



# ARTIFICIAL INTELLIGENCE 501

Lesson 4

Supervised Learning

# Formulating a Supervised Learning Problem

감독 학습 :

- 레이블이 지정된 데이터 집합 (기능 및 대상 레이블)을 수집합니다.
- 모델을 선택하십시오.
- 평가 측정 항목 선택 : 실적 측정에 사용합니다.
- 최적화 방법 선택 : 최상의 성능을 제공하는 모델 구성을 찾는 방법.

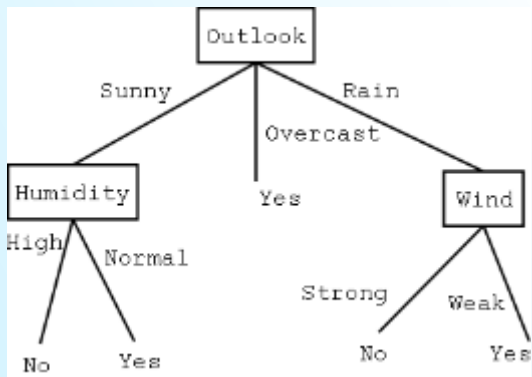
<sup>1</sup>*There are standard methods to use for different models and metrics.*

# Which Model?

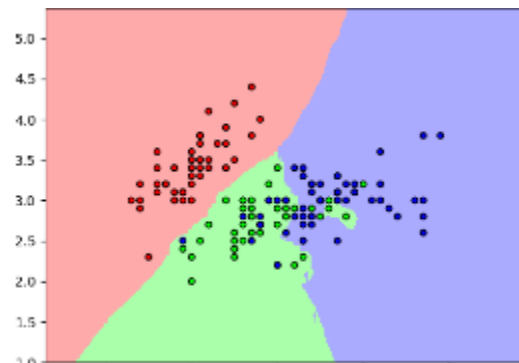
문제를 대표하고 자신의 장점과 단점을 가지고 각각 다른 방식으로 의사 결정을 내리는 많은 모델이 있습니다.

의사 결정 트리는 일련의 예 / 아니오 질문을 통해 예측을합니다.

가장 가까운 이웃이 가장 유사한 예를 투표로하여 예측을합니다.



*Decision tree*



*Nearest neighbors*

# Which Model?

Some considerations when choosing are:

- 훈련에 필요한 시간
- 예측 속도 향상
- 필요한 데이터의 양
- 데이터 유형
- 문제 복잡성
- 복잡한 문제를 해결하는 능력
- 간단한 것을 지나치게 복잡하게 만드는 경향

# Evaluation Metric

다음과 같이 성능을 측정 할 수있는 많은 메트릭이 있습니다.

- Accuracy : 예측이 실제 값과 얼마나 잘 일치 하는지를 나타냅니다.
- Mean Squared Error: 예측과 실제 값 사이의 평균 제곱 거리입니다.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

*Mean square error formula*



<sup>1</sup>The wrong metric can be misleading or not capture the real problem.

# Evaluation Metric

잘못된 측정 항목은 오해의 소지가 있거나 실제 문제를 포착 할 수 없습니다.

예 : 스팸 / 스팸이 아닌 정확도 사용을 고려하십시오.

- 100 개의 이메일 중 99 개의 이메일이 실제로 스팸 일 경우 매번 스팸을 예측하는 모델의 정확도는 99 %입니다.
- 이는 고정밀 측정 항목이 있더라도 중요한 실제 이메일을 스팸으로 강제 전송할 수 있습니다.



*Email*

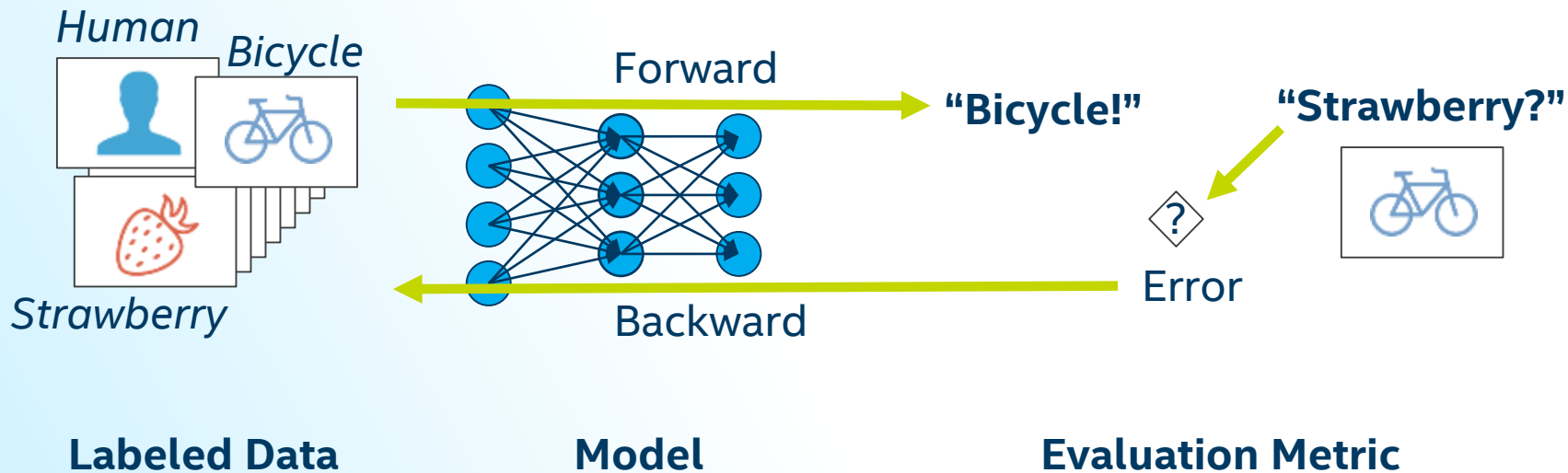
# Training

**Training Data:** 모델을 훈련하는 데 사용 된 데이터 집합.

**Optimization:** 최상의 성능을 발휘하도록 모델을 구성.

# Training

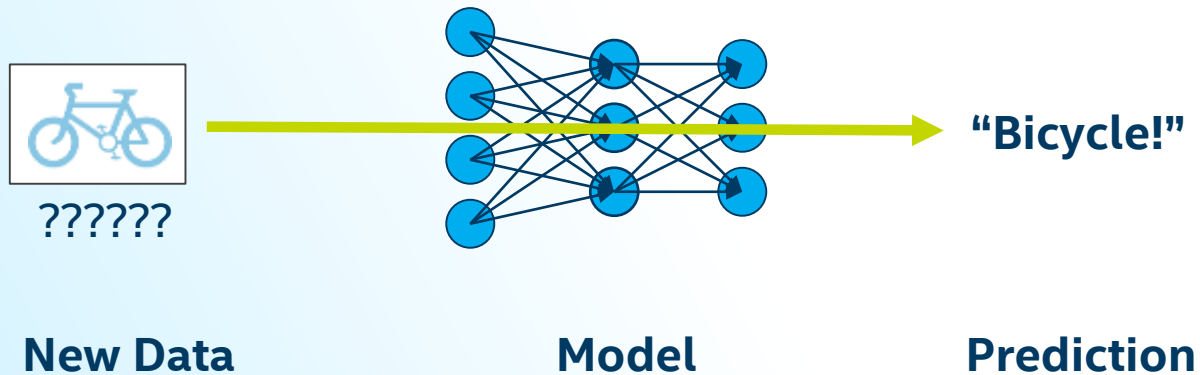
각각의 이미지들을 사용하여 최상의 구성을 찾기 위해 모델을 훈련시킵니다.





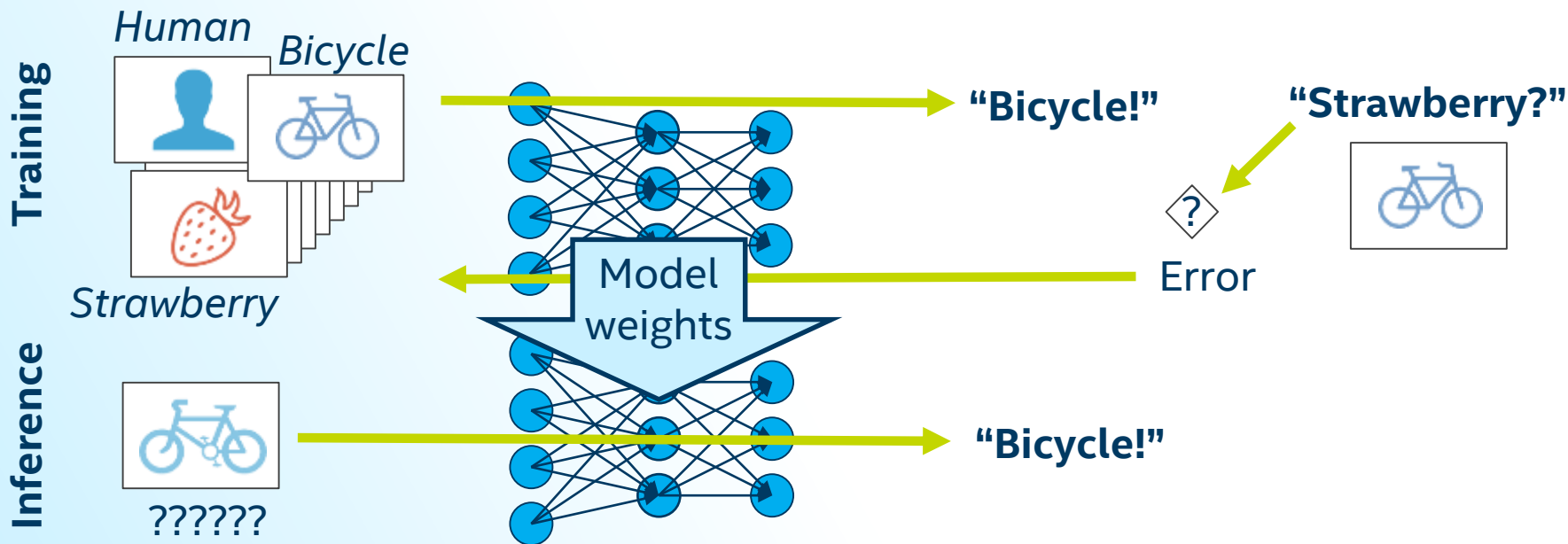
# Inference

모델이 트레이닝되면, 예측에 대한 새로운 틀이 제공되어 입력값이 무엇 인지를 예측합니다.



# Training vs. Inference

**Goal:** 많은 데이터를 이용해서 사물을 정확하게 인지할 수 있도록 모델을 훈련시켜 정확한 값을 예측할수 있는 모델을 만들고, 이 모델을 이용해서 다양한 사용자 어플리케이션에 적용이 가능하도록 합니다.



# Supervised Learning Overview

**Training:** 태그가 붙은 알려진 데이터로 훈련을 합니다.



**Inference:** 처음 입력되는 데이터를 훈련된 모델을 이용해 결과를 예측합니다.

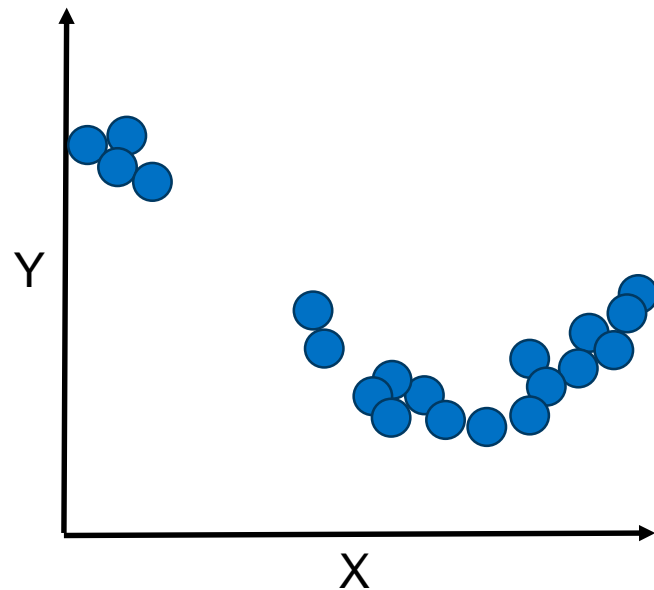




# OVERFITTING, TRAINING, AND TESTING DATA

# Curve Fitting: Overfitting vs. Underfitting Example

**Goal:** 데이터의 커브를 최적화 합니다.



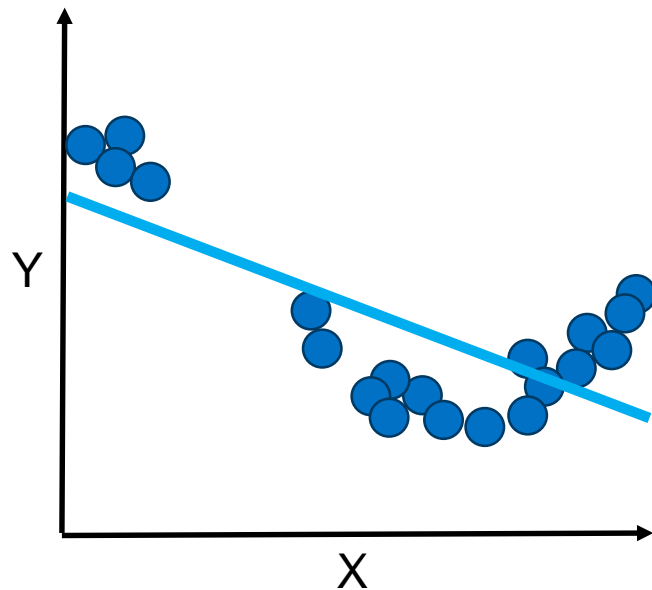
# Curve Fitting: Underfitting Example

데이터의 커브가 너무 단순합니다.

- 이것이 언더 피팅입니다.
- 훈련 데이터에 최적화 되지 않았습니다.
- 새로운 데이터에 대한 예측이 최적화 되지
- 않았습니다.

## Underfitting:

모델에 데이터의 체계적인 추세가 없습니다.



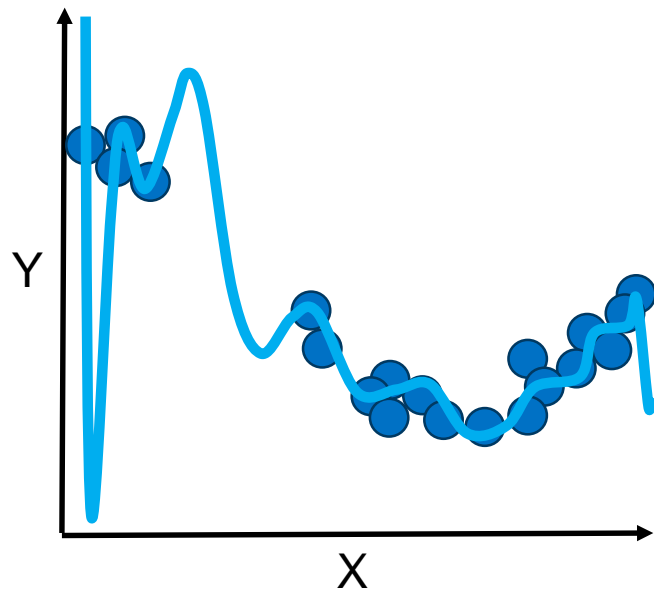
# Curve Fitting: Overfitting Example

데이터의 커브가 너무 복잡합니다.

- 이것이 오버피팅 입니다.
- 트레이닝 데이터에 최적화 되어 있습니다.
- 새로 입력되는 데이터 예측에 최적화 되지 않았습니다.

## Overfitting:

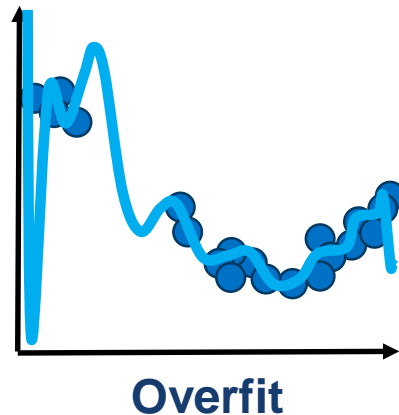
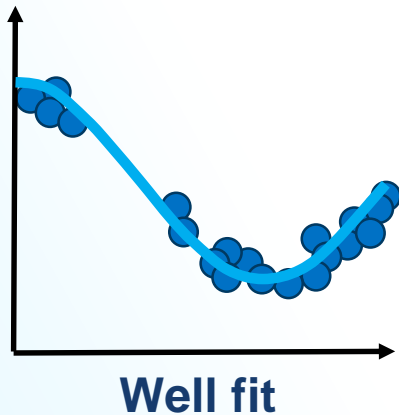
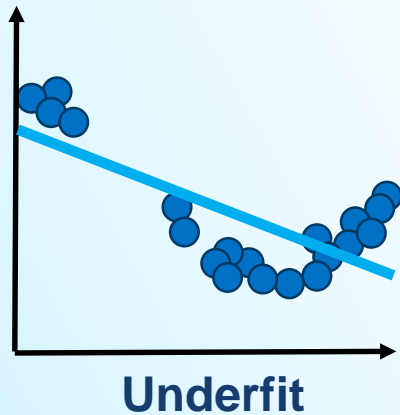
모델이 너무 민감해서 교육 데이터에 적합하지 않습니다.



# Curve Fitting Problem

**Problem:** 새로 입력될 데이터는 트레이닝에는 없습니다.

- 트레이닝 과정에서 어떻게 예측 성능을 극대화 할수 있을까요?
- 데이터를 지나치게 훈련시키면 주어진 데이터에만 만족해서 신규 데이터에 대해 예측 오류를 범합니다.





# Solution: Split Data Into Two Sets

**Training Set:** 트레이닝에 사용된 데이터

**Test Set:** 모델 성능을 평가하기 위해 사용되는 데이터

sepal length	sepal width	petal length	petal width	species
6.7	3.0	5.2	2.3	virginica
6.4	2.8	5.6	2.1	virginica
4.6	3.4	1.4	0.3	setosa
6.9	3.1	4.9	1.5	versicolor
4.4	2.9	1.4	0.2	setosa
4.8	3.0	1.4	0.1	setosa
5.9	3.0	5.1	1.8	virginica
5.4	3.9	1.3	0.4	setosa
4.9	3.0	1.4	0.2	setosa
5.4	3.4	1.7	0.2	setosa

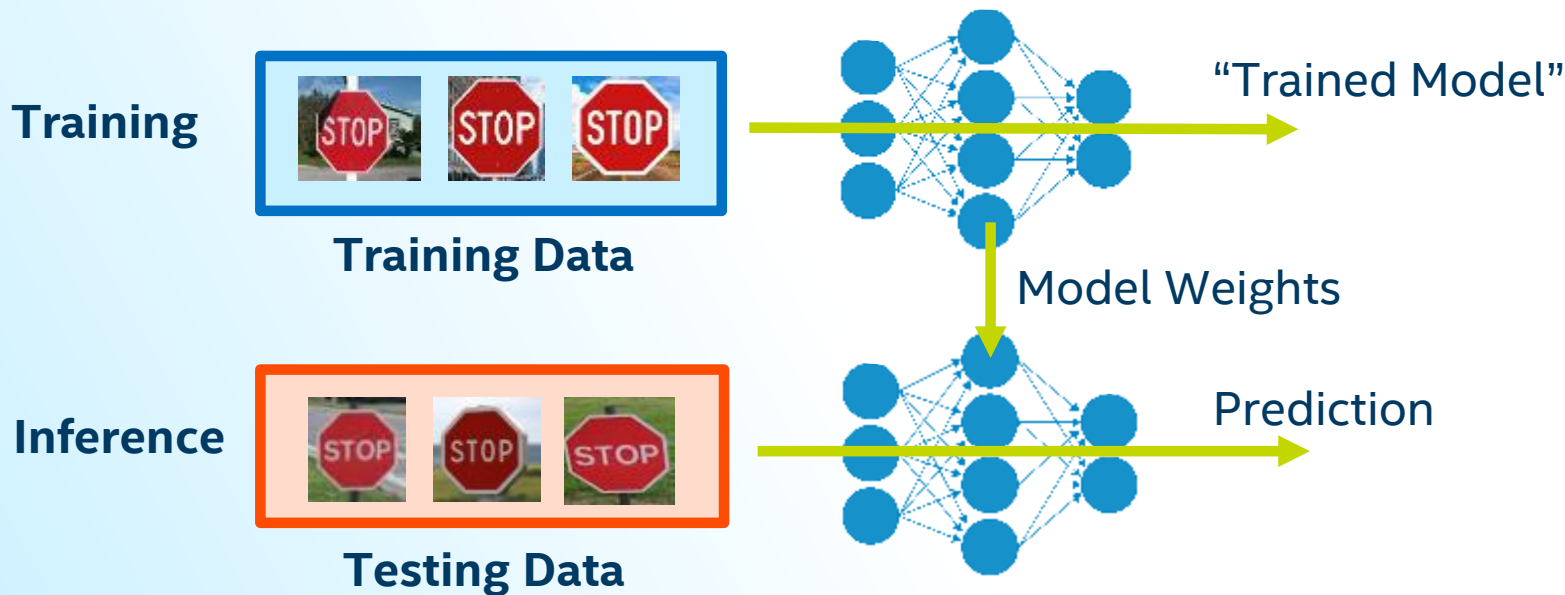
**Training Set**

**Testing Set**

<sup>1</sup> Not used during the training process.

# Train-Test Split

트레이닝에 사용되지 않은 데이터로 모델의 성능을 평가합니다.



























# PYTHON\* ECOSYSTEM

\*Other names and brands may be claimed as the property of others.

# Why Python\*?

- 범용 언어.
- Java \* 또는 C ++와 같은 다른 언어와 관련하여 간단하고 읽기 쉬운 구문입니다.
- C ++ 및 Fortran과 같은 다른 언어로 작성된 응용 프로그램을 쉽게 사용할 수 있습니다.
- 적극적인 커뮤니티.
- 광범위한 라이브러리.

# Python\*: Highest Ranked Language<sup>1</sup>

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

<sup>1</sup>Source:  
IEEE Spectrum

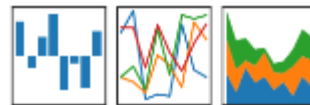
# Python\* Libraries for Data Science



**Fast numerical computing**

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



**Data analysis and manipulation**



**Machine learning**



**Deep Learning**

**matplotlib**

**Visualization**

# Intel® Distribution for Python\*

## Easy, Out-of-the-box Access to High Performance Python\*

- 수치 계산, 데이터 분석, 고성능 컴퓨팅 (HPC)을 위해 사전 구축되어 최적화되었습니다.
- 기존 Python을 대체하여 사용 가능 합니다. (코드를 변경하지 않아도 됨).
- Download from <https://software.intel.com/en-us/distribution-for-python>

# Intel® Distribution for Python\*

## Drive Performance with Multiple Optimization Techniques

- Accelerated NumPy / SciPy / Scikit-Learn 인텔® 수학 커널 라이브러리 (인텔® MKL).
- pyDAAL을 이용한 데이터 분석, 인텔® TBB (Thread Building Blocks), Jupyter \* 노트북 인터페이스, Numba, Cython \*을 통한 향상된 스레드 스케줄링.
- 최적화 된 MPI4Py \* 및 Jupyter notebook으로 쉽게 확장하십시오.

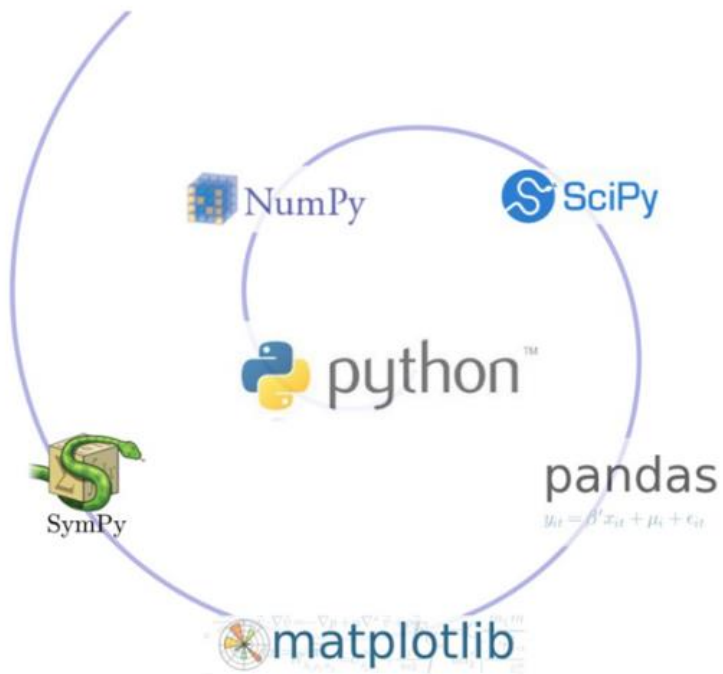


Python: 당연히, 파이썬을 이야기하지 않으면 앙금 없는 찌빵! 프로그래밍 언어

NumPy: 수치 계산을 위한 기본 패키지

Matplotlib: 그리기 패키지

SciPy: 신호 처리, 최적화, 통계 등의 기능을 갖는 패키지들을 묶어 놓은 라이브러리



pandas, scikit-learn, Jupyter 등의 다른 패키지들과 함께 사용할 수 있다고 합니다.

[출처] 파이썬(python) 공부하기 17 - SciPy 둘러보기|작성자 EngineerK

# Intel® Distribution for Python\*

## Faster Access to Latest Optimizations for Intel® Architecture

- Conda \* 및 Anaconda Cloud \*를 통해 distribution 및 개별 최적화 된 패키지를 제공 합니다.
- 파이썬 메인 트렁크로 업 스트림 된 최적화 솔루션 입니다.