

UVA CS 4774: Machine Learning

Lecture 1: Introduction

Dr. Yanjun Qi

University of Virginia
Department of Computer Science

Roadmap

- Course Logistics
- History and Now
- A Rough Plan of Course Content

ATT:

- I will video-record only when I talk/teach
- One TA will act as coordinator, please type in questions via Chats / He will read out and ask me to answer
- Each session will include multiple small modules --- each about 20mins (each into a different recorded video)

Welcome

- Course Website:
- <https://qiyanjun.github.io/2020f-UVA-CS-MachineLearningDeep/>

We focus on learning fundamental principles, algorithm design and deep learning methods and applications.

Objective

- To help students get able to build simple machine learning tools
 - (not just a tool user!!!)
- Key Results:
 - Able to build a few simple machine learning methods from scratch
 - Able to understand a few complex machine learning methods at the source code level
 - To re-produce / or invent one cutting-edge machine learning application / algorithm for GOOD USE

Course Staff

- Instructor: Prof. Yanjun Qi
 - QI: /ch ee/
 - You can call me “professor”, “professor Qi”;
 - I have been teaching Graduate-level and Under-Level Machine Learning course for years!
 - My research is about machine learning
- TA and Office Hour information @ CourseWeb

Course Material

- Text books for this class is:
 - NONE
 - Multiple good reference books are shared via CourseWeb
- My slides – if it is not mentioned in my slides, it is not an official topic of the course
- Your UVA Collab for Assignments and Project
- Google Forms for Quizzes

Course Background Needed

- **Background Needed**
 - Calculus, Basic linear algebra,
 - Basic probability and Basic Algorithm
 - Statistics is recommended.
 - **Python** is required for all programming assignments

Assignments

- Assignments (50%, with five assignments)
- See policy in <https://qiyajun.github.io/2020f-UVA-CS-MachineLearningDeep//About/>

Quiz

- Class quizzes (20%): Each takes 10 mins via google form;
 - We will have a total of 12 quizzes
 - Your top 10 scored will be counted into 20%
 - Within Our Zoom Synchronous sessions
 - Will be close book (please follow honor code!)
- See policy in <https://qiyanjun.github.io/2020f-UVA-CS-MachineLearningDeep//About/>

Course Project

- Final project (30%): Three potential types:
 - a. To produce one machine learning project on cutting-edge data applications with health or social impacts
 - b. Survey and benchmark multiple pytorch (or tensorflow if you prefer) libraries with a shared goal
 - c. To Reproduce a cutting-edge machine learning paper, for instance from Top Venues' most cited recent papers
- See policy in <https://qiyajun.github.io/2020f-UVA-CS-MachineLearningDeep//About/>

Thank You



Roadmap

- Course Logistics
- History and Now
- A Rough Plan of Course Content

Artificial Intelligence Today

Watson (IBM)



DeepMind (Google)



Echo
(Amazon)



SIRI
(Apple)

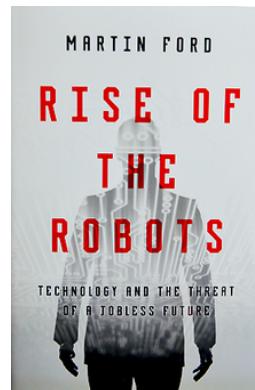
Boston Dynamics

Impact: Good and Bad?

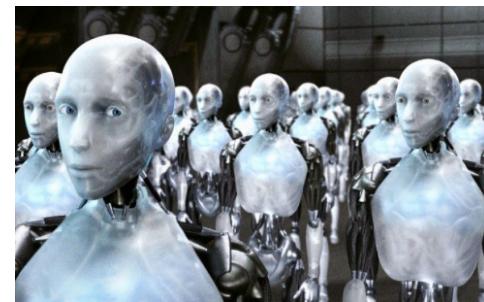


Economic, cultural, social, health... endless disruption

Martin Ford,
Rise of the Robots



Labor - McKinsey >50%
of jobs automated



Elon Musk, artificial intelligence...
existential threat

Artificial intelligence (AI)

The study of computer systems that attempt to model and apply the intelligence of the human mind.

What defines “intelligence”?

Why is it that we assume humans are intelligent?

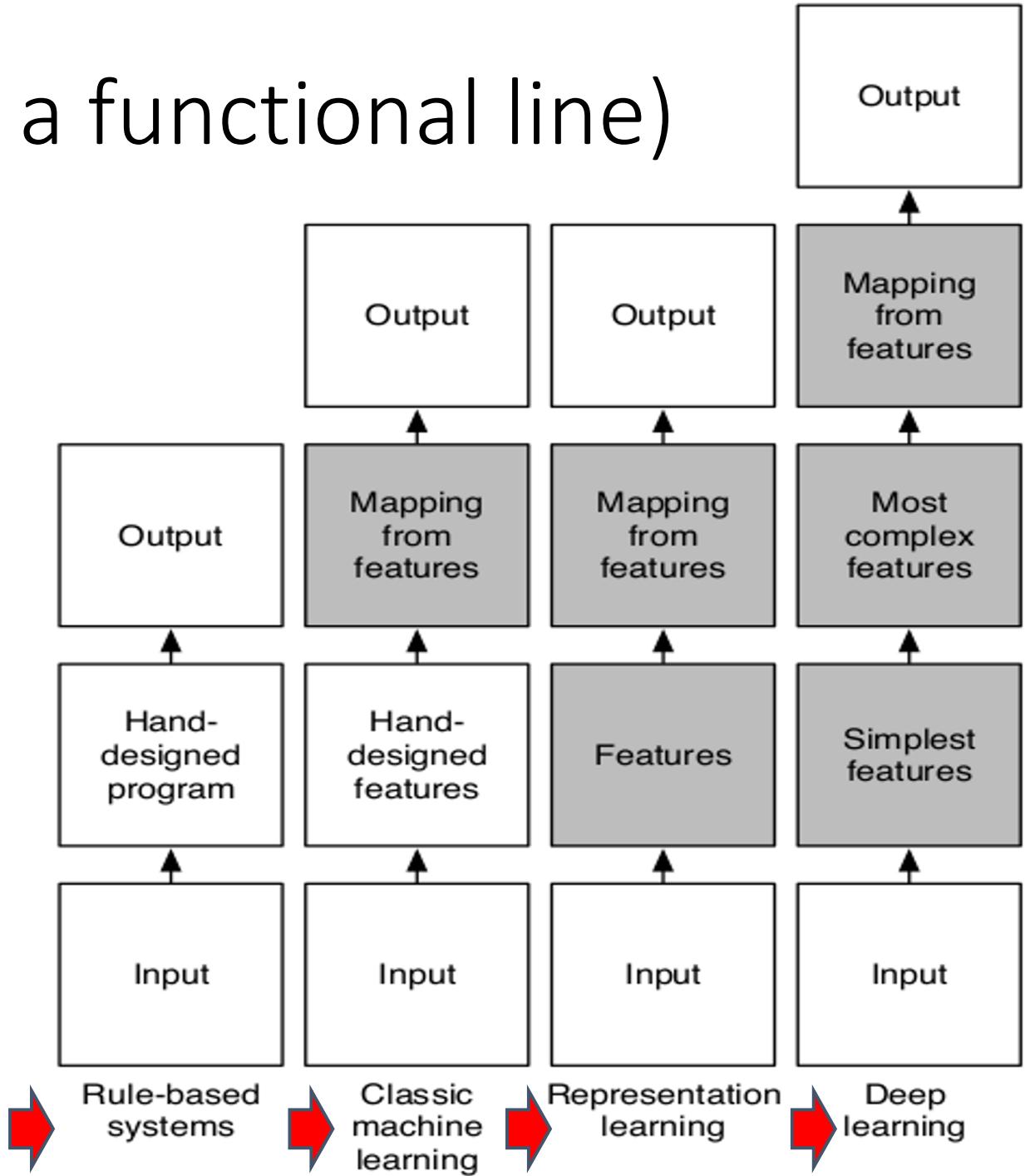
Are monkeys intelligent? Dogs? Ants? Pine trees?

How to build more intelligent computer / machine ?

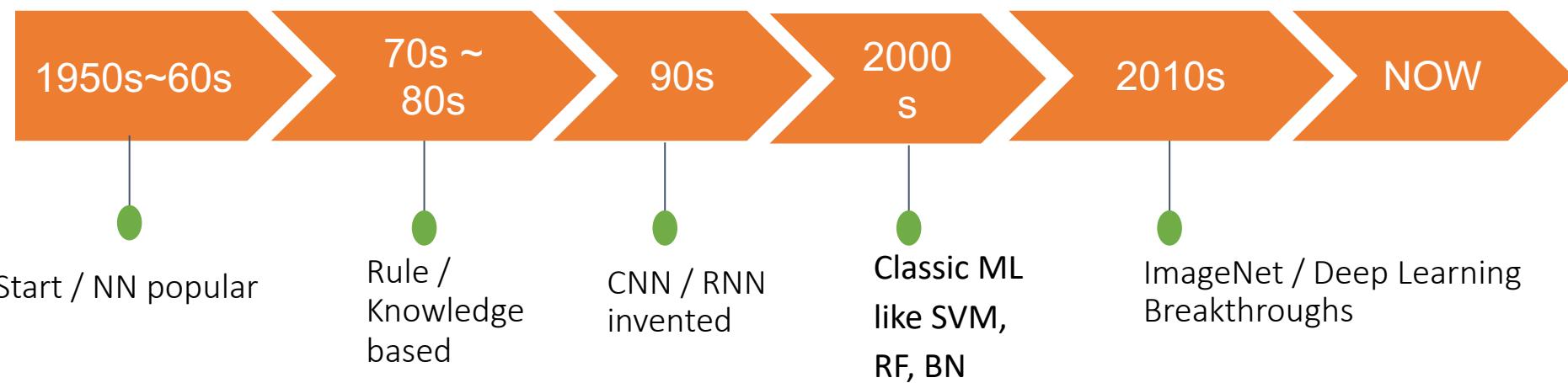
- Able to **perceive** the world,
 - e.g., objective recognition, speech recognition, ...
- Able to **understand** the world,
 - e.g., machine translation, text semantic understanding
- Able to **Interact** with the world,
 - e.g., AlphaGo, AlphaZero, self-driving cars, ...
- Able to **think / reason / learn**,
 - e.g., learn to program programs, learn to search deepNN architecture, ...
- Able to **imagine** / to make **analogy**,
 - e.g., learn to draw with styles,

16

History (on a functional line)



History (on a time line)



Auther Samuel popularized the term "[machine learning](#)" in 1959.^[4] The Samuel Checkers-playing Program was among the world's first successful self-learning programs;

Early History

- In 1950 English mathematician Alan Turing wrote a landmark paper titled “Computing Machinery and Intelligence” that asked the question: **“Can machines think?”**
- Further work came out of a 1956 workshop at Dartmouth sponsored by John McCarthy. In the proposal for that workshop, he coined the phrase a “study of artificial intelligence”
- Expert systems (70s, 80s)
 - A software system based the knowledge of human experts;
 - **Rule-based system**
 - processes rules to draw conclusions
 - Idea is to give AI systems lots of information to start with

MIT Technology Review

10 Breakthrough Technologies 2013

Think of the most frustrating, intractable, or simply annoying problems you can imagine. Now think about what technology is doing to fix them. That's what we did in coming up with our annual list of 10 Breakthrough Technologies. We're looking for technologies that we believe will expand the scope of human possibilities.

Deep Learning

10 Breakthrough Technologies 2017

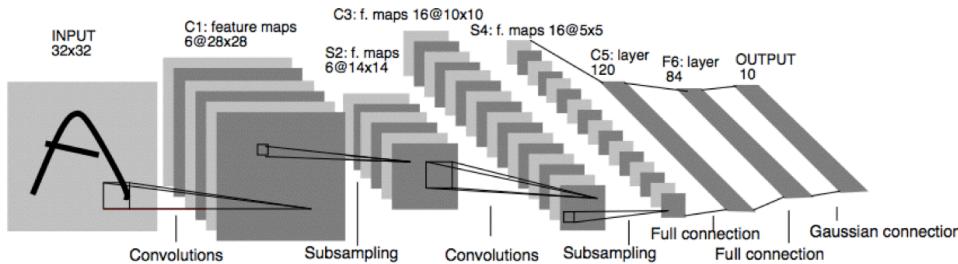
These technologies all have staying power. They will affect the economy and our politics, improve medicine, or influence our culture. Some are unfolding now; others will take a decade or more to develop. But you should know about all of them right now.



Deep
Reinforcement
Learning

Generative
Adversarial
Network (GAN)

- **1952-1969 Enthusiasm:** Lots of work on neural networks
- **1990s:** Convolutional neural network (CNN) and Recurrent neural network (RNN) were invented

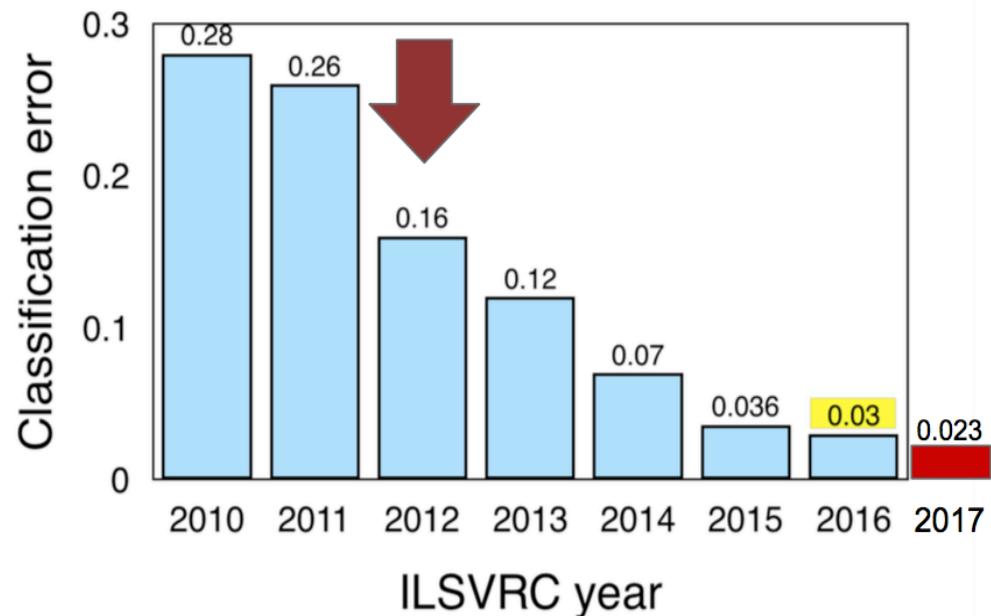


Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86(11): 2278–2324, 1998.

ImageNet Challenge



- 2010-11: hand-crafted computer vision pipelines
- 2012-2016: ConvNets
 - 2012: AlexNet
 - major deep learning success
 - 2013: ZFNet
 - improvements over AlexNet
 - 2014
 - VGGNet: deeper, simpler
 - InceptionNet: deeper, faster
 - 2015
 - ResNet: even deeper
 - 2016
 - ensembled networks
 - 2017
 - Squeeze and Excitation Network

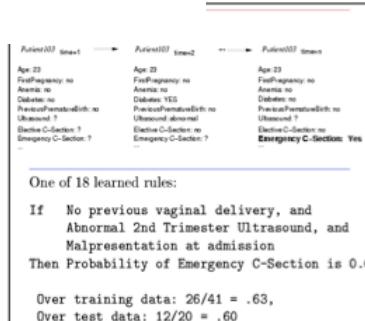


ImageNet Competition:

[Training on 1.2 million images [X] vs. 1000 different word labels [Y]]

Deep Learning is Changing the World

How may I help you, human?

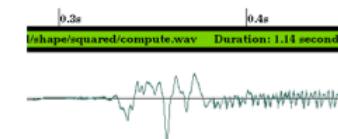


Mining Databases



Text analysis

Peter H. van Oppen, Chairman of the Board & Chief Executive Officer
Mr. van Oppen has served as chairman of the board and chief executive officer of ADIC since its acquisition by Interpoint in 1994 and a director of ADIC since 1986. Until its acquisition by Crane Co. in October 1996, Mr. van Oppen served as chairman of the board of Interpoint, president and chief executive officer of Interpoint. Prior to 1985, Mr. van Oppen worked as a consulting manager at Price Waterhouse LLP and at Bain & Company in Boston and London. He has additional experience in medical electronics and venture capital. Mr. van Oppen also serves as a director of Seacor Worldwide and Spacelabs Medical, Inc.. He holds a B.A. from Whitman College and an M.B.A. from Harvard Business School, where he was a Baker Scholar.



Speech Recognition



Control learning



Object recognition

Many more !

Reason of Recent Deep Learning breakthroughs:

Plenty of (Labeled)
Data for ML

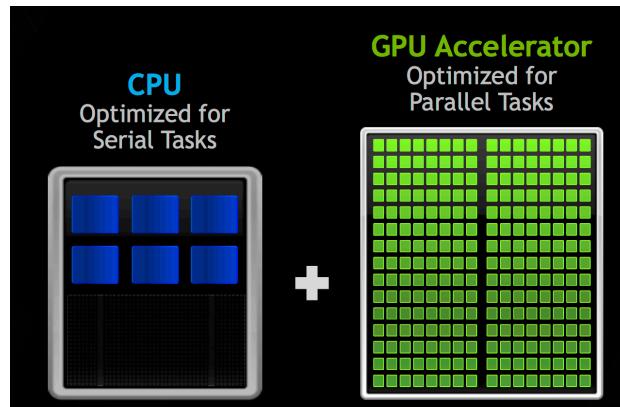
Advanced
Computer
Architecture that
fits DNNs

Powerful DNN
platforms /
Libraries

Reason: Plenty of (Labeled) Data

- **Text**: trillions of words of English + other languages
- **Visual**: billions of images and videos
- **Audio**: thousands of hours of speech per day
- **User activity**: queries, user page clicks, map requests, etc,
- **Knowledge graph**: billions of labeled relational triplets
- Genomics data:
- Medical Imaging data:

Reason: Advanced Computer Architecture that fits DNNs



http://www.nvidia.com/content/events/geoInt2015/LBrown_DL.pdf

Neural Networks	GPUs
Inherently Parallel	✓
Matrix Operations	✓
FLOPS	✓

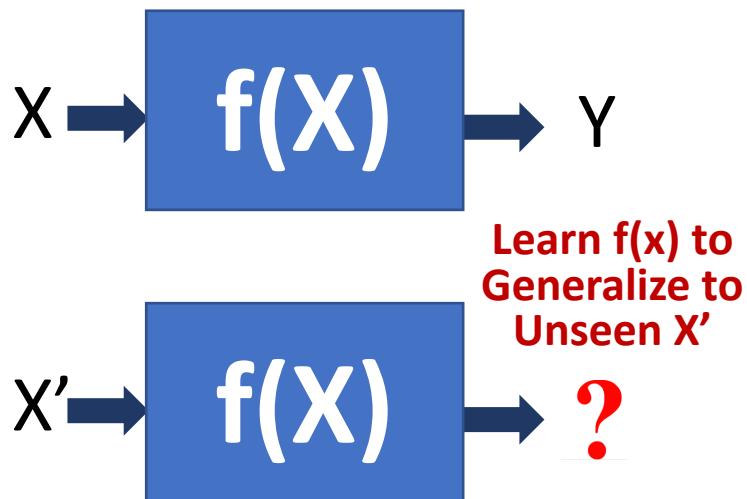
GPUs deliver --

- *same or better prediction accuracy*
- *faster results*
- *smaller footprint*
- *lower power*

Reason: Data-Driven Machine Learning Algorithms and Platforms

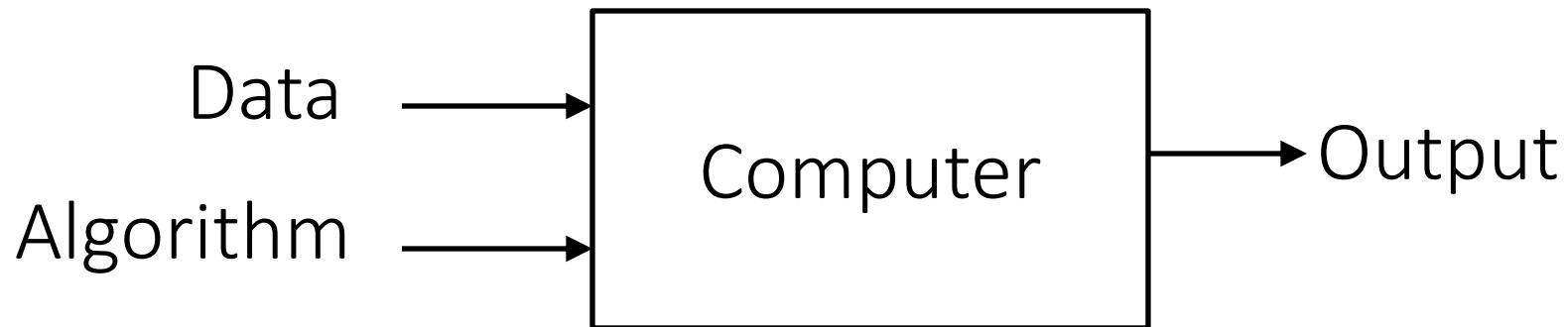
- Inductive reasoning

- Generalizations from observed data to unseen data



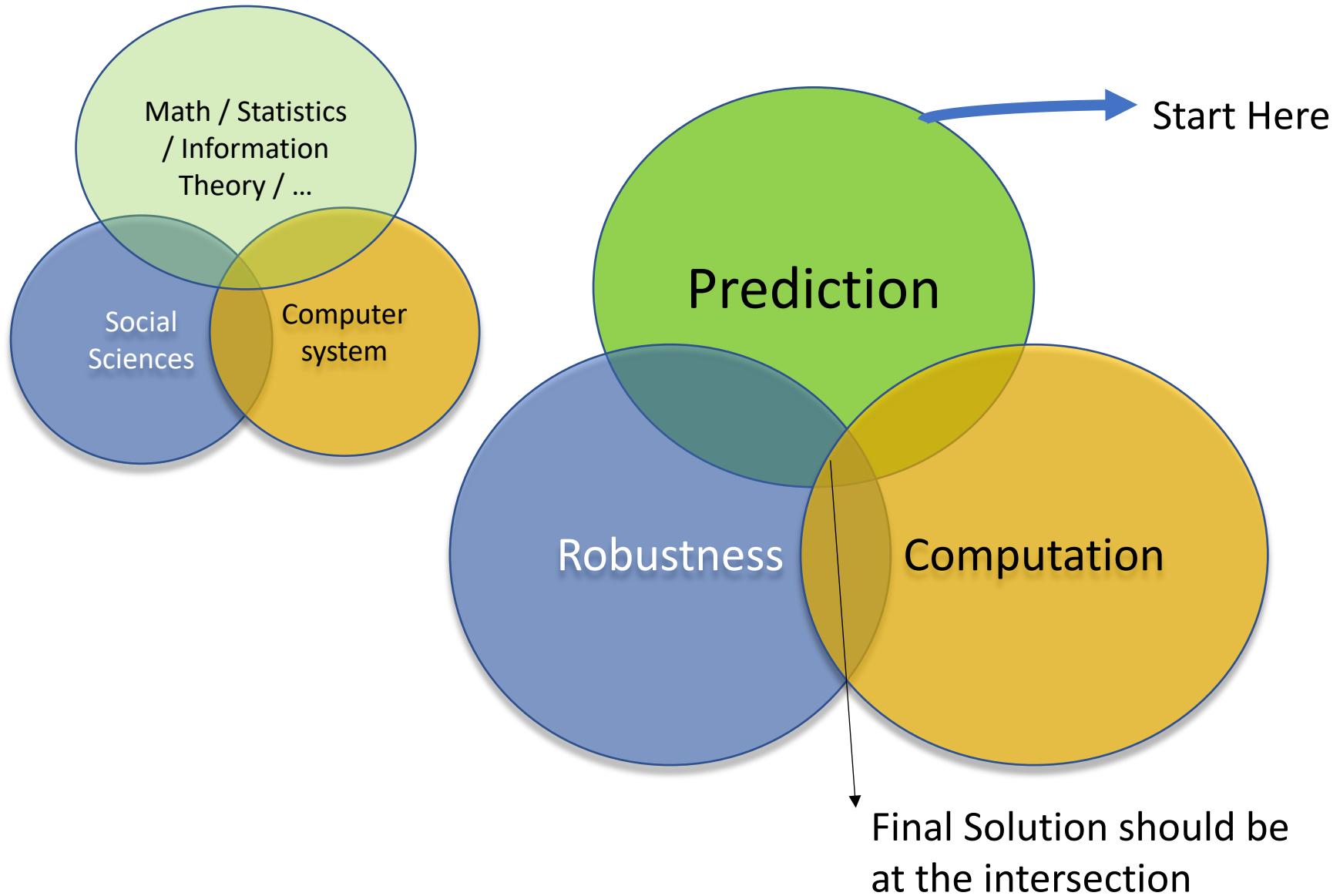
- Able to build computer systems that can learn and adapt from their experience
- Well-engineered software architectures to build upon
- Provide prediction accuracy
- Create software that improves over time

Traditional Programming



Machine Learning





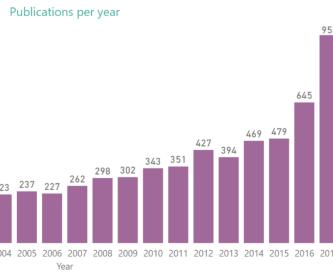
Future?

ML + Digital Data Platforms: Unprecedented Era

Hyper time compression
new disruptive innovations

Extreme convergence
of multiple domains

Exponential accelerating automation
– smart sensors and the billion IoT devices



Universal connectivity linked
by a digital mesh

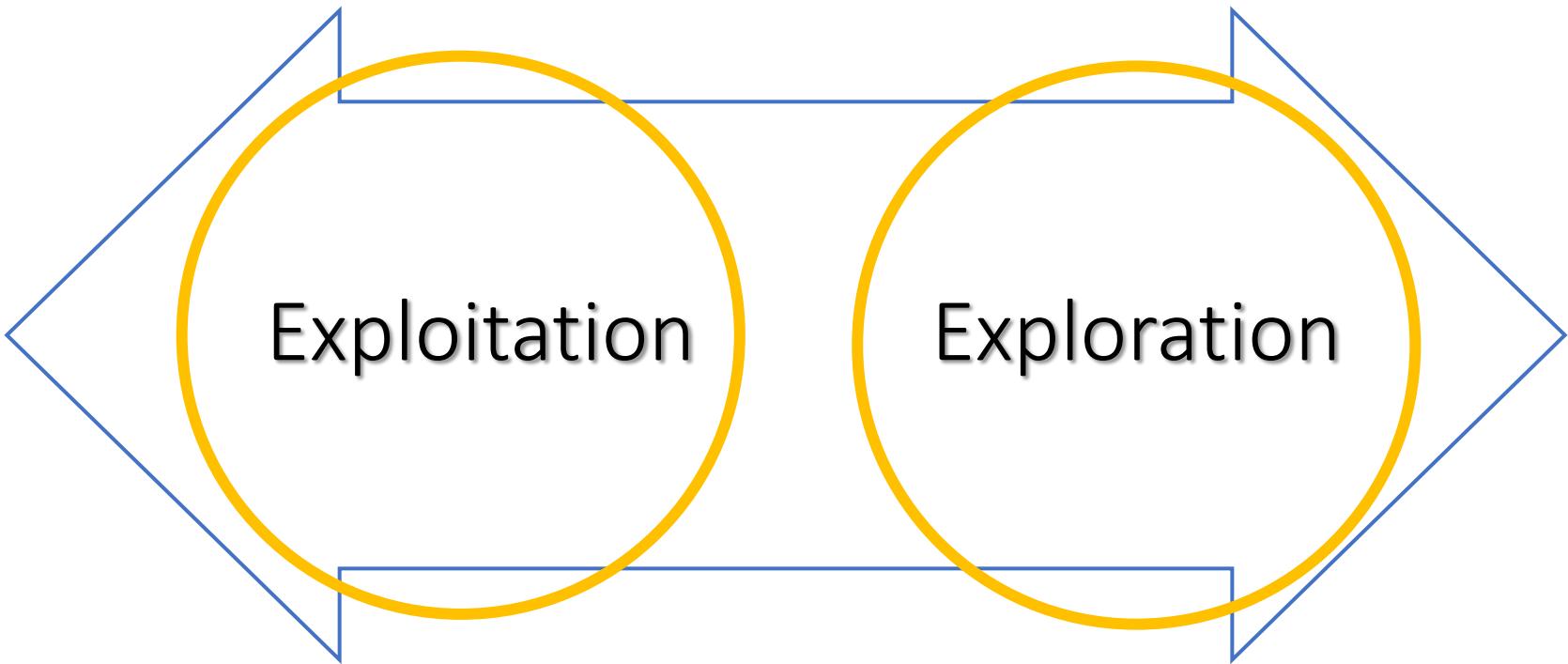


We always overestimate the change
that will occur in the next two years
and underestimate the change
that will occur in the next ten.

- Bill Gates, The Road Ahead, 1996

General Lessons for Excellence

- • Good breath in fundamentals is key
- • Strength in particular targeted topics help standing out



Highly Recommend Two Extra-curriculum books:

1. Book: By Dr. Domingos: Master Algorithm

So How Do Computers Discover New Knowledge?

1. **Symbolists**--Fill in gaps in existing knowledge
2. **Connectionists**--Emulate the brain
3. **Evolutionists**--Simulate evolution
4. **Bayesians**--Systematically reduce uncertainty
5. **Analogizers**--Notice similarities between old and new

SRC: Pedro Domingos ACM Webinar Nov 2015
<http://learning.acm.org/multimedia.cfm>

Highly Recommend Two Extra-curriculum books

- 2. Book: Homo Deus- A Brief History of Tomorrow
 - <https://www.goodreads.com/book/show/31138556-homo-deus>
 - “Homo Deus explores the projects, dreams and nightmares that will shape the twenty-first century—from overcoming death to creating artificial life. It asks the fundamental questions: Where do we go from here? And how will we protect this fragile world from our own destructive powers? This is the next stage of evolution. This is Homo Deus.””

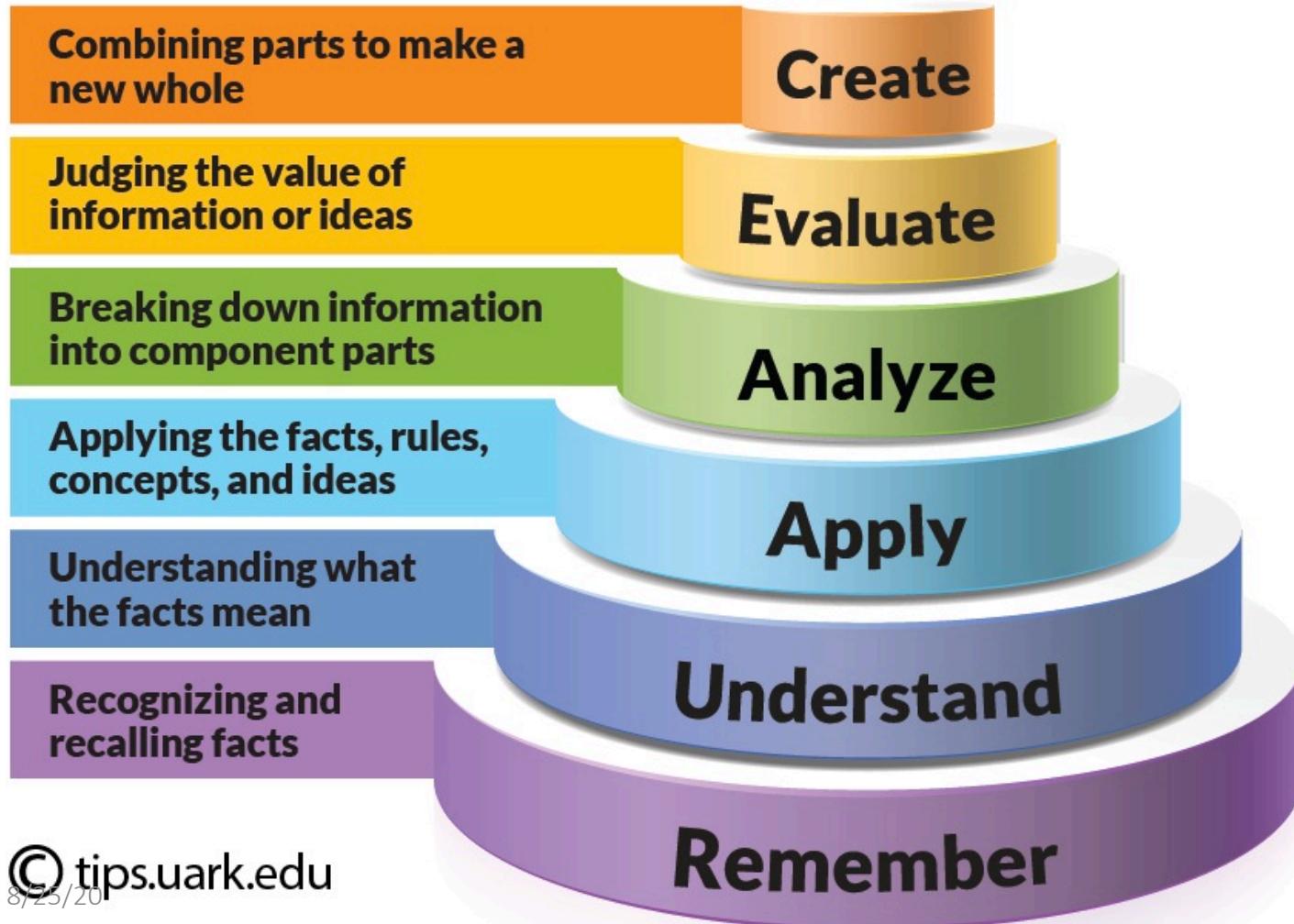
Thank You



Roadmap

- Course Logistics
- History and Now
- A Rough Plan of Course Content

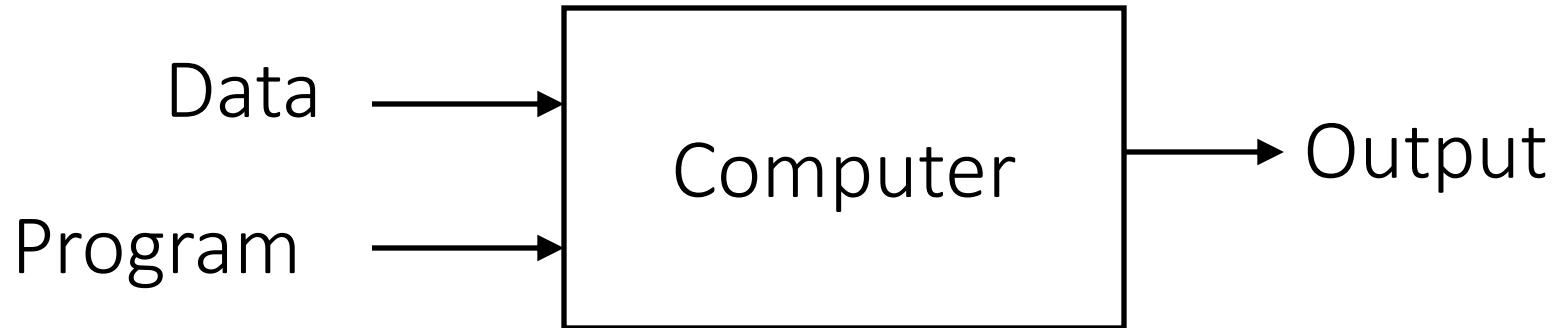
My Teaching Guide: Bloom's Taxonomy on Cognitive Learning



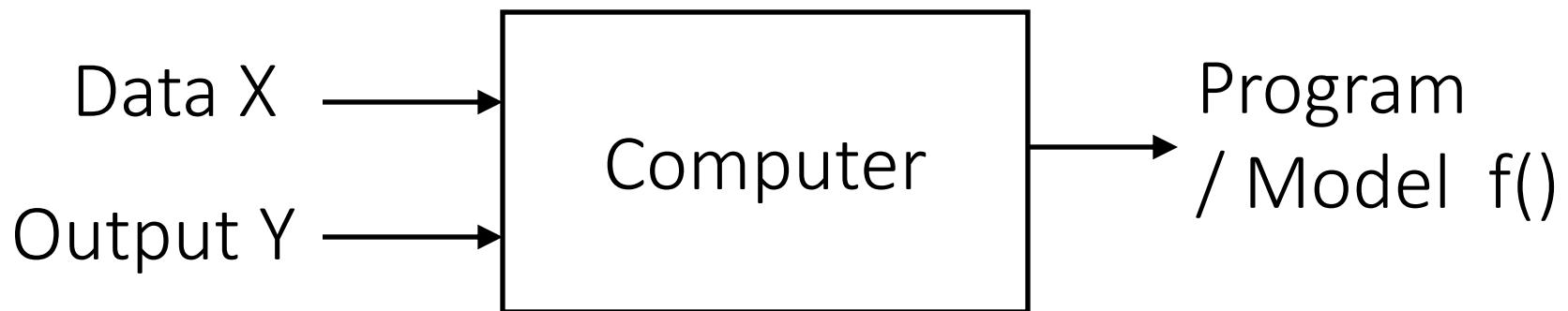
BASICS OF MACHINE LEARNING

- “The goal of machine learning is to build computer systems that can **learn and adapt from their experience.**” – Tom Dietterich
- “**Experience**” in the form of available **data examples** (also called as instances, samples)
- Available examples are described with properties (**data points in feature space X**)

Traditional Programming



Machine Learning (training phase)



e.g. SUPERVISED LEARNING

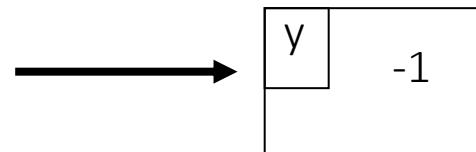
- Find function to map **input** space X to **output** space Y

$$f : X \longrightarrow Y$$

- So that the **difference** between y and $f(x)$ of each example x is small.

e.g.

x	I believe that this book is not at all helpful since it does not explain thoroughly the material . it just provides the reader with tables and calculations that sometimes are not easily understood ...
---	--

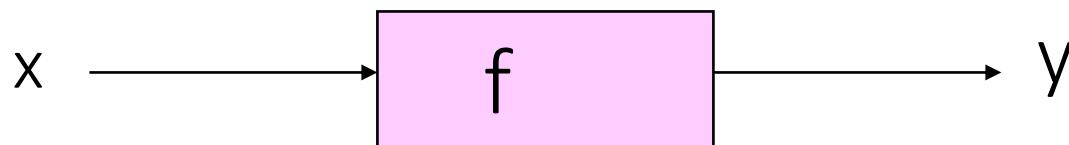


Output Y: {1 / Yes , -1 / No }
e.g. Is this a positive product review ?

Input X : e.g. a piece of English text

SUPERVISED Linear Binary Classifier

- Now let us check out a **VERY SIMPLE** case of

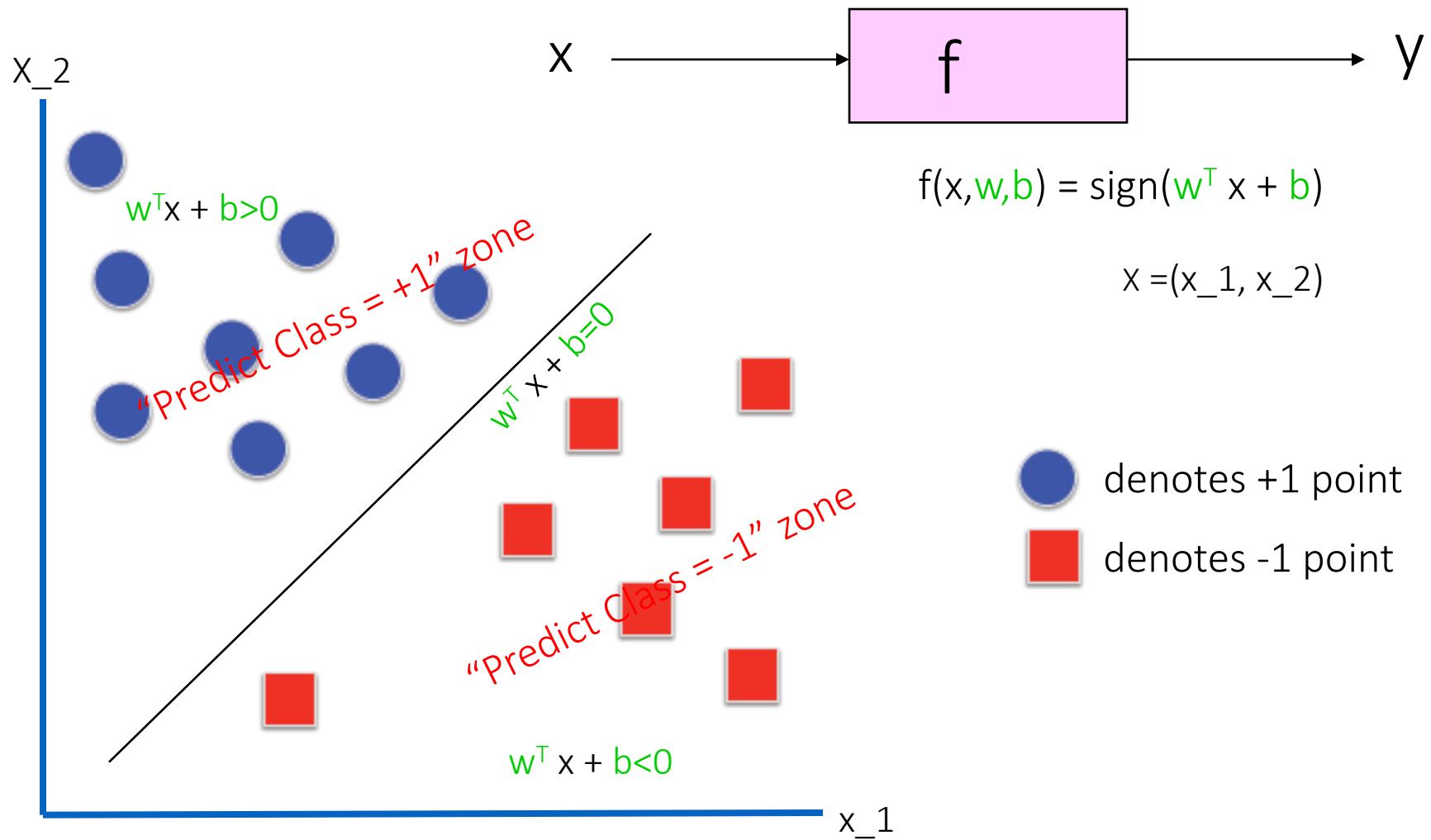


e.g.: Binary y / Linear f / X as \mathbb{R}^2

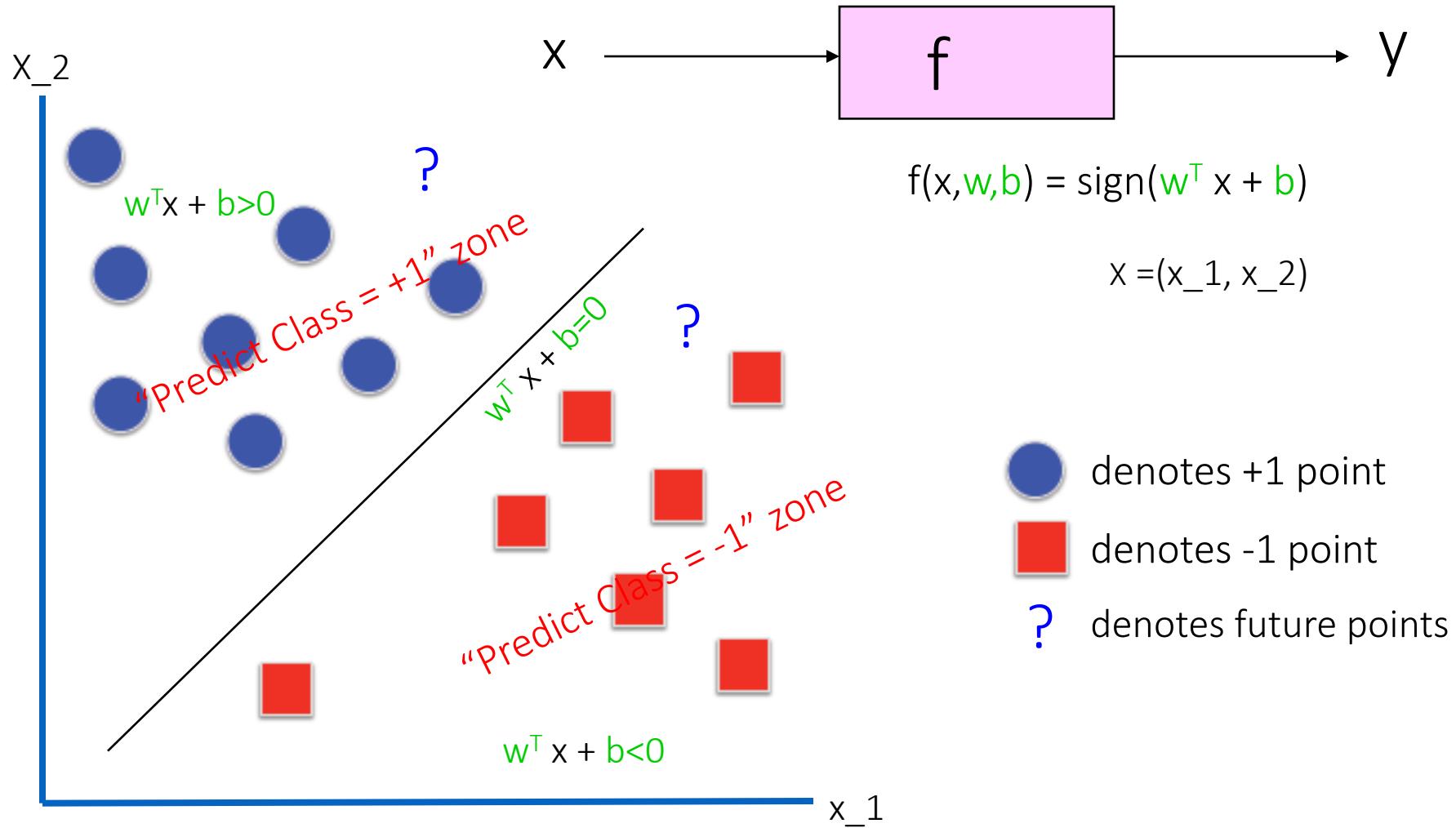
$$f(x, w, b) = \text{sign}(w^T x + b)$$

$$x = (x_1, x_2)$$

SUPERVISED Linear Binary Classifier

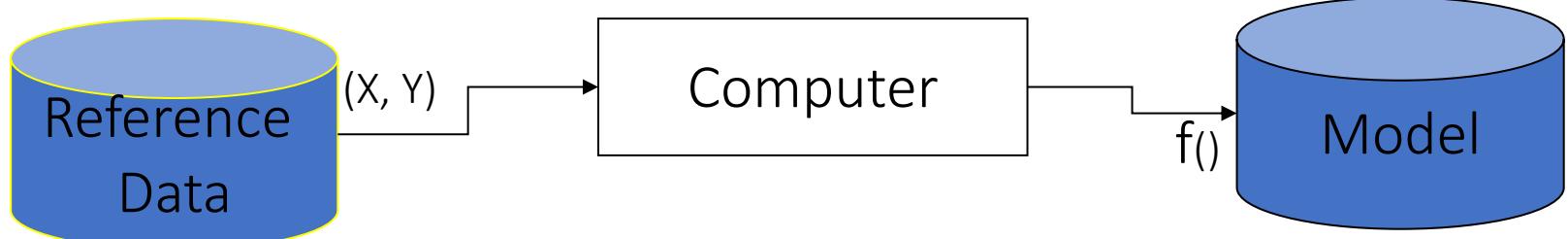


SUPERVISED Linear Binary Classifier

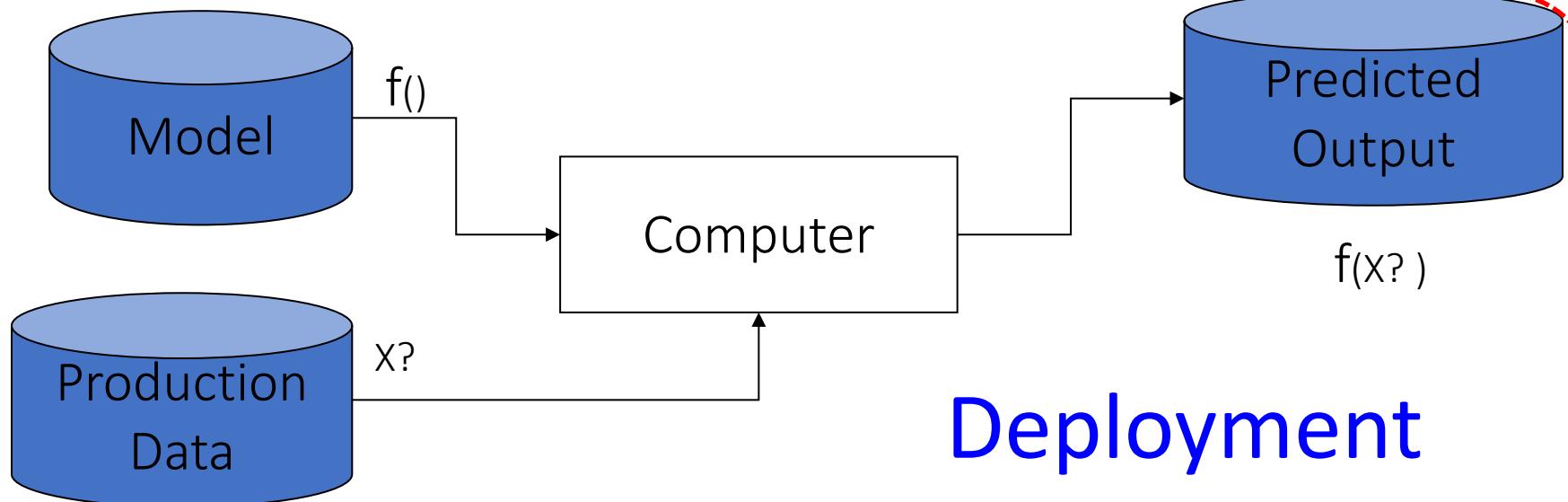


Two Modes of Machine Learning

Consists of **input-output** pairs

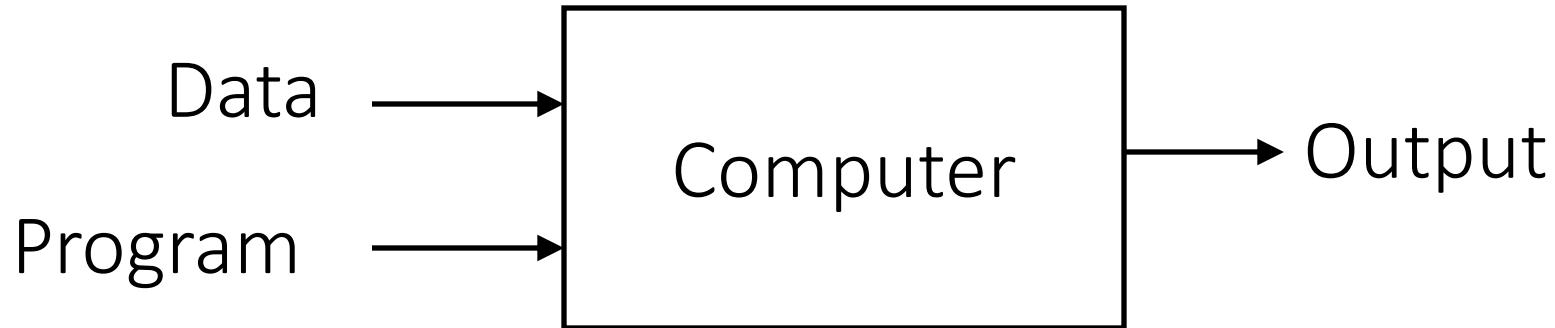


Training

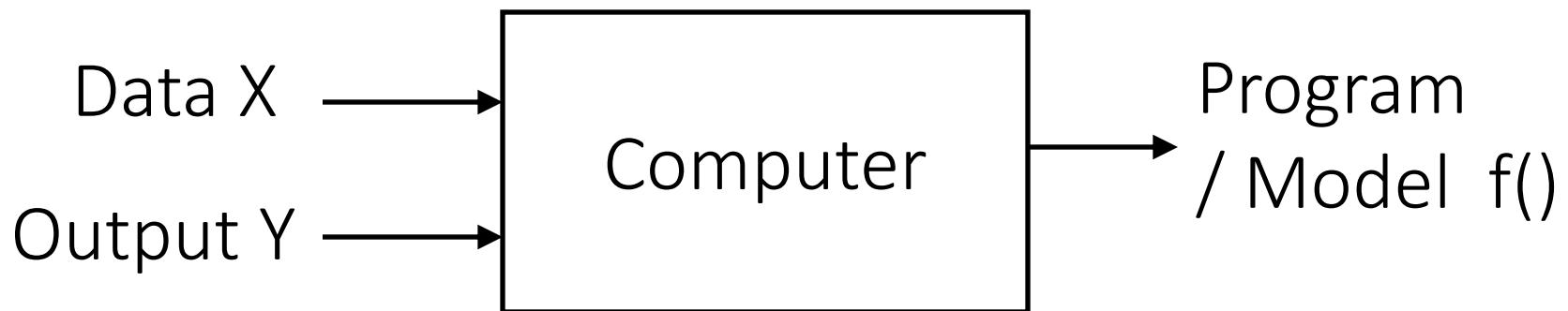


Deployment

Traditional Programming



Machine Learning (training phase)



Basic Concepts

- Training (i.e. learning parameters w, b)
 - Training set includes
 - available examples x_1, \dots, x_L
 - available corresponding labels y_1, \dots, y_L
 - Find (w, b) by minimizing loss / Cost function $L()$
 - (i.e. difference between y and $f(x)$ on available examples in training set)

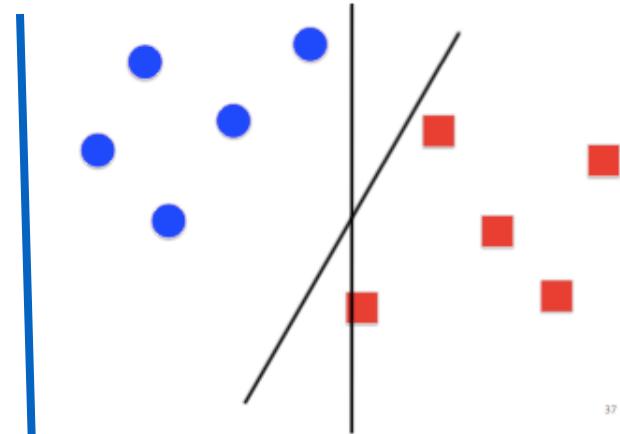
$$(W, b) = \operatorname{argmin}_{W, b} \sum_{i=1}^L \ell(f(x_i), y_i)$$

Basic Concepts

- Loss function

- e.g. hinge loss for binary classification task

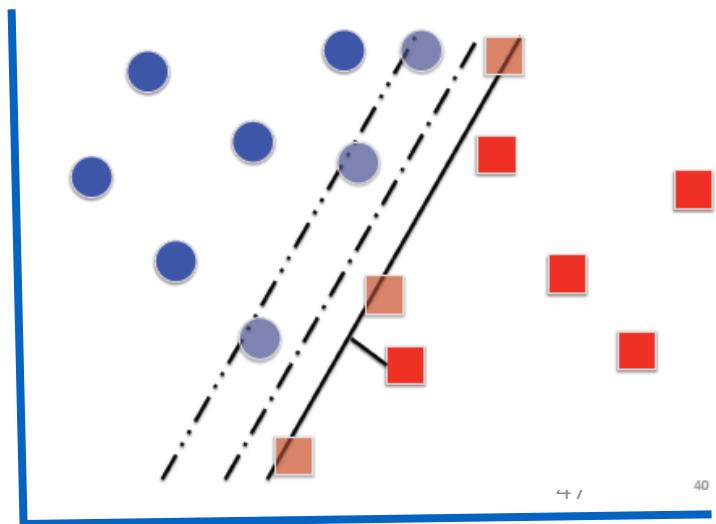
$$\sum_{i=1}^L \ell(f(x_i), y_i) = \sum_{i=1}^L \max(0, 1 - y_i f(x_i)).$$



- Regularization

- E.g. additional information added on loss function to control f

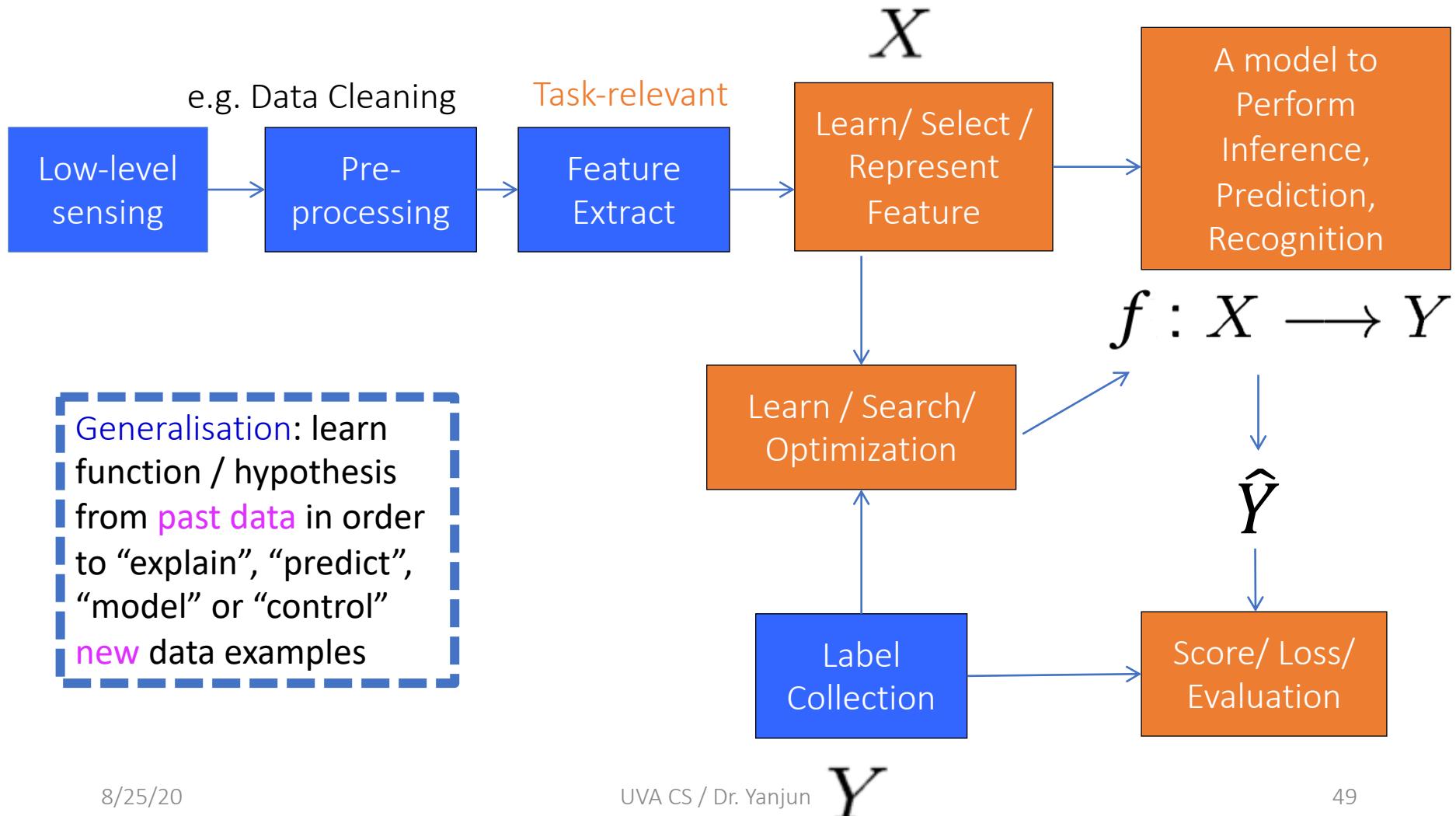
$$C \sum_{i=1}^L \ell(f(x_i), y_i) + \frac{1}{2} \|w\|^2,$$



Basic Concepts

- Testing (i.e. evaluating performance on “future” points)
 - Difference between true $y_?$ and the predicted $f(x_?)$ on a set of testing examples (i.e. testing set)
 - Key: example $x_?$ not in the training set
- Generalisation: learn function / hypothesis from past data in order to “explain”, “predict”, “model” or “control” new data examples

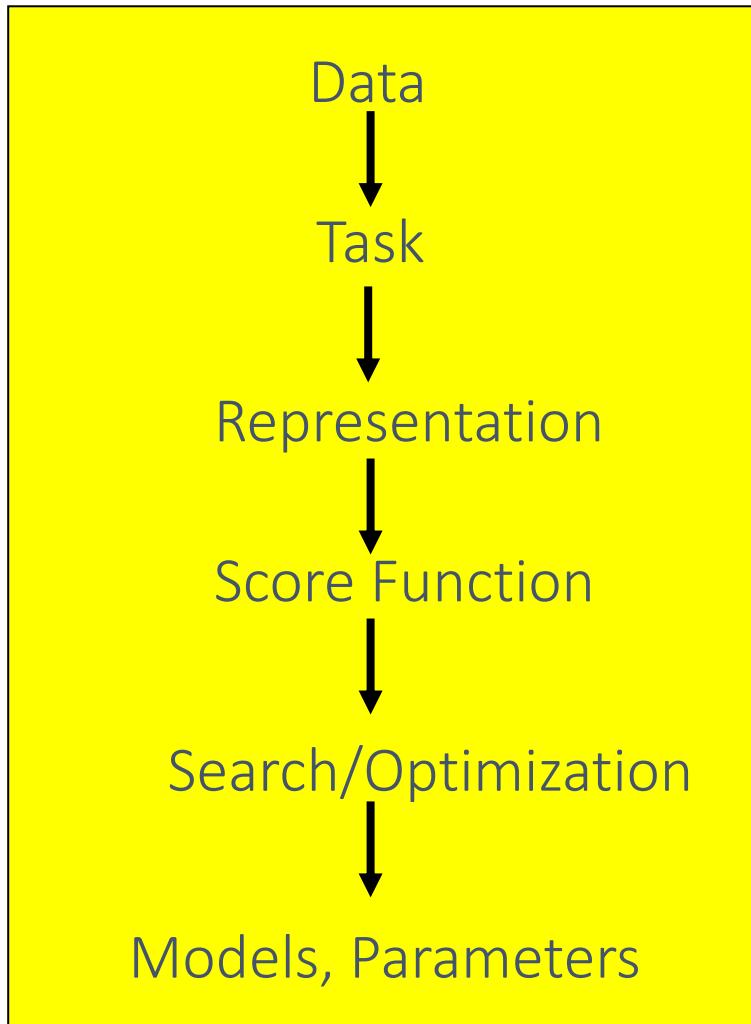
A Typical Machine Learning Application's Pipeline



When to use Machine Learning (Adapt to / learn from data) ?

- 1. Extract knowledge from data
 - Relationships and correlations can be hidden within large amounts of data
 - The amount of knowledge available about certain tasks is simply too large for explicit encoding (e.g. rules) by humans
- 2. Learn tasks that are difficult to formalise
 - Hard to be defined well, except by examples, e.g., face recognition
- 3. Create software that improves over time
 - New knowledge is constantly being discovered.
 - Rule or human encoding-based system is difficult to continuously re-design “by hand”.

Machine Learning in a Nutshell



ML grew out of work in AI

Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

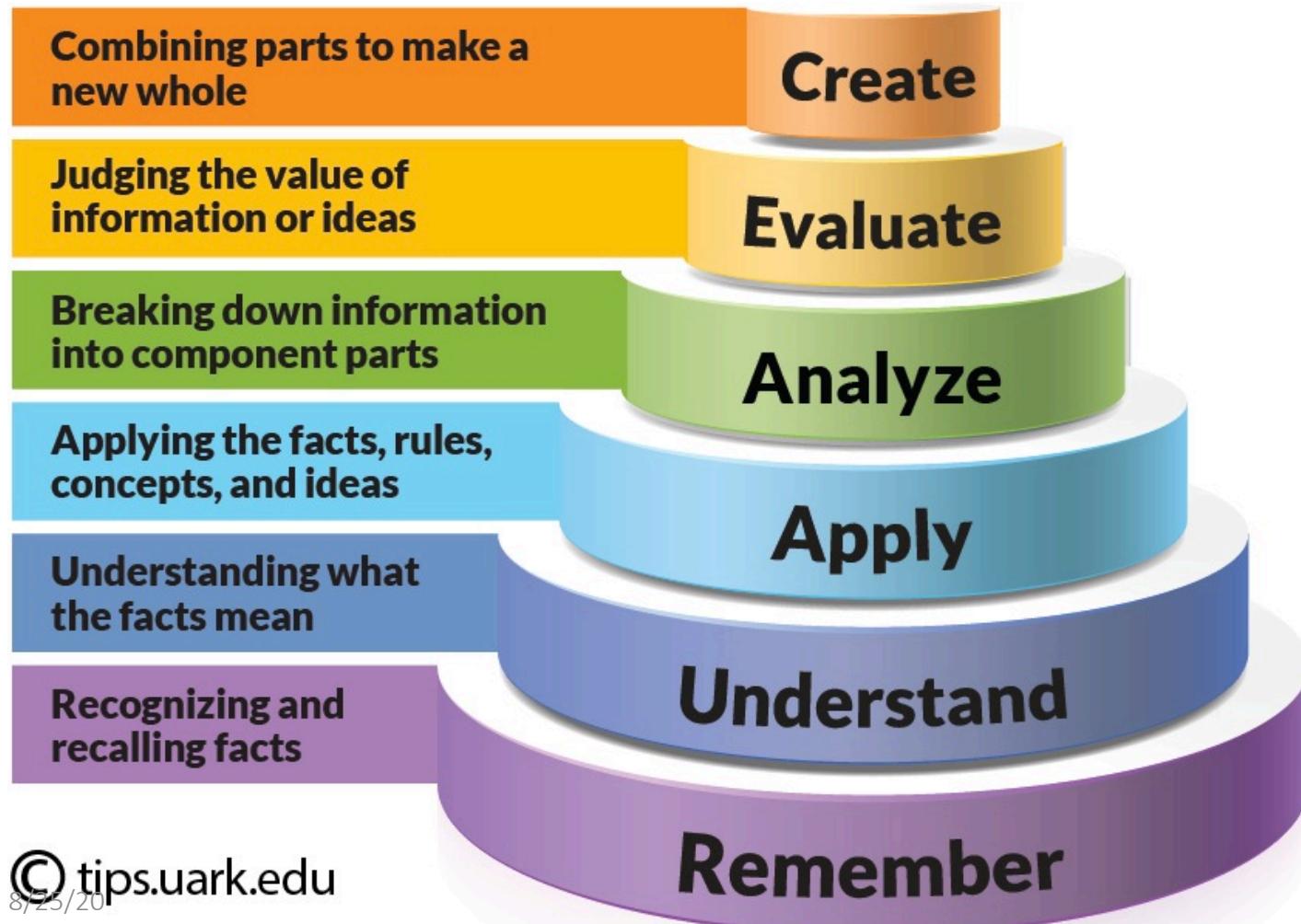
What we have covered

Data	
Task	
Representation	
Score Function	
Search/Optimization	
Models, Parameters	
8/25/20	

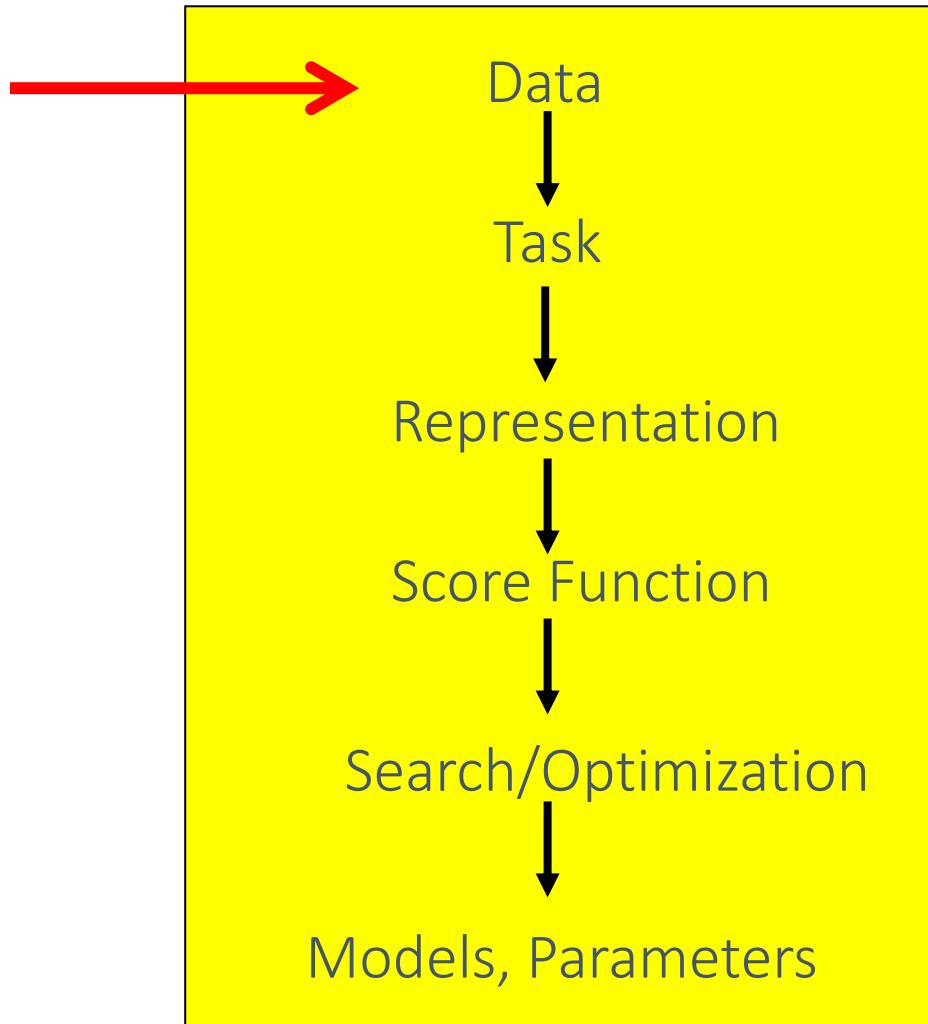
What we will cover

Data	Tabular, 1-D sequential, 2-D Grid like Imaging, 3-D VR, Graph, Set
Task	Regression, classification, clustering, dimen-reduction
Representation	Linear func, nonlinear function (e.g. polynomial expansion), local linear, logistic function (e.g. $p(c x)$), tree, multi-layer, prob-density family (e.g. Bernoulli, multinomial, Gaussian, mixture of Gaussians), local func smoothness, kernel matrix, local smoothness, partition of feature space,
Score Function	MSE, Margin, log-likelihood, EPE (e.g. L2 loss for KNN, 0-1 loss for Bayes classifier), cross-entropy, cluster points distance to centers, variance, conditional log-likelihood, complete data-likelihood, regularized loss func (e.g. L1, L2) , goodness of inter-cluster similar
Search/ Optimization	Normal equation, gradient descent, stochastic GD, Newton, Linear programming, Quadratic programming (quadratic objective with linear constraints), greedy, EM, asyn-SGD, eigenDecomp, backprop
Models, Parameters	Linear weight vector, basis weight vector, local weight vector, dual weights, training samples, tree-dendrogram, multi-layer weights, principle components, member (soft/hard) assignment, cluster centroid, cluster covariance (shape), ...
8/25/20	

My Teaching Guide: Bloom's Taxonomy on Cognitive Learning



Machine Learning in a Nutshell



ML grew out of work in AI

Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

	X_1	X_2	X_3	Y
s_1				
s_2				
s_3				
s_4				
s_5				
s_6				

$$f : \boxed{X} \longrightarrow \boxed{Y}$$

- Data/points/instances/examples/samples/records: [rows]
- Features/attributes/dimensions/independent variables/covariates/predictors/regressors: [columns, except the last]
- Target/outcome/response/label/dependent variable: special column to be predicted [last column]

Main Types of Columns

	X_1	X_2	X_3	Y
s_1				
s_2				
s_3				
s_4				
s_5				
s_6				

- *Continuous*: a real number, for example, weight
- *Discrete*: a symbol, like “Good” or “Bad”

training dataset


$$\mathbf{X}_{train} = \begin{bmatrix} \text{---} & \mathbf{x}_1^T & \text{---} \\ \text{---} & \mathbf{x}_2^T & \text{---} \\ \vdots & \vdots & \vdots \\ \text{---} & \mathbf{x}_n^T & \text{---} \end{bmatrix}$$

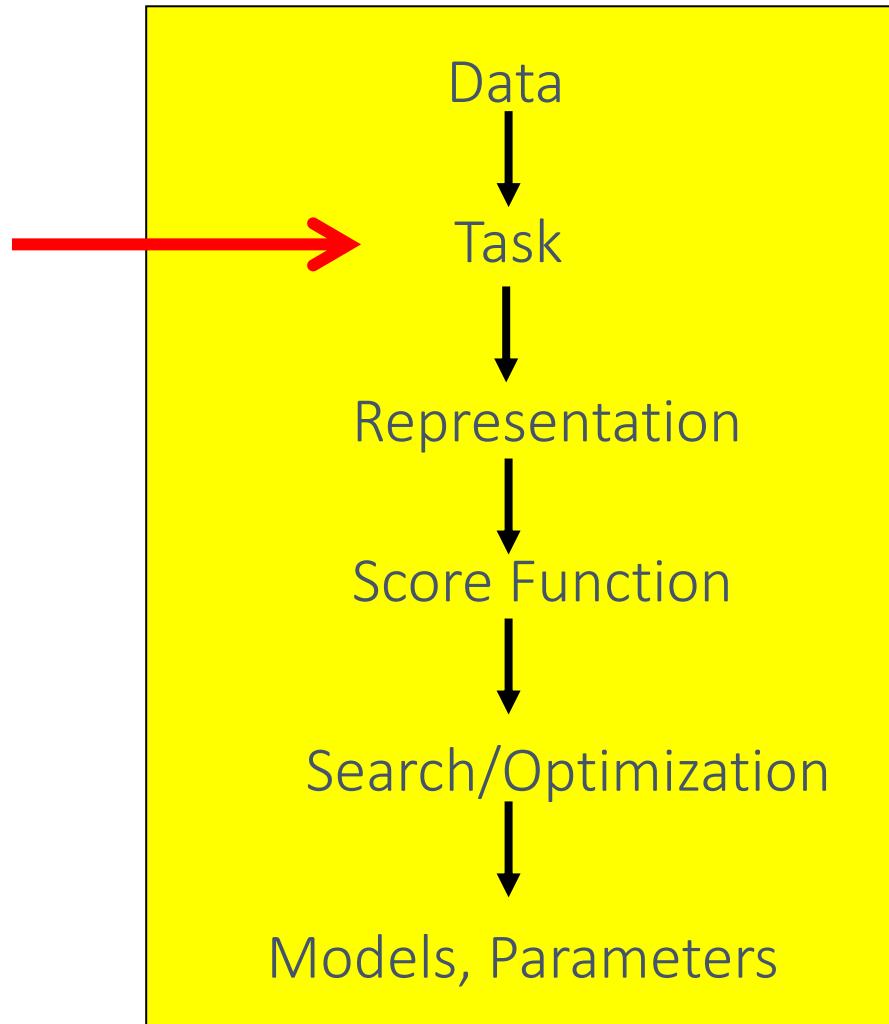
$$\vec{y}_{train} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

test dataset


$$\mathbf{X}_{test} = \begin{bmatrix} \text{---} & \mathbf{x}_{n+1}^T & \text{---} \\ \text{---} & \mathbf{x}_{n+2}^T & \text{---} \\ \vdots & \vdots & \vdots \\ \text{---} & \mathbf{x}_{n+m}^T & \text{---} \end{bmatrix}$$

$$\vec{y}_{test} = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ \vdots \\ y_{n+m} \end{bmatrix}$$

Machine Learning in a Nutshell

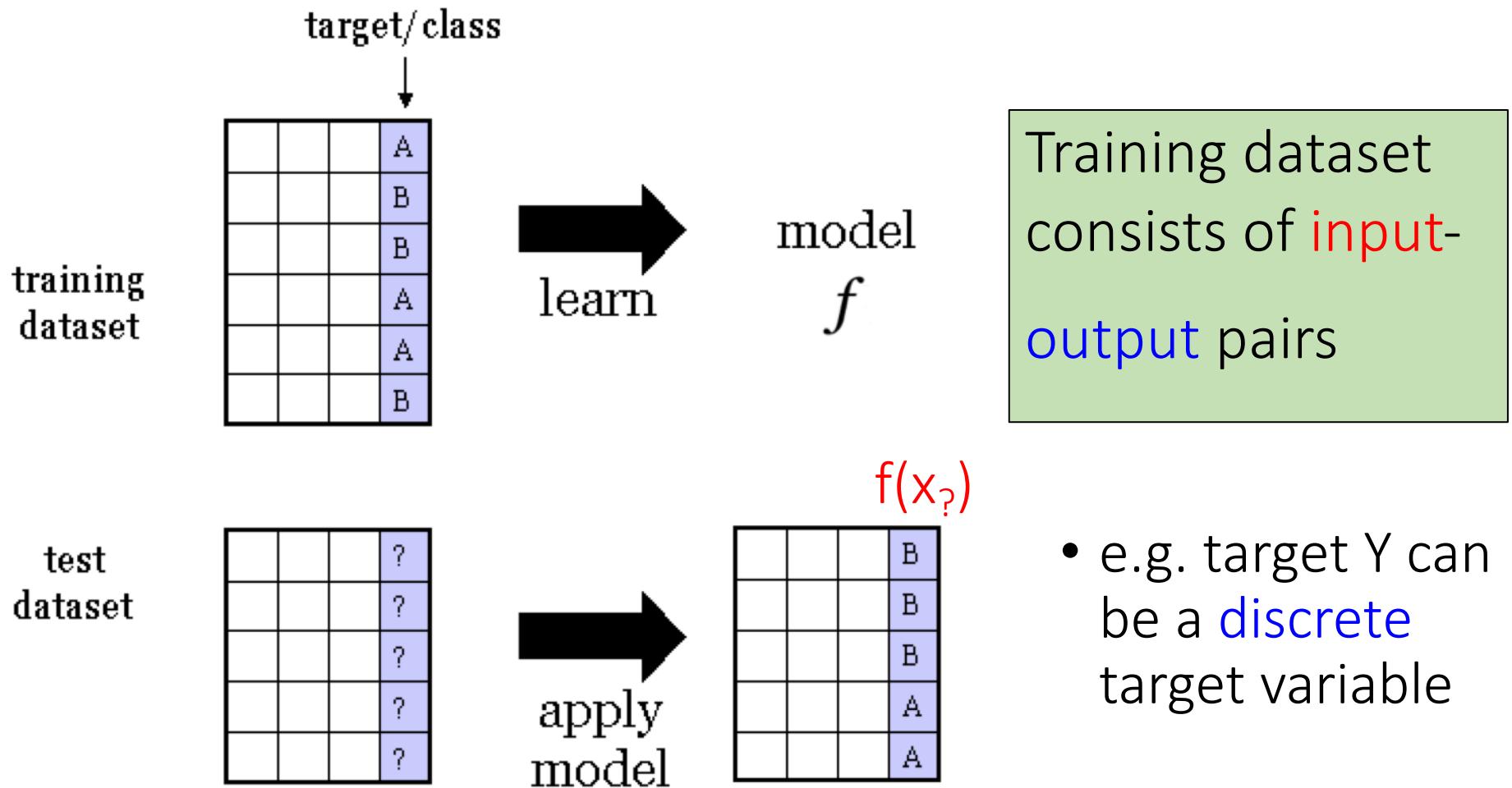


ML grew out of
work in AI

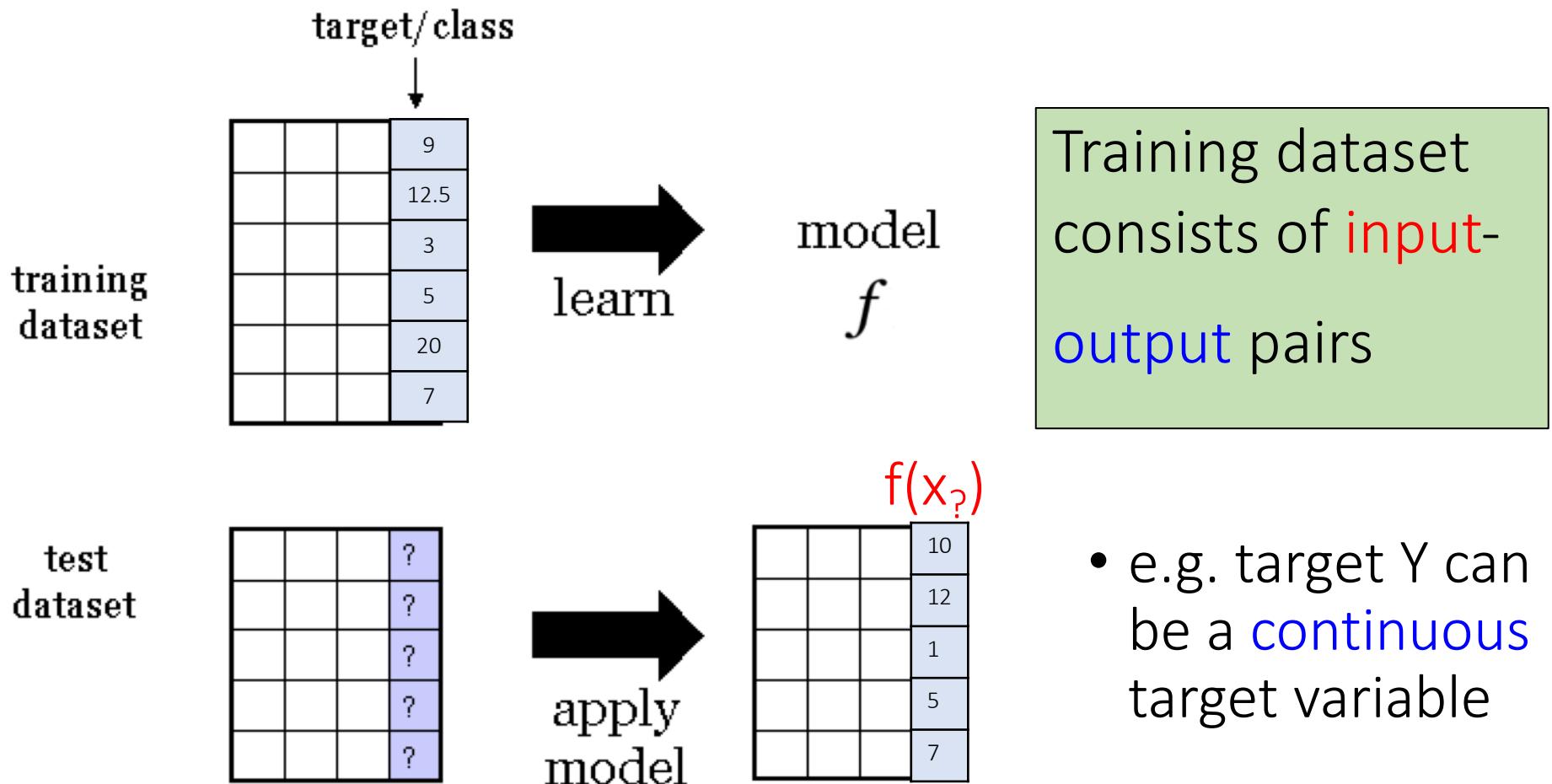
Optimize a
performance criterion
using example data or
past experience,

Aiming to generalize to
unseen data

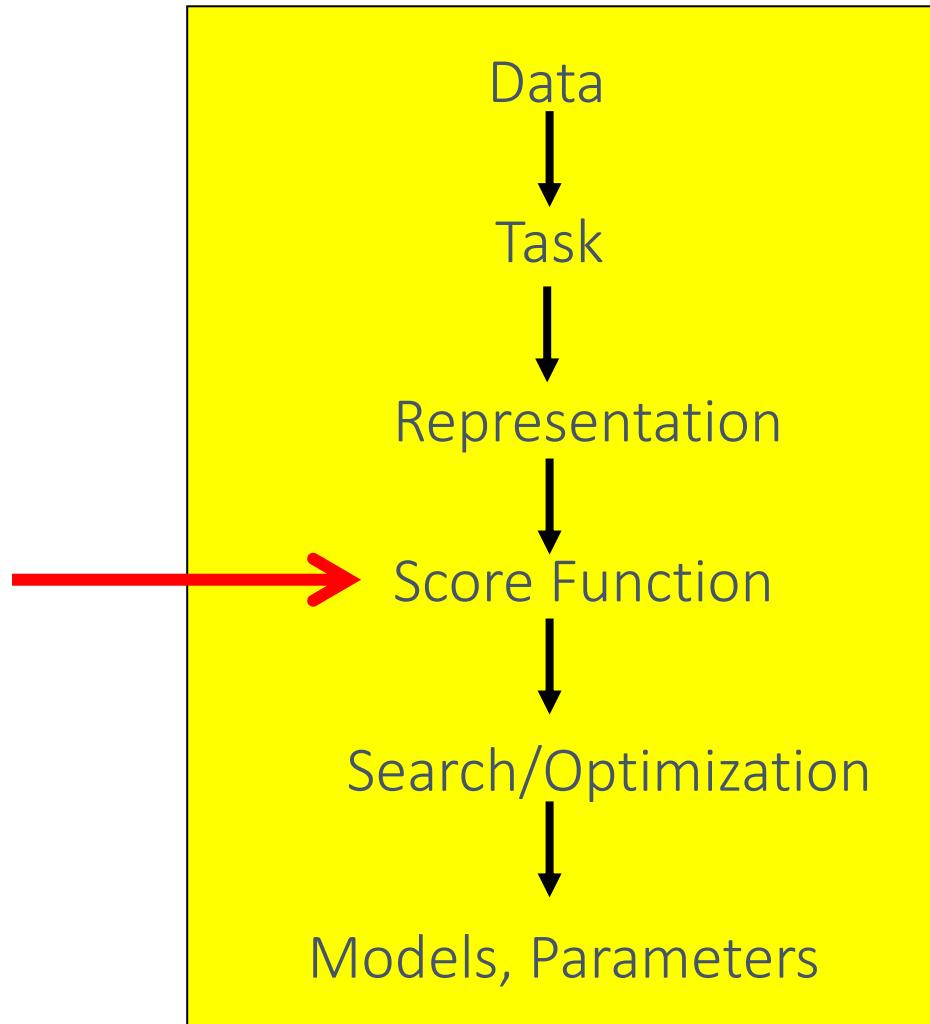
e.g. SUPERVISED Classification



e.g. SUPERVISED Regression



Machine Learning in a Nutshell



ML grew out of
work in AI

Optimize a
performance criterion
using example data or
past experience,

Aiming to generalize to
unseen data

Basic Concepts

- Training (i.e. learning parameters w, b)
 - Training set includes
 - available examples x_1, \dots, x_L
 - available corresponding labels y_1, \dots, y_L
 - Find (w, b) by minimizing loss
 - (i.e. difference between y and $f(x)$ on available examples in training set)

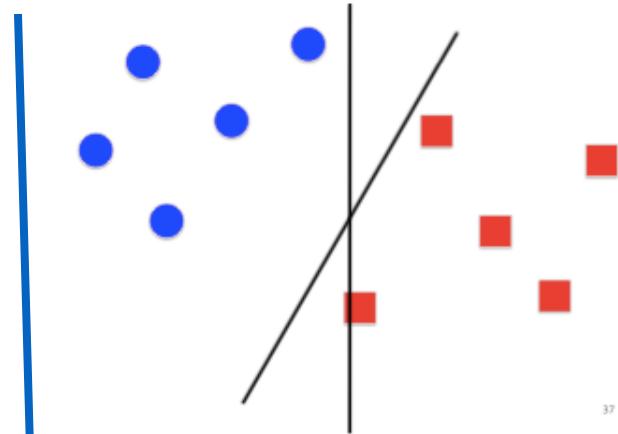
$$(W, b) = \operatorname{argmin}_{W, b} \sum_{i=1}^L \ell(f(x_i), y_i)$$

Basic Concepts

- Loss function

- e.g. hinge loss for binary classification task

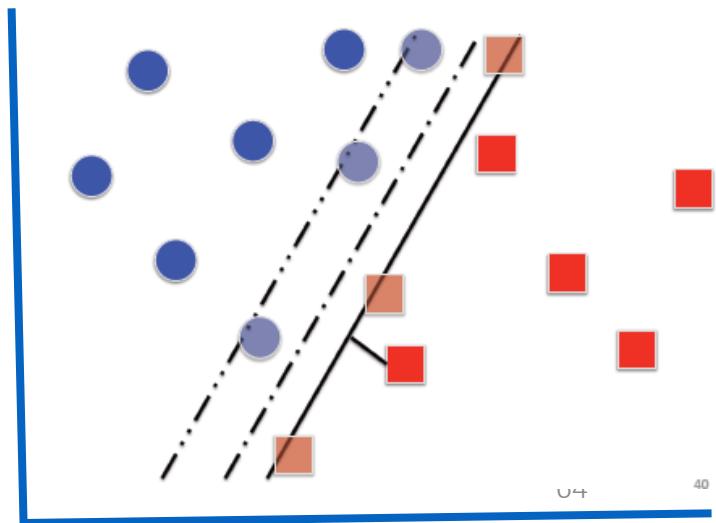
$$\sum_{i=1}^L \ell(f(x_i), y_i) = \sum_{i=1}^L \max(0, 1 - y_i f(x_i)).$$



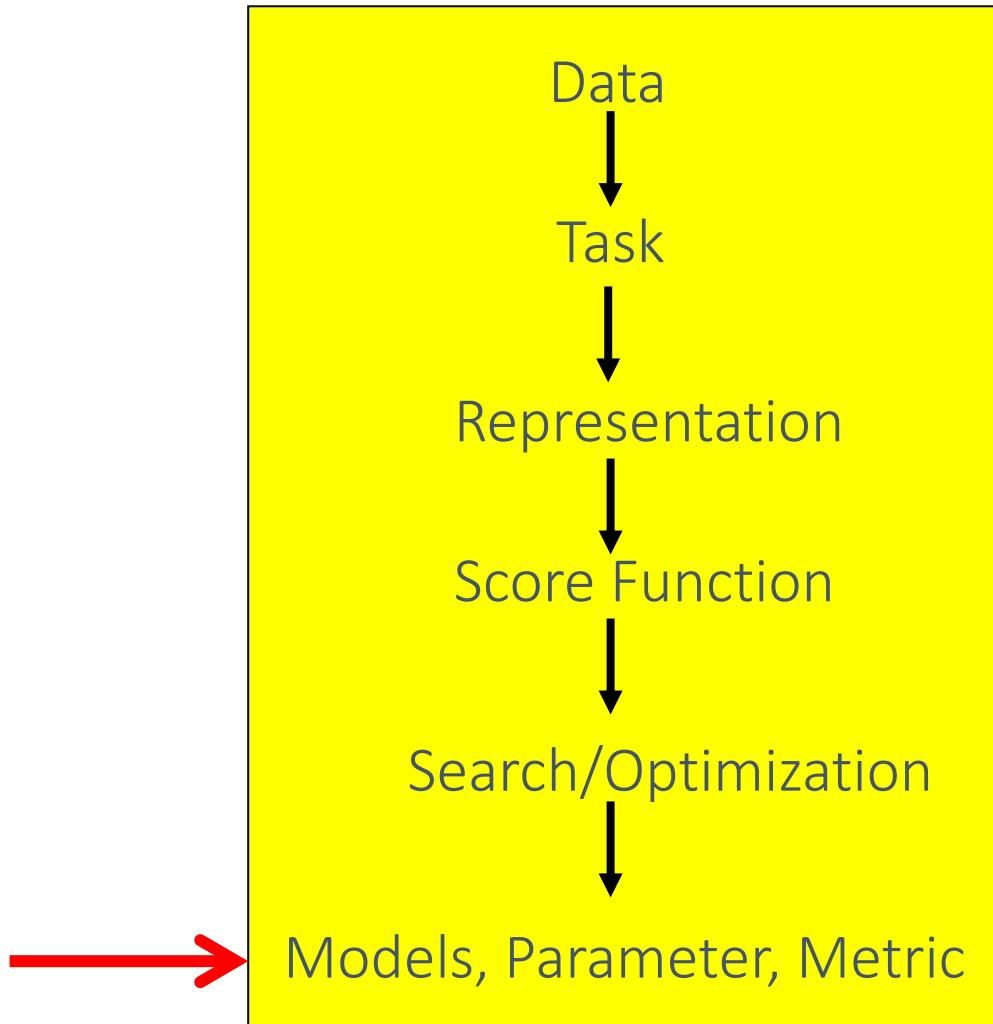
- Regularization

- E.g. additional information added on loss function to control f

$$C \sum_{i=1}^L \ell(f(x_i), y_i) + \frac{1}{2} \|w\|^2,$$



Machine Learning in a Nutshell

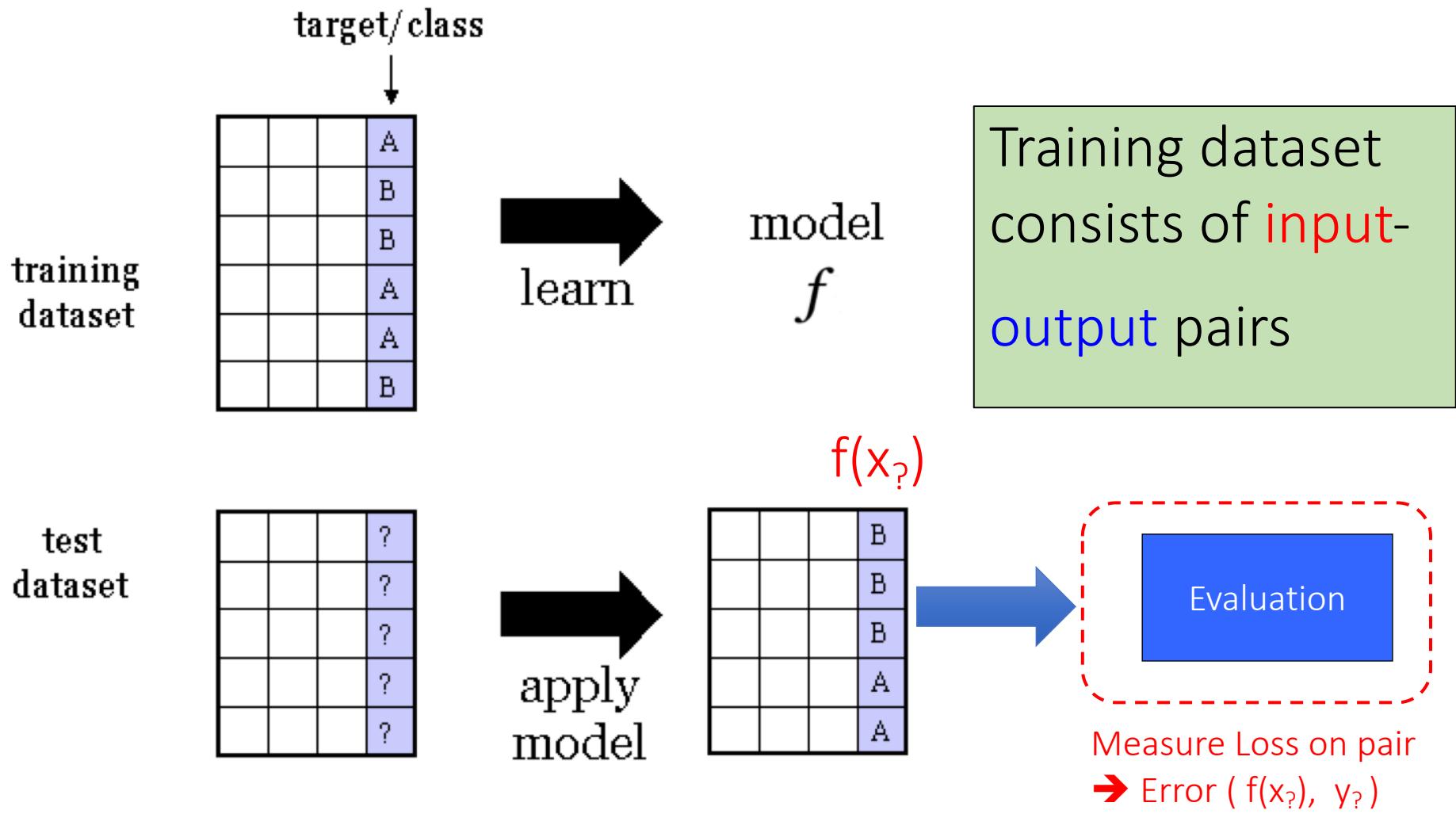


ML grew out of work in AI

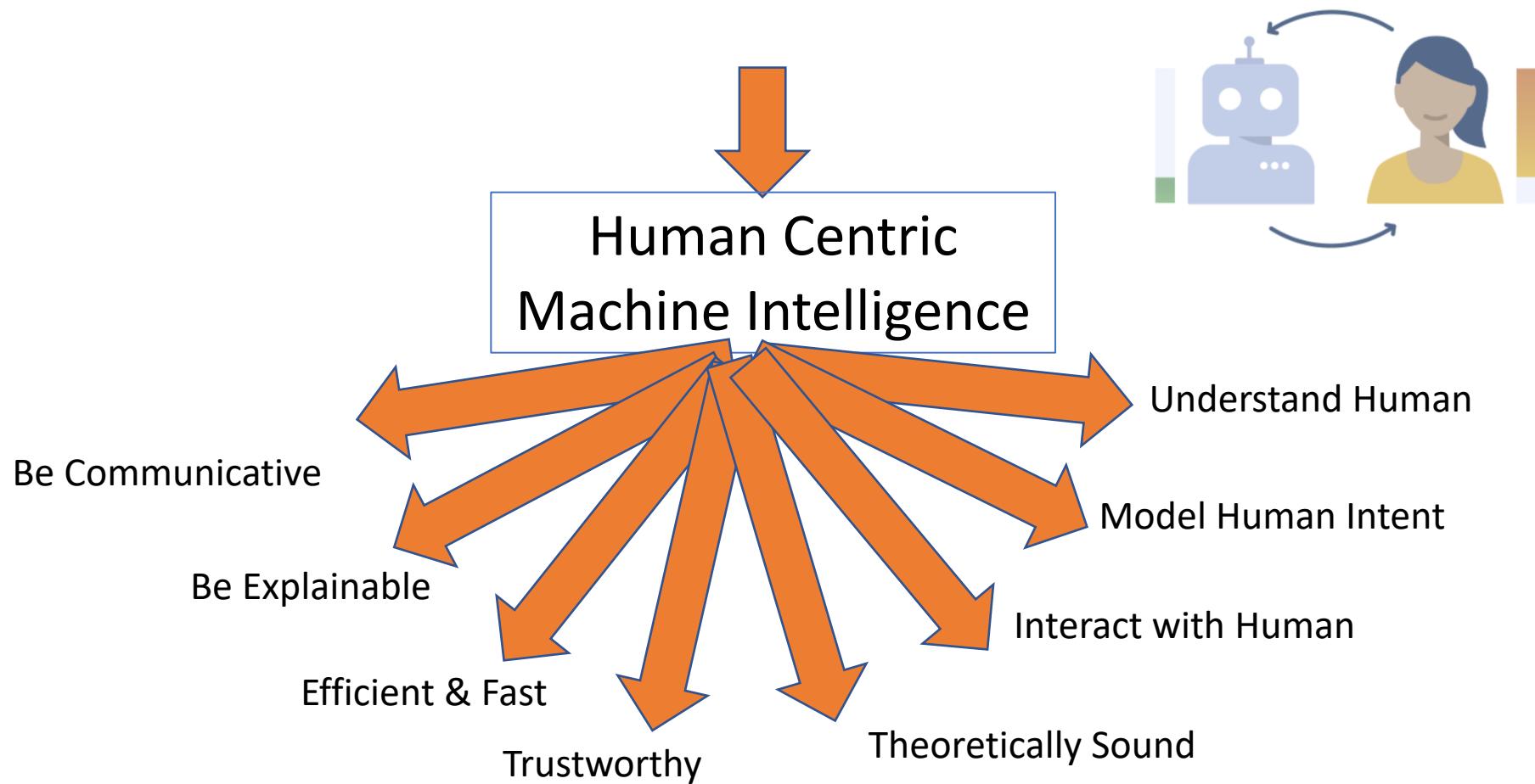
Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

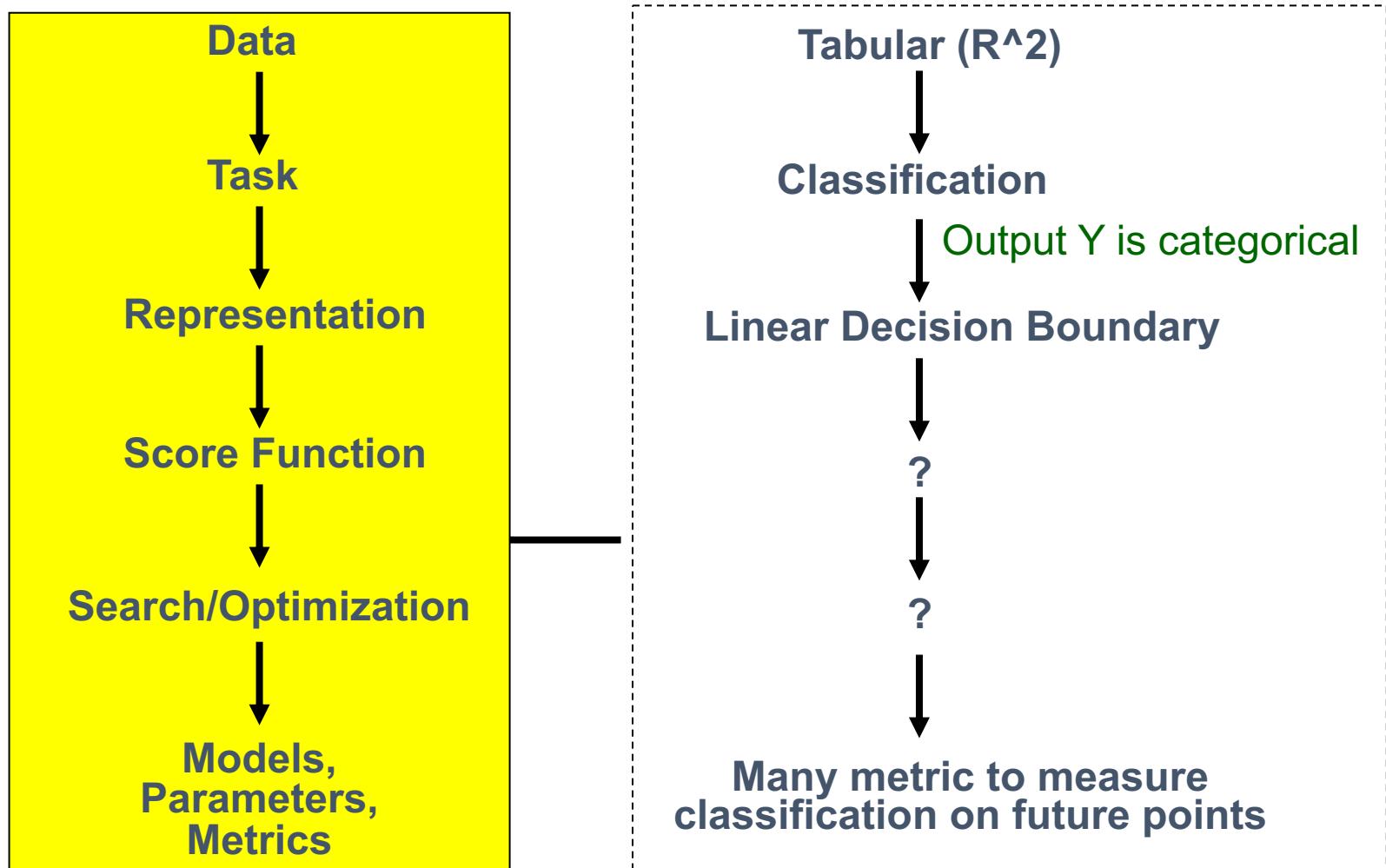
How to know the program works well: Measure Prediction Accuracy on Test Data



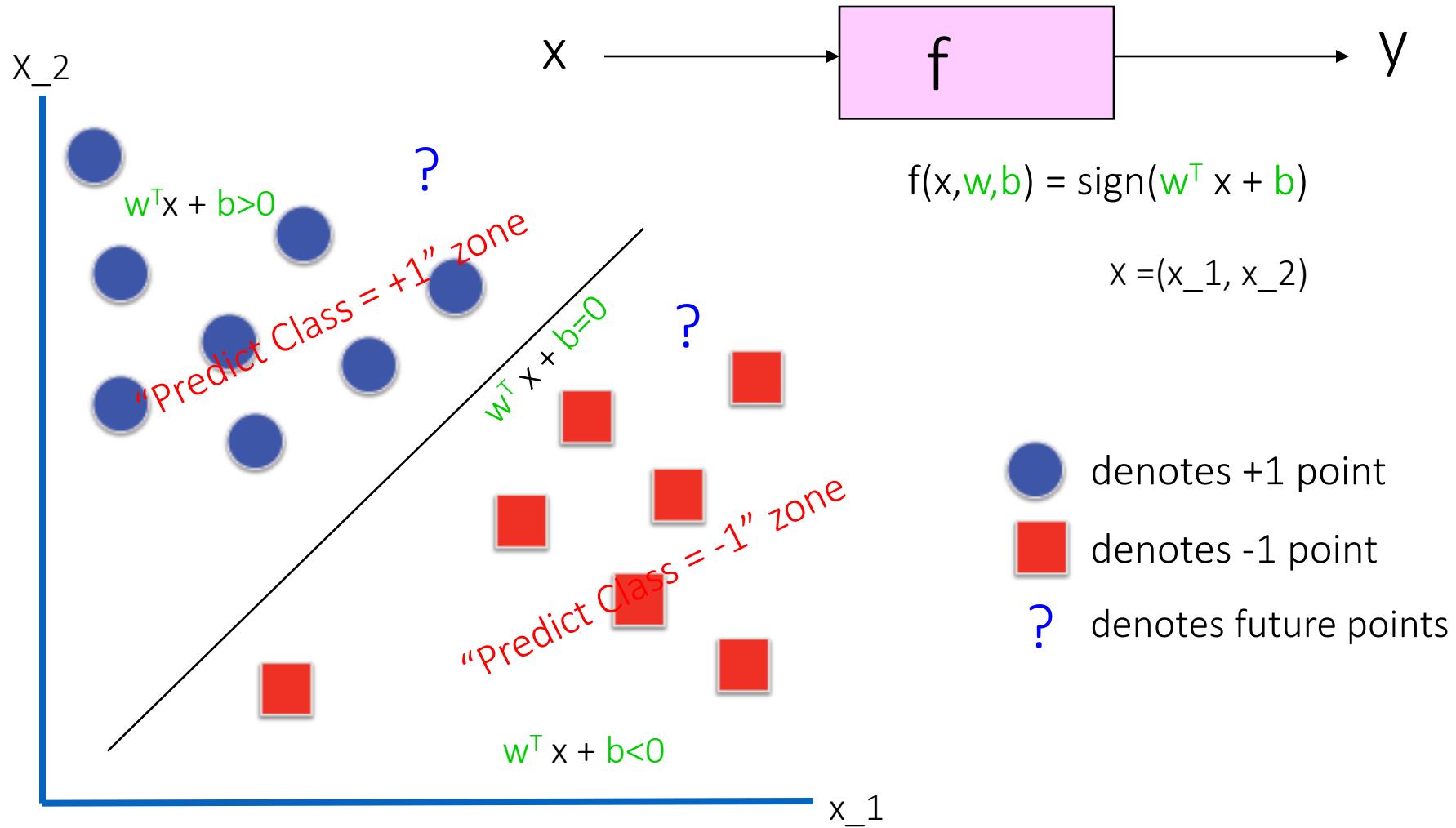
More Metrics: e.g., ML and AI Research @ UVA CS



Nutshell for the simple Linear Supervised Classifier



SUPERVISED Linear Binary Classifier



Rough Sectioning of this Course

- 1. Basic Supervised Regression + Tabular Data
- 2. Basic Deep Learning + 2D Imaging Data
- 3. Advanced Supervised learning + Tabular Data
- 4. Generative and Deep + 1D Sequence Text Data
- 5. Not Supervised

Section 1 - Basics & On Tabular Input Type								
Extra	Extra		Algebra Review	Slides: S1-AlgbReview		Notes: S1-linalg-extra.pdf	video	Useful Math
Platform	Platform		Basics scikit-learn	Slides: S1-scikit-learn	library			TA survey scikit-learn
0827	W1		Linear Regression	Slides: S1-LinearReg	tabular		video	LR more + ELS Ch3.2
0901	W2	Q1	Optimization for LR	Slides: S1-LR optimization	tabular		video	Useful SGD
0903	W2		Nonlinear Regression	Slides: S1-lrExtend SelectModel	tabular	Notes: S1-nonparametricR	video	NonLinear + API + ELS Ch5
0908	W3	Q2	Linear Prediction with Regularization	Slides: S1-lr Regularized	tabular		video	More Ridge
Extra	Extra		Lasso and Elastic Net	Slides: S1-lr Sparse	tabular	Notes: S1-Extra-lrReguOpm	video	Elastic paper
0910	W3		supervised classification	Slides: S1-Classify			video	Error Metrics
0915	W4	Q3	KNN and Theory	Slides: S1-kNearestN	tabular	Notes: S1-KNN-extra	video	Useful BiasVar
0917	W4		Bias Variance Tradeoff	Slides: S1-biasVariance		Notes: S1-Tibshirani-modelbasics	video	ESL Ch7
8/27/20	Platform		machine learning in the cloud	Slides: S1-CloudML	library		video	Invited Speaker

Section 2 - Deep and 2D Grid Type (e.g. Imaging)

0922	W5	Q4	ProbReview + MLE	Slides: S2-ProbReview		Notes: S2-MLE	video	MLE / MLE code
0924	W5		Logistic and NN	Slides: S2-LogisticReg	structured	Notes: L14extra-Logistic	video	code + compare classifiers
0929	W6	Q5	NN and Deep Learning	Slides: S2-basicDeepNN	structured	Notes: L15-lecun-98b	MLP video	
1001	W6		CNN	Slides: S2-CovNN	2d(vision)	Notes: L16-PCA	video	CNN
Extra	Extra		Quick survey of recent deep learning	Slides: S2-deepSurvey	structured	Notes: L16-PCA	video	DNN Cheatsheets
1006	W7	Q6	PCA, TSNE, UMAP	Slides: S2-PCA-VAE	tabular	Notes: L07feaSelc		DNN Cheatsheets
Extra	Extra		semi-supervised	Slides: S2-semi-self-learning				
Extra	Extra		auto differentiation	Slides: S2-auto-grad	library			
Platform	Platform		S2 L09 Ta Pytorch	Slides: S2-pytorch	library			TA survey pytorch

Section 3 - More Advanced Supervsied on Tabular Type

1008	W7		SVM	Slides: S3-SVM-basic	tabular	Notes: L11-LibSVMGuide	video	More SVM
1015	W8	Q7	SVM, Kernel	Slides: S3-SVM-kernel		Notes: L11-LibSVMGuide	video	VC Theory
Extra	Extra		SVM, Dual	Slides: S3-SVM-optimDual		Notes: L11Extra-SVMoptimDual	video	SMO + Paper SMO
1020	W9		DecisionTree and Bagging	Slides: S3-DecisionTree	tabular	Notes: L22-review	video	xgboost

Section 4 - on 1D Sequence Type (e.g. Language Text)

1027	W10		Generative Classification	Slides: S4-BayesClassify		Notes: L16-PCA	video	
1029	W10	Q9	Gaussian BC	Slides: S4-GenerDiscr		Notes: L17c-NBCtext	video	Paper Discr vs. Genera
1103	W11		NaiveBC on Text	Slides: S5-NBCtext	1D(Text)	Notes: L20-review	video	Multinomial MLE
1105	W11	Q10	Recent deep learning on Text	Slides: S4-deepText	1D(Text)	Notes: L16-PCA	video	DNN Cheatsheets
1110	W12		Learning to Recommend on Text	Slides: S4-deepRecommend	1D(Text)		video	
Extra	Extra		probabilistic programming	Slides: S2-prob-program				
Extra	Extra		Feature Selection and Model Selection	Slides: S4feaSelc		Notes: L07-FeatureSelect-jmlrPaper	video	API + ELS Ch3.4 and Ch3.3

Section 5 - Not Supervised

1112	W12	Q11	Clustering Hier	Slides: S5-clustering-Hier	tabular	Notes: L19c-clustering3-GMM	video	compare Hier clusterings
1117	W13		Clustering Partition	Slides: S5-clustering-kMeans	tabular	Notes: L19d-EMextra-EM	video	compare clusterings
Extra	Extra		Clustering GMM	Slides: S5-clustering-GMM	1DSignal(Audio)	Notes: L19d-EMextra-EM	video	EM primer
8/25/20 1119	W13	Q12	RL	Slides: S5-reinforcement-learning	None-IID	Notes: L18c-More-Boosting	video	xgboost

Thank You



References

- Prof. Andrew Moore's tutorials
- Prof. Raymond J. Mooney's slides
- Prof. Alexander Gray's slides
- Prof. Eric Xing's slides
- <http://scikit-learn.org/>
- Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.
- Prof. M.A. Papalaskar's slides