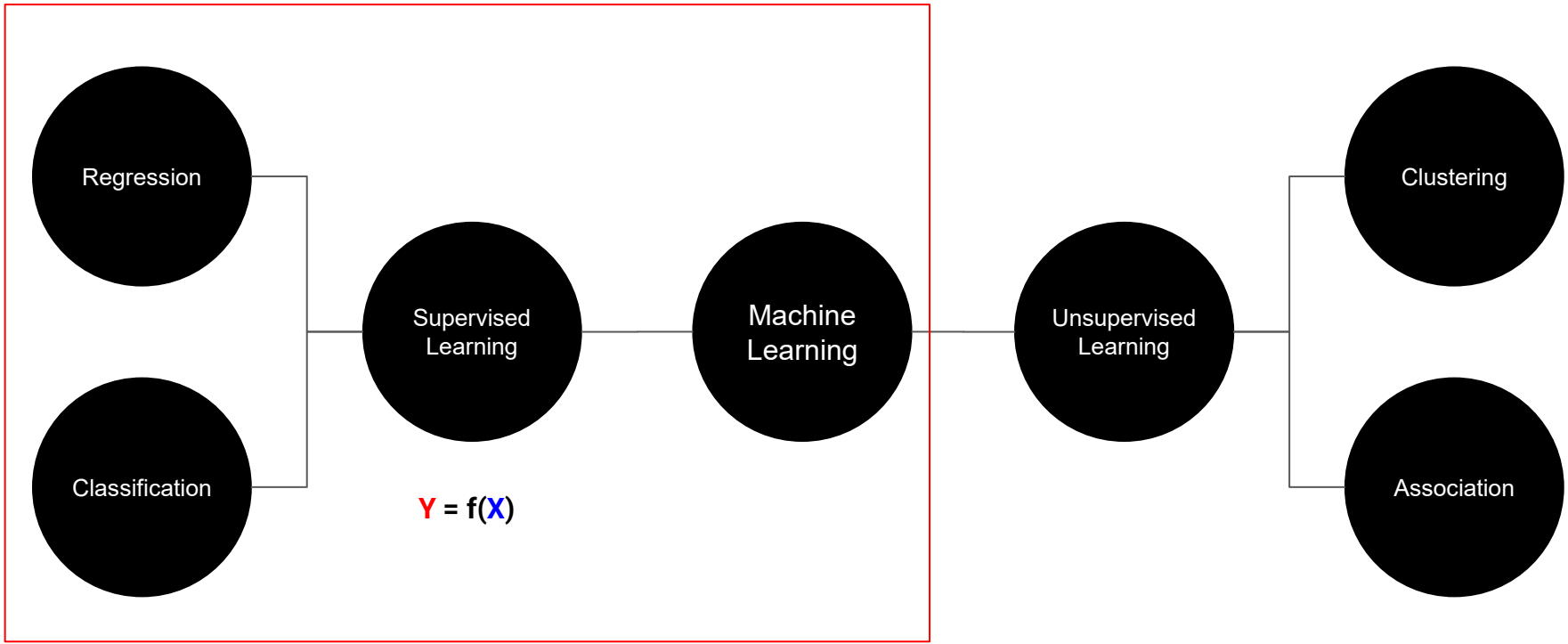
# Data Prediction Model and Machine Learning

**Online course #3**

Learning Type


Regression
Classification
Supervised Learning
Machine Learning
Unsupervised Learning
Clustering
Association
Y = f(X)

Supervised Learning

You, human
(Teacher, 쌤)

Machine
(Student, 과외돌(순)이)

Unsupervised Learning


Supervised Learning

- Solving problems with correct answers

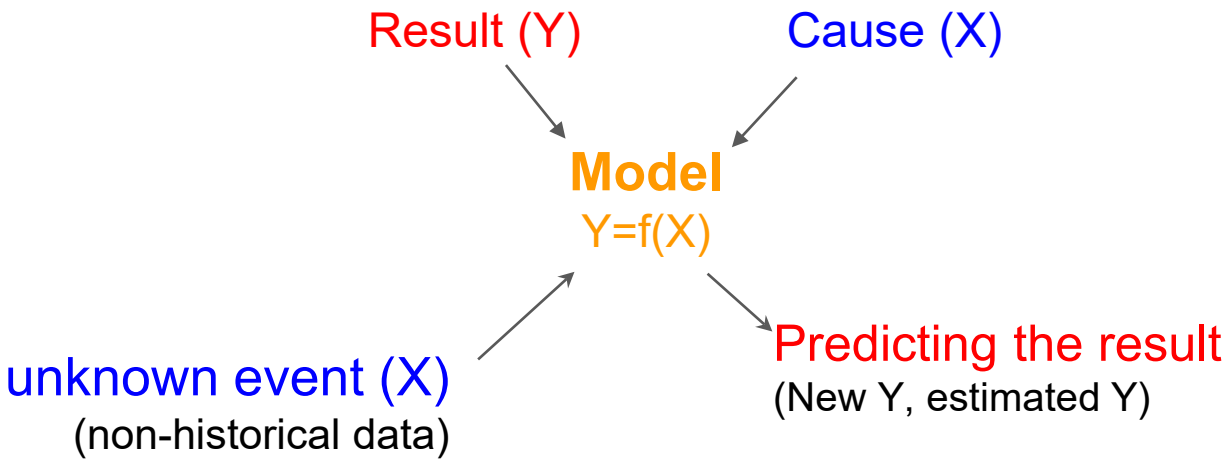
Unsupervised Learning

- Solving problems without correct answers.
- To reveal a new meaning or relationship through observation

Supervised Learning

Solving problems with correct answers

Correct answers ← from **history** (historical data)



Supervised Learning

Correct answers ← from **history** (historical data)

**history**

Cause (X)

Result (Y)

| Date | Day | Temp. | Sales |
|---|---|---|---|
| 2020.9.1. | Mon | 25 | 50 |
| 2020.9.2. | Tue | 24 | 49 |
| 2020.9.3. | Wed | 23 | 46 |
| 2020.9.4. | Thu | 27 | 52 |
| 2020.9.5. | Fri | 26 | 50 |
| 2020.9.6. | Sat | 25 | ?? |

**Model**

Y=f(X)

Supervised Learning

Correct answers ← from **history** (historical data)

| Date | Day | Temp. | Sales |
|---|---|---|---|
| 2020.9.1. | Mon | 25 | 50 |
| 2020.9.2. | Tue | 24 | 49 |
| 2020.9.3. | Wed | 23 | 46 |
| 2020.9.4. | Thu | 27 | 52 |
| 2020.9.5. | Fri | 26 | 50 |
| 2020.9.6. | Sat | 25 | Y |

Cause (X)

Result (Y)

**Model**
Y=f(X)

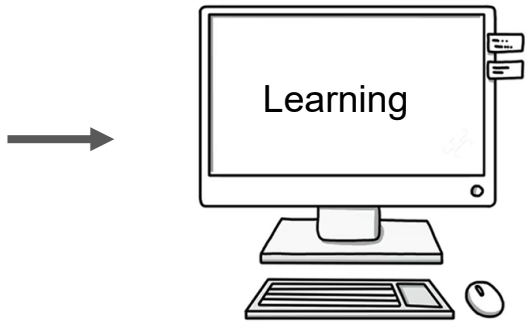
unknown event (X)
(non-historical data)

Predicting the result
(New Y, estimated Y)

Supervised Learning

Independent var. Dependent var.

| Temp. | Sales |
|---|---|
| 20 | 40 |
| 21 | 42 |
| 22 | 44 |
| 23 | 46 |



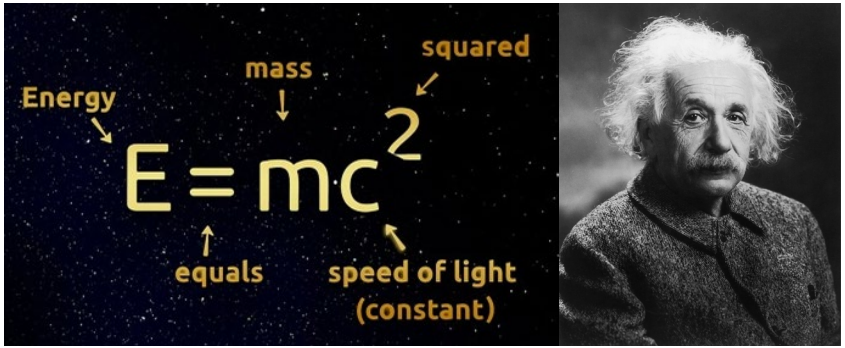

**Model**
Sales = Temp. × 2

**Model**
Independent var. × 2

# F=ma

Force = mass x acceleration

$$F = G\frac{m^1 m^2}{r^2}$$



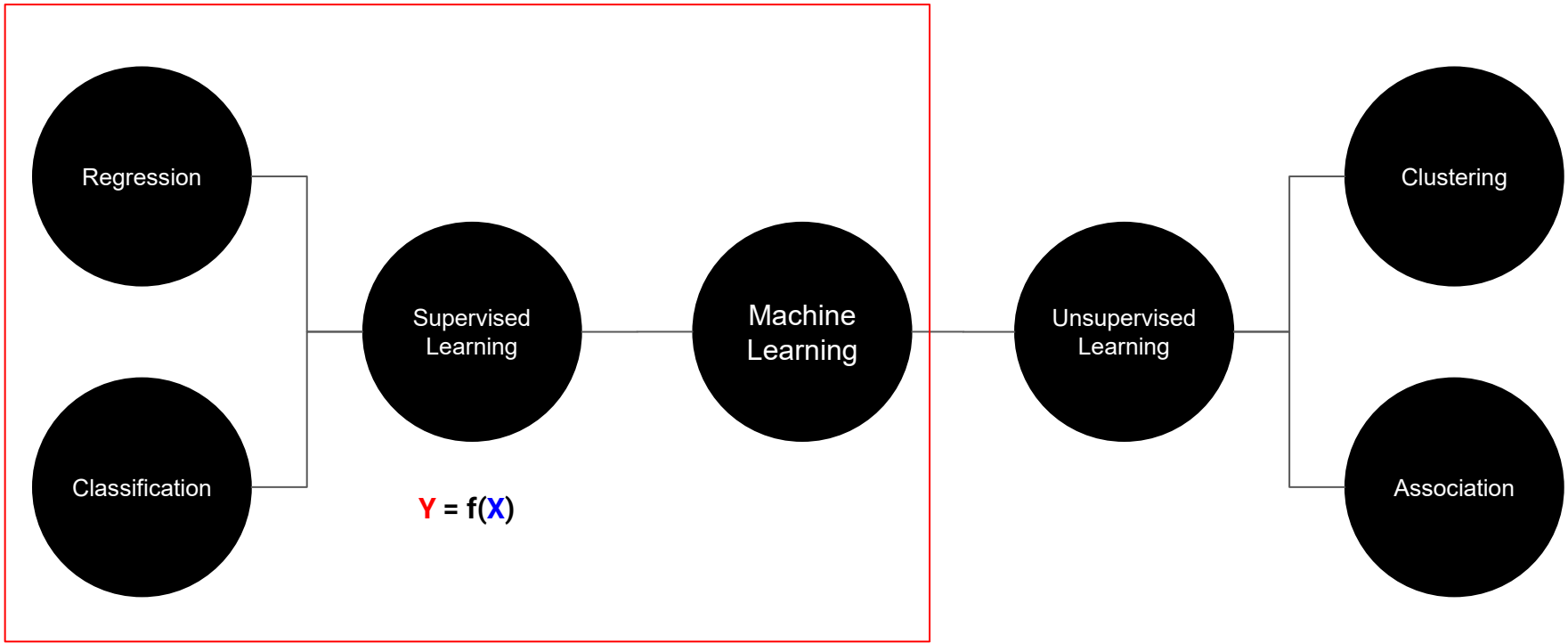

Supervised Learning

Supervised Learning

**Machine Learning**

**Popularization of the formula**
(공식의 대중화)

**Model**
Independent var. × 2


Regression
Classification
Supervised Learning
Machine Learning
Unsupervised Learning
Clustering
Association
Y = f(X)

| Temp. | Sales |
|---|---|
| 20 | 40 |
| 21 | 42 |
| 22 | 44 |
| 23 | 46 |

**Target:** Numeric variables
(Quantitative measure)

→

**Regression**
(회귀 분석)

| Speed (km/h) | Ticket |
|---|---|
| 60 | No ticket |
| 63 | No ticket |
| 65 | Ticket |
| 80 | Ticket |

**Target:** Dummy variables
(Categorical measure)

→

**Classification**
(분류 분석)

# Data Prediction Model and Machine Learning

**Online course #3**

Classification

# Preview

## Classification

- Response (or output, dependent) variable: discrete value (categorical variable)
- E.g.) 1(Patient) 0(Normal) or 2(Patient), 1(Observation), 0(Normal)
- E.g. ) Mobile carrier customer management
  - Classify customers into 3 (most loyal), 2 (loyal), 1 (medium), and 0 (dissatisfied)
  - For customers in category 3, sometimes providing good words,
    For customers in category 0, providing benefits like reduced fee, etc.

## Models for classification

- Decision tree
- Random forest
- k-NN (k-nearest neighbors)
- SVM (Support Vector Machine)
- Neural network
- Deep learning, etc.

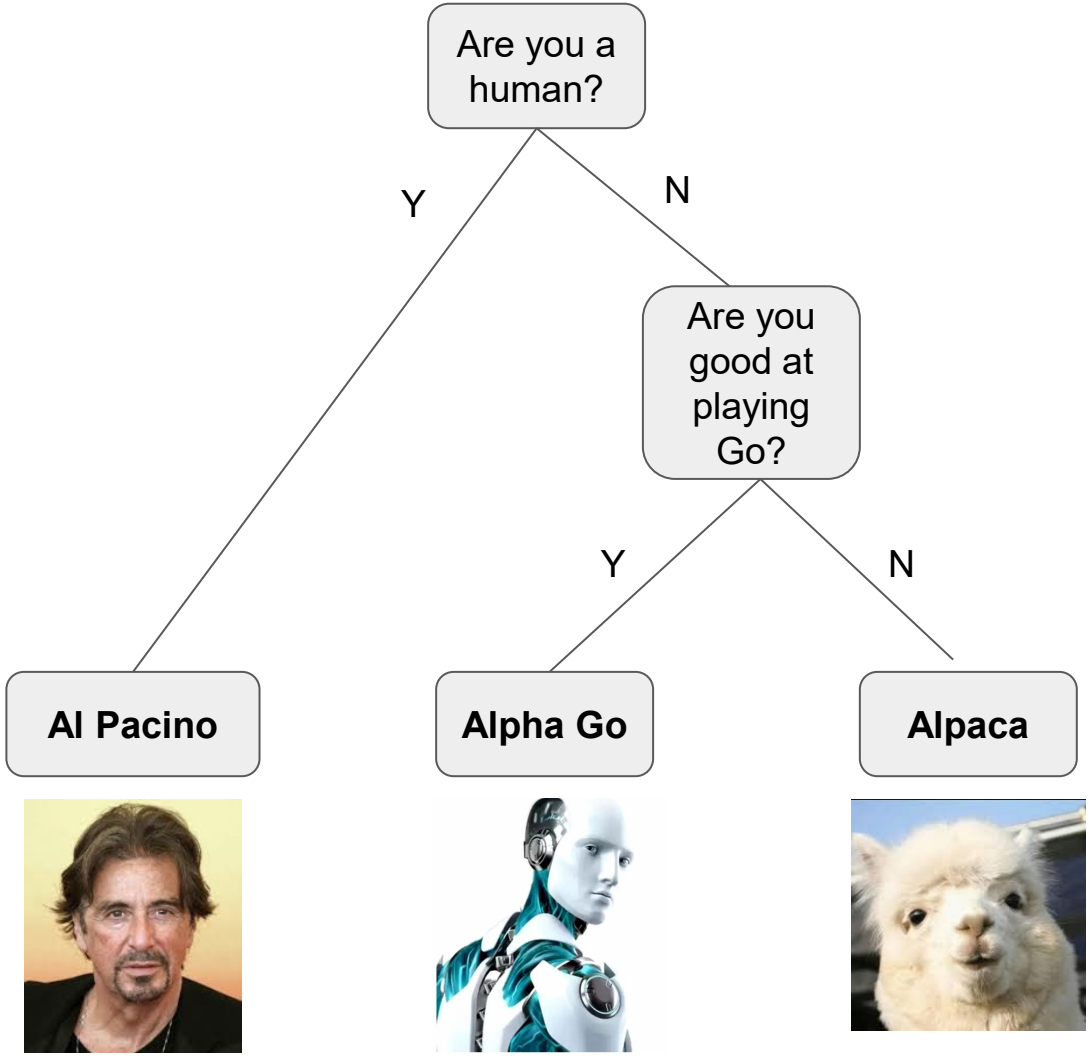
# Preview

## Regression models for classification

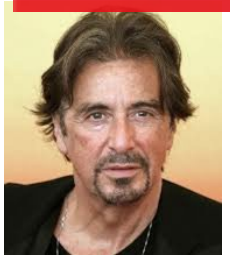
- Logistic regression: Regression model but for solving classification problems
- We call this kind of regression as generalized linear model (glm), will learn this model after understanding linear model (lm).
  - F.Y.I) Linear regression model: lm

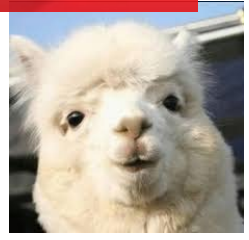
    Generalized linear model: glm with an option “family=binomial”

# Decision Tree

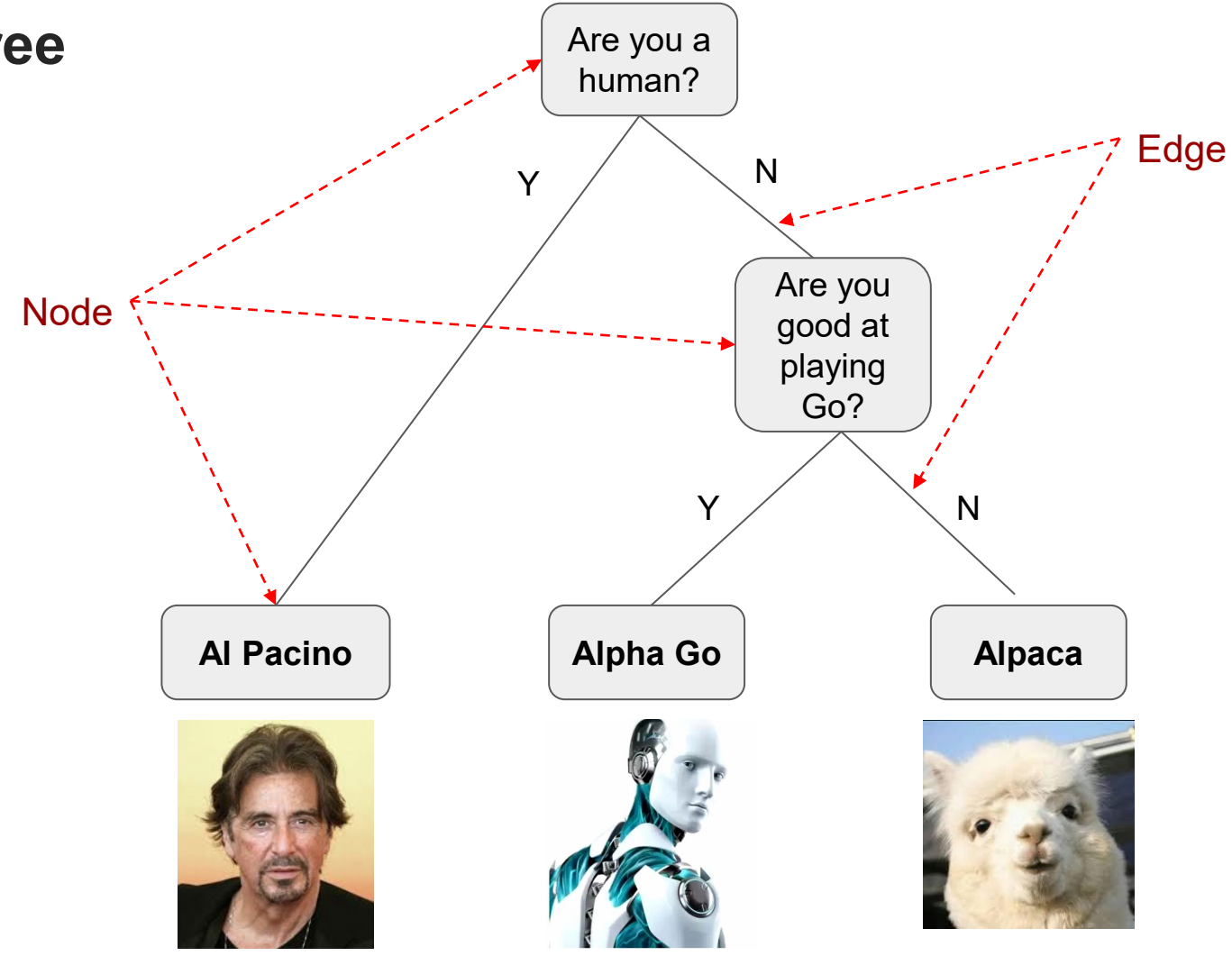


# Decision Tree

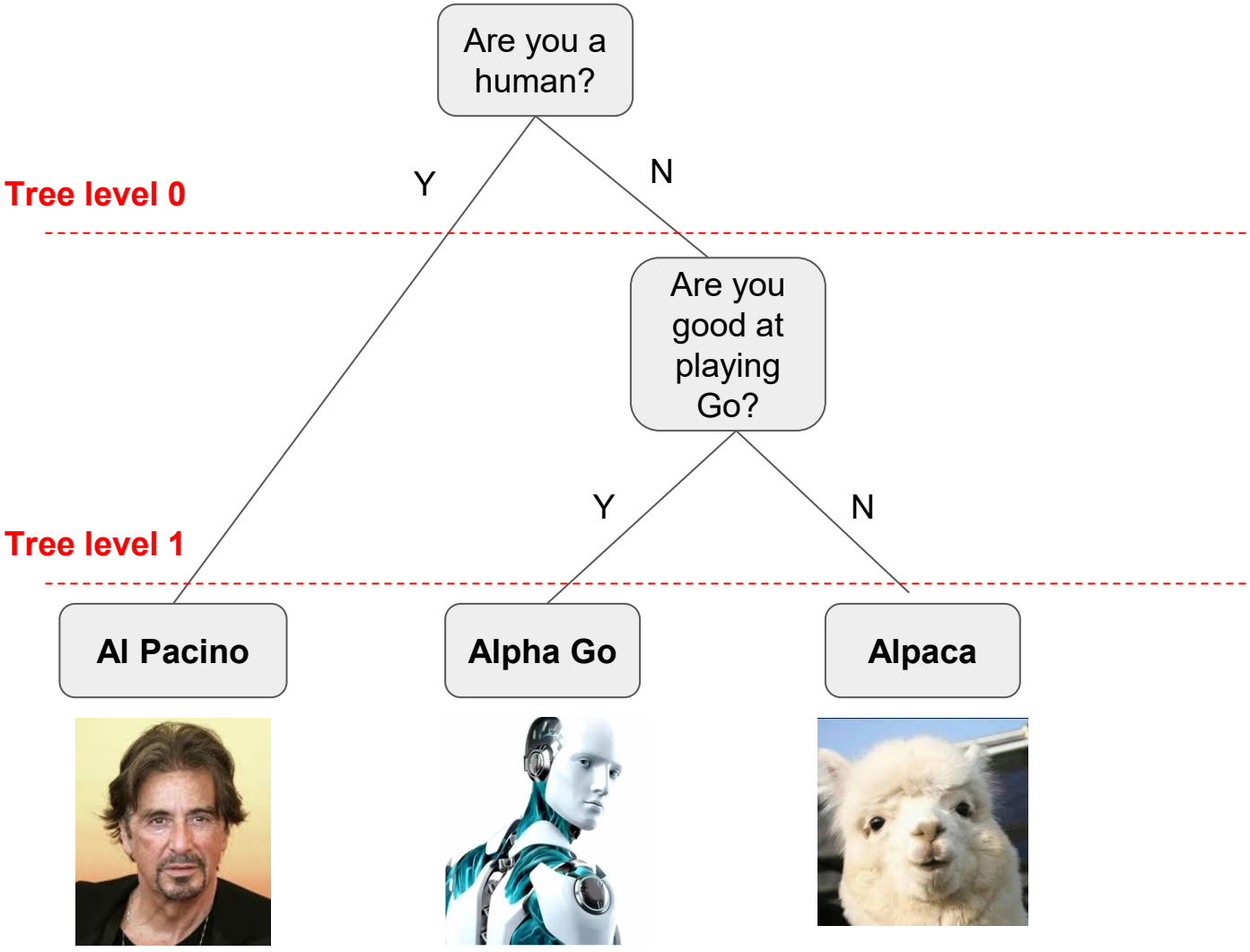





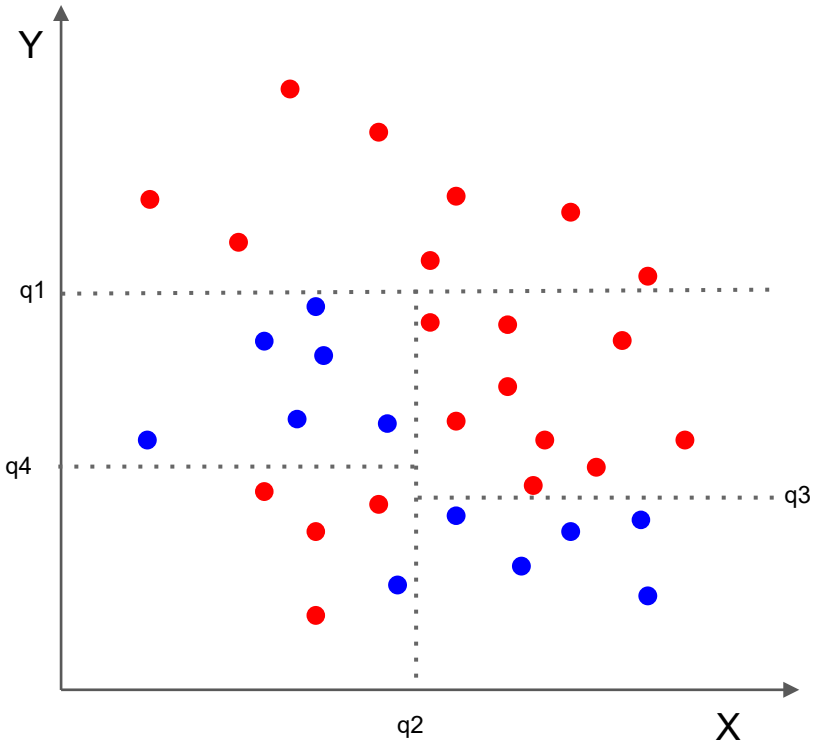
# Decision Tree

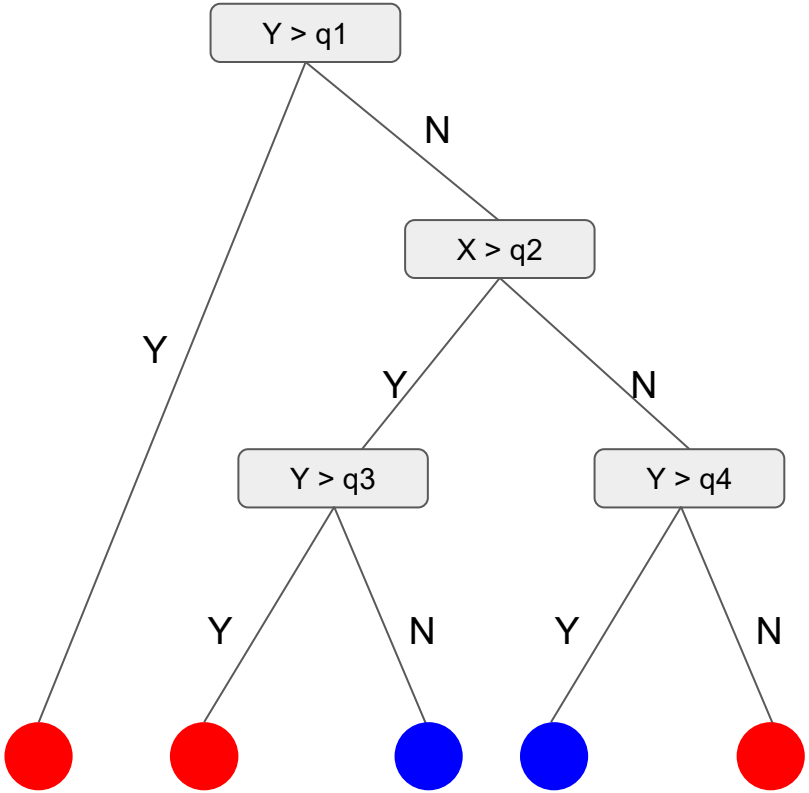


# Decision Tree

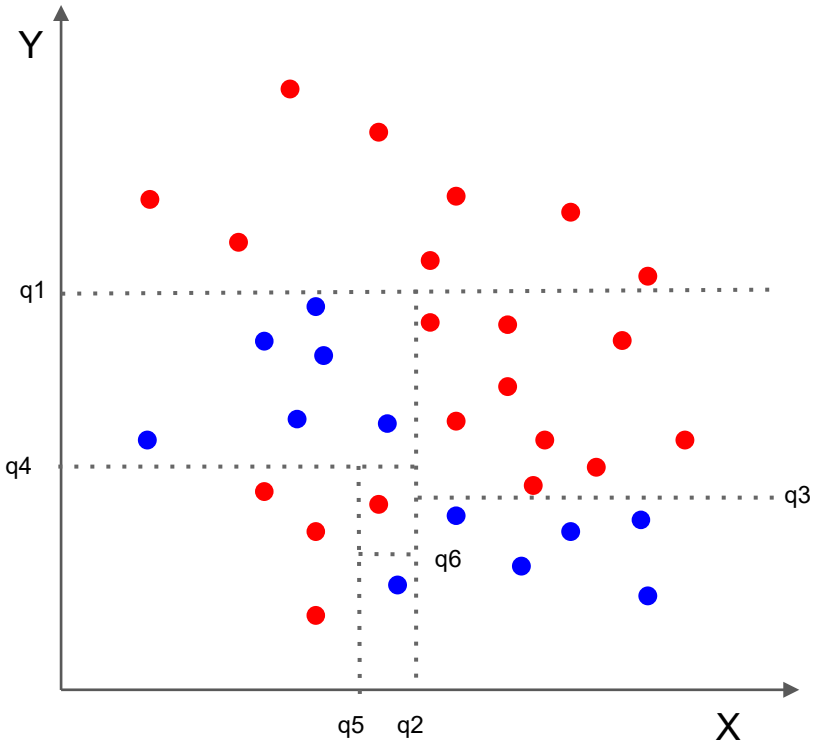


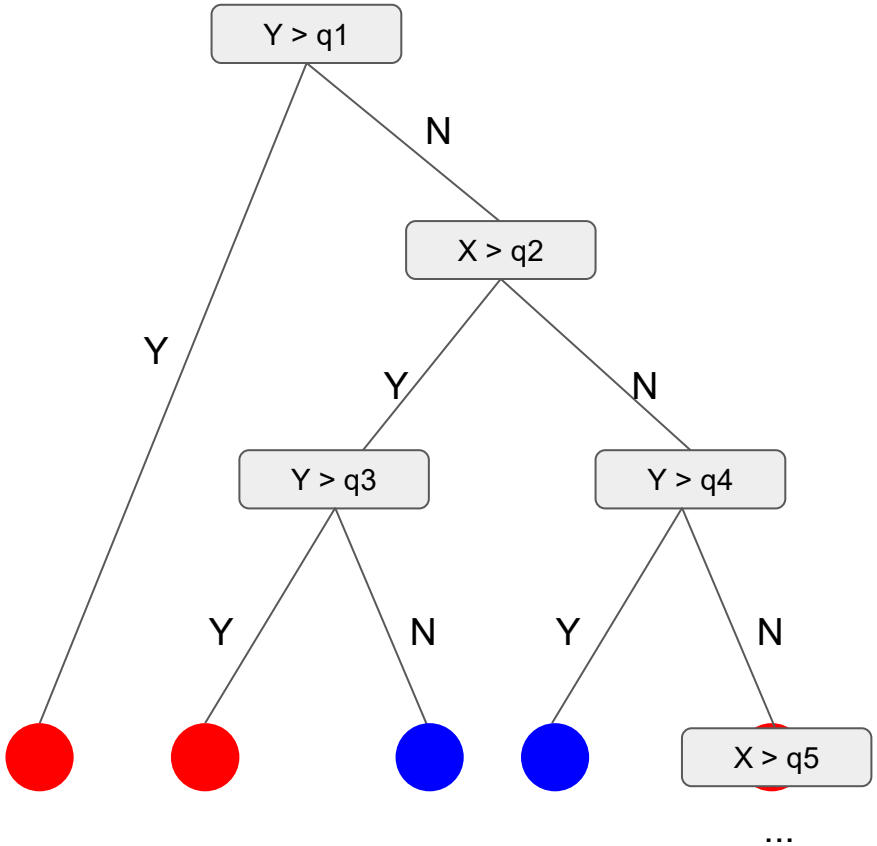
# Decision Tree Intuition



# Decision Tree Intuition




**How deep do we need to go??**

# How deep do we need to branch out?

If the decision tree branches out to the deepest it can?

The accuracy rate: 100%

# How deep do we need to branch out?

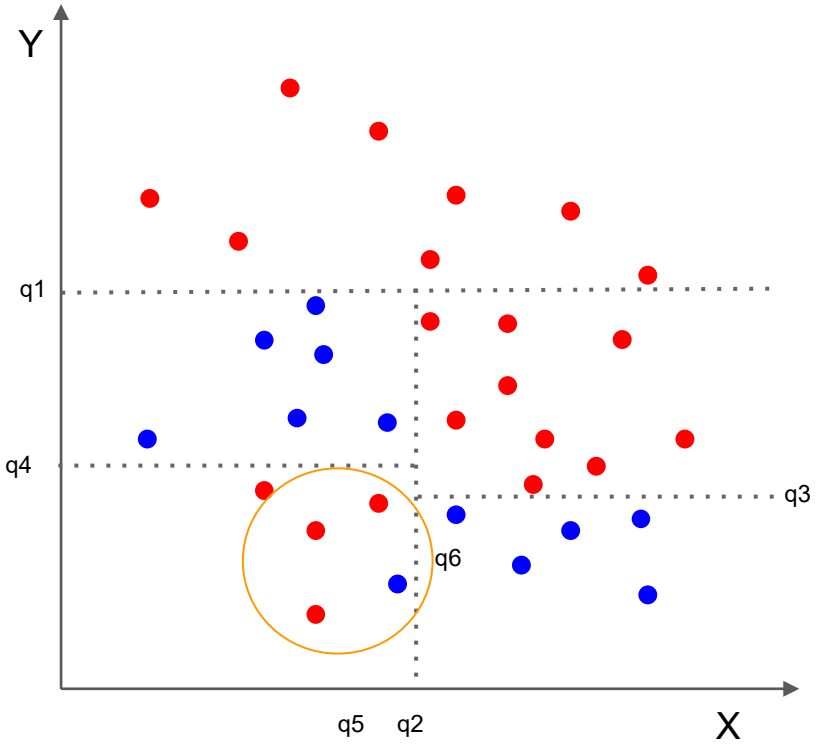
If the decision tree branches out to the deepest it can?

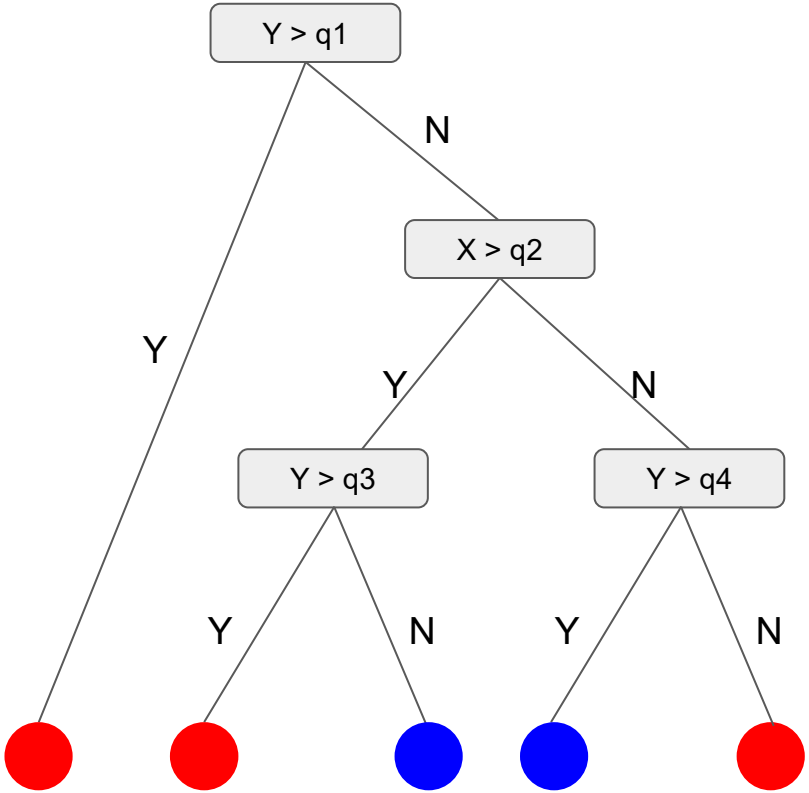
Another issue comes out: **Overfitting**

# Overfitting

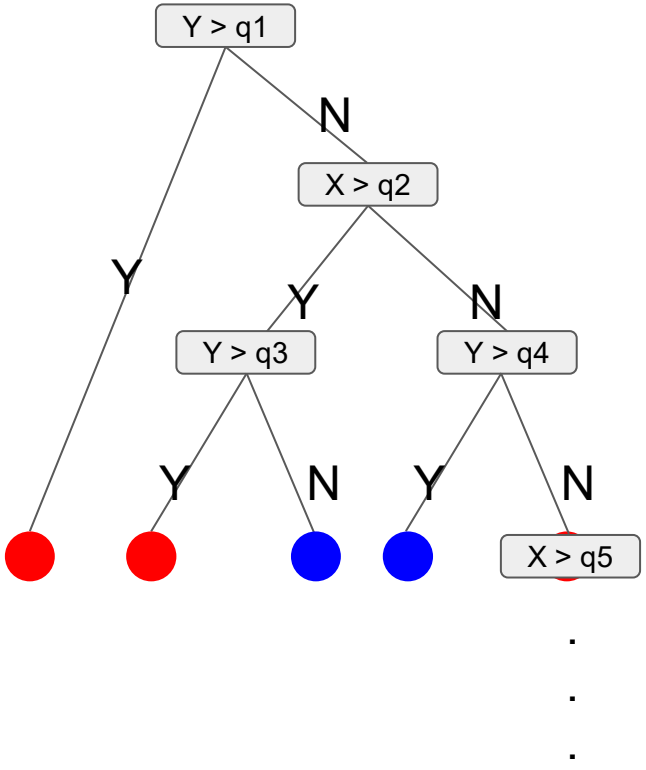
Overfitting is a phenomenon that tries to learn (or train with) the training set too completely, so that it spoils predicting performance for new samples.

# Decision Tree Intuition

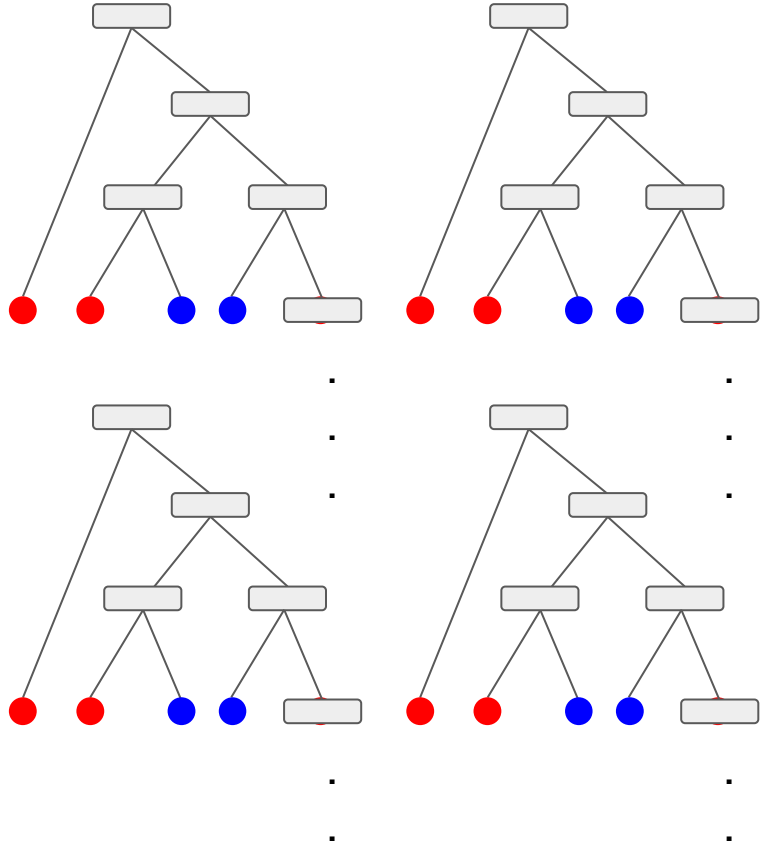


## Decision Tree




## Random Forest



- How the algorithm choose the appropriate condition?
- How the algorithm choose whether it needs to go further branching out or stops?

To explain the questions above, we have to start learning the concepts about Entropy, Information Gain function, Minimizing the objective function, Information Gain Ratio, and so on.

**Let’s just learn how to apply this amazing technique firstly!!**