빅데이터 분석처리 과정

<회귀분석>

머신러닝의 이해(1)

본 수업의 내용

- 머신러닝의 개요
  - 머신러닝의 의의, 사례, 절차, 학습방법의 구분
  - 머신러닝 라이브러리 소개 - Scikit-Learn, Tensorflow, Pytorch
- 회귀
  - 선형회귀
  - 경사하강법
  - 다항회귀
  - 머신러닝체크포인트 - 과적합, 분산 편향 트레이드 오프, 교차 검증, 매개변수 스코어링
  - 규제가 있는 선형 모델 - 라쏘회귀, 릿지 회귀, 엘라스틱 회귀
  - 로지스틱회귀
- 분류
  - 의사결정트리
  - 랜덤포레스트
  - 그래디언트부스팅-adaboosting, XGBoosting, LightGBM
  - 앙상블Ensemble
- 군집
  - K-Means
  - DBSCAN
- Text분석
  - 텍스트 분류
  - 감성분석(소셜 분석, 리뷰 분석)

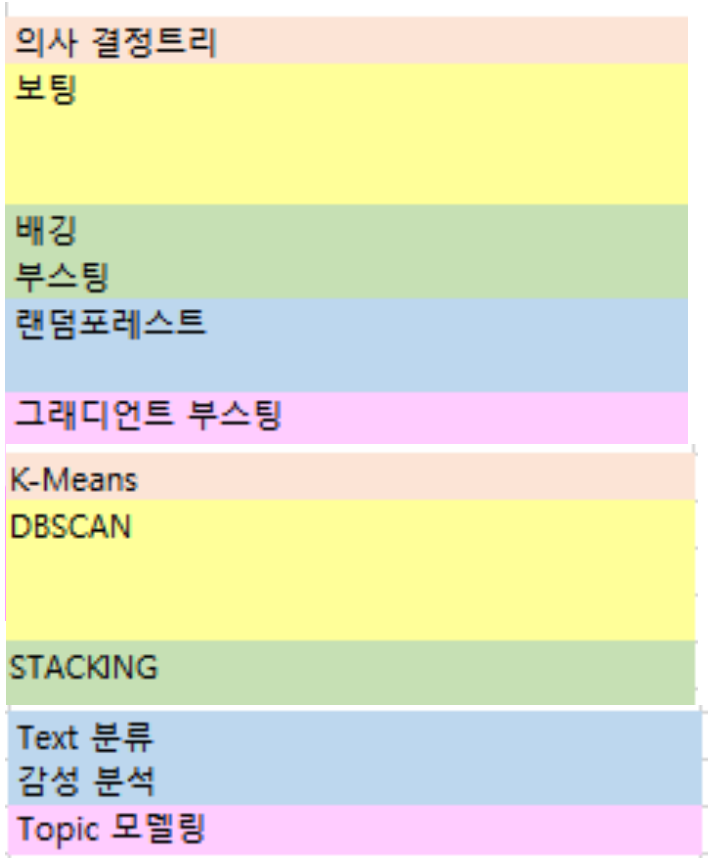
과정 소개

day	대주제	hour	관련 내용
1	0. 머신러닝의 개요	1	머신러닝 개요 머신러닝 기법 분류 사이킷런 기초 및 데이터 표현 방식 이해 사이킷런을 활용한 머신러닝 모델 만들기 훈련, 테스트 데이터 분할 iris-data 소개
	단순 선형 회귀	2	단순선형회귀 이론 설명 단순 선형 회귀 실습 -보스턴 집값(실습) 방 VS 가격
		3	맥주판매량 VS 기온 나이 VS 키
2		4	sklearn, statsmodel 비교 및 statsmodel 설명
	다중 선형회귀	5	다중 선형회귀 개념 - 회귀식 - 독립변수가 복수 : 상관분석 - 특성 스케일링
		6	보스턴 집값 다중회귀 실습 (응용)캘리포니아 집값(상관분석) 다중 회귀 (응용) 당뇨병 데이터 다중 회귀, 스케일링
	경사하강법	7	경사하강법 개념
3		8	실습) 보스톤 집값 SGDRegressor 실습(튜닝의 필요성) - 스케일링 - 하이퍼파라미터 튜닝(max_iter, eta0, tol) - 수정된 결정 계수
	다항회귀	9	다항회귀 개념 및 Polynomial (degree 2~7) 실습
		10	실습) 보스톤 집값(room vs price) lab) 당뇨병(bmi VS 진행정도)
4	과적합, 분산편향 트레이드오프, 교차검증	11	과적합(과대적합, 과소적합)
		12	분산편향 트레이드오프
		13	교차검증 (KFold)
5	규제된 선형회귀	14	릿지, 라쏘 회귀 이론 설명 및 실습
		15	엘라스틱넷 회귀 이론 설명 및 실습
	로지스틱 회귀	16	로지스틱 회귀 이론 설명 - 시그모이드 함수
		17	붓꽃 데이터를 활용한 로지스틱 회귀 실습 (lab) 종양 분류 데이터

- **표의 일정으로 진행**
- **각 단원 마다 실습으로 복습하는 시간을 갖는다.**

- 당장 이해가 되지 않아도 스트레스 받지 말자

과정 소개



- **표의 일정으로 진행**
- **각 단원 마다 실습으로 복습하는 시간을 갖는다.**

- 당장 이해가 되지 않아도 스트레스 받지 말자

목차

- **머신러닝의 개요**
  - 머신러닝의 의의, 사례, 절차, 학습방법의 구분
  - 머신러닝 라이브러리 소개 – Scikit-Learn, Tensorflow, Pytorch
- **회귀**
  - 선형회귀
  - 경사하강법
  - 다항회귀
  - 머신러닝체크포인트 – 과적합, 분산 편향 트레이드 오프, 교차 검증, 매개변수 스코어링
  - 규제가 있는 선형 모델 – 라쏘회귀, 릿지 회귀, 엘라스틱 회귀
  - 로지스틱회귀

01 머신러닝이해(1)

- 머신러닝 개요
- 머신러닝 기법 분류
- 사이킷런 기초
- Iris 데이터 소개

02 단순선형회귀 다중선형회귀

- 단순선형회귀 이론 설명/실습/ 결과 해석
- 다중선형회귀 이론 설명 / 실습 / 해석
- 보스턴 집값 예측
- 맥주 판매량 예측
- 당뇨병 진행도 예측

03 경사하강법

- 경사하강법 원리
- SGDRegressor 실습
- Scaling
- Hyper parameter tuning

04 다항회귀

- PolynomialRegressor 이론/실습

05 머신러닝 이해(2)

- 과적합
- 분산/편향
- 교차 검증

06 규제된 선형회귀

- Rigde, Lasso, Elasticnet 회귀

07 로지스틱회귀

- 로지스틱 회귀 이론
- 붓꽃 분류 실습

0. 머신러닝의 이해(1)

- 인공지능 개요
- 머신러닝 의의
- 머신러닝 사례
- 학습방법 구분
- 머신러닝 개발 환경

머신 러닝의 의의

- 머신 러닝의 의의
- 머신 러닝 사례

인공지능Artificial Intelligence

[사례]

바둑, 장기, 체스에서 인간을 이기는 인공지능 프로그램

병리 영상 속에서 정확히 암세포를 발견

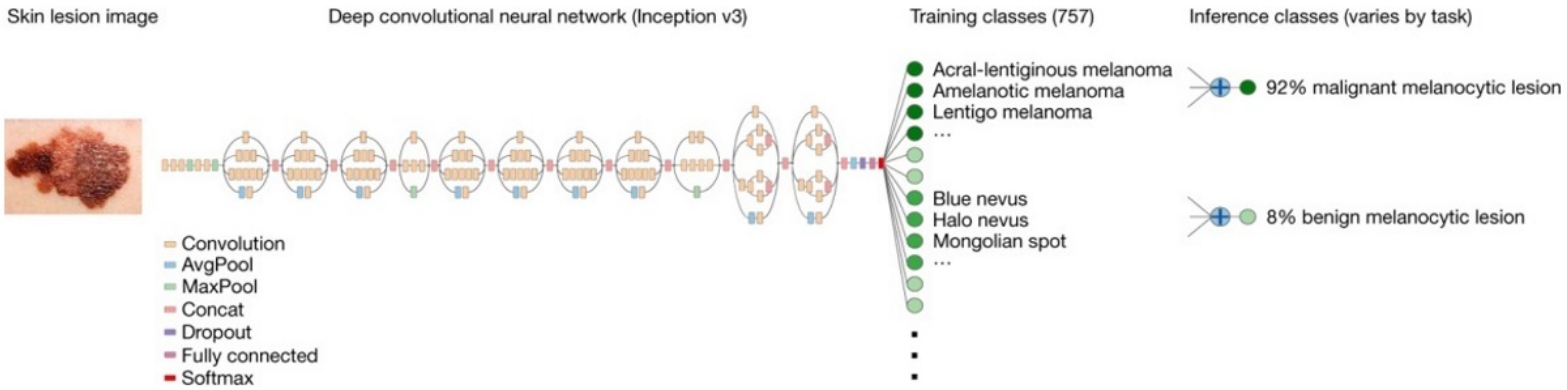
인간과 자연스럽게 대화하는 AI 챗봇

[정의]

인공 지능의 기본은 통계학과 확률론에 근거하여 **판정**과 **분류**를 하는 컴퓨터 프로그램

[특징]

대량의 데이터를 **학습**해 판정과 분류의 정밀도가 높아져(최적화) 점점 지능이 높아짐



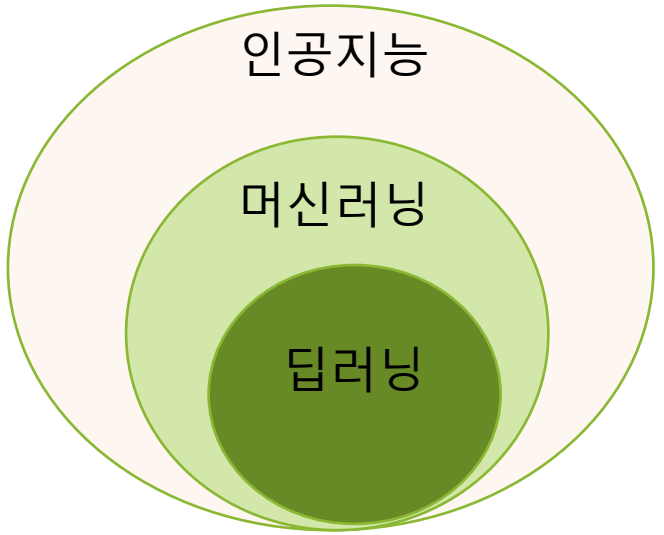
<상단 그림출처:
https://biz.chosun.com/site/data/html_dir/2016/03/02/2016030200243.html>
<하단 그림출처:
https://www.nature.com/articles/nature21056#Sec16

인공지능 artificial intelligence
vs 머신러닝 machine learning
vs 딥러닝 deep learning

인공지능: 지적 작업에 필요한 능력(지능)을 사람이 아닌 기계가 가질 수 있도록 하는 것
인간의 지적 능력을 컴퓨터를 통해 구현하는 기술의 총칭

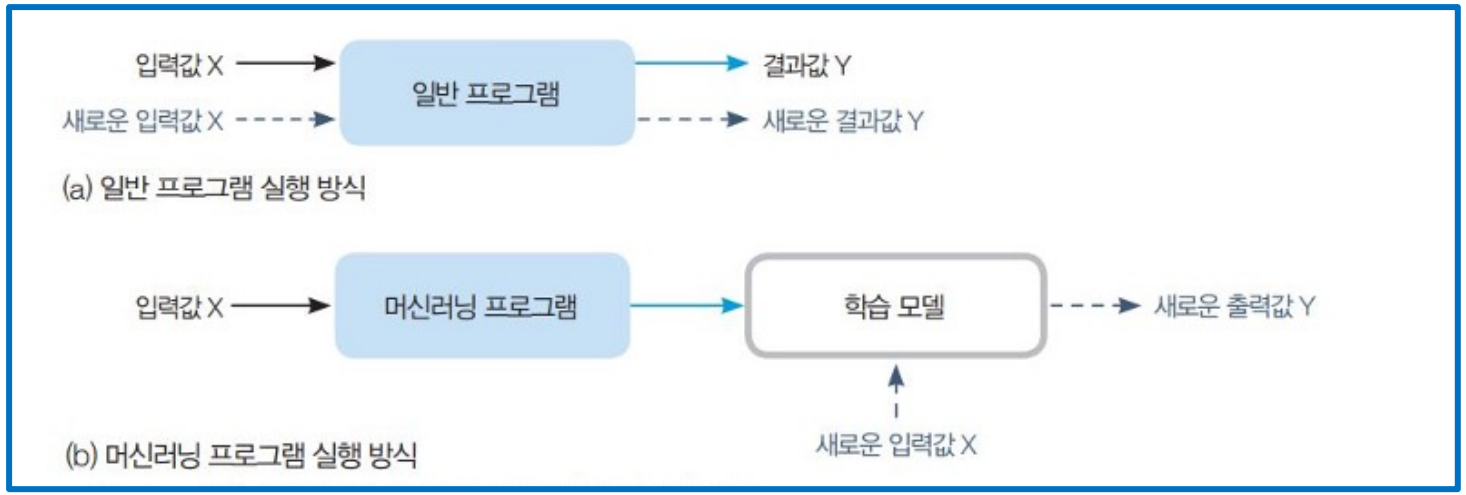
머신러닝: 주어진 데이터를 기반으로 학습하여 패턴을 파악하고 이를 이용해 예측하거나 분류를 수행하는 것

딥러닝: 머신러닝 알고리즘 중에 인공 신경망 구조를 기반으로 학습하는 방법



머신러닝Machine Learning

- 의의 :
  - 머신러닝(기계학습)은 ”기계가 사람이 일일이 코드로 명시하지 않은 동작을 데이터로부터 학습하여 실행할 수 있도록 하는 알고리즘을 개발하는 연구 분야 " (아서 새뮤얼Arthur Sammuel, 1959)
  - 어떤 작업 T에 대한 컴퓨터 프로그램의 성능을 P로 측정했을 때 경험 E로 인해 성능이 향상되었다면 이 컴퓨터 프로그램은 작업T와 성능 측정 P에 대해 경험 E로 학습한 것이다(톰 미첼Tom Michell, 1997)
- 일반 프로그램과 머신러닝 프로그램의 실행 방식 비교



<출처: 데이터과학기반의 파이썬 빅데이터 분석, p307, 한빛아카데미>

머신러닝 > 의의

- **데이터로부터 학습하도록 컴퓨터를 프로그래밍하는 분야**
- **과거 경험에서 학습을 통해 얻은 지식을 미래의 결정에 이용하는 컴퓨터 과학의 한 분야**
- **관측된 패턴을 일반화하거나 주어진 샘플을 통해 새로운 규칙을 생성하는 목표를 가짐.**

머신러닝 응용 사례

- **풀이 방법을 적시하여 컴퓨터가 할 일을 나열하기 어려운 분야**

<출처: '파이썬으로 배우는 머신러닝의 교과서', p20, 한빛미디어>

1. 이미지 분류 작업: 생산 라인에서 제품 이미지를 분석해 자동으로 분류
2. 텍스트 분류(자연어 처리): 자동으로 뉴스 기사 분류
3. 텍스트 분류: 토론 포럼에서 부정적인 코멘트를 자동으로 구분
4. 텍스트 요약: 긴 문서를 자동으로 요약
5. 자연어 이해 : 챗봇(chatbot) 또는 개인 비서 만들기
6. 회귀 분석: 회사의 내년도 수익을 예측하기
7. 음성 인식: 음성 명령에 반응하는 앱
8. 이상치 탐지: 신용 카드 부정 거래 감지
9. 군집 작업: 구매 이력을 기반으로 고객을 나누고 각 집합마다 다른 마케팅 전략을 계획
10. 데이터 시각화: 고차원의 복잡한 데이터셋을 명확하고 의미 있는 그래프로 표현하기
11. 추천 시스템: 과거 구매 이력을 기반으로 고객이 관심을 가질 수 있는 상품 추천하기
12. 강화 학습: 지능형 게임 봇(bot) 만들기 - 알파고

머신 러닝 학습 방법
- 지도학습 VS 비지도 학습(자율학습)

머신러닝 구분 > 지도학습 VS 비지도학습

- **학습의 의미:**

– 선을 긋는 것, 깔끔하게 분류할 수 있는 선을 긋거나, 설명할 수 있는 선을 긋는 것

– 학습에 의해 선을 그음으로써 미지의 데이터에 대한 예측이 가능, 즉 기존의 학습데이터에 나타나지 않은 상황까지도 일반화하여 처리할 수 있게 됨.

- **지도학습Supervised Learning:**

– 지도 학습이란 정답(레이블, label)에 근거한 학습 데이터에서 모델을 학습하여 데이터를 설명하는 하나의 함수(가설)를 유추하고 학습하지 않은 미래 데이터에 대해 예측을 만드는 머신러닝의 한 방법

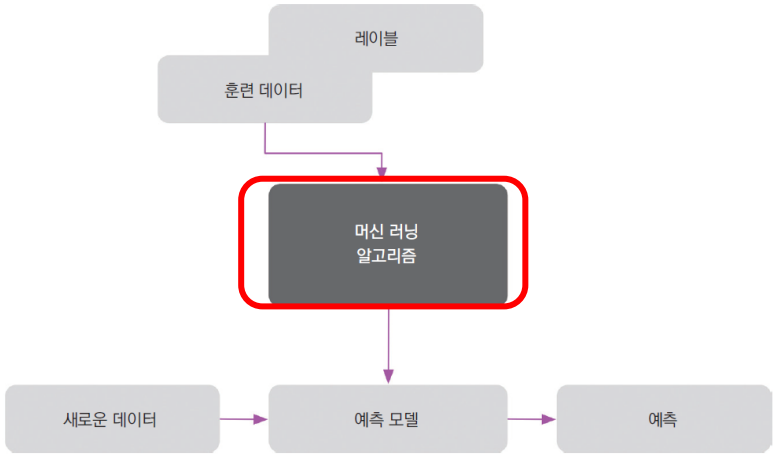
– 학습 데이터는 입력 객체에 대한 속성을 벡터 형태로 포함하고 있으며, 각각의 벡터에 대해 원하는 결과(종속값, 정답, 레이블)가 무엇인지도 포함되어 있음. 학습데이터가 입력, 출력(대상) 벡터 쌍으로 제공됨.

– 이렇게 유추된 함수 중 연속적인 값을 출력하는 것을 회귀 분석(Regression)이라고 하고, 이산적인 값을 출력하는 것을 분류(Classification)라고 함.(분류와 회귀의 차이 : 예측값(종속변수)의 형태(연속값, 이산값))

– **이미 정답이 있는 데이터를 컴퓨터에 학습시키는 방법.**

지도학습supervised Learning

- 머신러닝 지도학습 방식

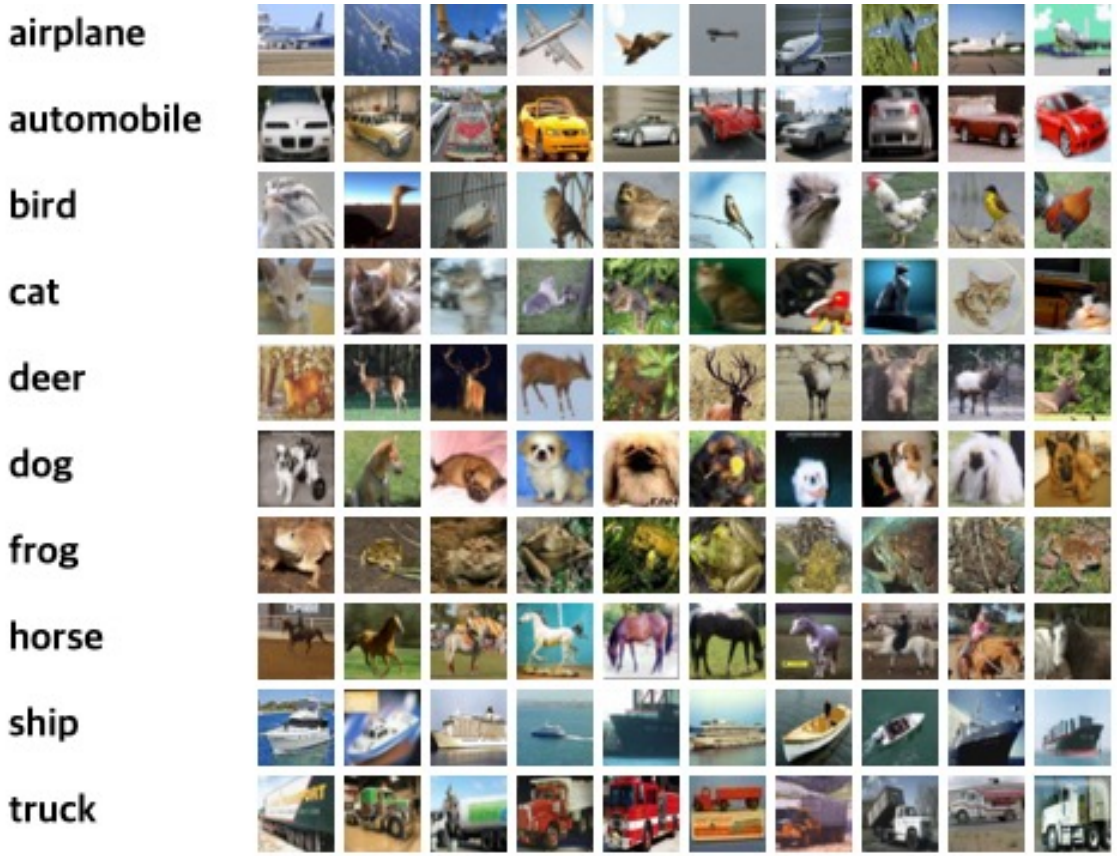


- 지도학습의 종류

1. 분류classification
   - 어떤 카테고리(종류)중 하나를 예측함
   - 예) 강아지인지 고양이인지 예측, 스팸메일 예측, 병명 진단
2. 회귀Regression
   - 연속적인 값을 예측함
   - 예) 주택의 면적을 고려하여 주택의 가격을 예측

[참고] 지도학습 데이터셋 예제(분류/회귀)

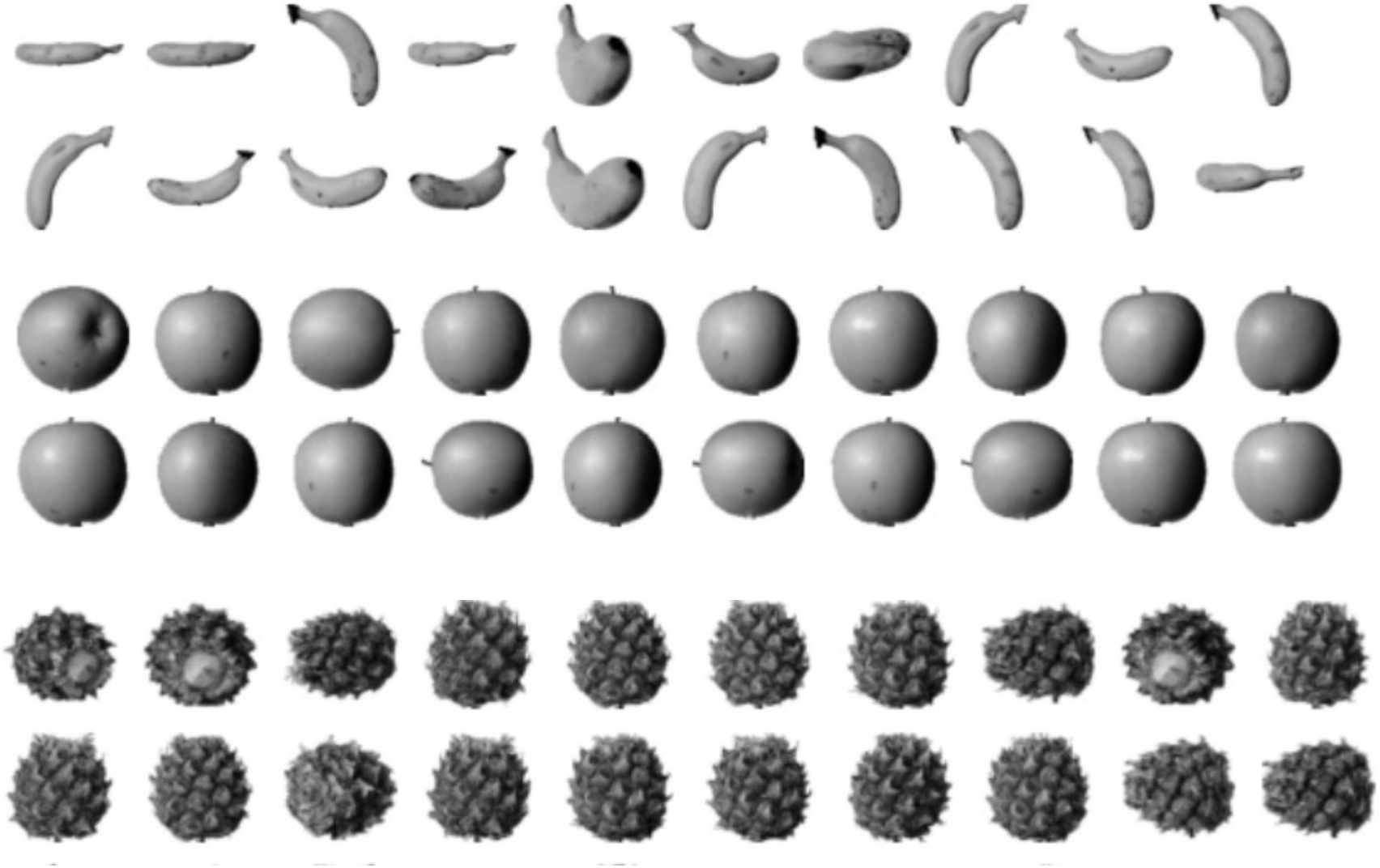
<The CIFAR-10 dataset>



<Boston Housing dataset>

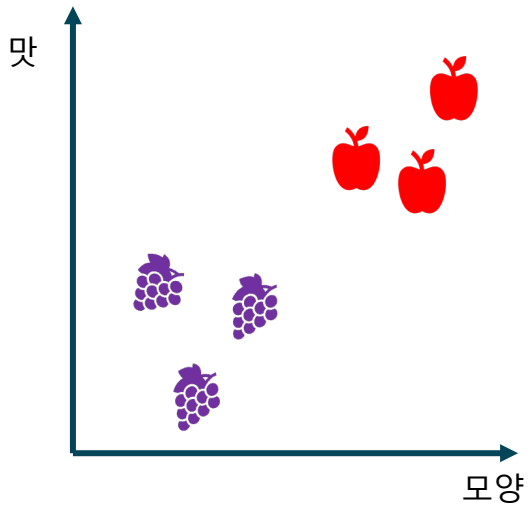
```
['housing.csv']
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  \
0  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1  296.0
1  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2  242.0
2  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2  242.0
3  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3  222.0
4  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222.0

   PTRATIO       B  LSTAT  MEDV
0     15.3  396.90   4.98  24.0
1     17.8  396.90   9.14  21.6
2     17.8  392.83   4.03  34.7
3     18.7  394.63   2.94  33.4
4     18.7  396.90   5.33  36.2
```

머신러닝(기계 학습) – 지도학습의 예



지도학습supervised Learning

- **지도학습Supervised Learning의 특징:**



<지도학습>

맛	모양	과일이름
3	1	포도
5	4	사과
1	2	포도
2	2	포도
6	6	사과
4	5	사과

<데이터>

- 데이터에 특성(맛, 모양)과 레이블(과일이름)이 모두 포함되어 있다.

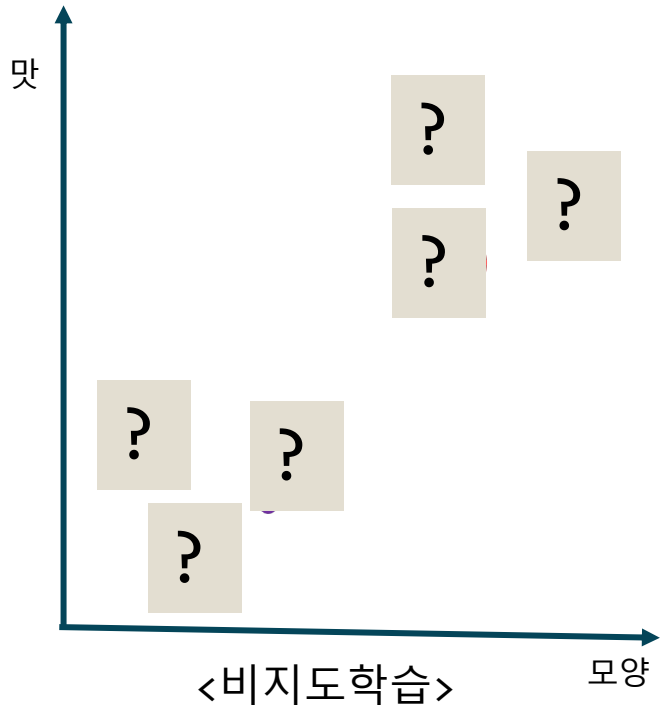
지도학습Supervised Learning

- 지도학습Supervised Learning 알고리즘 종류
  - k-최근접 이웃k-nearest neighbors
  - 선형 회귀linear Regression
  - 로지스틱 회귀logistic regression
  - 서포트 벡터 머신(SVM)support vector machine
  - 결정 트리decision tree와 랜덤 포레스트random forest
  - 신경망neural networks
- 어떤 알고리즘을 사용할 것인가?

  - 데이터나 목적에 따라 적합한 알고리즘을 찾아 사용해야 함

비지도학습Unsupervised Learning

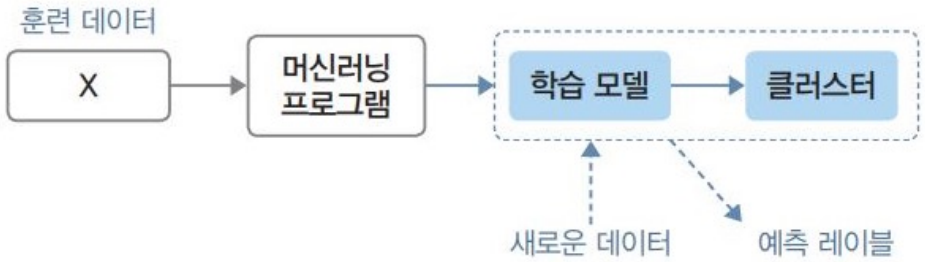
- **비지도 학습unsupervised learning:**
  - ✓ 정답 데이터를 얻을 수 없는 문제를 학습하는 것
  - ✓ 지도학습이 아닌 학습, 훈련 데이터에 레이블label이 없는 학습
  - ✓ 학습 데이터로 입력(특성 행렬)만 제공됨(정답이 없는 데이터를 사용함)
  - ✓ 수많은 사진을 보여주며 무엇이 사과인지 무엇이 포도인지 정답을 알려주지 않음.



<비지도학습>

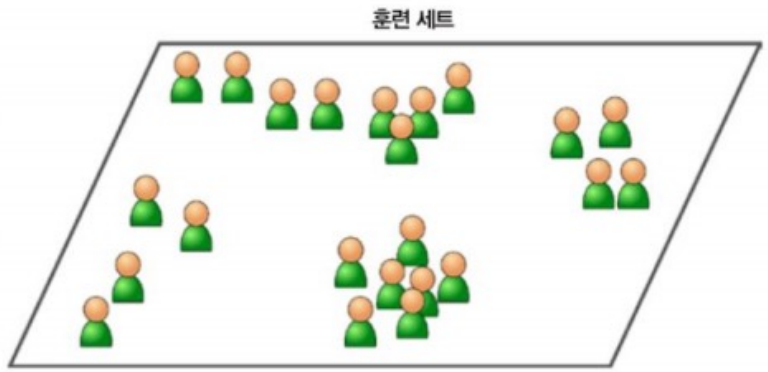
맛	모양
3	1
5	4
1	1
2	2
4	6
2	5

<데이터>



머신러닝 구분 > 지도학습 VS 비지도학습

- 비지도학습 기대 효과
  - 알려지지 않은 데이터 구조 탐색
- 비지도 학습 종류:
  1. 군집화clustering – 유사한 그룹으로 클러스터링
  2. 시각화visualizatio와 차원 축소dimensionality reduction– 하나의 관측 샘플에 있는 많은 특성(고차원)의 수를 줄임(저차원)으로써 알고리즘의 성능을 개선하거나 시각화에 도움을 줌
  3. 연관 규칙 학습association rule learning



<출처:핸즈온 머신러닝, p38, 한빛출판사>

머신러닝 관련 용어 및 배경 지식

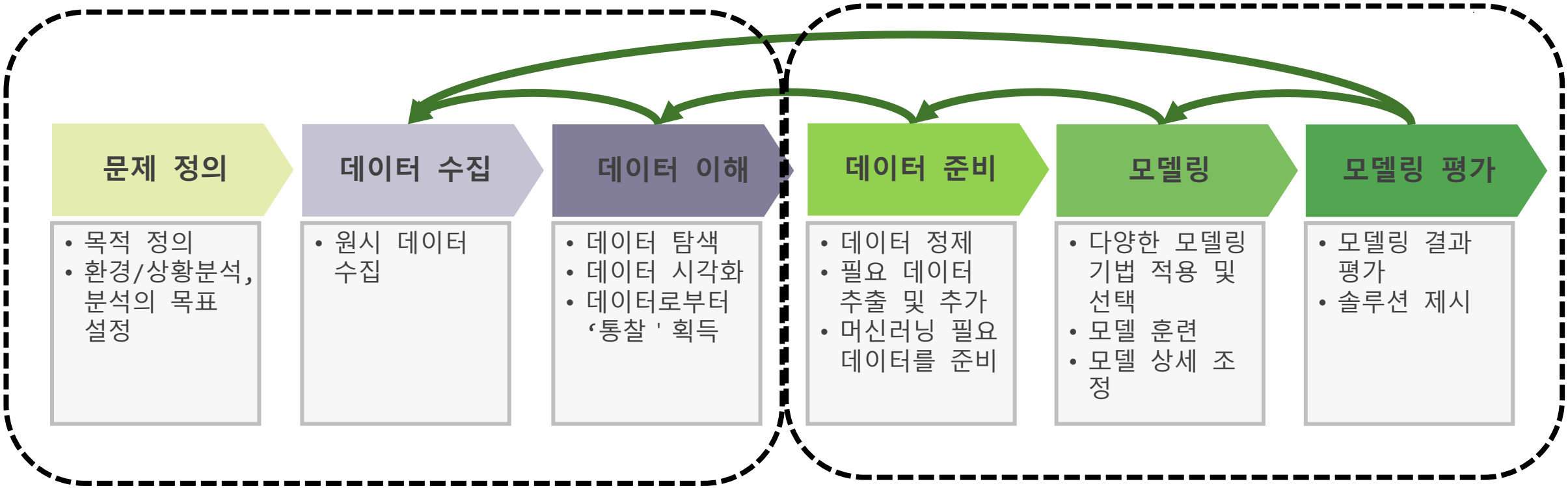
- 관련 용어
  - 모델(model) : 학습을 통해 판단하는 알고리즘을 구현한 프로그램
  - 파라미터(parameter) : 데이터에 기반한 값으로 머신 러닝 모델의 특징을 나타냄
  - 하이퍼파라미터(hyperparameter) : 주어진 데이터로부터 구하는 것이 아니라, 외부의 사용자가 직접 입력함.
  - 최적의 파라미터를 구하고, 하이퍼 파라미터 튜닝을 통해 모델의 성능을 높임.

- 배경 지식
  - 통계학
  - 선형 대수
  - 벡터
  - 미분

머신 러닝 개발 환경과 구현 프로세스

머신러닝 시스템 구축 과정

머신러닝을 이용한 시스템은 문제 정의, 데이터 이해 및 준비, 모델링, 평가를 통해 솔루션을 구현합니다.



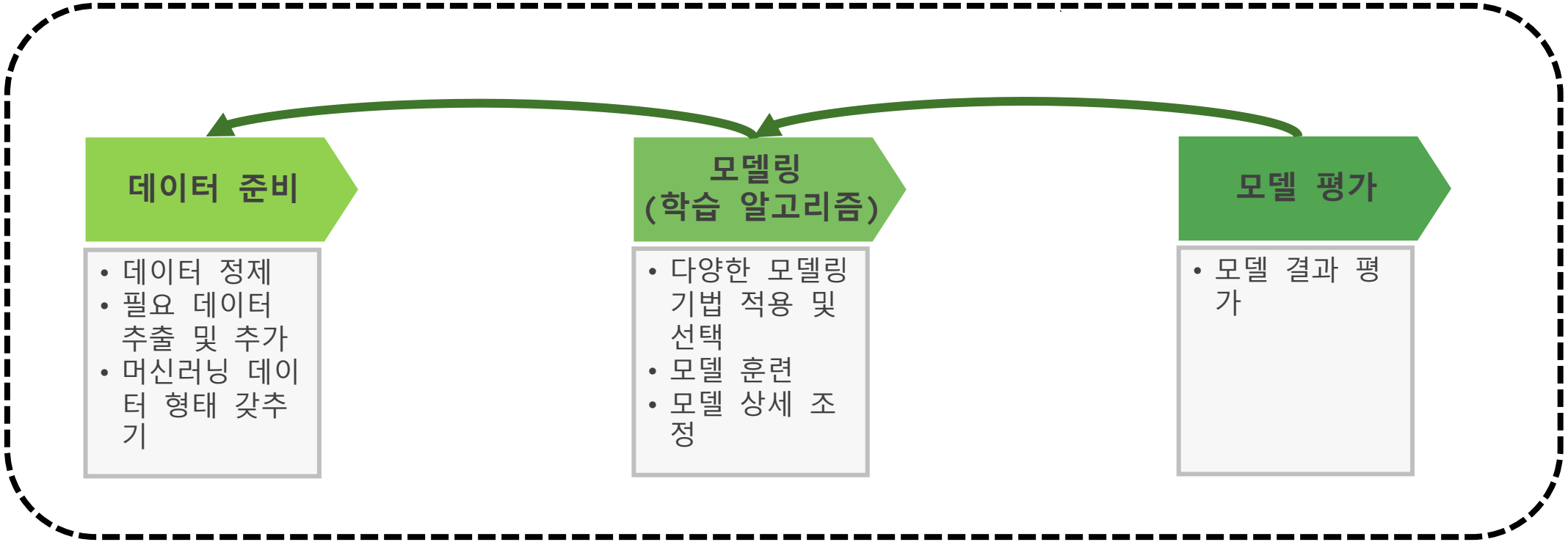
< 머신 러닝 전단계>

< 머신 러닝 단계>

머신러닝 전단계: 탐색적 데이터 분석

- **탐색적 데이터 분석**
  - 데이터 구조
  - 기초 통계 살펴보기
  - 결측값 확인
  - 시각화로 살펴보기(종속변수 -> 독립변수)
  - 단일 변수마다 살펴보기(히스토그램)
  - 변수간 상관관계
  - 단일변수 VS 타겟변수

머신러닝 프로젝트: 베이스라인Baseline 모델



① 데이터 전처리(라벨인코딩)
② 데이터 결합
③ 특성 선택 및 스케일 조정
④ 차원 축소, 샘플링
⑤ 학습데이타와 테스트데이터 분할

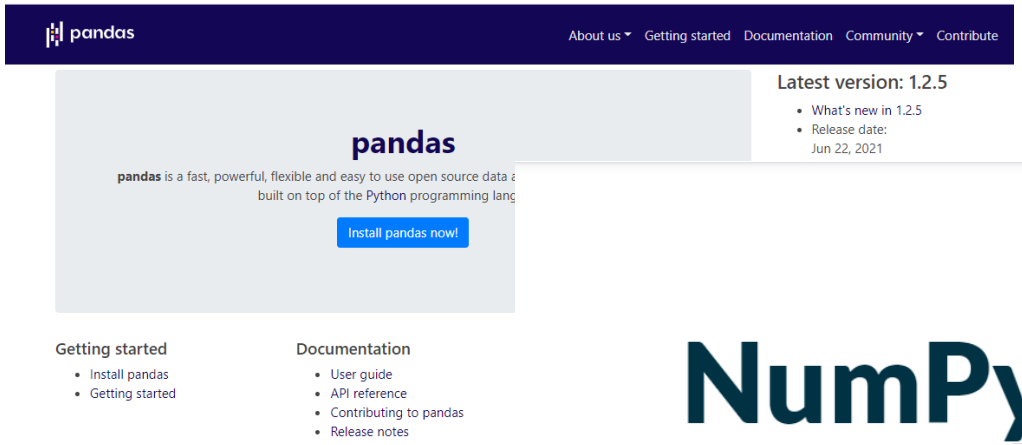
① 학습모델 정의
② 모델 학습 및 교차 검증
③ 성능 지표
④ 하이퍼파라미터 최적화

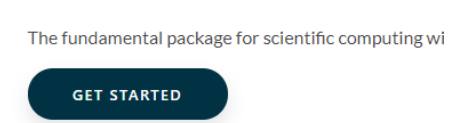
① 테스트(최종) 데이터 예측
② 고객 솔루션 제시

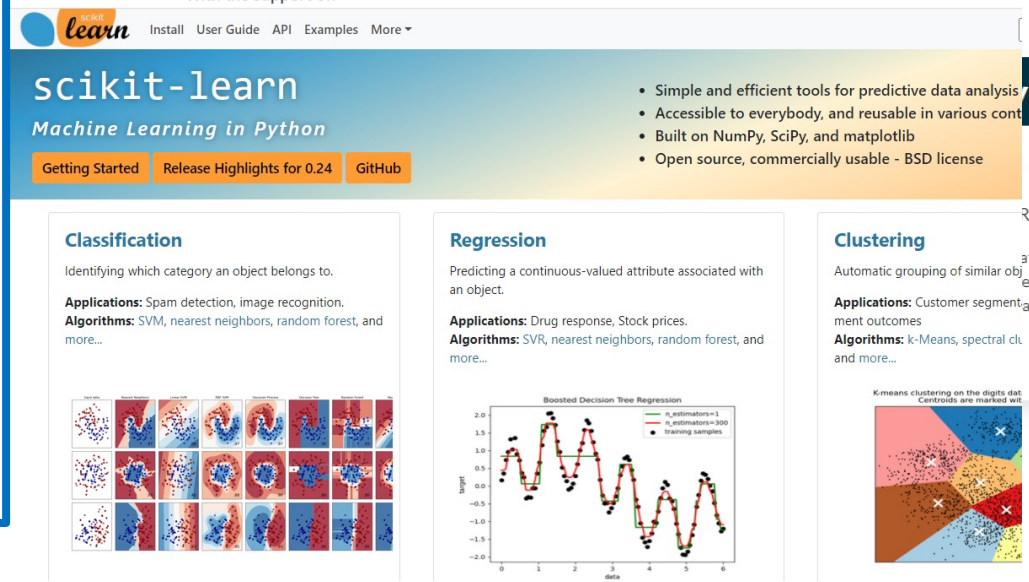
머신 러닝 개발 환경

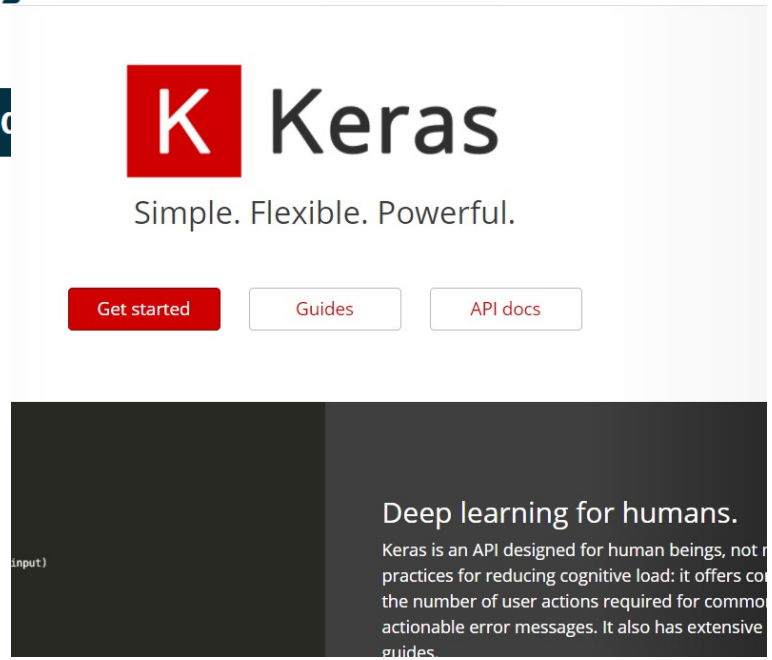
- scikit-learn
- numpy
- pandas
- scipy
- matplotlib
- seaborn

...









Scikit-Learn 기초

- 사이킷런 사용법 익히기
- 회귀, 분류와 같은 주요 머신러닝 모델을 직접 생성해보고 절차를 살펴보기
- 머신러닝 데이터를 훈련용과 테스트용으로 나누는 목적과 방법 이해하기

머신러닝 관련 라이브러리

- 사이킷런(Scikit-learn): https://scikit-learn.org
  - 머신러닝 알고리즘을 구현한 오픈소스 라이브러리 중 가장 유명한 라이브러리 중 하나
  - 회귀 분석을 비롯하여 다양하고 간단하며 배우기 쉬운 알고리즘을 제공
  - 다양한 평가 지표 제공
  - 일관되고 간결한 API가 강점이며, 문서화가 잘 되어 있음.
  - https://scikit-learn.org 에서 관련 정보 및 문서, 예제 등을 확인.
  - 아나콘다를 설치하면 자동적으로 사이킷런까지 설치되므로 별도의 추가 설치 또는 설정 없이 사용.
  - 별도로 재설치하고자 할 경우: Anaconda Prompt에서 다음 중 하나를 입력하여 설치
    - pip install scikit-learn 또는 conda install scikit-learn

https://scikit-learn.org/

사이킷런 API 활용 프로세스

데이터 -> 모델 훈련 -> 평가/예측

1.데이터 수집 및 탐색

2.데이터 전처리

3. 훈련데이터/ 테스트 데이터 분리

4. 모델 객체 생성, 학습: 훈련 데이터로 모델을 학습시킴

5. 평가, 예측: 새로운 데이터(테스트 데이터)로 예측

사이킷런 모듈 소개

분류	모듈명	설명
예제 데이터	sklearn.datasets	내장된 예제 데이터셋
데이터 전처리	sklearn.preprocessing	데이터 전처리에 필요한 다양한 기능(encoding, 정규화, 스케일링)
데이터 분리, 검증, 파라미터 튜닝	sklearn.model_selection	학습, 검증을 위한 데이터 분리, 파라미터 튜닝(Grid Search)
성능 평가	sklearn.metrics	각종 머신 러닝 알고리즘(회귀, 분류, 군집 등)별 성능 측정 함수 제공(r2_score, mean square error, accuracy, precision, etc.)
머신러닝 알고리즘	sklearn.linear_model sklearn.svm sklearn.tree sklearn.ensemble sklearn.cluster	회귀분석(선형, 릿지, 라쏘, 로지스틱 회귀 등) 서포트 벡터 머신 분류 알고리즘 의사 결정 트리 알고리즘 앙상블 알고리즘(Random Forest, AdaBoost, Gradient boosting etc) 군집화 알고리즘(K-means, DBScan etc)
지원 기능	sklearn.pipeline	피쳐 변환(transform)과 학습(fit), 예측(predict)등을 묶어 실행

Scikit-Learn의 데이터 표현방식

- **특성 행렬(Feature Matrix)**
  - 표본(sample): 데이터셋이 설명하는 개별 객체를 나타냄, 행렬의 행
  - 특성(feature) : 각 표본을 연속적인 수치, 부울값, 이산값으로 표현하는 개별 관측치를 의미
  - **학습의 결과를 결정하는 데 영향을 미치는 입력 데이터**
  - 행의 개수: n_samples
  - 열의 개수: m_features
  - 관례적으로 특성 행렬은 X에 저장
  - 2차원의 배열 구조를 사용 : 주로 Numpy 2차원 배열, Pandas DataFrame, SciPy 희소행렬을 사용
  - 독립변수, 입력변수라고도 함.

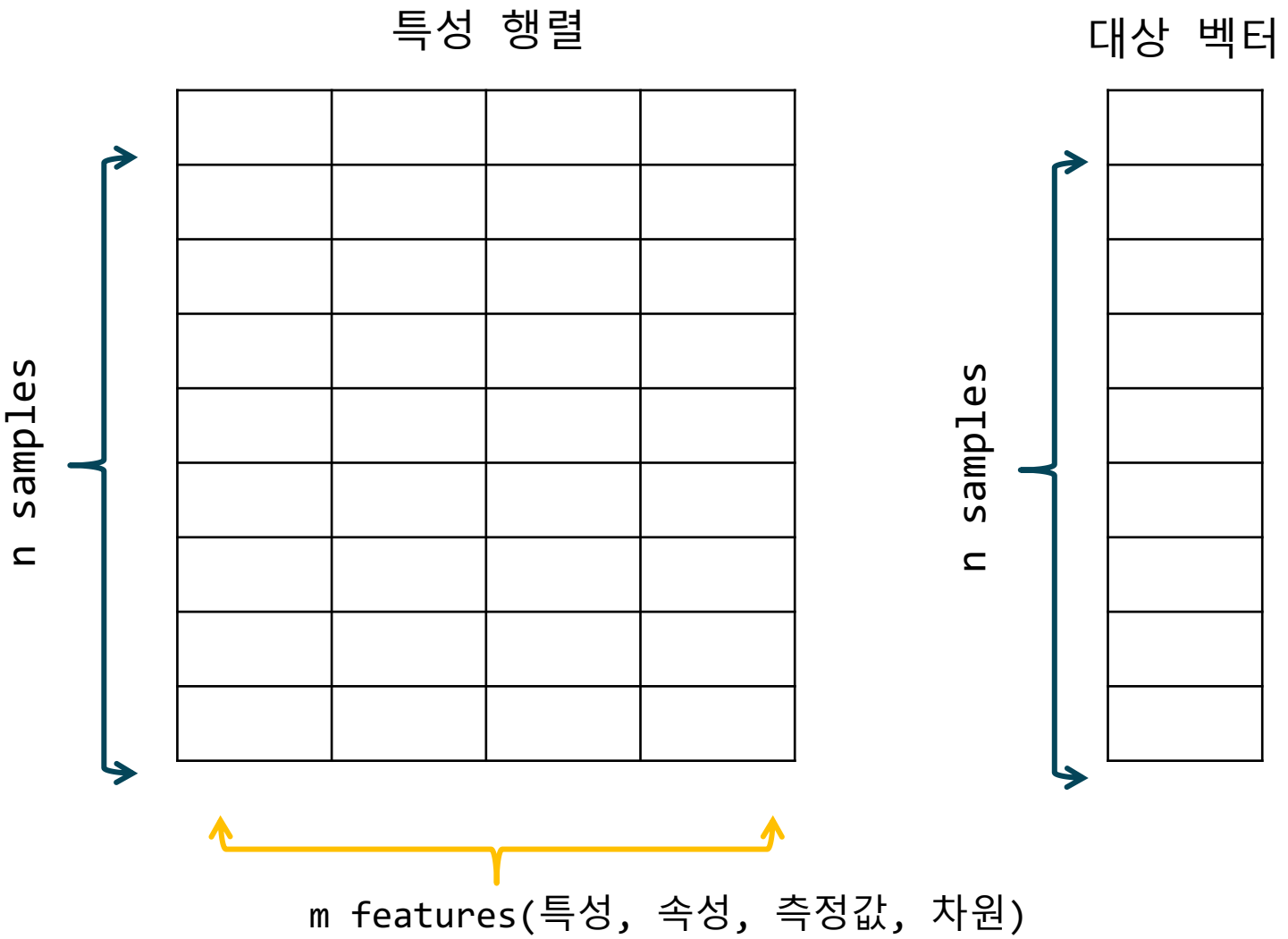
- 그렇다면, 스팸 메일을 결정하는데 영향을 미치는 특성은 어떤 것이 있을까?

Scikit-Learn의 데이터 표현방식

- **대상 벡터(Target Vector)**
  - 연속적인 수치값, 이산 클래스/레이블을 가짐
  - 길이: n_samples
  - 관례적으로 대상 벡터는 변수 y에 저장
  - 1차원 배열 구조를 주로 사용: Numpy 1차원 배열, Pandas Series
  - 특성 행렬로부터 예측하고자하는 값의 벡터
  - 종속변수, 출력변수, 결과변수, 반응변수,목표변수라고도 함.

Scikit-Learn의 데이터 이해

특성행렬과 대상벡터의 데이터 레이아웃



Scikit-Learn의 데이터 이해

- **사이킷런의 데이터셋**

종류	의미	함수 접두어
Toy dataset	크기가 작고 간단한 샘플 데이터셋	load_
Real dataset	레코드가 더 많은 실제 데이터를 다운로드하여 사용	fetch_
Generated dataset	사용자가 원하는 특성에 맞도록 데이터를 생성	make_

```
#sklearn의 toy dataset import
from sklearn.datasets import load_iris

iris = load_iris()
```

```
'fetch_lfw_people',
'fetch_olivetti_faces',
'fetch_openml',
'fetch_rcv1',
'fetch_species_distributions',
'get_data_home',
'load_boston',
'load_breast_cancer',
'load_diabetes',
'load_digits',
'load_files',
'load_iris',
'load_linnerud',
'load_sample_image',
'load_sample_images',
'load_svmlight_file',
'load_svmlight_files',
'load_wine',
'make_biclusters',
'make_blobs',
'make_checkerboard',
'make_circles',
'make_classification',
'make_friedman1',
```

Scikit-Learn 실습 – 붓꽃 품종 예측기

출처 : http://mirlab.org/의 아이리스 항목

Scikit-Learn 실습– 붓꽃 품종 예측기

- **내장 예제 데이터의 자료형 및 의미**

Key	설명	자료형
DESCR	데이터에 대한 전체적인 설명	str
data	데이터 집합의 특성(독립변수, 입력변수)값들	ndarray(2차원)
feature_names	특성(독립변수, 입력변수)들의 이름	ndarray 또는 list
target	데이터 집합의 레이블(종속변수, 출력변수) 값	ndarray
target_names	레이블(종속변수, 출력변수)들의 이름	ndarray 또는 list

```
1  type(iris)
```

```
sklearn.utils.Bunch
```

```
1  dir(iris)   # iris.keys()
```

```
['DESCR',
 'data',
 'feature_names',
 'filename',
 'frame',
 'target',
 'target_names']
```

Scikit-Learn 실습– 붓꽃 품종 예측기

- iris 데이터 탐색

```
#데이터셋 설명
print(iris.DESCR)
```

```
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

    :Number of Instances: 150 (50
in each of three classes)
    :Number of Attributes: 4 numer
ic, predictive attributes and the
```

```
# 특성 데이터 확인
print(iris.data)
```

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
```

```
#특성의 이름 확인
print(iris.feature_names)
```

```
['sepal length (cm)', 'sepal width
(cm)', 'petal length (cm)', 'petal
width (cm)']
```

```
# 타겟 데이터 확인
print(iris.target)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
# 타겟(레이블, 클래스)의 이름 확인
print(iris.target_names)
```

```
['setosa' 'versicolor' 'virginica']
```

Scikit-Learn 실습 – 붓꽃 품종 예측기

- iris 데이터 탐색

```
#pandas DataFrame으로 데이터 확인
import pandas as pd

df_X = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df_X
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...

Scikit-Learn 실습 - 데이터 분할

- 데이터 전처리 및 데이터 분할

sklearn.model_selection.train_test_split

train_test_split(arrays, test_size, random_state, shuffle)

- array: 데이터셋
- test_size: 전체 데이터 중 테스트 데이터셋 비중(0 ~ 1, 기본값: 0.25)
- random_state: 호출할 때마다 동일한 학습/테스트 데이터셋을 생성하기 위해 주는 난수값
- shuffle: 데이터를 분리하기 전에 미리 섞을지를 결정함(기본값: True)

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, test_size=0.3,₩
                                                    random_state=42)
```

학습데이터(훈련데이터): 파라미터 추정에 사용되는 데이터
테스트데이터: 일반화 오차를 평가하기 위해 학습에 사용되지 않고 남겨둔 데이터.

▪ 모델 구축 및 학습

모델링

- 다양한 모델링 기법 적용 및 선택
- 모델 훈련
- 모델 상세 조정

▪ 사이킷런의 분석 수행 절차

1. 모델 객체 생성 : 분석을 수행하기 위한 추정자(estimator) 객체를 생성한다
2. 학습 수행: 생성된 객체에 대해 학습을 수행한다(**fit()** 메소드)
3. 예측 결과 도출: 생성된 객체 또는 학습이 수행된 분석 결과에 대해 **predict()** 메소드를 호출하여 수행

▪ 모델 평가: 테스트와 검증

모델 평가

- 모델링 결과 평가
- 솔루션 제시

sklearn.metrics: Metrics

모델 평가:
생성된 객체 또는 학습이 수행된 분석 결과에 대하여 적합한 성능 평가 지표를 도출한다.

metrics.mean_absolute_error(y_true, y_pred, *)	Mean absolute error regression loss.
metrics.mean_squared_error(y_true, y_pred, *)	Mean squared error regression loss.
metrics.median_absolute_error(y_true, y_pred, *)	Median absolute error regression loss.
metrics.r2_score(y_true, y_pred, *[, ...])	R2 (coefficient of determination) regression score function.
metrics.mean_poisson_deviance(y_true, y_pred, *)	Mean Poisson deviance regression loss.

< Regression metrics >

머신러닝 프로젝트 전 과정 > 모델 평가

metrics.mean_absolute_error(y_true, y_pred, *)	Mean absolute error regression loss.
metrics.mean_squared_error(y_true, y_pred, *)	Mean squared error regression loss.
metrics.median_absolute_error(y_true, y_pred, *)	Median absolute error regression loss.
metrics.r2_score(y_true, y_pred, *[, ...])	R2 (coefficient of determination) regression score function.
metrics.mean_poisson_deviance(y_true, y_pred, *)	Mean Poisson deviance regression loss.
metrics.accuracy_score(y_true, y_pred, *)	Accuracy classification score
metrics.f1_score	
metrics.roc_auc_score	
metrics.precision_score	
metrics.recall_score	

테스트와 검증

- 모델을 평가하고 새로운 샘플에 실제로 적용해보며 필요에 따라 튜닝하는 과정
- 테스트 데이터 세트를 미리 분리시켜 놓고 학습이 끝난 모델을 검증
  - 데이터셋 크기에 따라 비율이 다름
  - 훈련(학습) 세트를 사용해 모델을 훈련하고 테스트 세트를 사용해 모델을 테스트
    - 일반화 오차(외부 샘플 오차): 시스템이 가지고 있지 않은 외부의 새로운 샘플에 대한 오류
    - 테스트 세트에서 모델을 평가함으로써 이 오차에 대한 추정값으로, 이전에 본 적이 없는 새로운 샘플에 모델이 얼마나 잘 작동할지 예측
    - 훈련 오차가 낮지만(즉, 훈련 세트에서 모델의 오차가 적음) 일반화 오차가 높다면 이는 모델이 훈련 데이터에 과대적합되었다는 뜻
- 하이퍼파라미터 튜닝과 모델 선택
  - 홀드아웃 검증
  - 교차 검증

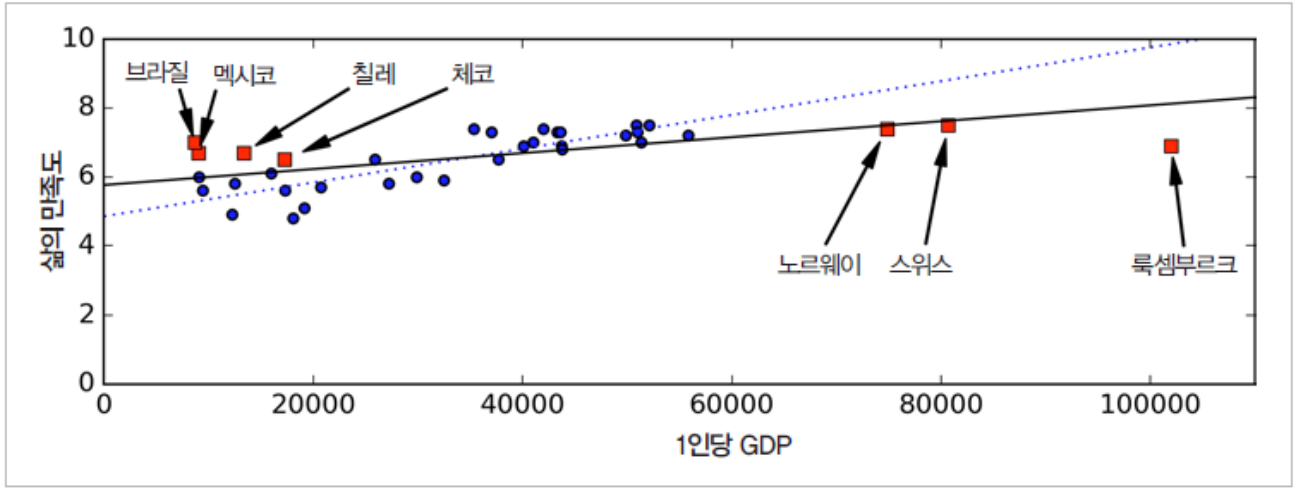
모델 튜닝

- 모델의 성능을 개선하기 위해 하이퍼파라미터를 조정해가는 과정

- GridSearchCV – 알고리즘에 사용되는 하이퍼 파라미터를 순차적으로 입력해가면서 편리하게 최적의 파라미터 조합을 찾아가는 방법 제공

머신러닝의 주요 도전 과제

- "나쁜 데이터"
  - 충분하지 않은 양의 훈련 데이터
  - 대표성 없는 훈련 데이터
  - 낮은 품질의 데이터
  - 관련 없는 특성



<출처:핸즈온 머신러닝, p56, 한빛출판사>

- "나쁜 알고리즘"
  - 훈련 데이터 과대적합
  - 훈련 데이터 과소적합

[참고] 머신러닝 관련 라이브러리

- 스탯츠모델(Statsmodels)
  - 다양한 통계 검정 및 추정, 회귀 분석, 시계열 분석 기능을 제공하는 통계 분석 라이브러리.
  - https://www.statsmodels.org 에서 관련 정보 및 문서, 예제 등을 확인.
  - 아나콘다를 설치하면 자동적으로 스탯츠모델이 설치되므로 별도 추가 설치 또는 설정 없이 사용.
  - 별도로 재설치하고자 할 경우: Anaconda Prompt에서 다음 중 하나를 입력하여 설치
    - pip install scikit-learn 또는 conda install scikit-learn
  - 다양한 옵션이 없지만 Stata 및 R과 같은 다른 통계 소프트웨어에 대해 검증된 통계 및 계량 도구를 제공
  - 다양한 선형 회귀 모델, 혼합 선형 모델, 회귀 분석이 필요한 경우 불연속 종속변수에 대한 옵션이 풍부함
  - R과 언어 형식이 비슷함.

[정리]

- 머신러닝의 학습 방법은 지도학습/비지도학습이 있으며 지도학습을 활용하는 대표적 알고리즘은 회귀와 분류이다.
- scikit-learn은 대표적인 머신러닝 라이브러리로써 2차원 구조로 특성(입력변수) 데이터(X)를 사용하고 1차원 구조로 타깃(종속 변수, 레이블) 데이터(y)를 사용한다.
- 머신러닝 프로세스:
  - 데이터 준비(특성, 타깃)
  - 모델 선택
  - 모델 훈련
  - 새로운 데이터에 대한 예측, 추론