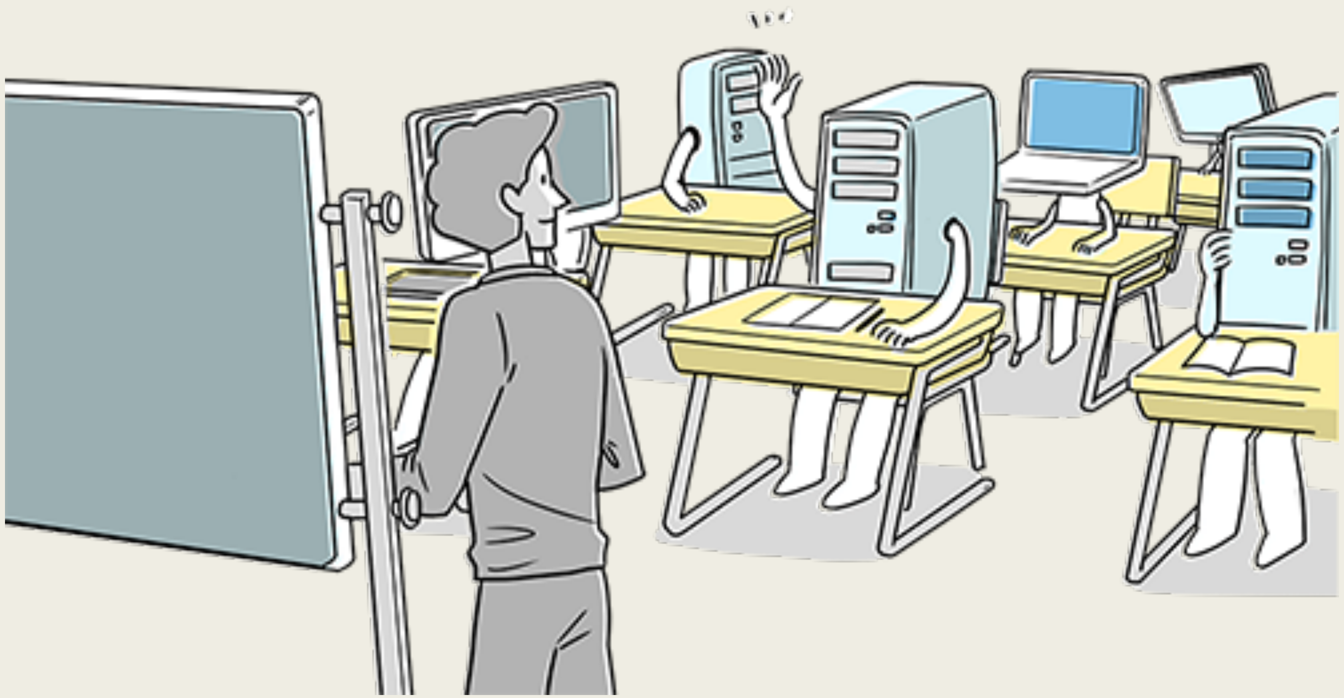
Sorawit Saengkyongam
sorawitj@gmail.com



INTRODUCTION TO MACHINE LEARNING

WHAT IS MACHINE LEARNING?

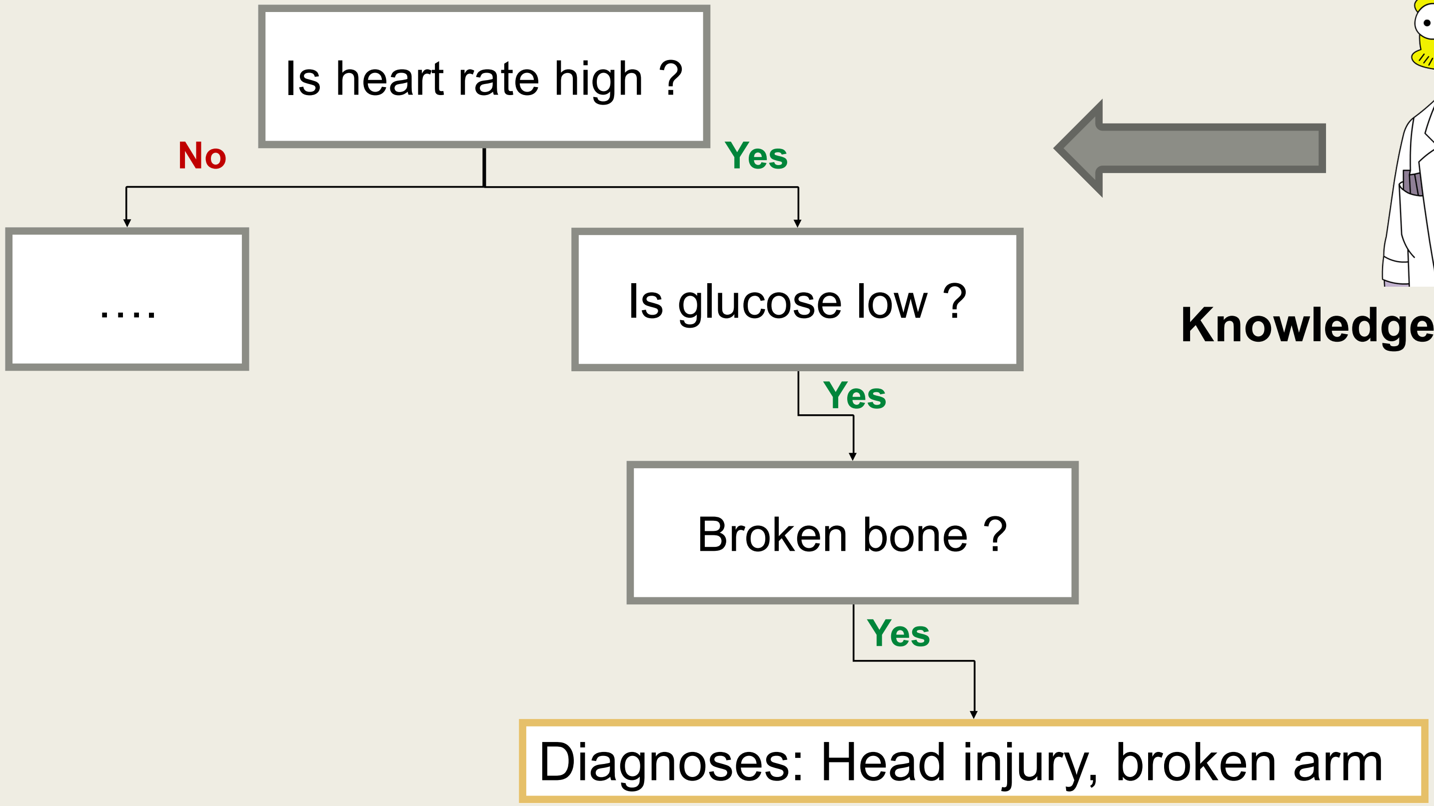
Machine Learning

- A subfield of Artificial Intelligence (AI)

- The field of study that gives computers the ability to learn without being explicitly programmed — Arthur Samuel

What is “NOT” machine learning

AI system for medical diagnosis



Knowledge from an expert

What is machine learning

Can we learn those rules from the data ?

Heart rate	Glucose	Broken bone	….	Diagnoses
High	Low	Yes		Head Injury, Broken arm
Normal	High	No		Diabetes
….	….	….		….

Supervised machine learning: Features **X**

Labels **Y**

Model Output

What is machine learning

- How would you teach someone to ___ ?

- Classify if an email is spam
- Estimate how many likes your post will get
- Recommend a movie to a friend
- Group customers into segments

What is machine learning

- Let’s listen from the world’s expert
- http://ta.virot.me/fb-ai-explainer/

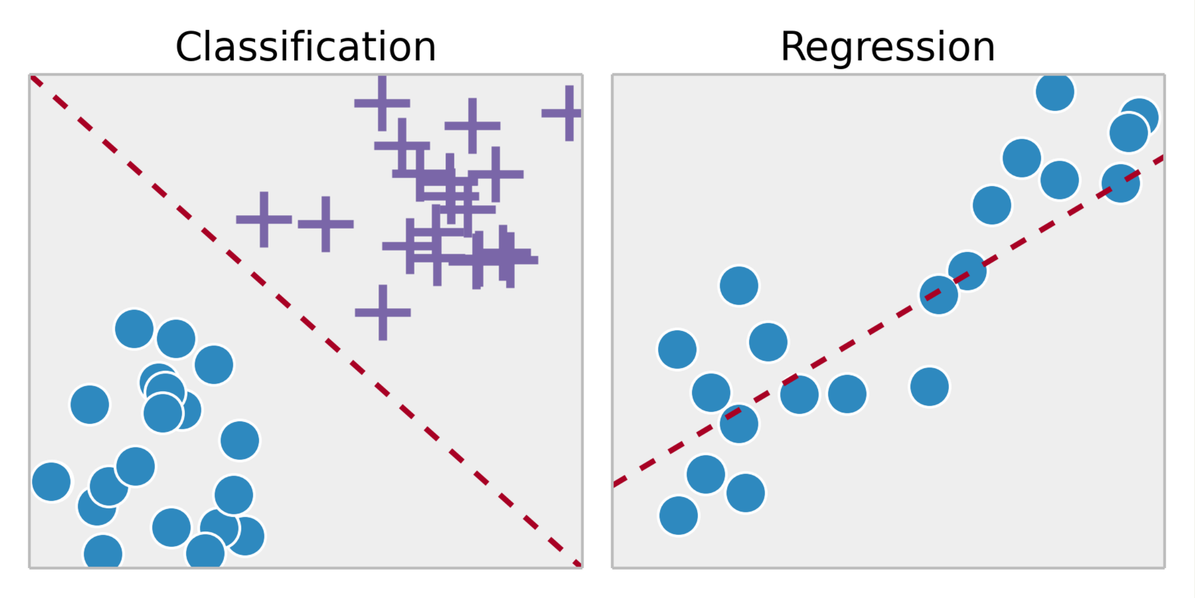
WHAT ARE MACHINE LEARNING USE CASES?

Agoda’s use cases

- Pay-Per-Click (PPC) Bidding Algorithm
- Search and Recommender Systems
- Dynamic Pricing
- Fraud Detection
- Photos Auto-tagging

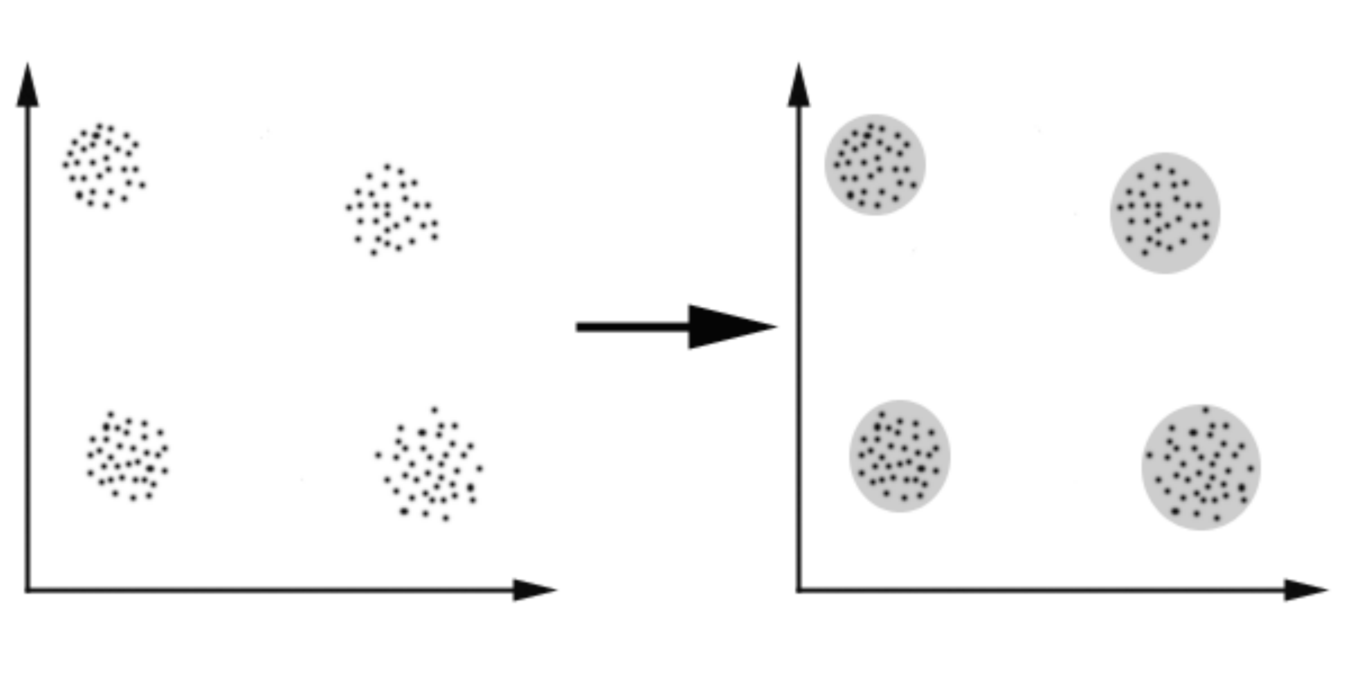
Type of machine learning

- **Supervised Learning**
  - The target values (**labels**) are provided
  - The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.
  - **Classification** (for discrete outcome) and **Regression** (for continuous outcome)
  - Example: Image classification



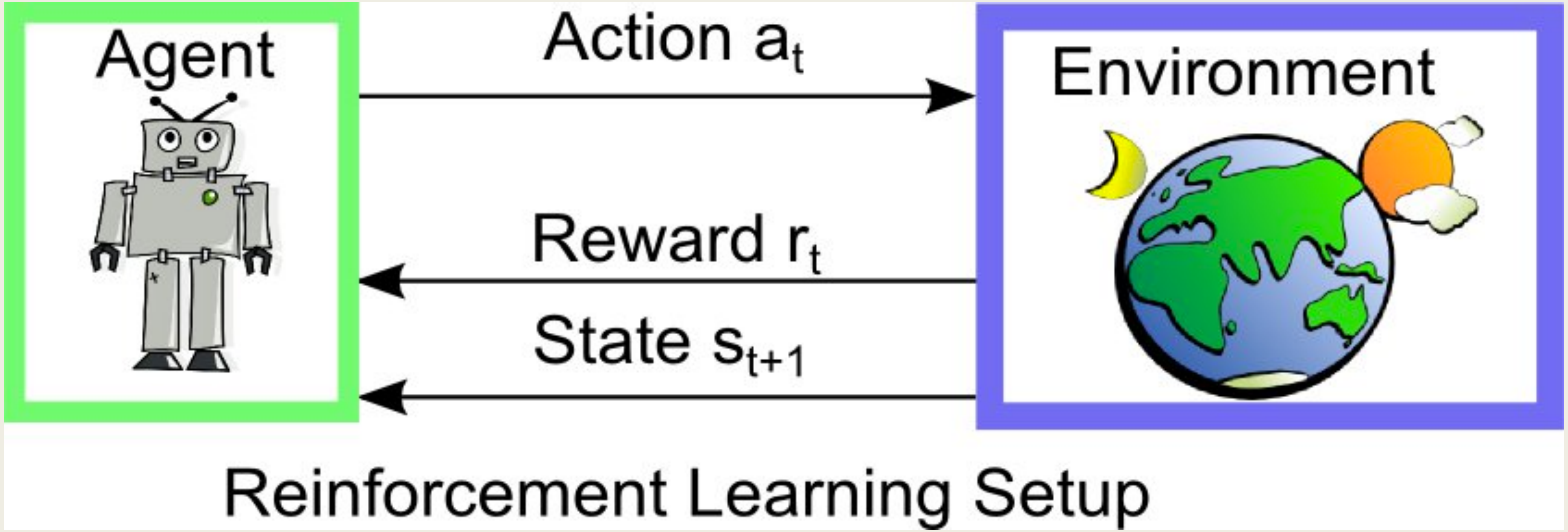
Type of machine learning

- **Unsupervised Learning**
  - No target values specified
  - The goal is to model the underlying structure or distribution in the data in order to learn more about the data
  - Example: Customer Segmentation

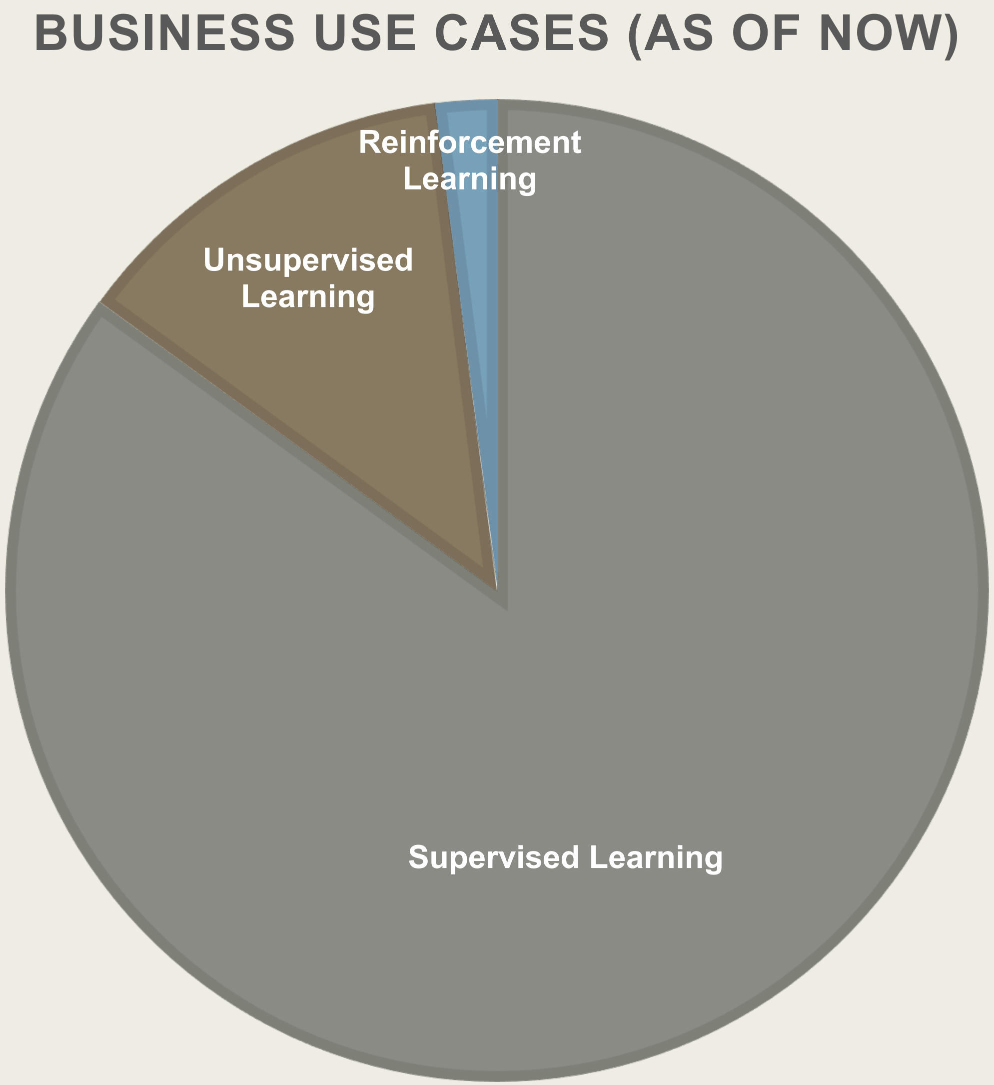


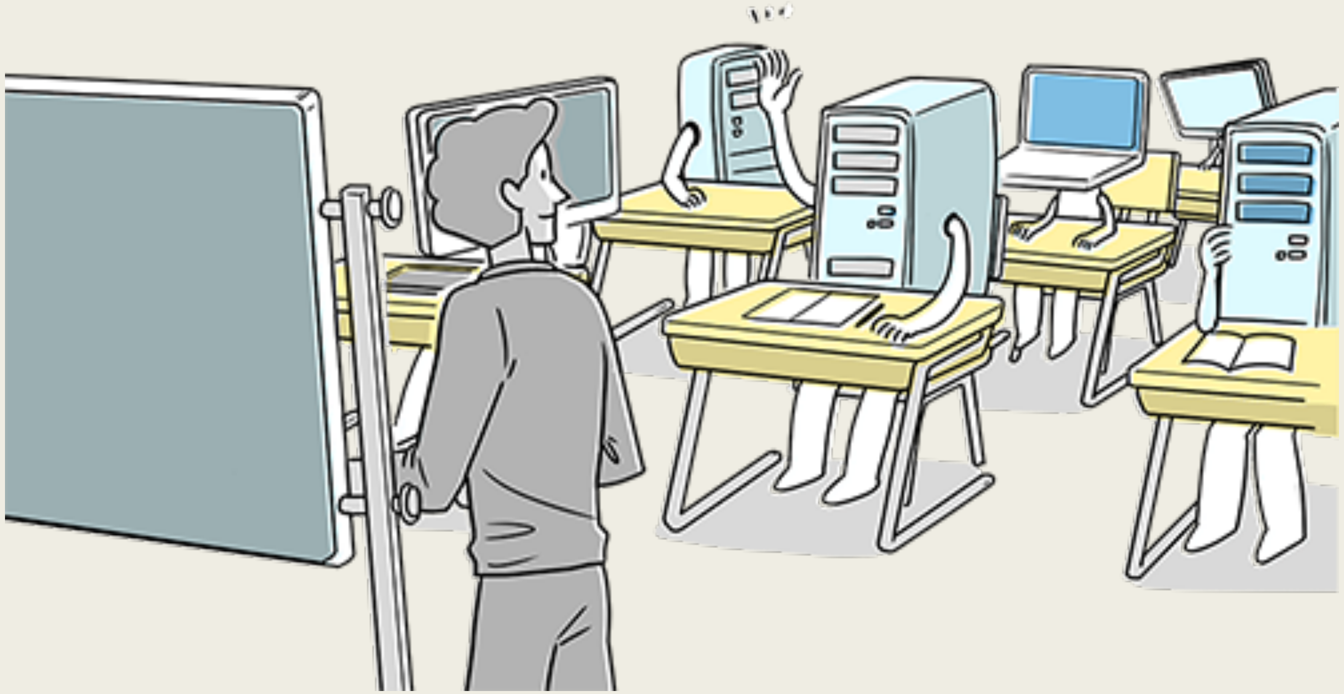
Type of machine learning

- **Reinforcement Learning**
  - The method aims at using observations gathered from the interaction with the environment to take actions that would maximize the reward
  - Example: AlphaGo



Type of machine learning



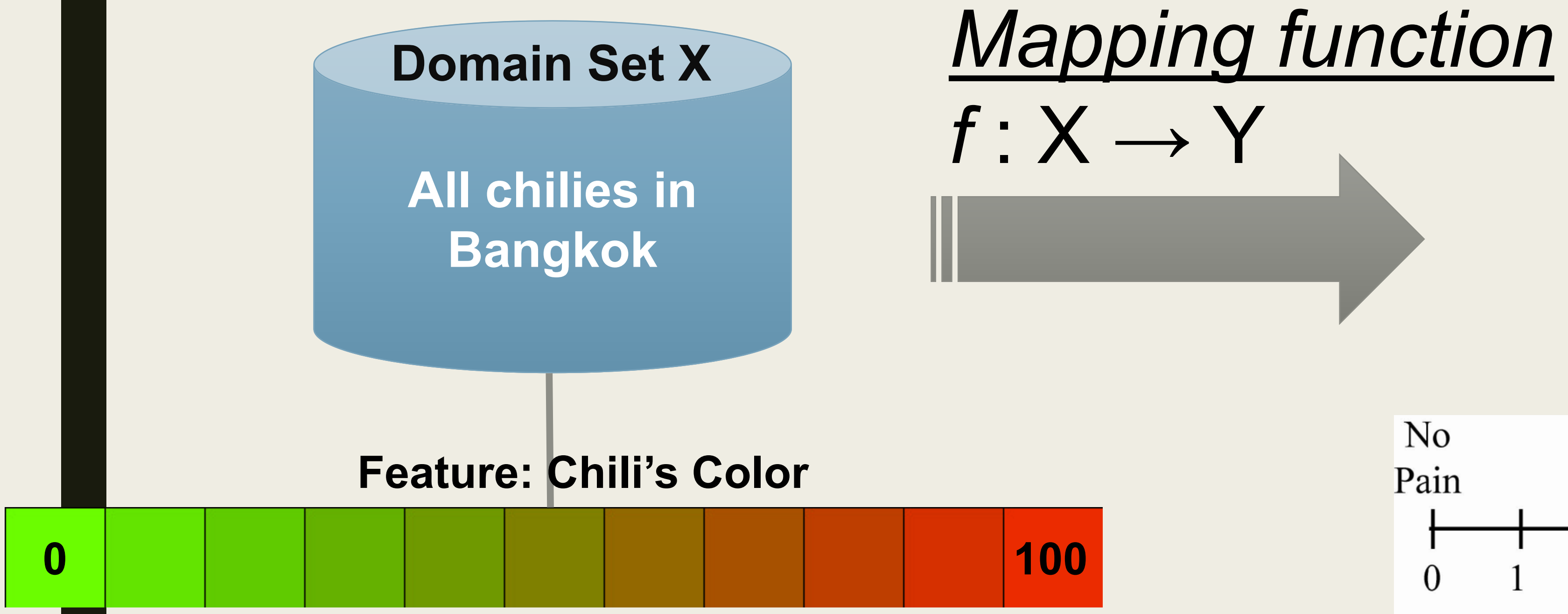


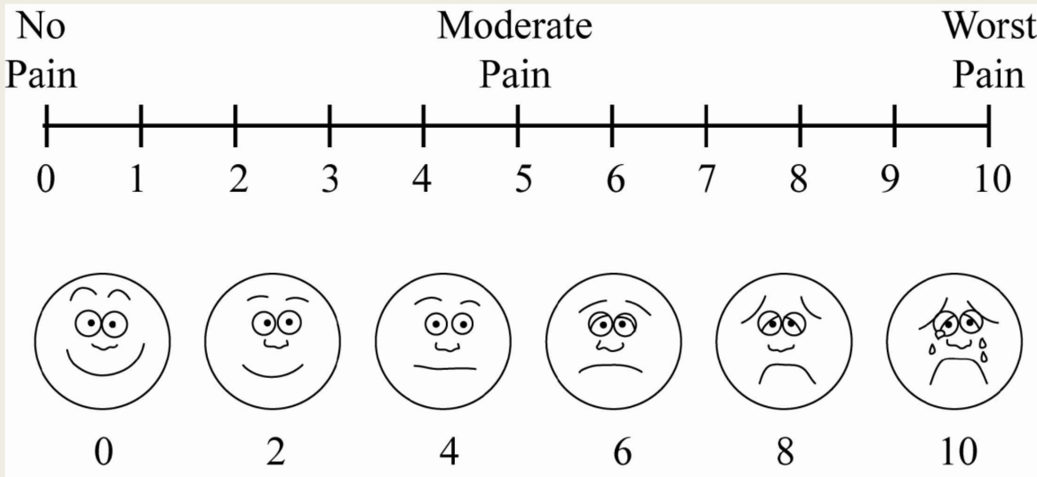
INTRODUCTION TO SUPERVISED LEARNING

Let’s start with example

LOVE

Learning Framework



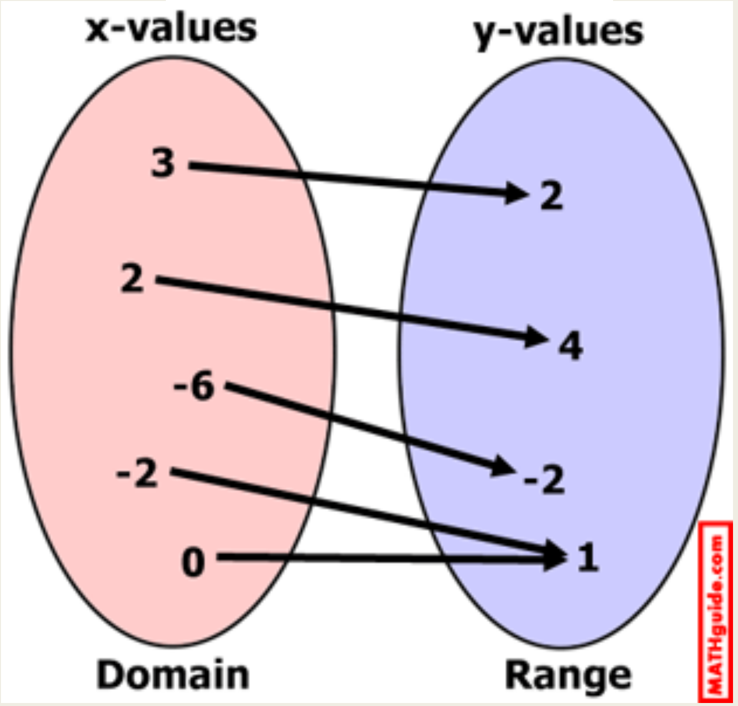


Learning Framework

SUPERVISED LEARNING IS ALL ABOUT FINDING THE **MAPPING FUNCTION**

What people think I do

What I actually do



Learning Framework

Learning Algorithm

Predictor

p: X → Y best represent

Mapping function *f*

<u>Measure of success</u>: Generalized Error

$$E_{(x,\,y)\sim D}\,(p(x)-y)^2$$

Domain Set X

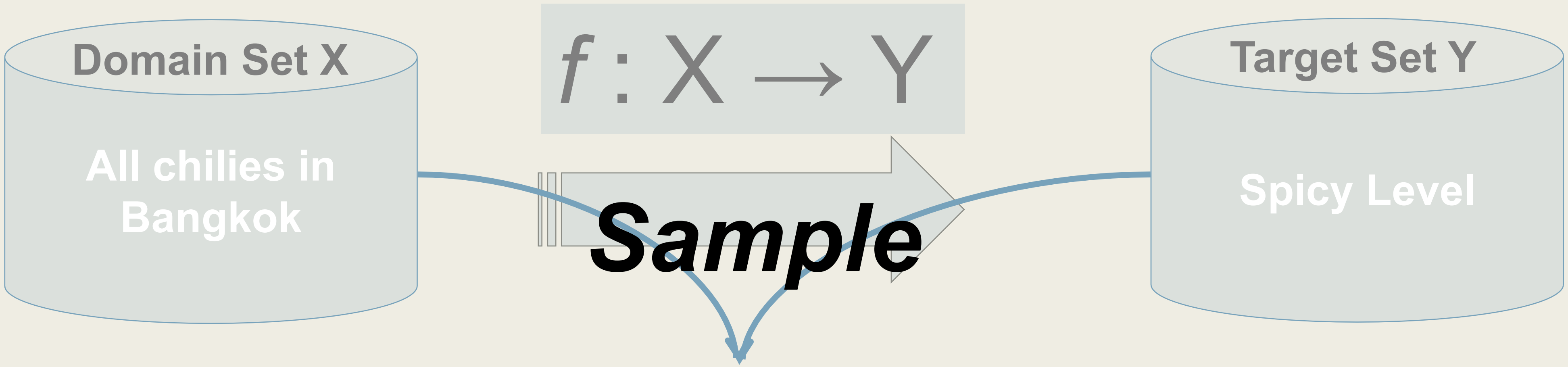
All chilies in Bangkok

Apply Predictor p

Observe actual y

$$(p(x)-y)^2$$

Learning Framework



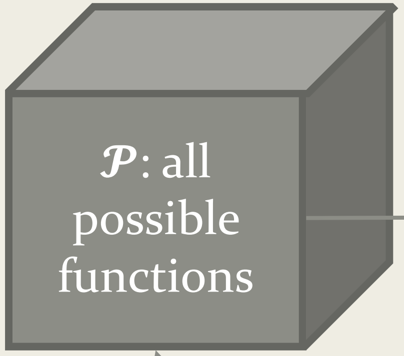
Training Data	
Color (X)	**Spicy Level (Y)**
2	2
5	2
12	3
37	5
48	6
67	6
88	7
95	9

Empirical Risk Minimization

Define: $\boldsymbol{\mathcal{P}}$ $as\ all\ possible\ functions\ mapping\ from\ X \rightarrow Y$

Find best p in $\boldsymbol{\mathcal{P}}$ that has the lowest Error on $\boldsymbol{Training\ Data}$

$$Min_{p\,\epsilon\,\mathcal{P}} \sum_{i=1}^{n} (p(x) - y)^2$$



$Pick\ p$

Training Data	
Color (X)	**Spicy Level (Y)**
2	2
5	2
12	3
37	5
48	6
67	6
88	7
95	9

$Calculate$

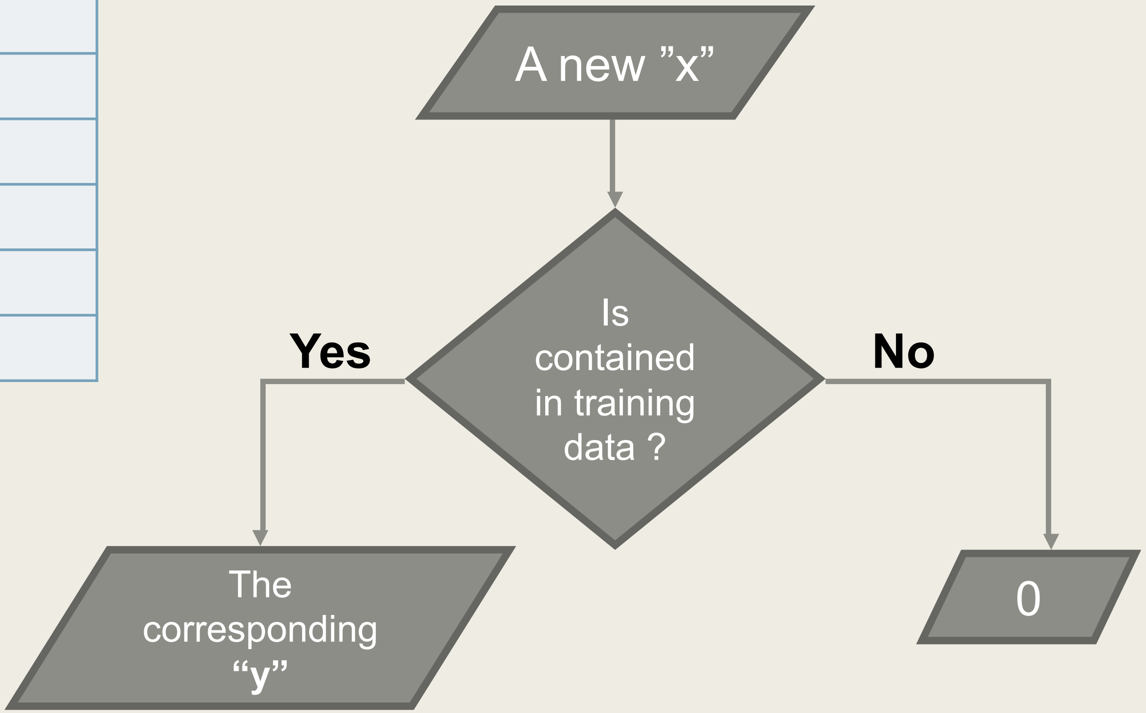
$\underline{Training\ Error}$

$$\sum_{i=1}^{n} (p(x) - y)^2$$

Empirical Risk Minimization

Training Data	
Color (X)	**Spicy Level (Y)**
2	2
5	2
12	3
37	5
48	6
67	6
88	7
95	9

$$p(x) = \begin{cases} y_i, & if\ x = \ x_i\ for\ some\ i \\ 0, & otherwise \end{cases}$$



Empirical Risk Minimization

Training Error

$$\sum_{i=1}^{n}(p(x)-y)^2 \; = 0$$

Generalized Error

HUGE!!

OVER FITTING

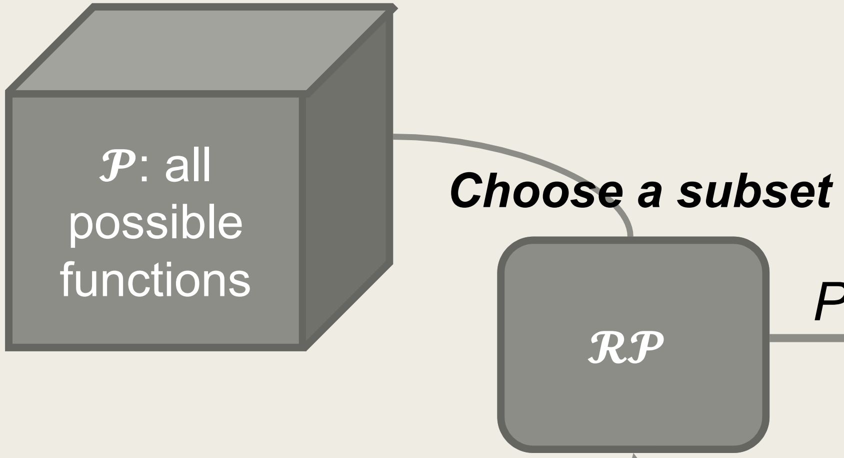
Bias (Assumption)

Apply the ERM learning rule over a **restricted** set

Define: $\boldsymbol{\mathcal{RP}}$ *as a chosen subset of* $\boldsymbol{\mathcal{P}}$

Find best p in $\boldsymbol{\mathcal{RP}}$ that has the lowest Error on $Training\ Data$

$$Min_{p\,\epsilon\,\mathcal{RP}} \sum_{i=1}^{n} (p(x) - y)^2$$



Pick p

Training Data	
Color (X)	**Spicy Level (Y)**
2	2
5	2
12	3
37	5
48	6
67	6
88	7
95	9

$Calculate$

Training Error

$$\sum_{i=1}^{n} (p(x) - y)^2$$

Linear Regression

Pick $\mathcal{RP}$ as: $p(x) = \boldsymbol{\theta_0} + \boldsymbol{\theta_1} x$, so we only search for best $\boldsymbol{\theta_0}$ and $\boldsymbol{\theta_1}$value

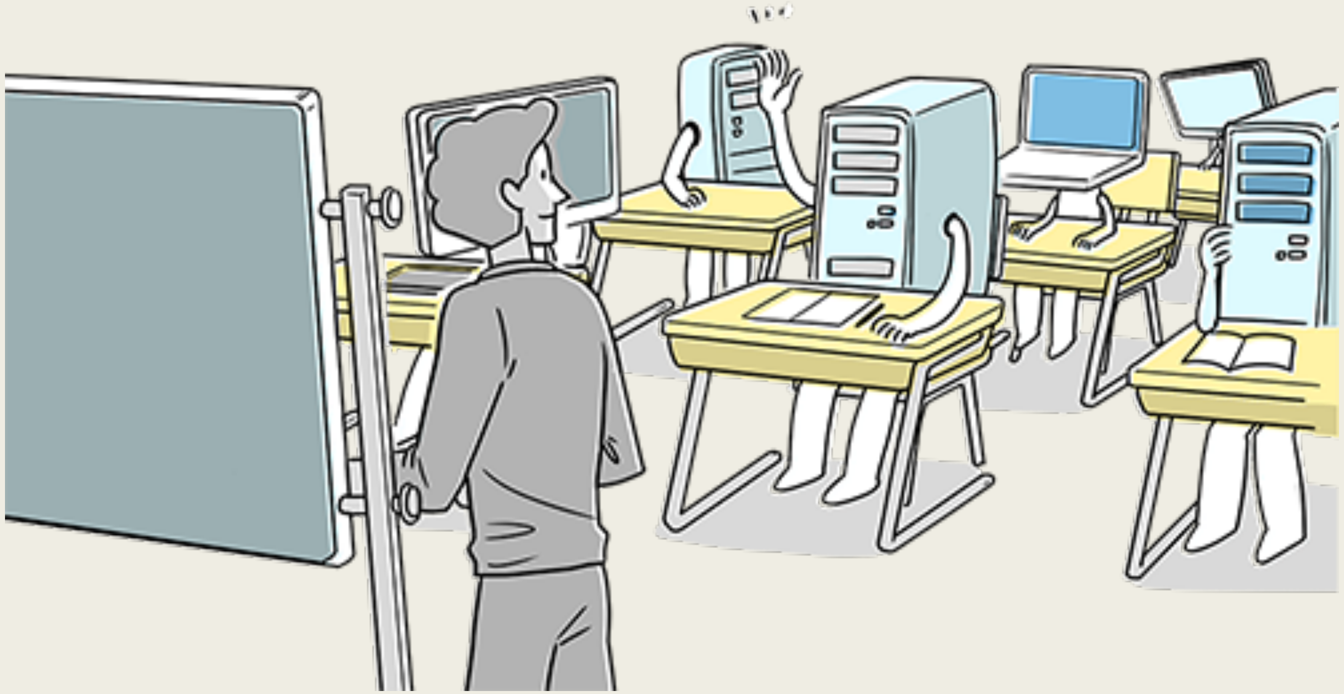
We call $\boldsymbol{\theta_0}, \boldsymbol{\theta_1}$ "$\boldsymbol{Model\ Parameters}$"

Training Error

$$\sum_{i=1}^{n} (p(x) - y)^2 = Small$$

Generalized Error

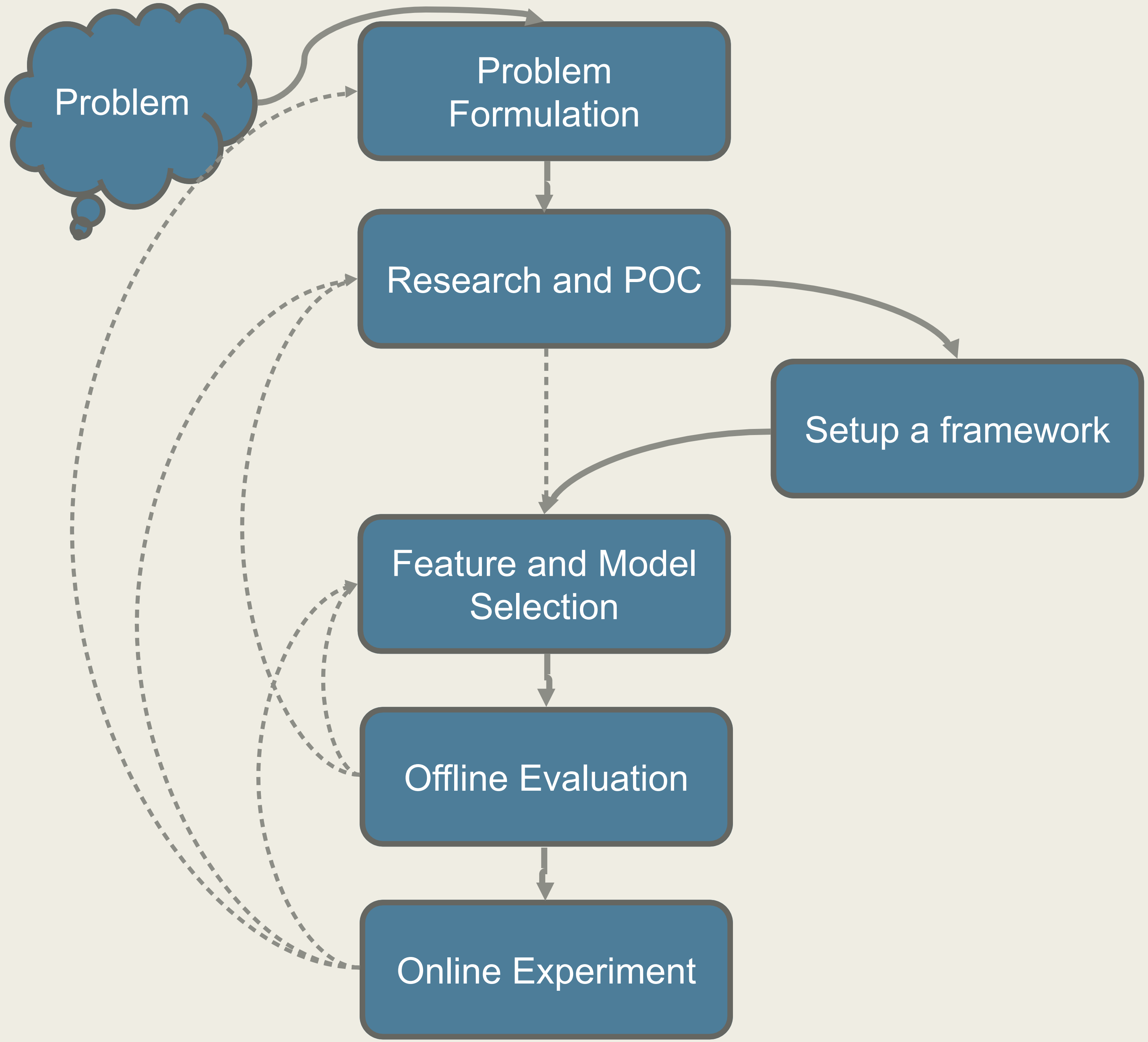
Small



MACHINE LEARNING PROCESS

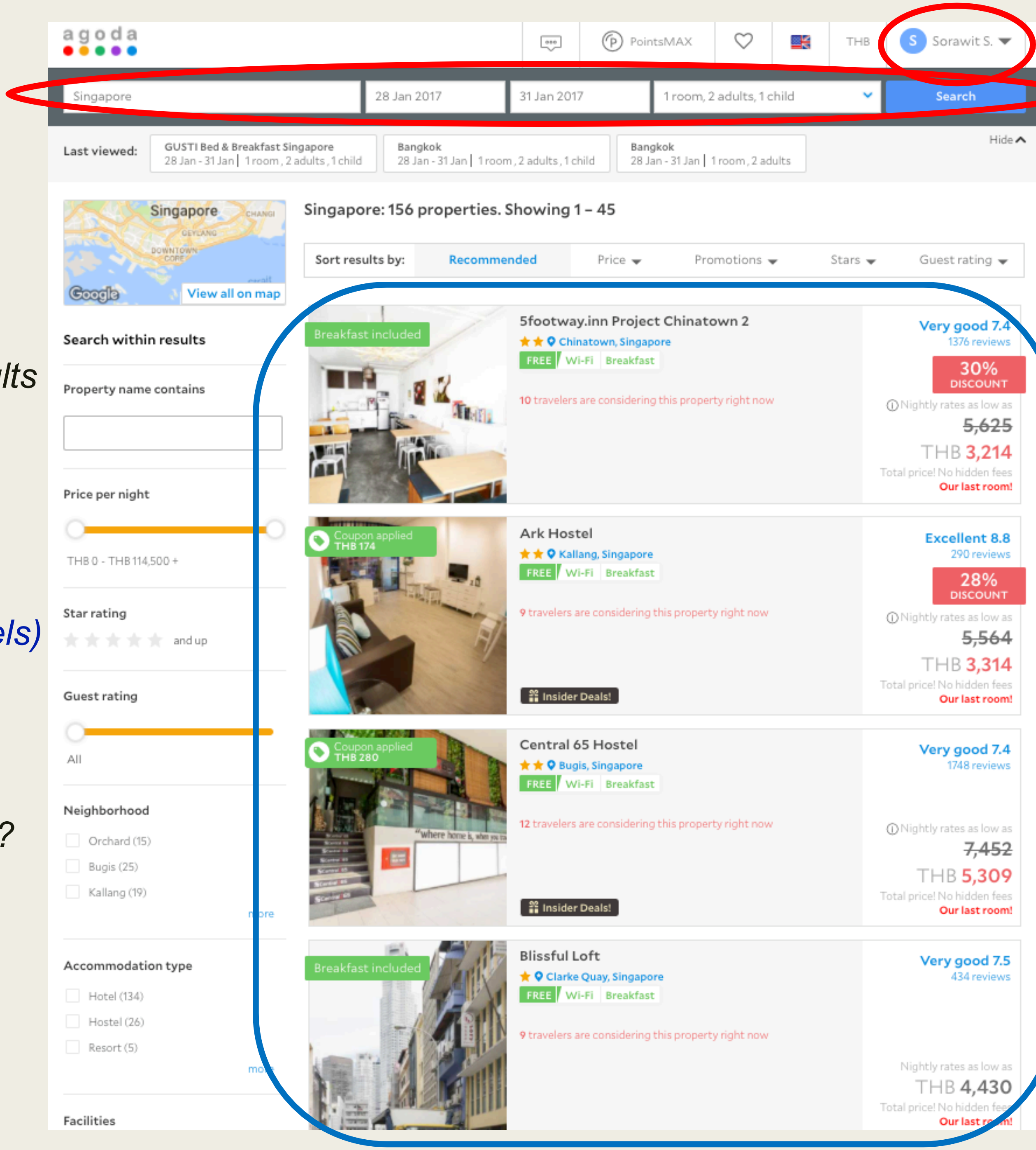
Onboarding Machine Learning

- How do we start adopting machine learning in practice ?
- What is a process for developing ML products ?


Problem
Problem Formulation
Research and POC
Setup a framework
Feature and Model Selection
Offline Evaluation
Online Experiment

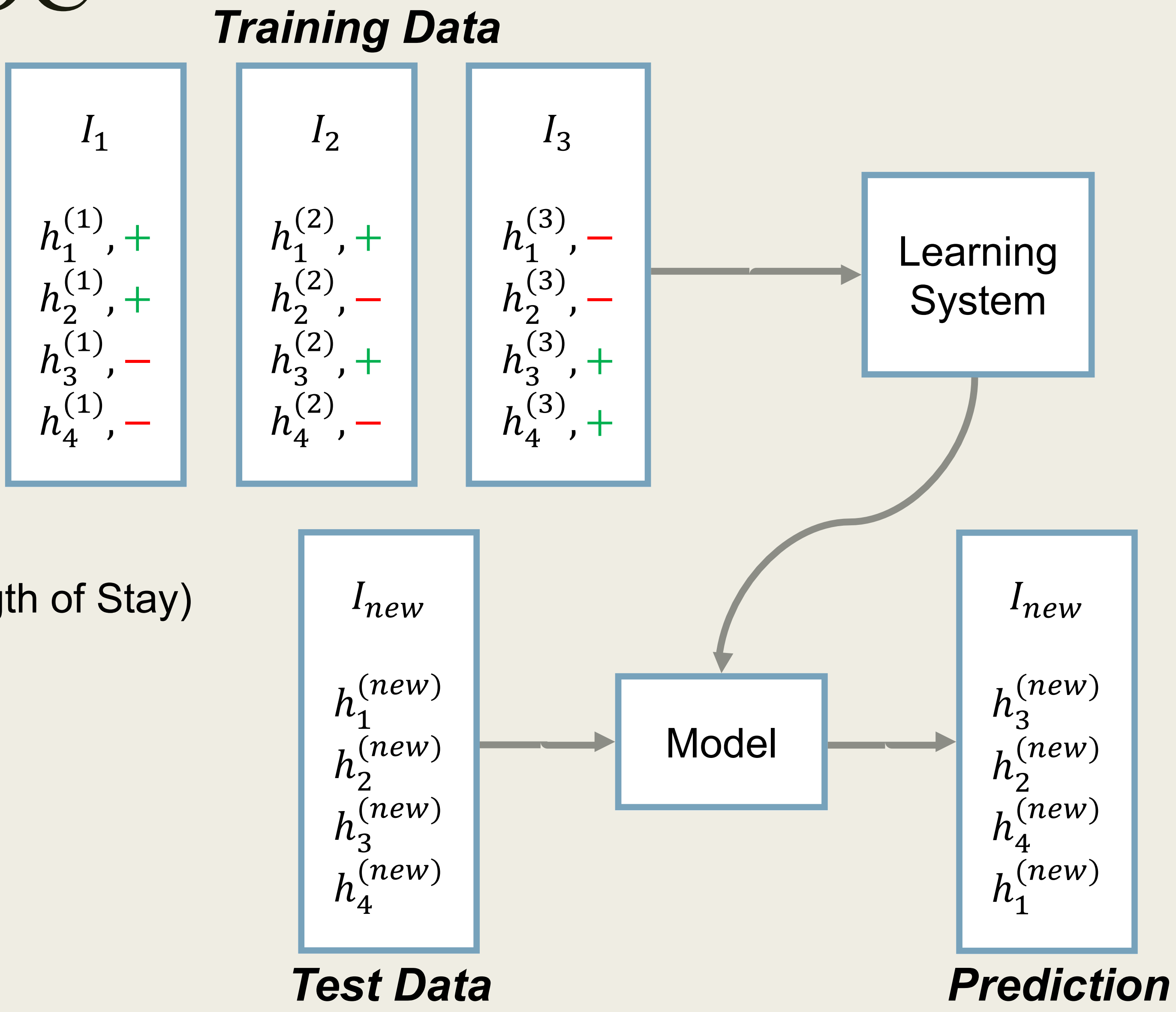
Problem Formulation

- **Objective**
  - *Maximizing reward by providing relevant search results*
- **Input Output**
  - ***Input**: (User, Length of Stay)*
  - ***Output**: {H1, H2, H3, H4 … } (a **permutation** of hotels)*
- **Measure of success**
  - *Uplift in Conversion rate, Click through rate or Profit ?*
- How would we solve this problem ?



Research and POC

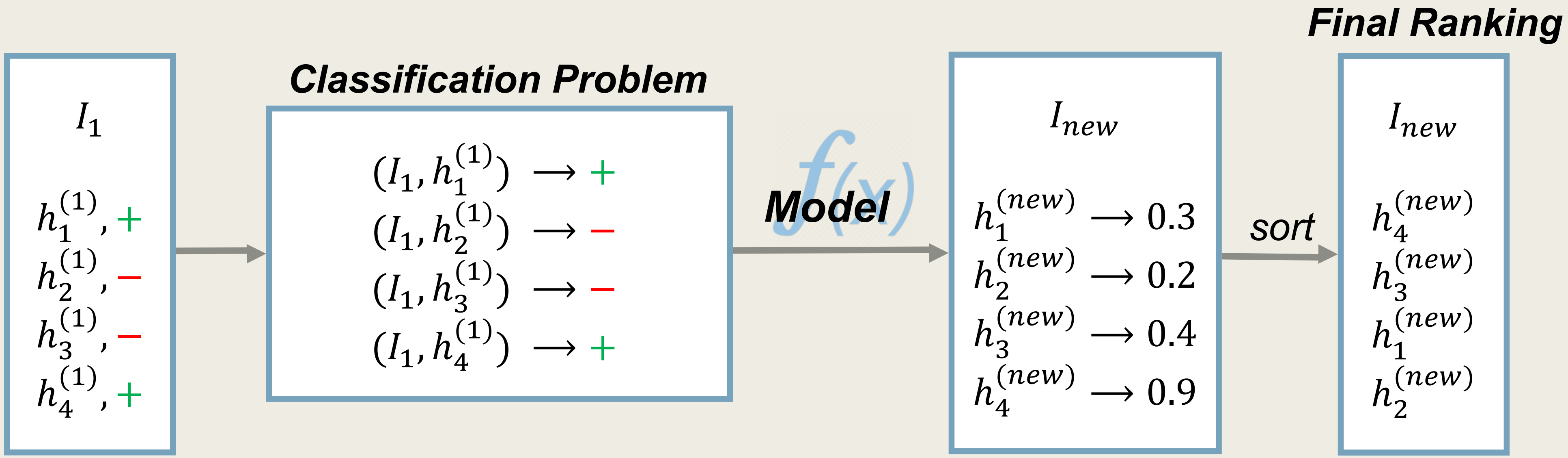
- **Avoid** reinventing the wheel
- **Information Retrieval:**
  - *Learning to rank*



Notation
I → (User, Length of Stay)
h → Hotel

Research and POC

- **Point-wise Approach**
  - Assign a score for each Input-Hotel pair



Research and POC

- **Evaluation metrics: each problem has its own appropriate set of metrics**

<u>Metrics</u>

- Precision at K: Compute % relevant in top K, ignores hotels ranked lower than K

Ex.

Prec@3 = 1/3

Prec@4 = 2/4

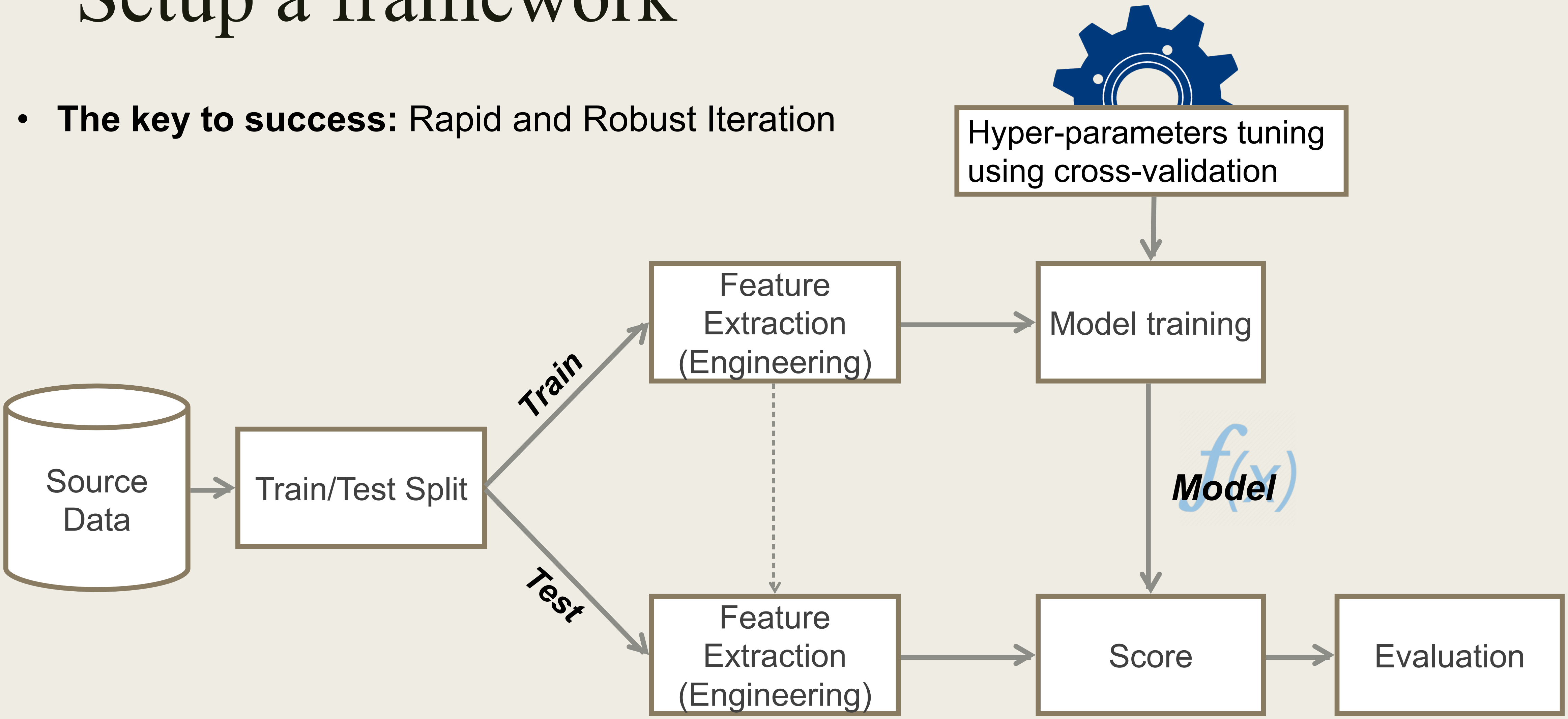
Prec@5 = 3/5

I_1

$h_1^{(1)}, +$
$h_2^{(1)}, -$
$h_3^{(1)}, -$
$h_4^{(1)}, +$
$h_5^{(1)}, +$

Setup a framework

- **The key to success:** Rapid and Robust Iteration



Feature Selection

Feature Selection/Engineering

- Domain knowledge plays an important role

– ***Hotel features***

- hotel_id, star rating
- price
- review score

– ***User features***

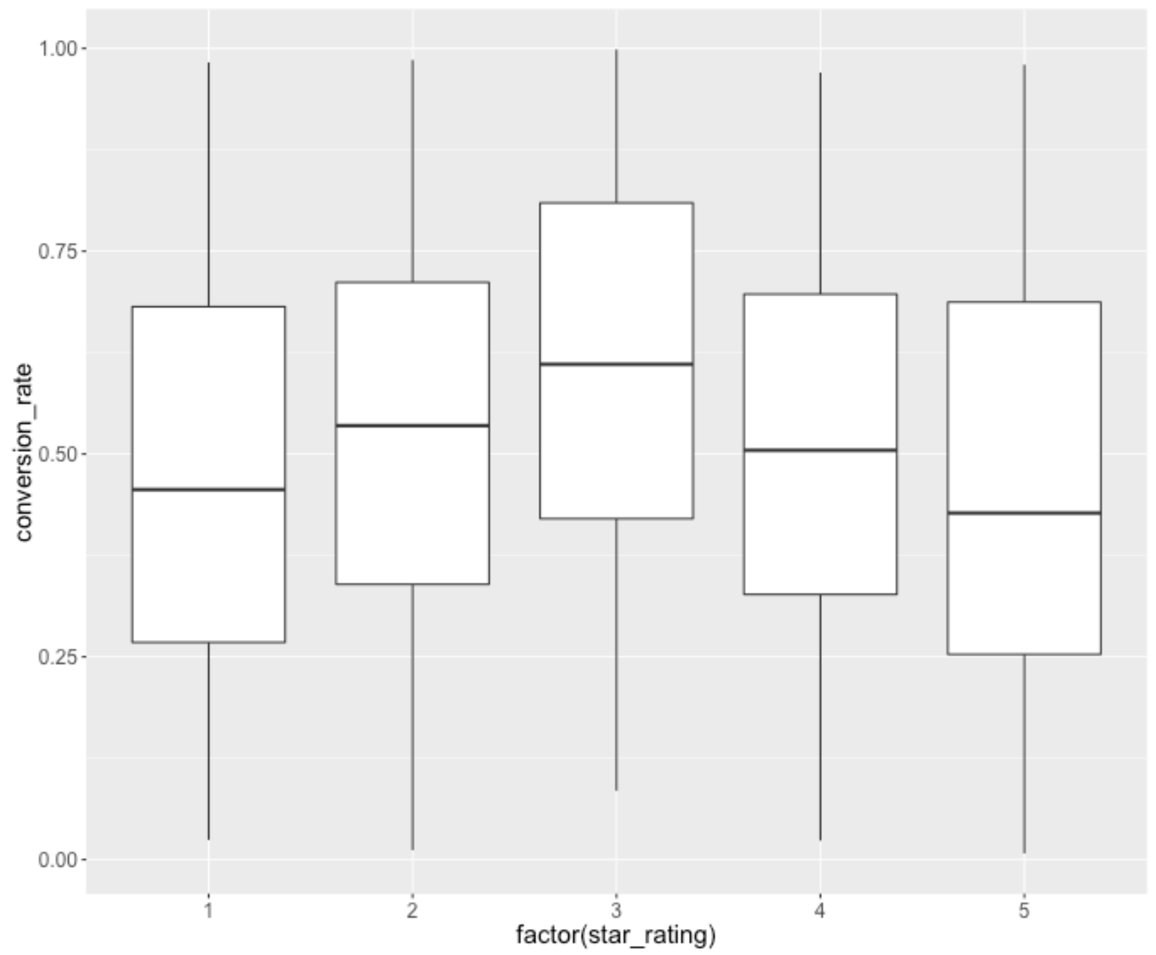
- user_id, length of stay
- user origin
- historical booking rate of user

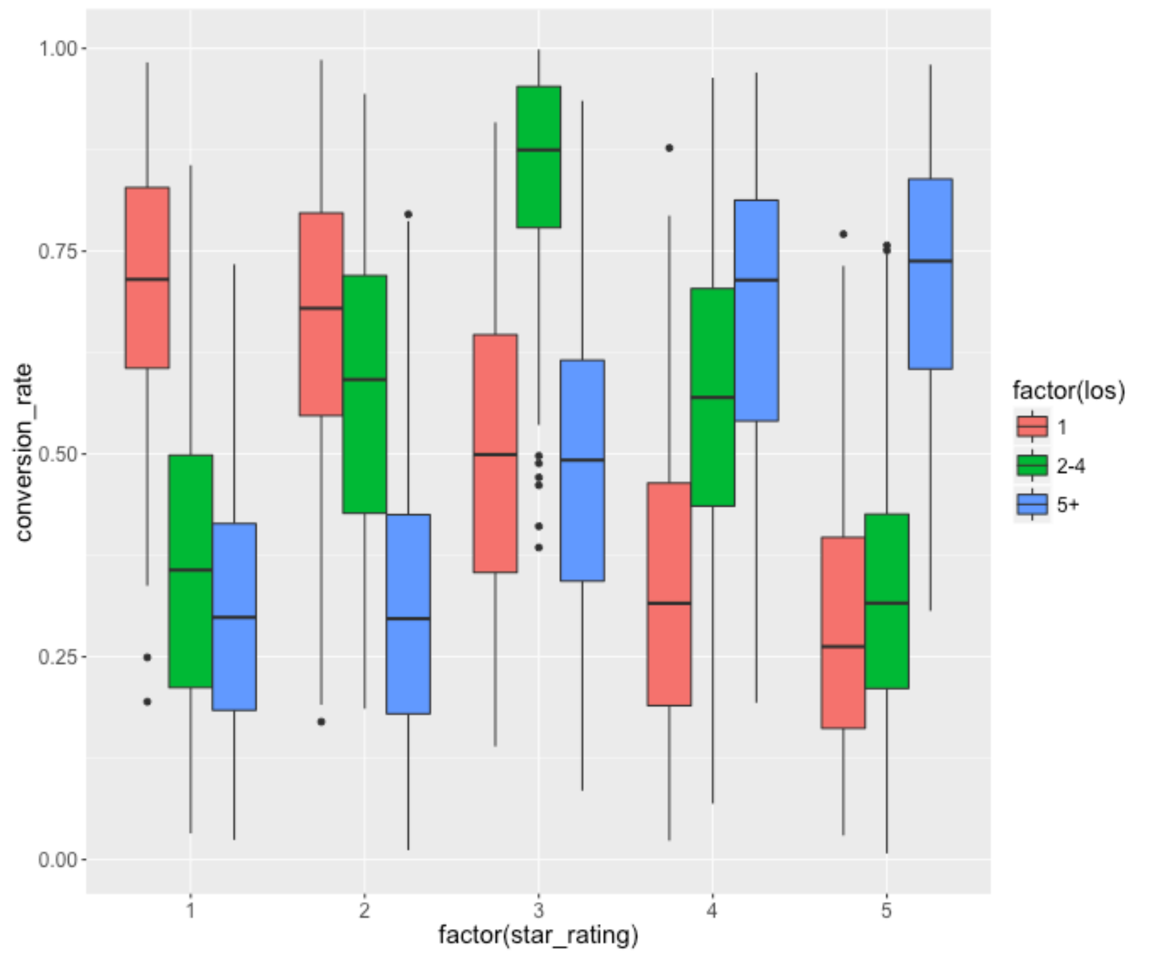
– ***User-Hotel features (the most important)***

- abs(historical bookings price of user – hotel price)
- length of stay * star rating
- hotel_id * user_id

Feature Selection

- Feature exploration: ***identify potential features***



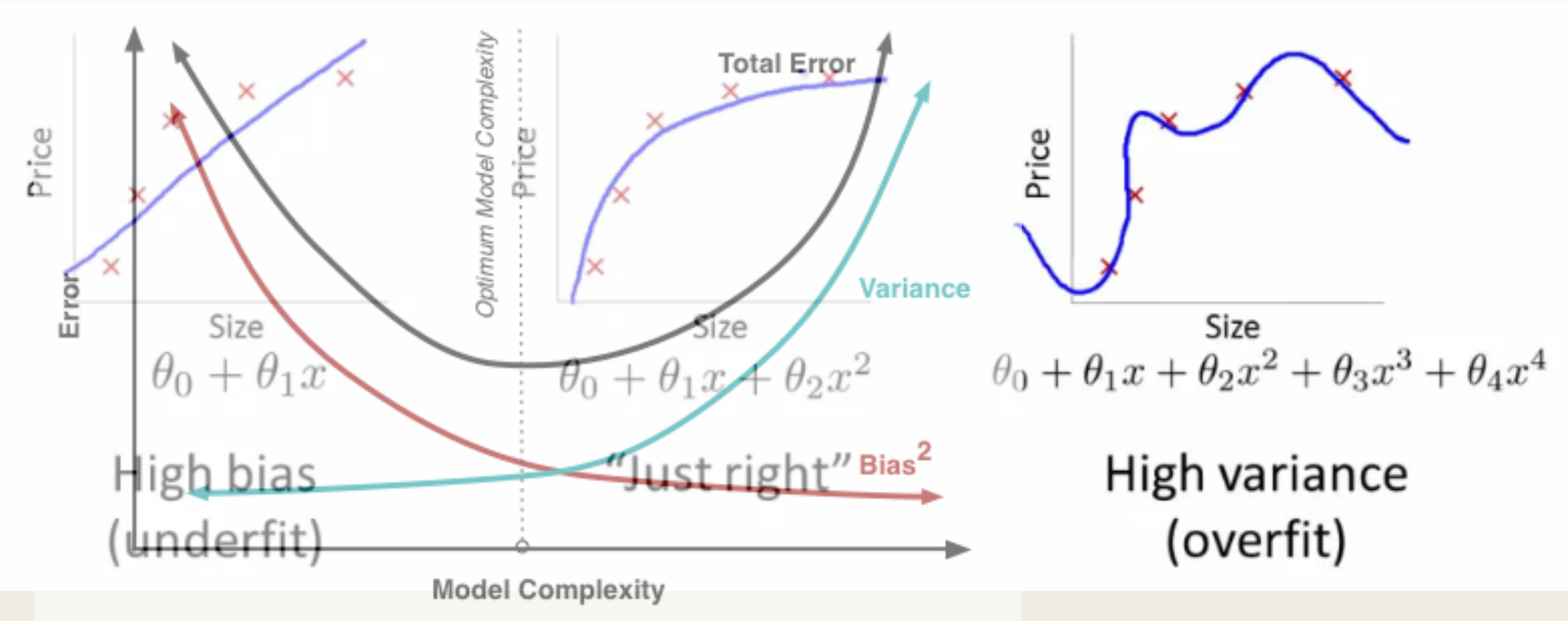


Model Selection

- Pick your candidate algorithms
- **Always always start with Baseline**: Rank by popularity
- Generalized Linear Model
- SVM
- Gradient Tree Boosting
- Factorization Machine
- Neural Network

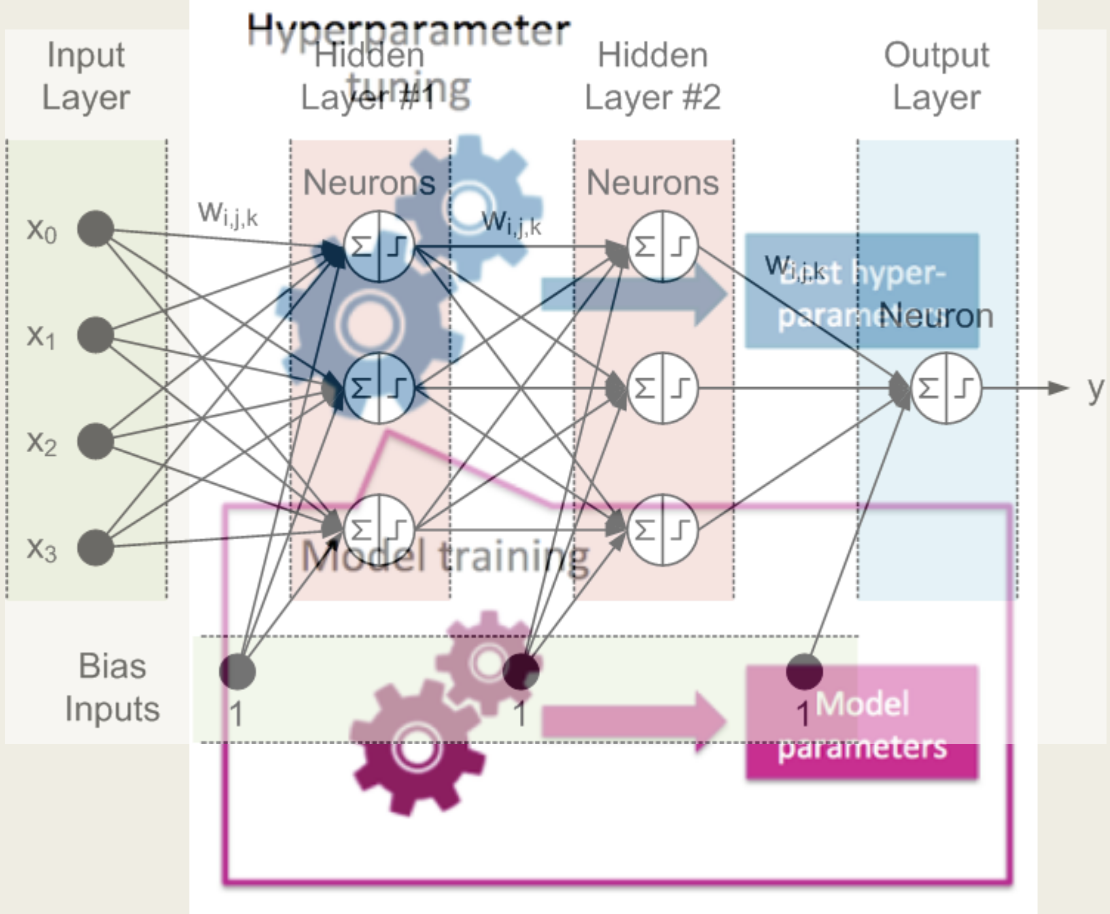
Model Selection

- **Model selection**: understand **bias-variance** trade off



Model Selection

- **Super important:** Understand hyper-parameters of your picked algorithms



Model Parameters

- Parameters that we optimize during training process

Hyper-parameters

- Parameters that express “higher-level” properties of the model such as its complexity

*It actually boils down to the **bias-variance** trade-off*

Tuning hyper-parameters

- Grid search with cross-validation
- Bayesian Optimization

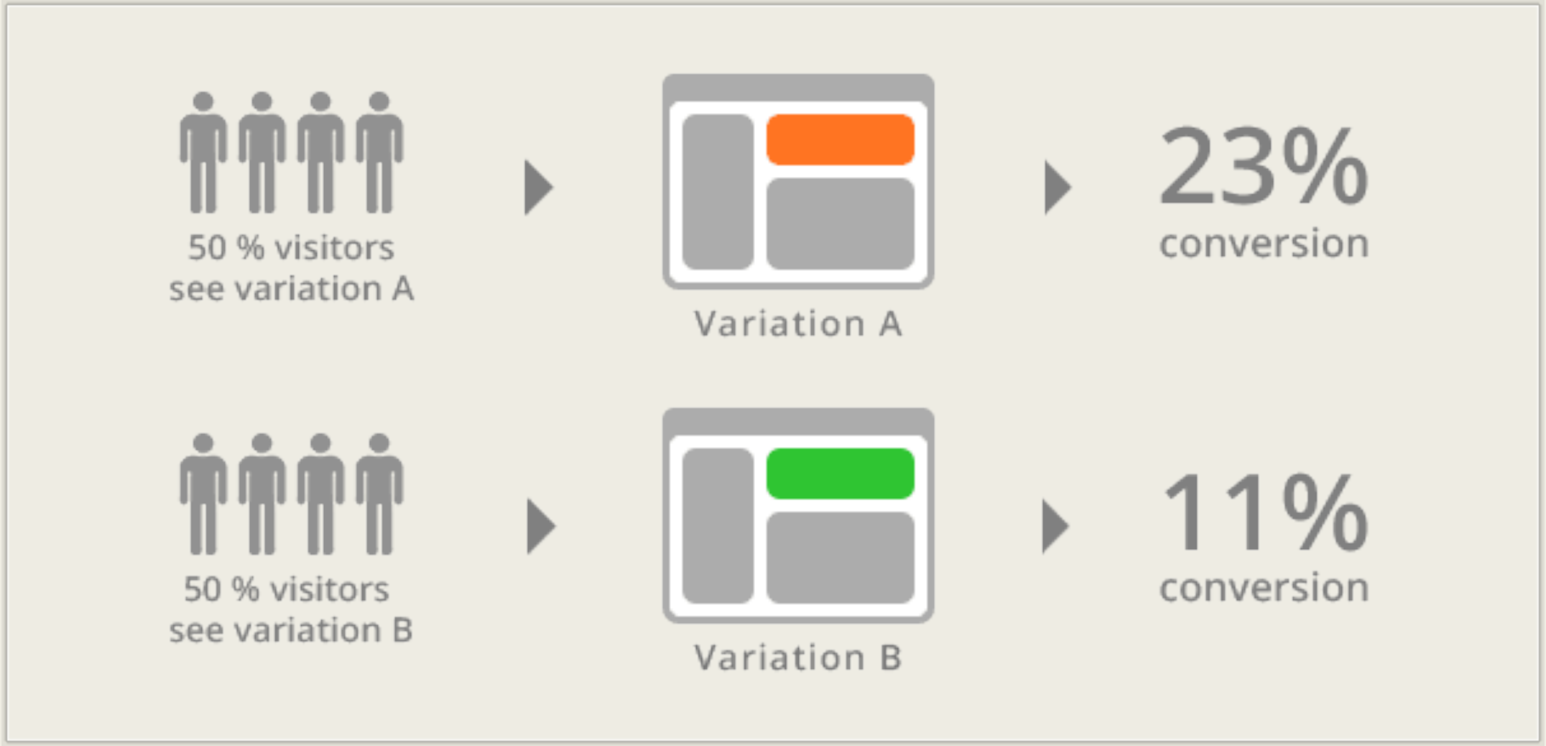
Offline Evaluation (Back testing)

- Remember that what only matters is **“Generalized Error**”
- – Train/Test splitting logic is very crucial
- – Make sure there is no information leakage

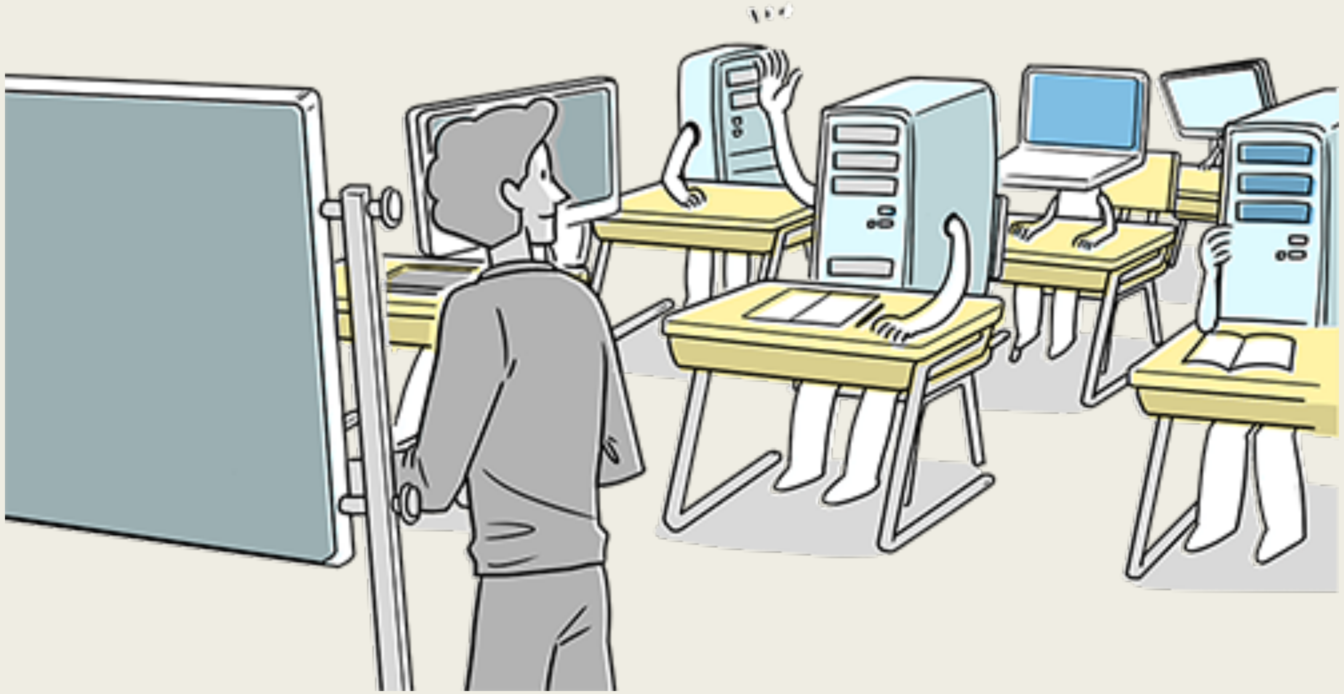
- Defining the right **metrics**
- – Indication to make decision on follow-up online experiments
- – A critical issue is how offline metrics correlate with online test results

Online Experiment

- A/B Testing



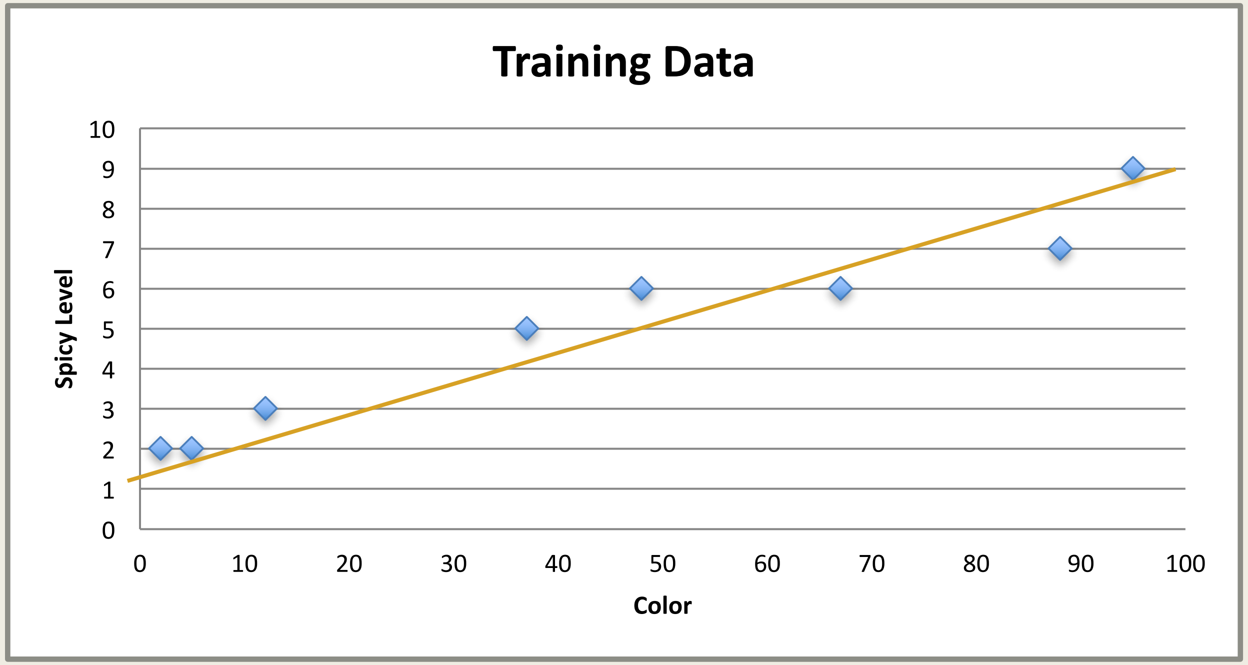
- Feedback Loop
  - *Observe how users interact with your product and make use of that data*



LINEAR REGRESSION FROM SCRATCH

Linear Regression

Model representation: $p(x) = \boldsymbol{\theta_0} + \boldsymbol{\theta_1} x$



Cost Function (Training Error)

$$J(\boldsymbol{\theta_0}, \boldsymbol{\theta_1}) = \frac{1}{2m} \sum_{i=1}^{m} \left(p_{\boldsymbol{\theta}}(x^i) - y^i\right)^2$$

$Select\ \boldsymbol{\theta_0}, \boldsymbol{\theta_1}\ that\ minimizes\ J(\boldsymbol{\theta_0}, \boldsymbol{\theta_1})$

Parameters Learning

Gradient descent algorithm

$$\text{repeat until convergence } \{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$
$$(\text{for } j = 1 \text{ and } j = 0)$$
$$\}$$

Linear Regression Model

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Parameters Learning

Gradient descent algorithm

$$\frac{\partial}{\partial\theta_0} J(\theta_0, \theta_1)$$

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)}$$

$$\theta_1 := \theta_1 - \alpha \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)}}$$

}

$$\frac{\partial}{\partial\theta_1} J(\theta_0, \theta_1)$$

update θ_0 and θ_1 simultaneously