

EZgoingAI

<https://www.youtube.com/watch?v=MShERLhFg70&t=0s>

이현봉

인공지능 얘기를 느긋하게 풀어나가려 합니다.

분량이 얼마나 될 지, 일정이 어떻게 될 지 정확히는 나도 모릅니다.

두 테마를 생각하는데요...

첫째 테마는 '**실용**'으로 세 과정으로 되었는데요,

1. '데이터사이언스'의 주요 내용인 **예측분석**을 다룹니다.
 - "**An Introduction to Statistical Learning**" 을 교재로 사용합니다.
 - '데이터사이언스'는 머신러닝을 그대로 활용할 수 있는 분야입니다
 - 저자들의 목표와 같이, 이공계열을 나오지 않은 사람이 공부해 (즉, 수학을 거의 쓰지 않고) 실무에 적용할 수 있게 내용을 다룹니다.
 - Python (Scikit-learn, Numpy, Pandas, Matplotlib) 으로 실습합니다.
 - 교재에 없거나 간단히 다루어진 자동화/Pipelining, Hyperparameter tuning, Ensembling/Stacking, Regularization, Gradient Descent, 배포를 다룹니다.
 - Stochastic Process(Signal/Time series), 신경망/딥러닝은 다루지 않습니다
 - <https://github.com/hyunblee/ISLR-with-Python> 내용을 중심으로, 더 천천히, 넓고 깊게 실습합니다.

인공지능,
머신러닝,
빅데이터,
데이터 사이언스

인공지능



Building intelligent machine

- [Reasoning, problem solving](#)
- [Knowledge representation](#)
- [Planning](#)
- [Learning](#)
- [Natural language processing](#)
- [Perception](#)
- [Motion and manipulation](#)
- [Social intelligence](#)
- [Creativity](#)
- [General intelligence](#)

- Wiki

이 정도는 되고 있죠

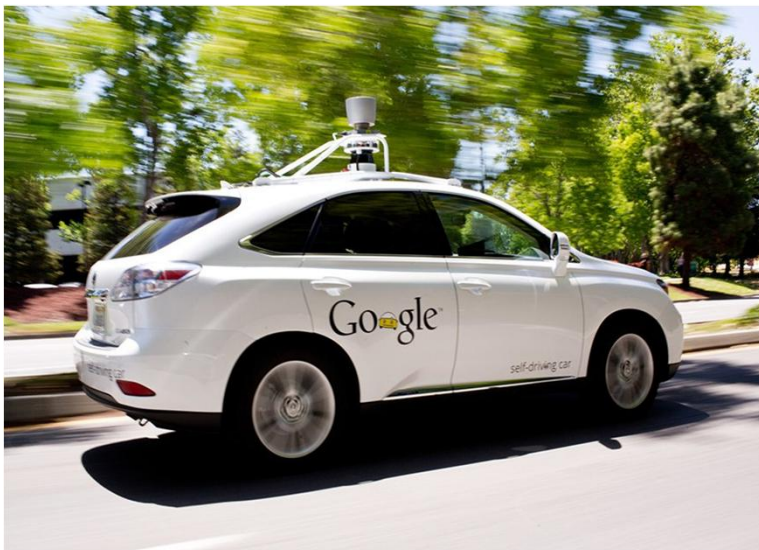


<https://ko.wikipedia.org/wiki/인공지능>
: 인공지능은? (위키)

<http://ai.stanford.edu/~nilsson/QAI/qai.pdf>
: 인공지능의 발자취

[Steps Toward Artificial Intelligence \(M. Minsky\)](#)
: Search, Pattern-Recognition, Learning, Planning, and Induction

<http://aima.cs.berkeley.edu/index.html>
: Artificial Intelligence: A Modern Approach



Machine Learning (기계학습)

인공 지능의 한 분야로, 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야를 말한다

(https://ko.wikipedia.org/wiki/기계_학습)

“A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .”

(Build computer agents that automatically improve with experience)

- Tom Mitchell (1997, Machine Learning)

Machine Learning – Supervised Learning

Example training set



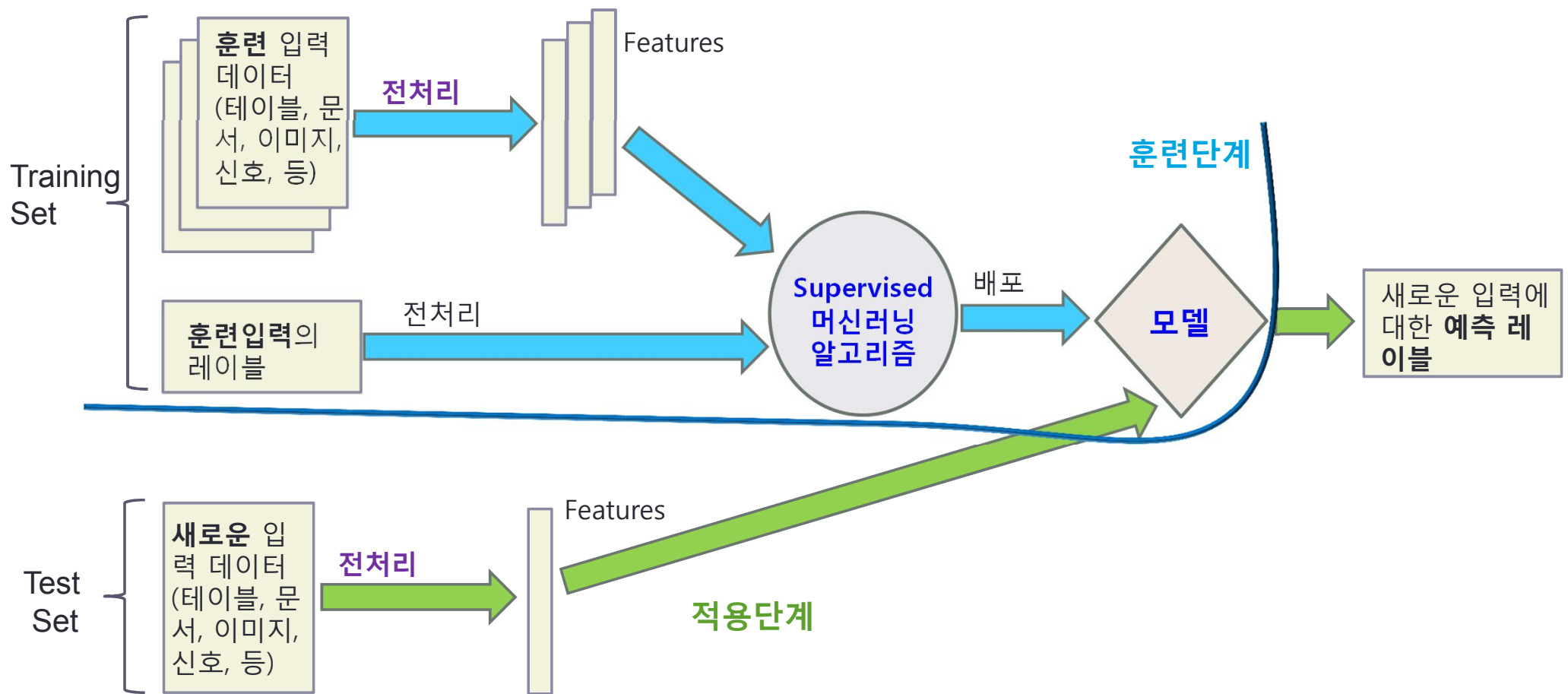
<http://cs231n.github.io/classification/>, Lecture 2-16

테스트 샘플



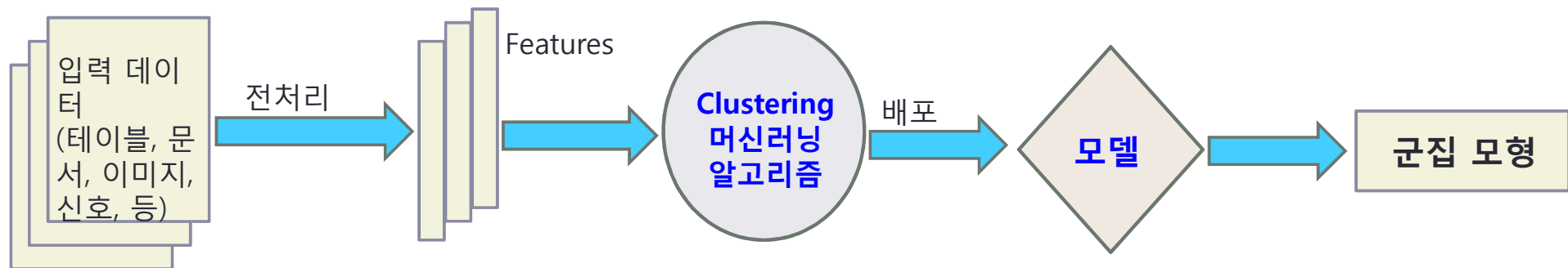
Machine Learning – Supervised Learning

레이블(원하는 결과)과 질문(입력 데이터)이 함께 있는 훈련(연습) 데이터를 이용해 머신러닝 알고리즘이 입력이 주어지면 레이블 답을 내는 메커니즘(함수)을 학습시키는 과정

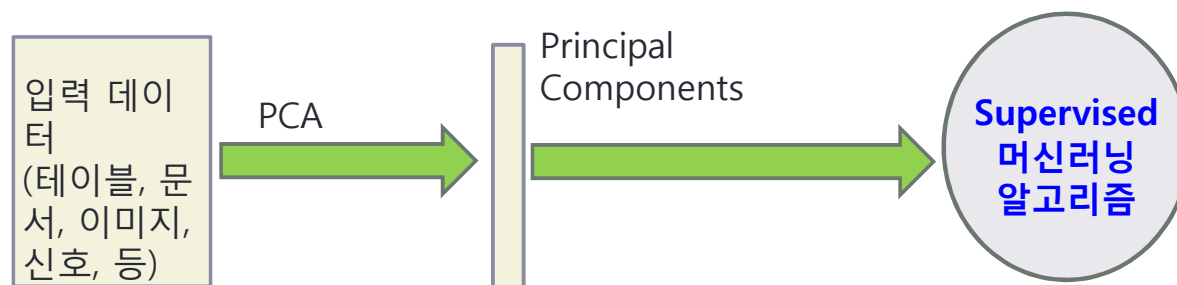


Machine Learning – Unsupervised Learning

레이블이 없는 데이터에서 데이터가 지니고 있는 특성을 분석해 군집화를 하거나 또는 요약(summarize)하는 방법 등...



- 입력을 PCA로 전처리하여 Supervised Learning에 적용



Machine Learning – Reinforcement Learning

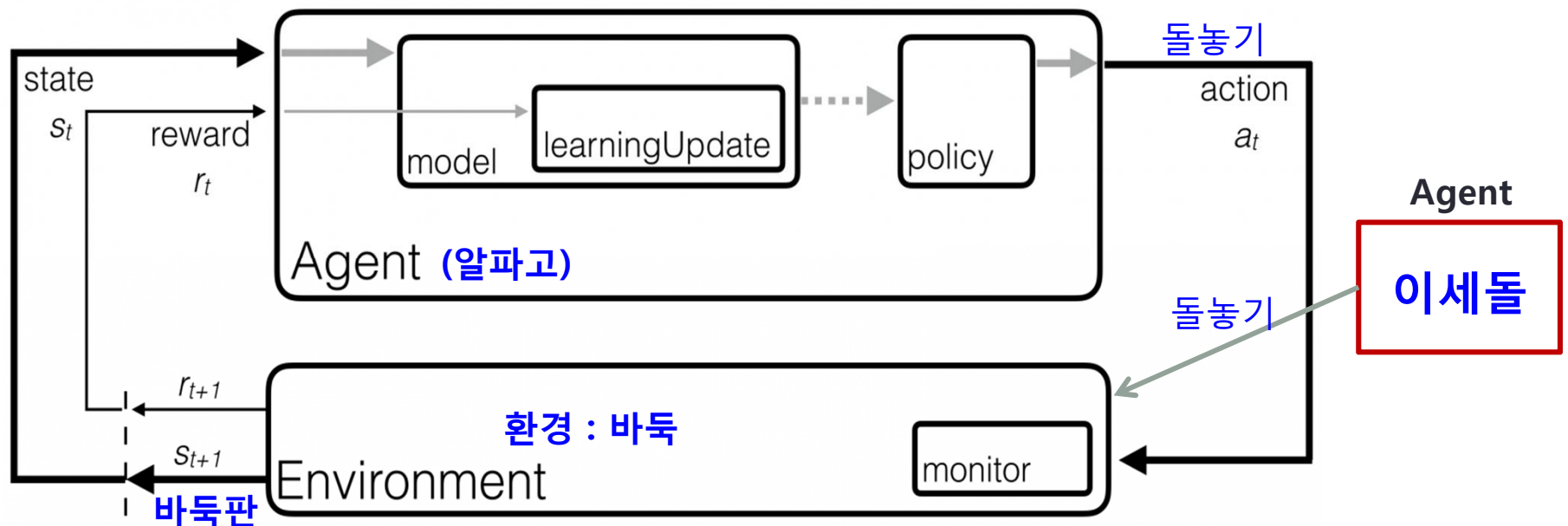
환경내에 있는 에이전트가 어떤 행동을 해야 미래에 최고로 보상(reward)을 받는가를 학습

자연계의 생물이 늘 하는 것. (어떻게 해야 나에게 유리할까?)

지금 내가 하는 행동이 향후 어떤 보상(손해)를 가져올까? 그 것을 언제쯤 알 수 있을까? 그 신호가 정확할까? 지금까지, 또 앞으로 있을 많은 행동중에 과연 어떤 것이 영향을 더 끼쳤을까?

알파고: 내가 어떻게 해야 이길 수 있을까? 내가 몰리고 있는데 나의 어떤 수가 문제였을까?

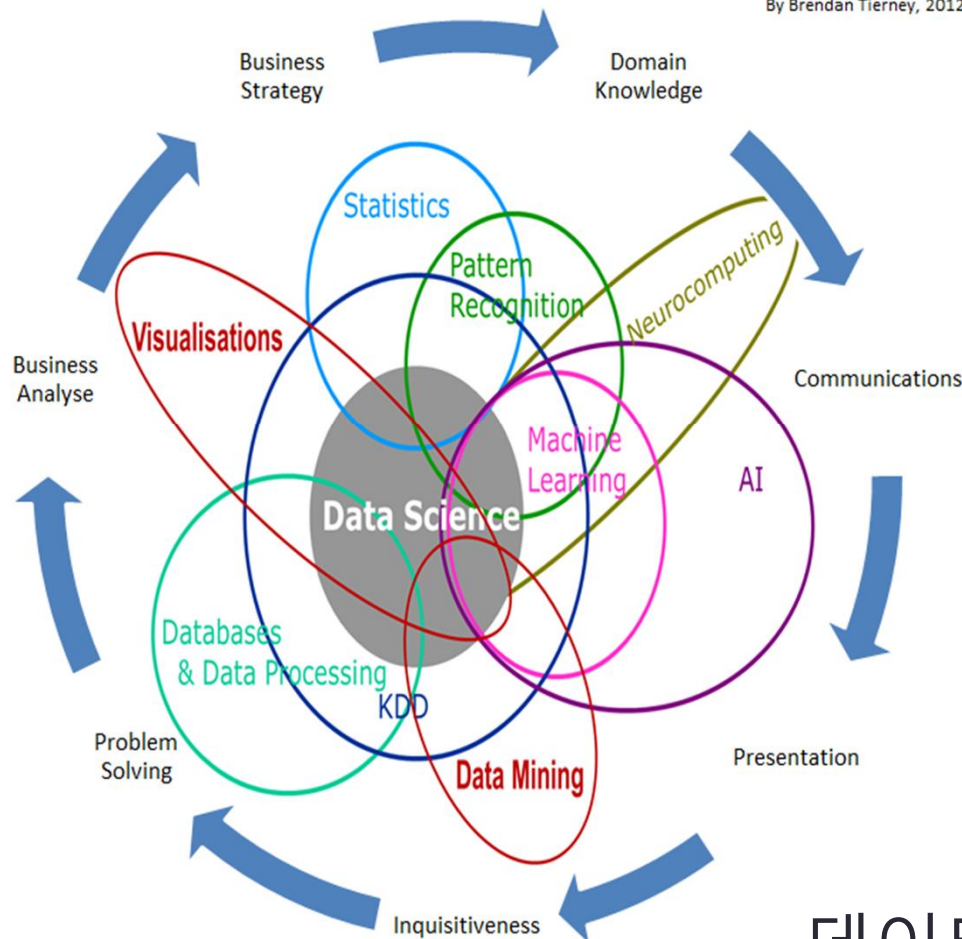
* 녹음에서 'reward'를 설명할 때 'policy' 부분을 섞는 오류를 했습니다. 감안해 들으세요.



데이터 사이언스

Data Science Is Multidisciplinary

By Brendan Tierney, 2012



데이터사이언스 – sexy rebranding
of statistics?

예측분석 (Predictive Analytics)

Predictive analytics encompasses a variety of statistical techniques from modeling, machine learning, and data mining that analyze current and historical facts to make **predictions** about future, or otherwise unknown, events.

- 데이터 사이언티스트가 많이 하는 일



머신러닝, 데이터사이언스, 예측분석, 그 말이 그 말 같고 헷갈리는데...

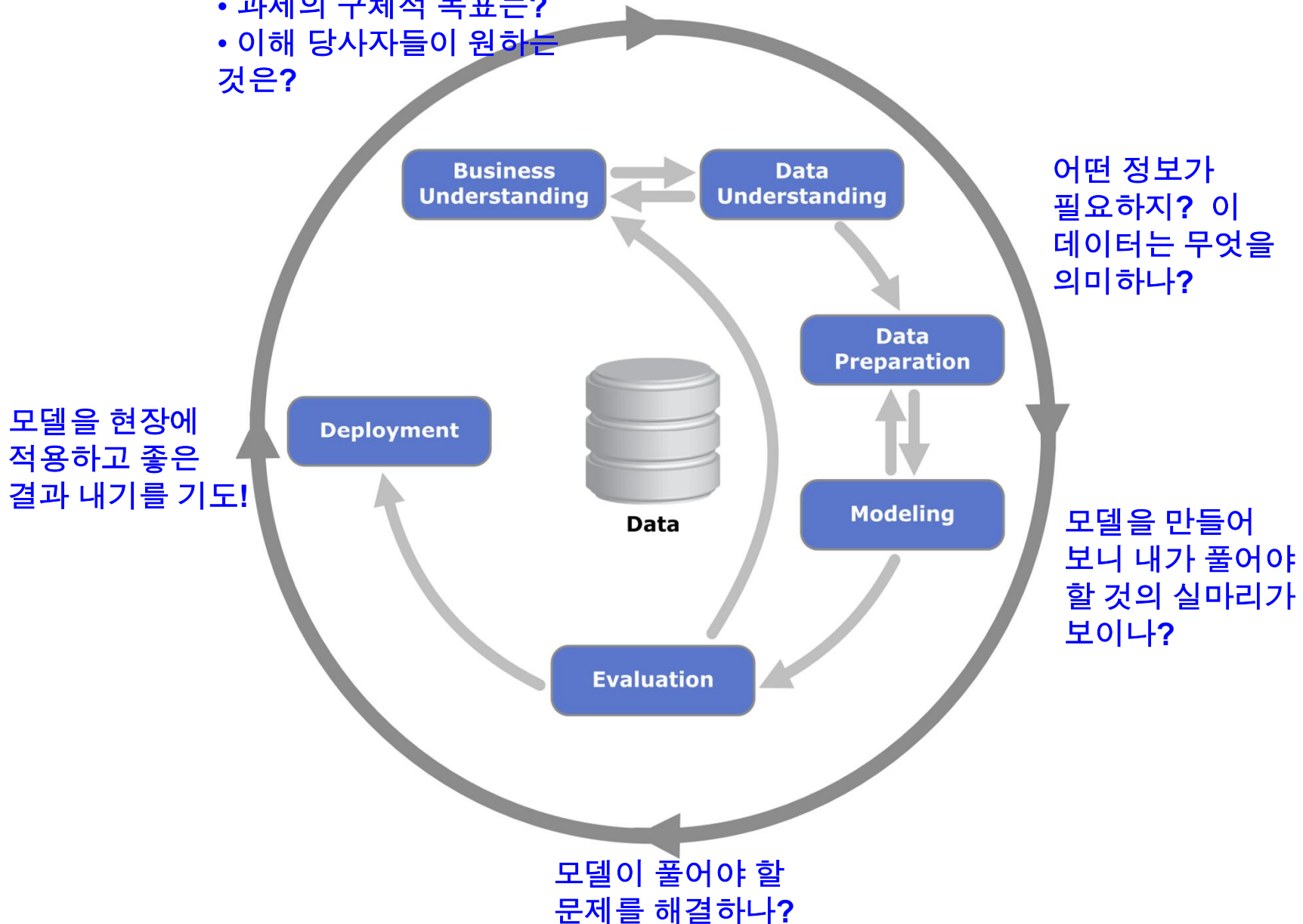
- 인공지능/머신러닝은 자동·자율적으로 동작하는 지능적 에이전트를 만들어 그 것이 인식-판단-행동토록 하여 상황을 의도한대로 만듬이 목표
- 데이터사이언스는 (사람이) 데이터를 잘 분석해 (What's the goal/question?, Get the data, Explore the data, Model the data, Make sense of the model, Communicate the results) 그 결과를 주로 (사람)에게 제공

R or Python?

데이터 분석/사이언스 프로세스 : CRISP-DM

시작:

- 내가 풀어야 하는 것은?
- 과제의 구체적 목표는?
- 이해 당사자들이 원하는 것은?



데이터 분석과제의 이해당사자

이해 당사자	역할
프로젝트 스폰서	프로젝트 발주, 비즈니스적 효용 대변, 과제 결과에 책임
고객	사용자적 효용 대변, 도메인 지식을 갖추고 있음
데이터 사이언티스트	데이터분석과제의 아키텍트, 분석과제의 실행, 다른 이해당사자들과 소통, 이해관계 중재
데이터 엔지니어	데이터 사이언티스트와 협조하여 데이터 수집, 저장, 관리
운용 부서	과제 결과를 배치/배포, 데이터 분석 인프라 관리

- 모든 이해당사자들의 의견이 중요. 그 중에도 프로젝트 스폰서가 제일
- 목표치 설정과 기대치 관리가 프로젝트 성공에 매우 중요

기대치 설정 & 관리

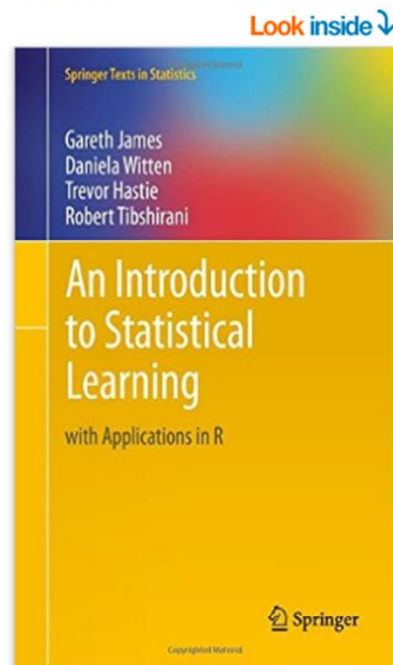
- 데이터 예측 분석이 “요술방망이”가 아님을 전달
- 데이터 사이언티스트가 현업보다 현업에 대한 통찰이 (정상적인 경우) 더 클 수가 없음을 전달
- 구체적, 검증가능한 목표를 잡는다. But, 여러 이유로 이 것이 불가능하면?
- 빅데이터에도 “Garbage In, Garbage Out”
- “투자한 만큼 결과를 얻는다”? 모든 IT 분야 중 데이터분석 과제만큼 이 문구가 많이 틀리는 경우가 없음을 전달
- 말이 안되면 “빠져나오기”를 준비. 없던 정보가 갑자기 생기는 경우도, 데이터가 스스로 똑똑해지는 경우도 없고, 데이터 분석 과제가 의지 싸움이 아님을 기억한다
- Good Luck

교재

An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics) 1st ed. 2013,

by Gareth James (Author), Daniela Witten (Author), Trevor Hastie (Author), Robert Tibshirani (Author)

★★★★☆ 141 customer reviews



ISBN-13: 978-1461471370

ISBN-10: 1461471370

[Why is ISBN important?](#)

Trade in your item
Get a **\$27.34**
Gift Card.



Trade in

[Learn More](#)

eTextbook

from \$55.43

Hardcover

\$16.99 - \$65.07

Other Sellers

[See all 4 versions](#)

☐ Rent

\$16.99 - \$21.08

☐ Buy used

\$61.77

☒ **Buy new**

\$65.07

In Stock.

Ships from and sold by Amazon.com. Gift-wrap available.

List Price: ~~\$79.99~~ Save: \$14.92 (19%)

[33 New from \\$44.62](#)

Want it Tuesday, May 16? Order within **1 hr 3 mins** and choose **Two-Day Shipping** at checkout. [Details](#)

FREE Shipping.

Qty: 1



Add to Cart

[Turn on 1-Click ordering](#)

Ship to:

Select a shipping address: ▼

More Buying Choices

71 offers from \$16.99

[33 New from \\$44.62](#) | [36 Used from \\$55.19](#) | [2 Rentals from \\$16.99](#)

[See All Buying Options](#)

Primestudent

College student?

FREE shipping and exclusive deals [Learn more](#)

교재 (ISLR) 사이트

- <http://www-bcf.usc.edu/~gareth/ISL/>

ISLR 강의 by Hastie & Tibshirani

- <https://www.youtube.com/playlist?list=PLwyWqcwXkazRpdF7AUDb4GK3Nn8f9jUoK>

Elements of Statistical Learning (참고)

- <https://statweb.stanford.edu/~tibs/ElemStatLearn/> : 사이트
- https://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf : 책

Web Resources

● Analytics/데이터사이언스

- www.kdnuggets.com
- www.datasciencecentral.com
- www.webscience.org
- [www.dmoz.org/Computers/Software/Databases/Data Mining](http://www.dmoz.org/Computers/Software/Databases/Data_Mining)
- www.kaggle.com
- www.kdnuggets.com/courses/index.html & www.kdnuggets.com/education/online.html
- [en.wikibooks.org/wiki/Data Science: An Introduction](http://en.wikibooks.org/wiki/Data_Science:_An_Introduction)
- www.coursera.org/category/stats & www.coursera.org/category/cs-ai
- [http://videolectures.net/Top/Computer Science/Machine Learning/#](http://videolectures.net/Top/Computer_Science/Machine_Learning/#)

Python Resources

- <https://docs.python.org/3/>
- <https://wikidocs.net/5>
- <http://m.blog.naver.com/koromoon/220578871433>
- <http://www.kdnuggets.com/2016/04/datacamp-learning-python-data-analysis-data-science.html>
- http://do1.dr-chuck.com/pythonlearn/EN_us/pythonlearn.pdf
- <https://www.continuum.io/downloads>
- <http://pandas.pydata.org/>
- <http://scikit-learn.org/stable/>
- <https://github.com/rasbt/python-machine-learning-book/blob/master/docs/references.md>

Python Data Munging

- <http://pandas.pydata.org/pandas-docs/stable/index.html> : Pandas Documentation (Tutorial 부터)
- <http://cs231n.github.io/python-numpy-tutorial/> : Numpy 소개
- <http://docs.scipy.org/doc/numpy> : Numpy Documentation
- <http://www.engr.ucsb.edu/~shell/che210d/numpy.pdf> : 짧은 Numpy and Scipy 소개
- <http://nbviewer.jupyter.org/github/jrjohansson/scientific-python-lectures/blob/master/Lecture-4-Matplotlib.ipynb> : Matplotlib 소개
- <https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook#gs.aOOhcyo> : Jupyter Notebook
- <http://www.cin.ufpe.br/~embat/Python%20for%20Data%20Analysis.pdf> : Wes McKinney 책
- <https://github.com/wesm/pydata-book> : Materials and IPython notebooks for "Python for Data Analysis" by Wes McKinney
- <http://www.cs.tufts.edu/comp/150VAN/demos/DataWrangling.pdf> : Pandas Quick reference
- <http://www.webpages.uidaho.edu/~stevel/504/Pandas%20DataFrame%20Notes.pdf> : Pandas Notes

Training sample: A training sample is a data point x in an available training set that we use for tackling a predictive modeling task. For example, if we are interested in classifying emails, one email in our dataset would be one training sample. Sometimes, people also use the synonymous terms *training instance* or *training example*.

Target function: In predictive modeling, we are typically interested in modeling a particular process; we want to learn or approximate a particular function that, for example, let's us distinguish spam from non-spam email. The *target function* $f(x) = y$ is the true function f that we want to model.

Hypothesis: A hypothesis is a certain function that we believe (or hope) is similar to the true function, the *target function* that we want to model. In context of email spam classification, it would be the *rule* we came up with that allows us to separate spam from non-spam emails.

Model: In machine learning field, the terms *hypothesis* and *model* are often used interchangeably. In other sciences, they can have different meanings, i.e., the hypothesis would be the "educated guess" by the scientist, and the *model* would be the manifestation of this *guess* that can be used to test the hypothesis.

Learning algorithm: Again, our goal is to find or approximate the *target function*, and the learning algorithm is a set of instructions that tries to *model* the target function using our training dataset. A learning algorithm comes with a *hypothesis space*, the set of possible hypotheses it can come up with in order to model the unknown target function by formulating the *final hypothesis*

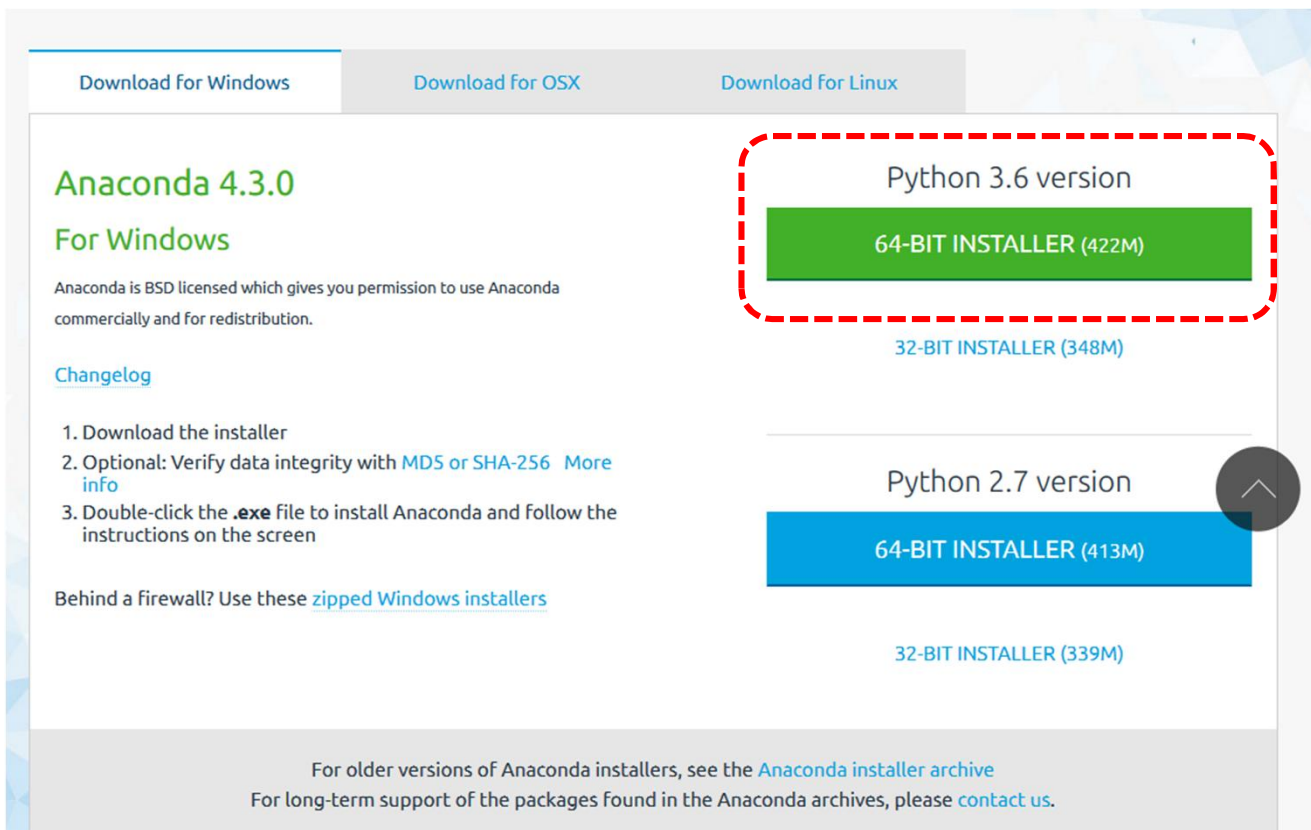
Classifier: A classifier is a special case of a *hypothesis* (nowadays, often learned by a machine learning algorithm). A *classifier* is a *hypothesis* or *discrete-valued function* that is used to assign (categorical) class labels to particular data points. In the email classification example, this classifier could be a hypothesis for labeling emails as spam or non-spam. However, a *hypothesis* must not necessarily be synonymous to a *classifier*. In a different application, our *hypothesis* could be a function for mapping study time and educational backgrounds of students to their future SAT scores.

파이썬 실습 환경 갖추기

우린 파이썬 실습에 파이썬 **아나콘다(Anaconda)**를 사용할 것입니다.

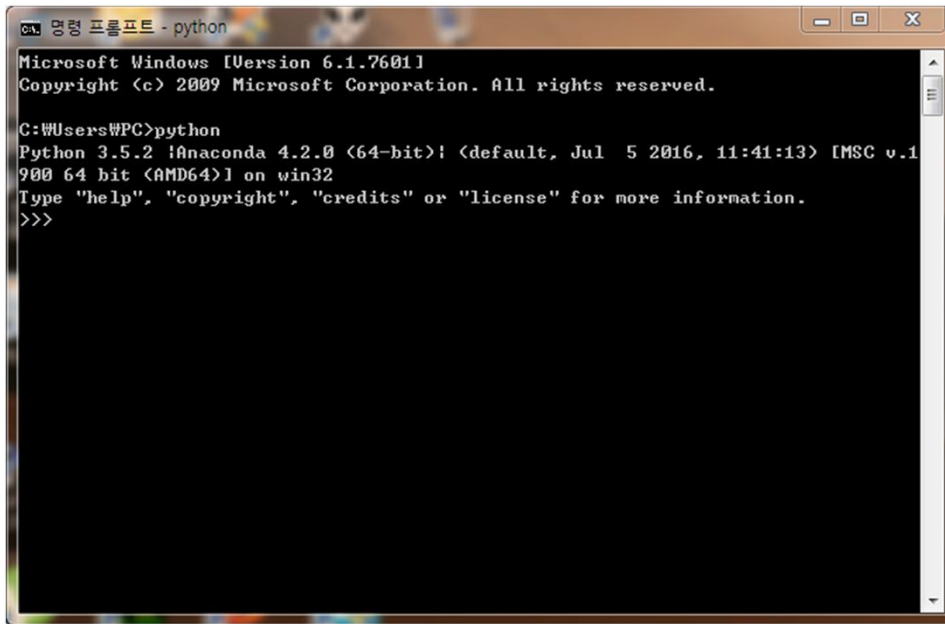
아나콘다는 기본 파이썬외에 여러 유용한 라이브러리를 포함한 번들 배포판입니다. 윈도우에서 설치하는 매우 쉽습니다. (파이썬, 아나콘다 다 뱀 종류죠)

1. <https://www.continuum.io/downloads> 로 갑니다.
2. 위 페이지에 가시면 아래와 같은 화면이 뜹니다.



3. 위 붉은 점선안의 아나콘다 설치 프로그램을 다운 받습니다. 윈도우용 64-bit 설치판입니다. (애플이나 리눅스를 사용하시는 분은 해당 페이지로 가셔서 설치 프로그램을 다운 받으시고 설명에 따라 설치하십시오)

4. "Anaconda3-xxx.exe" 같이 Anaconda로 시작하는 **.exe** 파일을 다운 받게 되실 것입니다. 다운받는데 시간이 꽤 걸릴 때도 있습니다.
5. 다운 받은 **.exe** 파일을 더블 클릭해 실행시키면 설치가 시작됩니다. 기본 옵션으로 설치하시면 됩니다.
6. 설치가 다 끝났으면 "명령 프롬프트" 창에서 "python"을 치고 '엔터'해 보세요.

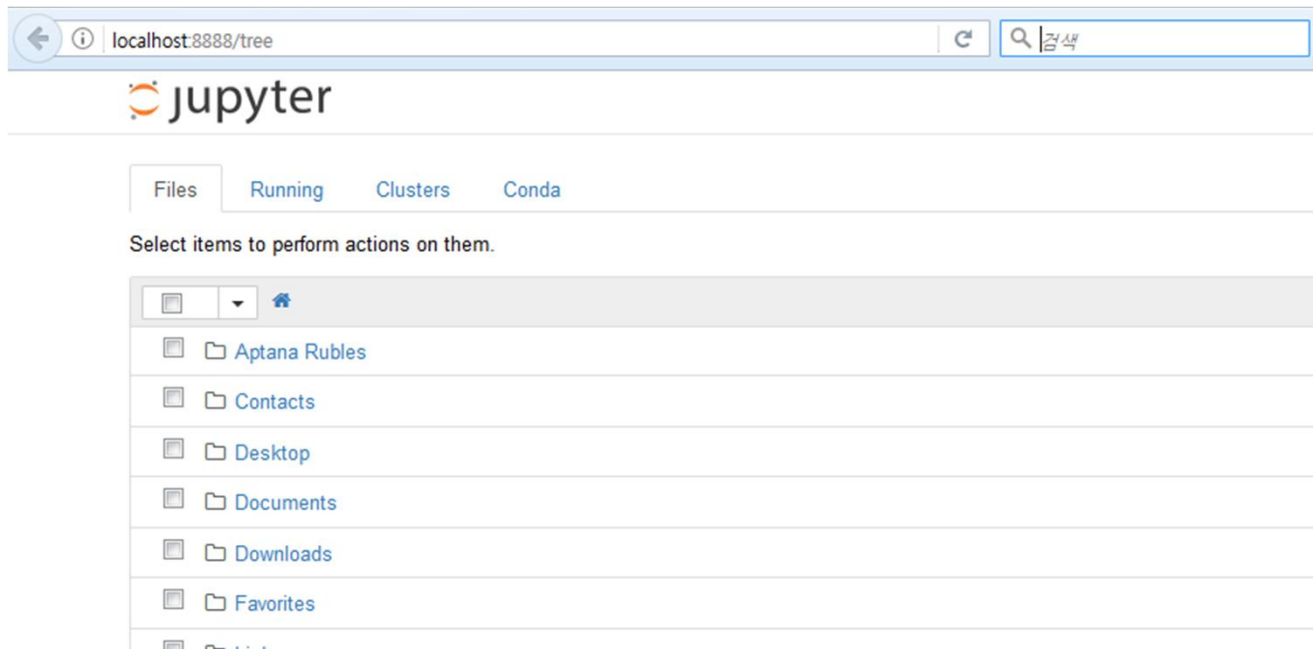


```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\WPC>python
Python 3.5.2 |Anaconda 4.2.0 (64-bit)| (default, Jul 5 2016, 11:41:13) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

위와 같은 창이 뜨면 잘 설치된 것입니다. 이 창은 파이썬 콘솔 창입니다. 여기서 파이썬 프로그램을 짤 수 있습니다. 창에서 $2 + 3$ 을 치고 엔터. 5가 뜨죠. 여러분이 막 파이썬을 쓰신 것입니다. `exit()` 을 치고 '엔터'해 보세요. 파이썬 콘솔 창에서 나가게 될 것입니다.

7. "명령 프롬프트" 창에서 "jupyter notebook" 을 치고 엔터해 보세요. 이상한 웹페이지가 웹브라우저에 뜰 것입니다. 대략 아래와 같은 모양일 것입니다.



이 건 "주피터 노트북" 이라고 하는 파이썬 개발(실습) 환경입니다. 우리가 사용할 파이썬 실습환경이죠. 앞으로는 "주피터 노트북" 환경을 그냥 "노트북"이라 할 께요.

웹브라우저에 뜬 노트북을 그냥 보시기만 하시고, 지금은 이것 저것 막 누르시려는 충동을 조금만 참아 주세요. (누르셔도 뭐 크게 잘 못 되는 것 없지만요. 잘 못 되면 책임 안집니다...). 강의할 때 노트북 쓰는 법을 알려드릴게요. 일단은 노트북 페이지를 그냥 닫으세요.

마음 급하신 분은 검색창에서 "주피터 노트북" 또는 "jupyter notebook" 으로 검색하면 많은 링크들이 보일 텐데 그 중에서 사용법을 따라 가 보는 것도 좋습니다.

지금은 여기까지...

웰컴 투 파이썬