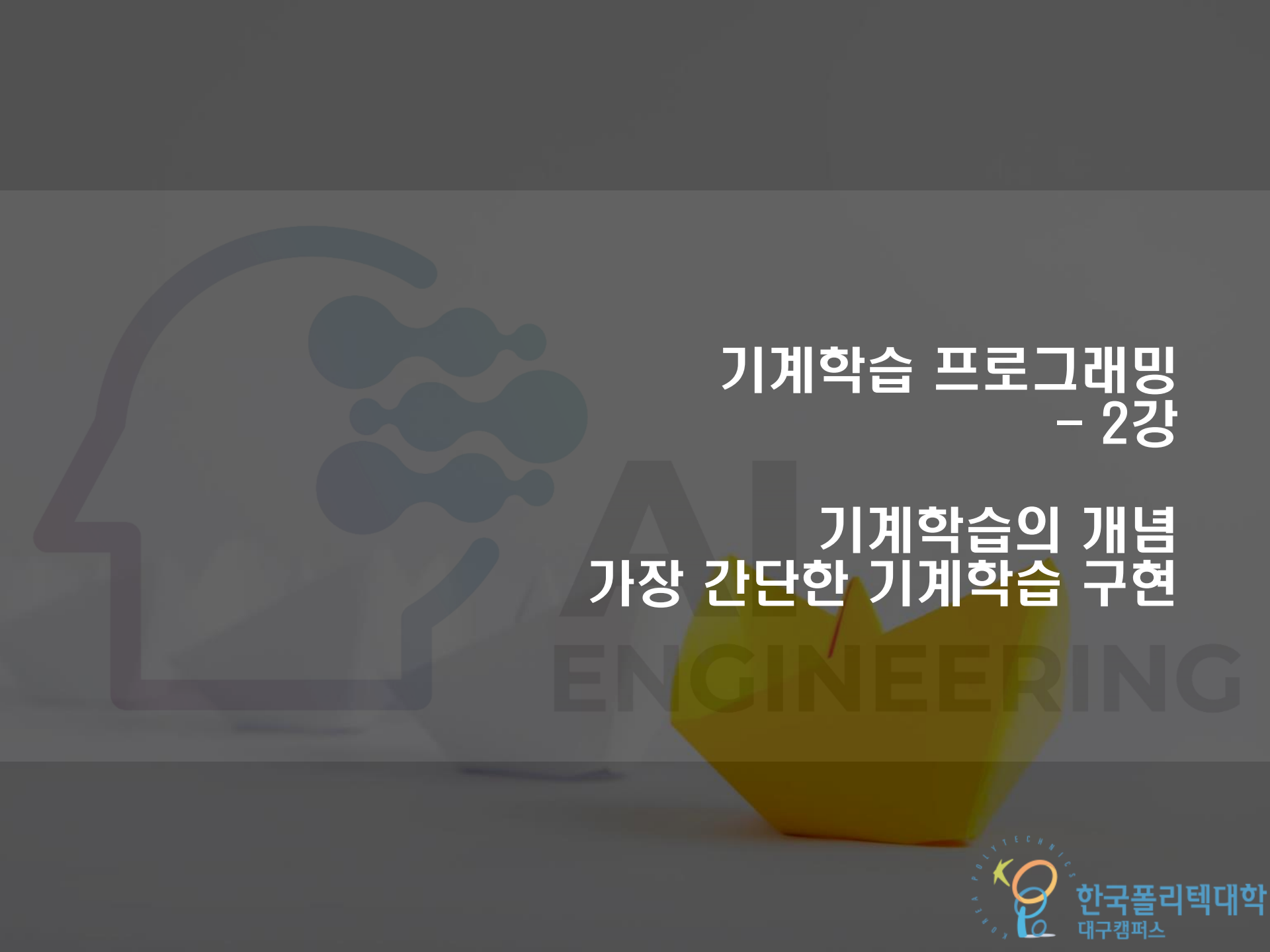
기계학습 프로그래밍

한국폴리텍대학 대구캠퍼스
AI엔지니어링과 강현우

기계학습 프로그래밍
- 2강

기계학습의 개념
가장 간단한 기계학습 구현



한국폴리텍대학
대구캠퍼스

기계학습의 등장 배경

◆ 왜? 기계 학습?

➢1980년대 인공 지능 연구의 대표적인 방법
= 전문가 시스템

➢ 사람이 직접 많은 수의 규칙을 만드는 것을 전제

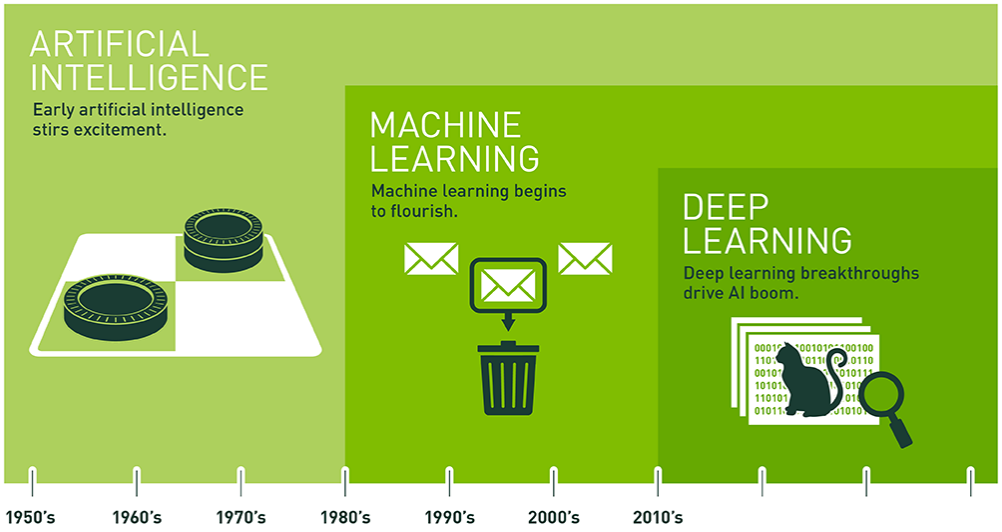
➢ 규칙을 정확하게 규정할 수 없는 분야는 어떻게?

➢ 사람조차 정확한 원리를 모르는 영역에 대해 요구

기계가 학습한다는 것?

◆ Machine Learning

- ➢ 어떤 컴퓨터 프로그램이 T라는 작업을 수행한다.
- ➢ 이 프로그램의 성능을 P 라는 척도로 평가했을 때
- ➢ 경험 E를 통하여 성능이 개선된다면
- ➢ 이 프로그램은 학습을 한다고 말할 수 있다.



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

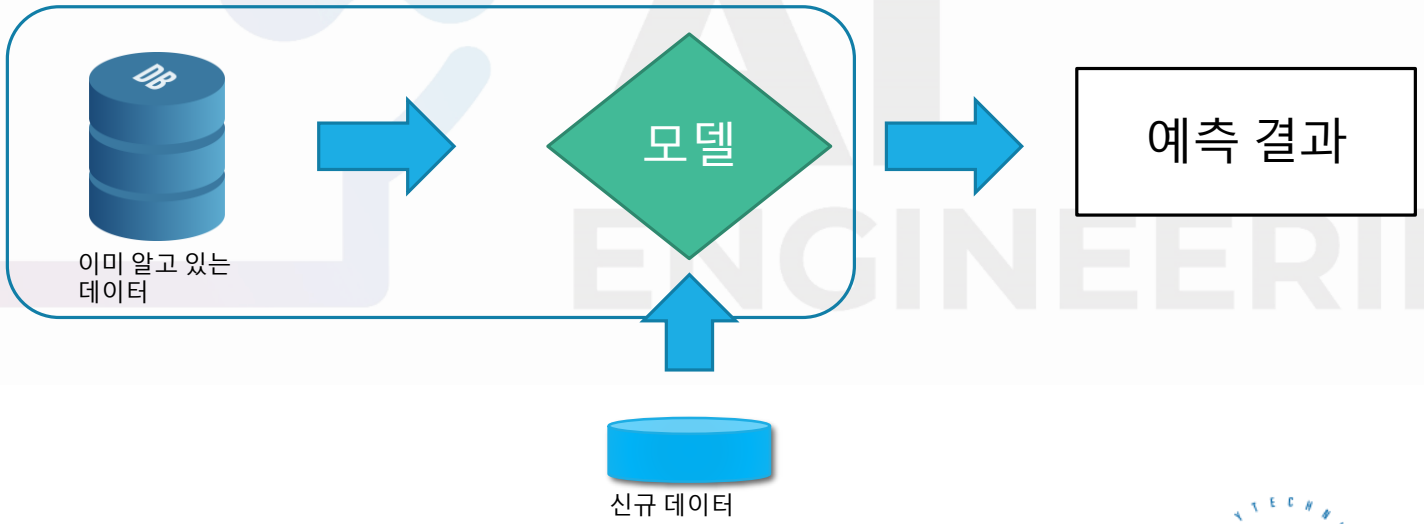
출처: Nvidia.com

기계 학습

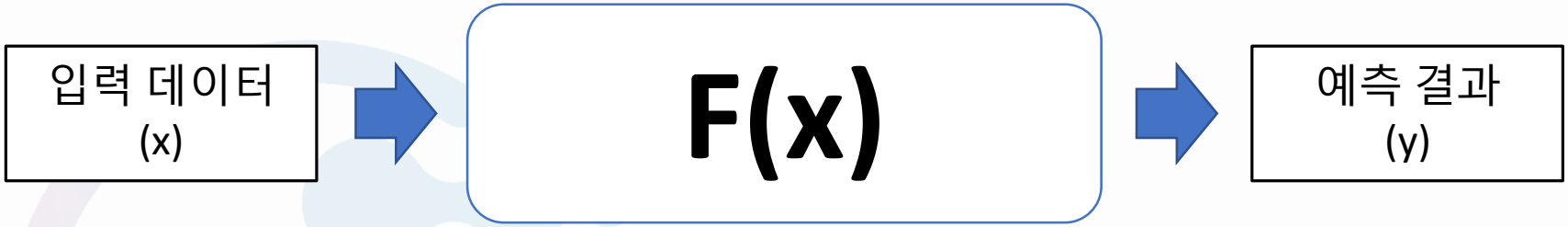
◆ Machine Learning

➢ **이미 알고 있는 데이터 (학습 데이터)로 모델을 생성해내는 과정**

✓데이터에서 패턴을 추출하여 스스로 지식을 획득



기계 학습



- 과거에는 F(x) 를 만드는데 집중
- 머신 러닝은 알고있는 데이터 x와 결과 y로 F(x) 를 만들어 내는 것

So What?

◆ 그래서, 무슨 문제를 풀고 싶은 건데?

- ➢ 분류 (Classification)
- ➢ 군집화 (Clustering)
- ➢ 회귀 (Regression)
- ➢ …
- ➢ 세상은 넓고 문제는 많다.

세상은 넓고 문제는 많다

- Kaggle
  - https://www.kaggle.com/
  - 예측 모델 및 분석 대회 플랫폼

- Competitions
  - HuBMAP – detect functional tissue units (FTUs)
  - **RANZCR CLiP** – detect the position of catheters
  - **VinBigData** – detect abnormalities in X-ray
  - ...

기계 학습 준비

◆ 어떤 문제를 머신 러닝으로 풀고 싶다면

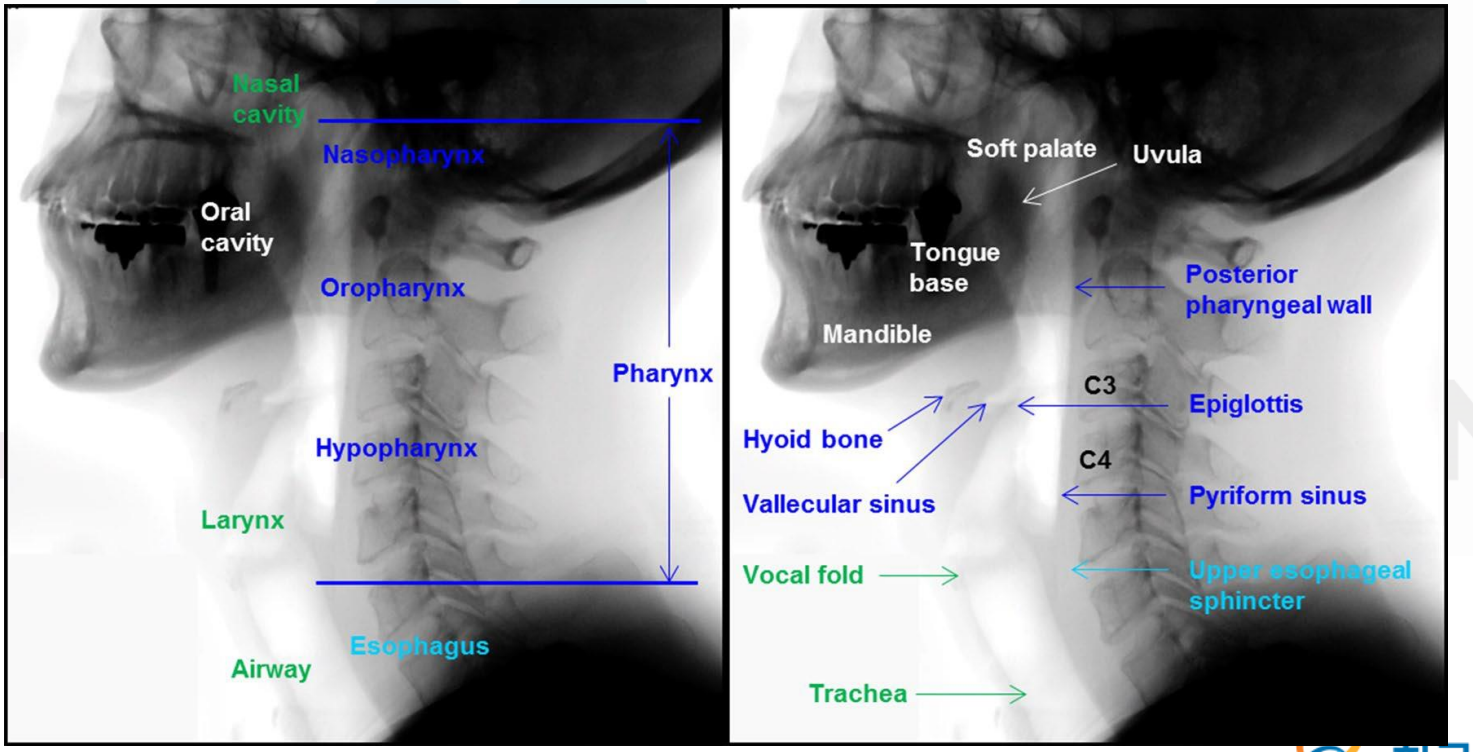
➢ **어떤 부류의 문제인지 파악**

➢ **데이터 세트**
- ✓ 학습 데이터
- ✓ 테스트 데이터
- ✓ Optional – 검증 데이터

➢ **모델을 설계**
- ✓ 알고리즘

Domain Knowledge

- Engineer 는 엔지니어…?
  - 이 사진은 뭐죠? → 논문
  - Kaggle에서 제공되는 것이 뭐였죠?



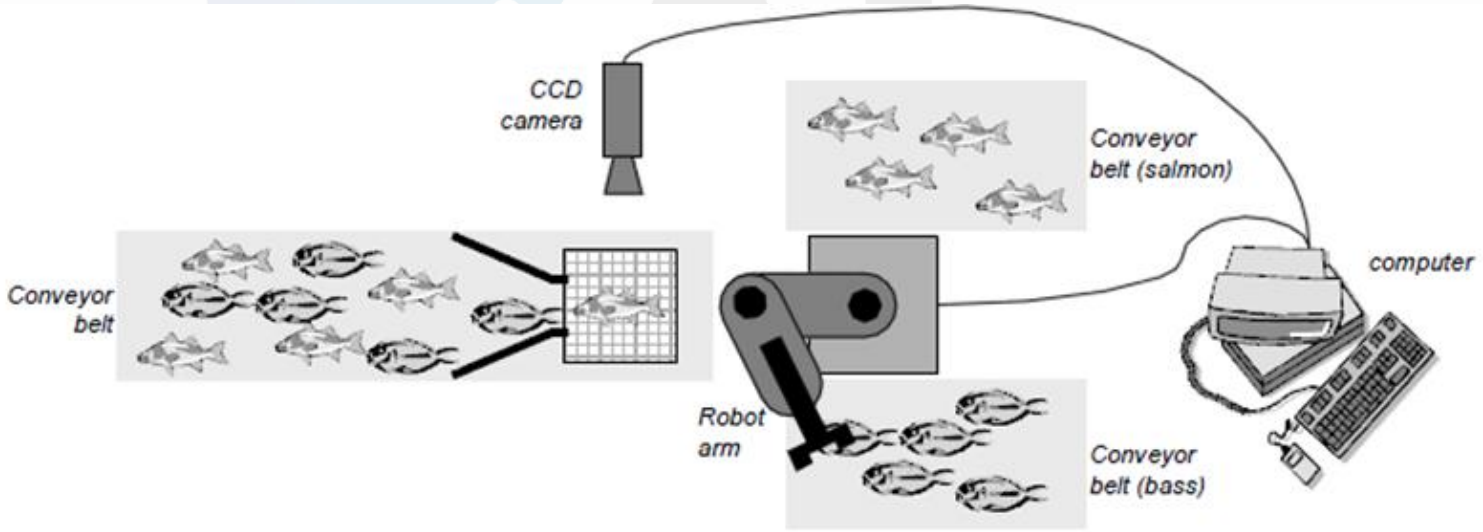
여러분들은

- 여러분들의 전문 분야에서
  혹은 전문분야가 아니더라도
  인공지능이나, 머신 러닝을 이용하여
  풀고 싶었던 문제가 있나요?

기계 학습 예제

◆ 시나리오

- ➢생선처리 공장에서 연어(salmon), 농어(sea bass)를 분류
- ➢CCD 카메라를 갖춘 비전 시스템
- ➢영상을 분석하여 로봇 암을 제어하여 생선을 이동



기계 학습 설계

◆ 데이터 수집

◆ 전처리

◆ 특징?

➢ 길이, 밝기 … Domain 지식이 없으니까…

◆분류기 설계

➢모델 선정, 분류기 학습

◆성능 테스트

➢학습에 사용하지 않은 데이터 사용

Feature – 특징

◆ 구분 대상을 어떻게 표현해야 하는가?

- ➢ 연어와 농어를 2가지 특징으로 표현
- ➢ [특징1: 길이, 특징2: 밝기, … 특징N: something]

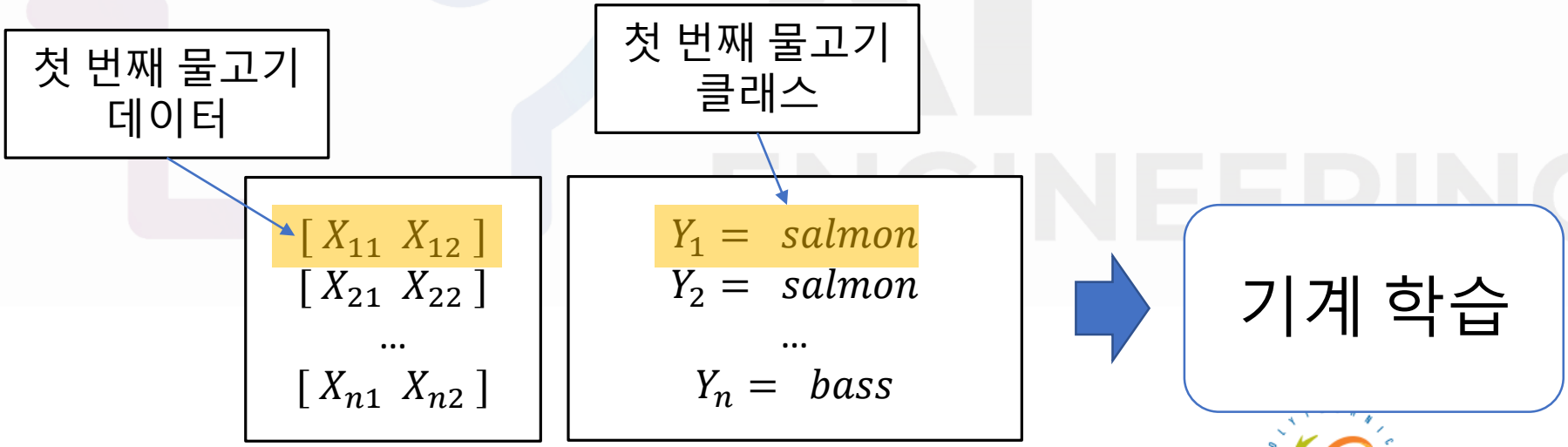
◆ Feature

- ➢ 관찰 대상에게서 발견된 개별적이고 측정가능한 경험적 속성
- ➢ 독립적인 변수를 잘 선택하는 것이 성공적인 분류를 위해 중요

지도 학습

◆ 지도 학습

- ➢ 훈련 데이터로부터 하나의 함수를 유추해내기 위한 기계 학습의 한 방법
- ➢ 훈련 데이터는 일반적으로 입력 객체에 대한 속성을 벡터 형태로 표현
- ➢ 각각의 벡터에 대해 원하는 결과가 무엇인지 표시



Feature Space – 특징 공간

◆ Feature Space

- ➢ 특징 벡터를 표현하는 공간
- ➢ 특징의 개수에 따라 다차원 공간으로 구성

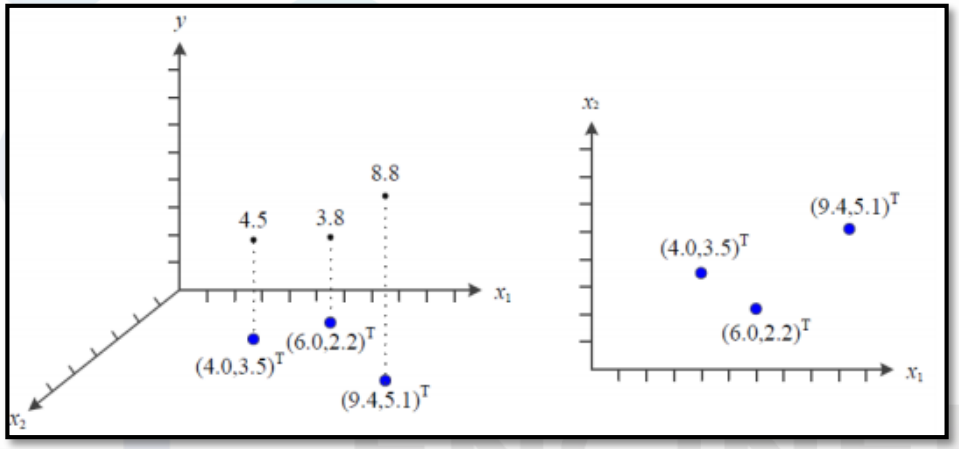
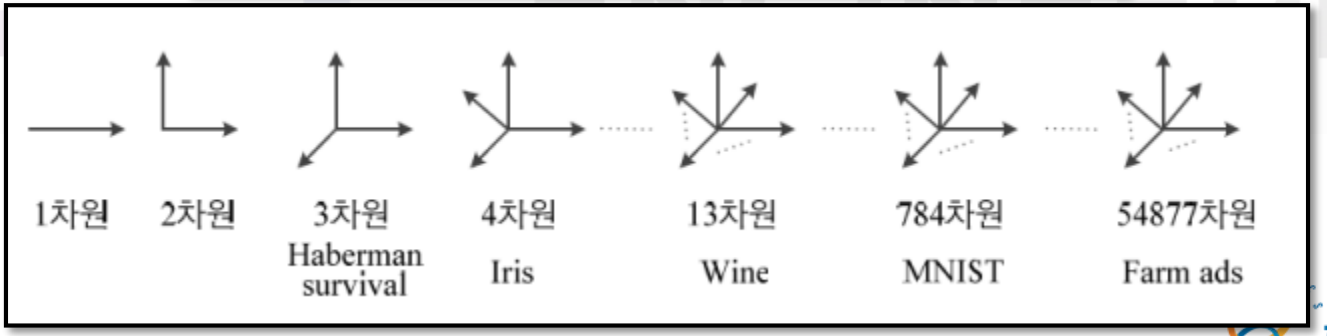


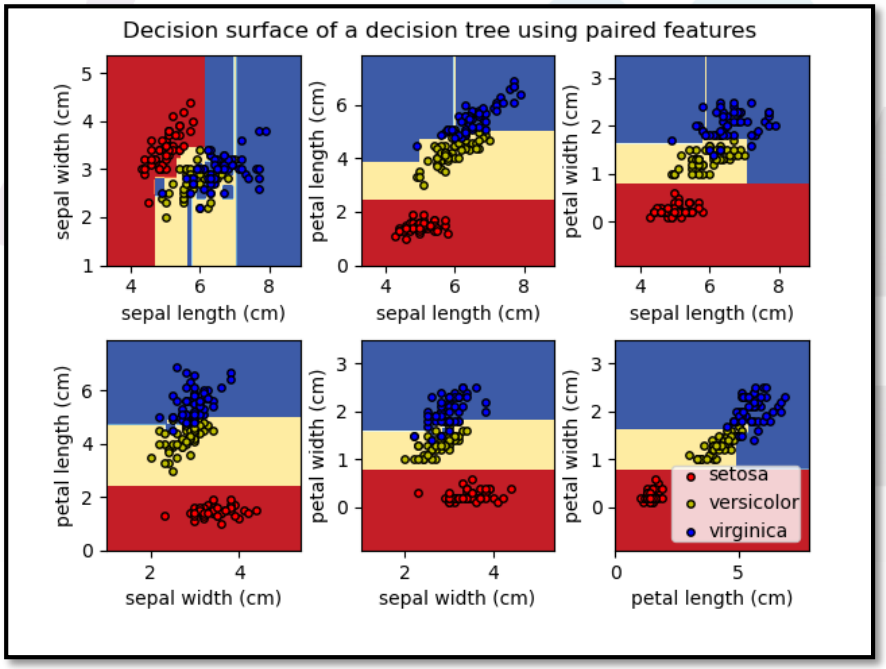
그림 출처: 기계학습(오일석 저)



Classification – 분류

◆ 특징 공간에서 Classification

➢특징 공간에서 대상을 분류할 수 있는 결정 경계를 구하는 것



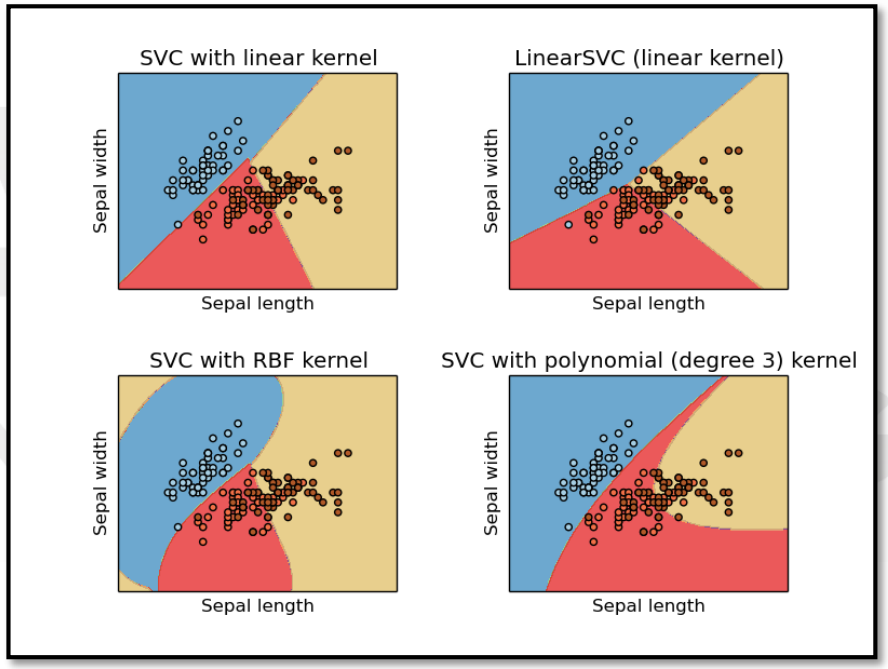


그림 출처: scikit-learn.org

모델 설계

- **Domain Knowledge**
  - 농어(sea bass)는 연어(salmon)보다 일반적으로 길다
- **선정 특징: Length**
- **분류 규칙**

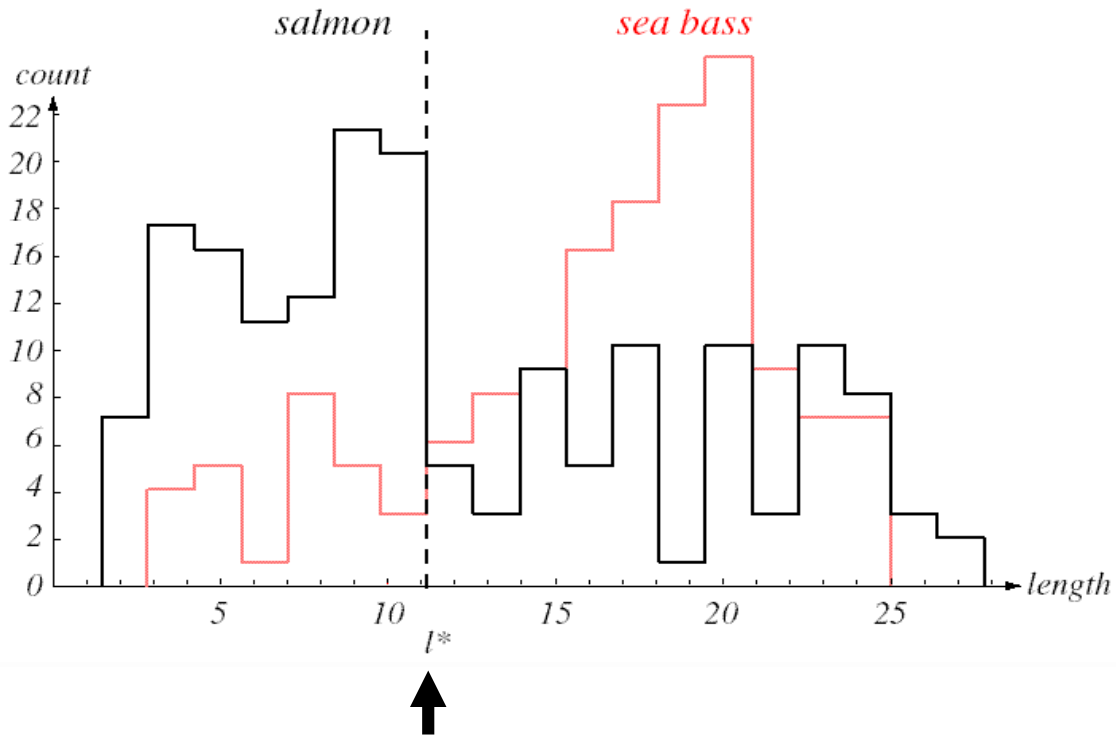
  If *Length* >= *l** then *sea bass*

  otherwise *salmon*

- ***l**** **를 고르는 방법?**

모델 학습

◆두 생선에 대한 Length 히스토그램



오분류가 제일 적은 지점

Training error: 90 / 316 = 28%

학습 결과

◆실험 결과

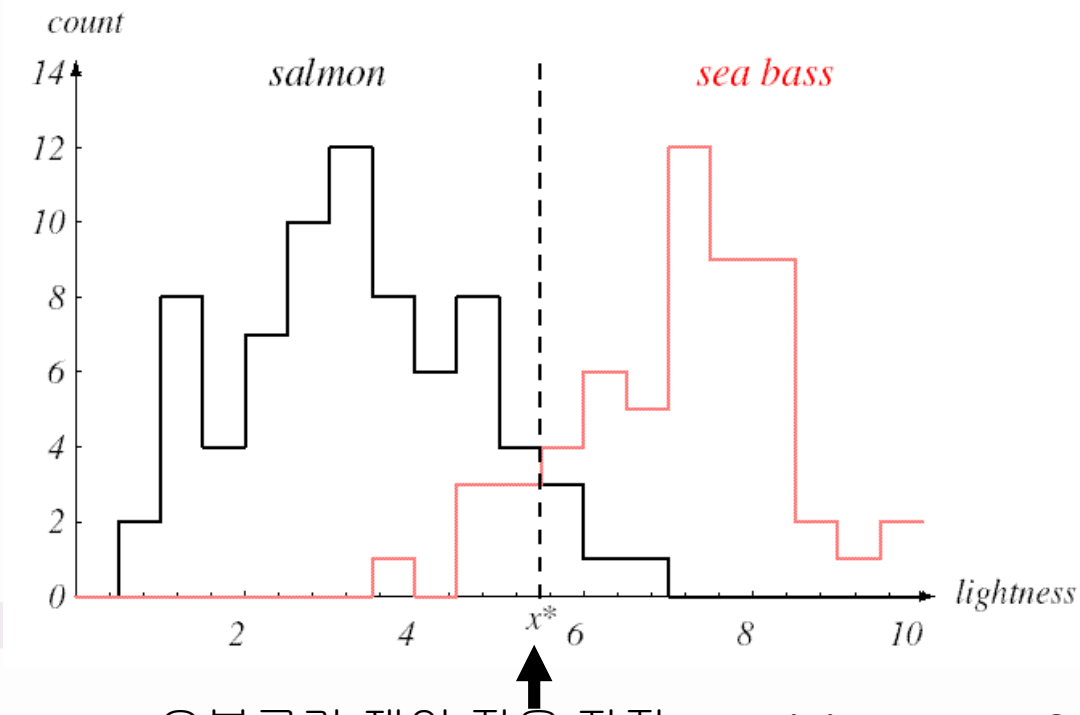
- ➢학습 데이터에 대한 분류율: 28%
- ➢너무 낮다!!!
- ➢다른 특징을 시도

◆ 밝기?

- ➢ New Feature -> Lightness

모델 학습

◆ 두 생선에 대한 Lightness 히스토그램



오분류가 제일 적은 지점 Training error: 16 / 316 = 5%

분류가 아까 보다 잘 되었음!

최선입니까?

◆단일 특징으로는 만족스럽지 못하다

◆복합 특징을 사용

➢Sea bass 가 salmon 보다 보통 폭이 넓다

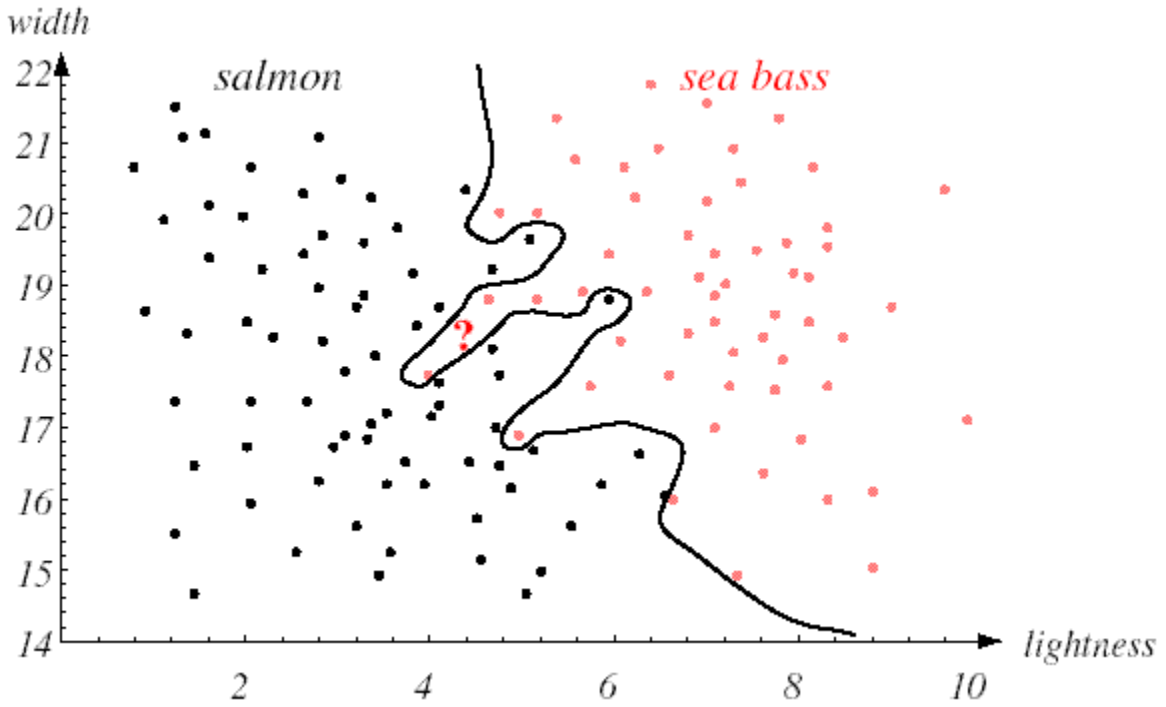
◆일반적으로 특징 공간의 차원이 높을 수록

➢ 분리에 유리

◆ 보다 복잡한 분류 모델을 사용할 필요

분류 함수를 좀 더 복잡하게?

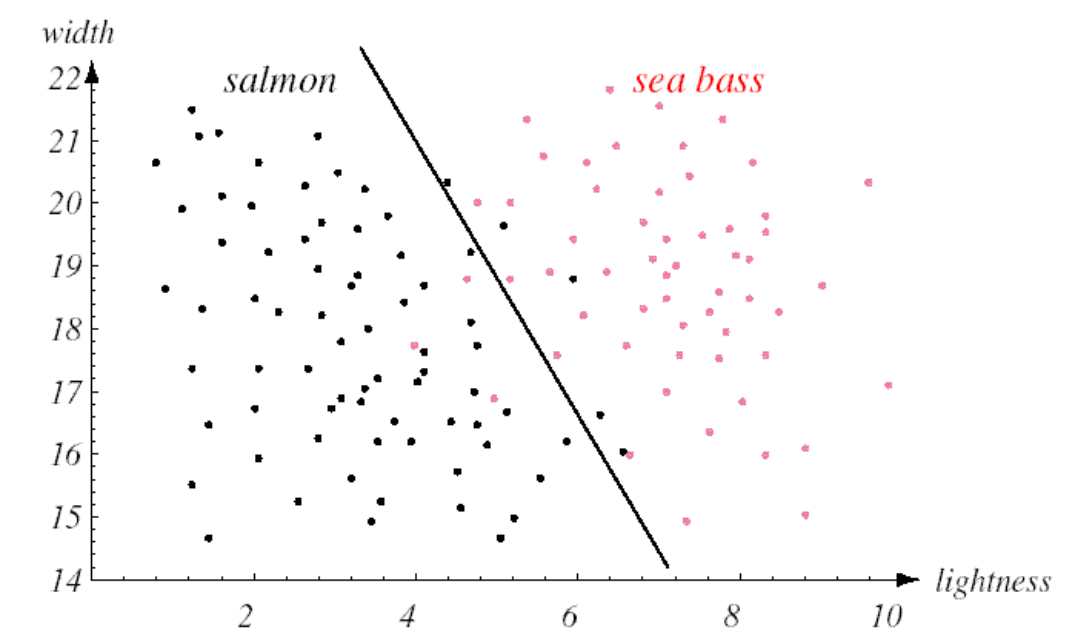
◆ 학습 데이터를 완벽하게 분류하는 모델



Complex decision function
Training error: 0 / 316 = 0%

일반화

◆ 앞 슬라이드의 분류 모델은 Overfitting



Linear decision function
Training error: 8 / 316 = 2.5%

Overfitting

◆ Overfitting 은 왜 생기나?

- ➢ 우리가 가지고 있는 데이터는 전체 데이터 중 얼마나 될까?
- ➢ 얼마나 일반화가 잘 되었는지는 어떻게 알지?
- ➢ 우리가 가지고 있는 데이터가 무진장 많다면?

모델 – ML Algorithm

◆분류

- ➢지도 학습 (Supervised Learning)
- ➢비지도 학습 (Unsupervised Learning)
- ➢강화 학습 (Reinforcement Learning)

◆ Algorithm

- ➢Support Vector Machine (SVM), Beyesian Network, Decision Tree, Random Forest, Neural Network…

◆ Deep Neural Network (심층 신경망)

ML Application

◆컴퓨터 비전

- ➢ 컴퓨터에서 카메라 등의 시각 매체를 통해 입력 받은 영상을 분석하여 유용한 정보를 생성하는 기술
  (ex. 보행자 검출, 얼굴 인식, 번호판 인식 등등)

◆ 데이터 마이닝

- ➢ 데이터 베이스 내에서 유용한 정보를 발견하는 기술
  (ex. 상품 추천, 마케팅)

◆ 자연어 처리

- ➢ 컴퓨터를 이용해 사람의 자연어를 분석하고 처리하는 기술
- ➢ 대량의 말뭉치 데이터를 활용하는 기계 학습 기반의 자연어 처리 기법이 주류

실습

◆ CSV 파일 읽기

- ➢ CSV : comma-separated values
- ➢ 몇 가지 필드를 쉼표(,)로 구분한 텍스트 데이터

◆ salmon_bass_data

- ➢ 3개의 column
- ➢ class : 물고기 종류
- ➢ length : 길이
- ➢ lightness : 밝기

```
Class,Length,Lightness
Salmon,2,0.8
Salmon,2,0.8
Salmon,2,1.2
Salmon,2,1.2
Salmon,2,1.2
Salmon,2,1.2
Salmon,2,1.2
Salmon,3,1.2
Salmon,3,1.2
Salmon,3,1.2
Salmon,3,1.6
```

파일 오픈

```
import java.io.FileReader;

public class Main {
    public static void main(String[] args) {
        FileReader fr = new FileReader("salmon_bass_data.csv");
    }
}
```

java: unreported exception java.io.FileNotFoundException; must be caught or declared to be thrown

- 파일이 없는 경우 에러가 발생할 수 있으니 반드시 처리되어야 한다는 경고
- 이럴 경우 try / catch 문법을 사용해야 한다.

Try / catch 문으로 변경

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;

public class Main {
    public static void main(String[] args) {
        FileReader fr;
        BufferedReader br;
        try
        {
            fr = new FileReader("./salmon_bass_data.csv");
            br = new BufferedReader(fr);
        }
        catch(FileNotFoundException e)
        {
            e.printStackTrace();
        }
    }
}
```

BufferReader를 사용하여
한 줄 씩 읽을 예정

java.io.FileNotFoundException: .\salmon_bass_data.csv1
(지정된 파일을 찾을 수 없습니다)

한국폴리텍대학
대구캠퍼스

ReadLine

```
fr = new FileReader("./salmon_bass_data.csv");
br = new BufferedReader(fr);

String line = br.readLine();
```

java: unreported exception java.io.IOException;
must be caught or declared to be thrown

◆ IOException 도 catch 하면 된다.

➢ ~~모옵시 불편…~~

```
try
{
    fr = new FileReader("./salmon_bass_data.csv");
    br = new BufferedReader(fr);

    String line = br.readLine();
}
catch(FileNotFoundException e)
{
    e.printStackTrace();
}

catch(IOException e)
{
    e.printStackTrace();
}
```

다른 방법

```java
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class Main {
    public static void main(String[] args) throws IOException {
        FileReader fr = new FileReader("./salmon_bass_data.csv");
        BufferedReader br = new BufferedReader(fr);

        String line = br.readLine();
        System.out.println(line);

        line = br.readLine();
        System.out.println(line);
    }
}
```

Main 함수 자체가
예외를 throws 하도록 변경

문자열 자르기

```
String line = br.readLine();
System.out.println(line);

String[] parse = line.split(",");
System.out.println(parse[0]);
System.out.println(parse[1]);
System.out.println(parse[2]);

line = br.readLine();
parse = line.split(",");
System.out.println(parse[0]);
System.out.println(parse[1]);
System.out.println(parse[2]);
```

Split 함수를 써서 나눈다.
구분자가 무엇인지 문자열로 넘겨준다.

Parse는 String이다.
2, 0.8 이 숫자가 아니다.
아래 코드의 실행 결과는??

```
System.out.println(parse[1]+parse[2]);
```

While 반복문

- **데이터가 몇 줄 일 줄 알고?**
  - 한 줄 한 줄 읽을 것인가?!
  - readline 을 했을 때 null 이 될 때까지 반복하도록 코딩하고 싶다!

```
while ( boolean )
{
    // 실행문
}
```

조건문이 참이면
실행문이 동작한다.

실행문 안에 조건을 거짓으로
만들 수 있는 코드가 있어야 한다.

문자열 10번 찍기

```
while (i < 10)
{
    System.out.println("안녕하세요 " + i);

    ++i;
}
```

```
while (true)
{
    System.out.println("안녕하세요 " + i);
    ++i;

    if (i >= 10)
    {
        break;
    }
}
```

의도적으로 무한 루프를 걸기도 한다..

루프를 break할 수 있는 코드를 넣어주었다.

실습

- while 루프를 사용하여 구구단을 출력해보라
  - 아래의 코드는 2단을 출력하는 샘플 코드
  - 참고로하여 9단까지 출력해보자.

```
int i = 1;
int dan = 2;

while (i < 10)
{
    System.out.printf("%d * %d = %d\n", dan, i, dan * i);
    ++i;
}
```

For / while 변경은 자유롭게…

```
int dan = 2;

while (dan < 10)
{
    for (int i = 1; i < 10; ++i)
    {
        System.out.printf("%d * %d = %d\n", dan, i, dan * i);
    }

    ++dan;
    System.out.println("==============");
}
```

Do / while

- **while 문과 거의 동일하다.**
  - 우선 한 번 실행하고 조건을 본다.

```
do {
    for (int i = 1; i < 10; ++i)
    {
        System.out.printf("%d * %d = %d\n", dan, i, dan * i);
    }

    ++dan;
    System.out.println("==============");
 } while (dan < 10);
```

CSV 파일 읽기

```
FileReader fr = new FileReader("./salmon_bass_data.csv");
BufferedReader br = new BufferedReader(fr);

String line;
do {
    line = br.readLine();
    if (line != null)
    {
        String[] parse = line.split(",");
        System.out.printf("%s %s %s\n", parse[0], parse[1], parse[2]);
    }
} while (line != null);
```

...
Bass 23 7.2
Bass 24 6.8
Bass 24 6.8
Bass 24 6.8
Bass 24 6.8
Bass 24 6.8
Bass 24 6.8

실습

- 몇 개의 데이터를 읽었는지 계수(count) 하라
- csv파일을 열어보면 318개의 record가 있음

정답

- 변수를 하나 쓰면 되는 일이다.
- 루프의 종료 조건을 생각해서 1을 빼야한다.

```
FileReader fr = new FileReader("./salmon_bass_data.csv");
BufferedReader br = new BufferedReader(fr);

String line = br.readLine();
int count = 0;
while (line != null)
{
    String[] parse = line.split(",");
    System.out.printf("%s %s %s\n", parse[0], parse[1], parse[2]);
    line = br.readLine();
    ++count;
}

// 마지막에 빈 문자열 (null)을 읽어야 루프가 종료 되므로 1개를 빼야 한다.
--count;

System.out.println("count = " + count);
```

실습

◆ 길이 데이터를 변수에 저장해 보자.

```
int num_records = 318;
int [] length_arr = new int[num_records];
int count = 0;

String line = br.readLine();
line = br.readLine();

while (line != null)
{
    String[] parse = line.split(",");
    //System.out.printf("%s %s %s\n", parse[0], parse[1], parse[2]);

    length_arr[count] = Integer.parseInt(parse[1]);
    ++count;

    line = br.readLine();
}

for (int i = 0; i < num_records; ++i)
{
    System.out.println(length_arr[i]);
}
```

배열의 크기는 데이터의 수 만큼.

Column 명은 읽어서 버림.

String 을 int로 바꾸었다.

실습

- 앞에서 구한 배열에서 가장 길이가 짧은 물고기의 길이를 구하라.
- 가장 길이가 긴 물고기의 길이를 구하라.
  - 즉, 배열에서 min / max 를 찾아보라.

정답

- min, max 변수의 초기값에 주의

```
int min = 999, max = -1;

for (int i = 0; i < num_records; ++i)
{
    if (length_arr[i] < min)
    {
        min = length_arr[i];
    }

    if (max < length_arr[i])
    {
        max = length_arr[i];
    }
}

System.out.printf("min = %d, max = %d\n", min, max);
```

히스토그램

◆ 길이에 대한 히스토그램을 구해보라

Summary

- **기계 학습이란?**
  - 데이터를 이용하여 일반화된 모델을 만드는 것
  - 데이터, 알고리즘이 필요
- **Domain Knowledge**
  - 해결하고자 하는 문제가 속한 분야의 지식
- **Overfitting**
  - 학습데이터에 과하게 적합된 모델