

Contents ⚙

- 1 학습 알고리즘(Learning Algorithm)
 - 1.1 통계주론에서 기계학습/딥러닝학습으로
 - 1.2 학습종류: 지도학습/비지도학습/강화학습
 - 1.3 학습 방식에 따른 세부 알고리즘
 - 1.4 정확성 vs. 설명력
- 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시분)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯츠모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)

1 학습 알고리즘(Learning Algorithm)

[Open in Colab](#)

"빅데이터 핸들링을 위한 기초 프로그래밍을 넘어 정교한 패턴 추출을 위한 알고리즘 모델링으로 넘어갈 수 있음"



• 세부 단계:

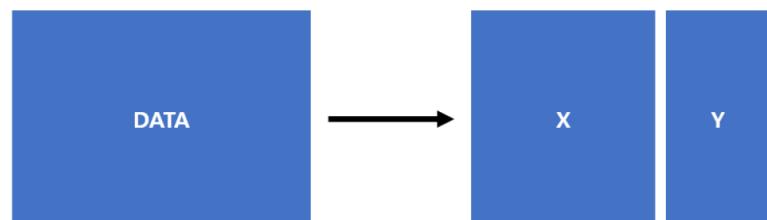
✓ 데이터 전처리: (0) 쓸모 없을 뻔한 Raw를 쓸모 있는 Data로 변환

100	T50	회수	111	TPU	...
few	Gds	Hvi	Rew	Fa	...
Fre	CT	QTP	D	합	...
'1'	1	23	22	NaN	43
76	NaN	43	32	1	8
'Hi'	NaN	NaN	NaN	NaN	87
23	98	NaN	64	46	NaN
90	NaN	'WW'	24	'KK'	4
NaN	2	NaN	NaN	NaN	6
64	NaN	90	'IU'	4	76

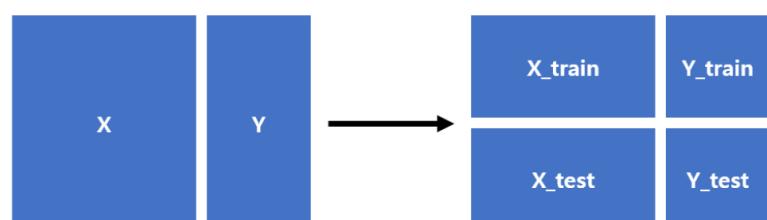
→

번호	시간	총량	기간	누적	...
1	1	23	22	21	43
76	33.3	43	32	1	8
5	33.3	52	35	21	87
23	98	52	64	46	61
90	33.3	2	24	33	4
55	2	52	35	21	6
64	33.3	90	11	4	76

✓ 데이터 분할: (1) 목표/종속변수 Y와 설명/독립변수 X설정



✓ 데이터 분할: (2) 학습데이터 Train과 예측 데이터 Test로 분할



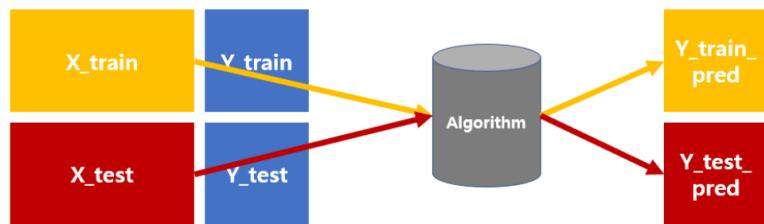
✓ 모델링: (3) 분석 목적에 맞는 알고리즘(Base & Advanced) 후보들 준비



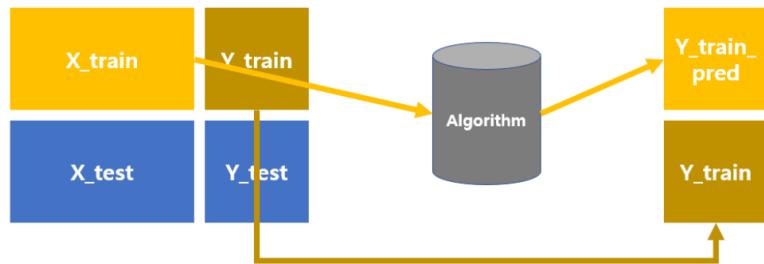
- 1.3 학습 알고리즘의 세부 내용은 뒤에 다룬다
- 1.4 정확성 vs. 설명력
- 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시분)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯츠모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)



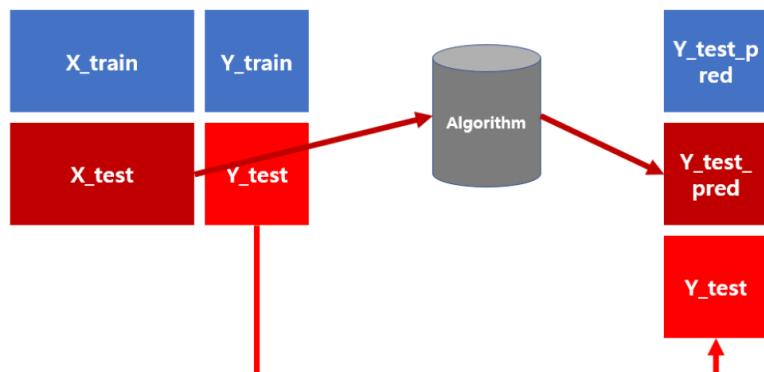
✓ 모델링 & 학습: (4) 알고리즘 평가를 위해 Train/Test의 예측값 추정



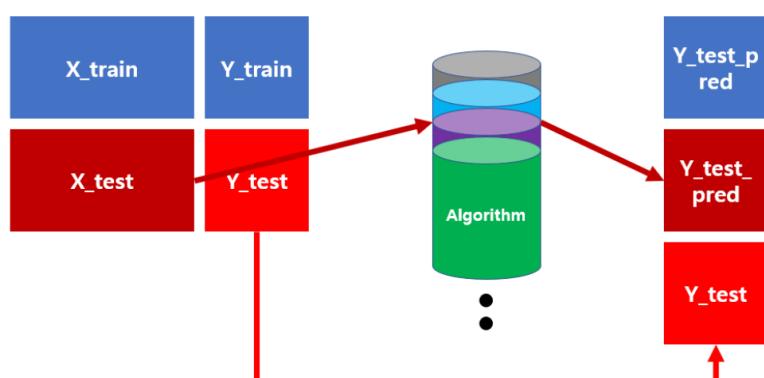
✓ 평가: (5) 학습(Train)이 잘 되었는지 알고리즘 성능검증



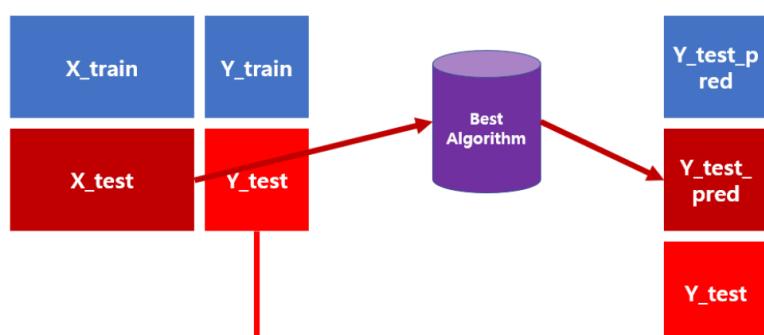
✓ 평가: (6) 예측(Test)이 잘 되었는지 알고리즘 성능검증



✓ 최적 알고리즘 선택: (7) 알고리즘을 변경하여 위 과정 반복 후



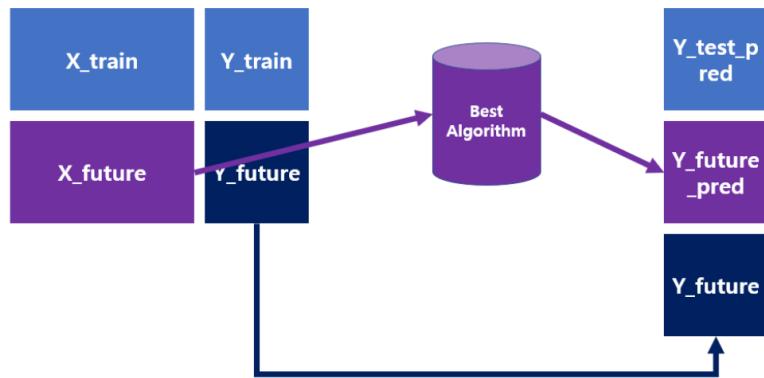
✓ 최적 알고리즘 선택: (7) 최고 성능의 알고리즘 선택



- 1.3 데일로그에 따른 세트업 트로픽
- 1.4 정확성 vs. 설명력
- ▼ 2 학습 알고리즘을 활용하는 Python 라이브러리
 - 2.1 실무에서 일반적 사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시분)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯츠모델즈)
 - 2.2.3 sklearn (사이킷런)
 - 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)

✓ 배포: (8) 실제 비즈니스 서비스 현업 적용 및 매출/수익/개선 정도 평가

- 1.3 학습 알고리즘 선택 및 평가
- 1.4 정확성 vs. 설명력
- 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시본)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯츠모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)



1.1 통계추론에서 기계학습/딥러닝학습으로

“데이터 과학은 크게 2가지 관점으로 발전”

- 통계학(Inferential Statistics):

- 데이터 과학의 근간
- 통계학 기반의 다양한 기능들은 딥러닝, 패턴인식, 기계학습 등에서 사용중

- 컴퓨터공학(Computer Science):

- 부품 가격, 절하와 성능 향상은 분석 성능에 영향
- 통계학에 컴퓨터 공학적인 접근을 받아들이게 하고, 통계와 기계학습 영역이 결합되어 시너지를 이루게 됨

- 통계학습(Statistical Learning) vs 기계학습(Machine Learning): 알고리즘 생성 방식

STATISTICS



Limited data samples
Focusing modeling & Tight assumptions

소수에서 다수를 추정

MACHINE LEARNING



Lots of data samples
Focusing optimization & No rigid assumptions

과거에서 미래를 예측

(루이, 다빈치랩스)

“데이터를 통해 문제 해결한다는 점은 일맥상통하나, 해결하는 목표/전략/방식에 대한 출발점이 다르며 점차 경계가 모호해지고 있음”

- 통계학습:

- 인공지능 및 기계학습에 대한 통계의 대응
- 기술통계, 추론통계에서 나아간 개념이지만 엄밀히 말하면 결국 추론 통계
- 실패의 위험을 줄여 신뢰성을 높이는 것
- 모델기반 사고방식과 데이터기반 사고방식을 모두 활용
- 정확도 자체에 매몰되지 않고 모델설명력과 다양한 가정 고려
- 모델들이 대부분 내부 구조 파악이 쉬운 화이트박스(White-box) 모형/알고리즘

- 기계학습:

- 인공지능, 패턴인식 등의 발전 역사와 결이 같음
- 에이전트라는 인간과 비슷한 녀석이 사물을 보고, 듣고, 인식하게 하는 것을 목적으로 발전
- 성공의 확률을 높이는 것
- 인과성보다 정확도에 굉장히 의존적이고 모델이나 가정에 크게

- 1.3 학습 알고리즘 선택 및 평가
- 1.4 정확성 vs. 설명력
- 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시본)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯츠모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)

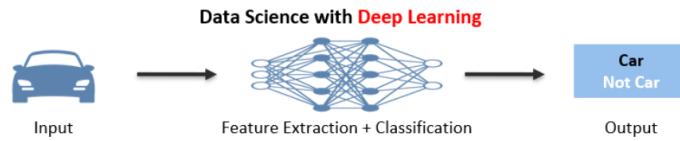
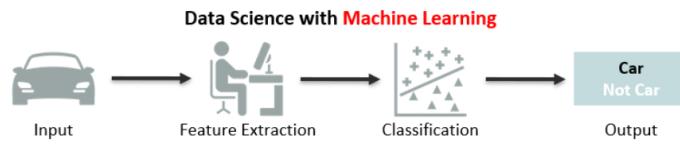
- 관심 없음
- 모델들의 내부 구조를 속속들이 알아야 할 이유가 없고 **맞추면 장땡**
 - 모델 내부 구조를 알 수가 없는 **블랙박스(Black-box)** 모형/알고리즘

통계학습(Statistical Learning)		기계학습(Machine Learning)
이론적 배경	통계학	컴퓨터과학
발전 기반	통계학, 수치해석 등	패턴인식, 인공지능 등
모형 구조	대부분 화이트박스	대부분 블랙박스
관심 목적	설명력, 실패위험 줄이기	정확성, 성공확률 높이기
주 사용 대상 데이터	관측치 및 변수가 적은 경우	관측치 및 변수가 많은 경우
상황/ 가정 반영	의존적 (독립성, 정규성, 등분산성 등)	독립적 (대부분 무시)
학습 방법	데이터에 맞게 최적화 중점	반복학습으로 모델 구축 중점
성능 평가	데이터의 해석과 가정 적합성 등	분할 데이터 반복 평가
특징	가설(Hypothesis), 모집단(Population), 표본(Sample)에 기반하여 데이터를 기술(Descriptive)하거나 추론(Inference)하는데 이용	예측력(Prediction) 중심의 다양한 문제 해결을 위한 지도(Supervised), 비지도(Unsupervised), 강화학습(Reinforcement) 등의 방법론 구축에 이용
문제 예시	대기오염과 호흡기 질환의 관계 배너위치에 따른 컨텐츠 클릭 빈도 변화 신규 장비의 불량률 감소 효과 분석 임상 상황을 통한 신약의 효능 분석	이미지 데이터의 객체 구분 상황이나 사물인식 성능 향상 음성인식을 통한 AI스피커 성능 향상 MRI데이터 사용 암 환자 조기 진단

1.3 학습 알고리즘 비교 세션

1.4 정확성 vs. 설명력

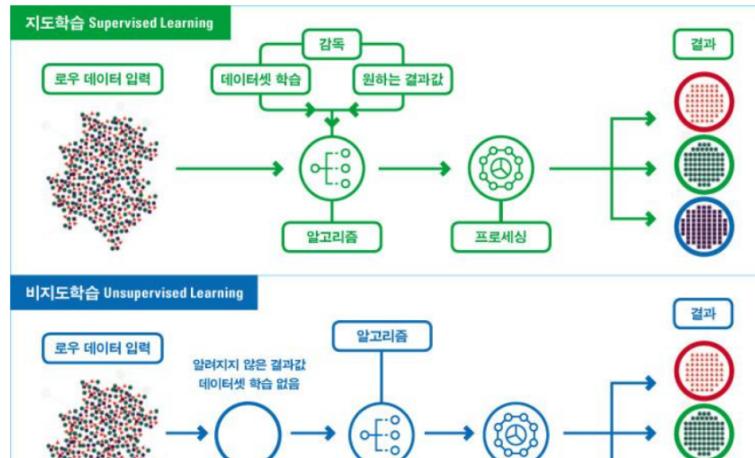
- 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (파andas)
 - 2.1.3 matplotlib (맷플롯립)
 - 2.1.4 seaborn (시본)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯스모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)



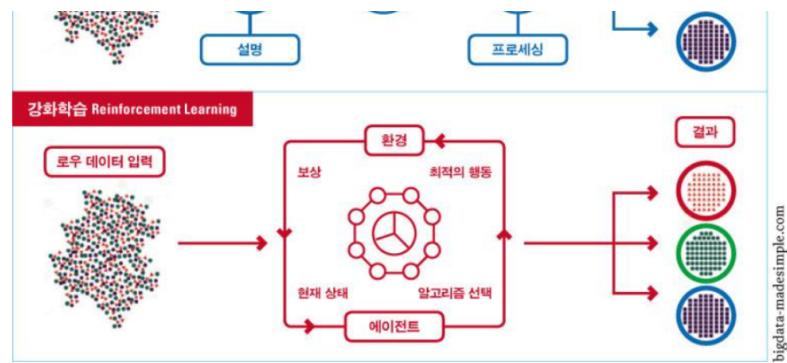
1.2 학습종류: 지도학습/비지도학습/강화학습

0) 어떤걸 풀수 있는가?

- 전체적으로 지도학습/비지도학습/강화학습 이라는 3가지 종류의 알고리즘으로 해결됨

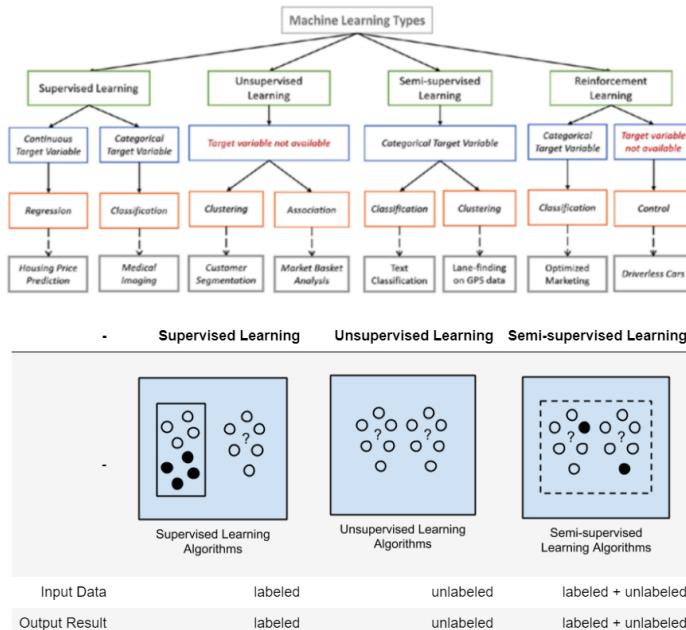


- 1.3 학습 모형에 따른 세가지 알고리즘
1.4 정확성 vs. 설명력
- ▼ 2 학습 알고리즘을 활용을 위한 Python 라이브러리
- ▼ 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시본)
 - ▼ 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯츠모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - ▼ 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)



bigdata-madesimple.com

- 4가지 종류의 알고리즘 분류:



알고리즘 종류	학습 방향
지도학습 알고리즘	기계에게 문제와 답을 학습 시킨 후 향후 답을 예측
비지도학습 알고리즘	기계에게 문제만 학습 시킨 후 스스로 패턴을 고려하여 답을 예측
강화학습 알고리즘	아무것도 모른채 일단 실전에 뛰어들어 시행착오로 학습하고 스스로 개선시켜 향후 답을 예측 데이터와 상호작용을 반복하기에 시간이 오래걸리나 가장 강력하고 진보적 인 방법

- 1.3 학습 모형에 따른 세가지 알고리즘
1.4 정확성 vs. 설명력
- ▼ 2 학습 알고리즘을 활용을 위한 Python 라이브러리
- ▼ 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시본)
 - ▼ 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯츠모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - ▼ 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)

- 각 알고리즘은 인간이 해결하고 싶어하는 특정 세부문제 맞춤해결 방식으로 진화중

알고리즘 종류	해결문제 종류	해결 예시
지도학습 알고리즘	예측문제	주관식 문제의 숫자형태의 정답을 해결
	분류문제	객관식 문제의 보기들 중 정답을 해결
비지도학습 알고리즘	군집문제	시험문제에 어떤 유형들이 있는지 해결
	차원변환 문제	시험문제의 풀이법을 다양한 관점으로 변환
강화학습 알고리즘		공교육/사교육 없이 스스로 지도학습/비지도학습 문제를 해결하고 오답노트도 스스로 만들고 학습하여 성적을 계속 끌어올림

Human Learning



Machine Learning

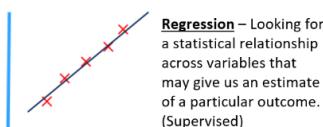




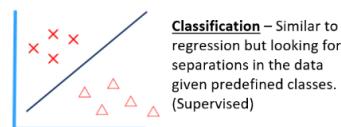
(루이, 다빈치랩스)

1) 실제 활용 사례:

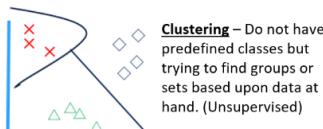
- How much is the stock of Samsung Electronics tomorrow?



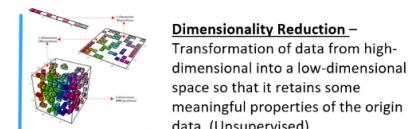
- Will Samsung Electronics' stocks rise or fall tomorrow?



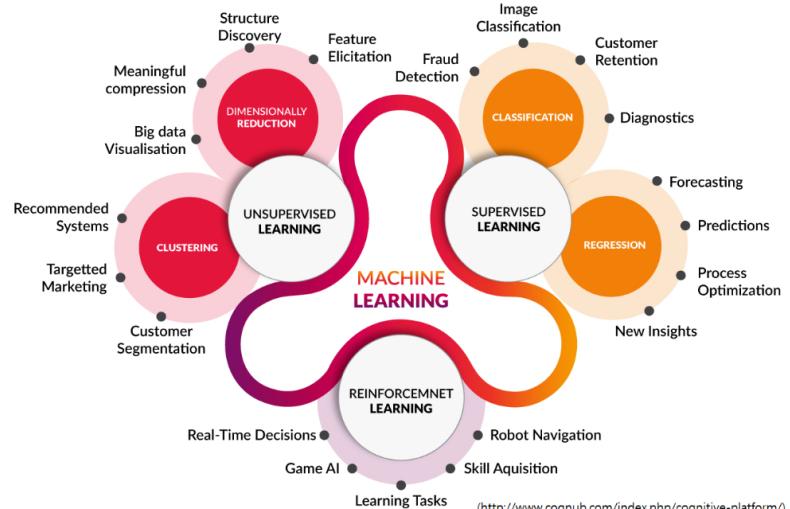
- Are Samsung Electronics and Naver similar business companies?



- What are the representatives among all stocks in the KOSPI?



- 1.3 학습 알고리즘 세부 내용
- 1.4 정확성 vs. 설명력
- ▼ 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시본)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯스모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)

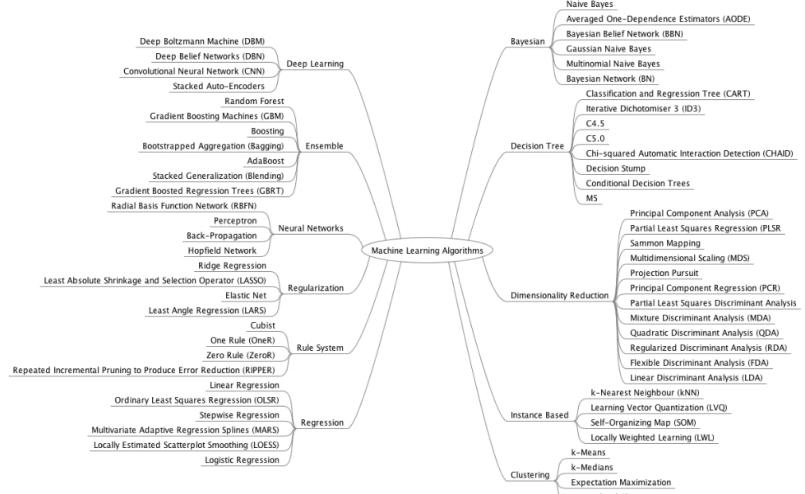


2) 정리:

알고리즘들은 크게 3~4가지 문제들을 해결하기 위해 만들어짐
내가 풀어야 할 문제가 무엇인지 알면 분석설정과 해결은 수동적인 작업(단순)

- 문제가 어디에 속하는지 → 문제정의 및 분석기획(가설설정) 가능
- 알고리즘의 입력은 무엇인지 → 분석기획(데이터준비 + 전처리) 가능
- 알고리즘의 출력은 무엇인지 → 분석기획(가설검정) 및 성능검증 가능

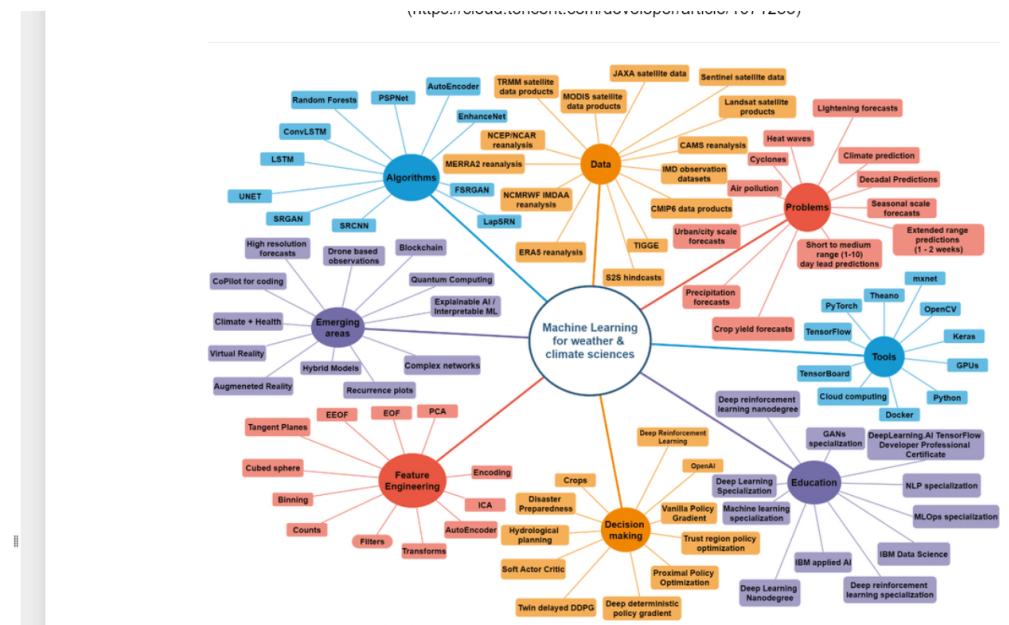
1.3 학습 방식에 따른 세부 알고리즘



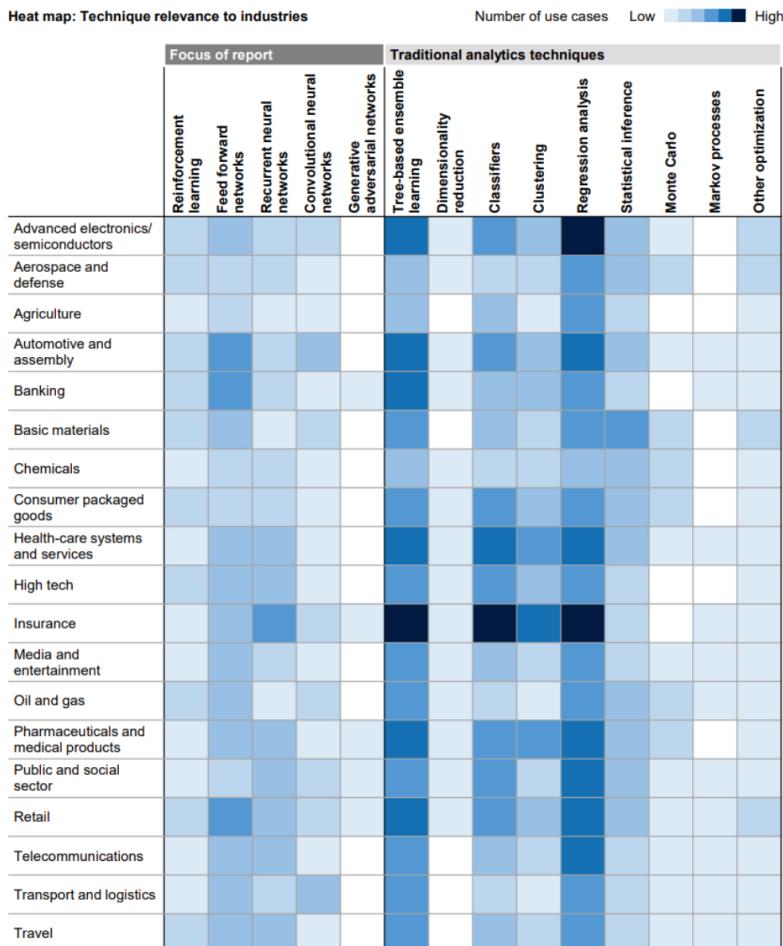
- 1.3 학습 알고리즘 세부 내용
- 1.4 정확성 vs. 설명력
- ▼ 2 학습 알고리즘을 활용을 위한 Python 라이브러리

<https://cloud.tencent.com/developer/article/1071985>

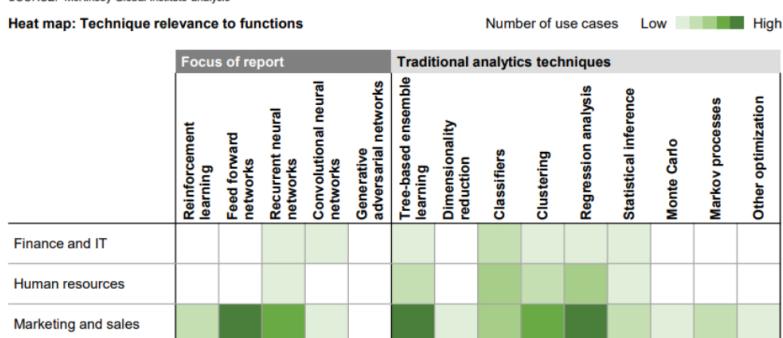
- ▼ 2.1 실무에서 딥러닝 사용 순서(정리)
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (플롯트립)
 - 2.1.4 seaborn (시본) - ▼ 2.2 통계주론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯스모델즈)
 - 2.2.3 sklearn (싸이킷런) - ▼ 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)

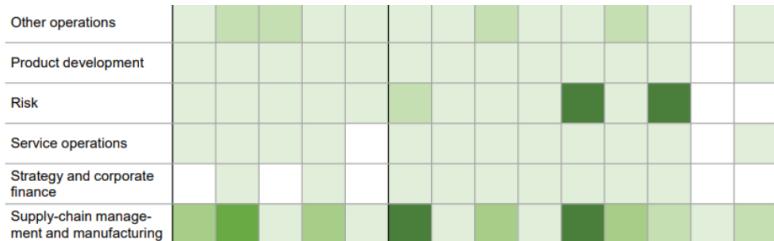


(Machine learning for Earth System Science (ESS): A survey, status and future directions for South Asia)



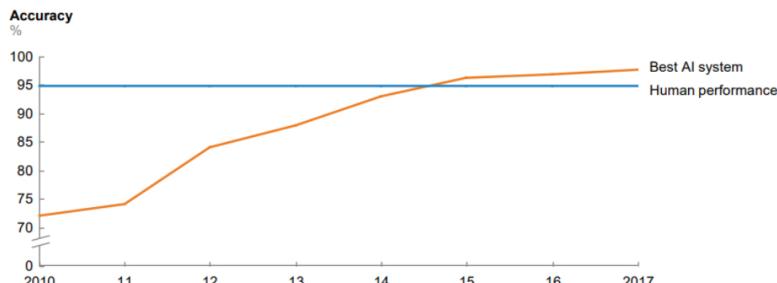
SOURCE: McKinsey Global Institute analysis





SOURCE: McKinsey Global Institute analysis

The ability of AI systems to recognize objects has improved markedly to the point where the best systems now exceed human performance



SOURCE: ImageNet Large Scale Visual Recognition Challenge; McKinsey Global Institute analysis

- 1.3 학습 알고리즘에 따른 세가지 라이브러리
- 1.4 정확성 vs. 설명력
- 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시본)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯스모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)

1.4 정확성 vs. 설명력

통계학습(Statistical Learning)		기계학습(Machine Learning)
이론적 배경	통계학	컴퓨터과학
발전 기반	통계학, 수치해석 등	패턴인식, 인공지능 등
모형 구조	대부분 화이트박스	대부분 블랙박스
관심 목적	설명력, 실패위험 줄이기	정확성, 성공확률 높이기
주 사용 데이터	관측치 및 변수가 적은 경우	관측치 및 변수가 많은 경우
상황/ 가정 반영	의존적 (독립성, 정규성, 등분산성 등)	독립적 (대부분 무시)
학습 방법	데이터에 맞게 최적화 중점	반복학습으로 모델 구축 중점
성능 평가	데이터의 해석과 가정 적합성 등	분할 데이터 반복 평가
특징	가설(Hypothesis), 모집단(Population), 표본(Sample)에 기반하여 데이터를 기술(Descriptive)하거나 추론(Inference)하는데 이용	예측(Prediction) 중심의 다양한 문제 해결을 위한 지도(Supervised), 비지도(Unsupervised), 강화학습(Reinforcement) 등의 방법론 구축에 이용
문제 예시	대기오염과 호흡기 질환의 관계 배너위치에 따른 컨텐츠 클릭 빈도 변화 신규 장비의 불량률을 감소 효과와 분석 일상을 통한 신약의 효능 분석	이미지 데이터의 객체 구분 상황이나 사물인식 성능 향상 음성인식을 통한 AI스피커 성능 향상 MRI데이터 사용 암 환자 조기 진단

1) 기계학습 활용 데이터분석의 현실:

"(이상적으로) 머신러닝 알고리즘에 데이터를 학습/적합/모델링 한다는 건.."

- (1) 사람/사물/시스템이 어떻게 동작하는지 이해 의 과정
- (2) 사람/사물/시스템이 만들어내는 데이터를 체계적으로 요약 하는 과정
- (3) 미래의 예측값과 실제값의 비교로 사람/사물/시스템을 일반화 하는 과정
- (4) 일반화된 사람/사물/시스템으로 더욱 효과적이고 체계적인 의사결정 하는 방법

"(현실적으로) 많은 시간과 비용이 투입되지만, 우리 사회를 이해 ((1),(2),(3)) <<< 빠른 의사결정 (4)에 집중되어 효과는 글쎄.."

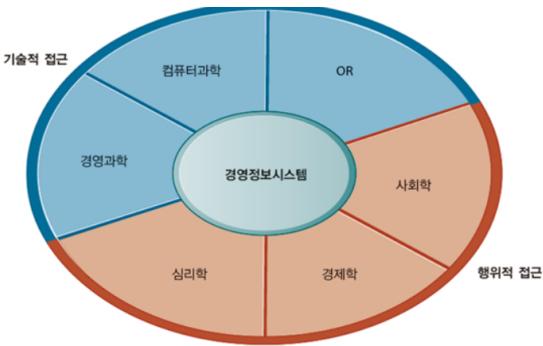
- 사회를 이해하기 위한 사회학/교육학/철학/경제학 등 의 학문은 ((1),(2),(3))에 집중
- 경영과학/컴퓨터과학/산업공학 등 의 학문은 기술적인 자동화나 인공지능을 반영하는 (4)에 집중
- 학문적 출신(?)에 따라 분석에 대한 관점 차이 또는 비즈니스 방향 차이가 존재할 수 있음

그림 1.9 정보시스템에 대한 현대적 접근

정보시스템에 대한 연구는 기술적·행위적 학문 분야에서 도출된 주제와 관점을 다루고 있다.

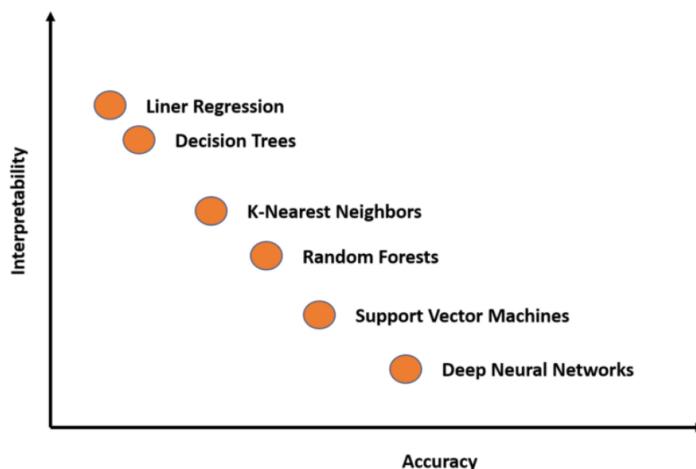
- 1.3 학습 알고리즘에 따른 세가지 라이브러리
- 1.4 정확성 vs. 설명력
- 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시본)

- ▼ 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯스모델즈)
 - 2.2.3 sklearn (싸이킷런)
- ▼ 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)



2) 정확성 vs. 설명력은 모두 욕심낼 수 없는 반비례 관계:

- 대부분의 기계학습 및 딥러닝 모델은 이론적 기반이 없기 때문에 1회성 추정을 반복하는 알고리즘
- 통계추론은 이론적 기반의 결과의 범위(신뢰구간)와 설명력을 제공하기에 반복추정이 필요없는 알고리즘



- 설명력 최근 연구동향:

- LIME
- DARPA

3) 반비례 관계 원인: 회귀분석(통계학습) vs 딥러닝(기계학습)

- 회귀분석(통계학습): 함수의 선형성을 추정하는 정확성보다 설명력에 집중하는 알고리즘
- 딥러닝(기계학습): 함수의 비선형성을 추정하는 설명력보다 정확성에 집중하는 알고리즘

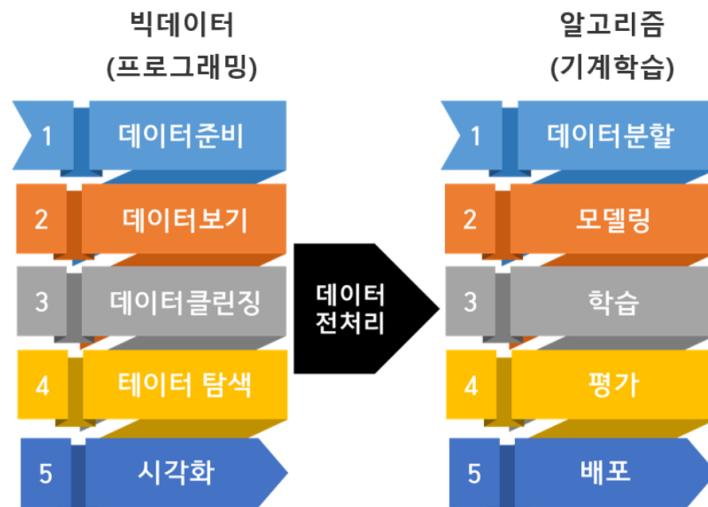
회귀분석		딥러닝
모델특징	-	-
분석목적	선형성파악(설명가능)	비선형성파악(설명불가)
이론적(수학적) 근거	존재	미존재
분석단계 특징(전처리)	-	-
데이터 로딩	Panel Data	다양(문이종으면 Panel)
데이터 빙간 채우기/삭제	분석필요	분석필요
데이터 컬럼 추가/삭제	분석필요+인감	분석필요+덜민감
데이터 분리	Train/Validate/Test	Train/Validate/Test
데이터 스케일링	분석필요/미필요	분석필요
분석단계 특징(모델링)	-	-
입력 확인 및 변환	Panel Data	다양(정해지지 않음)
데이터 모델연결	자동화	반자동화
비용함수(Cost)	최소제곱에러(MSE)	다양
추정함수(Optimizer)	고정(미분1회 대체가능)	다양(미분지속)
분석단계 특징(검증)	-	-
정확성지표	다양	다양
잔차진단활용	가능(분석필요)	불가
분석단계 특징(결과해석)	-	-
관계성 시각화/영향력 해석	가능(분석필요)	불가

- 1.3 회귀 분석에서의 시각화 방법
- 1.4 정확성 vs. 설명력
- ▶ 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (맷플롯립)
 - 2.1.4 seaborn (시본)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯스모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)

2 학습 알고리즘을 활용을 위한 Python 라이브러리

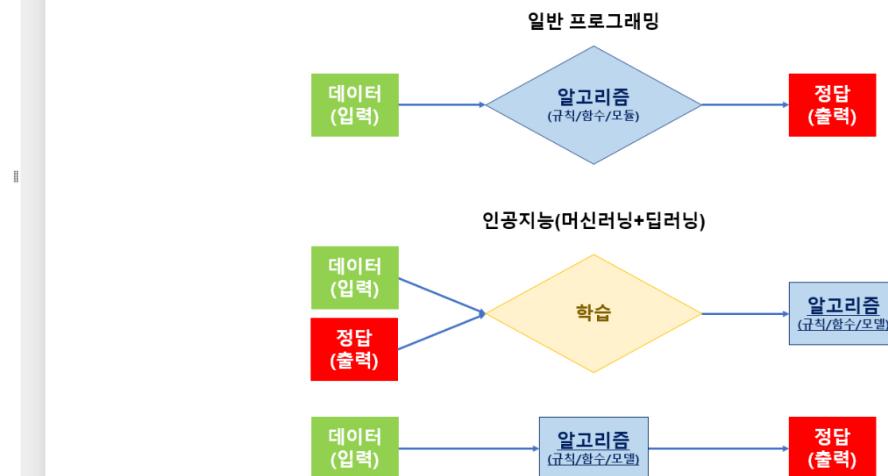
"빅데이터 핸들링을 위한 기초 프로그래밍을 넘어 정교한 패턴 추출을 위한 알고리즘 모델링으로 넘어갈 수 있음"

- 1.3 빅데이터 라이브러리
- 1.4 정확성 vs. 설명력
- 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시본)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯츠모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)



• 프로그래밍 vs 머신러닝:

- 데이터로 결과를 찾는 것이 아니라 데이터로부터 규칙성을 찾는 것
- 기존 프로그래밍 방식의 한계를 해결하게 된 새로운 계기



• 예시:

일반 프로그램	머신러닝 프로그램
다음 수식을 계산한 결과는?	□와 △에 들어갈 정수는?
(문제) <ul style="list-style-type: none"> ▪ $3 \times 2 =$ ▪ $4 \times -3 =$ ▪ $5 \times 8 =$ ▪ $7 \times -6 =$ 	✓ 입력 A : 숫자 ✓ 조건 B : 곱하기 x ✓ 동작 X : 숫자를 곱해서 결과 출력

"인공지능, 기계학습 그리고 딥러닝" 강의자료 15p 일부 변경, <https://www.slideshare.net/jinwonLee/ss-7044612>

• 요약:

문제	사람	일반 프로그래밍	인공지능
수천개의 숫자 곱하기	어려움	쉬움	쉬움
사진에서 사람얼굴 찾기	쉬움	어려움	쉬움

2.1 실무에서 일반적사용 분석용 패키지

- 직접/간접적인 데이터 패턴 확인을 위해 간단한 데이터 탐색 및 시각화를 포함하여 모든 종류의 알고리즘 모델링에 많이 사용
- 자주 사용하는 패키지들은 짧은 별명(alias)으로 import하여 사용하는 편

- 1.3 빅데이터 라이브러리
- 1.4 정확성 vs. 설명력
- 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (matplotlib)
 - 2.1.4 seaborn (시본)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯츠모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - 2.3 딥러닝 분석용 패키지

- 2.3.1 tensorflow (텐서플로우)
- 2.3.2 keras (케라스)
- 2.3.3 pytorch (파이토치)

2.1.1 numpy (넘파이)

선형대수 또는 수치해석 등의 계산기능 제공하며 수학 연산에서 가장 기본

```
# 설치
!pip install numpy
!conda install numpy
```

```
# 불러오기
import numpy as np
```

2.1.2 pandas (판다스)

테이블 형태의 데이터를 다루는 DataFrame 자료형 제공하며 데이터의 탐색과 정리에 유용한 필수 패키지

- R의 데이터프레임을 Python에 제공하는 목적이었으나 기능 추가

```
# 설치
!pip install pandas
!conda install pandas
```

```
# 불러오기
import pandas as pd
```

2.1.3 matplotlib (맷플롯립)

각종 그래프나 차트 등을 그리는 시각화 기능 제공

```
# 설치
!pip install matplotlib
!conda install matplotlib
```

```
# 불러오기
import matplotlib.pyplot as plt
```

2.1.4 seaborn (시본)

matplotlib에서 지원하지 않는 고급 통계 차트 시각화

```
# 설치
!pip install seaborn
!conda install seaborn
```

```
# 불러오기
import seaborn as sns
```

2.2 통계추론, 머신러닝 분석용 패키지

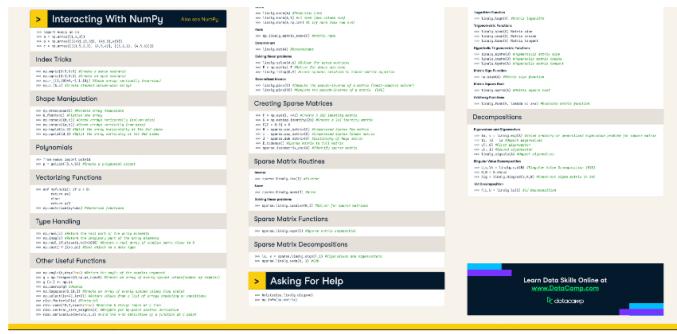
2.2.1 scipy (사이파이)

고급 수학함수, 수치적 미적분, 미분방정식 계산, 최적화 등의 과학기술 계산기능

```
# 설치
!pip install scipy
!conda install scipy
```

```
# 불러오기
import scipy as sp
```





2.2.2 statsmodels(스탯츠모델즈)

R에서만 가능했던 분석 기능들을 그대로 파이썬에서 이용 가능

- 예제 데이터셋
- 검정 및 추정
- 선형/로지스틱/강건/일반화 회귀분석
- 혼합효과모형
- 이산종속변수
- 요인분석
- 생존분석
- 시계열 분석
- 상태공간모형

설치

```
!pip install statsmodels  
!conda install statsmodels
```

불러오기

```
import statsmodels.api as sm
```

2.2.3 sklearn(싸이킷런)

다양한 머신러닝 모델(알고리즘)을 하나의 패키지로 제공하는 필수 패키지

- 예제 데이터셋
- 데이터 전처리
 - 스케일링
 - 누락데이터 처리
 - 데이터 분리 및 교차검증
 - Feature Selection
- 지도 학습(Supervised learning)
 - 회귀분석
 - LDA/QDA
 - Gauss Process
 - Naive Bayes
 - SVD
 - Bagging/Boosting
 - SGD, Perceptron
- 비지도 학습(Unsupervised learning)
 - Clustering
 - PCA
 - Gauss Mixture

1.3	numpy
1.4	정확성 vs. 설명력
▼ 2	학습 알고리즘을 활용을 위한 Python 라이브러리
▼ 2.1	실무에서 일반적사용 분석용 패키지
2.1.1	numpy (넘파이)
2.1.2	pandas (판다스)
2.1.3	matplotlib (맷플롯립)
2.1.4	seaborn (시본)
▼ 2.2	통계추론, 머신러닝 분석용 패키지
2.2.1	scipy (사이파이)
2.2.2	statsmodels (스탯츠모델즈)
2.2.3	sklearn (싸이킷런)
▼ 2.3	딥러닝 분석용 패키지
2.3.1	tensorflow (텐서플로우)
2.3.2	keras (케라스)
2.3.3	pytorch (파이토치)

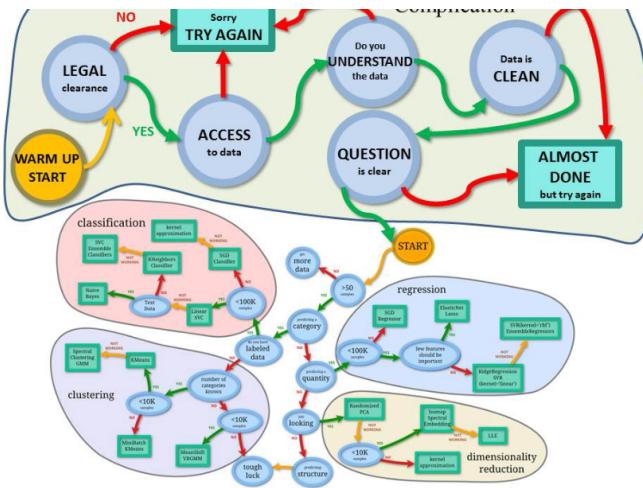
1.3	numpy
1.4	정확성 vs. 설명력
▼ 2	학습 알고리즘을 활용을 위한 Python 라이브러리
▼ 2.1	실무에서 일반적사용 분석용 패키지
2.1.1	numpy (넘파이)
2.1.2	pandas (판다스)
2.1.3	matplotlib (맷플롯립)
2.1.4	seaborn (시본)
▼ 2.2	통계추론, 머신러닝 분석용 패키지
2.2.1	scipy (사이파이)
2.2.2	statsmodels (스탯츠모델즈)
2.2.3	sklearn (싸이킷런)
▼ 2.3	딥러닝 분석용 패키지
2.3.1	tensorflow (텐서플로우)
2.3.2	keras (케라스)
2.3.3	pytorch (파이토치)

설치

```
!pip install sklearn  
!conda install sklearn
```

불러오기

```
import sklearn as sk
```



2.3 딥러닝 분석용 패키지

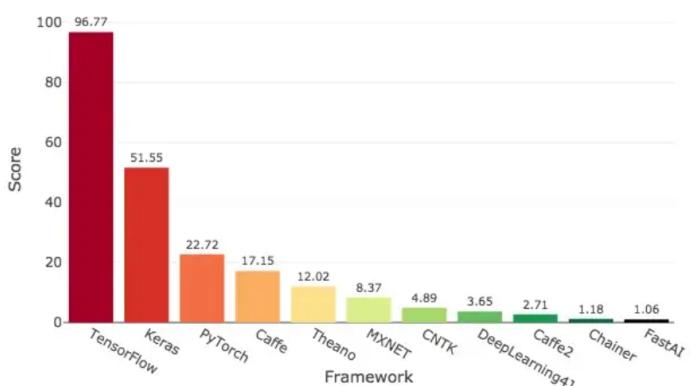
1) 머신러닝 및 딥러닝 플랫폼: Scikit-learn, Theano, Tensorflow, PyTorch, MXNet, NLTK, Keras

- 1.3 학습 모형의 학습 세부 알고리즘
- 1.4 정확성 vs. 설명력
- ▼ 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - 2.1 실무에서 일반적사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (판다스)
 - 2.1.3 matplotlib (맷플롯립)
 - 2.1.4 seaborn (시본)
 - 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯츠모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - ▼ 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)



(<https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c>)

Deep Learning Framework Power Scores 2018

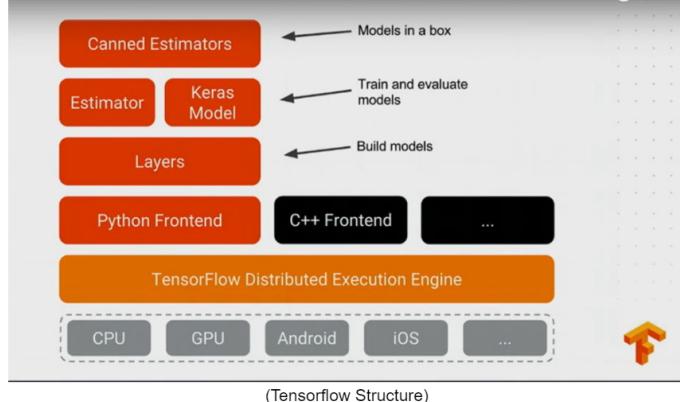


(<https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a>)

2.3.1 tensorflow (텐서플로우)

- 구글의 인공지능 개발부서에서 개발하여 내부적으로 사용하다 2015년 오픈 소스로 공개
- 파이썬의 문법과 다른 방식으로 동작하기에 텐서플로우 이해가 쉽진 않음
- 프로그래밍 언어들 중 파이썬은 느린 언어라 파이썬으로 딥러닝 구현 어려움
- 우수한 기능과 서비스로 가장 많은 사용자 확보
- 병렬처리를 지원하며 고급 신경망 구현 용이
- 웹, 모바일, 임베디드 시스템 등에서 머신러닝 적용 라이브러리 제공

- (1) 데이터 플로우 그래프를 활용하여 풍부하게 표현 가능
- (2) 단순 아이디어 테스트부터 실제 협업 서비스 단계까지 두루 사용
- (3) 목표와 계산/연산 방식만 설정하면 자동으로 미분과 최적화를 포함한 처리 완료
- (4) 파이썬 뿐만 아니라 C++, Go, Java, R 등 다양한 언어 지원



```
# 설치
!pip install tensorflow
!conda install tensorflow
!pip install --upgrade --user tensorflow
!pip install -U tensorflow-addons
```

```
# 불러오기
import tensorflow as tf
```

2.3.2 keras (케라스)

“파이썬으로 작성된 오픈소스(Open Source) 인공신경망 라이브러리”

- Tensorflow, MXNet, DeepLearning4j 등을 뒤에서 작동시켜 대부분의 신경망 프로그래밍 또는 모델링을 최소한의 방식으로 빠르고 간편하게 만들고 학습 시킬 수 있는 프레임워크
- 일반적인 사례들을 간단하게 분석할 수 있는 인터페이스 제공하여 사용자의 오류를 명확하게 피드백하는 편
- 백엔드 텐서플로우를 사용하지만 저수준 연산기반이 아닌 고수준 구성요소 연산 제공
- 모듈 구조로 구성되어 있지만 여러가지 새로운 아이디어를 반영하기 위해 쉽게 Customize 가능하여 현존하지 않는 신규 알고리즘을 개발하거나 기존 기능이나 엔진들과 연동이 쉬움

- (1) 다양한 딥러닝 네트워크 구조를 쉽게 생성 가능
- (2) 동일한 코드로 CPU와 GPU 모두에서 동일하게 실행 가능
- (3) 사용하기 쉬운 API로 딥러닝 모델의 프로토타입을 쉽게 구현 가능하고 자유롭게 조합도 가능

케라스 ► Keras

백엔드 엔진 ► TensorFlow / Theano / CNTK / ...

하드웨어를 위한 라이브러리 ► CUDA / cuDNN BLAS, Eigen

하드웨어 ► GPU CPU

딥러닝 소프트웨어와 하드웨어 선택

설치

```
!pip install tensorflow
!conda install tensorflow
!pip install --upgrade --user tensorflow
!pip install -U tensorflow-addons
!pip install keras
!conda install keras
!pip install keras-tqdm
```

1.3 학습 모델의 높은 시각화 능력
1.4 정확성 vs. 설명력

- ▼ 2 학습 알고리즘을 활용을 위한 Python 라이브러리
 - ▼ 2.1 실무에서 일반적 사용 분석용 패키지
 - 2.1.1 numpy (넘파이)
 - 2.1.2 pandas (파andas)
 - 2.1.3 matplotlib (맷플롯립)
 - 2.1.4 seaborn (시분)
 - ▼ 2.2 통계추론, 머신러닝 분석용 패키지
 - 2.2.1 scipy (사이파이)
 - 2.2.2 statsmodels (스탯츠모델즈)
 - 2.2.3 sklearn (싸이킷런)
 - ▼ 2.3 딥러닝 분석용 패키지
 - 2.3.1 tensorflow (텐서플로우)
 - 2.3.2 keras (케라스)
 - 2.3.3 pytorch (파이토치)

```
# 불러오기
import tensorflow as tf
import keras
```

2.3.3 pytorch(파이토치)

- 파이썬 언어로 사용하며, 진화된 토치 C/CUDA 백엔드를 사용
- 강력한 GPU 가속이 적용되는 파이썬으로 된 텐서와 동적 신경망
- 동적 신경망(Dynamic Neural Network)은 반복할 때마다 변경이 가능한 신경망
- 각 반복 단계에서 즉석으로 그래프를 재생성 하며, 텐서플로우는 단일 데이터 흐름 그래프
- 속도를 극대화하기 위해 인텔 MKL, 엔비디아 cuDNN, NCCL과 같은 가속 라이브러리를 통합

설치

START LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, 1.12 builds that are generated nightly. Please ensure that you have met the prerequisites below (e.g., numpy), depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also [install previous versions of PyTorch](#). Note that LibTorch is only available for C++.

Additional support or warranty for some PyTorch Stable and LTS binaries are available through the [PyTorch Enterprise Support Program](#).

PyTorch Build	Stable (1.12.0)	Preview (Nightly)	LTS (1.8.2)
Your OS	Linux	Mac	Windows
Package	Conda	Pip	LibTorch
Language	Python		C++ / Java
Compute Platform	CUDA 10.2	CUDA 11.3	CUDA 11.6
Run this Command:	NOTE: 'conda-forge' channel is required for cudatoolkit 11.6 conda install pytorch torchvision torchaudio cudatoolkit=11.6 -c pytorch -c conda-forge		

```
# 불러오기
import torch
torch.__version__
torch.cuda.is_available()
```