Lecture 4: Machine Learning Basics

Shuai Li

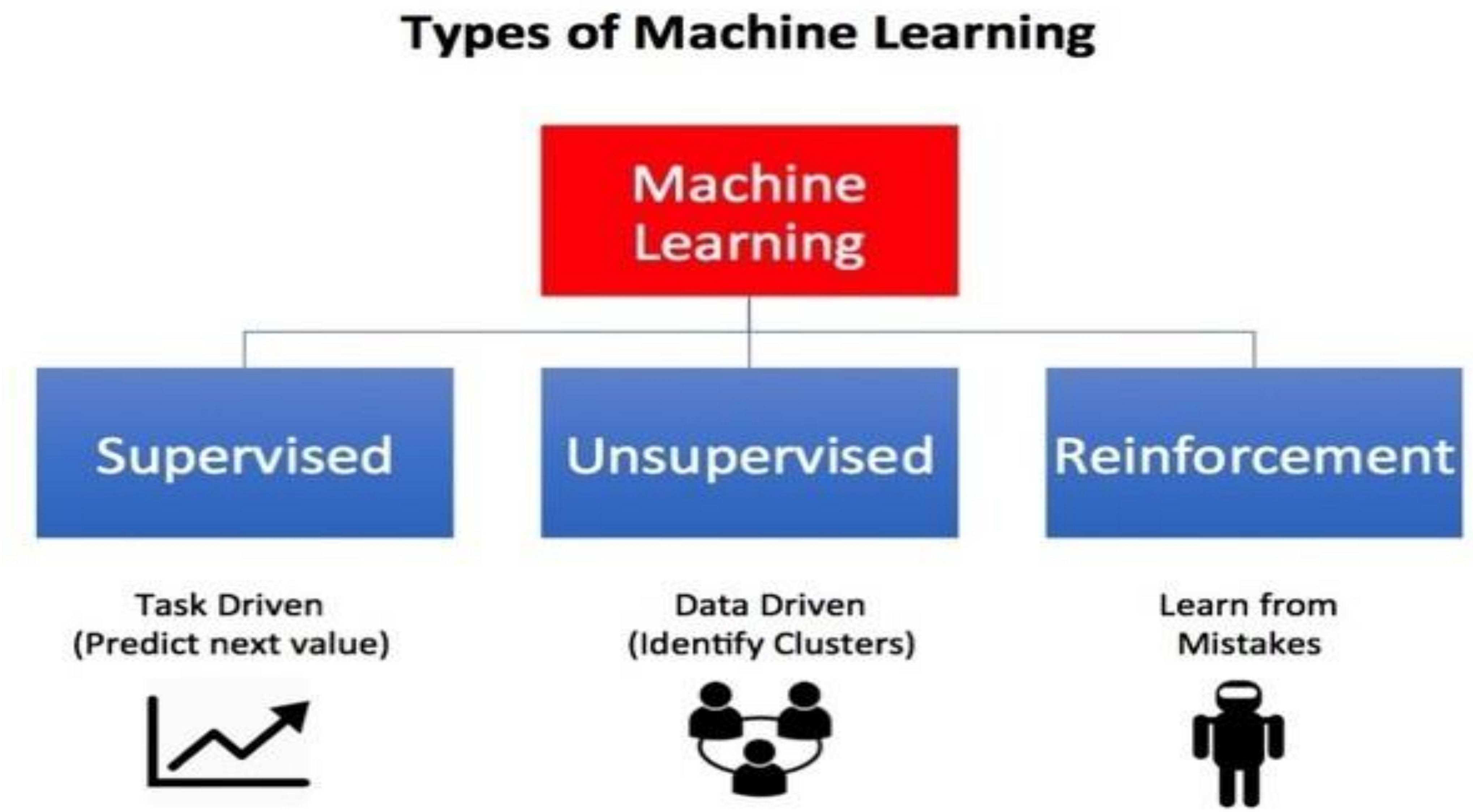
John Hopcroft Center, Shanghai Jiao Tong University

https://shuaili8.github.io

https://shuaili8.github.io/Teaching/CS410/index.html

Outline

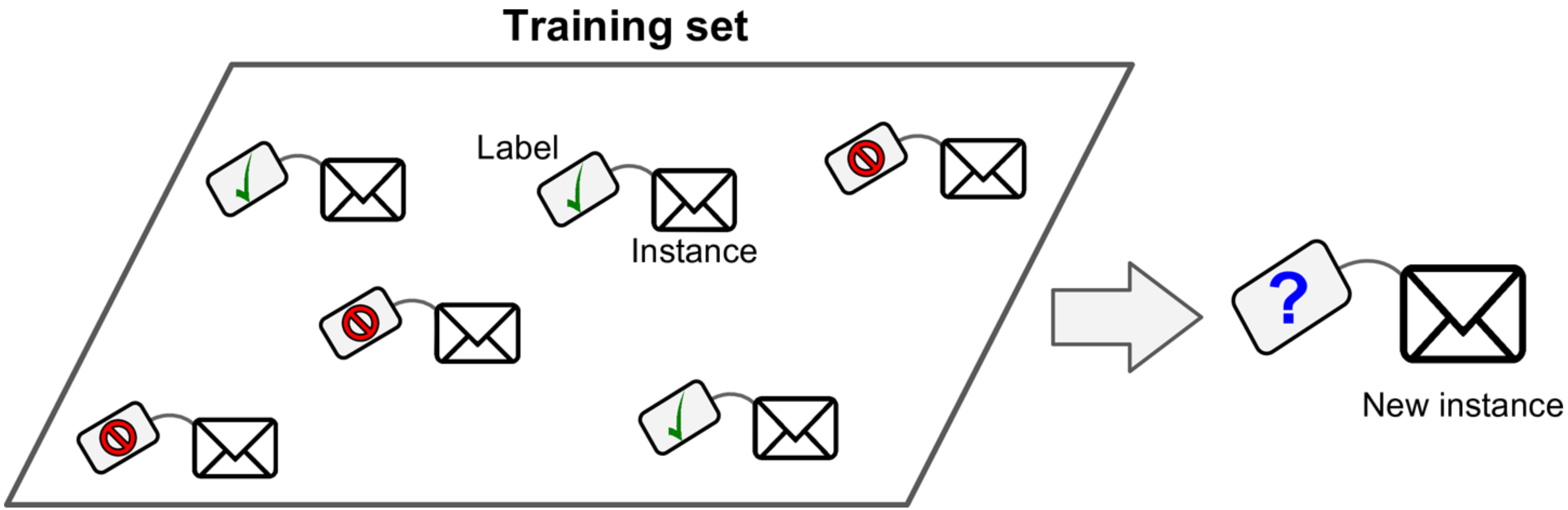
- The classification of machine learning
  - Supervised/unsupervised/reinforcement
- Supervised learning
  - Evaluation metrics for classification
    - Accuracy/Precision/Recall/F1 score
  - Model selection: bias/variance/generalization
  - Machine learning process


Types of Machine Learning
Machine Learning
Supervised
Unsupervised
Reinforcement
Task Driven
(Predict next value)
Data Driven
(Identify Clusters)
Learn from
Mistakes

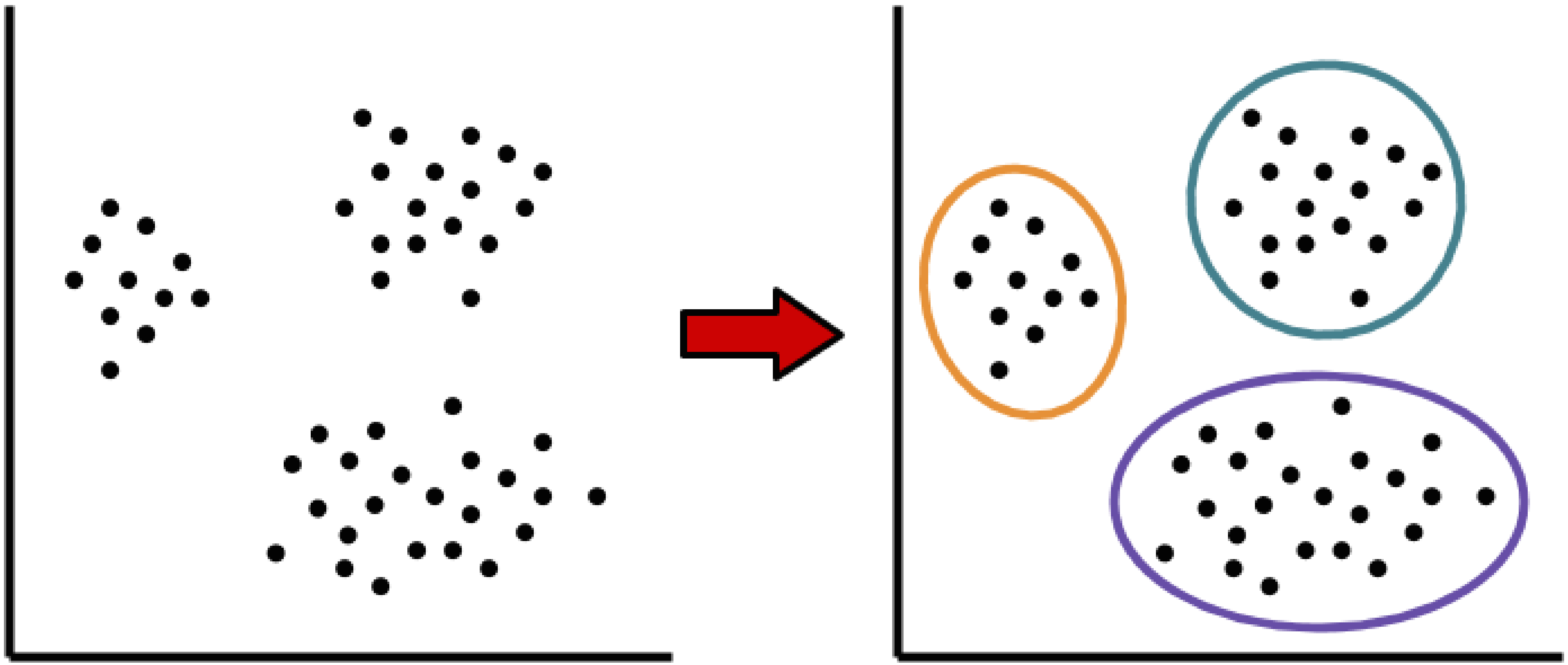
Machine Learning Categories

- Unsupervised learning
  - No labeled data
- Supervised learning
  - Use labeled data to predict on unseen points
- Semi-supervised learning
  - Use labeled data and unlabeled data to predict on unlabeled/unseen points
- Reinforcement learning
  - Sequential prediction and receiving feedbacks

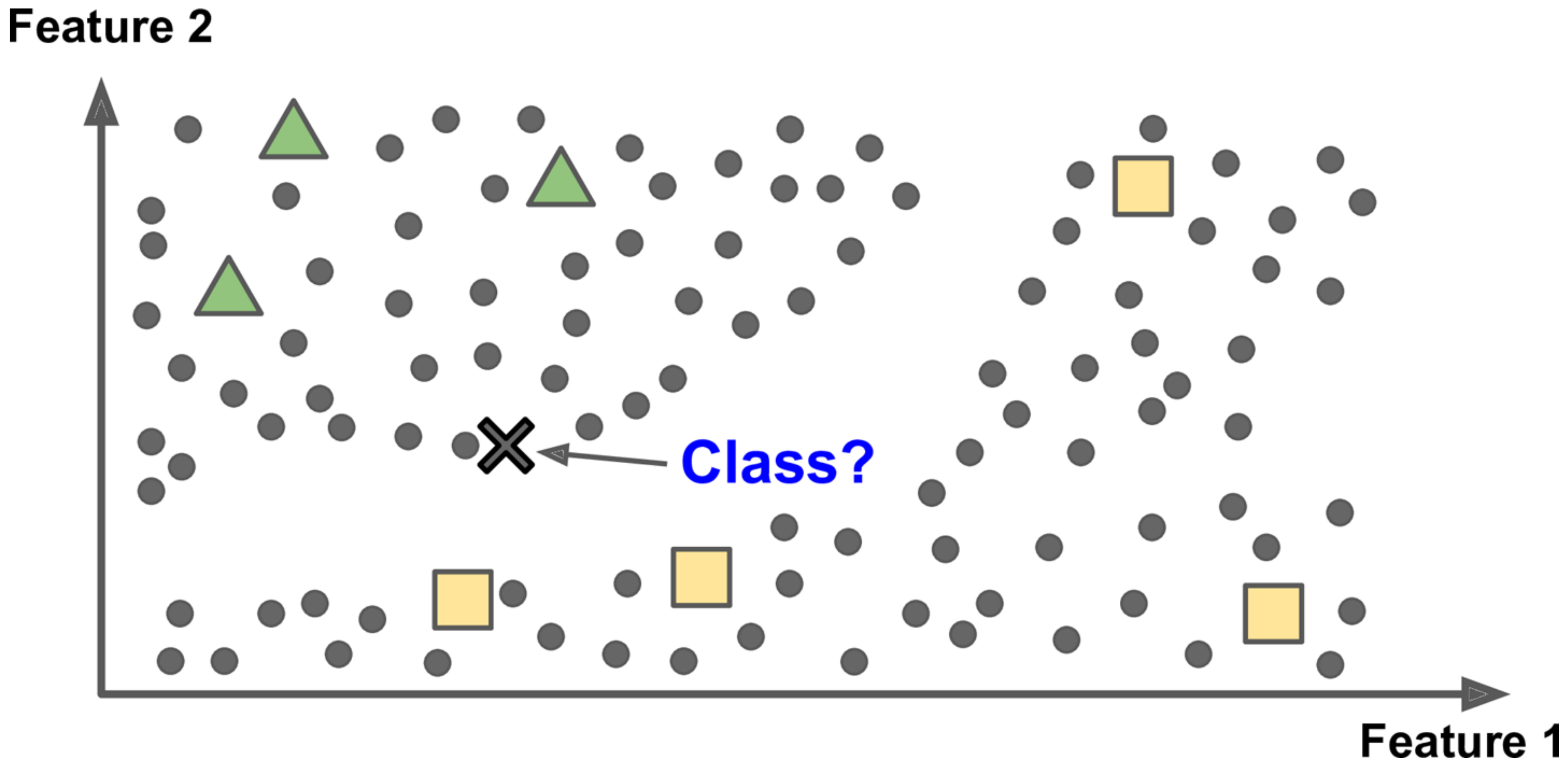
Supervised learning example



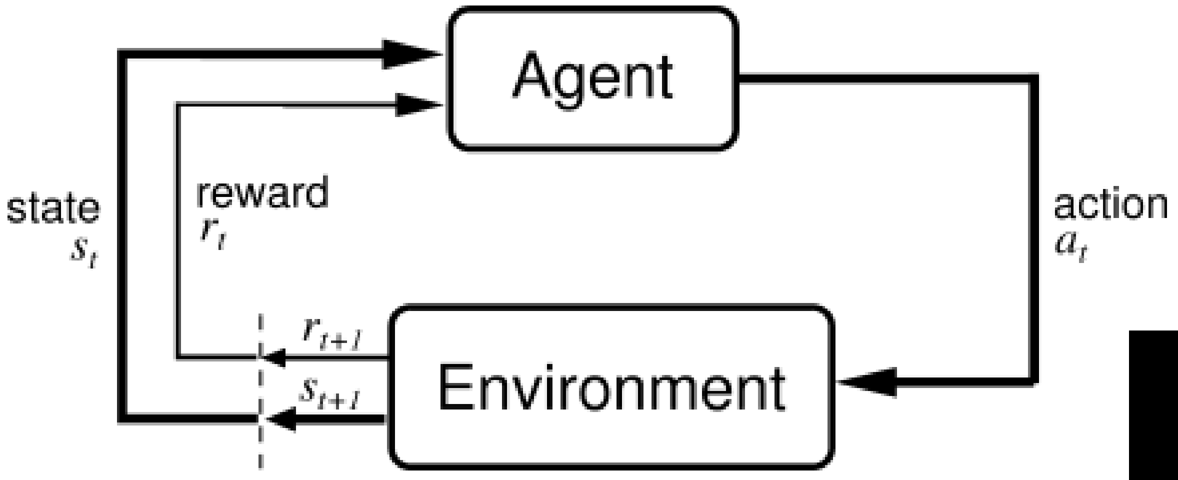
Unsupervised learning example

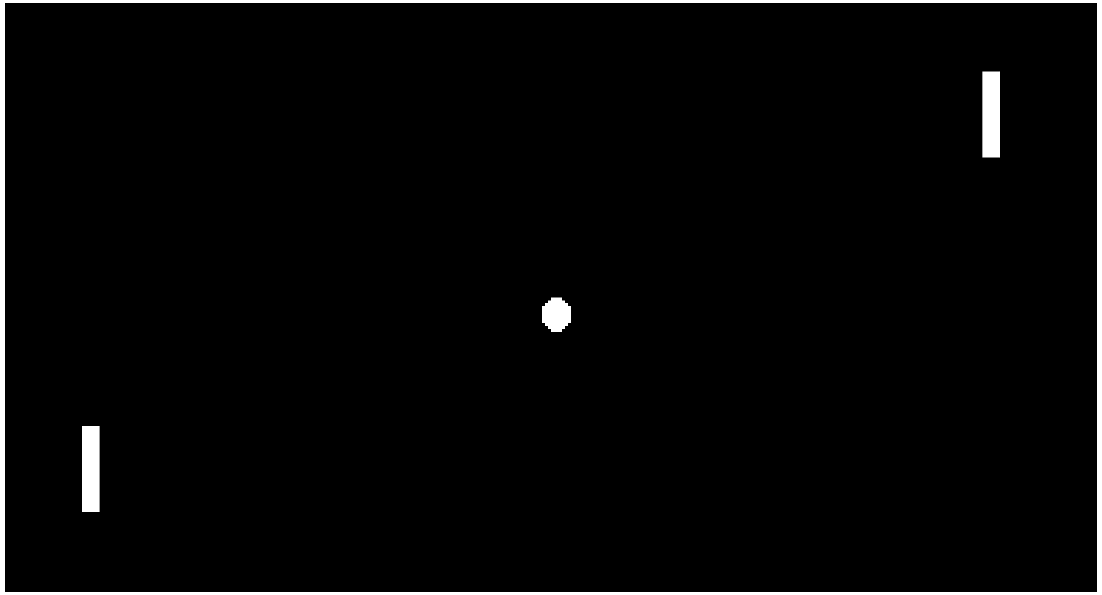


Semi-supervised learning example

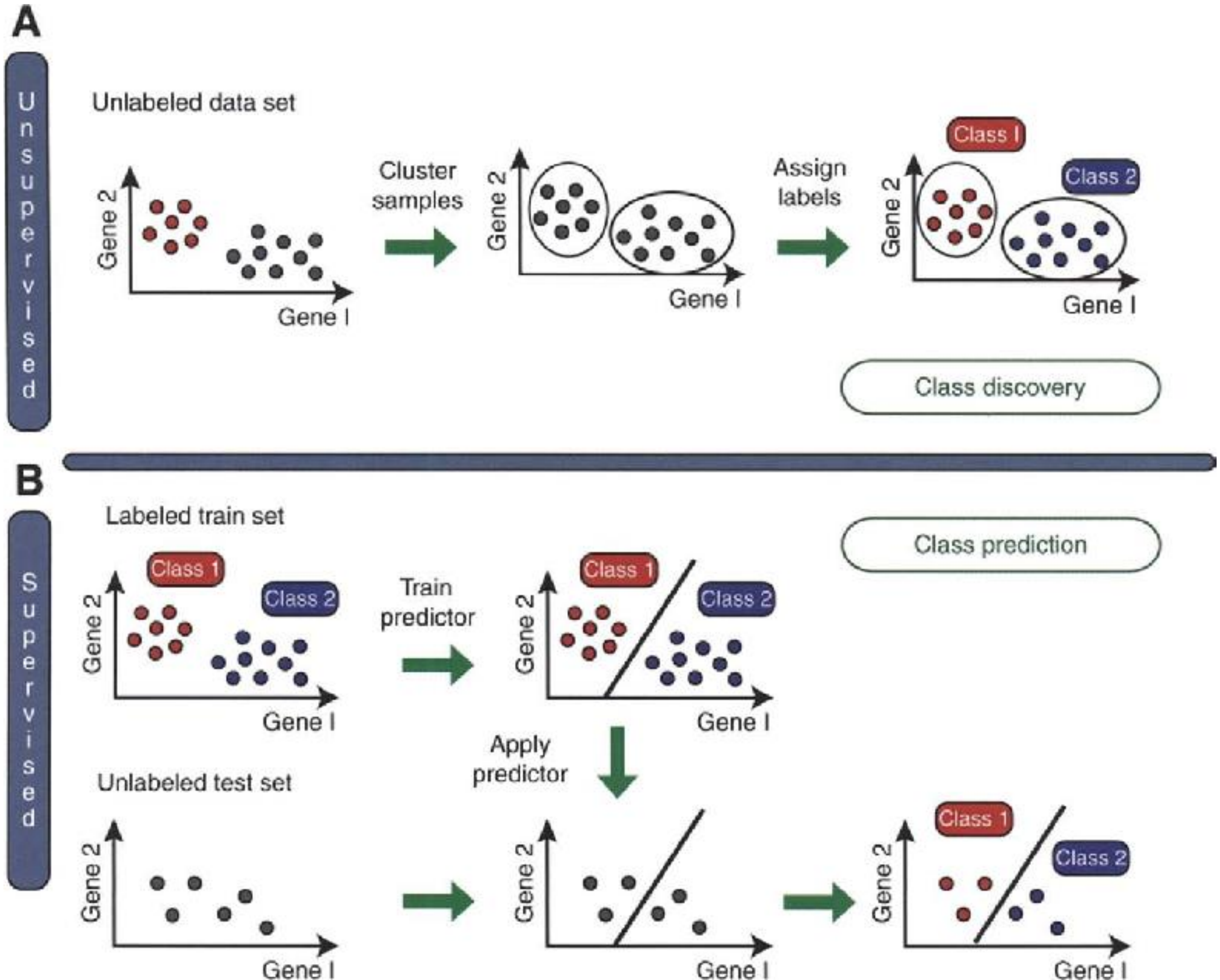


Reinforcement learning example

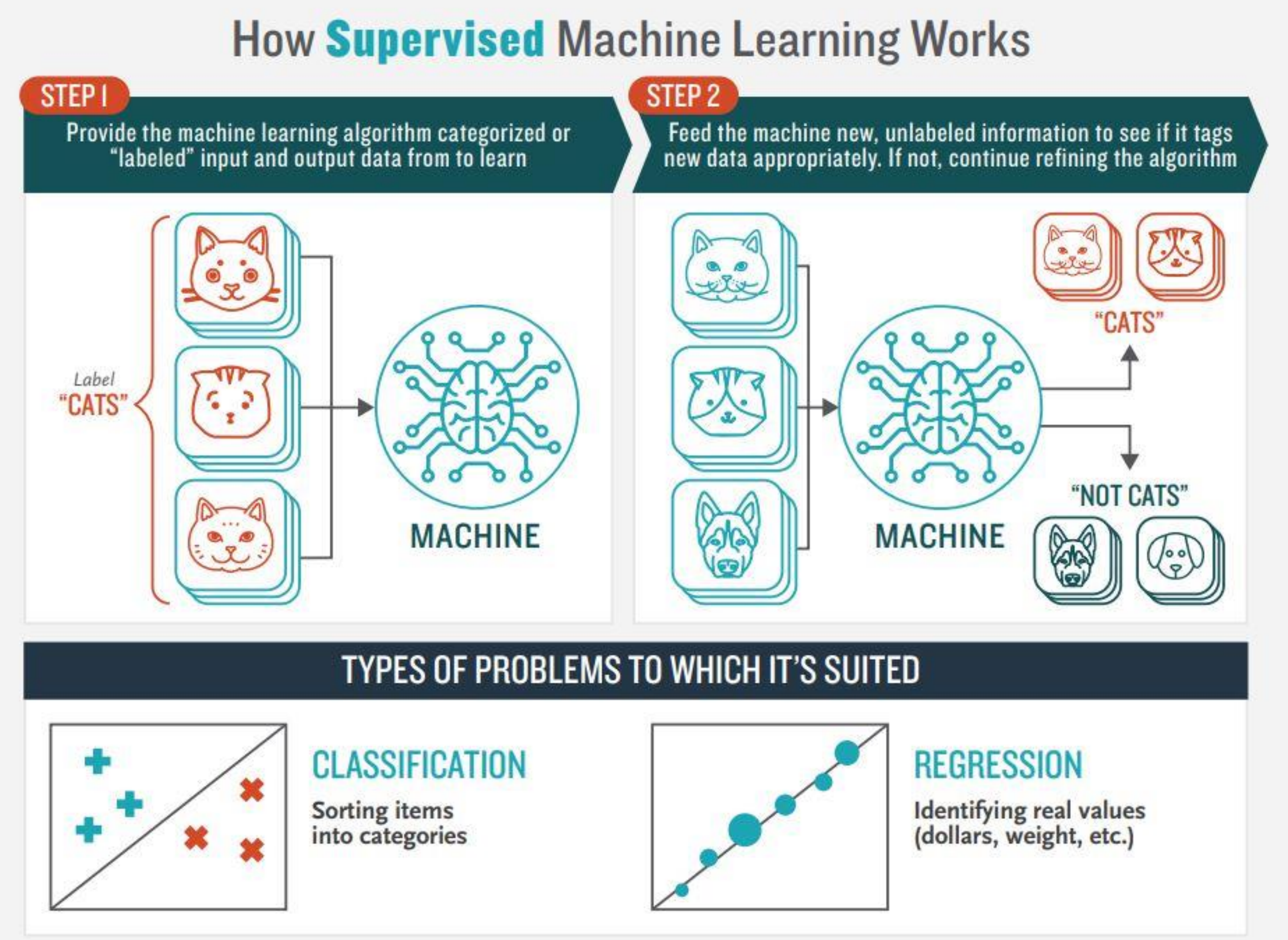


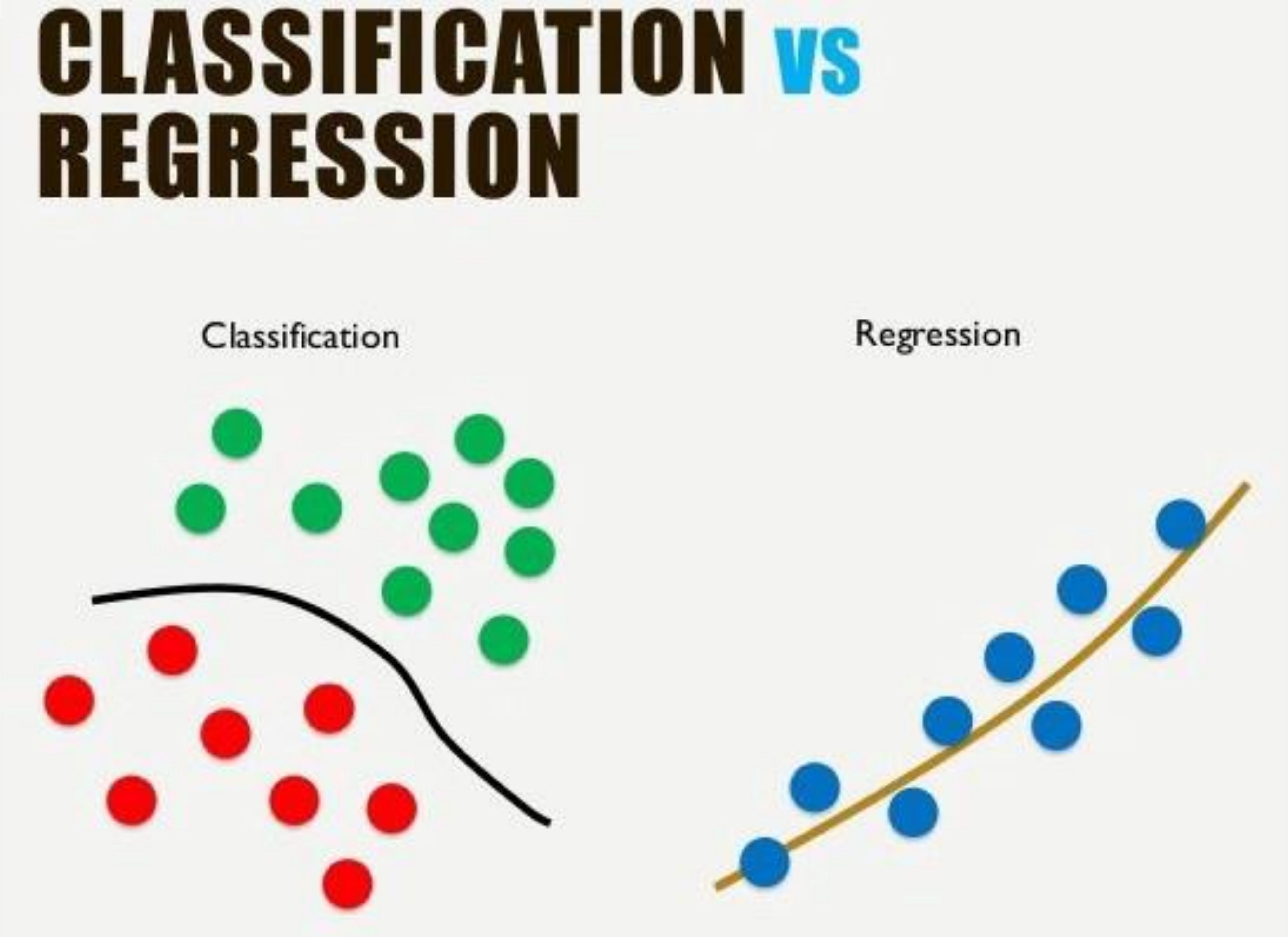


Supervised Learning	Unsupervised Learning
Input data is labelled	Input data is unlabeled
Uses training dataset	Uses just input dataset
Used for prediction	Used for analysis
Classification and regression	Clustering, density estimation and dimensionality reduction


A
Unsupervised
Unlabeled data set
Gene 2
Gene I
Cluster samples
Gene 2
Gene I
Assign labels
Class I
Class 2
Gene 2
Gene I
Class discovery
B
Supervised
Labeled train set
Class 1
Class 2
Gene 2
Gene I
Train predictor
Class 1
Class 2
Gene 2
Gene I
Class prediction
Apply predictor
Unlabeled test set
Gene 2
Gene I
Gene 2
Gene I
Class 1
Class 2
Gene 2
Gene I

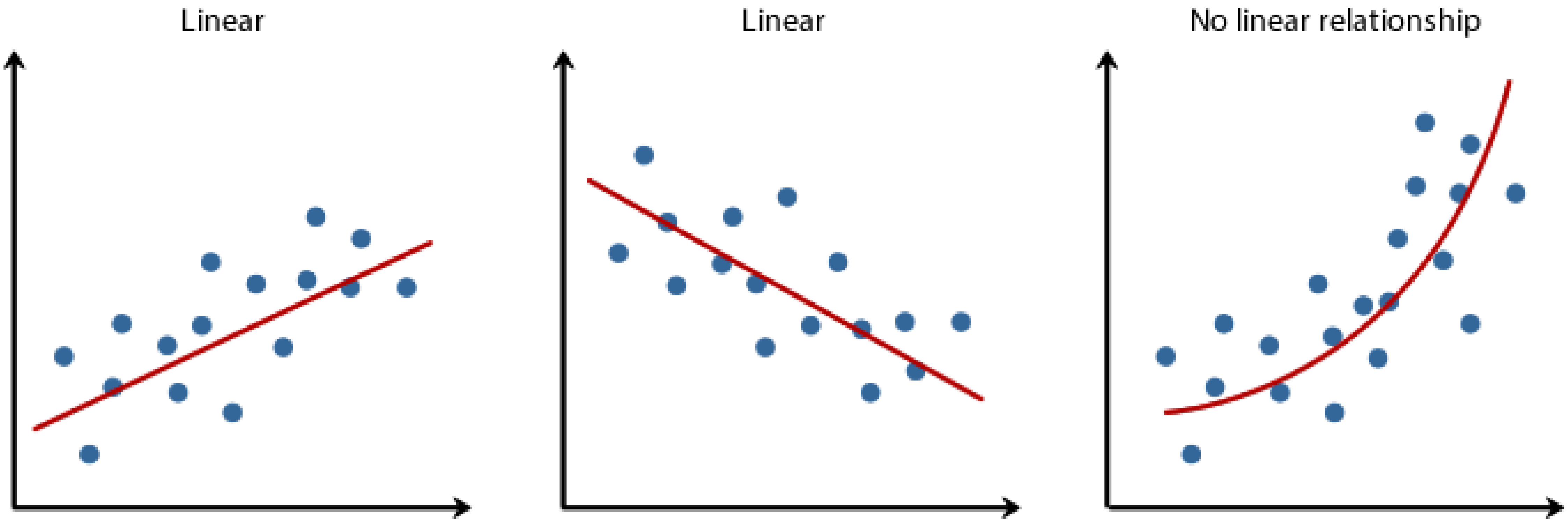
Supervised Learning


How Supervised Machine Learning Works
STEP 1
Provide the machine learning algorithm categorized or "labeled" input and output data from to learn
Label
"CATS"
MACHINE
STEP 2
Feed the machine new, unlabeled information to see if it tags new data appropriately. If not, continue refining the algorithm
MACHINE
"CATS"
"NOT CATS"
TYPES OF PROBLEMS TO WHICH IT'S SUITED
CLASSIFICATION
Sorting items into categories
REGRESSION
Identifying real values (dollars, weight, etc.)


CLASSIFICATION VS
REGRESSION
Classification
Regression

Classification -- Handwritten digits

Regression example



Copyright 2014. Laerd Statistics.

Model Evaluations for Classification

Confusion Matrix

- Confusion Matrix
  - TP – True Positive ; FP – False Positive
  - FN – False Negative; TN – True Negative

		Predicted Class	
Actual Class		Class = Yes	Class = No
	Class = Yes	a (TP)	b (FN)
	Class = No	c (FP)	d (TN)

$$\text{Accuracy} = \frac{a + d}{a + b + c + d} = \frac{TP + TN}{TP + TN + FP + FN}$$

Confusion Matrix 2

- Given a set of records containing positive and negative results, the computer is going to classify the records to be positive or negative.

- Positive: The computer classifies the result to be positive
- Negative: The computer classifies the result to be negative
- True: What the computer classifies is true
- False: What the computer classifies is false

Limitation of Accuracy

- Limitation of Accuracy
  - Consider a 2-class problem
    - Number of Class 0 examples = 9990
    - Number of Class 1 examples = 10
  - If a “stupid” model predicts everything to be class 0, accuracy is 9990/10000 = **99.9** %

- The accuracy is misleading because the model does not detect any example in class 1

Other measures

- Cost-sensitive measures

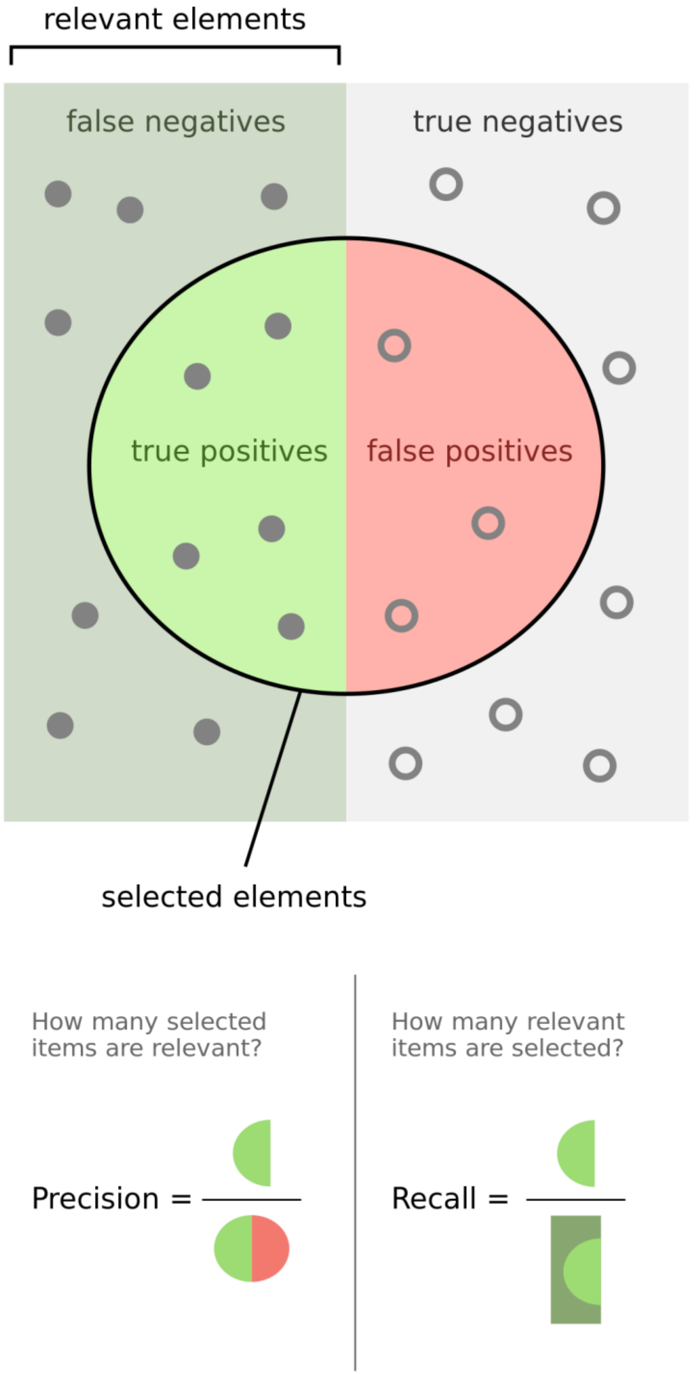
	Predicted Class	
Actual Class	Class = Yes	Class = No
Class = Yes	a (TP)	b (FN)
Class = No	c (FP)	d (TN)

$$\text{Precision (p)} = \frac{TP}{TP + FP} = \frac{a}{a + c}$$

$$\text{Recall (r)} = \frac{TP}{TP + FN} = \frac{a}{a + b}$$

$$\text{F - measure (F)} = \frac{2rp}{r + p} = \frac{2a}{2a + b + c}$$

Harmonic mean of Precision and Recall
(Why not just average?)


relevant elements
false negatives
true negatives
true positives
false positives
selected elements
How many selected items are relevant?
How many relevant items are selected?
Precision =
Recall =

How to understand

- A school is running a machine learning primary diabetes scan on all of its students
  - Diabetic (+) / Healthy (-)
  - False positive is just a false alarm
  - False negative
    - Prediction is healthy but is diabetic
    - Worst case among all 4 cases

- Accuracy
  - Accuracy = (TP+TN)/(TP+FP+FN+TN)
  - How many students did we correctly label out of all the students?

How to understand (cont.)

- A school is running a machine learning primary diabetes scan on all of its students
    - Diabetic (+) / Healthy (-)
    - False positive is just a false alarm
    - False negative
        - Prediction is healthy but is diabetic
        - Worst case among all 4 cases

- Precision
    - Precision = TP/(TP+FP)
    - How many of those who we labeled as diabetic are actually diabetic?

How to understand (cont.)

- A school is running a machine learning primary diabetes scan on all of its students
  - Diabetic (+) / Healthy (-)
  - False positive is just a false alarm
  - False negative
    - Prediction is healthy but is diabetic
    - Worst case among all 4 cases

- Recall (sensitivity)
  - Recall = TP/(TP+FN)
  - Of all the people who are diabetic, how many of those we correctly predict?

F1 score (F-Score / F-Measure)

- F1 Score = 2*(Recall * Precision) / (Recall + Precision)
- Harmonic mean (average) of the precision and recall
- F1 Score is best if there is some sort of balance between precision (p) & recall (r) in the system. Oppositely F1 Score isn’t so high if one measure is improved at the expense of the other.
- For example, if P is 1 & R is 0, F1 score is 0.

Which to choose

- Accuracy
  - A great measure
  - But only when you have symmetric datasets (FN & FP counts are close)
  - Also, FN & FP have similar costs
- F1 score
  - If the cost of FP and FN are different
  - F1 is best if you have an uneven class distribution
- Recall
  - If FP is far better than FN or if the occurrence of FN is unaccepted/intolerable
  - Would like more extra FP (false alarms) over saving some FN
  - E.g. diabetes. We'd rather get some healthy people labeled diabetic over leaving a diabetic person labeled healthy
- Precision
  - Want to be more confident of your TP
  - E.g. spam emails. We'd rather have some spam emails in inbox rather than some regular emails in your spam box.

Example

- Given 30 human photographs, a computer predicts 19 to be male, 11 to be female. Among the 19 male predictions, 3 predictions are not correct. Among the 11 female predictions, 1 prediction is not correct.

	Predicted Class		
Actual Class		Male	Female
	Male	a = TP = 16	b = FN = 1
	Female	c = FP = 3	d = TN = 10

Example

	Predicted Class		
Actual Class		Male	Female
	Male	a = TP = 16	b = FN = 1
	Female	c = FP = 3	d = TN = 10

- Accuracy = (16 + 10) / (16 + 3 + 1 + 10) = 0.867
- Precision = 16 / (16 + 3) = 0.842
- Recall = 16 / (16 + 1) = 0.941
- F-measure = 2 (0.842)(0.941) / (0.842 + 0.941)

  = 0.889

Discussion

- “In a specific case, precision cannot be computed.” Is the statement true? Why?
- If the statement is true, can F-measure be computed in that case?

	a	b	c
a	TP	FN	FN
b	FP	TN	TN
c	FP	TN	TN

←Classified as

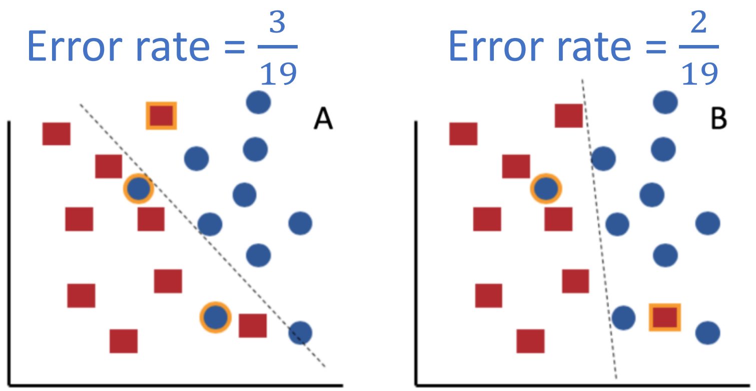
a: positive
b: negative
c: negative

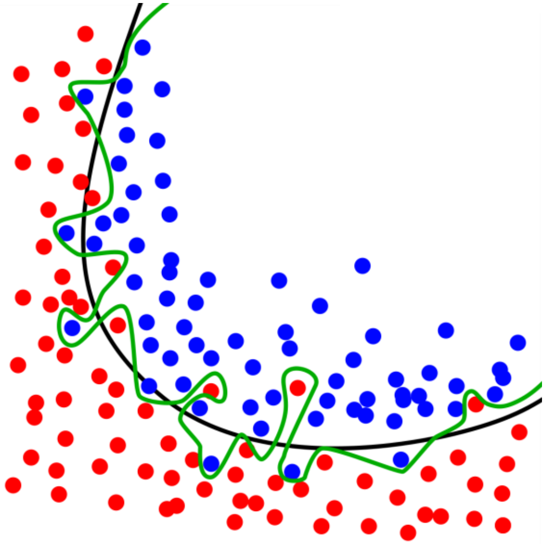
- How about if b is positive, a and c are negative, or if c is positive, a and b are negative ?

Model Selections

Minimize the error rate?

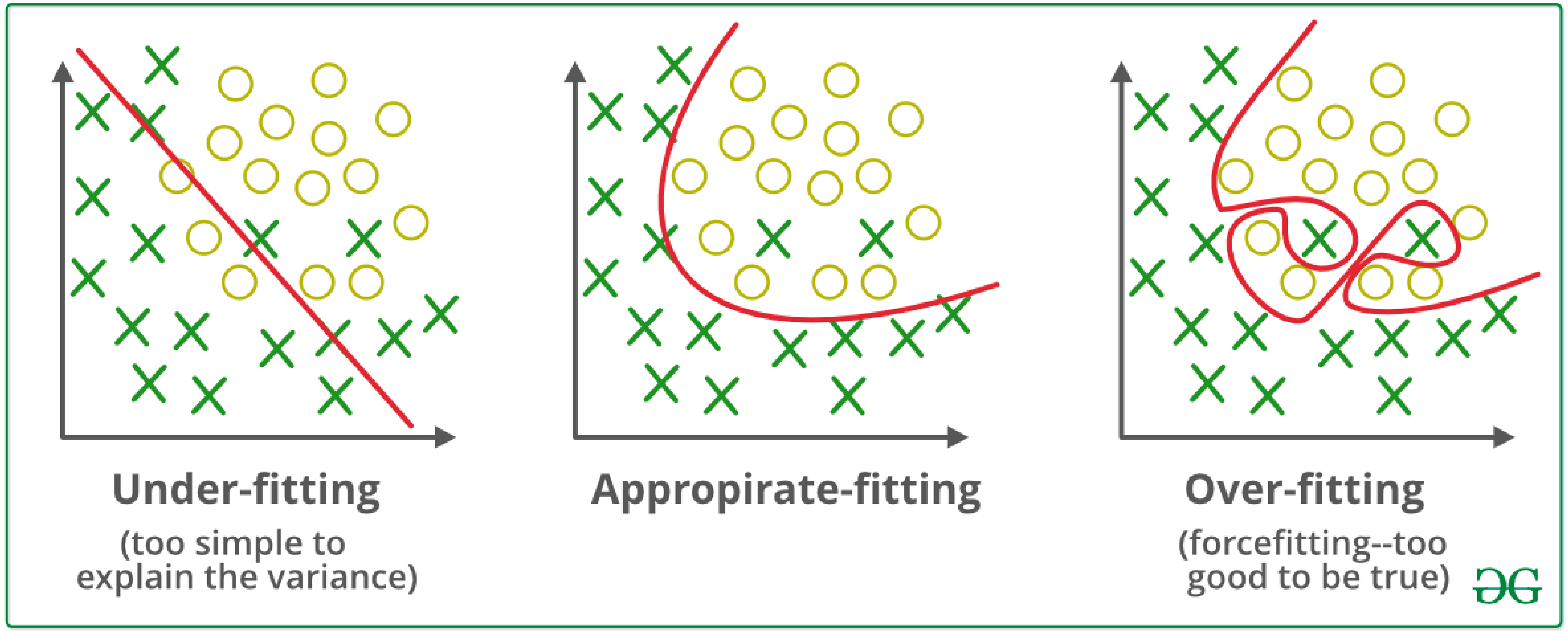
- Given a data set S
- Error rate = $\dfrac{\#\text{ of Errors}}{\#\text{ of Total Samples}}$
- Accuracy = 1 - Error rate





https://malware.news/uploads/default/original/3X/6/d/6df12e50b7f97cdba92697ce164cbe4a5502a349.png
https://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Overfitting.svg/1200px-Overfitting.svg.png

Fitting



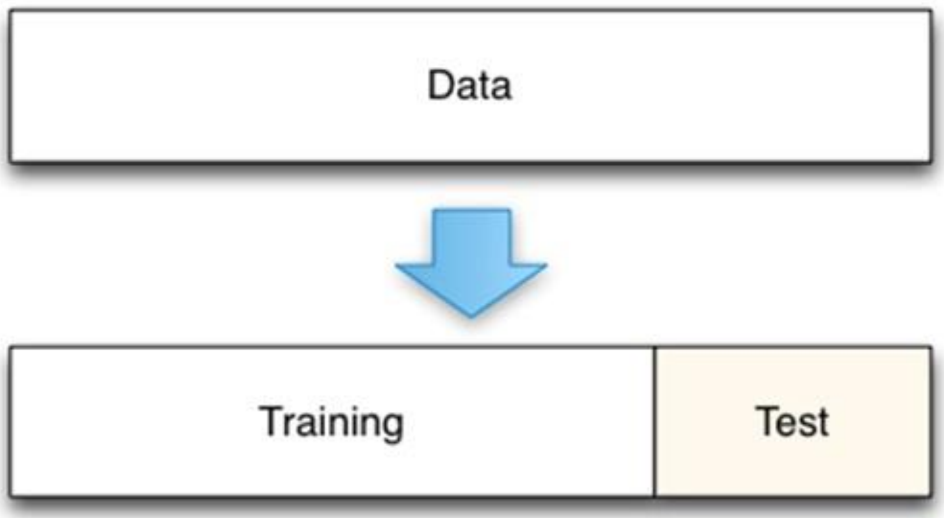
https://media.geeksforgeeks.org/wp-content/cdn-uploads/20190523171258/overfitting_2.png

Split training and test

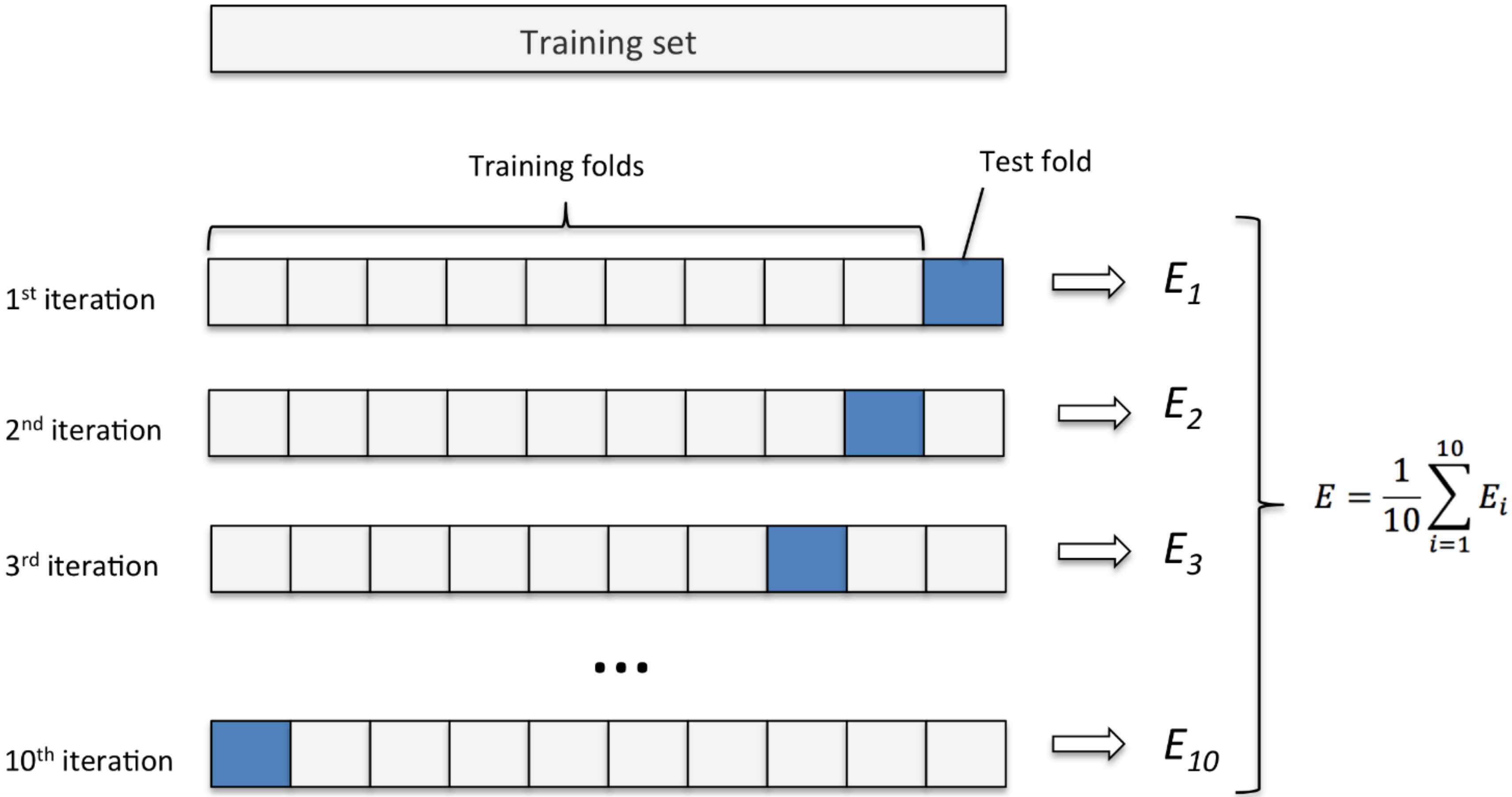
- Split dataset to training and test

- Train models on training dataset
- The evaluation of the model is the error on test dataset

- Might overfit the training dataset

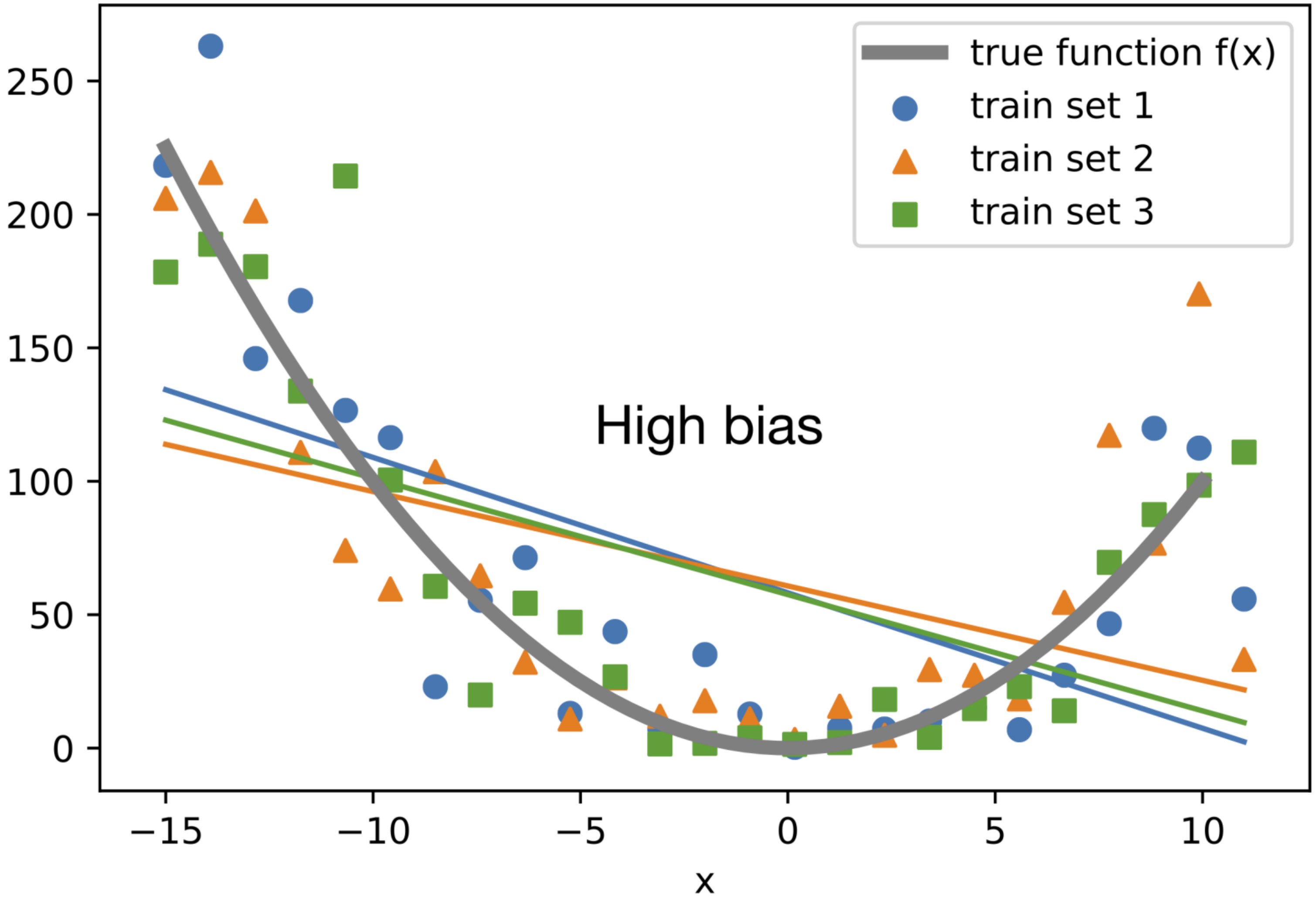


Cross validation

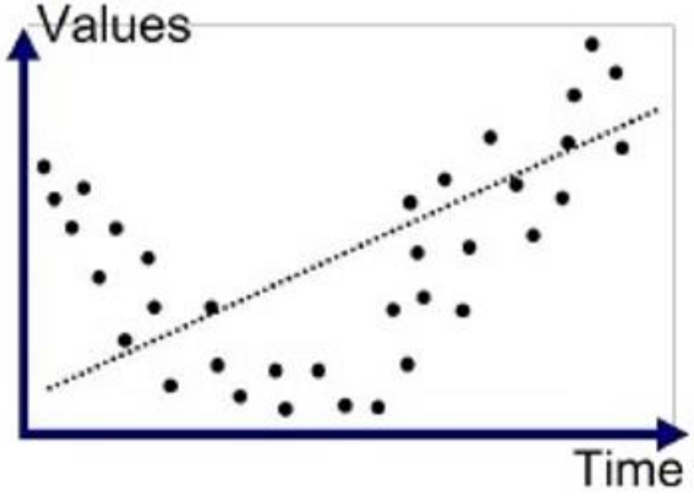


Bias

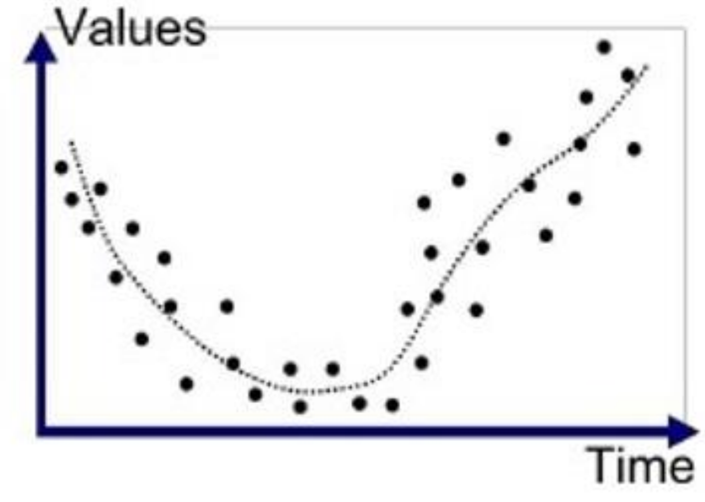
$$\text{Bias} = E[\hat{\theta}] - \theta.$$



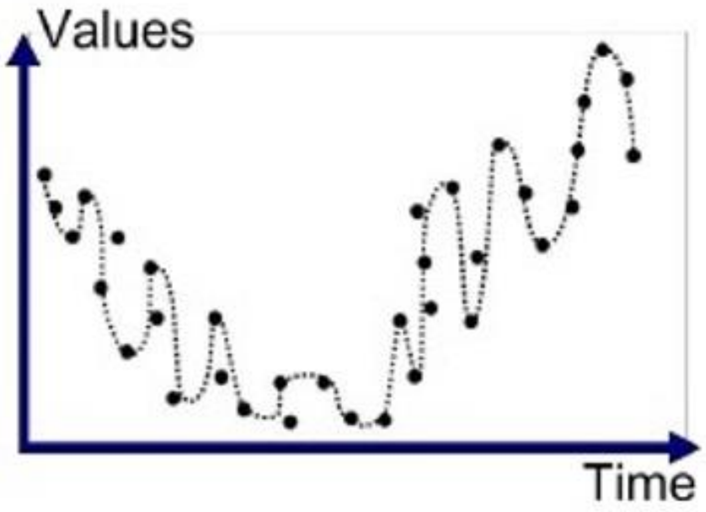
Underfitting



Underfitted



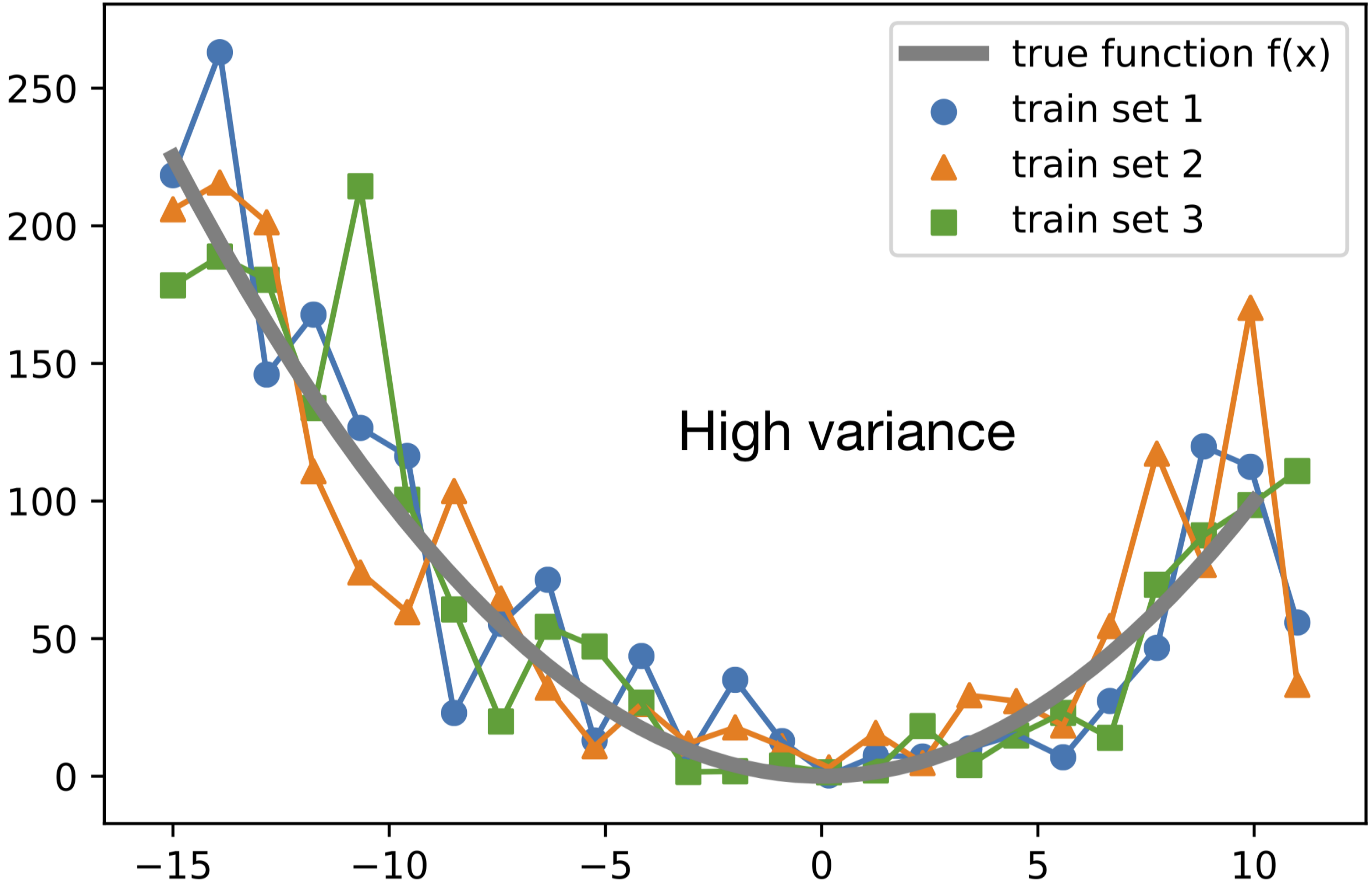
Good Fit/Robust



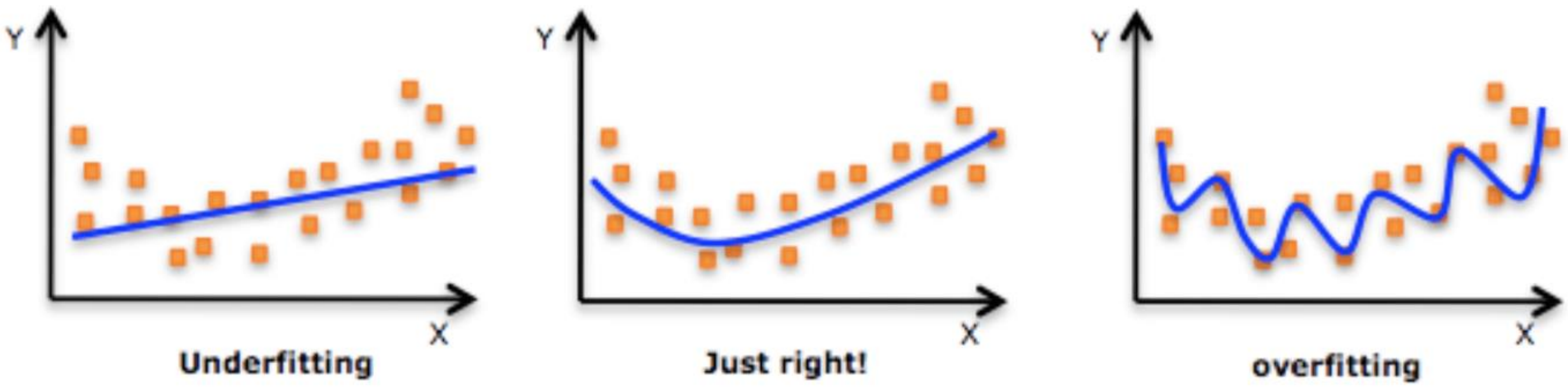
Overfitted

Variance

$$\mathrm{Var}(\hat{\theta}) = E[(E[\hat{\theta}] - \hat{\theta})^2].$$



Overfitting



Bias-variance decomposition

True value

Estimated value

$$S = (y - \hat{y})^2$$

$$\begin{aligned}(y - \hat{y})^2 &= (y - E[\hat{y}] + E[\hat{y}] - \hat{y})^2 \\ &= (y - E[\hat{y}])^2 + (E[\hat{y}] - y)^2 + 2(y - E[\hat{y}])(E[\hat{y}] - \hat{y}).\end{aligned}$$

$$\begin{aligned}E[2(y - E[\hat{y}])(E[\hat{y}] - \hat{y})] &= 2E[(y - E[\hat{y}])(E[\hat{y}] - \hat{y})] \\ &= 2(y - E[\hat{y}])E[(E[\hat{y}] - \hat{y})] \\ &= 2(y - E[\hat{y}])(E[E[\hat{y}]] - E[\hat{y}]) \\ &= 2(y - E[\hat{y}])(E[\hat{y}] - E[\hat{y}]) \\ &= 0.\end{aligned}$$

$$E[S] = E[(y - \hat{y})^2]$$

$$\begin{aligned}E[(y - \hat{y})^2] &= (y - E[\hat{y}])^2 + E[(E[\hat{y}] - \hat{y})^2] \\ &= [\text{Bias}]^2 + \text{Variance}.\end{aligned}$$

Can be understood by interpreting y and $\hat{y}$ as outputs from model θ and $\hat{\theta}$

Training vs. Generalization Error

- Training error: $$E_{train} = \frac{1}{n}\sum_{\underbrace{i=1}_{\text{training examples}}}^{n} \overbrace{error}^{\text{same? different by how much?}}(\underbrace{f_D(\mathbf{x}_i)}_{\text{value we predicted}}, \underbrace{y_i}_{\text{true value}})$$
- Generalization error:
  - how well we will do on future data
  - don't know what future data x_i will be
  - don't know what labels y_i it will have
  - but know the "range" of all possible $\{x,y\}$
    - x: all possible 20x20 black/white bitmaps
    - y: {0,1,…,9} (digits)

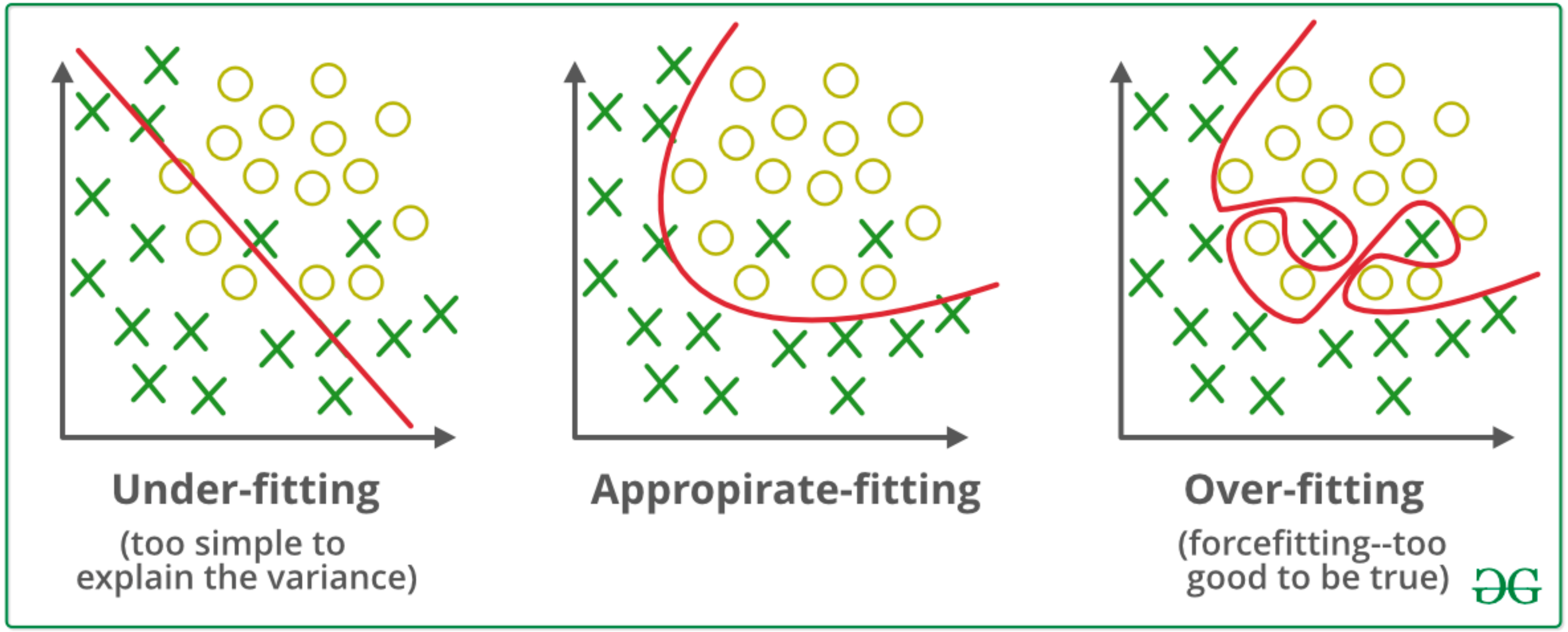
Usually
$E_{train} \leq E_{gen}$

$$E_{gen} = \underbrace{\int}_{\text{over all possible x,y}} \underbrace{error(f_D(\mathbf{x}), y)}_{\text{error as before}} \underbrace{p(y, \mathbf{x})}_{\text{how often we expect to see such x and y}} d\mathbf{x}$$

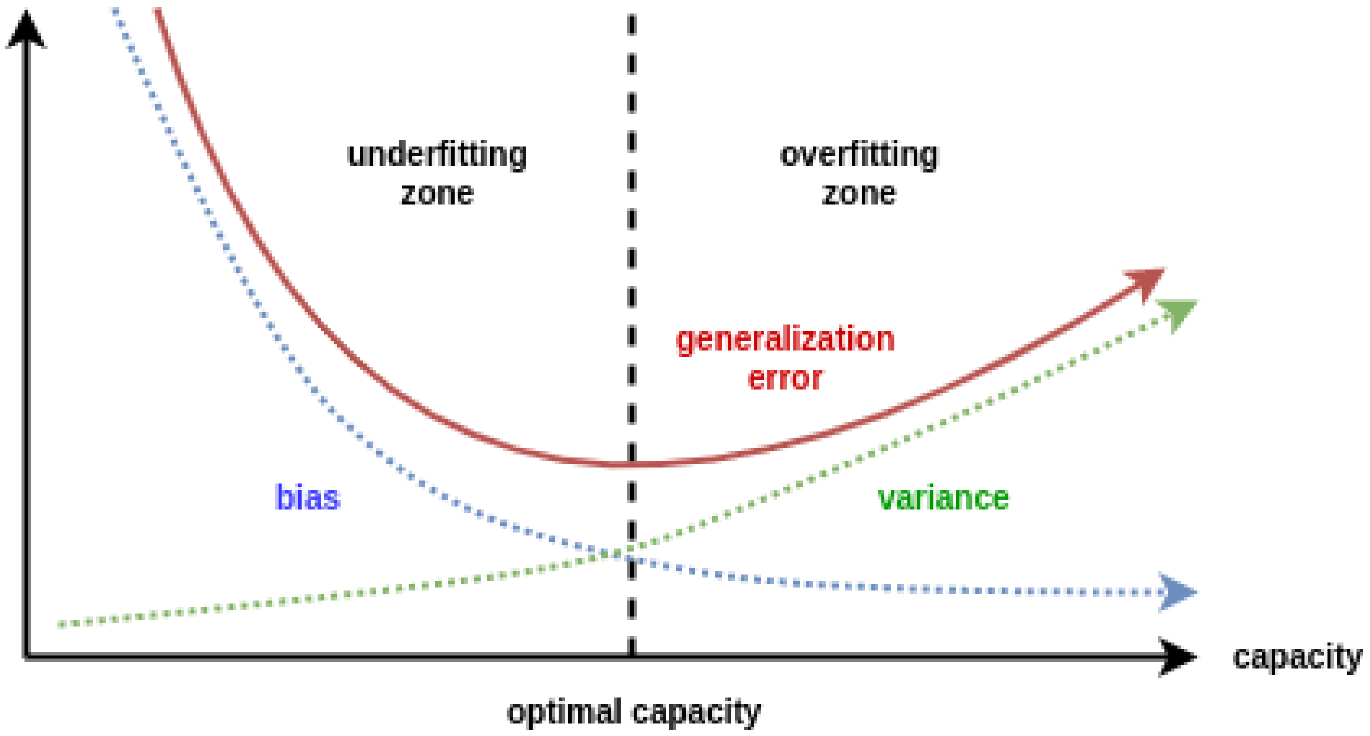
Can never compute generalisation error

Generalization

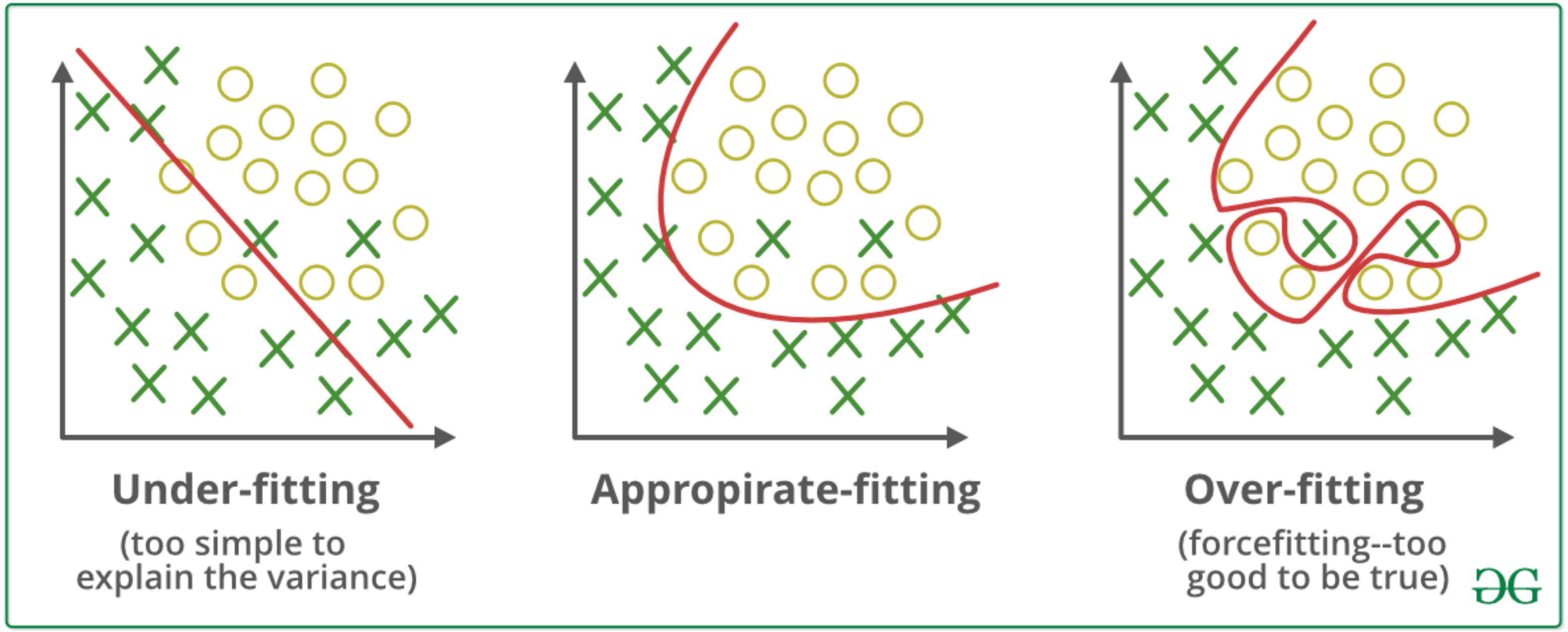
- Observations:
  - The best hypothesis on the sample may not be the best overall
  - Complex rules (very complex separation surfaces) can be poor predictors
  - trade-off: complexity of hypothesis set vs sample size (underfitting/overfitting)



Balance bias-variance trade-off

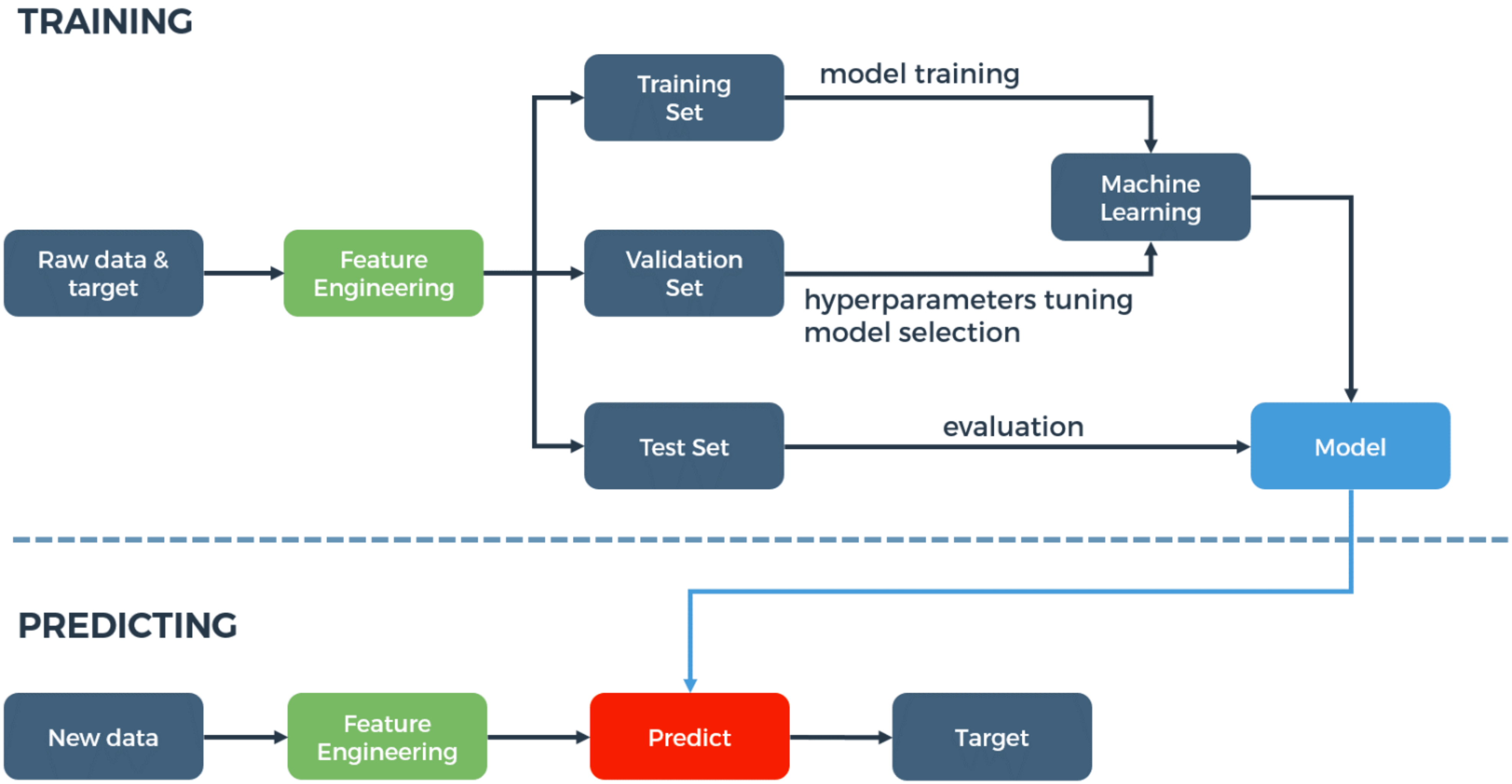


Learning ≠ Fitting



- Notion of simplicity/complexity
- How to define complexity
- Model selection

Machine Learning Process

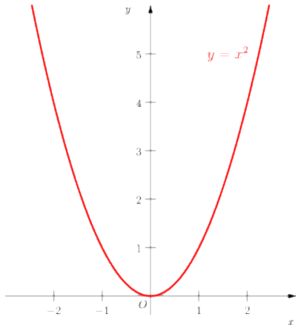


https://techblog.cdiscount.com/assets/images/DataScience/automl/ML_process.png

Problem Formulation

Problem Definition

- Spaces:
  - Input space (feature space) X, output space (labeled space) Y
- Loss function: $L: Y \times Y \to \mathbb{R}$
  - $L(\hat{y}, y)$: loss of predicting $\hat{y}$ when the true output is y
  - Binary classification: $L(\hat{y}, y) = 1_{\hat{y} \neq y}$
  - Regression: $L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$



- Hypothesis set: $H \subseteq Y^X$(mappings from X to Y)
  - Space of possible models, e.g. all linear functions
  - Depends on feature structure and prior knowledge about the problem

Set-up

- Training data:
  - Sample S of size N drawn i.i.d. from $\mathrm{X} \times Y$ according to distribution D:
$$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$
- Objective:
  - Find hypothesis $h \in H$ with small generalization error

- Generalization error
$$R(h) = \mathbb{E}_{(x,y)\sim D}[L(h(x), y)]$$
- Empirical error
$$\hat{R}(h) = \frac{1}{N}\sum_{i=1}^{N} L(h(x_i), y_i)$$

Model Selection

- For any $h \in H$

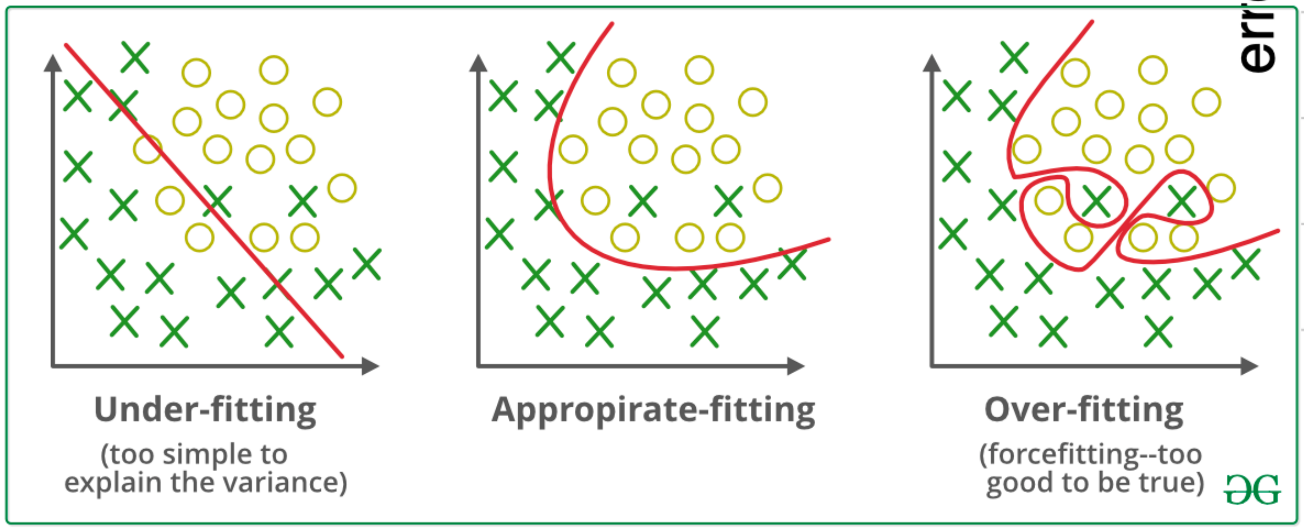
$$R(h) - \min_{h'} R(h') = \overbrace{\left(R(h) - \min_{h' \in H} R(h')\right)}^{\text{estimation}} + \overbrace{\left(\min_{h' \in H} R(h') - \min_{h'} R(h')\right)}^{\text{approximation}}$$

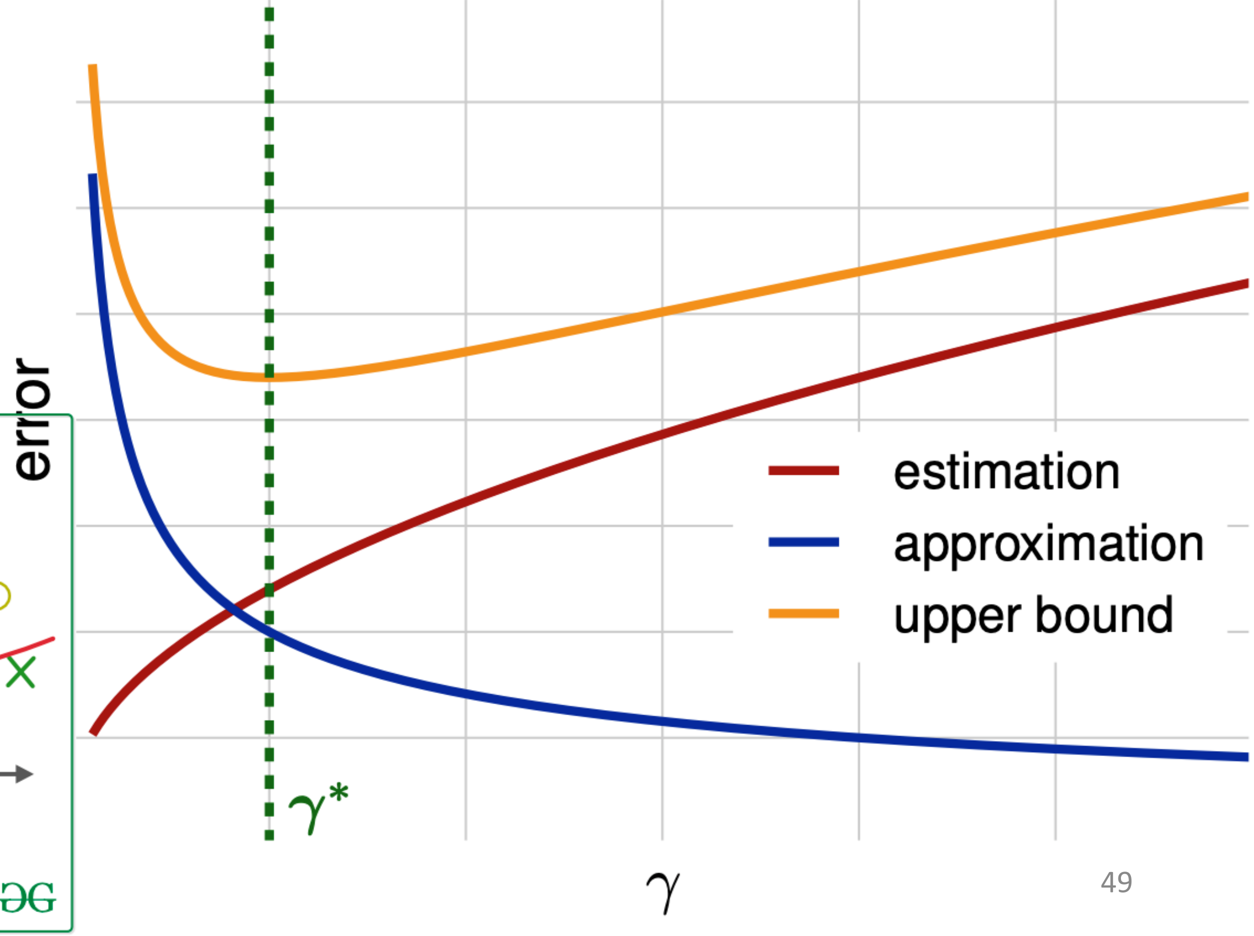
  - Approximation: only depends on H
  - Estimation
    - Recall $R(h) = \mathbb{E}_{(x,y)\sim D}[L(h(x), y)]$
    - Empirical error: $\hat{R}(h) = \frac{1}{N}\sum_{i=1}^{N} L(h(x_i), y_i)$
- Empirical risk minimization:

$$h = \text{argmin}_{h \in H} \hat{R}(h)$$

Model Selection

- $R(h) - \min_{h'} R(h') = \overbrace{\left(R(h) - \min_{h' \in H} R(h')\right)}^{\text{estimation}} + \overbrace{\left(\min_{h' \in H} R(h') - \min_{h'} R(h')\right)}^{\text{approximation}}$
- ERM $h = \text{argmin}_{h \in H} \hat{R}(h)$

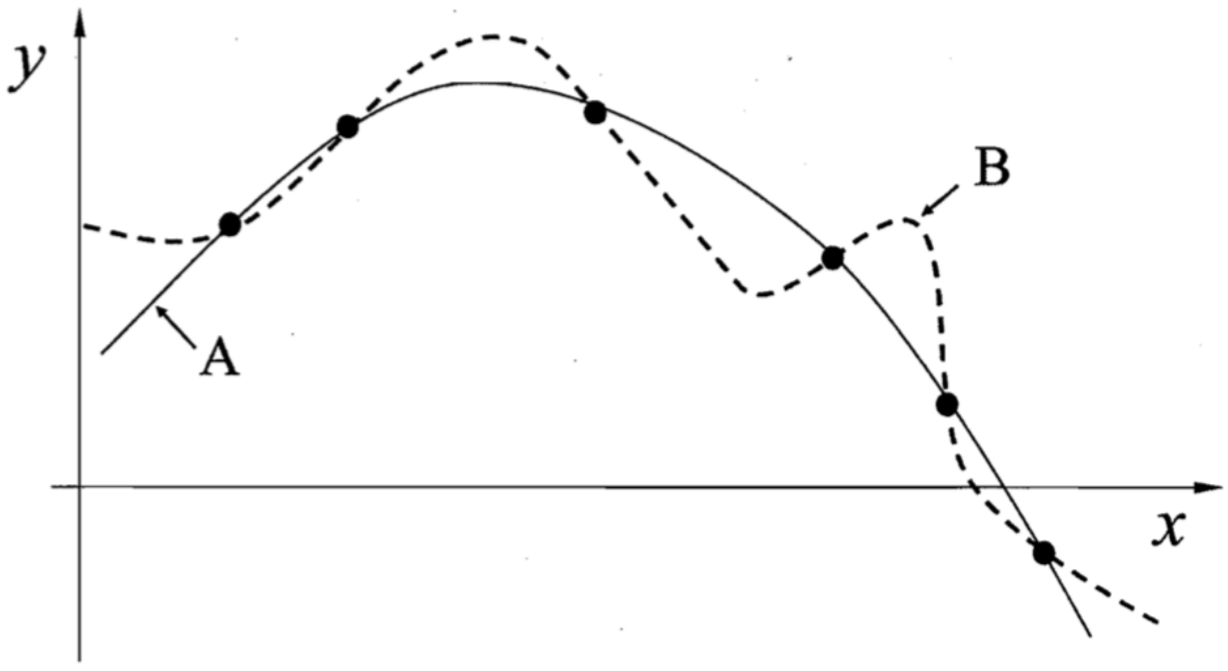




Principle of Occam's Razor

Suppose there exist two explanations for an occurrence.

The one that requires the least assumptions is usually correct.



存在多条曲线与有限样本训练集一致

Figure credit: Zhihua Zhou

Regularization

- Recall empirical risk minimization(ERM):

$$h = \text{argmin}_{h \in H} \hat{R}(h)$$

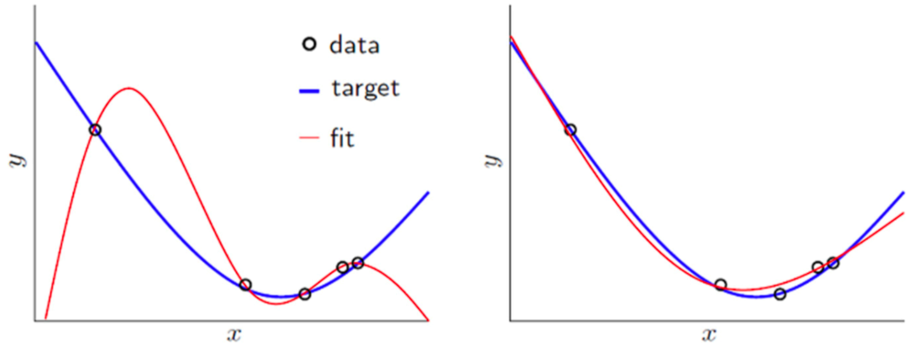
The above equation can be over-optimized

- Regularization-based algorithms

$$h = \text{argmin}_{h \in H} \hat{R}(h) + \lambda \Omega(h)$$

regularization parameter

Complexity of h



(a) without regularization (b) with regularization

Figure credit: Weinan Zhang

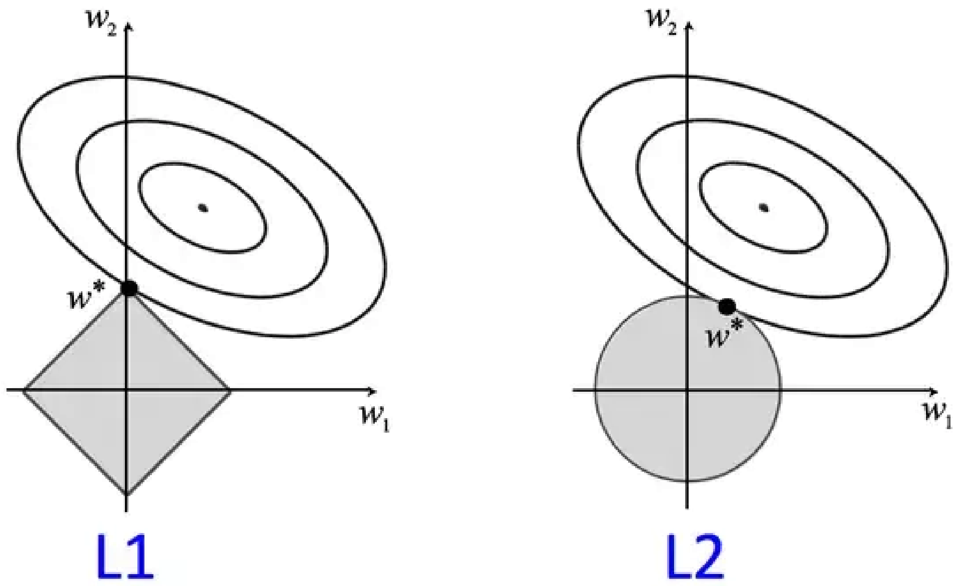
Regularization (cont.)

- E.g. L^2-norm (Ridge):

$$\Omega(h = ax + b) = a^2 + b^2$$

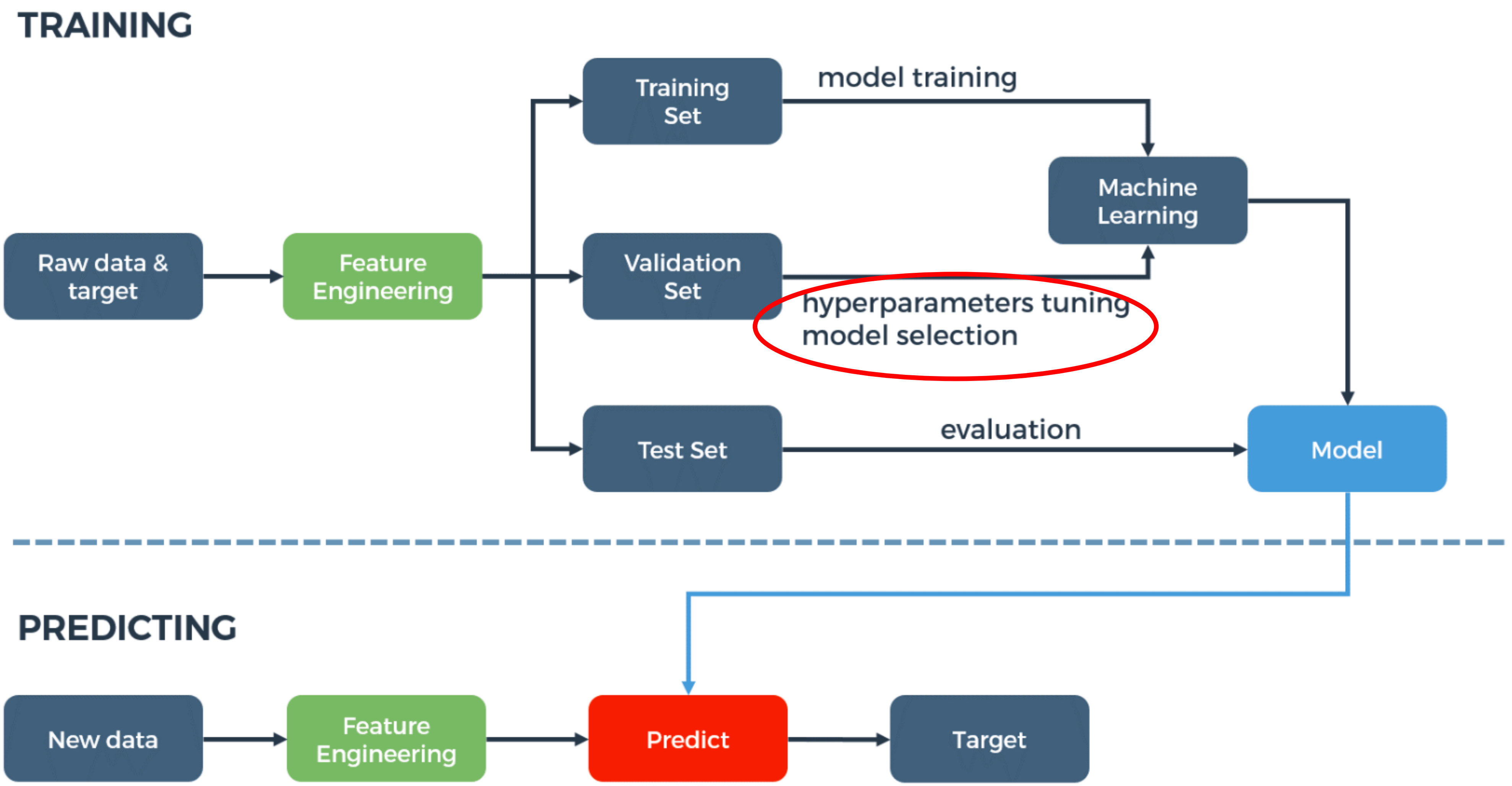
- E.g. L^1-norm (Lasso):

$$\Omega(h = ax + b) = |a| + |b|$$



https://miro.medium.com/max/1200/1*o6H_R3Do1zpch-3MZk_fjQ.png

Machine Learning Process



https://techblog.cdiscount.com/assets/images/DataScience/automl/ML_process.png

Summary

Shuai Li

https://shuaili8.github.io

- The classification of machine learning
  - Supervised/unsupervised/reinforcement
- Supervised learning
  - Evaluation metrics for classification
    - Accuracy/Precision/Recall/F1 score/AUC/AUPR
  - Model selection: bias/variance/generalization
  - Machine learning process

Questions?