

인공지능! 체험과 실습을 통한 이해

건양대학교 박 헌 규 교수

010-5084-8123 / ingenium@konyang.ac.kr

대학연계 참학력 공동교육과정 (23.7.25 ~ 8.2)

수업 일정

일차	날짜	차시	시간	수업내용	비고
1	7. 25 (화)	1~3 (3H)	2:30 ~5:20	<ul style="list-style-type: none"> • 오리엔테이션 • 인공지능의 정의, 역사, 종류 • 인공지능 체험 1 	구글 계정
2	7. 26 (수)				
3	7. 27 (목)	4~7 (4H)	2:00 ~5:50	<ul style="list-style-type: none"> • 인공지능, 머신러닝, 딥러닝 관계 이해 • 머신러닝의 종류 (지도학습, 비지도학습, 강화학습 사례) • 인공지능 체험 2 • 인공지능 실습환경 구축 (구글 코랩 환경 설정) 	구글 계정
4	7. 28 (금)	8~10 (3H)	2:00 ~4:50	<ul style="list-style-type: none"> • 인공지능으로 구현한 틱택토 게임 • 틱택토 게임으로 보는 인공지능 원리 학습 • 인공지능으로 구현한 오목 게임 • 인공지능 오목 게임의 원리 	구글 계정
5	7. 31 (월)	11~14 (4H)	2:00 ~5:50	<ul style="list-style-type: none"> • 인공지능 바둑 "알파고" 구현 원리 이해 • 머신러닝 지도 학습의 종류 (분류, 회귀) • 구글 코랩을 이용한 인공지능 지도학습 실습 • 학습한 모델을 통해 새로운 데이터의 예측 	구글 계정
6	8. 2 (수)	15~17 (3H)	2:00 ~4:50	<ul style="list-style-type: none"> • 내가 쓴 손글씨 숫자 인식시키기 (이미지 인식) • 구글 코랩을 이용한 MNIST 이미지 인식 실습 • 이미지를 인식하는 인공지능(CNN) 학습 	구글 계정

5일차

- 인공지능 바둑 “알파고” 구현 원리 이해
- 머신러닝 지도학습의 실습
- 구글 코랩을 이용한 인공지능 지도학습 실습
- 학습한 모델을 통한 새로운 데이터의 예측

5일차 1교시

인공지능 바둑 “알파고” 구현 원리 이해

알파고(AlphaGo) 분해

2016년 3월 9일~15일 인간과의 바둑대결에서 승리하여 인공지능의 역사를 새로 쓴 알파고에 대해 잠깐 살펴봅시다.

(이세돌 바둑 9단과 대결하여 4:1로 이긴 인공지능)

11.8 알파고: 컨볼루션 신경망과 강화 학습, 몬테카를로 트리 탐색의 결합

■ 보드 게임의 복잡도(탐색 공간의 크기)

- 너비(후보가 되는 수의 개수) b 와 깊이(끝날 때까지 두는 수의 개수) d 로 좌우

- 탐색 공간의 크기는 b^d

- 예) 틱택토($b=9, d=9 : 9^9$), 체스($b=35$ 가량, $d=80$ 가량 : 35^{80}),

바둑($b=250$ 가량, $d=150$ 가량 : 250^{150})

- 바둑의 탐색 공간은 250^{150} 으로서 인공지능이 인간을 이기는 일은 불가능하거나 수십년 걸릴 것으로 예상했는데 알파고가 2016년에 인간을 이김

11.8 알파고: 컨볼루션 신경망과 강화 학습, 몬테카를로 트리 탐색의 결합

■ 알파고

- 딥러닝과 몬테카를로 트리 탐색을 결합하여 만듦
- 딥러닝으로는 컨볼루션 신경망과 강화 학습을 사용

■ 최적 정책으로 승리하는 알파고

- 정책 $p(a|s)$: 보드 상태 s 에서 행동(두는 수) a 의 좋은 정도를 나타내는 확률
- [그림 11-13]의 경우, 알파고는 빈 곳 각각을 a 로 두고 $p(a|s)$ 를 계산한 다음, 확률이 가장 큰 a 를 취하여 둬
- 가장 좋은 수에 가장 높은 확률을 부여하는 "최적 정책"이 있다면 항상 승리
- 최적 정책을 어떻게 알아내나?

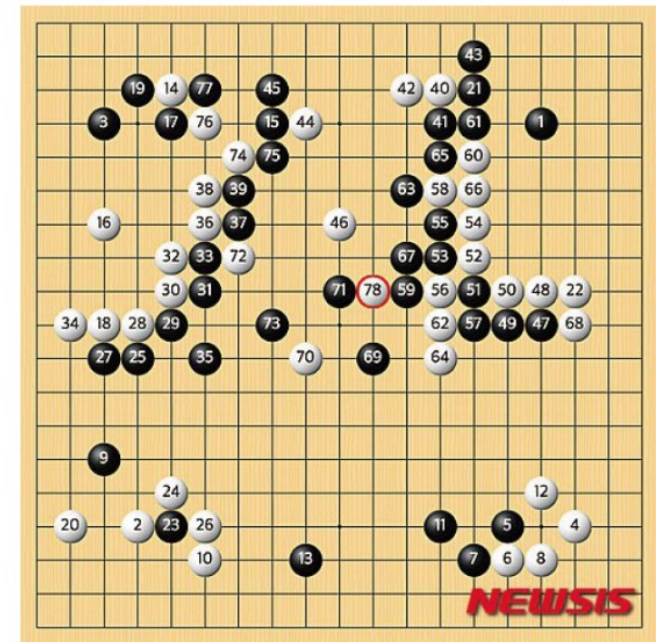


그림 11-13 이세돌이 1승을 거둔 네 번째 대국의 한 장면(이세돌이 둔 78번째 수를 신의 한 수라 부름)

11.8 알파고: 컨볼루션 신경망과 강화 학습, 몬테카를로 트리 탐색의 결합

■ 최적 정책 찾기

- 바둑은 상태가 무한에 가깝기 때문에 최적에 가까운 정책을 컨볼루션 신경망으로 근사함

■ 세 개의 정책 신경망 제작

- SL(supervised learning) 정책 신경망 p_σ
 - 프로 기사들이 축적해놓은 기보를 훈련 집합으로 활용하여 컨볼루션 신경망을 학습
 - 13개 층을 가진 컨볼루션 신경망. 출력층은 361개의 노드를 가지며 softmax 사용
 - σ 는 컨볼루션 신경망의 가중치
 - SL 정책 신경망의 한계 : 어떤 상태에서 다음 수를 예측하는 방식으로 학습되었기 때문에 성능 한계
- RL(강화 학습) 정책 신경망 p_ρ
 - 자율 학습을 통해 SL 정책 신경망을 개선
 - p_σ 의 가중치 σ 를 p_ρ 의 ρ 로 복사한 다음 자율 플레이를 통해 ρ 를 개선
 - RL 정책 신경망은 SL 정책 신경망을 80% 승률로 이김
- 정확률은 떨어지지만 매우 빠른 정책 신경망 p_π 를 별도로 학습

11.8 알파고: 컨볼루션 신경망과 강화 학습, 몬테카를로 트리 탐색의 결합

■ 한 개의 가치 신경망 제작

- RL 정책 신경망 p_ρ 에 대한 가치 신경망 $v_\theta(s)$ 를 추가로 학습
 - 신경망 구조는 정책 신경망과 비슷. 상태 s 의 좋은 정도만 출력하면 되므로 출력 노드가 하나라는 점만 다름
 - 훈련 집합은 p_ρ 의 자율 플레이에서 수집한 샘플 사용. 레이블은 상태 s 가 승리로 이어지면 1, 패배면 -1을 부여
 - 과잉 적합을 피하기 위해 한 대국에서 한 샘플만 랜덤하게 추출

■ 이제 바둑을 둘 수 있는 세 개의 신경망 확보

- SL 정책 신경망 p_σ
- RL 정책 신경망 p_ρ
- 가치 신경망 v_θ
- RL 정책 신경망이 가장 우수하는데 파치 프로그램과 겨루어 85% 승률을 달성(SL 정책 신경망은 파치에 11% 승률로 뒤짐)
- 하지만 RL 정책 신경망이 프로 기사를 이길 정도는 아님

11.8 알파고: 컨볼루션 신경망과 강화 학습, 몬테카를로 트리 탐색의 결합

- 알파고는 몬테카를로 트리 탐색을 이용하여 인간을 넘어섬
 - 순수 몬테카를로 트리 탐색을 벗어나, SL 정책 신경망 p_{σ} , 정확률은 낮지만 매우 빠른 정책 신경망 p_{π} 가치 신경망 v_{θ} 를 투입
 - 노드의 통계량으로 w/v 대신, 행동 가치 $Q(s,a)$, 방문 횟수 $N(s,a)$, 사전 확률 $P(s,a)$ 를 저장
 - 플레이아웃을 만들 때, 랜덤 샘플링 대신 p_{π} 를 이용해 샘플링
 - 단말 노드의 값은 가치 신경망 v_{θ} 의 추정값과 플레이아웃의 승패 정보를 혼합하여 계산
- 알파고의 성능
 - CPU 48개와 GPU 8개 사용한 단일 버전과 CPU 1202개와 GPU 176개 사용한 분산 버전
 - 분산 버전은 단일 버전을 77% 승률로 앞서고 기존 프로그램에 대해서는 100% 승률
- 알파고 제로
 - 프로 기사 기보를 전혀 사용하지 않고 자율 학습만 사용
 - 알파고 제로는 알파고를 100% 승률로 이김

9.1.1 다중 손잡이 밴딧 문제

■ ϵ -탐욕 알고리즘

- 탐욕 알고리즘 greedy algorithm
 - 과거와 미래를 전혀 고려하지 않고 **현재 순간의 정보만 가지고 현재 최고 유리한 선택**을 하는 알고리즘 방법론
 - 탐사형에 치우친 알고리즘
- ϵ -탐욕 알고리즘은 기본적으로 탐욕 알고리즘인데, ϵ 비율만큼 탐험을 적용하여 탐사와 탐험의 균형을 추구

■ 몬테카를로 방법

- 현실 세계의 현상 또는 수학적 현상을 **난수를 생성하여 시뮬레이션** 하는 기법
- 인공지능은 다양한 목적으로 몬테카를로 방법 활용.

5일차 2교시

머신러닝 지도학습의 종류 복습

머신러닝 - 지도학습

3일차 내용 복습

인공지능, 머신러닝, 딥러닝 맛보기



인공지능

Artificial Intelligence

사고방식이나 학습 등
인간이 가지는 지적 능력을
컴퓨터를 통해 구현하는 기술



머신러닝

Machine Learning

컴퓨터가 스스로 학습하여
인공지능의 성능을
향상 시키는 기술 방법



딥러닝

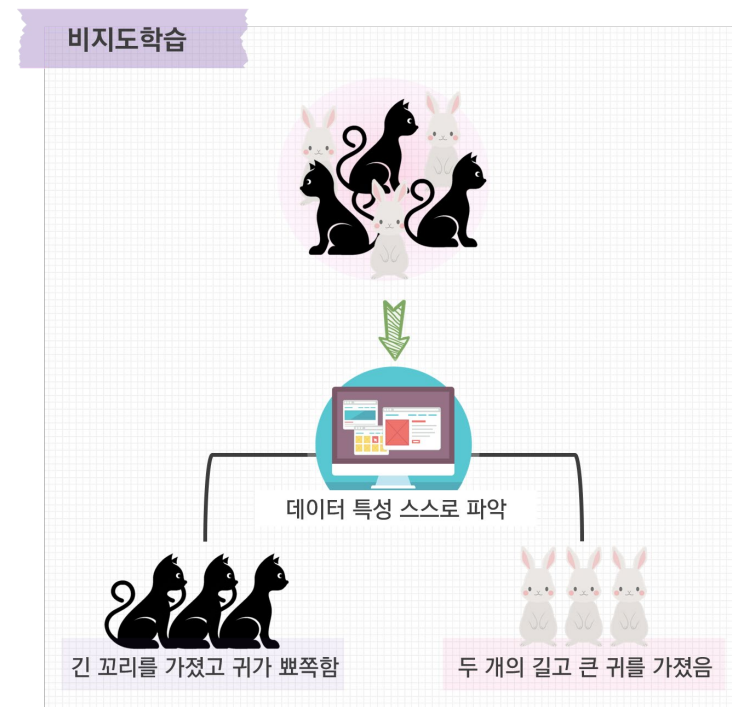
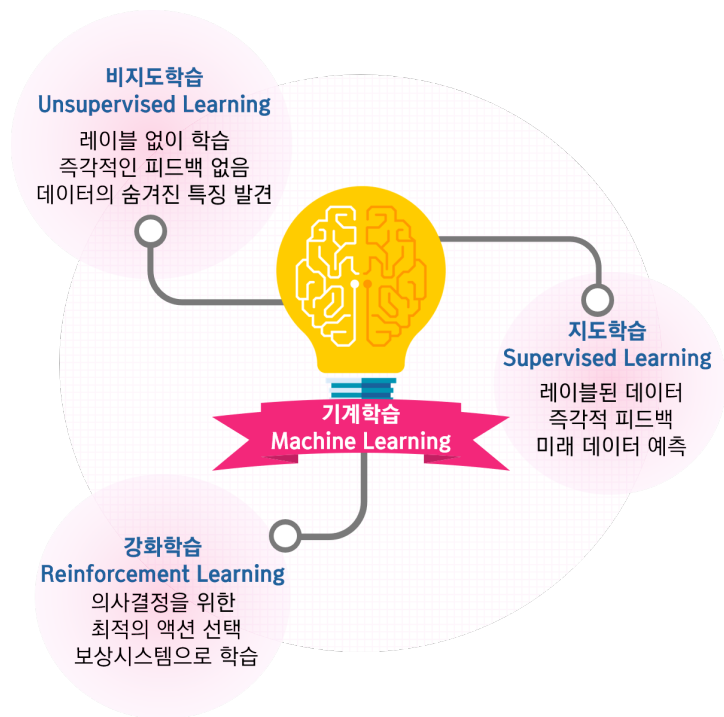
Deep Learning

인간의 뉴런과 비슷한
인공신경망 방식으로
정보를 처리

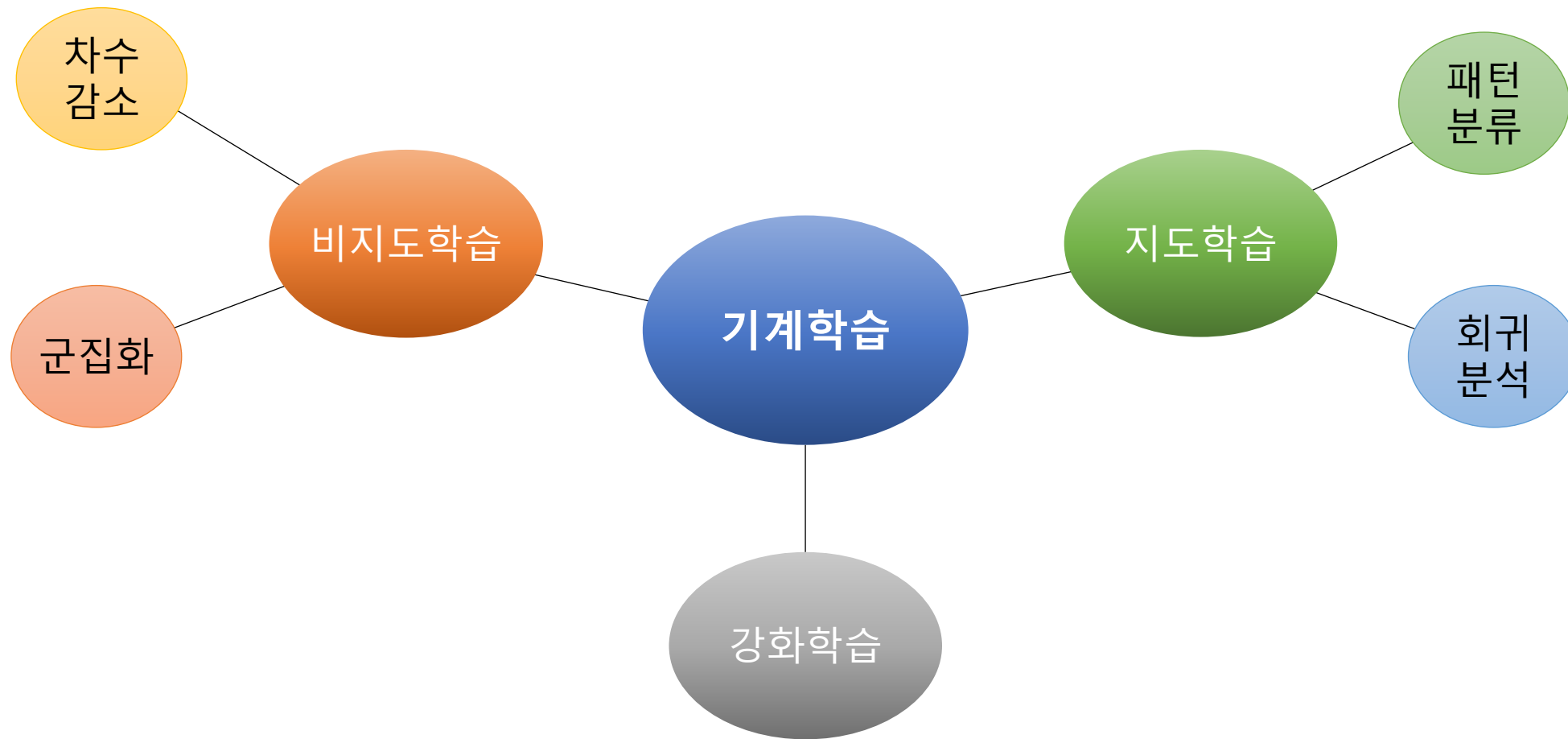


그림 출처 : <https://blogsabo.ahnlab.com/2605>

머신러닝의 종류



머신러닝 알고리즘의 분류



분류, 회귀분석, 군집

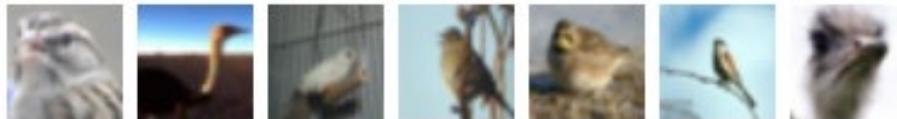
비행기



자동차



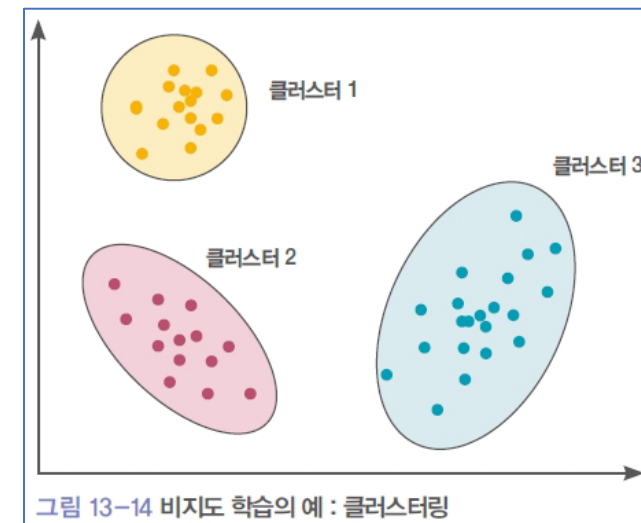
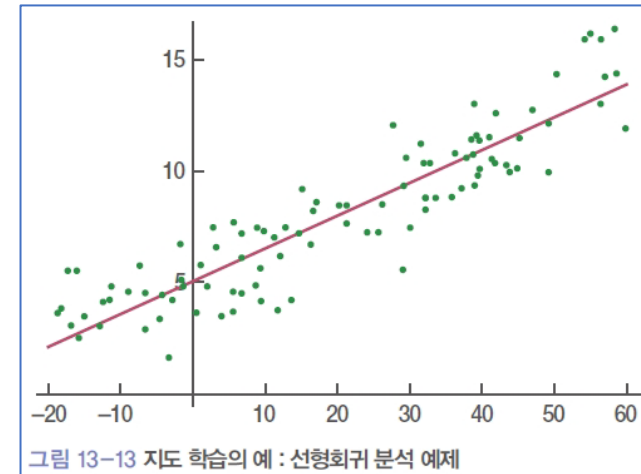
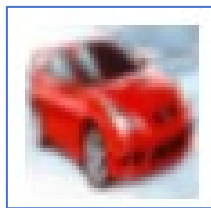
새



고양이

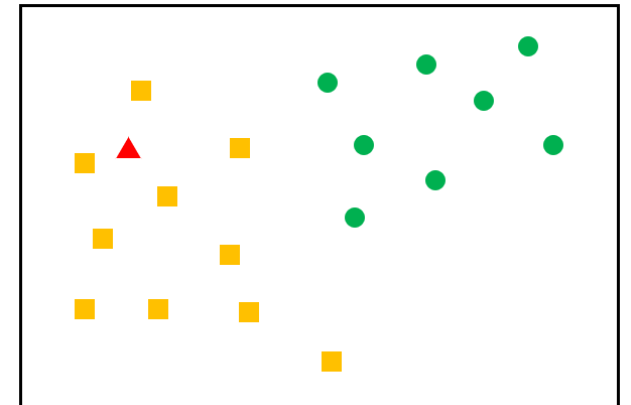


이것은 어떤
그룹으로
분류되나요?



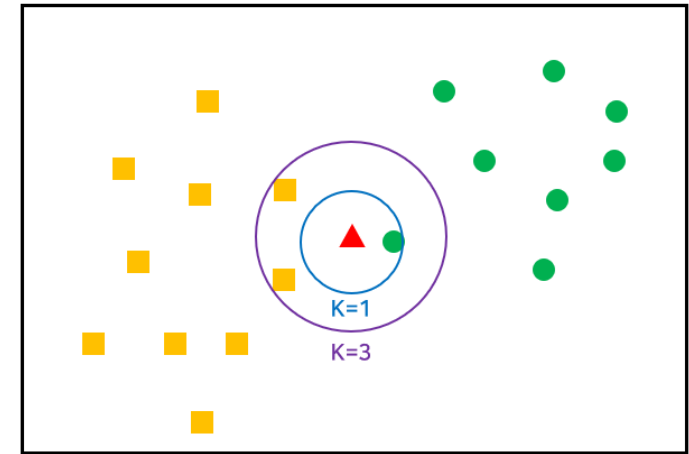
분류, 회귀, 군집화

- 분류 classification 1)
 - 입력을 두 개 이상의 유형으로 분할하고, 학습자가 한 번도 보지 못한 입력을 이들 유형 중의 하나로 분류하는 시스템
 - 예) 스팸 필터링(스팸 or 스팸 아님), 수능 등급 판별
- 분류 알고리즘
 - KNN (k-nearest neighbor) : 데이터로부터 거리가 가까운 k개의 다른 데이터의 레이블을 참조하여 분류하는 알고리즘
 - Decision Tree (의사결정트리)
 - Random Forest
 - Naive Bayes (나이브 베이즈)
 - SVM (Support Vector Machine) 등



knn 알고리즘의 문제점

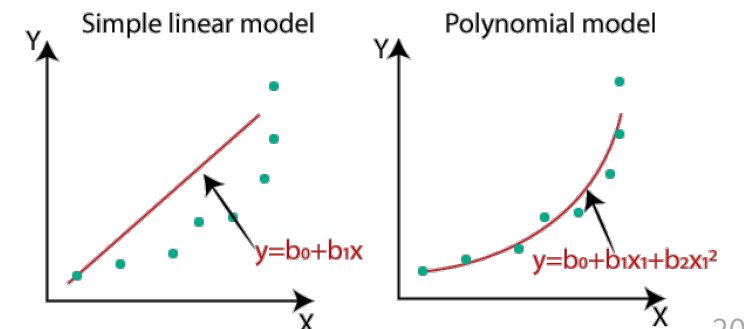
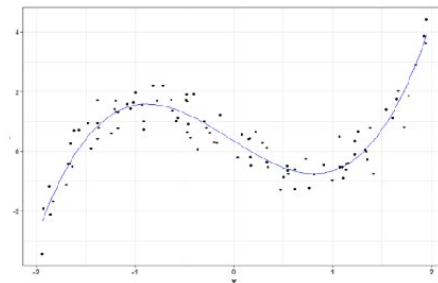
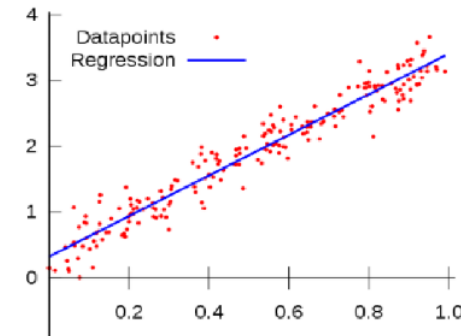
- K의 값에 따라 분류가 달라질 수 있다
 - $k=1$ 이면 초록색이지만, $k=3$ 이면 노란색
- 최선의 K를 선택하는 것은 데이터마다 다르게 접근해야한다
 - 일반적으로 총 데이터수의 제곱근 값
- 데이터가 많아지면 분류단계가 느려진다
- 모델을 생성하지 않으므로 특징과 클래스 간 관계를 이해하는데 제한적



<https://rebro.kr/183>

분류, 회귀, 군집화

- 회귀 regression ¹⁾
 - 데이터들간의 함수관계를 파악하여 통계적 추론을 하는 기술
- 회귀분석모델 종류
 - 선형 Linear, 비선형 Non-Linear
 - 단변량 Univariate, 다변량 Multivariate
 - 단순 Simple, 다중 Multiple
- 선형회귀모델 Linear regression Model
 - 단순 선형 회귀 : 독립 변수가 하나
 - 다중 선형 회귀 : 독립 변수가 둘 이상



1) <https://bangu4.tistory.com/100>

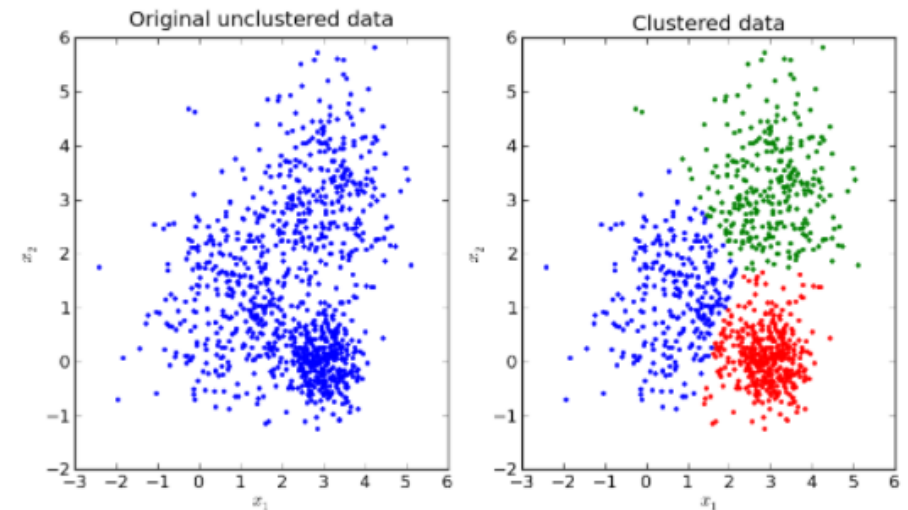
분류, 회귀, 군집화

- 군집화 clustering 1)

- 훈련용 데이터집합에서 서로 유사한 것들을 스스로 묶어서 군집을 형성하는 과정 (비지도 학습)
- 군집화를 위해서는 유사성의 판단 기준을 미리 정해야한다.
- 유사성은 데이터 간의 '거리' 를 이용하여 판단할 수 있다.
- '거리' 라는 개념은 다양할 수 있다.

- 대표적 알고리즘 : K-평균 알고리즘

1. 무작위로 K개의 중심점을 선정한다
2. 중심점과 각 데이터간의 거리를 계산한다
3. 가장 가까운 거리의 중심점에 속하도록 한다
4. k개의 클러스터의 중심점으로 재조정한다.
5. 위 2~4 과정을 반복한다



5일차 3교시

구글 코랩을 이용한 인공지능 지도학습 실습

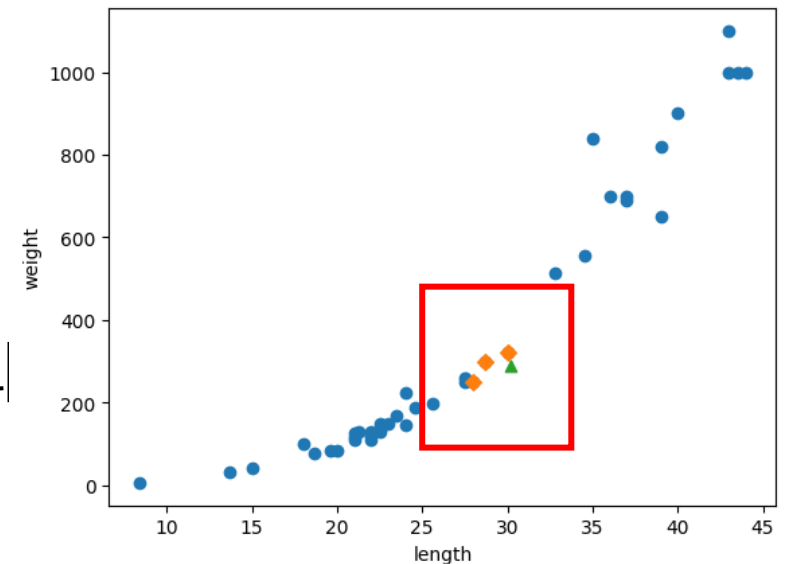
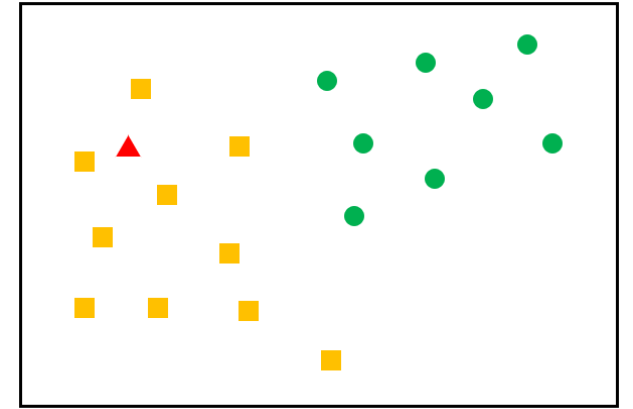
머신러닝 – 지도학습 실습

생선 무게 예측하기 version 1

- 당신은 생선 가게 사장입니다. (한 종류만 판매합니다)
- 생선을 무게 단위로 판매하려합니다.
 - 예) 200g : 2,000원, 483g : 4,830원
- 그런데 생선의 무게를 달 수 있는 저울이 없습니다.
- 하지만 생선의 길이를 잴 수 있는 줄자는 있습니다.
- 그리고 이전에 56마리의 생선의 길이와 무게를 재어 놓은 데이터가 있습니다.
 - 8.4cm : 5.9g, 13.7cm : 32.0g, 15cm : 40.0g, 16.2cm : 51.5cm ...
- 어떻게 하면 생선을 사가는 손님들이 무게에 따른 적절한 가격이라고 생각하게 할 수 있을까요?

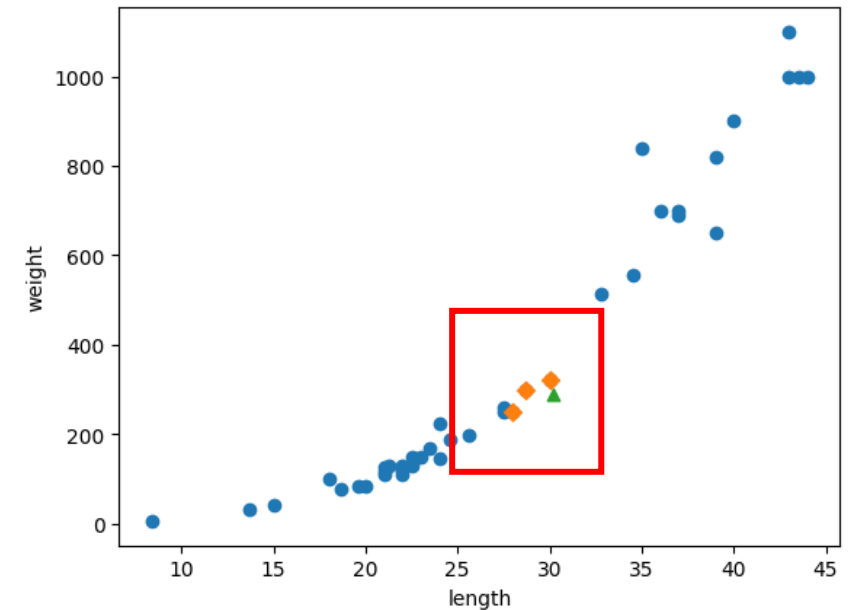
머신러닝 만들기

- 먼저 knn 방법으로 생선의 무게를 예측해봅시다.
- **KNN** (k-nearest neighbor)
 - 데이터로부터 거리가 가까운 k개의 다른 데이터의 레이블을 참조하여 분류하는 알고리즘
- 따라서, 이미 알고 있는 56개의 데이터를 이용하여 그래프를 그리고, 새로 들어오는 생선의 길이와 근접한 생선 3개의 무게를 이용하여 무게를 예측합니다.
- 예를 들어, 우측 그래프에서 녹색 삼각형의 생선의 무게를 알려면 인접한 주황색 마름모 데이터 3개의 무게를 평균내는 방법입니다.



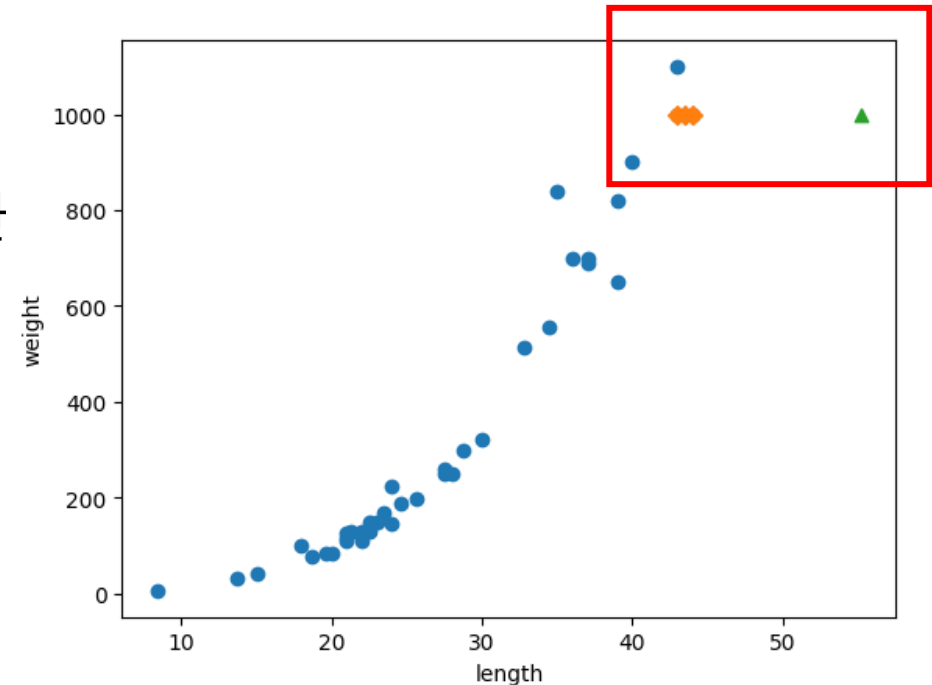
머신러닝 만들기 - knn알고리즘

- "code5 fish_weight_knn.py" 다운로드
- 메모장에서 읽어서 복사
- 구글 코랩에 붙여넣기
- 실행 버튼 or Ctrl+Enter
- 무게를 예측할 농어의 길이를 입력한다
 - 5 ~ 45 사이의 숫자
 - 예) 30.2 입력하면 농어의 무게를 290g으로 예측한다.
- 제법 잘 예측합니다.
 - 이대로 장사하면 고객의 불만은 없을 것 같습니다.



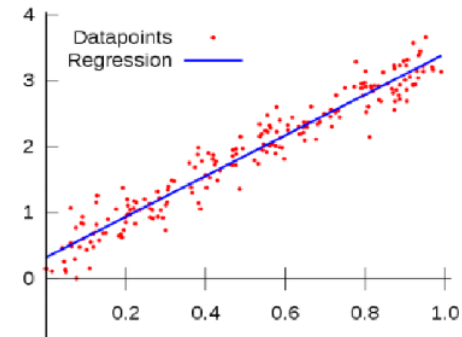
머신러닝 만들기 - knn알고리즘의 문제점

- 그런데 이상한 점을 발견하였습니다.
- 50cm 이상되는 농어의 무게가 더이상 증가하지 않는 문제점이 발견되었습니다.
- 무게를 예측할 농어의 길이를 입력한다
 - 50보다 훨씬 큰 숫자
 - 예) 55.2 입력하면 농어의 무게를 1,000g으로 예측한
- 상식적으로 1,500g이 넘어야합니다.
- 왜 이런일이 생겼을까요?
- 55.2와 가장 가까운 데이터 3개를 찾아 평균을 내다보니 그렇게 되었습니다.



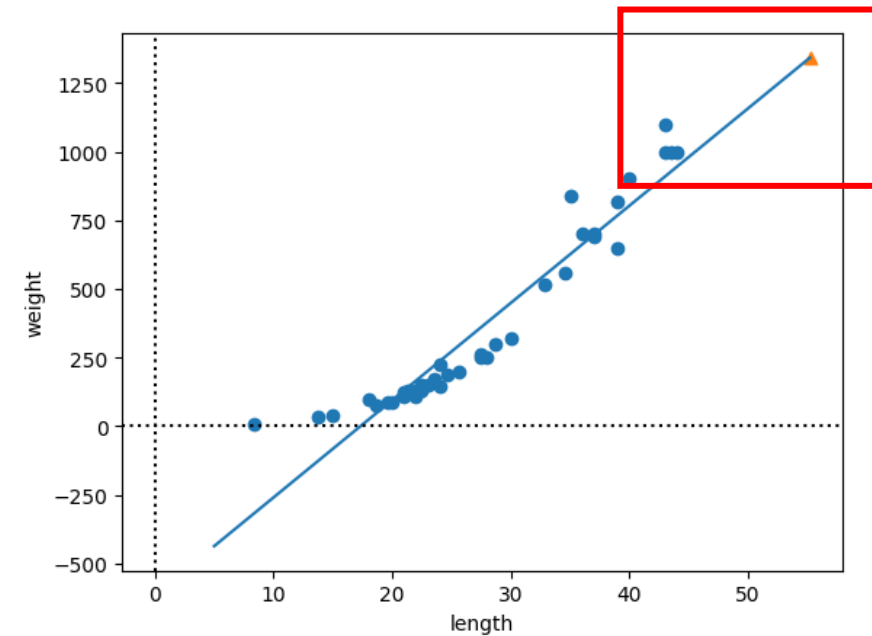
생선 무게 예측하기 version 2

- knn 알고리즘으로는 입력된 데이터 범위를 벗어나는 입력값에 대해서는 예측을 잘하지 못한다는 것이 발견되었습니다.
- 그렇다면 해결 방법은?
- 회귀분석 방법으로 해결해봅시다.
- 회귀 regression : 데이터들간의 함수관계를 파악하여 통계적 추론을 하는 기술



머신러닝 만들기 - 회귀분석

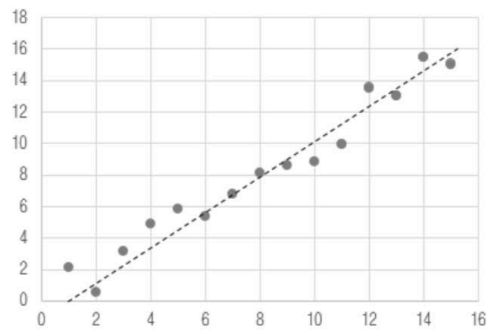
- "code6 fish_weight_linear.py" 다운로드
- 메모장에서 읽어서 복사
- 구글 코랩에 붙여넣기 (새로운 셀로)
- 실행 버튼 or Ctrl+Enter
- 무게를 예측할 농어의 길이를 입력한다
 - 5 ~ 100 사이의 숫자 가능
 - 예) 55.2 입력하면 농어의 무게를 1,342g으로 예측한다.
- 상식적인 값으로 잘 예측합니다.
- 어떻게 예측한 것일까요?



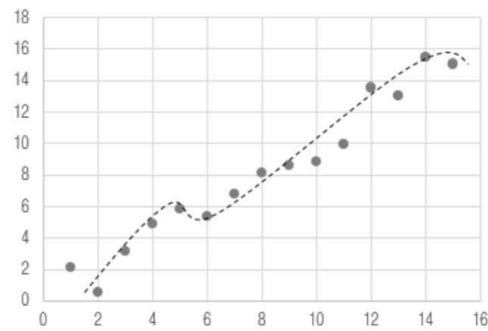
회귀분석이란?

- 회귀분석

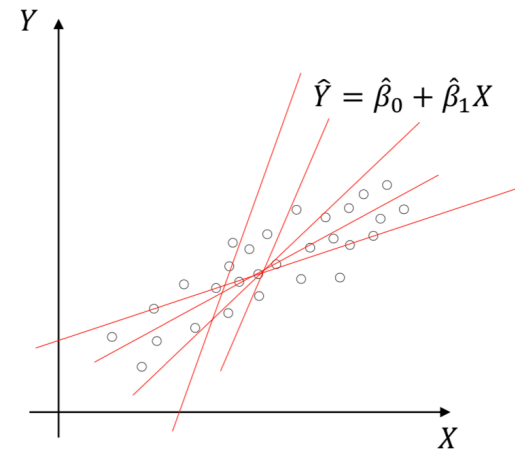
- 입력변수인 x 의 정보를 활용하여 출력 변수인 y 를 예측하는 방법
- 선형회귀분석(좌측그림)과 비선형회귀분석(우측그림) 있음



선형회귀



비선형회귀

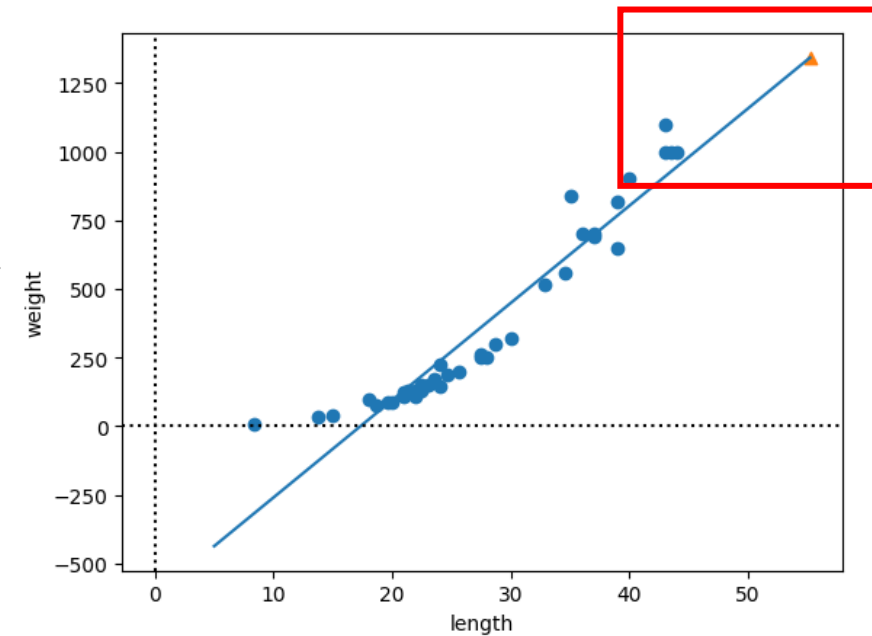


- 여러개의 직선 중 가장 좋은 직선은?

- 직선과 데이터의 차이가 평균적으로 가장 작아지는 직선
⇒ 인공지능이 이를 찾아냅니다.

머신러닝 만들기 - 회귀분석

- 주어진 데이터 56개를 이용하여 데이터간의 1차 방정식을 구했습니다.
 - $y = ax + b$
 - 기울기(a) 35.4, 절편(b) -614.0
- 즉, 생선의 무게 = $35.4 \times \text{길이} - 614$ 의 1차 방정식을 이용하여 예측하였습니다.
- 그래서 입력된 데이터의 범위를 벗어나더라도 회귀분석을 이용하여 값을 예측할 수 있었습니다.
- 그런데 그래프를 자세히 보면 뭔가 이상한 점이 몇가지 발견됩니다.

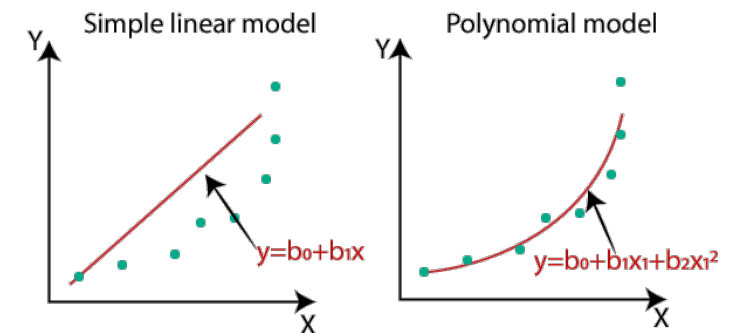
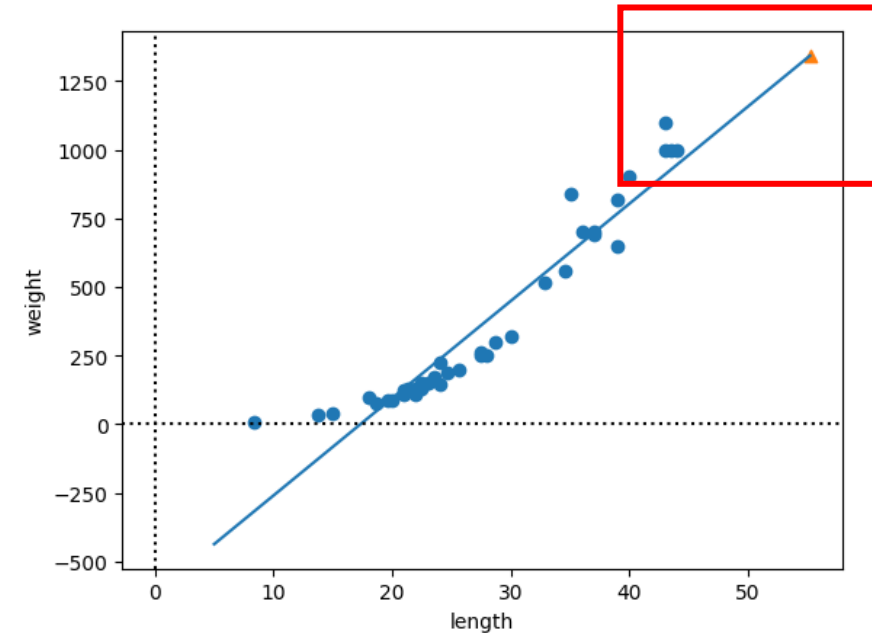


머신러닝 만들기 - 회귀분석

- 이상한 점이 몇가지 있습니다.
 1. 생선의 길이와 무게의 관계가 일차방정식이 아닌 듯합니다.
 - 약간 휘어지는 것이 2차 방정식으로보입니다.
 2. 생선의 무게가 음수(-)가 나옵니다.
 - 5g을 입력하면 -436.8g으로 예측합니다.

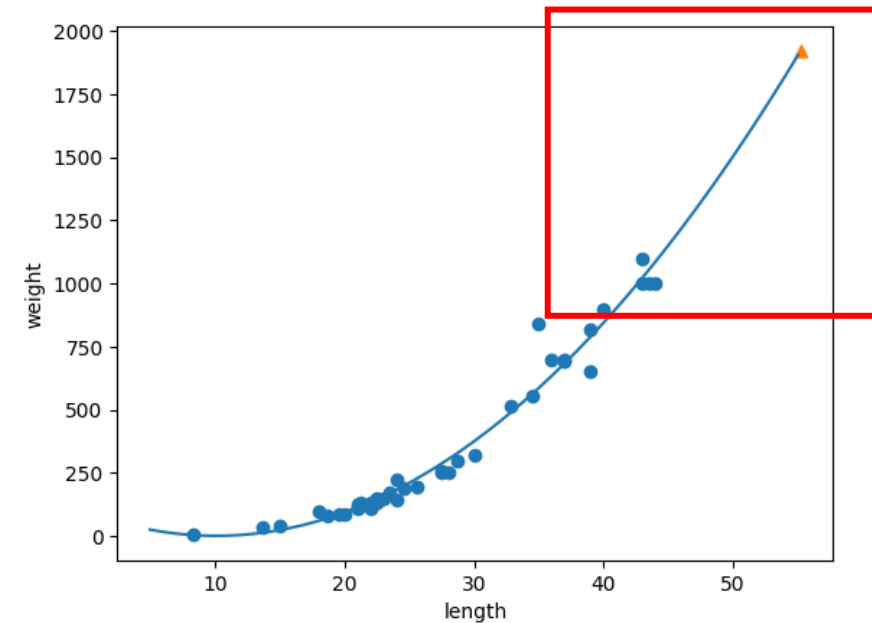
- 데이터들을 2차 선형 방정식으로 표현해야겠습니다.

$$y = ax^2 + bx + c$$



머신러닝 만들기 - 회귀분석

- "code7 fish_weight_poly.py" 다운로드
- 메모장에서 읽어서 복사
- 구글 코랩에 붙여넣기 (새로운 셀로)
- 실행 버튼 or Ctrl+Enter
- 무게를 예측할 농어의 길이를 입력한다
 - 5 ~ 100 사이의 숫자 가능
 - 예) 55.2 입력하면 농어의 무게를 1,922g으로 예측한다.
※ 1차 방정식 : 1,342g 보다 현실적
- 현실적인 값으로 잘 예측합니다.
- 데이터를 이용하여 2차 방정식 공식을 찾도록 한 결과입니다. (다항회귀방식)



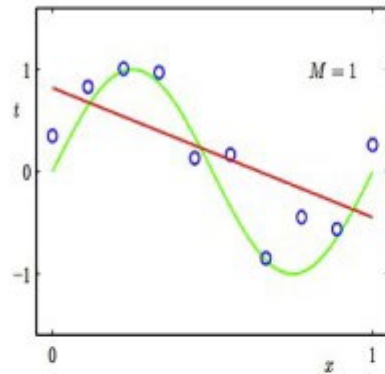
회귀분석의 문제점

- 그럼 데이터들간의 상관관계를 밝히기 위해 고차원 다항방정식을 사용하는 것이 항상 좋은 결과를 가져올까요?
- 과다적합 vs 과소적합 Overfitting vs Underfitting
 - **과다적합** : 머신러닝 모델 학습시 학습 데이터셋에 지나치게 최적화하여 발생하는 문제
 - 모델 성능은 높지만 새로운 데이터가 주어졌을 때 정확한 예측/분류를 수행못함
 - **과소적합** : 머신러닝 모델이 충분히 복잡하지 않아 (최적화가 제대로 수행되지 않아) 학습 데이터의 구조/패턴을 정확히 반영하지 못하는 문제

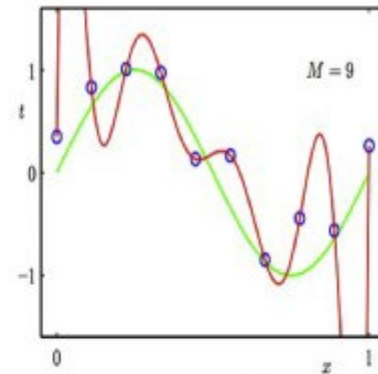
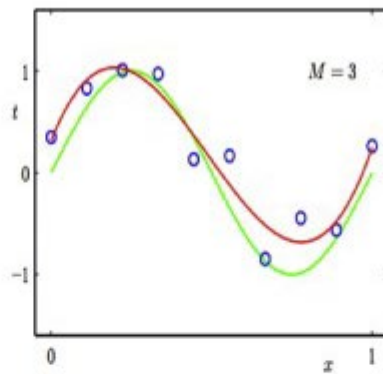
과대적합 vs 과소적합 예

- 과대적합은 머신러닝 사용 시 가장 어렵고 조심해야하는 문제

Regression:

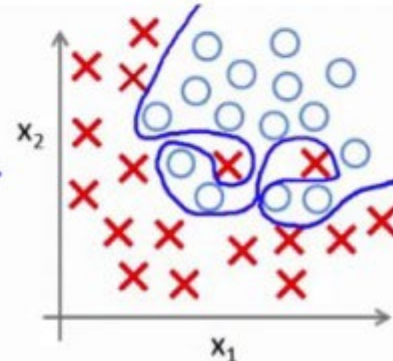
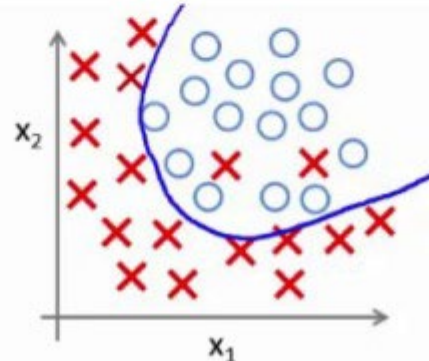
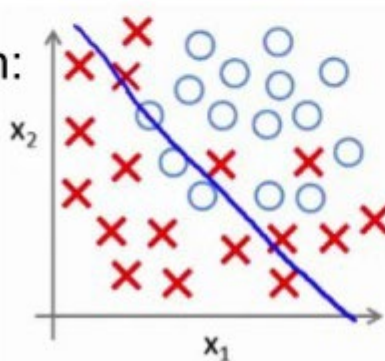


predictor too inflexible:
cannot capture pattern



predictor too flexible:
fits noise in the data

Classification:



Copyright © 2014 Victor Lavrenko

<https://m.blog.naver.com/qbxlvnf11/221324122821>

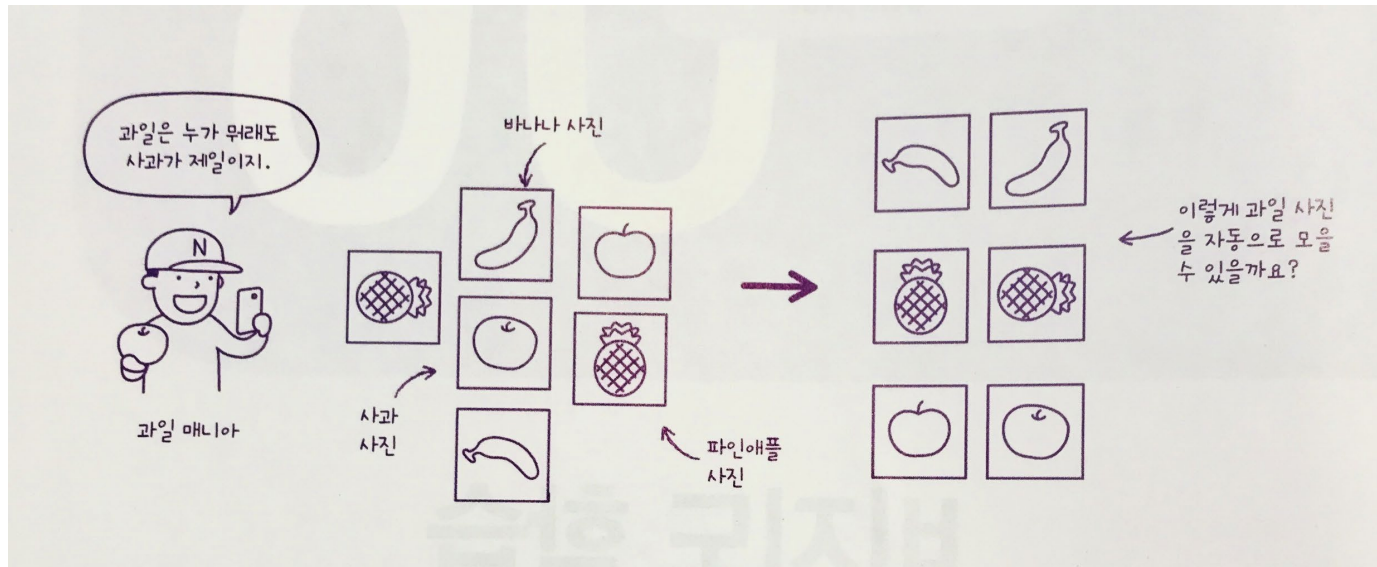
5일차 4교시

학습한 모델을 통한 새로운 데이터의 예측

머신러닝 – 비지도학습 실습

고객이 올린 사진 분류하기

- 당신은 과일 가게 사장입니다. (사과, 바나나, 파인애플 팝니다.)
- 고객이 사고 싶은 과일 사진을 올리면, 가장 많은 요청이 있는 과일을 특가에 판매하고 싶습니다.



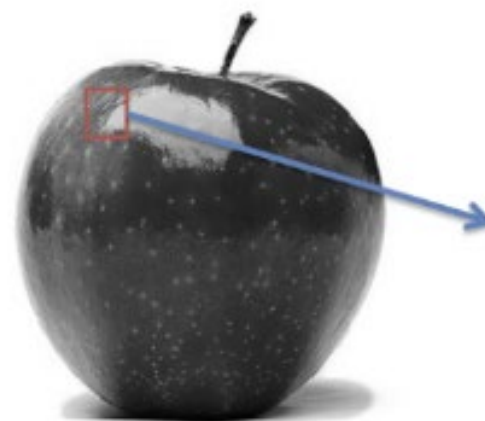
- 사람이 일일이 분류할 수도 있겠지만, 너무 번거롭습니다.
- 사람이 가르쳐 주지 않아도 데이터에 있는 무언가를 학습하게 할 수 있을까요?

사진 분류 방법?

- 어떤 방법으로 사진을 분류할 수 있을까요?
- 문제를 단순화하기 위해 흑백 사진을 다룬다고 합시다.

- 사진은 픽셀로 구성되어 있다.
- 픽셀은 0~255사이의 숫자로 이루어져 있다. (1Byte)
 - 0 : 흰색, 255 : 검정색

- 사진의 픽셀값을 평균 내면 비슷한 과일끼리 모이지 않을까?

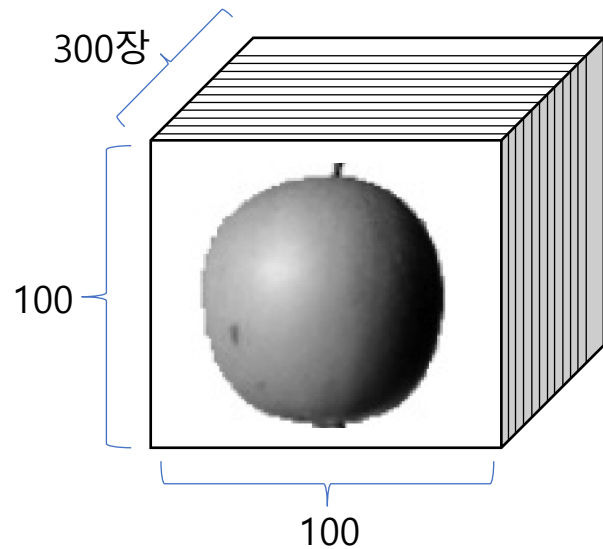


Gray-scale Image

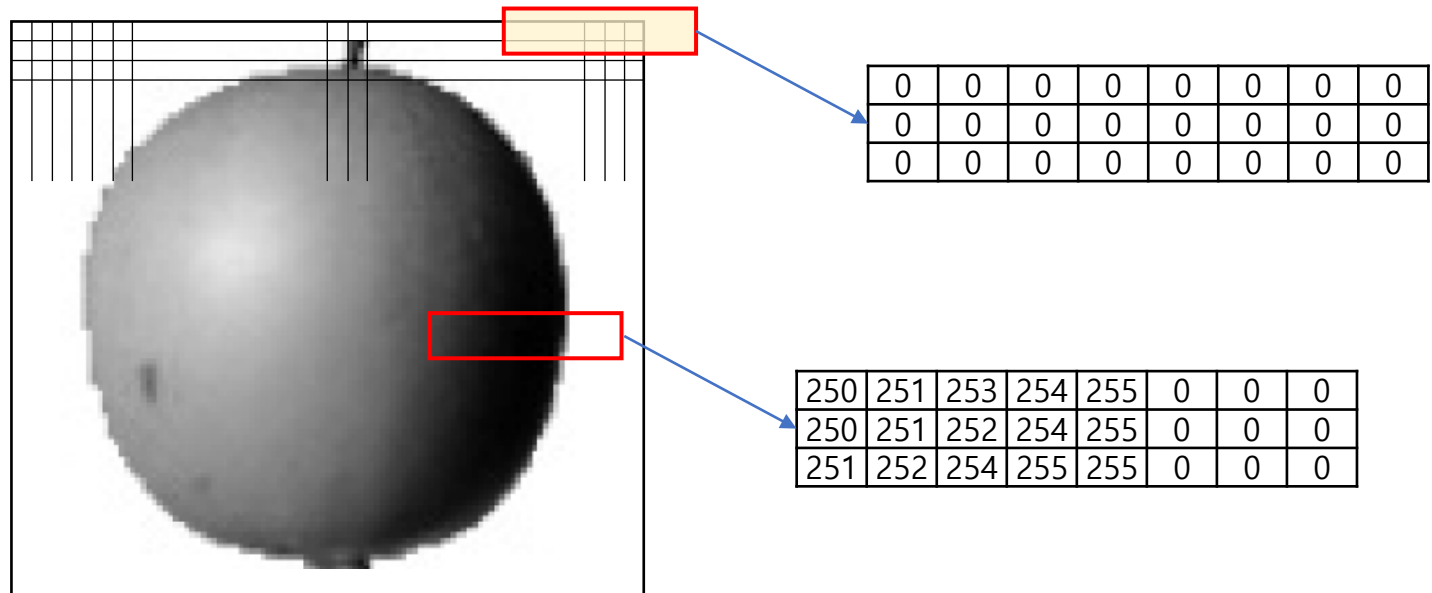
206	225	193	185	182	174	151	137	132	161	140	119	132	120	141	121	125
230	194	191	197	189	160	164	143	156	157	99	160	127	113	121	122	107
195	188	213	185	148	178	153	160	148	116	123	123	155	142	107	151	117
183	191	190	170	177	167	148	164	145	134	127	158	140	112	128	97	146
194	177	189	184	171	139	179	149	108	142	146	140	122	114	115	109	164
177	171	170	175	161	165	172	121	149	153	127	116	131	148	133	133	140
180	177	157	156	168	171	137	145	153	134	138	149	133	128	137	123	119
185	160	171	149	157	142	132	147	124	129	118	138	132	118	165	138	104
165	159	145	176	159	125	159	137	131	142	152	152	116	135	147	106	122
153	180	186	168	139	160	151	158	114	155	172	83	125	154	107	124	152
176	191	153	127	166	140	144	149	164	158	71	184	166	81	147	150	132
177	145	124	151	152	154	140	179	156	92	161	201	108	101	165	128	131
139	131	152	146	140	158	173	159	92	170	171	89	123	161	124	136	99
145	136	169	150	141	134	175	106	158	155	142	121	144	137	102	112	107
141	157	158	121	139	169	137	135	165	124	145	129	105	104	118	112	118
158	149	122	135	153	140	107	156	121	152	156	118	124	129	118	104	94
165	142	145	132	156	117	135	146	127	138	107	95	116	120	102	94	93
130	168	151	132	132	134	125	139	116	132	126	111	129	106	99	102	123
171	173	149	136	133	111	130	121	120	102	104	127	120	111	106	102	118
185	171	150	109	133	125	120	114	105	121	109	111	111	103	115	100	96
181	138	124	129	102	123	107	138	119	101	108	109	114	95	102	109	125
155	137	131	109	114	105	128	119	104	102	103	121	104	129	103	124	110
140	120	139	128	103	116	110	122	110	106	103	112	110	108	124	120	104
119	111	136	112	125	125	122	115	90	119	105	98	132	101	126	91	122
125	127	132	91	134	121	82	117	109	96	97	112	130	109	113	126	129

이미지 처리를 위한 사전학습

- 이미지크기가 100 x 100인 사진 300장 모여있다.

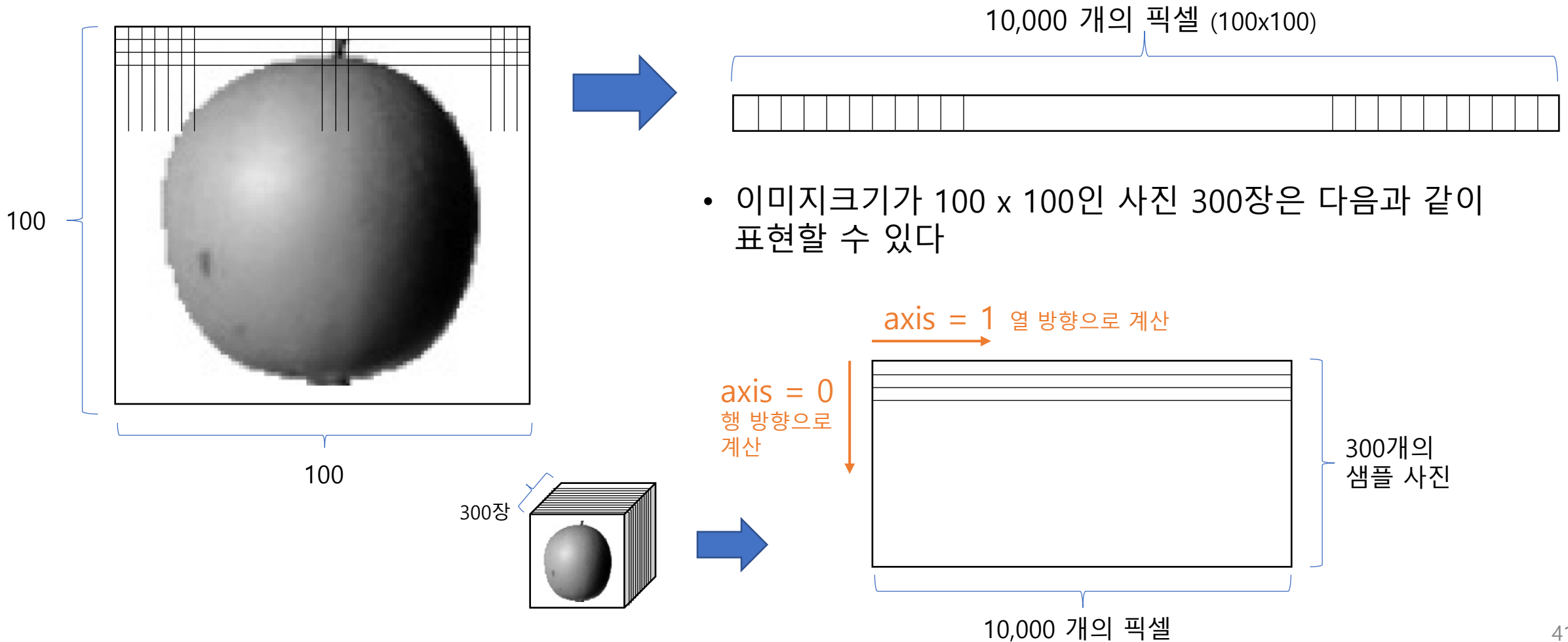


- 하나의 사진은 다음과 같이 픽셀로 나타난다.
- 각 픽셀 안에는 색상을 나타내는 숫자가 입력되어 있다.



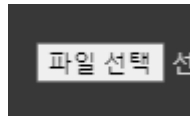
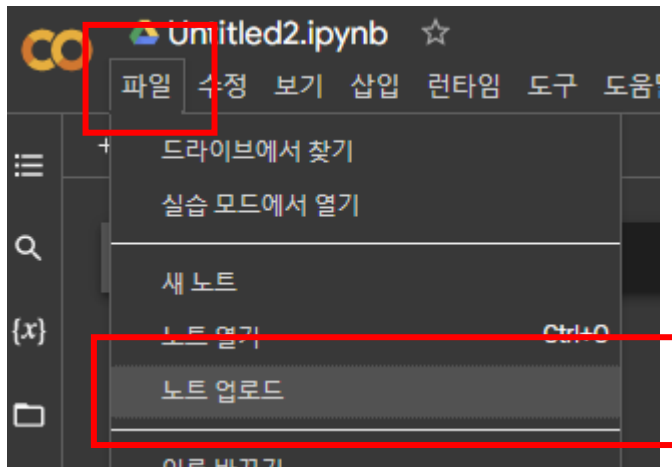
이미지 처리를 위한 사전학습

- 픽셀값 분석을 위해 2차원 배열을 1차원 배열로 변환함 : **Flattening** (배열 계산에 편리)




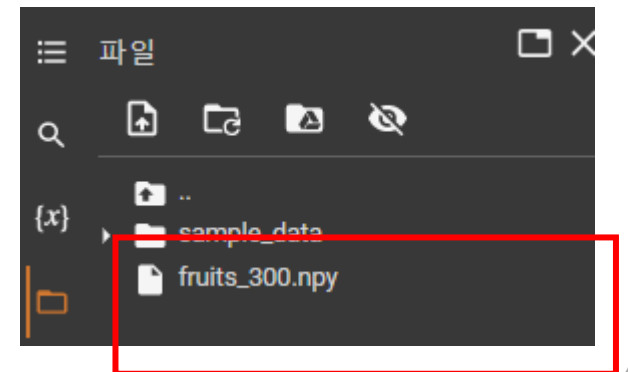
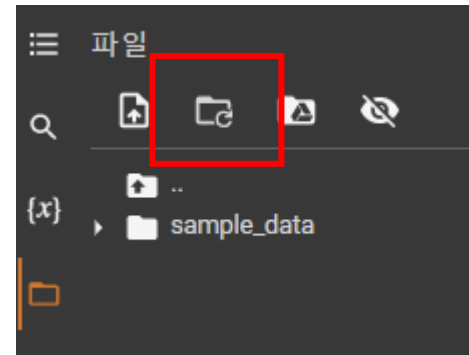
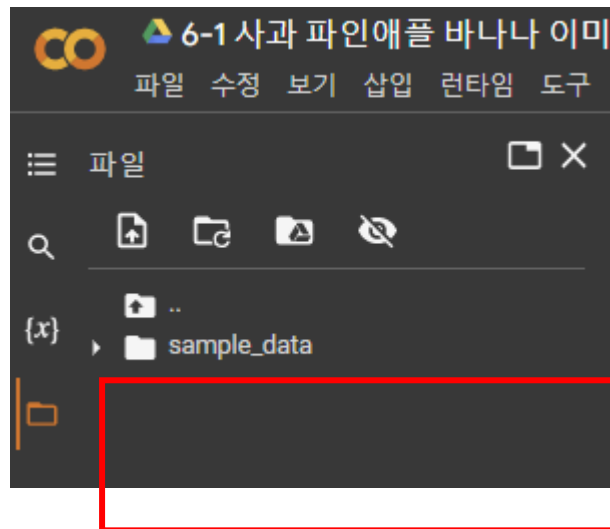
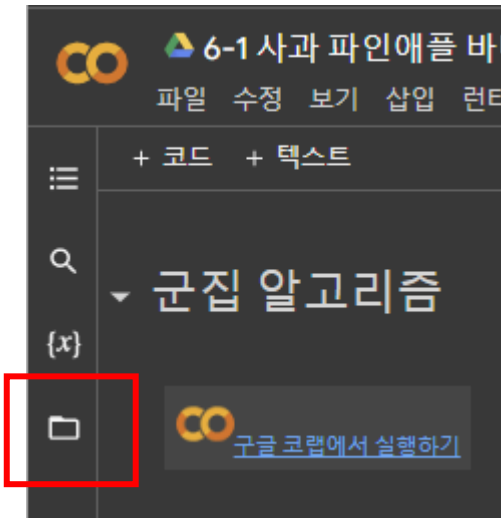
구글 코랩에서 소스코드 불러오기

- 'code8 머신러닝_비지도학습실습_이미지_평균값_활용.ipynb' 다운로드
- 구글 코랩 - 새노트 - 파일 - 노트업로드 - (열린 창에서) 파일선택 click - (다운 받은 폴더로 가서 해당 파일 클릭) 열기 - '업로드 중' - 아래 화면 열리면 성공



코드를 하나씩 실행

-  버튼 누르거나 'Ctrl + Enter'
- 데이터 셋 다운로드 확인하기
 - !wget https://bit.ly/fruits_300_data -O fruits_300.npy 실행 후



K-평균 이용한 비지도학습 실습

분류, 회귀분석, 군집화

비행기



자동차



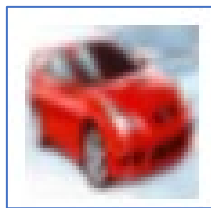
새



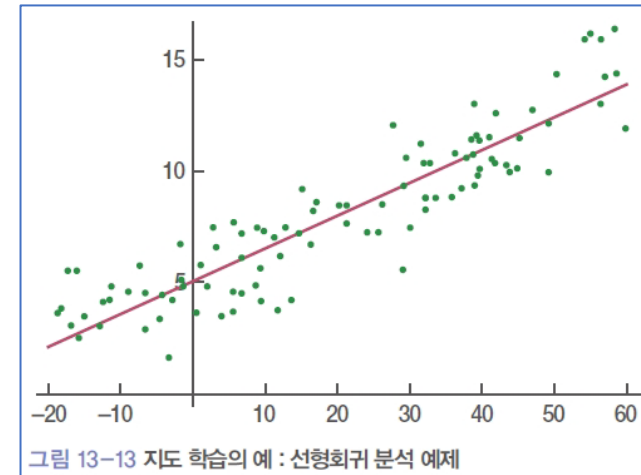
고양이



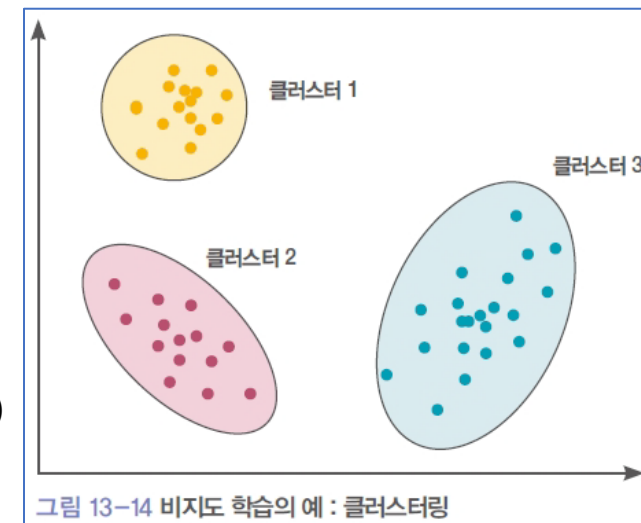
이것은 어떤
그룹으로
분류되나요?



회귀분석



군집화
(클러스터링)



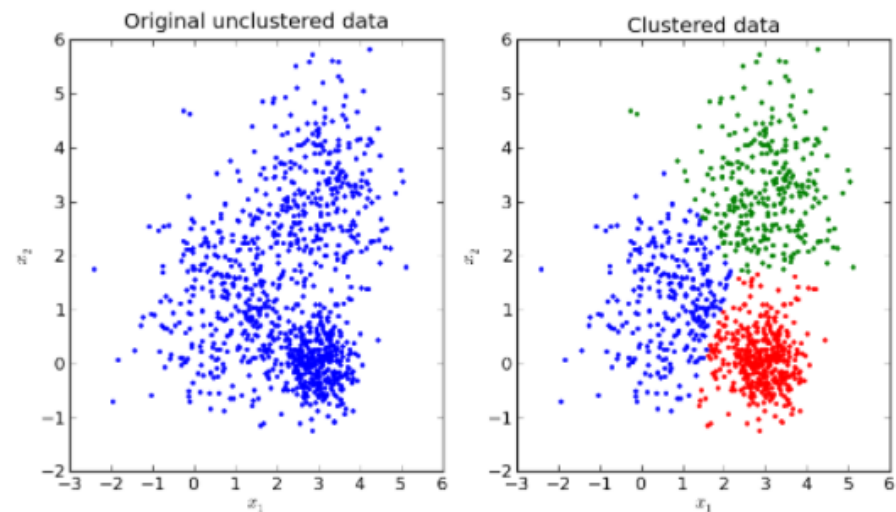
분류, 회귀, 군집화

- 군집화 clustering 1)

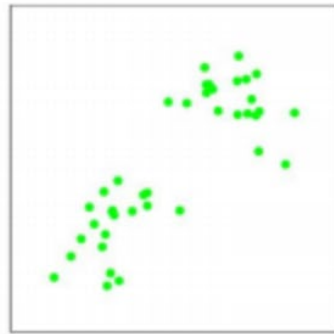
- 훈련용 데이터집합에서 서로 유사한 것들을 스스로 묶어서 군집을 형성하는 과정 (비지도 학습)
- 군집화를 위해서는 유사성의 판단 기준을 미리 정해야한다.
- 유사성은 데이터 간의 '거리' 를 이용하여 판단할 수 있다.
- '거리' 라는 개념은 다양할 수 있다.

- 대표적 알고리즘 : K-평균 알고리즘

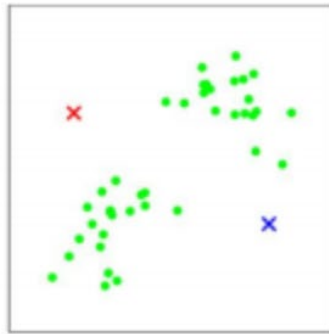
1. 무작위로 K개의 중심점을 선정한다
2. 중심점과 각 데이터간의 거리를 계산한다
3. 가장 가까운 거리의 중심점에 속하도록 한다
4. k개의 클러스터의 중심점으로 재조정한다.
5. 위 2~4 과정을 반복한다



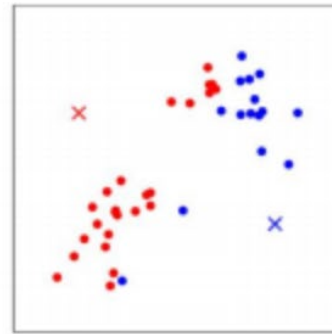
K-평균 군집화 예제



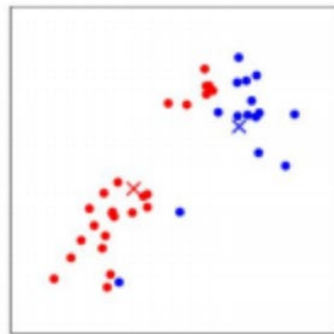
(a)



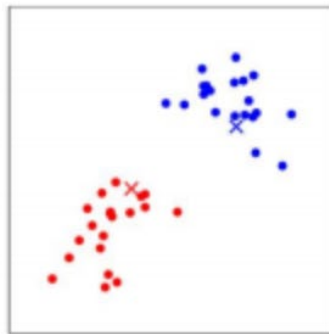
(b)



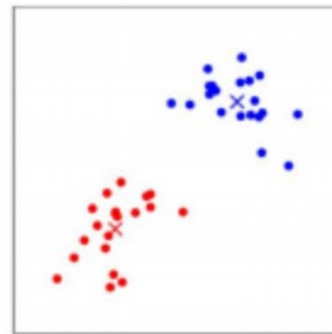
(c)



(d)



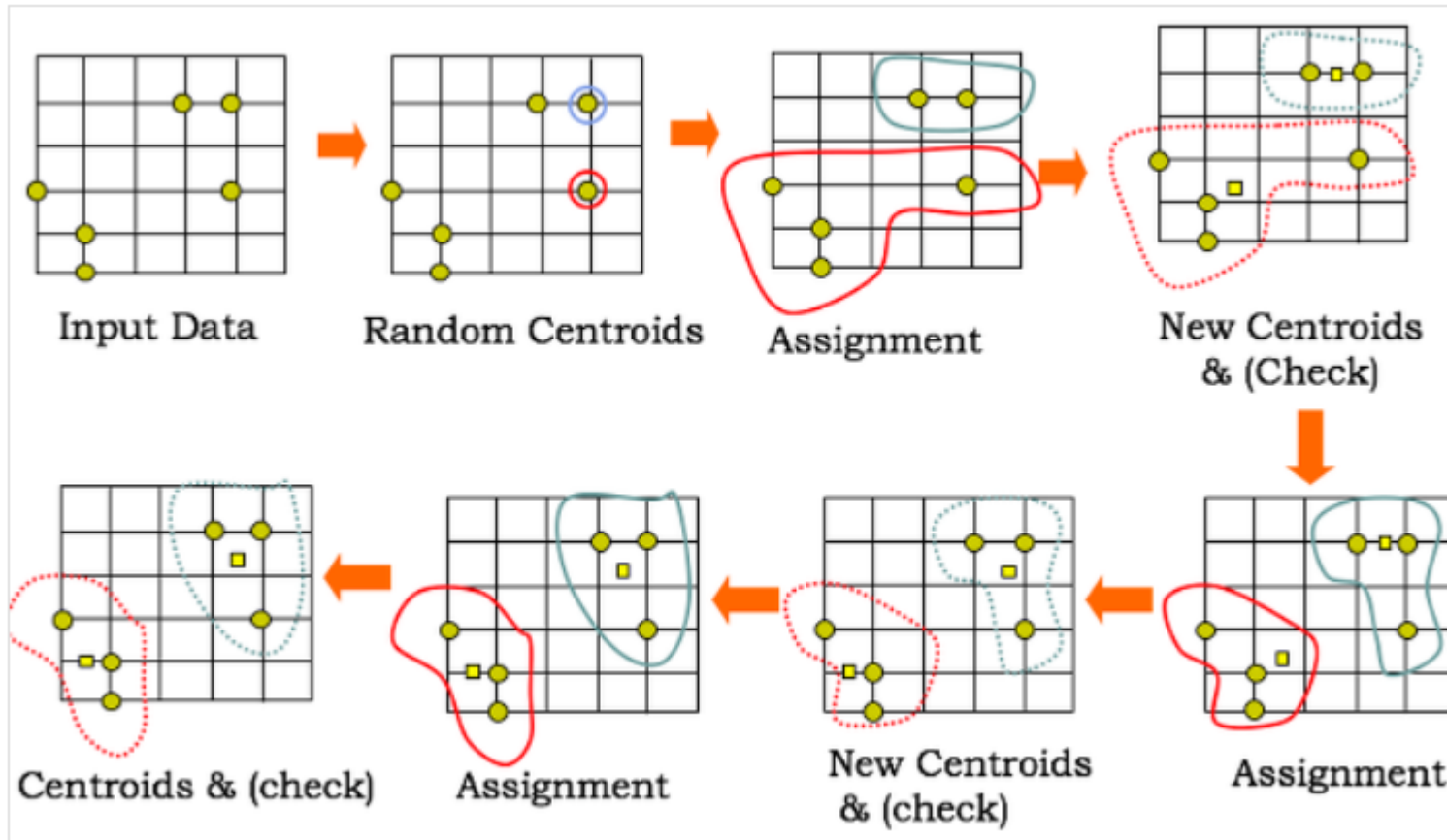
(e)



(f)

<https://hleecaster.com/ml-kmeans-clustering-concept/>

K-평균 군집화 예제



<https://nicola-ml.tistory.com/1>

구글 코랩에서 소스코드 불러오기

- "code9 머신러닝_비지도학습실습_k_평균_이용.ipynb" 다운로드 후 활용
- 구글 코랩에서 '노트 업로드' 메뉴로 업로드

다음 시간

- 6일차 (8.2, 수)
 - 수업 오후 2시에 시작해서 3시간 수업 (2:00~4:50)