AI6102:
L1: Introduction

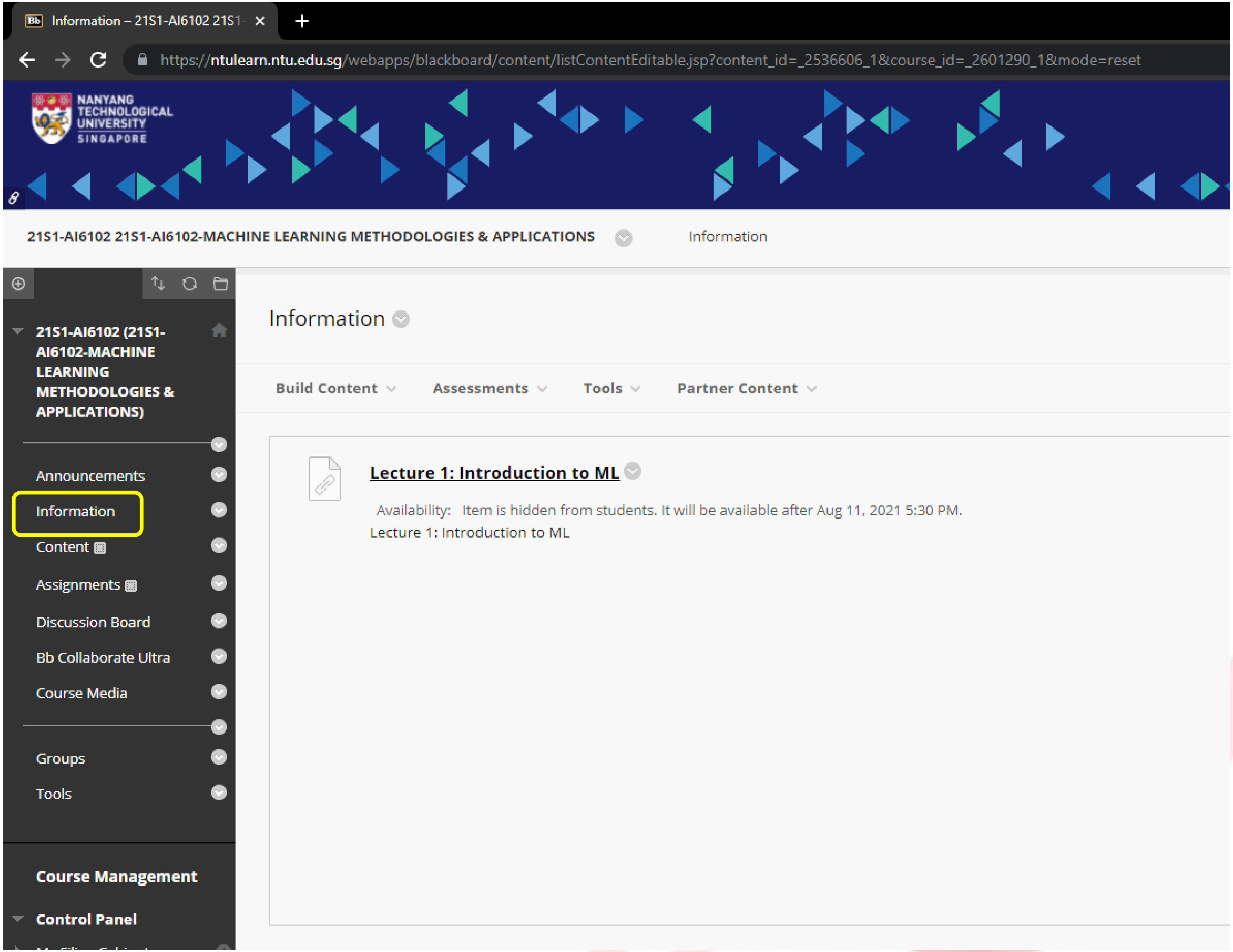
Sinno Jialin Pan

Nanyang Technological University, Singapore

Homepage: http://www.ntu.edu.sg/home/sinnopan

General Information

- Instructor:
  - Dr. Sinno Jialin PAN
- Time/venue
  - Wednesdays 6:30 – 9:30pm via Microsoft Teams
  - Webpage of AI6102 in NTULearn → Information → Teams link (available at 5:30pm on Wednesdays, links for each week are different)



https://ntulearn.ntu.edu.sg/webapps/blackboard/content/listContentEditable.jsp?content_id=_2536606_1&course_id=_2601290_1&mode=reset

21S1-AI6102 21S1-AI6102-MACHINE LEARNING METHODOLOGIES & APPLICATIONS   Information

Information

Lecture 1: Introduction to ML

Availability: Item is hidden from students. It will be available after Aug 11, 2021 5:30 PM.

Lecture 1: Introduction to ML

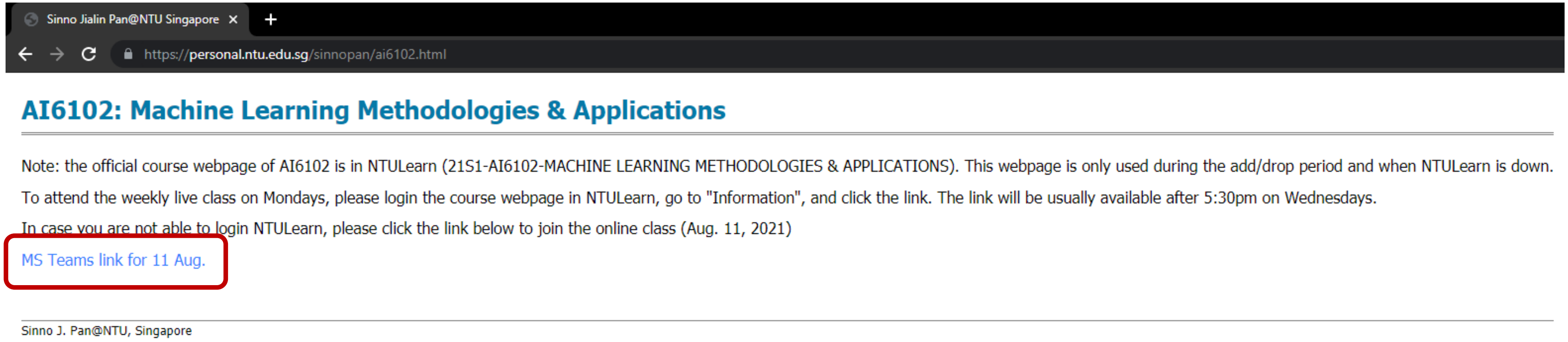
General Information (cont.)

Q&A

- Live Q&A after class
- Make an appointment via email sinnopan@ntu.edu.sg
- Send me questions via email

Course Webpage

- AI6102 @ NTULearn (official course webpage)
- https://www.ntu.edu.sg/home/sinnopan/ai6102.html (be used during the add/drop period, and when NTULearn is down)


Sinno Jialin Pan@NTU Singapore
https://personal.ntu.edu.sg/sinnopan/ai6102.html

AI6102: Machine Learning Methodologies & Applications

Note: the official course webpage of AI6102 is in NTULearn (21S1-AI6102-MACHINE LEARNING METHODOLOGIES & APPLICATIONS). This webpage is only used during the add/drop period and when NTULearn is down.

To attend the weekly live class on Mondays, please login the course webpage in NTULearn, go to "Information", and click the link. The link will be usually available after 5:30pm on Wednesdays.

In case you are not able to login NTULearn, please click the link below to join the online class (Aug. 11, 2021)

MS Teams link for 11 Aug.

Sinno J. Pan@NTU, Singapore

Prerequisites

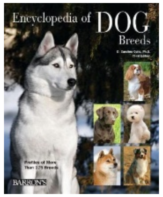
- The prerequisites for taking this module is "AI6104: MATHEMATICS FOR ARTIFICIAL INTELLIGENCE"
- Alternatively, showing evidence on taking the probability and linear algebra courses in your previous studies
  - Transcript containing the related courses need to be submitted for verification
- Otherwise, taking this module in Sem 2

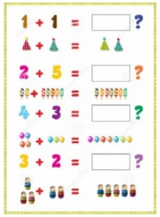
Evaluation

- Individual assignment (25%)
  - Details will be released by Week 4
- Open book quiz (35%)
  - ~1 hour (scheduled in Week 13 tentatively)
  - Scope: Weeks 2 – 11
  - Details will be released later
- Term paper (40%)
  - Details will be released by Week 4
  - Submission deadline: 21st Nov.

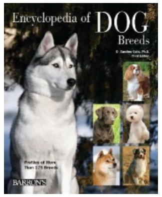
What is Machine Learning?

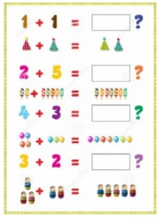
- Motivation: human beings can always learn from examples/experience





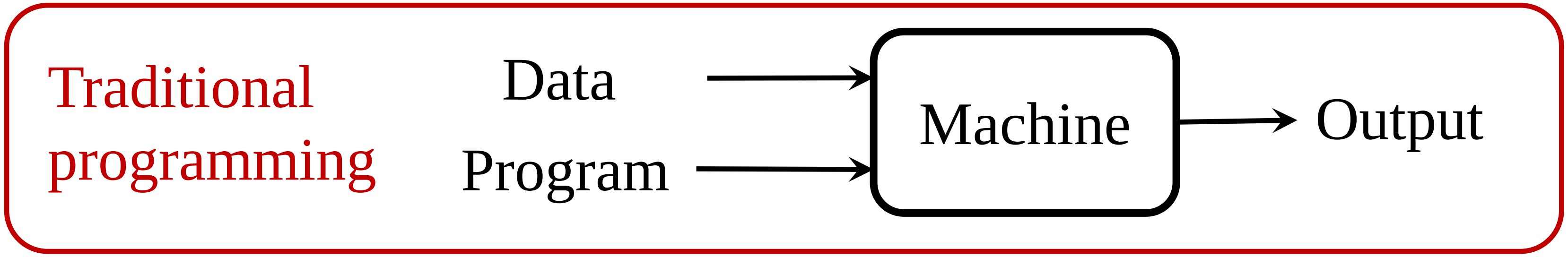
- Can machines teach themselves to grow from data and change when exposed to new data?





What is Machine Learning? (cont.)

- A type of artificial intelligence that provides computers with the ability to learn from examples/experience without being explicitly programmed



Traditional programming

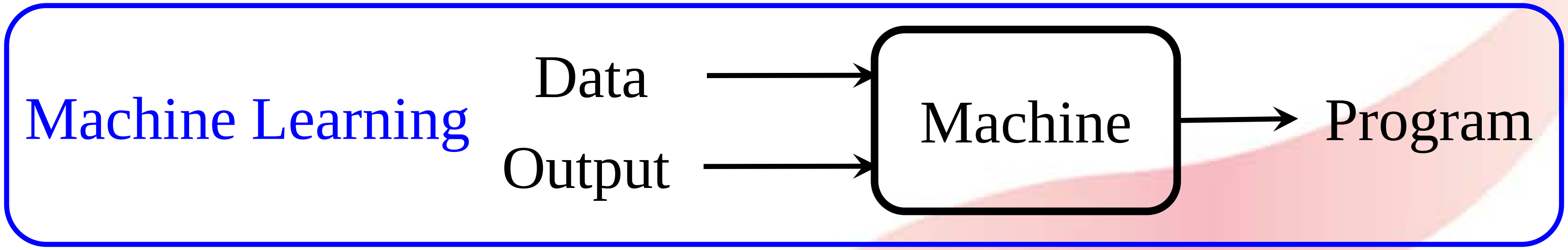
Data

Program

Machine

Output

V.S.



Machine Learning

Data

Output

Machine

Program

A Motivating Example:
Credit risk estimation

- Goal: to automatically evaluate the risk that a potential borrower will repay a loan or not
- Accessible information:
  - Profile of the loan applicant: name, age, gender, income, saving, marital status, profession, past financial history, etc
  - Records of past loans containing such profile data of previous borrowers and whether they finally repaid the loans or not

A Motivating Example:
Credit risk estimation

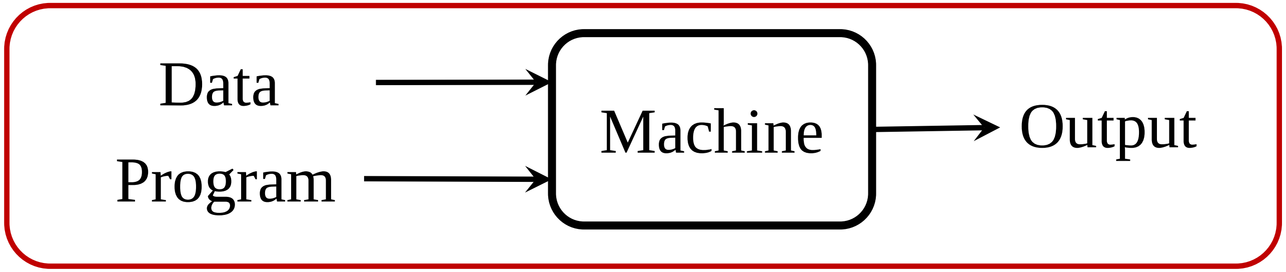
Records of past loans

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
3	M	Teacher	56k	100k	Yes
4	F	Student	12k	15k	Yes
5	M	Lawyer	80k	60k	No
6	M	Lawyer	100k	250k	Yes
7	F	Teacher	70k	34k	Yes
8	M	Engineer	85k	110k	No
9	M	Teacher	90k	250k	Yes
10	M	Student	8k	5k	No

Information of
a new applicant

ID	Gender	Profession	Income	Saving	Repay
11	F	Lawyer	70k	100k	?

Credit Risk Estimation
A traditional approach

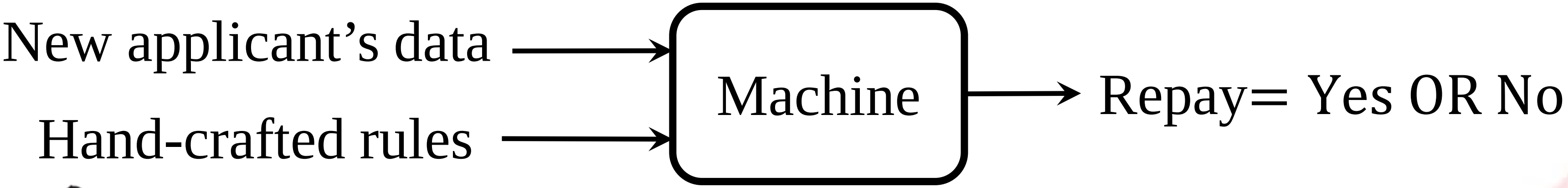


Data

Program

Machine

Output



New applicant's data

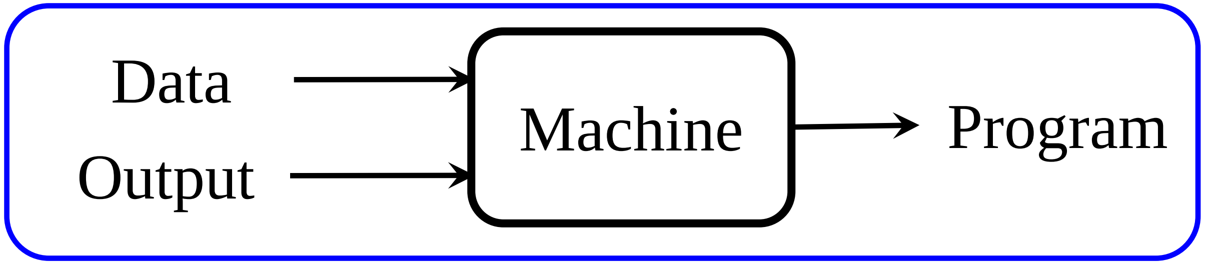
Hand-crafted rules

Machine

Repay= Yes OR No

E.g., IF income > 100k AND saving > 150k, THEN Repay = Yes

Credit Risk Estimation
A machine learning approach

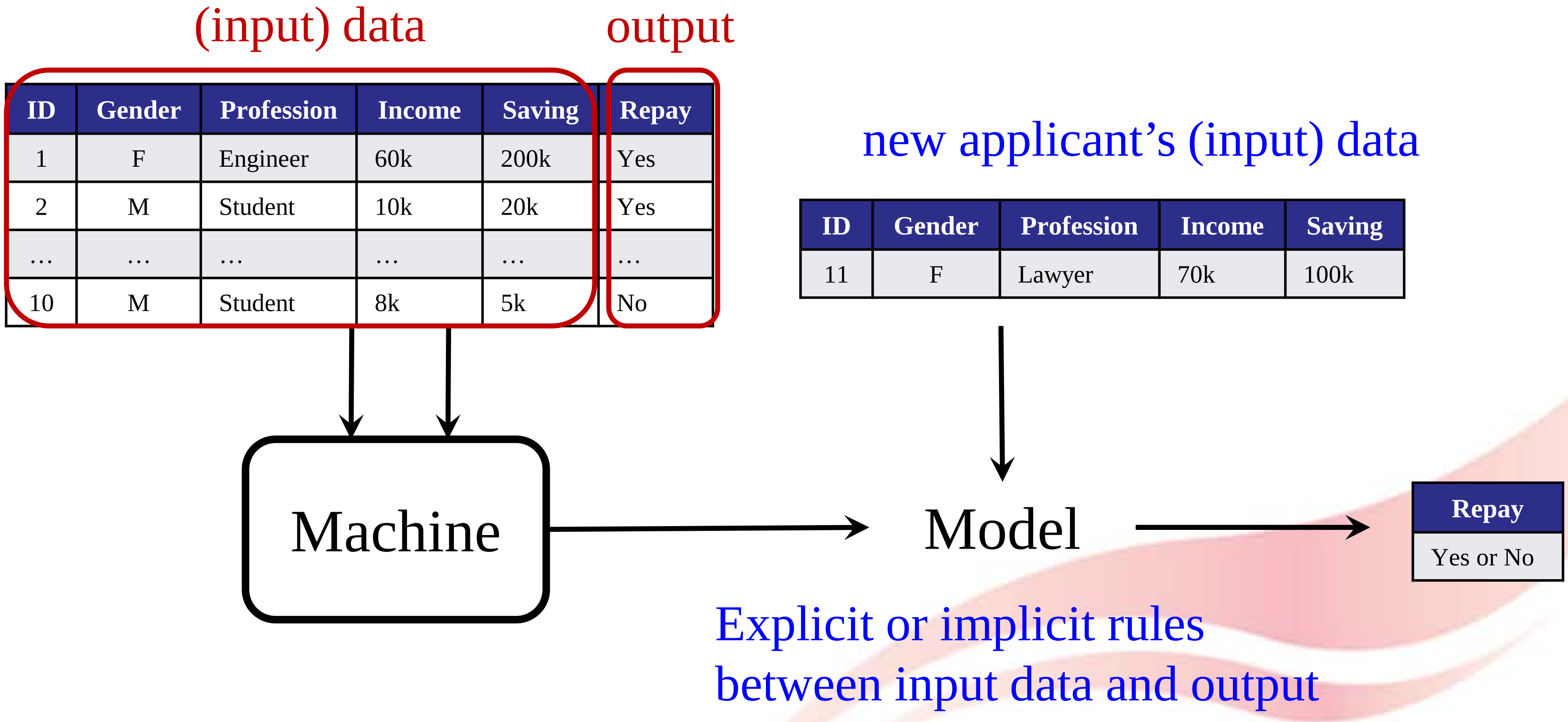


Data

Output

Machine

Program



(input) data

output

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
…	…	…	…	…	…
10	M	Student	8k	5k	No

new applicant's (input) data

ID	Gender	Profession	Income	Saving
11	F	Lawyer	70k	100k

Machine

Model

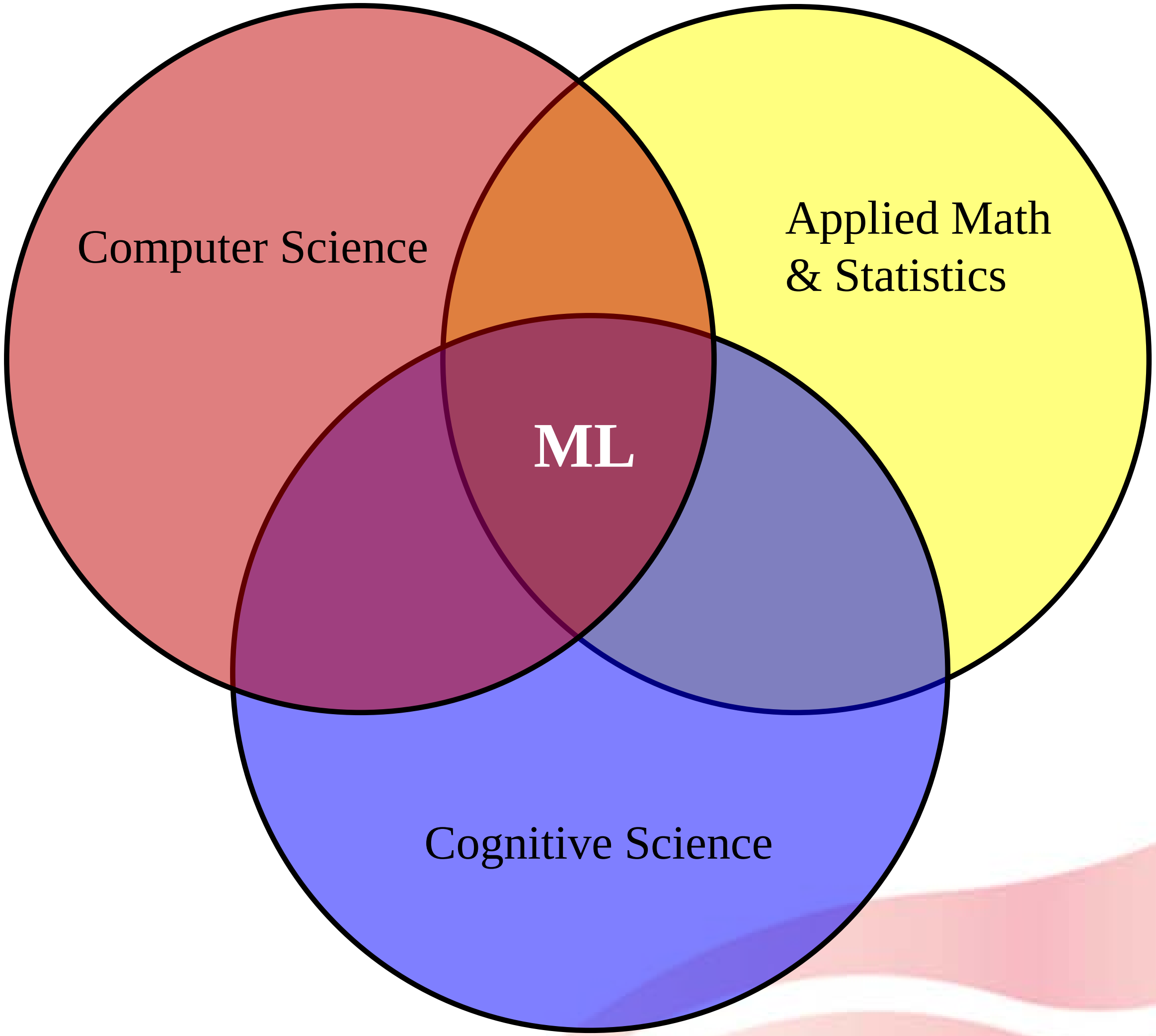
Repay
Yes or No

Explicit or implicit rules
between input data and output

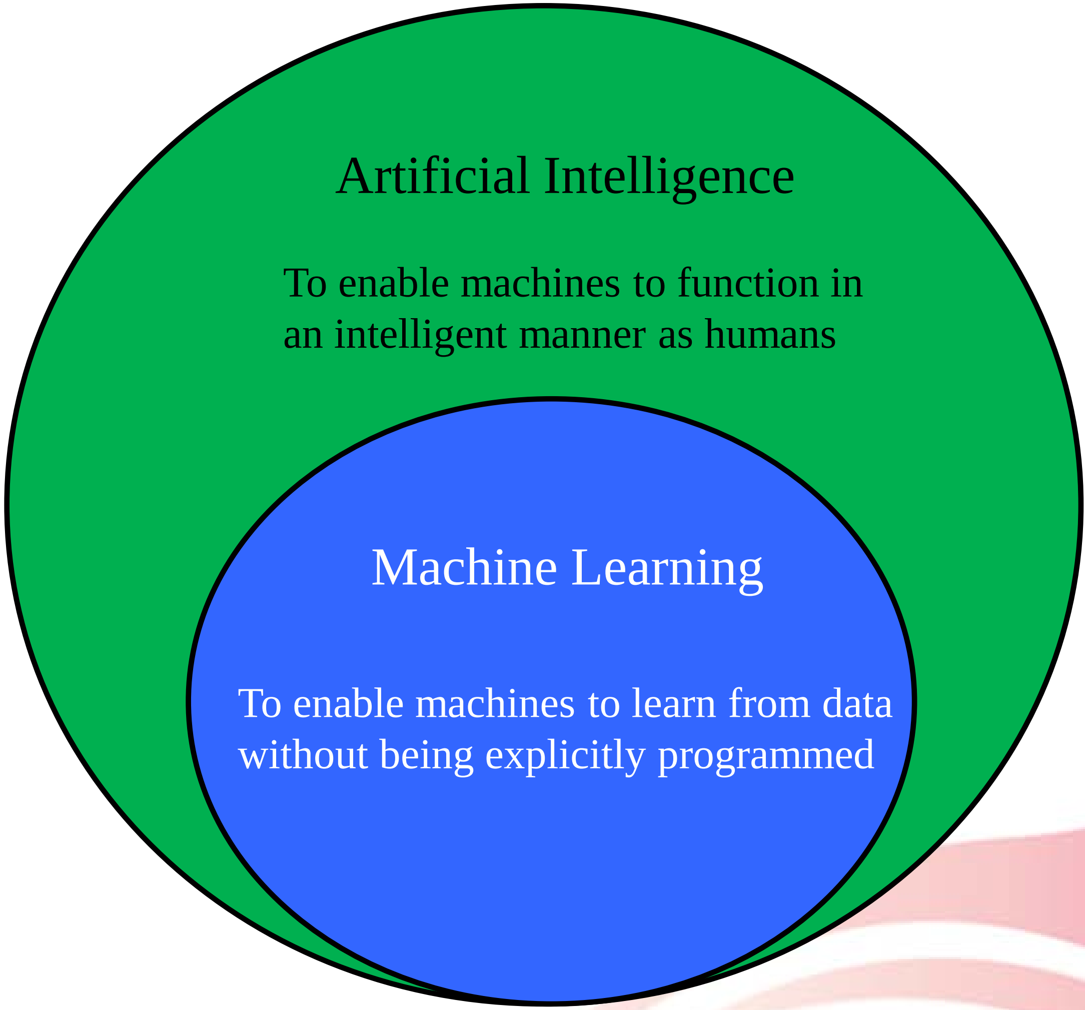
Machine Learning Characteristics

- Data driven
  - Aims to automatically learn a model from data
- Focus on the development of general algorithms to learn a model from data
  - Not focus on specific application problems

ML is Interdisciplinary



Machine Learning & AI



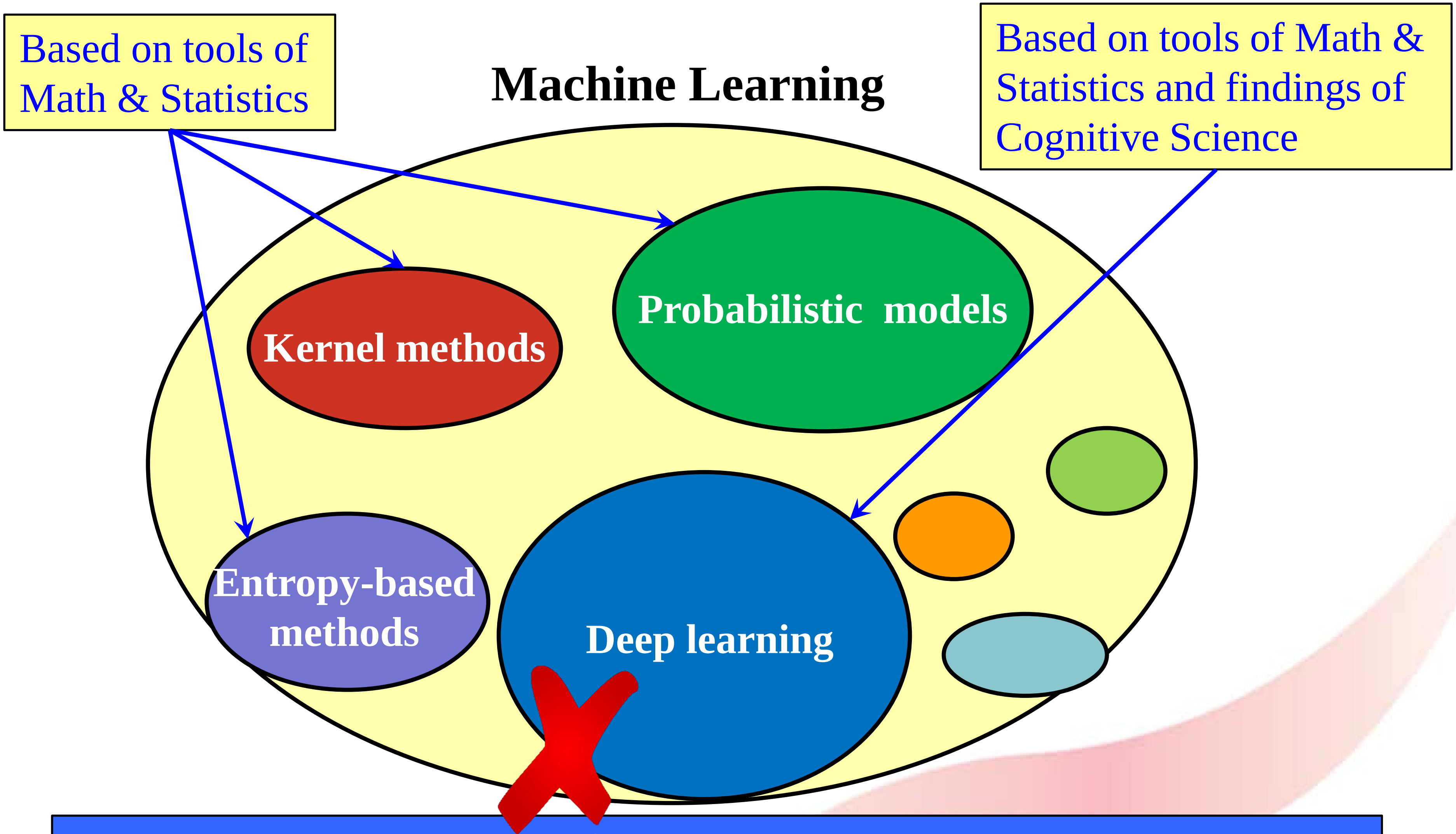
Artificial Intelligence

To enable machines to function in an intelligent manner as humans

Machine Learning

To enable machines to learn from data without being explicitly programmed

Machine Learning Methodologies



Based on tools of Math & Statistics

Machine Learning

Based on tools of Math & Statistics and findings of Cognitive Science

Kernel methods

Probabilistic models

Entropy-based methods

Deep learning

Not covered in this module. Refer to AI6103: Deep Learning and Applications

Different Learning Paradigms

- Three important paradigms:
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
- Some advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning
  - etc.

Supervised Learning

- Learning with “supervision”
- A set of examples are presented to a machine, and each example is a pair of an input data instance and a desired output

labeled data

- The goal is to “learn” a function or prediction model from inputs to outputs

supervision or label

Supervised Learning (cont.)

In mathematics

- Given: a set of $\{\boldsymbol{x}_i, y_i\}$ for $i = 1, \dots, N$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ is m-dimensional vector of numerical values, and y_i is a scalar
- Goal: to learn a mapping $f\colon \boldsymbol{x} \to y$ by requiring $f(\boldsymbol{x}_i) = y_i$
- The learned mapping f is expected to make precise predictions on any unseen $\boldsymbol{x}^*$ as $f(\boldsymbol{x}^*)$

Supervised Learning Tasks

Use discrete numerical values
to represent **categories**

- Classification v.s. Regression
- If the value of each output y_i is discrete, then it is a classification task
  - Only two distinct values: binary classification
  - More than two distinct values: multi-class classification
- If the value of each output y_i is continuous, then it is a regression task

Credit Risk Estimation: Binary

(input) data

output

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
…	…	…	…	…	…
10	M	Student	8k	5k	No

new applicant’s (input) data

ID	Gender	Profession	Income	Saving
11	F	Lawyer	70k	100k

Yes: 1

No: -1

	X_1	X_2	…	X_{m-1}	X_m	Y	
$\boldsymbol{x}_1$	1	0	…	60	200	1	y_1
$\boldsymbol{x}_2$	0	1	…	10	20	1	y_2
	…	…	…	…	…	…	
$\boldsymbol{x}_{10}$	0	1	…	8	5	-1	y_{10}

	X_1	X_2	…	X_{m-1}	X_m
$\boldsymbol{x}^*$	1	0	…	70	100

$$f: \boldsymbol{x} \rightarrow y \text{ s.t. } f(\boldsymbol{x}_i) = y_i, i = 1 \ldots, 10$$

Learning

$$f: \boldsymbol{x}^* \rightarrow y^* \text{via } f(\boldsymbol{x}^*) = y^*$$

Using

Credit Risk Estimation: Multiclass

(input) data

output

ID	Gender	Profession	Income	Saving	Repay Time
1	F	Engineer	60k	200k	On Time
2	M	Student	10k	20k	Late
…	…	…	…	…	…
10	M	Student	8k	5k	Never

new applicant’s (input) data

ID	Gender	Profession	Income	Saving
11	F	Lawyer	70k	100k

	X_1	X_2	…	X_{m-1}	X_m	Y	
$\boldsymbol{x}_1$	1	0	…	60	200	1	y_1
$\boldsymbol{x}_2$	0	1	…	10	20	2	y_2
	…	…	…	…	…	…	
$\boldsymbol{x}_{10}$	0	1	…	8	5	3	y_{10}

On Time: 1
Late: 2
Never: 3

	X_1	X_2	…	X_{m-1}	X_m
$\boldsymbol{x}^*$	1	0	…	70	100

$$f: \boldsymbol{x} \rightarrow y \text{ s.t. } f(\boldsymbol{x}_i) = y_i, i = 1 \dots, 10$$

Learning

$$f: \boldsymbol{x}^* \rightarrow y^* \text{via } f(\boldsymbol{x}^*) = y^*$$

Using

Credit Risk Estimation: Regression

(input) data

output

ID	Gender	Profession	Income	Saving	Repay %
1	F	Engineer	60k	200k	100
2	M	Student	10k	20k	95
…	…	…	…	…	…
10	M	Student	8k	5k	0

new applicant’s (input) data

ID	Gender	Profession	Income	Saving
11	F	Lawyer	70k	100k

	X_1	X_2	…	X_{m-1}	X_m	Y	
$\boldsymbol{x}_1$	1	0	…	60	200	100	y_1
$\boldsymbol{x}_2$	0	1	…	10	20	95	y_2
	…	…	…	…	…	…	
$\boldsymbol{x}_{10}$	0	1	…	8	5	0	y_{10}

	X_1	X_2	…	X_{m-1}	X_m
$\boldsymbol{x}^*$	1	0	…	70	100

$f: \boldsymbol{x} \to y$ s.t. $f(\boldsymbol{x}_i) = y_i, i = 1 \dots, 10$

Learning

$f: \boldsymbol{x}^* \to y^*$via $f(\boldsymbol{x}^*) = y^*$

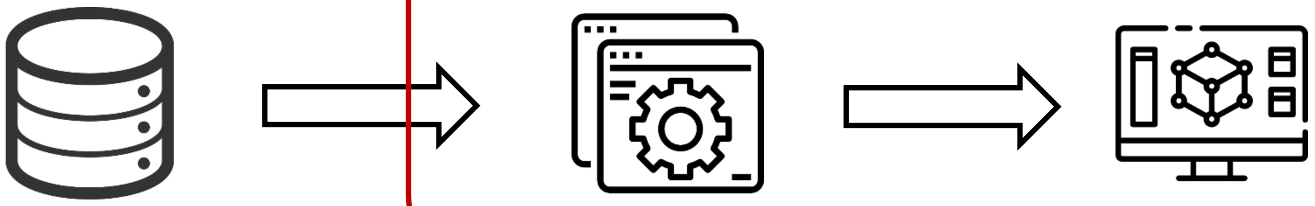
Using

Typical Procedure of Supervised Learning

ID	Gender	Profession	Income	Saving	Repay
1	F	Engineer	60k	200k	Yes
2	M	Student	10k	20k	Yes
…	…	…	…	…	…
10	M	Student	8k	5k	No

X_1	X_2	…	X_{m-1}	X_m	Y
1	0	…	60	200	1
0	1	…	10	20	1
…	…	…	…	…	…
0	1	…	8	5	-1

$$f: x \rightarrow y$$



Training phase

Training data

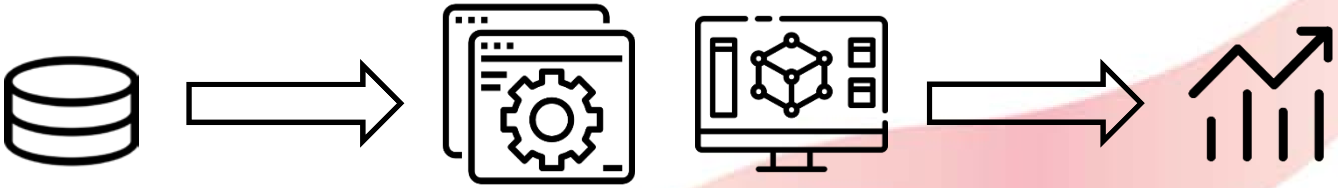
Feature engineering

Learning a model

ID	Gender	Profession	Income	Saving
11	F	Lawyer	70k	100k

X_1	X_2	…	X_{m-1}	X_m
1	0	…	70	100

$$f(x^*)$$



Test phase

Test data

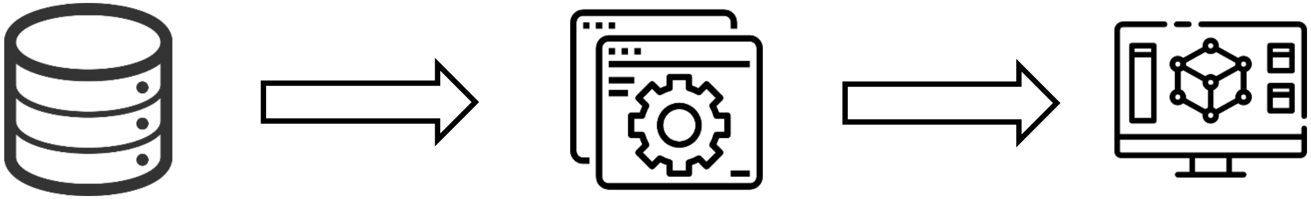
Engineered features + Trained model

Predictions

Content of Supervised Learning

$\{\boldsymbol{x}_i, y_i\}$ for $i = 1, \dots, N$

$f: \boldsymbol{x} \rightarrow y$



Training data

Feature engineering

Learning a model

In this module

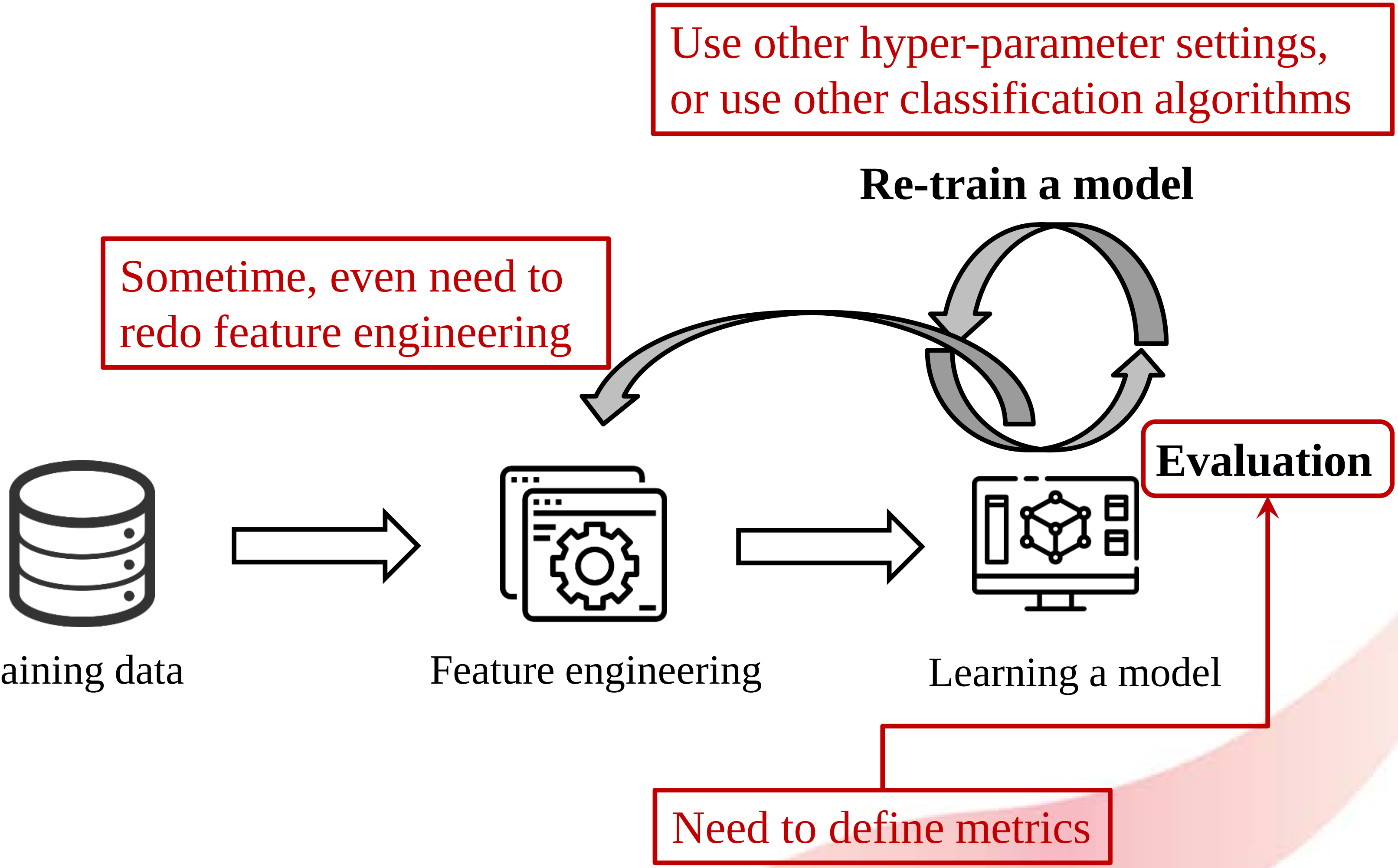
- Basic transformations to numerical vectors

Note: domain knowledge or experience is required to manually design good features

- Linear models
- Kernel methods
- Tree-based methods
- Bayesian classifiers
- KNN classifiers (a lazy classifier)

An end-to-end solution: deep learning, effective on unstructured data, like images, text, speech data, etc. (out of scope)

Loop of Training a Model



Use other hyper-parameter settings, or use other classification algorithms

Re-train a model

Sometime, even need to redo feature engineering

Evaluation

Training data

Feature engineering

Learning a model

Need to define metrics

Common Performance Metrics

- Classification: accuracy or error rate

Ground-truth y_i	Prediction $f(\boldsymbol{x}_i)$	
$+1$	-1	✗
-1	-1	✓
$+1$	$+1$	✓
-1	$+1$	✗
-1	-1	✓

The rate of correct predictions

$$\text{Accuracy } = 3/5$$

The rate of incorrect predictions

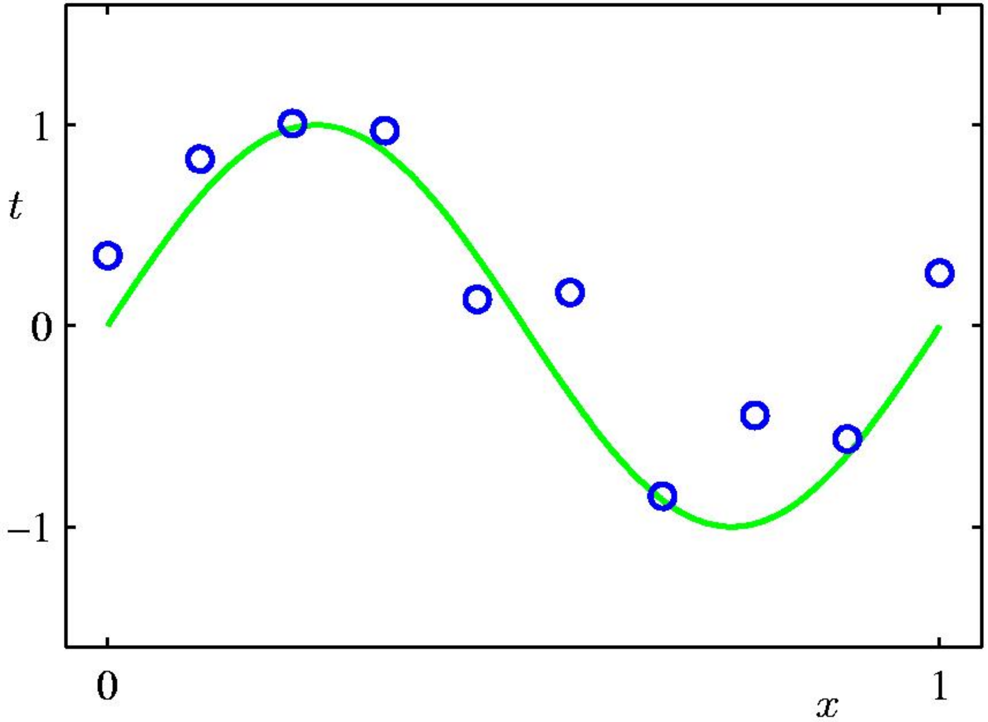
$$\text{Error rate } = 2/5$$

- Regression:
  - Mean Absolute Error (MAE): $\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|f(\boldsymbol{x}_i) - y_i|$
  - Root Mean Squared Error (RMSE): $\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f(\boldsymbol{x}_i) - y_i)^2}$

Model Evaluation

- With a performance metric, how to evaluate the performance of a trained model?
- A straight-forward solution:
  - Step 1: Use the training data to train a model
  - Step 2: Apply the trained model to make predictions on the training data, and calculate the performance metric, e.g., error rate
- The model highly risky to “over-fit” the training data

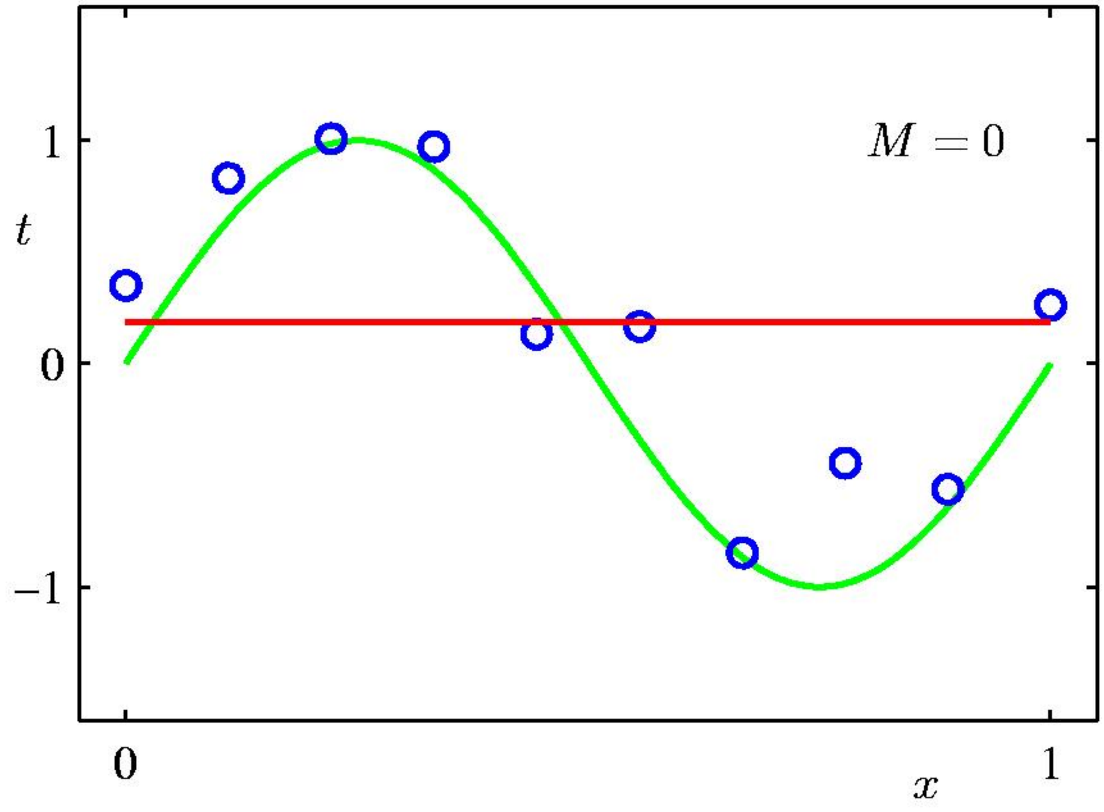
Example: Polynomial Curve Fitting



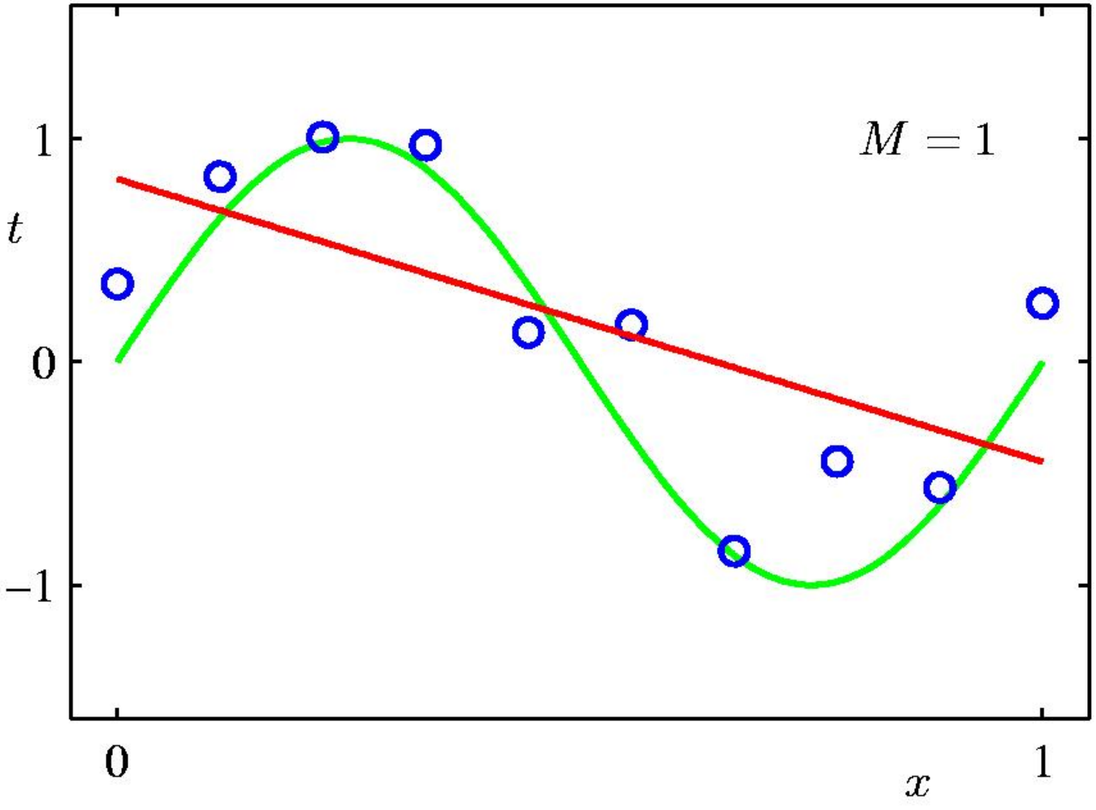
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Evaluation metric: Root Mean Squared Error (RMSE)

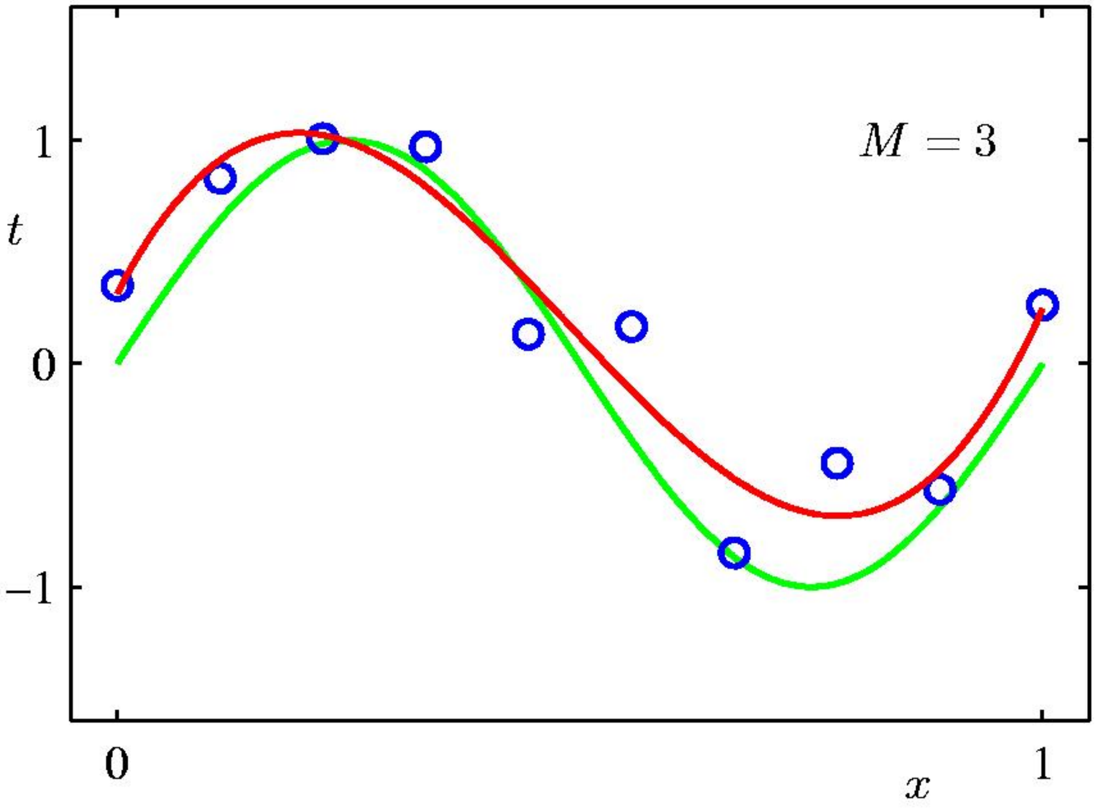
0th Order Polynomial



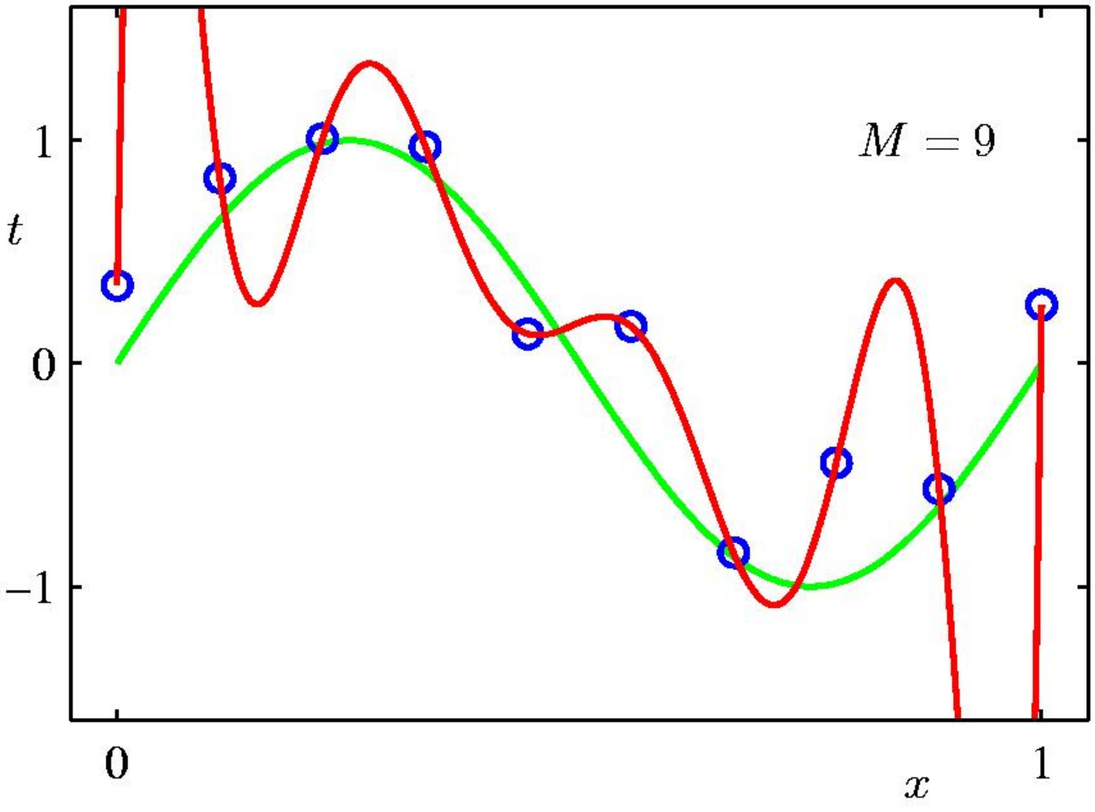
1st Order Polynomial



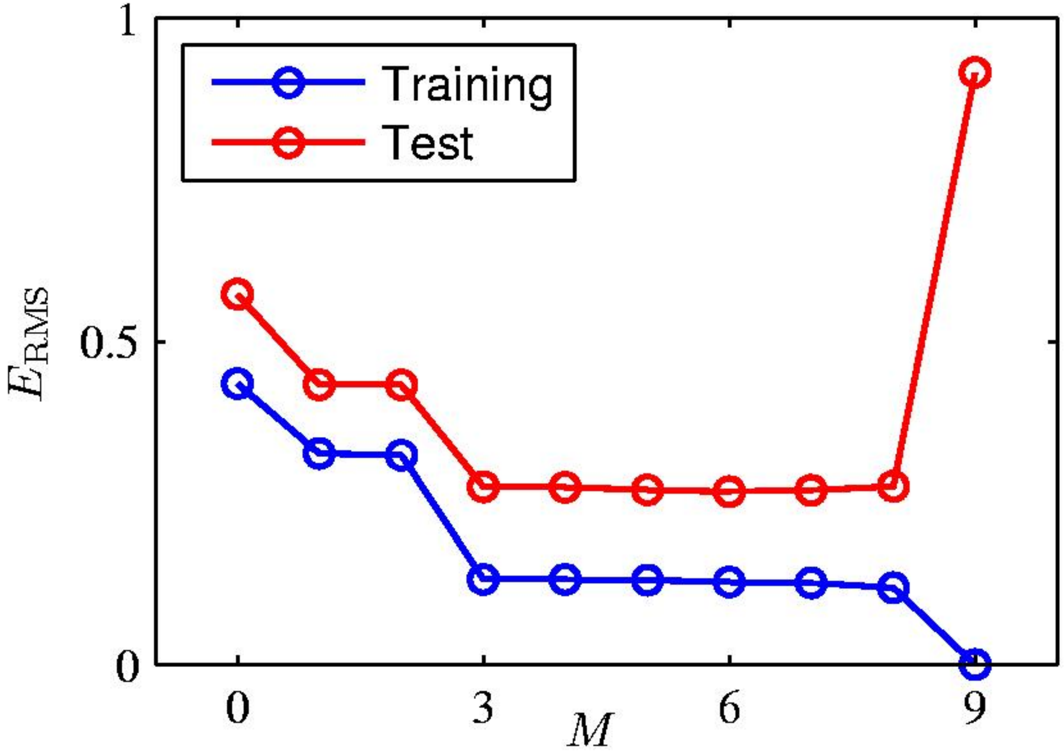
3rd Order Polynomial



9th Order Polynomial



Over-fitting



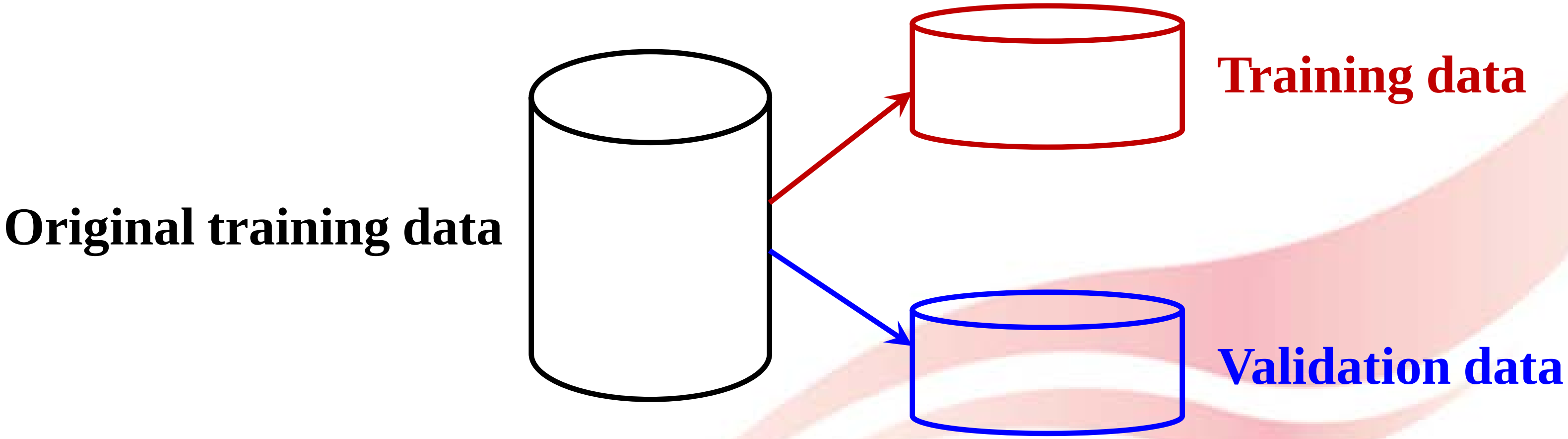
Root Mean Square Error (RMSE)

Note on Over-fitting

- Recall: the goal of supervised learning is learn a model such that it makes precise predictions on any unseen data (i.e., test data) NOT training data
- Performance calculated on training data fails to provide a good estimate of how well the model will perform on previously unseen data
- Solution: validation set

Validation Set

- Split the whole training dataset into two disjoint sets: "training" set and "validation" set
- The split "training" set is used to train a prediction model and the "validation" set is used to evaluate the performance of the trained model
- How to split the dataset?



Random Subsampling

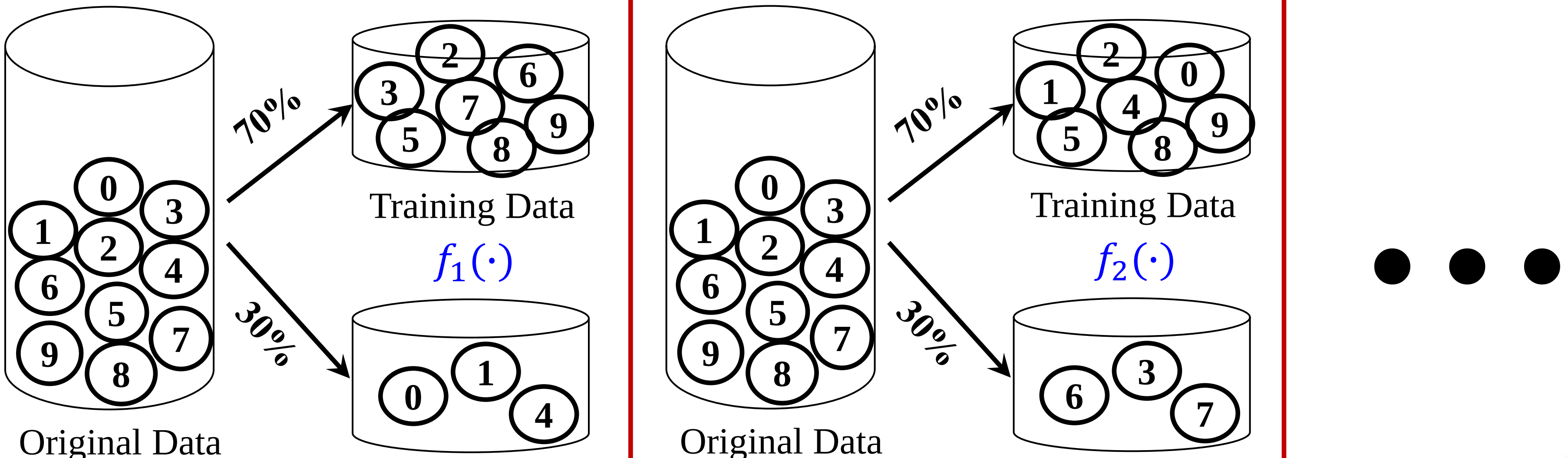
For a classification algorithm $\mathcal{A}$, set hyper-parameter $\Theta = \theta_1$

1st random sampling

(Sampled without replacement)

2nd random sampling

(Sampled without replacement)



Original Data

Training Data

$f_1(\cdot)$

Validation Data

accuracy_1

Original Data

Training Data

$f_2(\cdot)$

Validation Data

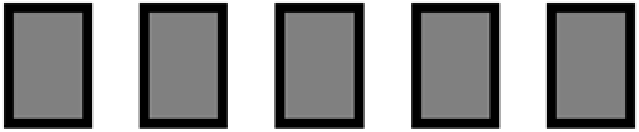
accuracy_2

• • •

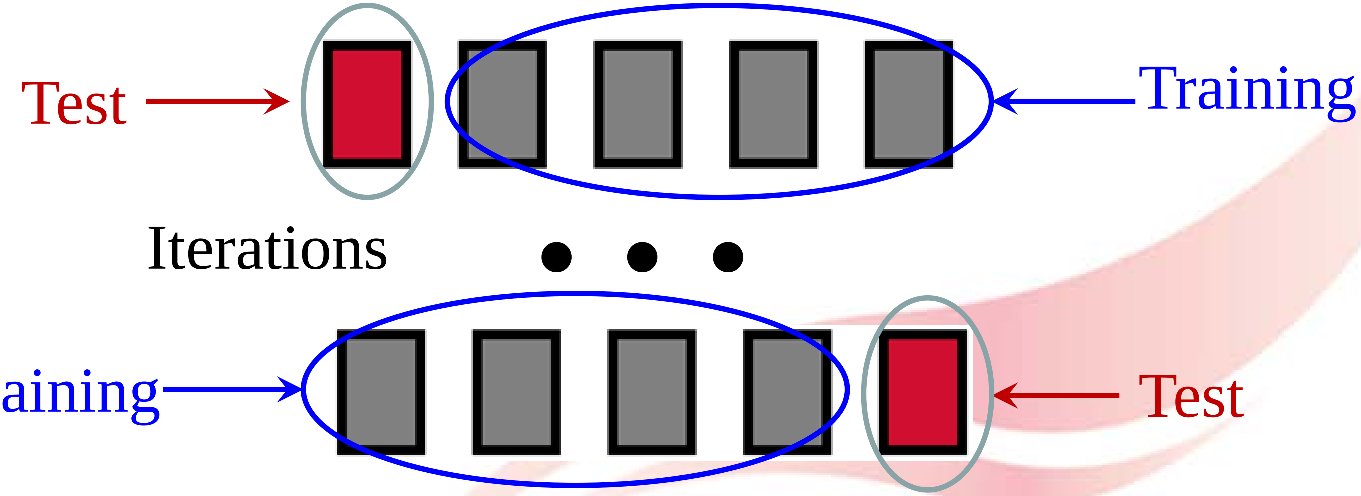
$$\text{acc} = \frac{1}{k}\sum_{i=1}^{k} \text{accuracy}_i \longrightarrow$$ Performance evaluation for $\mathcal{A}$ with hyper-parameter $\Theta = \theta_1$

K-fold Cross Validation

- *k*-fold cross-validation: partition data into *k* subsets of the same size



- Hold aside one group for testing and use the rest to build model



An Example: 5-fold

For a classification algorithm $\mathcal{A}$, set hyper-parameter $\Theta = \theta_1$

Partition into 5 subsets	**1** $\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	**2** $\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	**3** $\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	**4** $\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	**5** $\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}$

Hold aside 1 group for testing, use the rest 4 for training	**1: Test** $\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	**2: Train** $\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	**3: Train** $\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	**4: Train** $\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	**5: Train** $\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}$
	1: Train $\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	**2: Test** $\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	**3: Train** $\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	**4: Train** $\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	**5: Train** $\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}$
	1: Train $\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	**2: Train** $\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	**3: Test** $\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	**4: Train** $\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	**5: Train** $\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}$
	1: Train $\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	**2: Train** $\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	**3: Train** $\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	**4: Test** $\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	**5: Train** $\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}$
	1: Train $\{\boldsymbol{x}_1, y_1\}, \{\boldsymbol{x}_2, y_2\}$	**2: Train** $\{\boldsymbol{x}_3, y_3\}, \{\boldsymbol{x}_4, y_4\}$	**3: Train** $\{\boldsymbol{x}_5, y_5\}, \{\boldsymbol{x}_6, y_6\}$	**4: Train** $\{\boldsymbol{x}_7, y_7\}, \{\boldsymbol{x}_8, y_8\}$	**5: Test** $\{\boldsymbol{x}_9, y_9\}, \{\boldsymbol{x}_{10}, y_{10}\}$

Evaluation, e.g., accuracy for $\mathcal{A}$, with $\Theta = \theta_1$	1: Prediction $\hat{y}_1, \hat{y}_2$	2: Prediction $\hat{y}_3, \hat{y}_4$	3: Prediction $\hat{y}_5, \hat{y}_6$	4: Prediction $\hat{y}_7, \hat{y}_8$	5: Prediction $\hat{y}_9, \hat{y}_{10}$
	1: Ground-truth y_1, y_2	2: Ground-truth y_3, y_4	3: Ground-truth y_5, y_6	4: Ground-truth y_7, y_8	5: Ground-truth y_9, y_{10}

Different Learning Paradigms

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning

Unsupervised Learning

- Learning without “supervision”
- A set of examples presented to a machine only contains input data instances without desired outputs, the goal is to “learn” intrinsic structures or patterns underlying the input data instances

Unlabeled data

Unsupervised Learning (cont.)

In mathematics

- Given: a set of $\{\boldsymbol{x}_i\}$ for $i = 1, \dots, N$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ is m-dimensional vector of numerical values
- Goal: to learn a model $g: \boldsymbol{x} \to \boldsymbol{z}$, where $\boldsymbol{z}$ captures patterns or hidden structure of the $\boldsymbol{x}$

Unsupervised Learning Tasks

- Clustering, (unsupervised) dimensionality reduction, density estimation
  - Clustering: to automatically assign each input data instance $\boldsymbol{x}_i$ to a group
    - $g: \boldsymbol{x}_i \rightarrow z_i$, here z_i is the index of a group
  - Dimensionality reduction: to map each input data instance $\boldsymbol{x}_i$ to a lower-dimensional vector
    - $g: \boldsymbol{x}_i \rightarrow \boldsymbol{z}_i$, here $\boldsymbol{z}_i$ is a vector of d dimensions, $d < m$
  - Density estimation: to learn an density function
    - $g: \boldsymbol{x}_i \rightarrow z_i$, here z_i is the value of the density function at the data point $\boldsymbol{x}_i$

Clustering: User Segmentation

Common hyper-parameter of most clustering algorithms

Suppose we want to cluster potential customers into $\boxed{3}$ groups, and advertise a different loaning plan to different groups

ID	Gender	Profession	Income	Saving
1	F	Engineer	60k	200k
2	M	Student	10k	20k
…	…	…	…	…
10	M	Student	8k	5k

	$\boldsymbol{X}_1$	$\boldsymbol{X}_2$	…	$\boldsymbol{X}_{m-1}$	$\boldsymbol{X}_m$
$\boldsymbol{x}_1$	1	0	…	60	200
$\boldsymbol{x}_2$	0	1	…	10	20
	…	…	…	…	…
$\boldsymbol{x}_{10}$	0	1	…	8	5

$$g : \boldsymbol{x} \rightarrow z$$

	$\boldsymbol{X}_1$	$\boldsymbol{X}_2$	…	$\boldsymbol{X}_{m-1}$	$\boldsymbol{X}_m$	Z
$\boldsymbol{x}_1$	1	0	…	60	200	1
$\boldsymbol{x}_2$	0	1	…	10	20	3
	…	…	…	…	…	
$\boldsymbol{x}_{10}$	0	1	…	8	5	1

Different Learning Paradigms

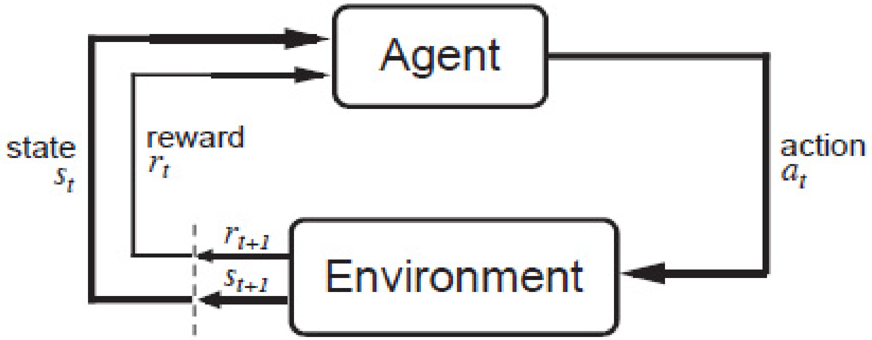
- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning

Reinforcement Learning

- Learning by interacting with an environment to achieve a goal

$p: \boldsymbol{s}_i \rightarrow a_i$

- Objective: to learn an optimal policy mapping states to actions

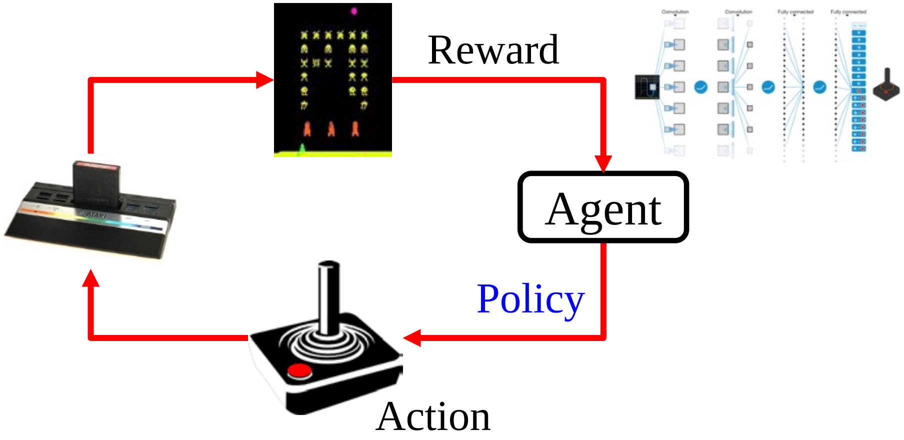


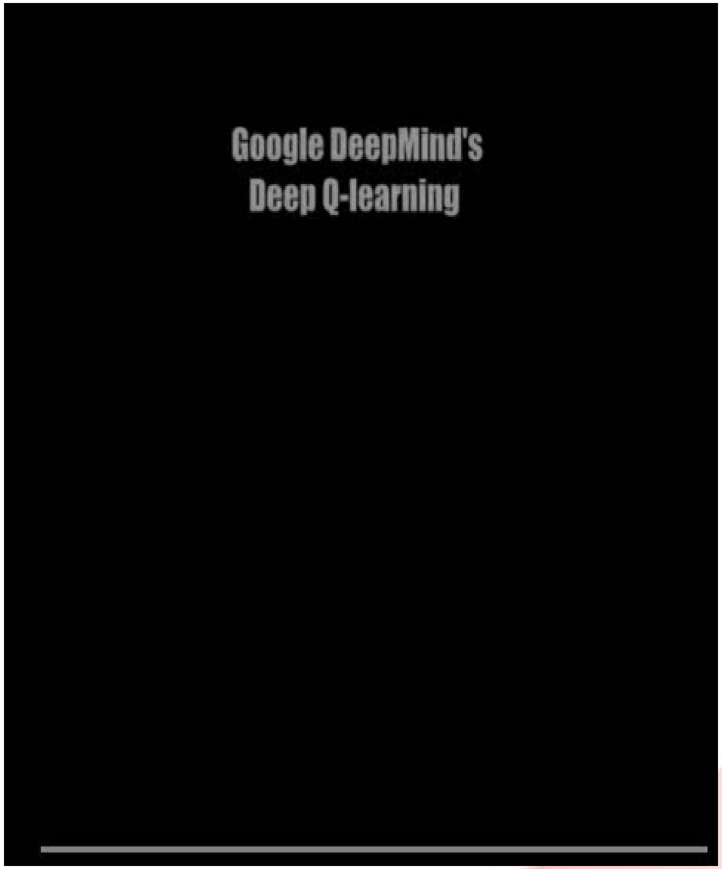
- Each state S_i is represented by a feature vector $\boldsymbol{s}_i$
- Each action is represented by a discrete value (categorical)
- Each reward is a scalar (indirect supervision)

$$S_0 \xrightarrow[r_0]{a_0} S_1 \xrightarrow[r_1]{a_1} S_2 \xrightarrow[r_2]{a_2} S_3$$

Reinforcement Learning (cont.)

- Deep Q-Network (DQN) [1]
  - Play Atari 2600 Games




Google DeepMind's
Deep Q-learning

[1] Mnih et al, Human-level control through deep reinforcement learning. *Nature*, 2015

Different Learning Paradigms

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning

Semi-supervised Learning

- Recall: in supervised learning, a set of labeled data is given for training. In general, sufficient labeled data set is required to train a precise prediction model
- However, in some real-world application scenarios, labeled data is in short supply while unlabeled data is easy to collect
- Semi-supervised learning aims to make use of unsupervised learning techniques with unlabeled data to help a supervised learning task with insufficient labeled data

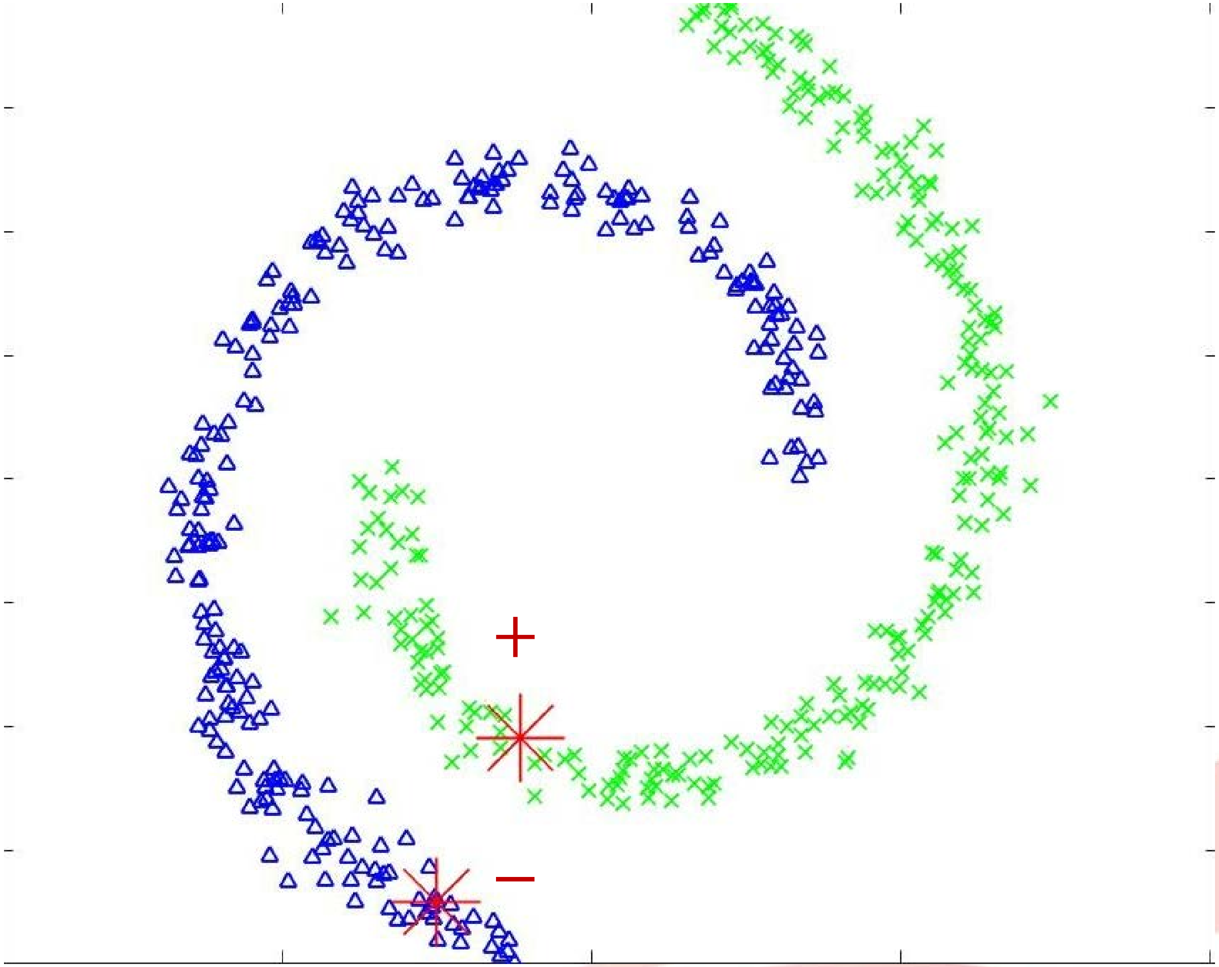
Semi-supervised Learning (cont.)

In mathematics

- Given: a small set of labeled data $\{\boldsymbol{x}_i, y_i\}$ for $i = 1, \dots, L$, and a set of unlabeled data $\{\boldsymbol{x}_j\}$ for $j = 1, \dots, U$. In general, $L \ll U$
- Goal: to learn a precise mapping $f: \boldsymbol{x} \rightarrow y$ by requiring $f(\boldsymbol{x}_i) = y_i$ and making using of the patterns underlying unlabeled data $\{\boldsymbol{x}_j\}$'s
- The learned mapping f is expected to make precise predictions on any unseen $\boldsymbol{x}^*$ as $f(\boldsymbol{x}^*)$

A Motivating Example

Two moons dataset



A Real-world Example

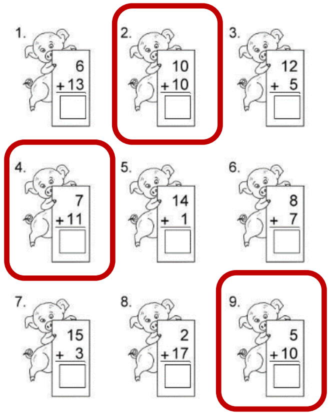
- Suppose our task is to predict whether a patient has a rare gene disease based on some symptoms and some basic tests
- Input: symptoms and basic tests results
- Output: whether the patient has the gene disease or not
- However, to get the “true” label, a gene test is needed, which is very expensive and time consuming
  - The labeled dataset available for training is small
  - Meanwhile, we may have a lot of unlabeled data, i.e., patients with symptoms and basic tests results
  - Semi-supervised learning can help!

Different Learning Paradigms

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning

Motivation

To understand a mathematical concept, a teacher may assign a lot of exercises to practice



If a student is smart and active, he can first analyze the exercises to figure out which are the key ones to understand the concept, then he can selectively do exercises rather than doing all of them

Active Learning

In mathematics

- Given: a pool of unlabeled data $\{\boldsymbol{x}_i\}$ for $i = 1, \ldots, N$, and a budget to query labels from an oracle
- Goal:
  - 1) to an selectively choose unlabeled data to query the oracle based on the budget to retrieve the corresponding labels : $\{\boldsymbol{x}_j, y_j\}$, $j = 1, \ldots, L$, and $L \ll N$
  - 2) learn a mapping $f: \boldsymbol{x} \rightarrow y$ with $\{\boldsymbol{x}_j, y_j\}$ using a supervised learning algorithm
- The learned mapping f is expected to make precise predictions on any unseen $\boldsymbol{x}^*$ as $f(\boldsymbol{x}^*)$

Gene Disease Example

- Suppose we have budget to conduct the gene test on 20 more patients to verify whether they have the gene disease or not
  - i.e., we are able to get 20 more labeled data instances
- Which 20 patients should we conduct the gene test on?
- Active learning is focused on selecting 20 more patients to test their “labels” such that with these 20 more labeled data instances, a <u>precise</u> classifier can be trained

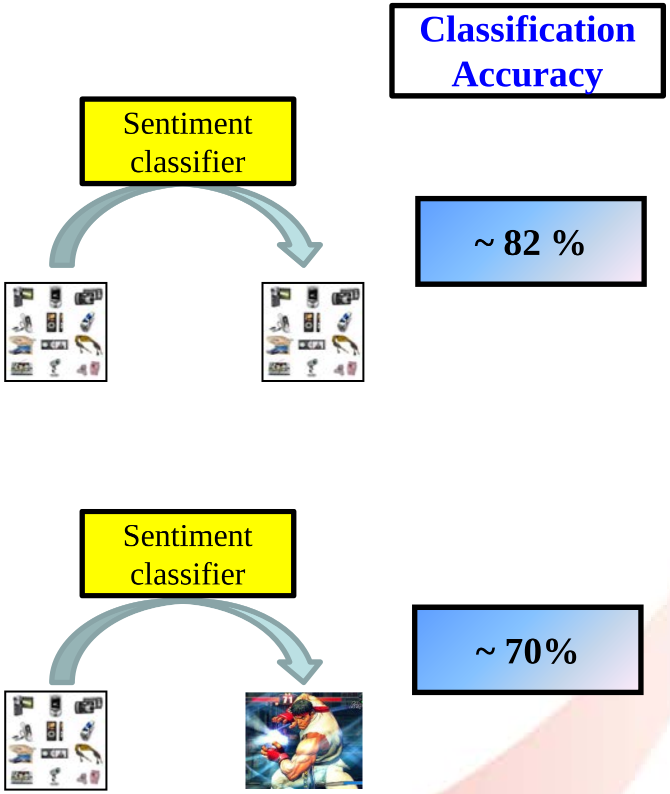
Different Learning Paradigms

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Advanced paradigms:
  - Semi-supervised learning
  - Active learning
  - Transfer learning

Motivating Example I

	Electronics	Video Games
👍	(1) **Compact**; easy to operate; very good picture quality; looks **sharp**!	(2) A very good game! It is action packed and full of excitement. I am very much **hooked** on this game.
👍	(3) I purchased this unit from Circuit City and I was very excited about the quality of the picture. It is really nice and **sharp**.	(4) Very **realistic** shooting action and good plots. We played this and were **hooked**.
👎	(5) It is also quite **blurry** in very dark settings. I will never buy HP again.	(6) The game is so **boring**. I am extremely unhappy and will probably never buy UbiSoft again.

Product reviews on different domains



Classification Accuracy

Sentiment classifier

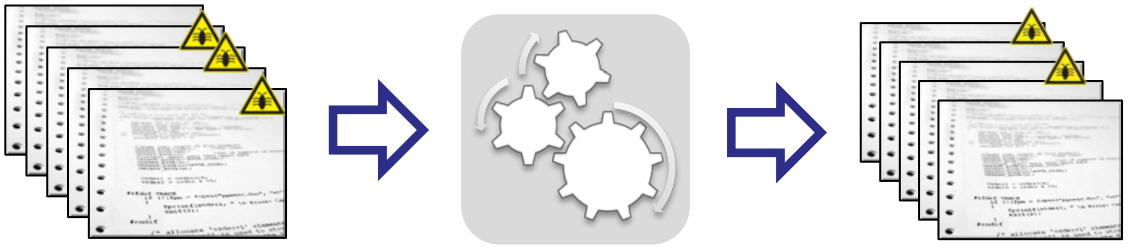
~ 82 %

Sentiment classifier

~ 70%

Motivating Example II

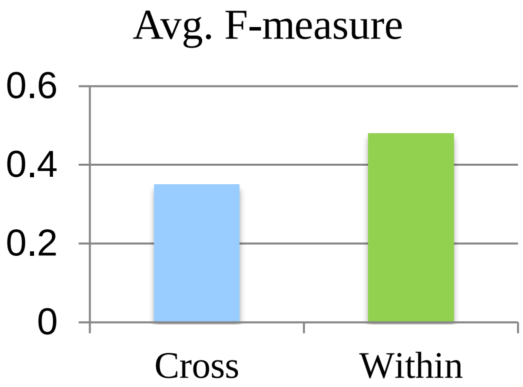
For a particular project:



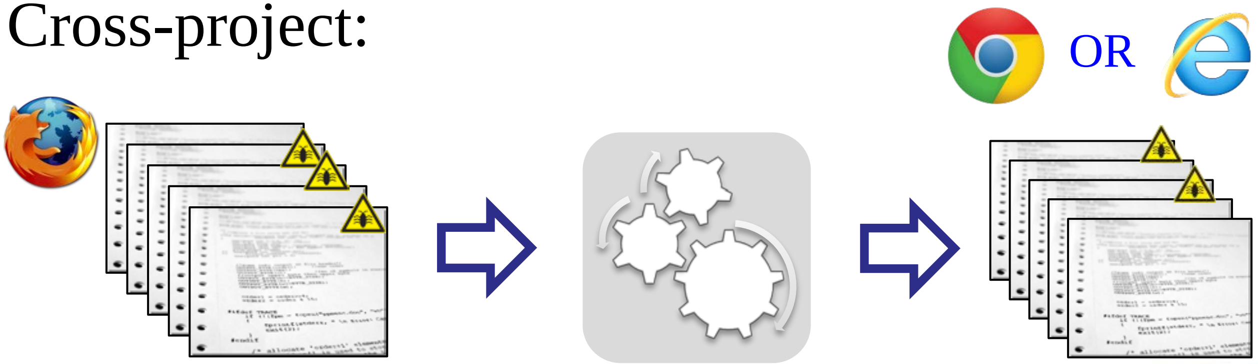
Program with defect information

Predictive Model

Future defects



Cross-project:



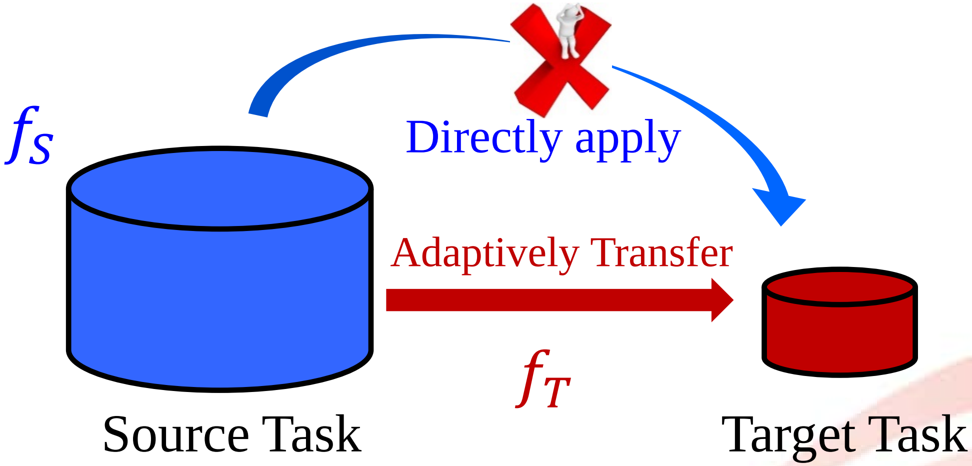
Program with defect information

Predictive Model

Program in another Project

Motivation

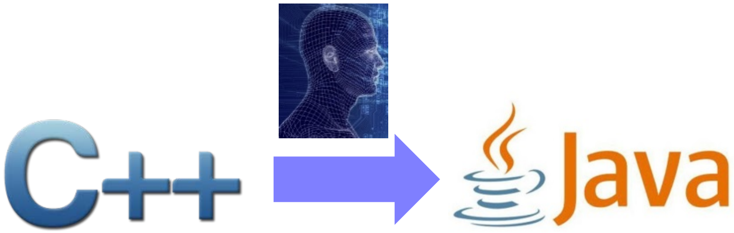
- **Assumption:** training and test data are assumed to be
  - Represented in the same feature space, AND
  - Follow the same data distribution
- **In practice:** training and test data come from different domains
  - Represented in different feature spaces, OR
  - Follow different data distributions



What if machines have transfer learning ability?

Transfer of Learning

- The study of dependency of human conduct, learning or performance on prior experience
  - [Thorndike and Woodworth, 1901] explored how individuals would transfer in one context to another context that share similar characteristics.



Transfer Learning

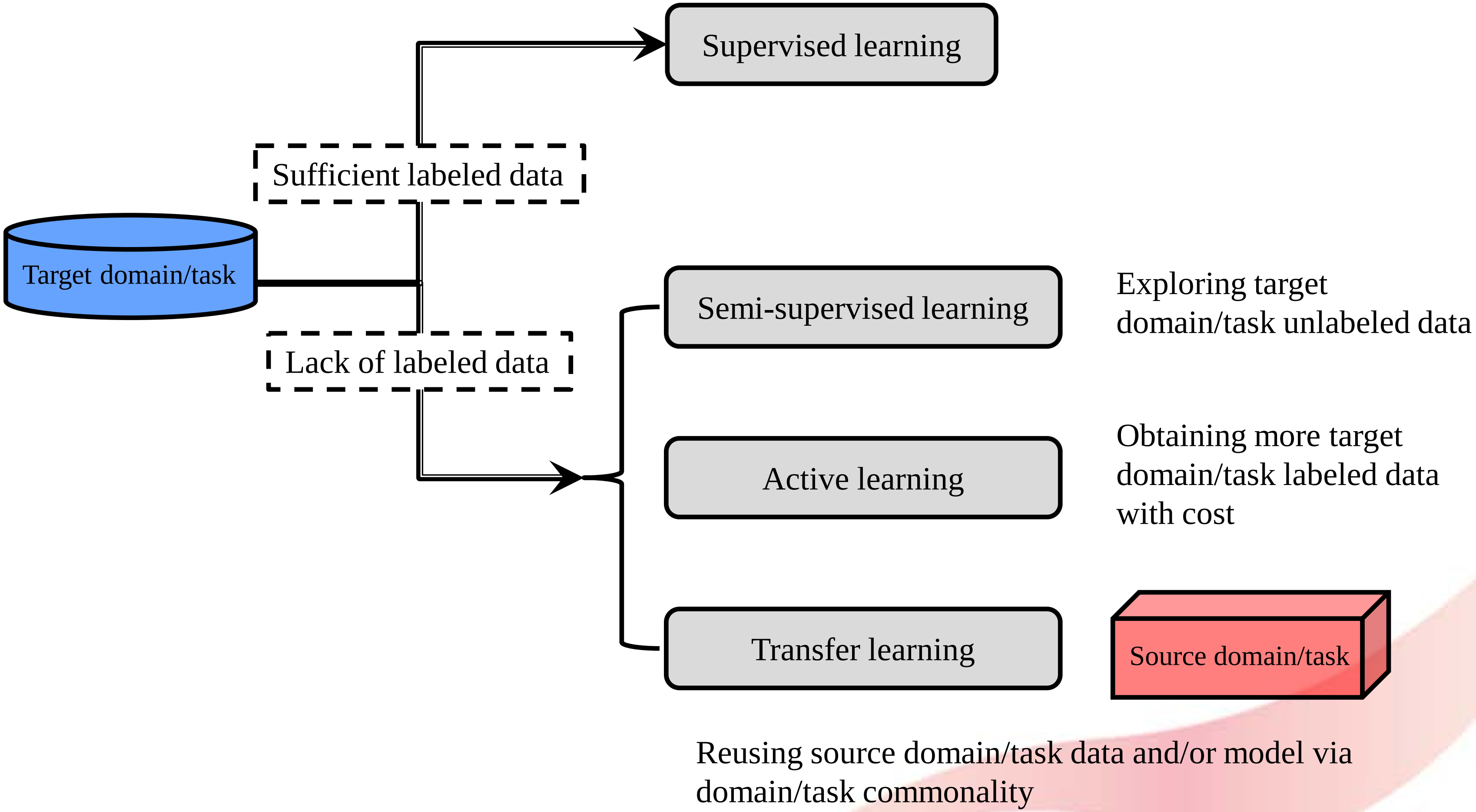
In mathematics

- Given: a small set of labeled data $T = \{\boldsymbol{x}_i, y_i\}$ for $i = 1, \dots, L$, for a target domain/task, and a set of plenty labeled data $S = \{\boldsymbol{x}_j, y_j\}$ for $j = 1, \dots, N$, from a source domain/task. In general, $L \ll N$
- Goal: to identify the commonality between the source data S and target data T, and utilize S to help learning a precise model $f: \boldsymbol{x} \to y$ from T
- The learned mapping f is expected to make precise predictions on any unseen $\boldsymbol{x}^*$ as $f(\boldsymbol{x}^*)$ for the target domain/task

Gene Disease Example

- We may have labeled data of other gene diseases
- Different gene diseases may have some correlations or share some common characteristics
- If the correlations/common characteristics can be automatically extracted and utilized, then labeled data from other gene disease may be used to help learning a classifier for the target gene disease
- Transfer learning can help！

Supervised Learning v.s. Advanced Paradigms



Target domain/task

Sufficient labeled data

Supervised learning

Lack of labeled data

Semi-supervised learning

Exploring target domain/task unlabeled data

Active learning

Obtaining more target domain/task labeled data with cost

Transfer learning

Source domain/task

Reusing source domain/task data and/or model via domain/task commonality

Different Learning Paradigms

- Supervised Learning ✓
- Unsupervised Learning ✓
- Reinforcement Learning ✗

AI6101 Introduction to AI and AI Ethics
OR AI6125 Multi-agent System

- Advanced paradigms:
  - Semi-supervised learning ✗

Semi-supervised learning literature survey

  - Active learning ✗

Active learning literature survey

  - Transfer learning ✓

Course Schedule (Tentative)

Date		Topics	Note
Week 1	11th Aug.	Introduction	
Week 2	18th Aug.	Data and Operations	
Week 3	25th Aug.	Linear models: regression	
Week 4	1st Sept.	Linear models: classification	Details of term paper released
Week 5	8th Sept.	Kernel methods	Assignment released (1 or 2 week(s) to finish)
Week 6	15th Sept.	Tree-based methods	
Week 7	22nd Sept.	Bayesian classifiers	
Recess Week	No class		
Week 8	6th Oct.	KNN + Ensemble learning	
Week 9	13th Oct.	Dimensionality reduction	
Week 10	20th Oct.	Clustering	
Week 11	27th Oct.	Density estimation	
Week 12	3rd Nov.	Recommender systems	
Week 13	10th Nov.	Transfer learning	In-person quiz! Scope: Weeks 2 – 11, ~1 hour open book Lecture will be pre-recorded
21st Nov.			Term paper submission deadline

Reference

- Reference:
  - **Introduction to Machine Learning (2nd Ed.)**, by Ethem Alpaydin, The MIT Press, 2010.
  - **Introduction to Data Mining**, by Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Addison Wesley, 2005.
  - **Pattern Recognition and Machine Learning**, by Christopher M. Bishop, Springer, 2006.
  - **Transfer Learning**, by Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan, Cambridge University Press, 2020.
- Regarding Mathematics:
  - Part I of the MIT Press book "*Deep Learning*" http://www.deeplearningbook.org/

Machine Learning Practice

- Important note: different from many other modules in the Master of Science in Artificial Intelligence (MSAI) programme, this machine learning module can be considered as an “applied mathematics” module, focusing on introducing the principles of different machine learning methods
- For practice:
  - Kaggle (highly recommended): http://www.kaggle.com/
  - UCI Repository: http://www.ics.uci.edu/~mlearn/MLRepository.html

Libraries and Platforms

- scikit-learn (Python):
  - http://scikit-learn.org/stable/
- Weka (Java)
  - http://www.cs.waikato.ac.nz/ml/weka/
- MALLET (Java)
  - http://mallet.cs.umass.edu/
- Tensorflow:
  - https://www.tensorflow.org/
- Pytorch:
  - https://pytorch.org/
- Many other libraries on deep learning
  - http://deeplearning.net/software_links/

Deep learning related, necessary in this module

Top-tier Academic Conferences

- International Conference on Machine Learning (ICML)
- Neural Information Processing Systems (NIPS)
- Conference on Learning Theory (COLT)

General machine learning

- Uncertainty in Artificial Intelligence (UAI)
- International Conference on AI & Statistics (AISTATS)

Statistical AI methods

- International Joint Conference on Artificial Intelligence (IJCAI)
- AAAI Conference on Artificial Intelligence (AAAI)

General AI

- International Conference on Learning Representations (ICLR)

Deep learning

- International Conference on Knowledge Discovery and Data Mining (KDD)

Data mining

Top-tier Academic Journals

- Journal of Machine Learning Research (JMLR)
- Machine Learning (MLJ)
- IEEE Transactions on Neural Networks and Learning Systems (TNNLS)

General machine learning

- IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)

Machine learning application in CV

- Artificial Intelligence (AIJ)
- Journal of Artificial Intelligence Research (JAIR)

General AI

Thank you!