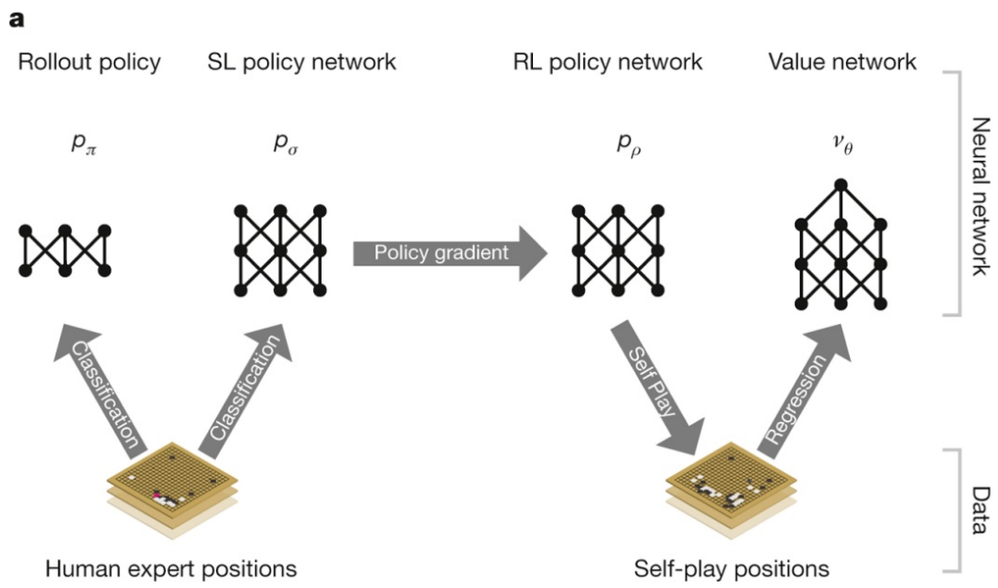
$$\mathcal{L}_{\text{content}}(\,,\,) \approx 0$$

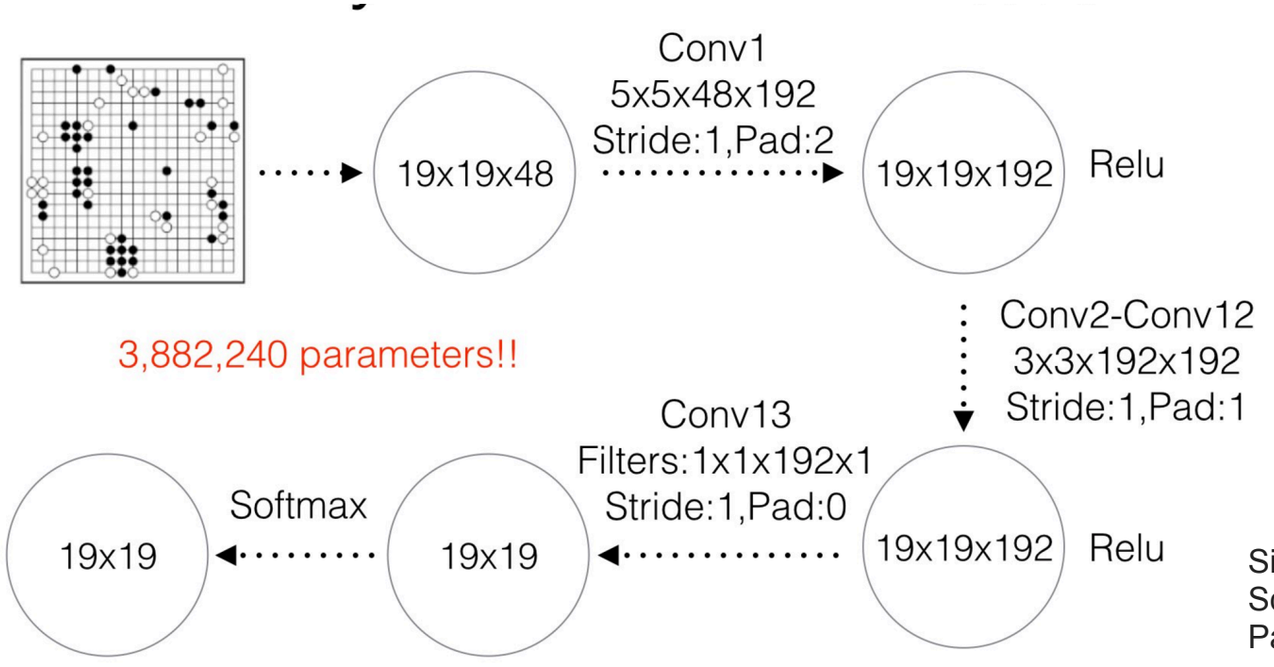
$$\mathcal{L}_{\text{style}}(\,,\,) \approx 0$$

Picture credit to Harish Narayanan.
Gatys, Leon A. et.al. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576* (2015).

Game Playing

- **AlphaGo, AlphaGo Zero**





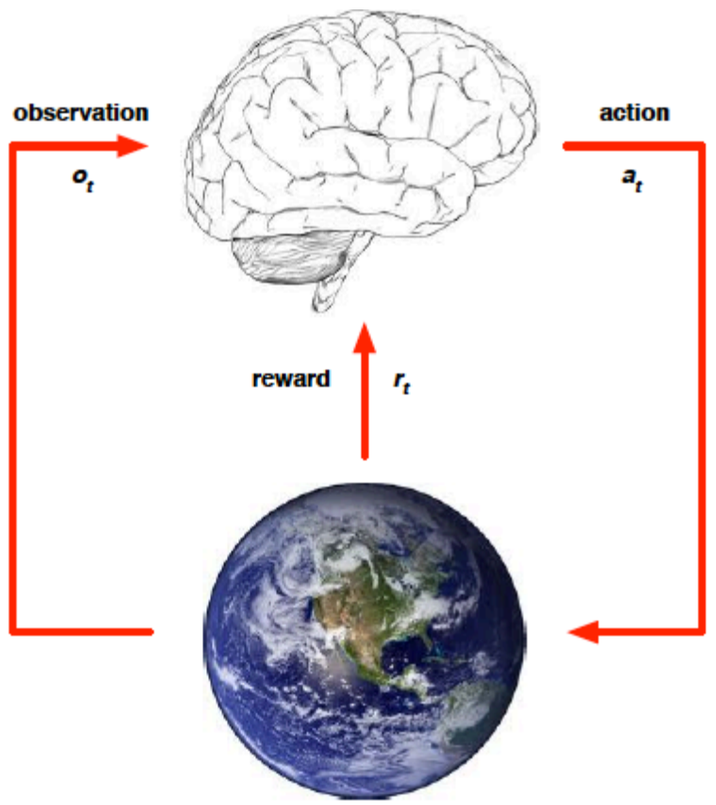
Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I,
Panneershelvam V, Lanctot M, Dieleman S. Mastering the game of Go with deep neural networks and tree search.
Nature. 2016 Jan;529(7587):484.

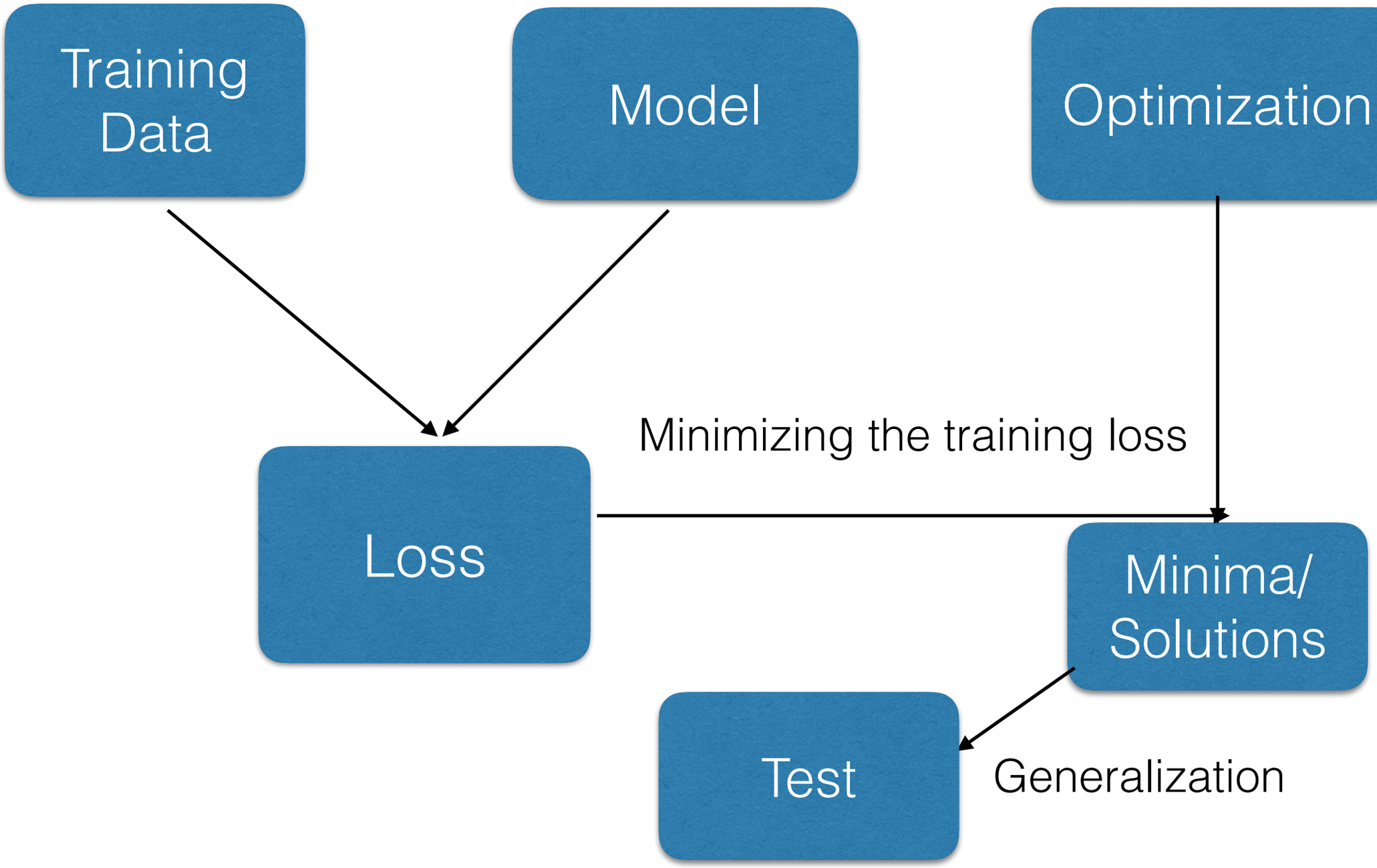
- Dota 2 1v1, 5v5
  - OpenAI Five beat 5 professional players
  - Experience (data) learned ~**180 years per day**
  - Based on deep neural networks and reinforcement learning

https://blog.openai.com/openai-five/

Overview of Machine Learning

- Supervised learning
  - Given data: both input x and its label y $\{x_i, y_i\}_{i=1}^{N}$
  - Learn a mapping from x to y $y = f(x; \theta)$
  - Regression (y continuous), classification (y discrete)
- Unsupervised learning
  - Given data: only x, no label
  - Learn hidden patterns or grouping data
  - Dimensionality reduction, clustering, generative modeling
- Reinforcement learning
  - Given data: delayed feedback signal (reward) after action
  - Learning an optimal policy (i.e. how to act) to achieve a goal




Training Data
Model
Optimization
Loss
Minimizing the training loss
Minima/ Solutions
Test
Generalization

Contents Summary

- Data preprocessing and basic signal processing
- Regression: linear regression and its regularization (ridge regression, Lasso), non-linear regression
- Classification: k-nearest neighbors, decisions trees, naive Bayes, Logistic regression (how to solve: gradient descent), support vector machine (SVM), basis of statistical learning theory
  Ensemble models: random forest, boosting, AdaBoot
- Clustering: k-means, hierarchical clustering, spectral clustering, density-based clustering

- Dimensionality reduction: principle component analysis (PCA), Multi-dimensionality reduction, manifold embedding, sparse coding…

- Probabilistic graphical models: directed and undirected, inference (variational and sampling), learning

- Analyze text, graph, and networks

- Deep learning: convolutional neural network, optimization (SGD and its variants)

- Distributed computing and storage: MapReduce, Hadoop, Spark, MPI, distributed optimization in deep learning

- Reinforcement learning and beyond

Common difficulties in DS/ML

- Modeling
  - Formulate the task at hand as a learning problem
  - Select suitable models for the tasks
- High dimensionality
  - Curse of dimensionality
  - Feature selection, dimensionality reduction
  - Representation learning
- Large-scale data
  - Stochastic approximation
  - Scalable algorithms

Review of Preliminary Knowledge

Review of preliminary knowledge

- Calculus, numerical methods
- Linear algebra
- Probability and statistics

Calculus

- Derivative (i.e. gradient)
  - Single-variable
  - Multi-variable: vector, matrix
  - Chain rule
- Taylor expansion (one-dimensional and multi-dimensional)
- Some important inequalities:
  Cauchy-Schwarz inequality (many versions), Jensen's inequality (convexity)
- Convex optimization
  - Gradient descent

Linear Algebra

- Eigenvalue, eigenvector, matrix diagonalizaton, singular value decomposition (SVD)
- Positive (semi-)definite matrix, matrix inverse
- Solving linear systems
- Vector norm
  - Lp norm, p = 0, 1, 2…
- Matrix norm
  - Frobenius norm, spectral norm..
- Vector/matrix derivative

The Matrix Cookbook:
https://www.ics.uci.edu/~welling/teaching/KernelsICS273B/MatrixCookBook.pdf

Probability and Statistics

- Probability distribution (mean, variance)
  - Gaussian distribution, Gamma distribution, Beta distribution, Multi-nomial distribution, Dirichlet distribution
- Probability rules
- Central Limit Theorem (CLT)
- Bayes Theorem
- Independence, conditional independence
- Maximum likelihood estimation (MLE)

Exercise

- Derive the Cauchy-Schwarz inequality on covariance case
- Derive the Jensen's inequality in the context of probability theory
- Compute the derivative of the followings

$$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b})^T\boldsymbol{W}(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}) =?$$

$$\frac{\partial \mathrm{Tr}(\boldsymbol{A}\boldsymbol{A}^T)}{\partial \boldsymbol{A}} =?$$

matrix trace

Model Evaluation and Model Selection

Model Evaluation and Model Selection

- Empirical error/risk and overfitting
- Evaluation methods
- Performance measure
- Bias and variance trade-off

Empirical error/risk and overfitting

- Error rate, accuracy

- Training error/empirical error, population risk/generalization error (empirically measured by test error)

$$\mathbb{E}_{P_{emp}}[\ell(f(x), y)] = \frac{1}{N}\sum_{i=1}^{N}\ell(f(x_i), y_i)$$ what we actually do

$$\mathbb{E}_{P_{data}}[\ell(f(x), y)] = \int \ell(f(x), y) P_{data}(x, y) dx dy$$ what we really want

- Goal:

  - Based on the training data, learn a model that generalize well. i.e. capture the “general pattern” hidden in data.

  - **Overfitting**: learn an over-complicated model that even captures the **specific** characteristics of the training data

- Alleviate the overfitting: restrict the model complexity

- Underfitting: increase the model complexity

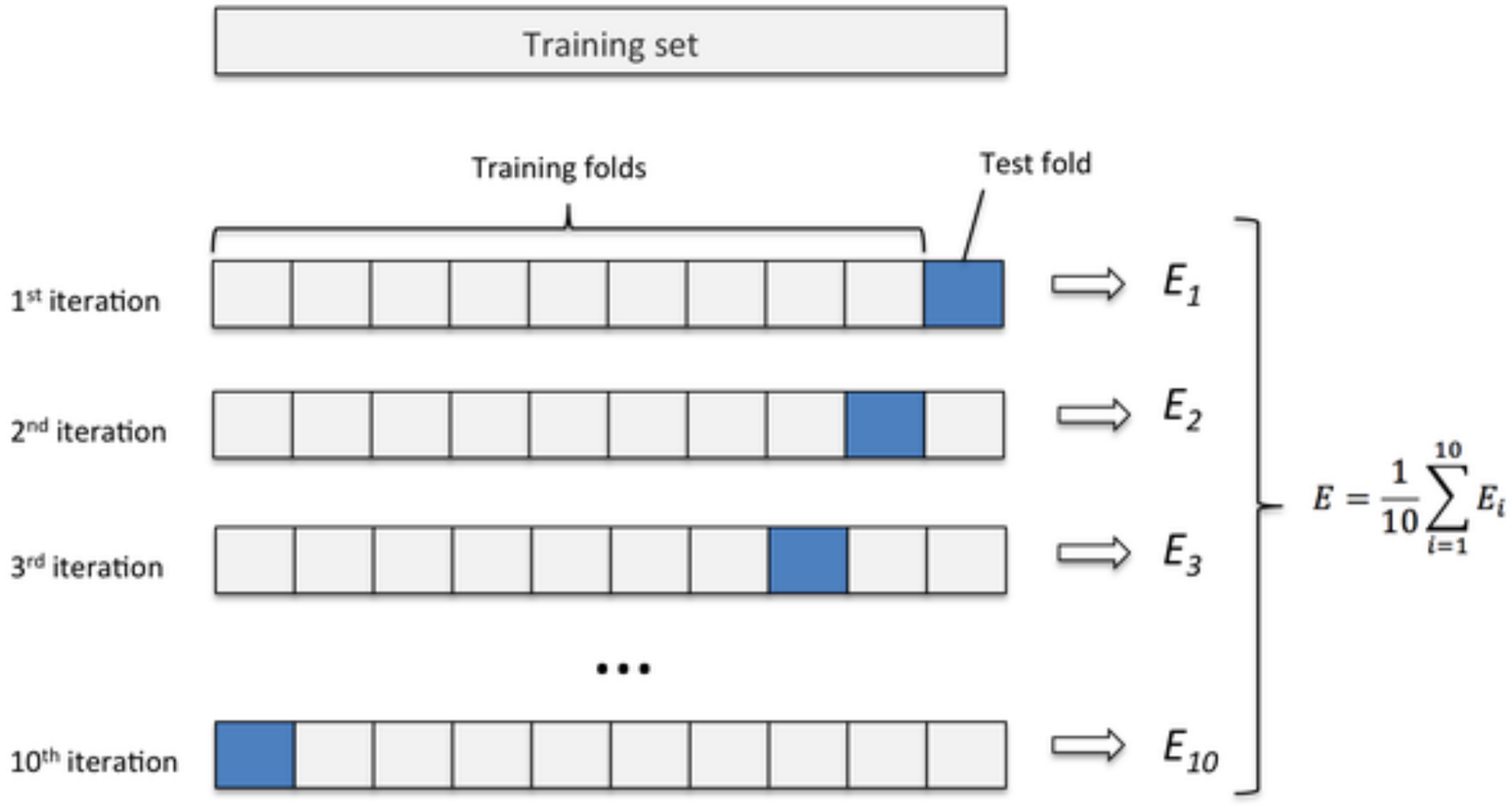
- Overfitting v.s. underfitting: a crucial concept in machine learning

  - Many techniques and analysis to balance the trade-off

Evaluation methods

- Hold-out
  - Split the data into training and test (random and stratified sampling), mutually exclusive.
  - Multiple random splitting and the average
  - Dilemma: large training or large test size. Typically 2/3-4/5 for training and the remaining for test

- Cross-validation
  - Split the data into k mutually exclusive subsets; select k-1 subsets as training and the remaining one for test: **k-fold cross validation**
  - Also can randomly split m times
  - Leave-One-Out (LOO): computationally expensive



- Bootstrapping sampling
  - Sampling with replacement m times
  - For a single sample, after m times, the probability of not being sampled is

$$\lim_{m\to\infty}(1-\frac{1}{m})^m=\frac{1}{e}\approx 0.368$$

  - Suitable for small-size dataset
  - Typically used in ensemble learning models (we'll talk about this in later lectures)
  - Bootstrapping changes data distribution, which might introduce estimation bias

- Hyper-parameter tuning

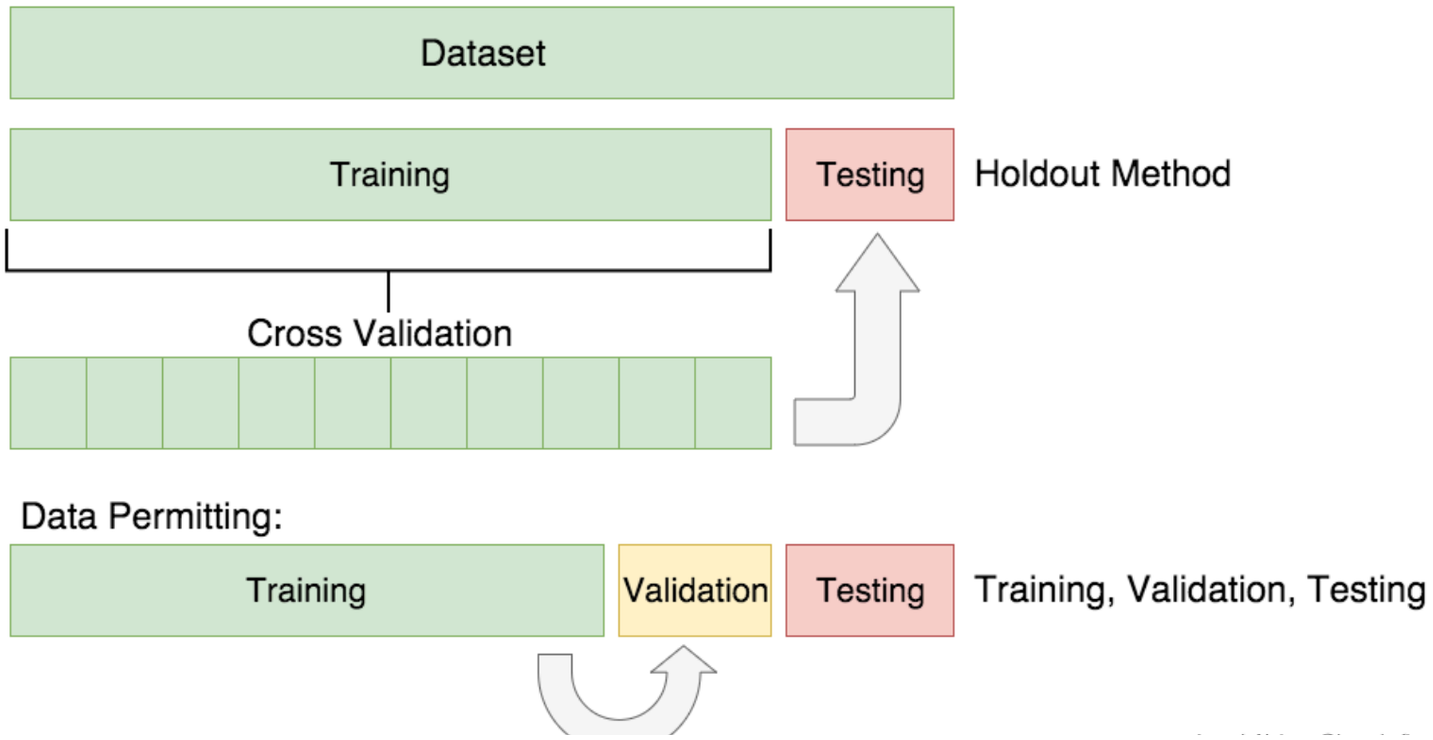
  - Hyper-parameters: configurations of models or learning algorithms
    E.g. depth of neural networks, number of decision trees in random forest, learning rate in optimization algorithms

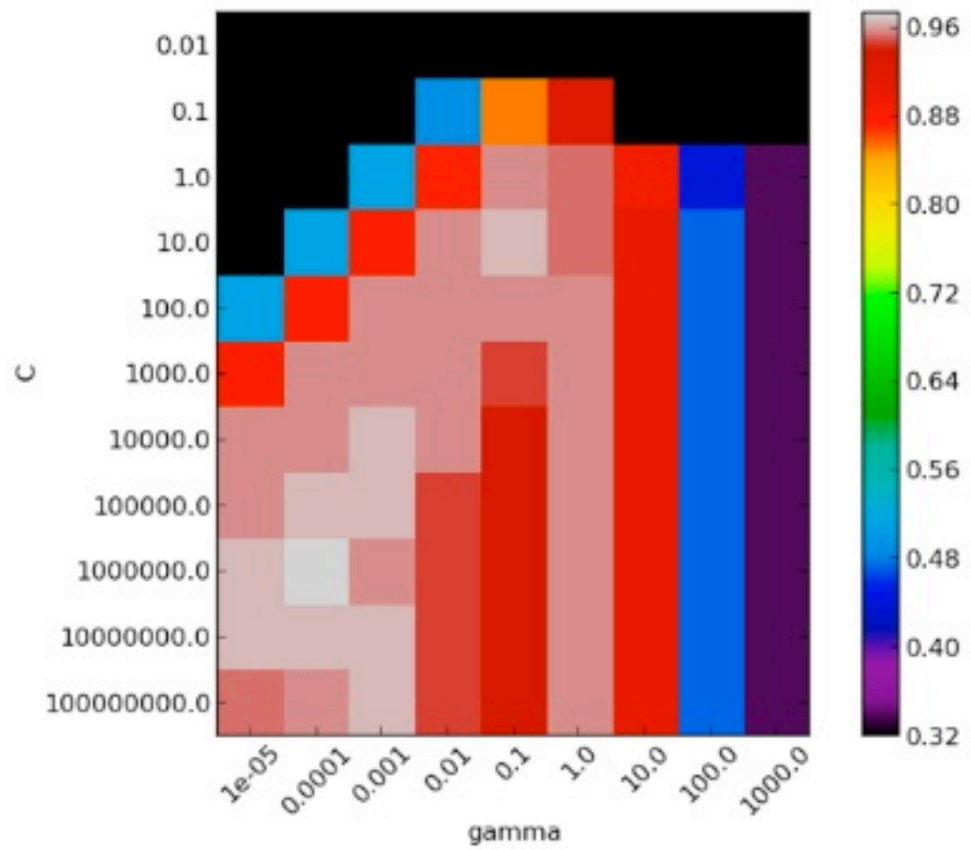
  - The settings of the hyper-parameters typically significantly affect the model performance

  - How to tune the hyper-parameter?

    - Grid search based on validation set

    - Curse of dimensionality if many parameters





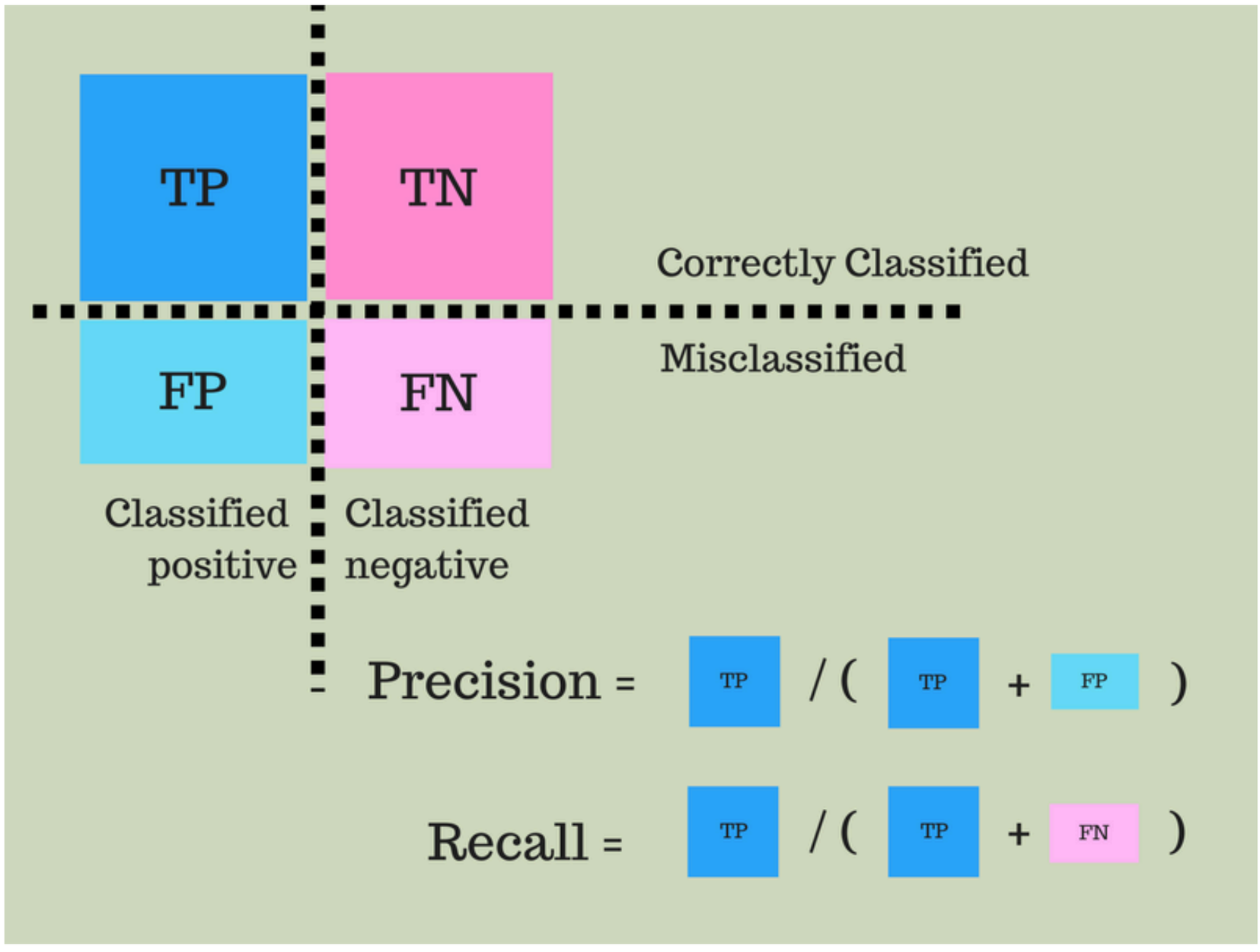
Performance Measure

- Accuracy or error rate is not enough in many cases

- Precision and recall in binary classification problem

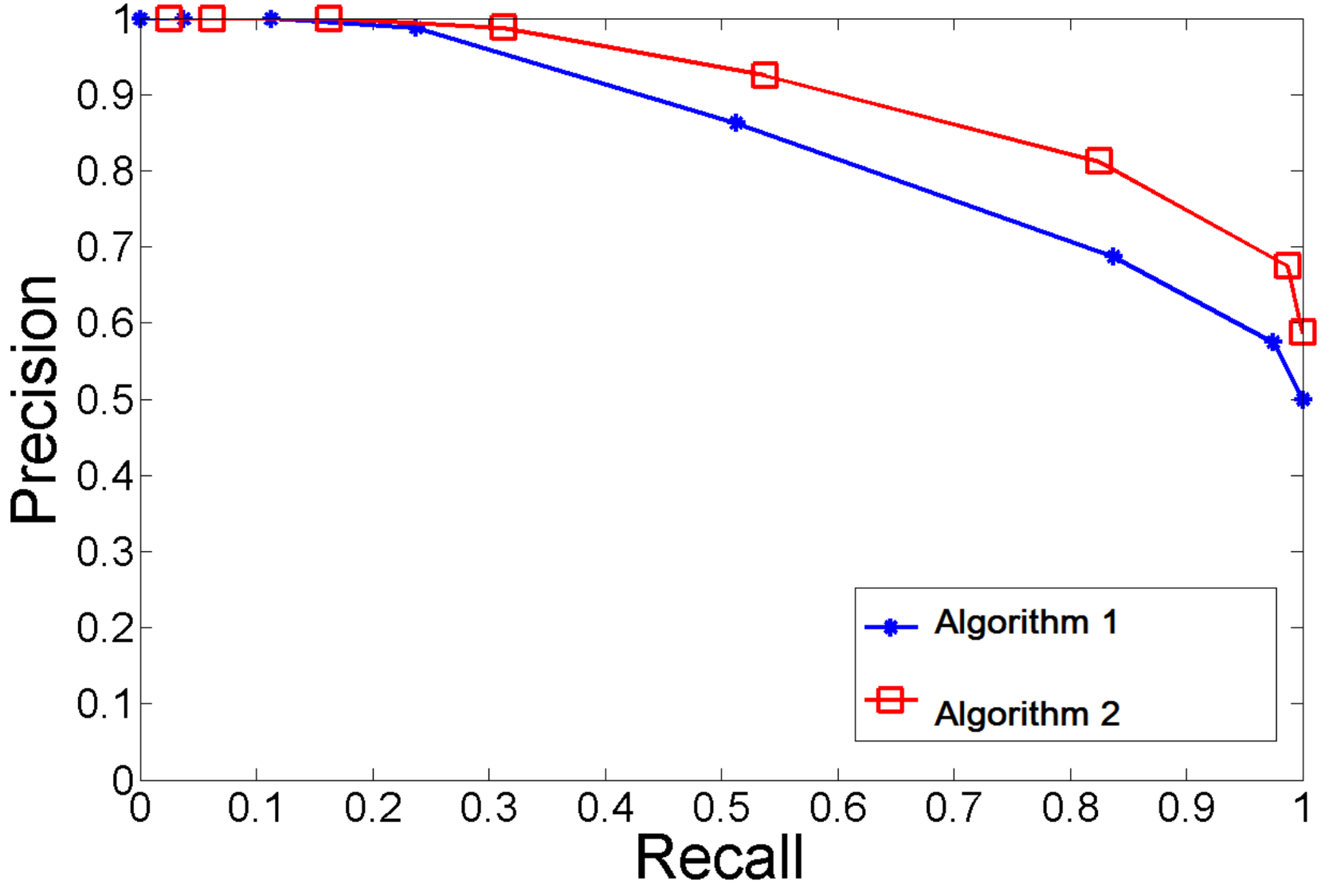
  true positive
  true negative
  false positive
  false negative

Confusion matrix



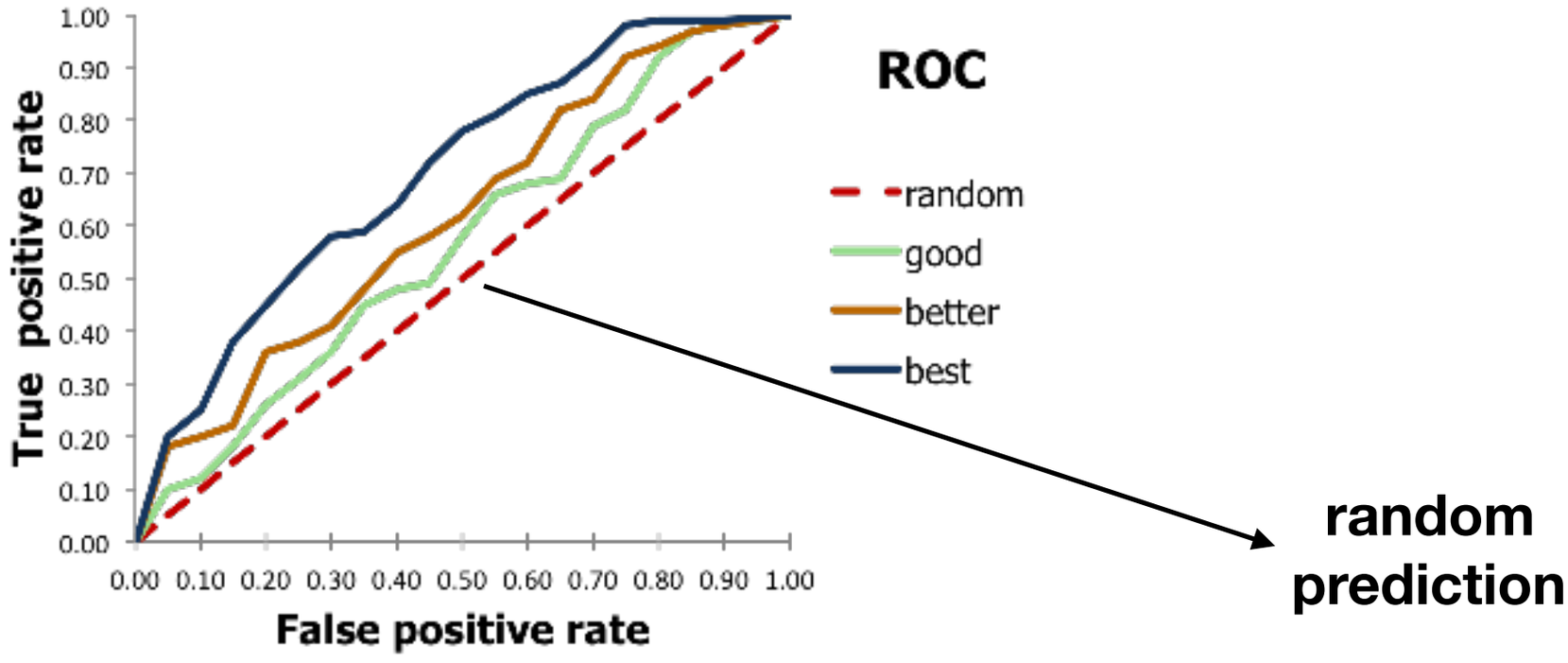
Precision-Recall Curve

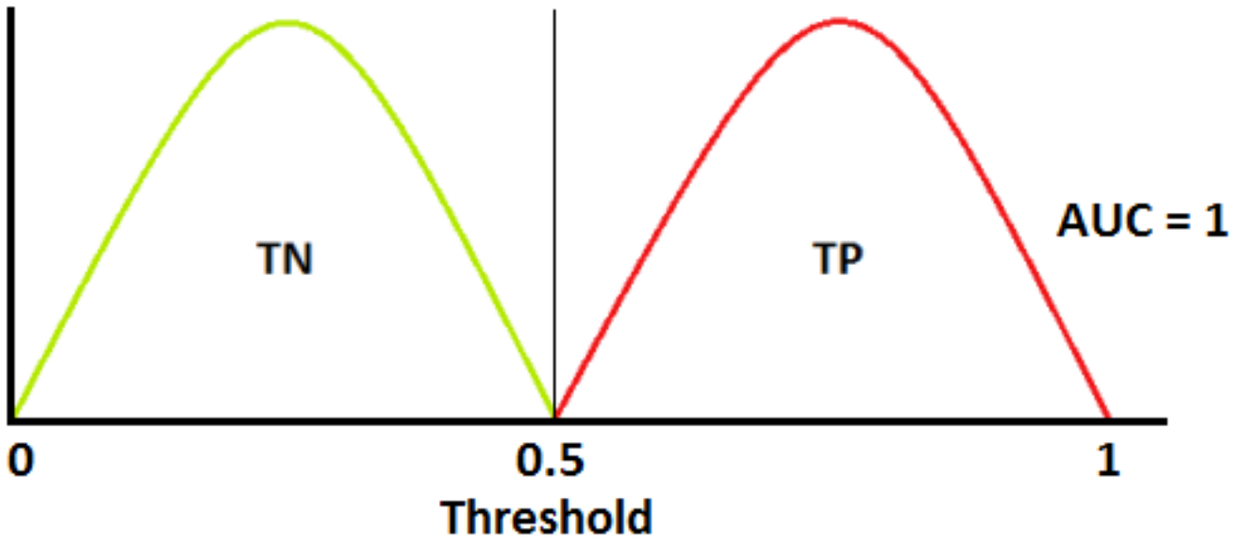
Algorithm 2 better than 1

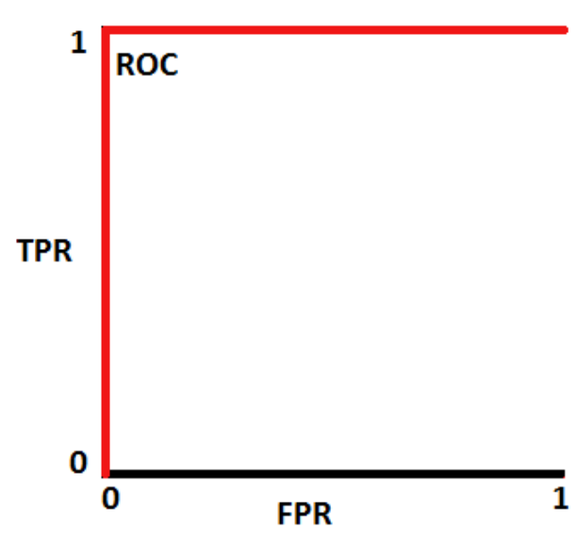


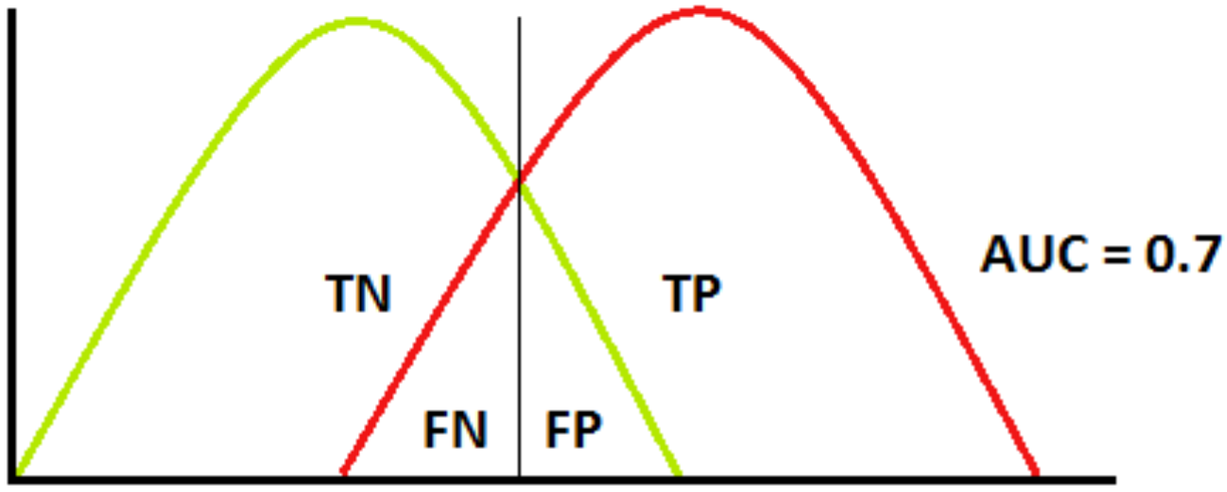
- Receiver Operating Characteristic Curve (**ROC curve**)
  - Ranking the samples according to the predicted probability
    - if P(predicted prob > threshold), classified as a positive one
    - each point on the curve is a pair
      (False Positive Rate, True Positive Rate)

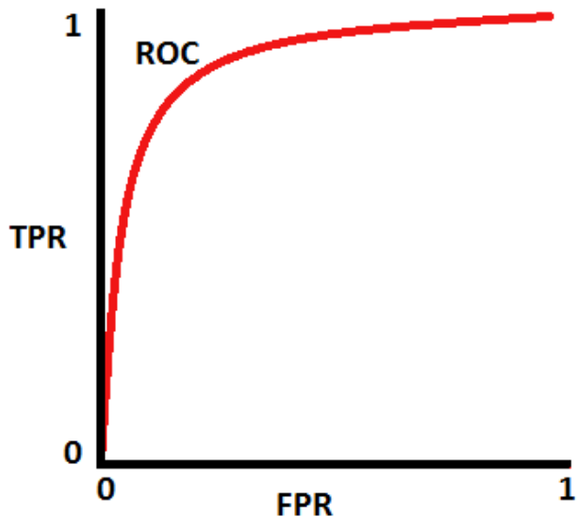
      TPR = TP / (TP + FN); FPR = FP / (TN + FP)

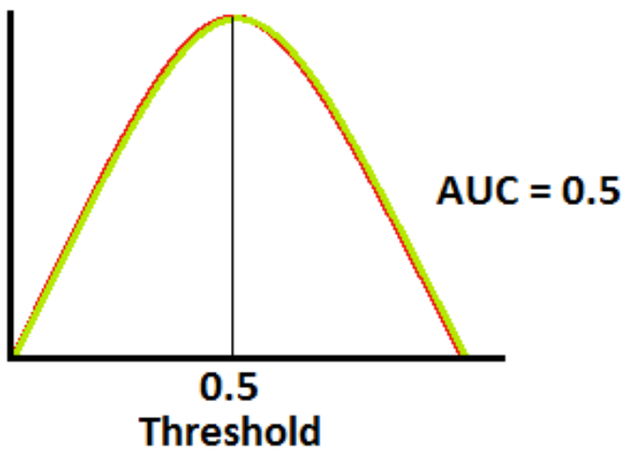


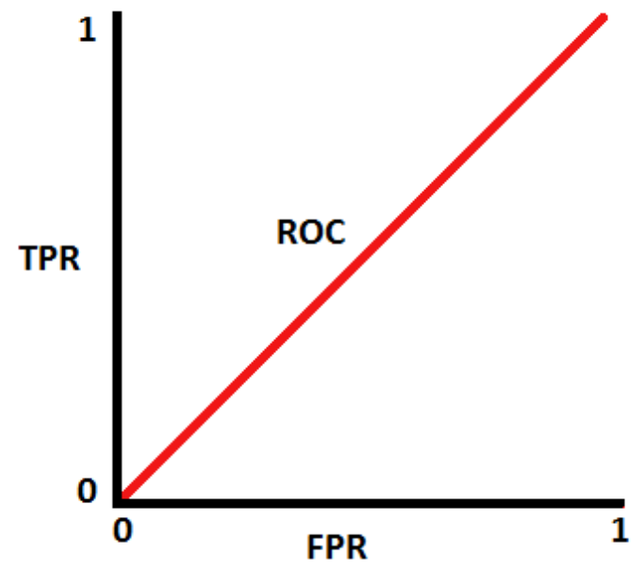






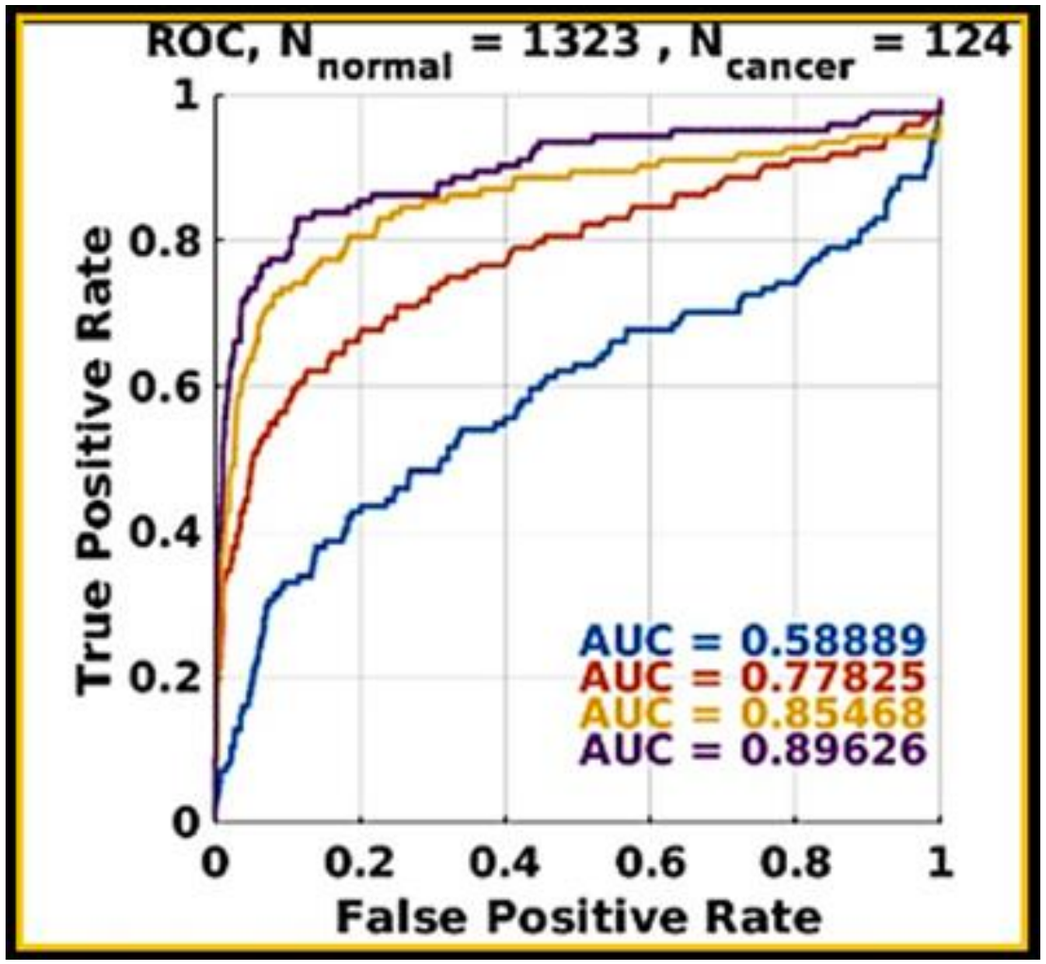






Source: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

- Area Under ROC Curve (**AUC**)
  - One way to measure the performance of a classifier



Bias-variance tradeoff

- Bias–variance decomposition of squared error

$$
\begin{aligned}
\mathbf{E}\left[(y-\hat{f})^2\right] &= \mathbf{E}[y^2+\hat{f}^2-2y\hat{f}] \\
&= \mathbf{E}[y^2]+\mathbf{E}[\hat{f}^2]-\mathbf{E}[2y\hat{f}] \\
&= \mathbf{Var}[y]+\mathbf{E}[y]^2+\mathbf{Var}[\hat{f}]+\left(\mathbf{E}[\hat{f}]\right)^2-2f\,\mathbf{E}[\hat{f}] \\
&= \mathbf{Var}[y]+\mathbf{Var}[\hat{f}]+\left(f^2-2f\,\mathbf{E}[\hat{f}]+(\mathbf{E}[\hat{f}])^2\right) \\
&= \mathbf{Var}[y]+\mathbf{Var}[\hat{f}]+(f-\mathbf{E}[\hat{f}])^2 \\
&= \sigma^2+\mathbf{Var}[\hat{f}]+\mathbf{Bias}[\hat{f}]^2
\end{aligned}
$$