



# 예제를 통해 살펴보는 머신 러닝 접근법

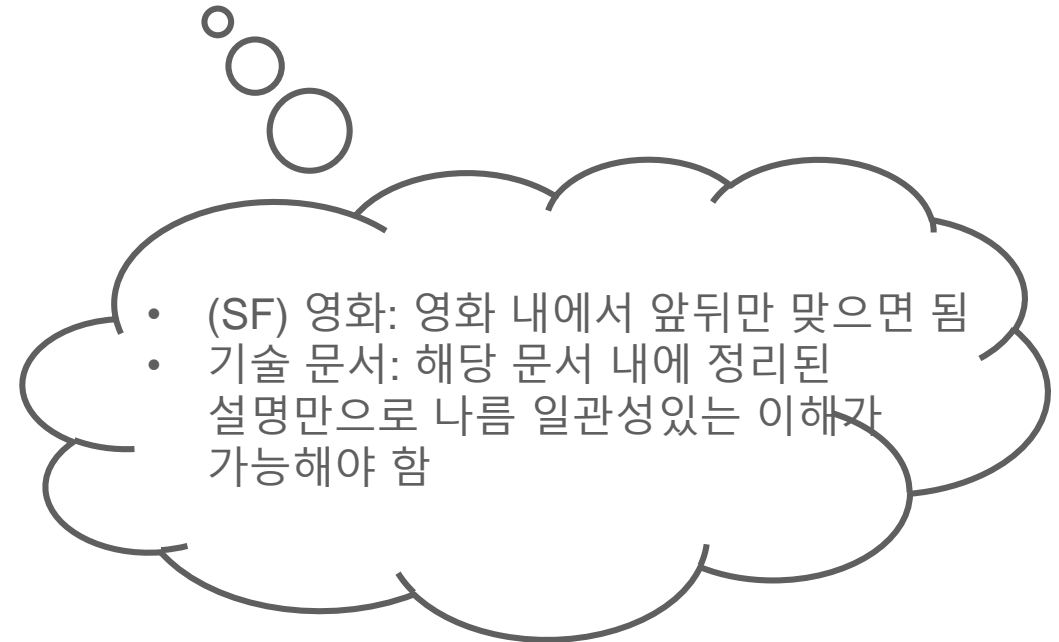
부제: *Machine Learning vs. Running Machine*

Autonomous DB 팀  
이상윤

Aug. 2019

# Safe Harbor Statement

- 본 문서는 절대적이거나 보편적인 규범이 아닌 제 개인적인 관점을 기술했을 뿐입니다.
- 본 문서에서 다루는 기술적 내용은 엄밀성이 떨어지거나 맞지 않는 부분이 있을 수 있습니다. 하지만 제 이해 범위 내에서는 나름의 **내적 일관성**을 갖추고 있는 문서입니다.



# 차례

## Chapter 1. Intro

- 머신 러닝을 대하는 "우리"들의 자화상
- 머신 러닝의 정의
- 머신 러닝의 개요
- 예제 정의

## Chapter 2. Running Machine으로 풀어보는 예제

## Chapter 3. Machine Learning으로 풀어보는 예제 (Minimum Length Description)

- 오컴의 면도날
- Shannon의 Information Theory
- 예제에의 적용
- 오컴의 면도날 원칙에 대한 증명 시도

## Chapter 4. 우리는 머신 러닝에 어떻게 접근해야 하는가

## Chapter 5. 이상윤 상무는 왜 머신 러닝 가이드 완성본을 내놓지 않는가?

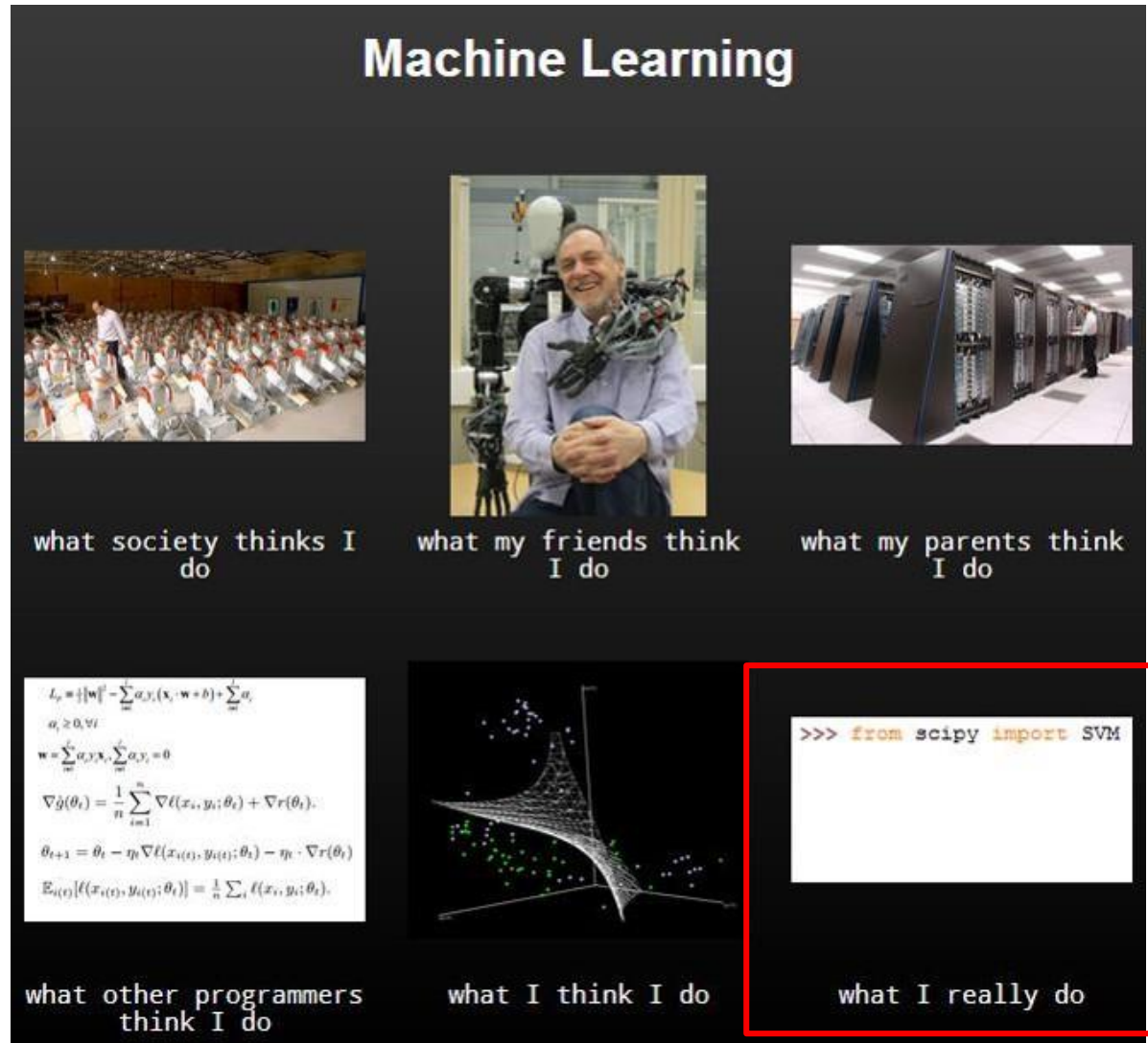
- History & status
- 수학의 분류
- 수학의 분류에 따른 머신 러닝 Algorithm의 분류
- 업무/비즈니스 관점

## Chapter 6. Outro

# Chapter 1. Intro

# 머신 러닝을 대하는 “우리”들의 자화상

**Machine Learning**



what society thinks I do

what my friends think I do

what my parents think I do

what other programmers think I do

what I think I do

what I really do

`>>> from scipy import svm`

⇒ 웃픈 이야기...

# “우리”가 느끼는 감정의 정체

## *Machine Learning vs. Running Machine*

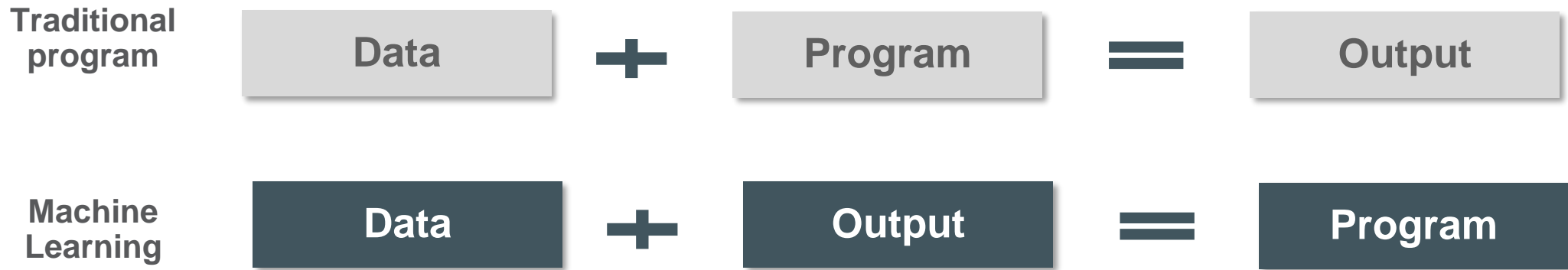
- 사실 내가 주로 하는 일은 제공되는 API들을 불러다 쓰고, 레퍼런스 매뉴얼에 나온 설명을 전달하는 것이다. 누군가 상세한 의미를 물어보면 그것은 제 전문 영역이 아니니다라고 답변을 하면 대개 이해해준다...
- 그런데... 내가 하는 일이 이 정도라면, 그런 일에 소요되는 지적(知的) 에너지의 양이 running machine을 타고 운동할 때 필요한 지적 에너지와 별반 다를 게 없지 않을까? ...
- 나는 가끔 머신 러닝을 하는 사람이 아니라 그냥 머신 자체가 된 것 같다는 느낌이다. 왜 그 알파고가 이세돌을 이길 때에도 정작 알파고 자신은 자기가 바둑을 두고 있다는 사실 자체를 몰랐다고 하지 않던가?...

**단, 모두가 같은 걸 느끼는 것은 아님**



# 머신 러닝의 정의

## 흔히 보는 정의 한가지



하지만 위 표현은 정의라기 보다는 일종의 추상적인 비유일 뿐  
• 게다가 머신 러닝의 실체를 과장하고 호도할 가능성도 있음

# 머신 러닝의 정의

좀 더 딱딱한, 하지만 구체적인 정의

## 머신러닝 = 응용 수학

- 목표: (숨겨진) 패턴의 인식
- 도구들
  - 확률/통계적 접근 또는 기하학적 (linear algebra) 접근
  - 기반 수학: 대수(algebra), 미적분 (as advanced algebra)
  - 추가 옵션: numerical analysis, etc.
- 요구 수준
  - 최소한 이공계(非수학과) 학부 레벨
  - “우리”들의 수준은?

## 프로그래밍/툴은?

- 당연히 관련 스킬 필요. **하지만 main은 어디까지나 수학**
- 참고: Why Python?
  - (Higher level language)
  - Interactive (Notebook)
  - General purpose
    - Unlike MATLAB/octave, SAS/R
  - 가장 넓은 community
    - Unlike SQL, PL/SQL

**최대의 난관: 수학을 익혀야 하고, 또 수학은 따로 익혀야 함**



# 머신 러닝의 개요

## Tablet Data

	Attr.	Attr.	....	Attr.	....	Attr.
Case						
Case						
⋮						

## 머신 러닝의 목표

- “Predictive Analytics”
- 예측하고자 하는 target 속성과 나머지 속성들 (predictors) 사이에 숨어있는 수학적 패턴들을 발견/이용
- Supervised learning이 기본

## 머신 러닝 function의 분류

- Numerical 데이터에 대한 예측은 regression, categorical 데이터에 대한 예측은 classification
  - 이때 regression과 classification은 마치 동전의 양면같은 관계를 가짐
    - Regression을 통해 예측하고자 하는 숫자가 확률값이라면 그건 곧 classification
    - 따라서 같은 algorithm이 regression과 classification의 두 function에 공통적으로 사용될 수 있음
- “정답”에 해당하는 target 값이 주어지지 않는 형태가 unsupervised learning
  - 하지만 대부분의 경우 unsupervised learning은 supervised learning을 위한 전처리 작업 정도...

# 예제 정의

소스: Oracle 매뉴얼

- Goal
  - Case ID: CUST\_ID
  - Target: HOME\_THEATER\_PACKAGE (binary classification)
- 문제
  - 그런데 이때 모든 predictor들의 “predicting power”는 동일할까?
  - 만일 그렇지 않다면 전체 속성이 아닌 일부 “적절한” 속성들만을 골라 predictor로 사용하는 것이 여러모로 바람직하지 않을까?
    - 예
      - 속성들이 너무 많은 경우
      - OAC에서 처음 분석(aggregation)을 시작해야 할 때



## SH.SUPPLEMENTARY\_DEMOGRAPHICS

❖ COLUMN_NAME	❖ DATA_TYPE
CUST_ID	NUMBER
EDUCATION	VARCHAR2 (21 BYTE)
OCCUPATION	VARCHAR2 (21 BYTE)
HOUSEHOLD_SIZE	VARCHAR2 (21 BYTE)
YRS_RESIDENCE	NUMBER
AFFINITY_CARD	NUMBER (10,0)
BULK_PACK_DISKETTES	NUMBER (10,0)
FLAT_PANEL_MONITOR	NUMBER (10,0)
HOME_THEATER_PACKAGE	NUMBER (10,0)
BOOKKEEPING_APPLICATION	NUMBER (10,0)
PRINTER_SUPPLIES	NUMBER (10,0)
Y_BOX_GAMES	NUMBER (10,0)
OS_DOC_SET_KANJI	NUMBER (10,0)
COMMENTS	VARCHAR2 (4000 BYTE)

그러면 어떻게 고르면 잘 고를 수 있을까?

다시 말해 어떤 속성들이 다른 속성들에 비해 target을 상대적으로 잘 “설명”할까?

# Chapter 2. Running Machine으로 풀어보는 예제

# 간단하게 풀어낼 수 있음!

## In Oracle ML

```
BEGIN
  DBMS_PREDICTIVE_ANALYTICS.EXPLAIN(
    data_table_name      => 'SUPPLEMENTARY_DEMOGRAPHICS',
    explain_column_name  => 'HOME_THEATER_PACKAGE',
    result_table_name    => 'DEMOGRAPHICS_EXPLAIN_RESULT',
    data_schema_name     => 'SH');
END;
/
```

```
SELECT ATTRIBUTE_NAME, EXPLANATORY_VALUE, RANK
FROM demographics_explain_result;
```

ATTRIBUTE_NAME	EXPLANATORY_VALUE	RANK
Y_BOX_GAMES	.524	1
YRS_RESIDENCE	.503	2
HOUSEHOLD_SIZE	.147	3
AFFINITY_CARD	.060	4
EDUCATION	.033	5
COMMENTS	.025	6
OCCUPATION	.022	7
FLAT_PANEL_MONITOR	.000	8
OS_DOC_SET_KANJI	.000	9
CUST_ID	.000	9
BULK_PACK_DISKETTES	.000	9
BOOKKEEPING_APPLICATION	.000	9
PRINTER_SUPPLIES	.000	9

13 rows selected.

## In OAC

The screenshot shows the OAC 'Data' pane with a list of attributes. A context menu is open over the 'HOME\_THEATER\_PACKAGE' attribute, offering options to create visualizations or filters. The 'Explain HOME\_THEATER\_PACKAGE' option is highlighted.

Click here or drag data to add a filter

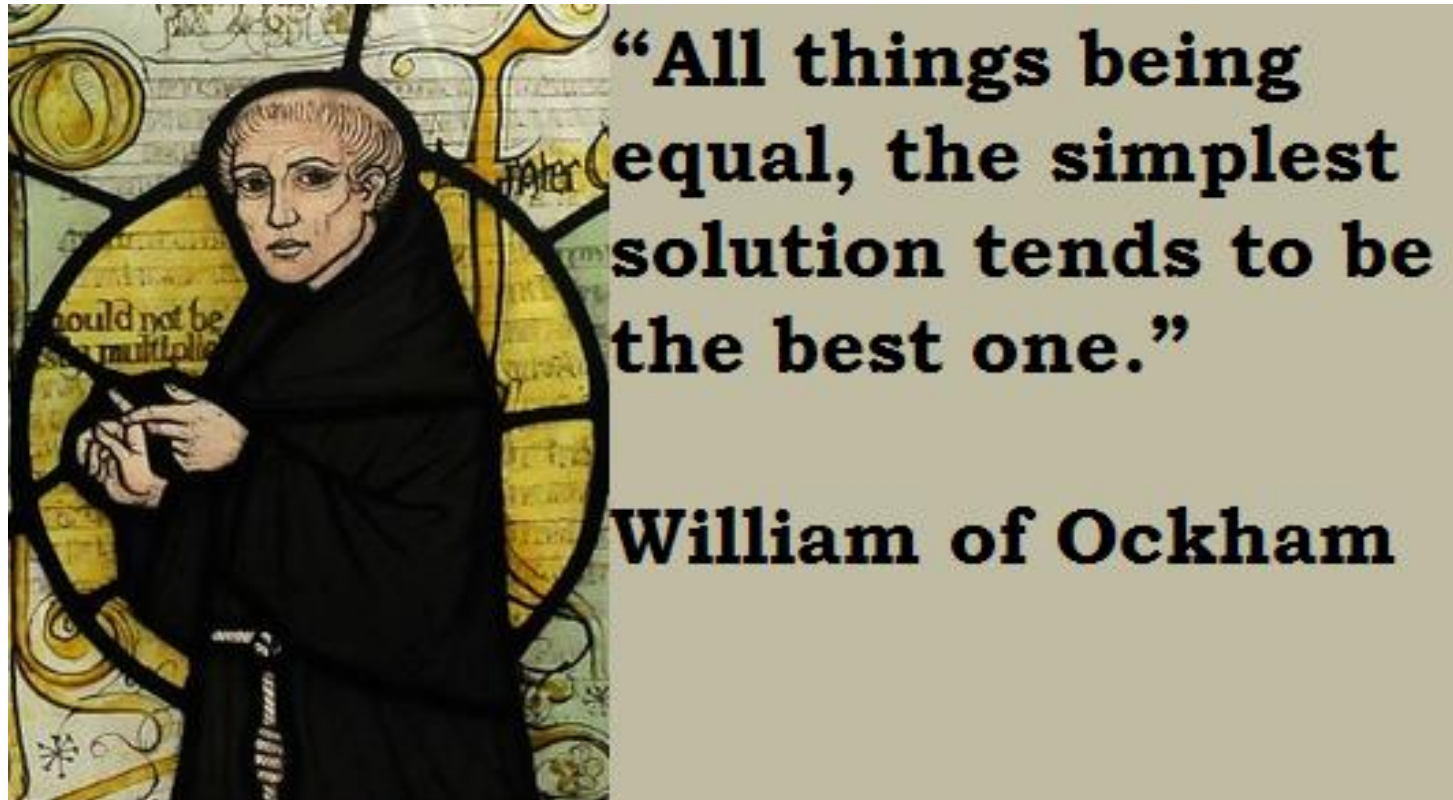
Create Best Visualization  
Pick Visualization...  
Create Filter  
Explain HOME\_THEATER\_PACKAGE

Select Visualization to View Details

# That's it!

# Chapter 3. Machine Learning으로 풀어보는 예제 (Minimum Length Description)

# Ockham's Razor



- Razor: 어떠한 현상이나 원리를 나타내기 위한 논리 구조에서 쓸모없는 비약, 전제, 논거들을 **잘라내라!**
- 각 predictor 별로 target을 “설명”해보고 그중 가장 짧은 설명을 제공하는 predictor를 선택하자!
- 원래는 경험칙 또는 논리적/철학적인 명제. 하지만 **Shannon의 Information Theory**에 의해 과학의 영역으로 들어옴

# 정보량의 개념

## *Tentative Definition*

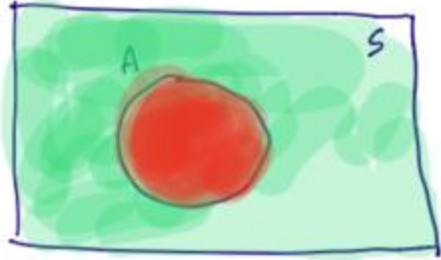
- 예: 주식 투자를 위한 정보 두 가지
  1. 내일은 해가 뜬다.
  2. 내일 미국방부 클라우드 프로젝트가 오라클에 의해 전량 수주될 것이다.
- 누구라도 1번은 버리고 2번을 선택
  - 1번에는 (쓸만한) 정보라고는 전혀 담겨있지 않지만 2번은 (중요한) 정보를 담고 있다.
    - 1번의 정보량  $\ll$  2번의 정보량
  - 수학적 해석: 2번 사건의 발생 확률이 1번 사건에 비해 매우 낮다
    - 정보량의 첫번째 정의:
      - 어떤 사건이 담고 있는 정보량 = 해당 사건의 발생 확률의 역수
        - “정보량은 놀라움에 비례한다”
        - “드문 사건일 수록 정보량이 크다”

$$\frac{1}{P}$$

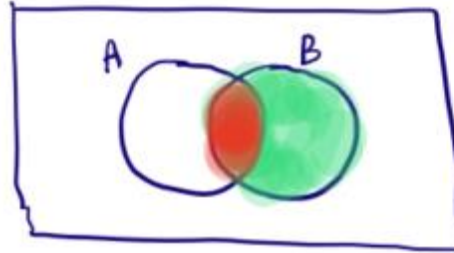


# 여기서 잠깐! 꼭 필요한 수학 설명

확률, 조건부 확률, 독립 사건



$$P(A) = \frac{m(A)}{m(S)}$$



$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(A|B) = P(A)$$

$$\frac{P(A \cap B)}{P(B)} = P(A)$$

$$P(A \cap B) = P(A)P(B)$$

Log의 기본

Logarithmic form

Exponent

$$\log_a x = y$$

Base

Exponential form

Exponent

$$a^y = x$$

Base

$$\log_b AB = \log_b A + \log_b B \quad \text{Log Property 1}$$

$$\log_b \frac{1}{A} = -\log_b A \quad \text{Log Property 2}$$

# 정보량의 정의를 수정해보자

- 정보량을 단순히 발생 확률의 역수로만 정의하면 모순 발생
  - 상호 독립적인 두 사건 A, B가 각각 p, q의 확률로 발생. 이 두 사건이 동시에 발생한 사건에 담긴 정보량은?

$$\text{정의: } \frac{1}{pq} \quad \text{적용: } \frac{1}{p} + \frac{1}{q}$$

$\neq$

- Log to the rescue! 정보량의 두번째 정의:
  - 어떤 사건이 담고 있는 정보량 = 해당 사건의 발생 확률의 역수에 log를 취한 값

$$\log \frac{1}{p} \quad \log \frac{1}{pq} = \log \frac{1}{p} + \log \frac{1}{q}$$

# 평균 정보량의 정의

- 일반적으로 우리는 단일 사건이 아닌 여러 개의 사건으로 구성된 어떤 종합적인 상황을 다룬다. 이때 그 상황이 갖는 정보량은 그 상황을 구성할 수 있는 모든 가능한 사건들의 평균적인 정보량으로 나타내야 한다.

$$\sum_n p_i \log \frac{1}{p_i}$$

- Shannon은 위에서 정의한 양을 **정보 엔트로피**라고 명명
  - 왜 그냥 평균 정보량이라는 평이한 용어 대신 엔트로피라는 물리학 용어를 선택했을까?...

$$H = k_B \log \Omega \quad \dots$$

양자역학

...

정보 우주론 / Simulation 우주론

## More Concrete Example

*어떤 정보를 bit 스트림으로 표현하면 정보량이란 말 그대로 길이로 계량할 수 있다*

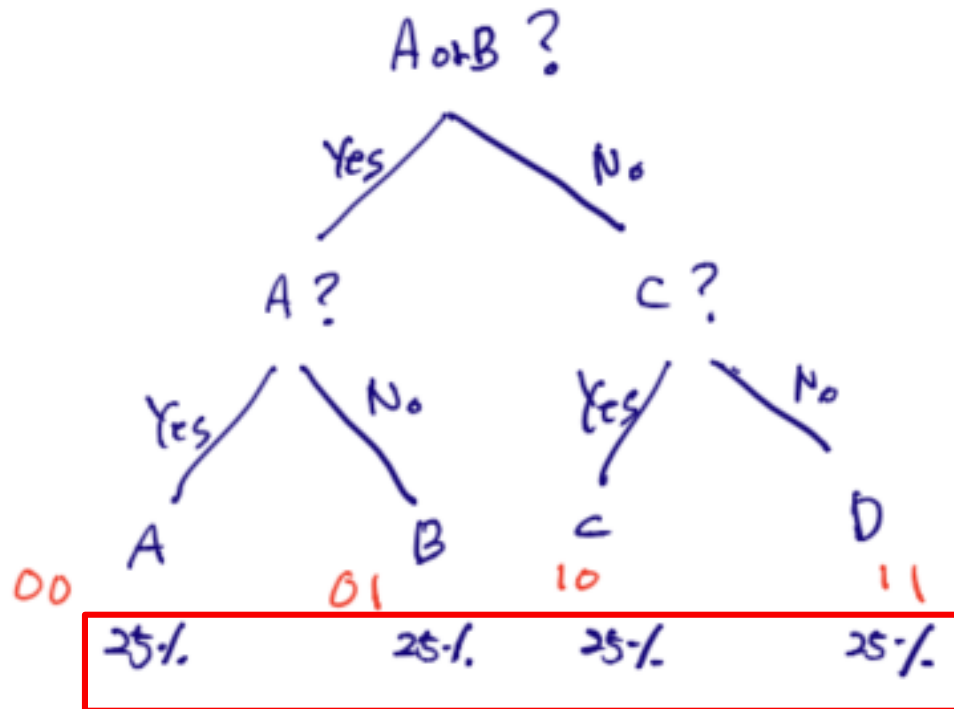
- A, B, C, D 네개의 symbol로 이루어진 메시지를 생성하는 기계 M1이 있다고 가정. 그 메시지를 bit 스트림으로 표현한다면?

00	A
01	B
10	C
11	D

⇒ 2 bit per symbol

# Shannon의 아이디어

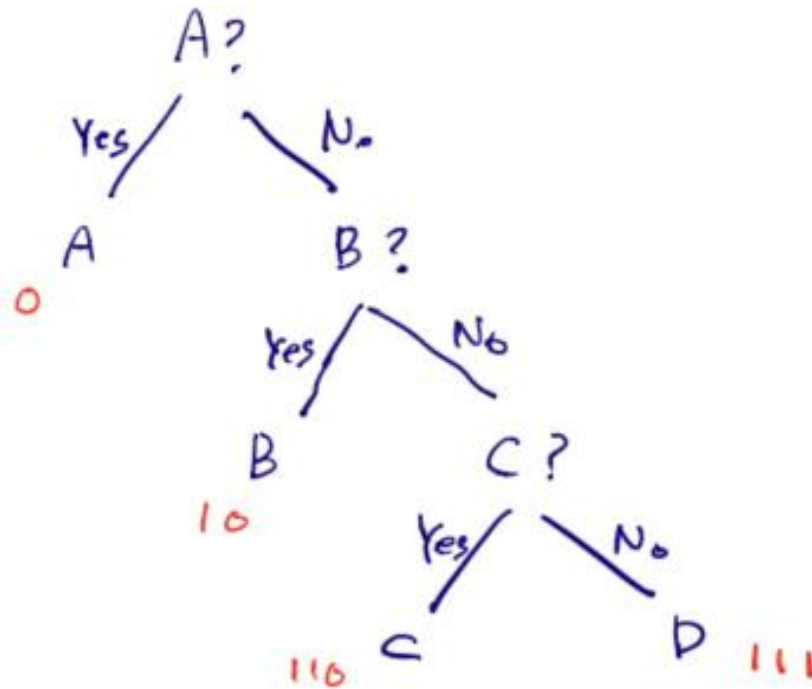
- 메시지를 이루는 각 symbol은 일련의 yes/no 질문을 통해 파악할 수 있다.
- 그리고 필요한 yes/no 질문의 갯수가 바로 해당 symbol을 표현하기 위해 필요한 bit 수이다.



$$\text{bit length} = \text{ceil} \left( \log_2 \frac{1}{P} \right)$$

# 앞서 정리한 정보량 개념과의 관계

- 발생 확률이 다른 symbol들은 해당 bit 수도 달라져야 한다.
  - $P(A) : P(B) : P(C) : P(D) = 50\% : 25\% : 12.5\% : 12.5\%$  인 기계 M2의 경우



# 평균 정보량의 비교

- M1과 M2는 평균 정보량이 다르다.

두 메시지의 entropy

$$M1: \sum_n P_i \log_2 \frac{1}{P_i} = \left( 0.25 \times \log_2 \frac{1}{0.25} \right) \times 4 = 2$$

$$\begin{aligned} M2: \sum_n P_i \log_2 \frac{1}{P_i} &= 0.5 \times \log_2 \frac{1}{0.5} \\ &+ 0.25 \times \log_2 \frac{1}{0.25} \\ &+ 0.125 \times \log_2 \frac{1}{0.125} \times 2 = 1.75 \end{aligned}$$



## 마지막 예

- 왜 “내일 해가 뜬다”는 statement에는 정보가 전혀 담겨있지 않은가?



# 예제에의 적용

앞서 예를 든 메시지 생성 기계와 동일한 원리:

- 기계 M1, M2, ..., Mn이 있고, 각 기계는 하나의 predictor의 값들만으로 구성된 메시지를 생성한다. 이 메시지를 통해 target을 예측하는 것.
- 이때 각 기계의 평균 정보량, 즉 정보 엔트로피를 비교하여 작은 것을 고르면 된다.
- 이때
  - Target 값 자체가 아닌 predictor 값들로 메시지가 생성되므로, 확률은 target의 predictor에 대한 조건부 확률로 기술되어야 한다.
  - Predictor 자체에 대한 정보량도 포함이 되어야 한다.

$$\sum_n P(D|h) \log_2 \frac{1}{P(D|h)} + \sum_n P(h) \log_2 \frac{1}{P(h)}$$

D = Data (target)

h = hypothesis (attribute)



Y\_BOX\_GAMES is  
more relevant than  
AFFINITY\_CARD

(계산 편의 상 두 개 컬럼만 비교함)

**마지막 질문: 그런데 오컴의 면도날은 정말 믿고 쓸 수 있는 원칙인가?**

# 여기서 잠깐! 꼭 필요한 수학 설명

## Total Probability Theorem

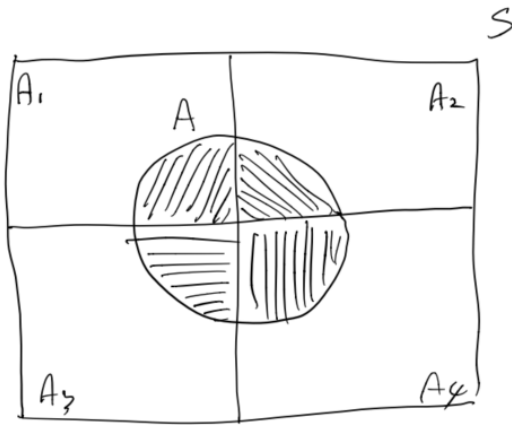
Let  $\{A_1, A_2, \dots, A_n\}$  be partitions of  $S$ .

$$\Leftrightarrow \begin{cases} \circ A_i \cap A_j = \emptyset & \text{for } i \neq j \\ \circ \bigcup_{i=1}^n A_i = S \end{cases}$$

Proposition 1.1  $\{A_1, A_2, \dots, A_n\}$  = partitions of  $S$ ,  $P(A_i) > 0$

$$\Rightarrow P(A) = \sum_{i=1}^n P(A_i) P(A|A_i)$$

proof) ex)



$$\Rightarrow \begin{cases} (A_i \cap A) \cap (A_j \cap A) = \emptyset \\ \text{for } i \neq j \end{cases}$$
$$\bigcup_{i=1}^n (A_i \cap A) = A$$

$$\Rightarrow P(A) = P(A_1 \cap A) + P(A_2 \cap A) + \dots + P(A_n \cap A)$$

$$\text{or say, } P(A_i \cap A) = P(A \cap A_i), \quad \frac{P(A \cap A_i)}{P(A_i)} = P(A|A_i)$$

$$\Rightarrow P(A_i \cap A) = P(A_i) P(A|A_i)$$

$$\therefore P(A) = P(A_1) P(A|A_1) + \dots + P(A_n) P(A|A_n)$$

# An Intuition of Ockham's Razor

Target에 대한 예측이 Total Probability Theorem으로 기술될 수 있는 가상의 sample space를 생각해보자. 그리고  $h$ 를 편의상 그냥 단일 event인 것처럼 취급해보자.

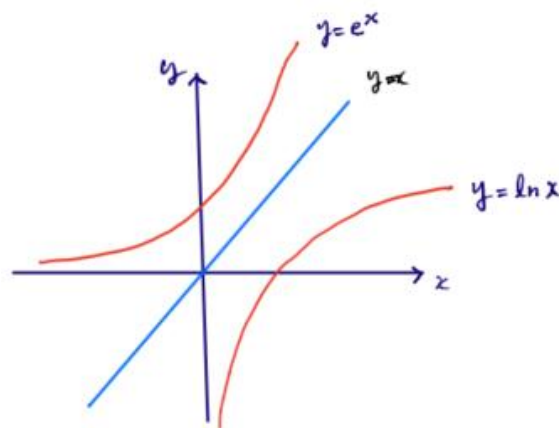
$$P(D) = P(h_1)P(D|h_1) + P(h_2)P(D|h_2) + \dots + P(h_n)P(D|h_n)$$

Target을 가장 잘 예측하는 predictor를 찾는 것은 다음 식의 solution을 구하는 것과 같다.

$$\operatorname{argmax}_h \{P(h)P(D|h)\}$$

Log 함수는 monotonously increasing 함수이므로 위 식은 아래 식과 동일하다.

$$\operatorname{argmax}_h \{\log P(h)P(D|h)\}$$



# An Intuition of Ockham's Razor

Log Property 1을 적용하면 다음 식으로 변형된다.

$$\operatorname{argmax}_h \{ \log P(h) + \log P(D|h) \}$$

부호를 바꾸어주면 argmax가 argmin으로 바뀐다.

$$\operatorname{argmin}_h \{ -\log P(h) - \log P(D|h) \}$$

Log Property 2를 적용하면 다음 식으로 변형된다.

$$\operatorname{argmin}_h \left\{ \log \frac{1}{P(h)} + \log \frac{1}{P(D|h)} \right\} \iff \sum_n P(D|h) \log_2 \frac{1}{P(D|h)} + \sum_n P(h) \log_2 \frac{1}{P(h)}$$

위 식은 결국 “정보량이 작은, 다시 말해 좀 더 간단한 설명을 찾으라!”라는 말과 동의어이다!

# Chapter 4. 우리는 머신 러닝에 어떻게 접근해야 하는가

# Machine Learning vs. Running Machine

- For IC

- Machine Learning은 어렵지만 Running Machine은 그리 어렵지 않습니다. 따라서 미리 주눅들거나 너무 큰 부담을 갖지 않아도 됩니다.
  - 다만 고객들에게는 조심스럽게 다가가야 할 것입니다.
- 혹시라도 마음 깊은 곳에 더 나아가고 싶은 열망이 있다면 그때부터 차근차근 정진하시면 됩니다.
  - One small tip: 수학도 영어로 공부하세요.

- For M

- 어떤 기술들은 쉽고 빠르게 얻는 것이 불가능하다는 점을 잊지 말아 주십시오.
  - Machine Learning을 하고자 하는 IC가 있다면 서둘러 평가를 내리기보다는 인내심을 갖고 지원해 주시길 부탁드립니다.
- 모든 구성원이 분석과 머신러닝을 지향하는 조직은 가능하지도 않을 것이고 심지어는 바람직하지도 않을 것입니다.
  - One small tip: 강점 혁명을 기억해 주세요.



# Chapter 5. 이상윤 상무는 왜 머신러닝 가이드 완성본을 내놓지 않는가?

# History & Current Status

- 올해 1월 version 0.37 발표
  - 공약 아닌 공약: scope는 7개의 function을 default algorithm으로 구사하는 내용
- 3월 경 공약의 절반을 지킴
  - 정해둔 scope의 내용을 **가까스로** “1회독”
  - 하지만 문서 정리로 막막해 하던 중에 어떤 SE의 feedback을 받음
    - “그렇게 수학적으로 접근하는 방법을 쓰면 다른 사람들이 지레 머신 러닝이 어렵다고 생각해서 더 안 하게 될 위험이 있습니다.”
    - 위 feedback에 일리가 없는 건 아니라고 생각하여 문서 정리는 사실상 포기
- 대신에 나름 체계적인 (수학) 스터디 시작
  - 8월 현재에도 아직 갈 길이 멀
  - **그리고 오늘 완성본 아닌 완성본을 발표하며 방점을 찍는 중**

Function	Type	Default Algorithms
Attribute Importance	Supervised	Minimum Description Length
Classification	Supervised	Naive Bayes
Regression	Supervised	Support Vector Regression
Clustering	Unsupervised	k-Means++
Anomaly Detection	Unsupervised	One-Class Support Vector Machine
Association Rules	Unsupervised	Apriori
Feature Extraction	Unsupervised	Non-Negative Matrix Factorization

# 수학의 분류

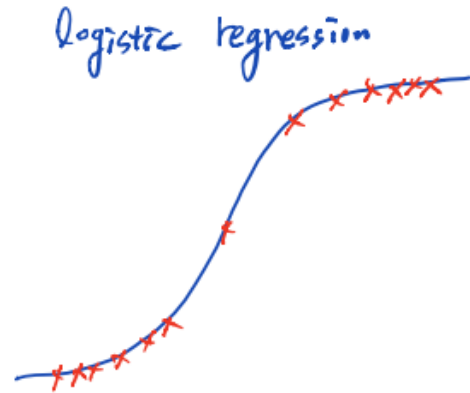
소스: “틀리지 않는 법”

심오함	저자가 다루고자 하는 내용	전업 수학자의 영역: - 페르마의 정리, - 푸앵카레 추측, - 리만 가설...
얕음	Arithmetic, Algebra, Trigonometry, etc.	Calculus
	단순함	복잡함

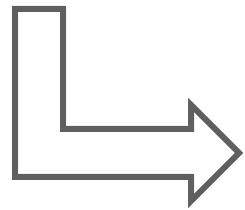
# 수학의 분류에 따른 머신 러닝 Algorithm의 분류

심오함	Minimum Length Description Naïve Bayes	
얕음	Apriori	Support Vector Regression k-Means++ One-Class Support Vector Machine Non-Negative Matrix Factorization ... 기타 나머지 전부
	단순함	복잡함

# 얼핏 간단해 보이는 것도 충분히 복잡!



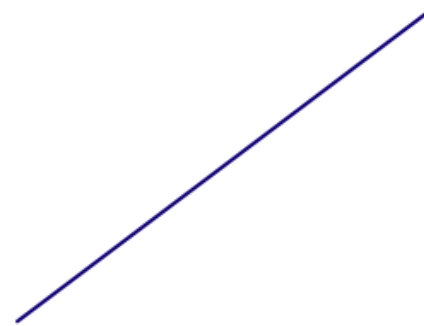
$$N(t) = \frac{N_0 k}{(k - N_0) e^{-rt} + N_0}$$



Easy?

No. “내적 완결성”을  
추구하는 문서라면...

linear regression



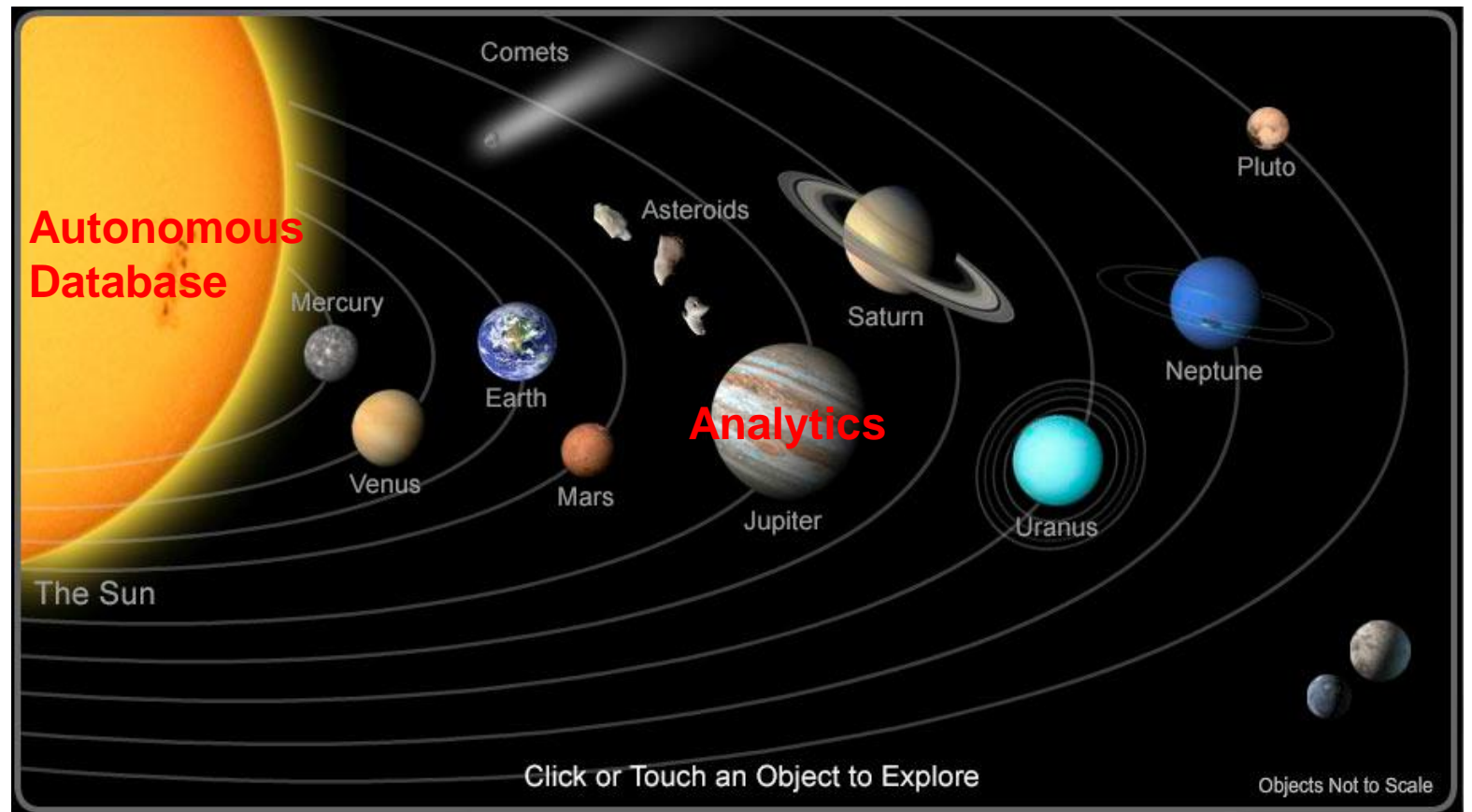
$$y = mx + b$$

slope      y-intercept

# 업무/비즈니스 관점

제가 보는 업무: Autonomous DB 팀 일

제가 생각하는 우리 비즈니스:



# Chapter 6. Outro



# Ockham's Razor Revisit

왜 이상윤 상무는 머신 러닝 가이드 완성본을 내놓지 못하는가?

## One Explanation

하게 될 위험이 있습니다.”

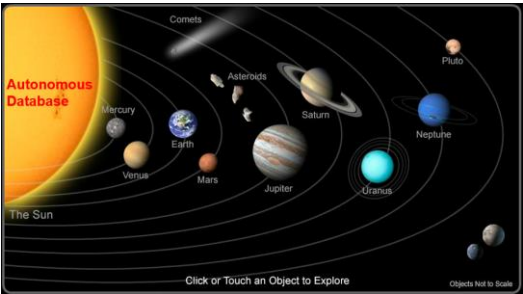
- 위 feedback에 일리가 없는 건 아니라고 생각하여 문서 정리는 사실상 포기

-트 체계적이 /스하\ 스택의 시작

심오함	Minimum Length Description Naïve Bayes	
알음	Apriori	Support Vector Regression k-Means++ One-Class Support Vector Machine Non-Negative Matrix Factorization ... 기타 scope 외의 나머지 전부
	단순함	복잡함

내적 완결성

업무 부하...



## Another Explanation

역량 부족

ORACLE®