

한국정당학회 하계 사회과학 방법론 특강 인과추론과 데이터 사이언스

데이터 사이언스와 사회과학 연구: 머신러닝

김서영

2024년 7월 10일

서강대학교 정치외교학과

목차

들어가며: 데이터 사이언스와 R Programming 소개

데이터 사이언스란 무엇인가?

컴퓨팅 환경 소개

분류(Classification)와 예측(Prediction)

머신러닝이란 무엇인가?

예측 및 분류의 성과평가

지도 학습(Supervised Learning)

훈련-테스트 패러다임(Training-testing Paradigm)

교차 검증(Cross-validation)

트리 기반 모형(Tree-based Models)

비지도 학습(Unsupervised Learning)

주성분 분석(Principal Component Analysis)

K-평균 군집분석(K-means Clustering)

들어가며: 데이터 사이언스와 R Programming 소개

들어가며: 데이터 사이언스란 무엇인가?

- **Statistics + computer science + domain expertise**
- 90년대부터 개념이 정립되기 시작
(Knowledge Discovery in Databases, data mining, ...)
- **데이터 마이닝/데이터 모으기, 저장(storage), 클리닝(cleaning, wrangling)/전처리(pre-processing), 시각화(visualization), 분석 (analysis), ...**
- 일상생활에서, 산업계에서, 시민단체/정부에서, 연구에서 다양하게 활용
- “The volume, velocity, variety, and veracity of data”

데이터 사이언스란 무엇인가?

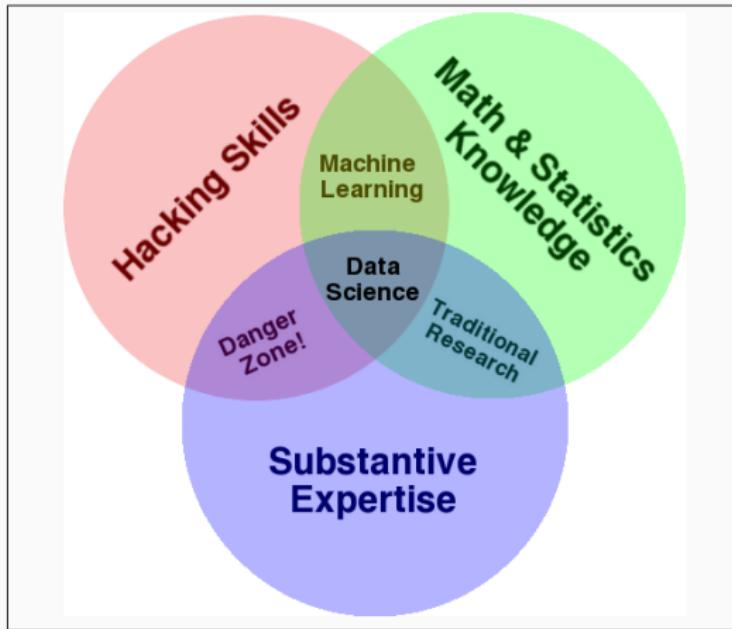


Figure 1-1. Drew Conway's Venn diagram of data science

Figure from O'Neil, Cathy, and Rachel Schutt. 2013.
Doing Data Science: Straight Talk from the Frontline. O'Reilly Media, Inc.

데이터 사이언스란 무엇인가?

Table 1 The seven activities of data science^a

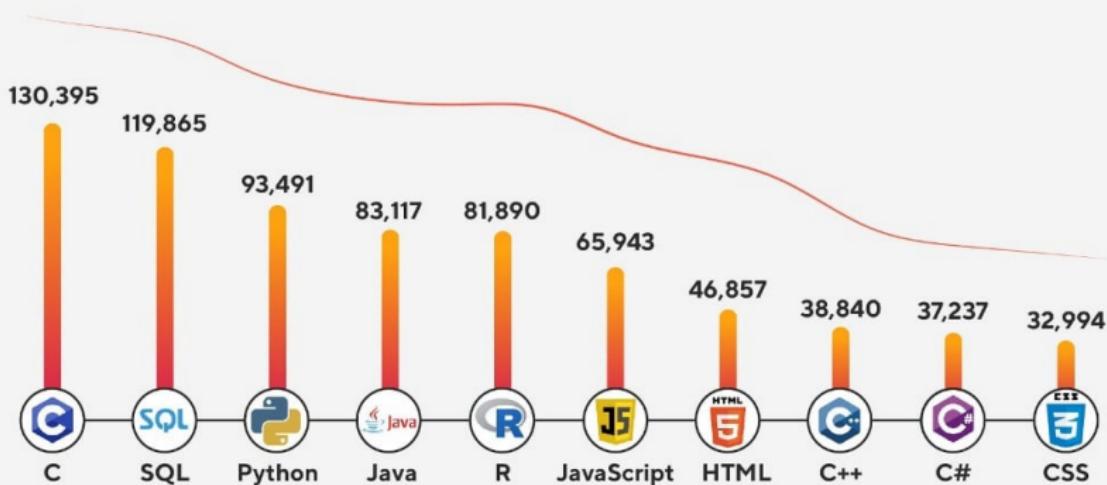
Activities	Examples
Data gathering, preparation, and exploration	Survey data, experimental data, genomic data, textual data, administrative data, image data, web data, and sensor data Data cleaning and exploratory data analysis methods for checking on outliers and data quality
Data representation and transformation	Relational and nonrelational databases Networks and graphs Other mathematical structures for data
Computing with data	R and Python Programming packages, text manipulation languages Cluster and cloud computing Reproducible workflows
Data modeling	Determining or hypothesizing data generating probability functions, structural and predictive modeling
Data visualization and presentation	Types of visualizations and graphs Rules for labeling and presenting data Psychological impacts of various displays
Data archiving, indexing, and search and data governance	Standards for open data and reproducibility Determining rules for access and privacy protection where necessary
Science about data science	How people do data science Impacts of data science and big data on society

^aThe activities are quoted from Donoho (2017, p. 755) except for “Data archiving, indexing, and search and data governance,” which is my addition. The examples are my own.

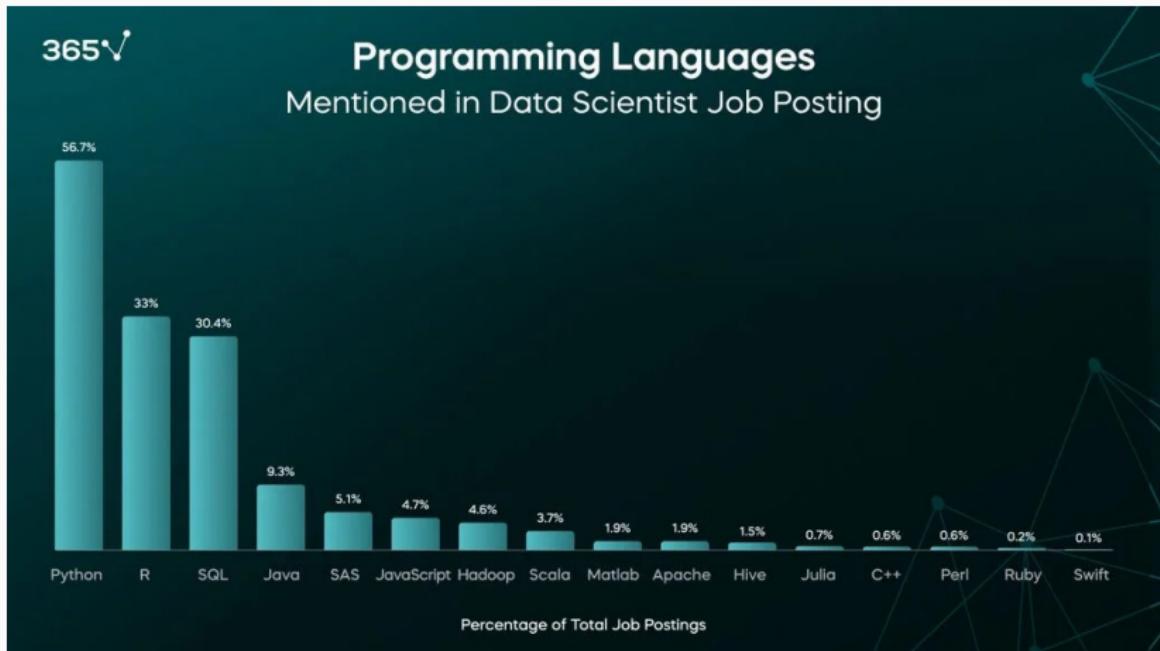
From Brady, Henry E. 2019. “The Challenge of Big Data and Data Science.” Annual Review of Political Science 22(1): 297–323. doi:10.1146/annurev-polisci-090216-023229.

R은 왜?

Top Programming Languages by Job Openings



R은 왜?

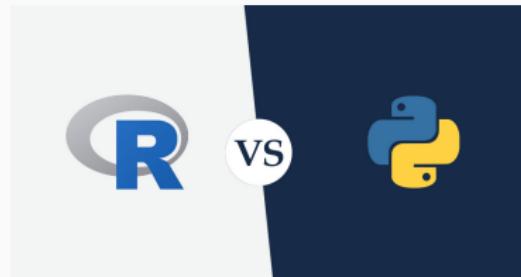


The Data Scientist Job Market in 2024, 365 Data Science

R을 선호하는 이유

- 객체지향적 프로그래밍(Object-oriented programming, OOP) + 함수형 프로그래밍(functional program)
- **무료!**
- 전문가로 구성된 코어 그룹에 의해 체계적 관리
- 자료 분석 전 과정에서 두루 사용될 수 있음
- 끊임없이 업데이트 + 발전
- 다른 언어와 연동이 쉽고 확장성이 뛰어남
- R을 사용하는 회사 리스트:
<https://github.com/ThinkR-open/companies-using-r>
- 초보자에게 굉장히 친절한 커뮤니티 유지

파이썬을 쓰면 안 되나요?



Technology Holy Wars are Coordination Problems

R과 RSTUDIO

엔진 = R



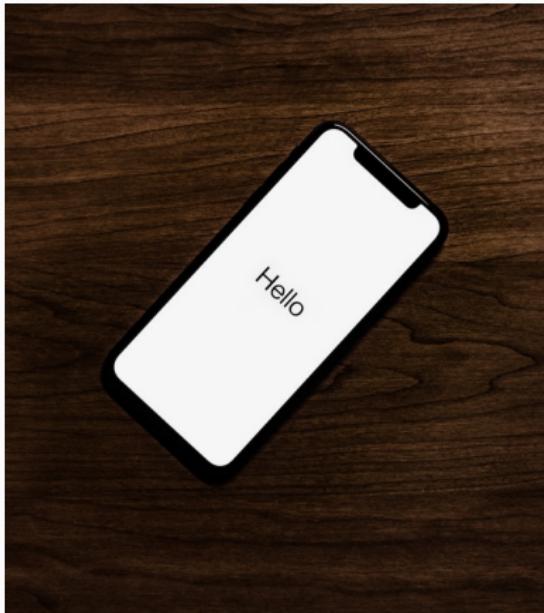
인터페이스 = RStudio



언어 vs. 통합개발환경(IDE) Borrowed from [Thomas Mock's slides](#)

R 패키지

새 폰 = R

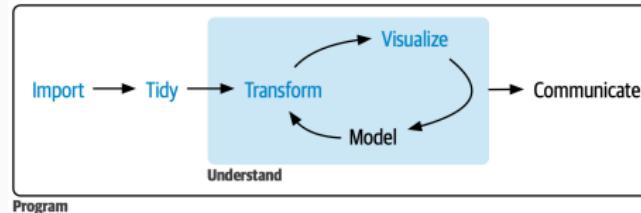


앱스토어 앱 = R 패키지

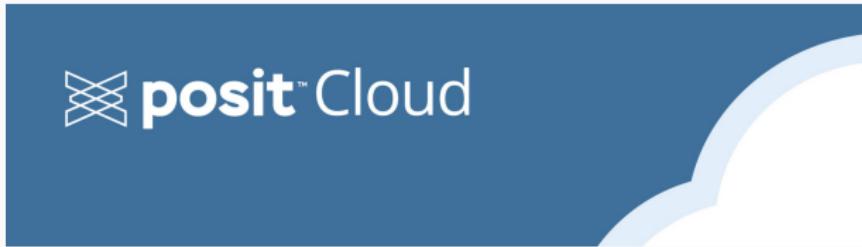


TIDYVERSE(타이디버스)

- 해들리 위컴 (Hadley Wickham)
- 자료 분석에 대한 위컴의 접근법 = “tidy” 접근법
- <https://www.tidyverse.org/>
- R for Data Science



POSIT CLOUD(구: RSTUDIO CLOUD)



- 컴퓨터에 설치 불필요
- 정당학회에 등록한 이메일로 초대
- 수업에서 사용한 코드 공개

RSTUDIO의 4개 패널

The screenshot displays the RStudio interface with four main panels:

- (1) Source, or Code Editor:** The top-left panel shows an R Markdown script named "example.Rmd". The code includes YAML front matter and an R code chunk. A red box highlights this panel.
- (2) Console:** The bottom-left panel shows the R console output for the command "head(ces)". A red box highlights this panel.
- (3) Environment/History:** The top-right panel shows the Global Environment and Data pane, listing datasets like "ces" and "fips". A green box highlights this panel.
- (4) Files/Plots/Help:** The bottom-right panel shows the Project Explorer, displaying files and folders within a "project" directory. A purple box highlights this panel.

연습

- Posit Cloud에서 첫 프로젝트 실행하기
- 콘솔에서 $5+3$ 계산해보기
- `#` 를 사용해서 “코멘트” 남겨보기
- `swirl` 패키지 설치하기
- `swirl :: install_course_github("kosukeimai", "qss-swirl")` 실행하기

팁: Tools → Global Options에서 Save/Restore 전부 해제

```
1 5 + 3 ## addition  
2 5 -3 ## subtraction  
3 5 / 3 ## division  
4 5 ^ 3 ## power operation  
5 5 * (10 -3)  
6 sqrt(4) ## if you are curious what "sqrt" is going to do, run ?sqrt
```

Adobe Acrobat으로 열면 슬라이드의 코드를 복사붙여넣기 가능

R 기초 문법

- `#` 을 한 개 이상 사용 = 주석/코멘트
- 괄호를 여는 단어는 함수. 예시: `sqrt(4)`
- 할당:
 - x라는 변수에 숫자형 데이터 할당: `x <- 123`
 - x라는 변수에 문자형 데이터 할당: `x <- "123"`
- 객체 타입 파악하기: `class()` 함수
- 출력하기: `print("Hello World")`

분류(Classification)와 예측(Prediction)

머신러닝이란 무엇인가?

- **머신(machine)** = 자동화, 계산, 알고리즘, 컴퓨터 사이언스, 인공지능(artificial intelligence)의 하위 분야
- **러닝(learning)** = 데이터를 통해 학습, 점진적 개선

머신러닝이란 무엇인가?

머신 러닝이란?

머신 러닝은 인간이 학습을 통해 정확도를 점진적으로 개선하는 방식을 모방하기 위한 데이터와 알고리즘의 사용에 초점을 맞춘 [인공지능\(AI\)](#) 및 컴퓨터 사이언스의 한 분야입니다.

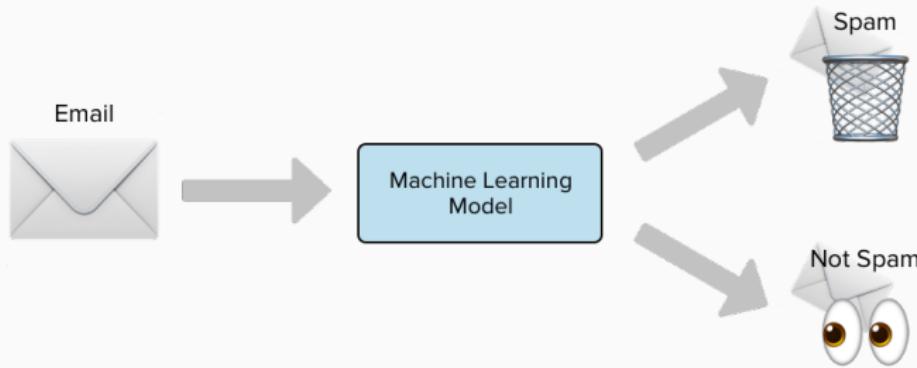
IBM은 머신 러닝 분야에서 깊은 [역사](#)를 가지고 있습니다. 그 중에서 Arthur Samuel은 체커 게임과 관련된 자신의 [연구](#) (IBM 외부 링크)에서 "머신 러닝"(machine learning)이라는 용어를 처음 만들었다고 인정받고 있습니다. 자칭 체커의 달인이라고 주장하는 Robert Nealey는 1962년에 IBM 7094 컴퓨터에서 게임을 실행했으며, 컴퓨터에게 졌습니다. 오늘날 할 수 있는 것에 비하면 이 정도의 업적은 사소한 것처럼 보이지만, 이는 인공지능 분야에서 중요한 이정표로 간주되고 있습니다.

지난 20년 동안 스토리지 및 프로세싱 기능의 기술적 발전으로 인해 Netflix의 추천 엔진과 자율주행 차와 같은 머신 러닝 기반의 혁신적 제품들이 탄생했습니다.

머신 러닝은 계속 발전 중인 데이터 사이언스 분야의 중요한 구성요소입니다. 통계적 방법을 사용하여 알고리즘은 데이터를 분류 또는 예측하고, 데이터 마이닝 프로젝트에서 중요 인사이트를 도출하도록 훈련을 받습니다. 이러한 인사이트는 결과적으로 핵심 성장 메트릭에 이상적으로 영향을 미치는 애플리케이션 및 비즈니스 내의 의사결정을 가속화합니다. 빅데이터 분야가 계속 확장되고 성장함에 따라 데이터 사이언티스트에 대한 시장 수요는 증가할 것입니다. 데이터 사이언티스트는 가장 중요한 비즈니스 질문을 찾아내고 이러한 질문에 답하기 위해 필요한 데이터를 확보해야 할 것입니다.

[IBM 웹사이트](#)

머신러닝: 예시



Picture from [Email Spam Classifier Using Naive Bayes](#)

머신러닝: 예시

4	$\rightarrow 4$	2	$\rightarrow 2$	3	$\rightarrow 3$
4	$\rightarrow 4$	9	$\rightarrow 9$	0	$\rightarrow 0$
5	$\rightarrow 5$	1	$\rightarrow 7$	1	$\rightarrow 1$
9	$\rightarrow 9$	0	$\rightarrow 0$	3	$\rightarrow 3$
6	$\rightarrow 6$	7	$\rightarrow 7$	4	$\rightarrow 4$

Picture from Wolfram Alpha

머신러닝: 예시

NETFLIX JOIN NOW Sign Out

Only on Netflix

Netflix is the home of amazing original programming that you can't find anywhere else. Movies, TV shows, specials and more, all tailored specifically to you.

Popular on Netflix

Recently Added
Ballerina

A Time Called You

Recently Added
Beckham

Asian TV Shows

The Rise of Phoenixes

Meteor Garden

ALICE IN BORDERLAND

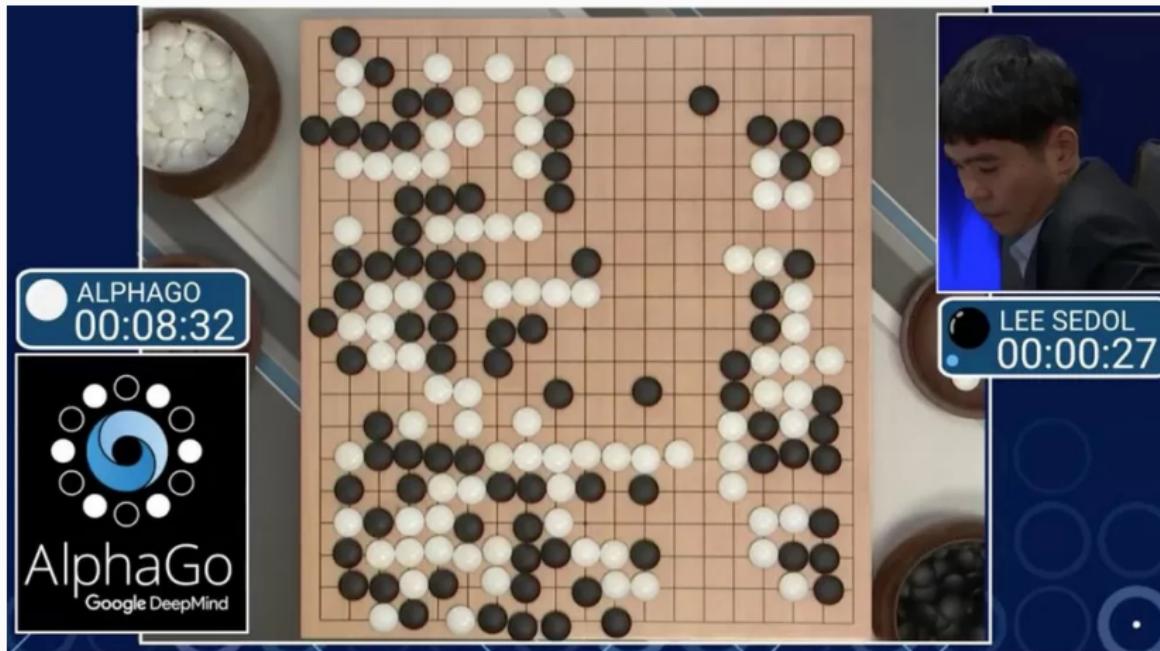
Korean TV Shows

D.P.

Song of the Bandits

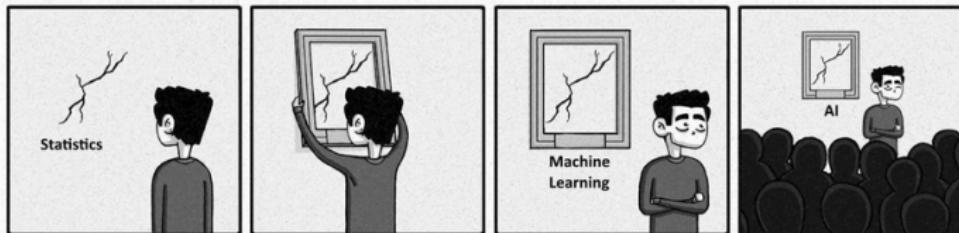
Juvenile Justice

머신러닝: 예시



Picture from BBC

머신러닝과 통계학



Artist: [sandserifcomics](#)

 Rex "garbage in" Douglass Ph.D. 
@RexDouglass

Because I never get tired of this joke:

It's statistics if it's in R.
It's machine learning if it's in Python.
It's AI if it's in PowerPoint.

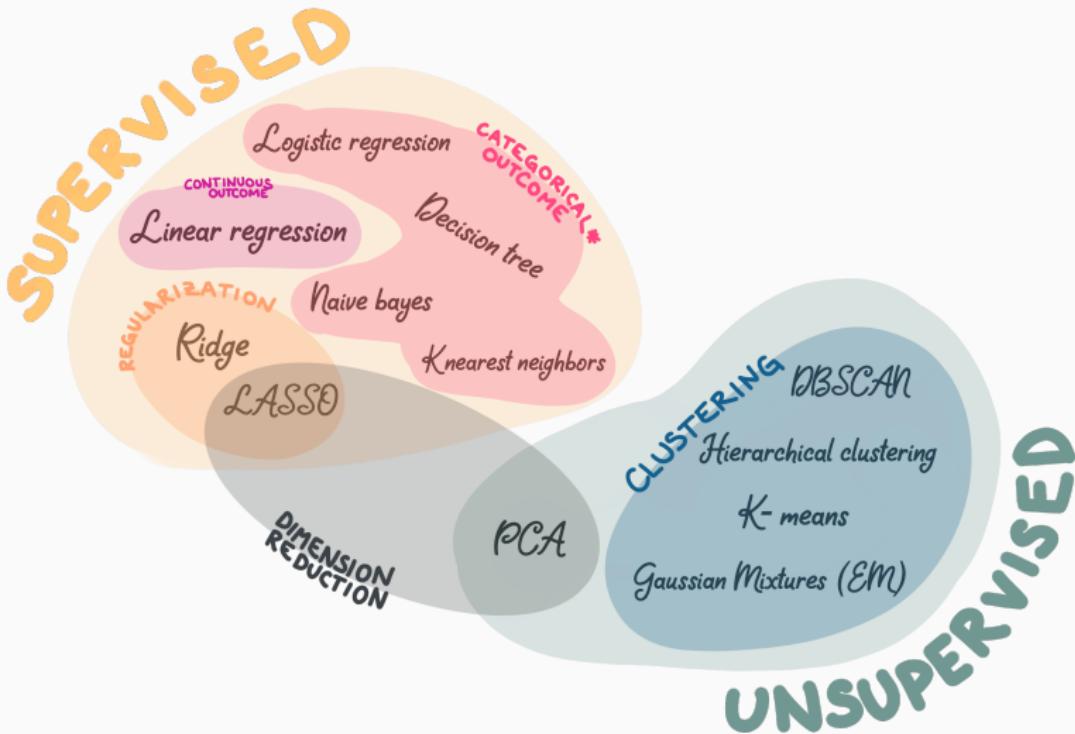
5:22 PM · Jun 27, 2024 · **288.9K** Views

22 586 6.5K 812 ↗

머신러닝과 데이터 사이언스

- 머신러닝 \subset 데이터 사이언스?
- 데이터 사이언스 \subset 머신러닝?

머신러닝: 지도 학습 vs. 비지도 학습



지도 학습 vs. 비지도 학습

지도 학습(supervised learning)

- 결과 변수 Y 와 예측 변수 X (p 개 변수, 머신러닝에서는 feature라고도 함)
- 회귀(regression) 문제라면 Y 는 보통 연속변수
- 분류(classification) 문제라면 Y 는 순서가 없고 유한한 변수
e.g., 생존/사망, 투표함/안함, 스팸인가 아닌가, 수능 등급, 0-9 사이의 자연수, ...
- Training set을 통해 아직 안 본 test set에 대해 예측하고, 어떤 변수가 결과에 가장 큰 영향을 미치는지 보고, 예측 성과를 평가하고 싶을 때 사용

지도 학습 vs. 비지도 학습

비지도 학습(unsupervised learning)

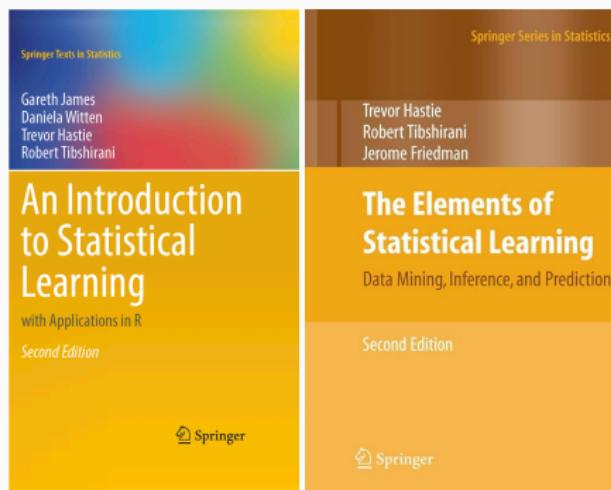
- 결과 변수가 없고 대신 설명 변수만 있음
- 목적이 좀 다름
 - 비슷한 성향을 보이는 샘플 내 그룹 찾기
 - 비슷한 성향을 보이는 feature 찾기
 - 데이터 요약 등등
- 클러스터링(clustering), 차원 축소(dimension reduction)
- 성과를 측정하고 평가하기가 어려울 수 있음
- 지도 학습을 하기 위한 디딤돌로서 유용한 경우도 있음

통계적 학습(STATISTICAL LEARNING) vs. 머신러닝

약간의 차이

- 인공지능 → 머신러닝: 예측의 정확성과 대규모 데이터에 집중 e.g., 마케팅
- 통계학 → 통계적 학습: 모형, 정밀도, 불확실성, 해석 가능성에 집중

ISLR 및 ESL은 통계적 추론에 관한 책



회귀분석 vs. 머신러닝

- ISLR의 3장 = 선형회귀!
- **지도 학습의 가장 기본이 되는 방법**
- 대표적인 모수적 모형(parametric model)
다른 방법들은 선형회귀에서 출발해서 응용한 경우가 많음
- 오차가 커 보일 수 있지만 편향-분산 트레이드오프 문제
- 좀 더 복잡한 머신러닝은 해석가능성을 희생하고 예측가능성을 높인 경우가 많음
- “Essentially, all models are wrong, but some are useful” (George Box)

하지만 근본적으로 “사회과학”은 추론(inference)이 중심인 반면
머신러닝은 예측(prediction)이 중심

그럼 왜 머신러닝을 사회과학에서 사용하나?

- 사회과학에서도 예측이 중요할 때가 있음 e.g., Kim, Seo-young Silvia, and Jan Zilinsky. 2022. “Division Does Not Imply Predictability: Demographics Continue to Reveal Little About Voting and Partisanship.”
- 유연하고 비모수적인 관계를 포착하는 것이 중요할 수 있음
- p 가 n 에 비해 커서 차원의 저주(curse of dimensionality)가 있을 경우 변수 선택(variable selection)에 효과적일 수 있음
- 이미지, 음성, 비디오 등 데이터를 다룰 때 유용
- 과적합을 방지하기 위한 표본외 예측

참고: Efron, B. (2020) Prediction, Estimation, and Attribution. International Statistical Review, 88: S28–S59. <https://doi.org/10.1111/insr.12409>.

머신러닝과 예측

예측이 중심이 되는 것이라면 예측의 성과(performance)를 어떻게 평가할까?

모형 부적합: 과적합(OVERFITTING) 및 과소적합(UNDERFITTING)

· 과적합(Overfitting)

- 연구자가 관측자료의 설명에 적합한 방식으로 모형을 지나치게 조정한 나머지 관측자료는 잘 설명하지만 새로운 자료에 대해 설명력이 취약한 상태
- 관측자료에 비해 모수의 수가 너무 많거나, 설명변수가 종속변수의 내생변수거나, 모형이 관측자료에 맞게 지나치게 유연하게 설계된 경우

· 과소적합(Underfitting)

- 연구자가 설정한 모형이 관측자료를 제대로 설명하지 못하고 관측자료의 극히 일부분만을 설명하고 있는 상태
- 모형이 관측자료의 생성과정을 제대로 반영하지 못함
- 추정된 모수가 견고성(robustness)이 떨어지고 모형의 설정 변경에 민감하게(sensitive) 반응하기 때문에 신뢰하기 어려움

과적합 vs. 과소적합

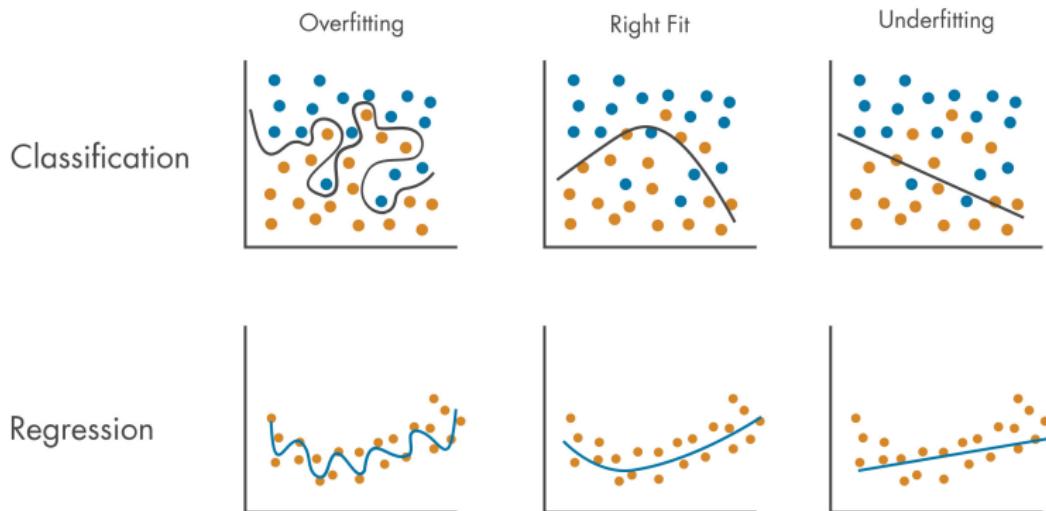
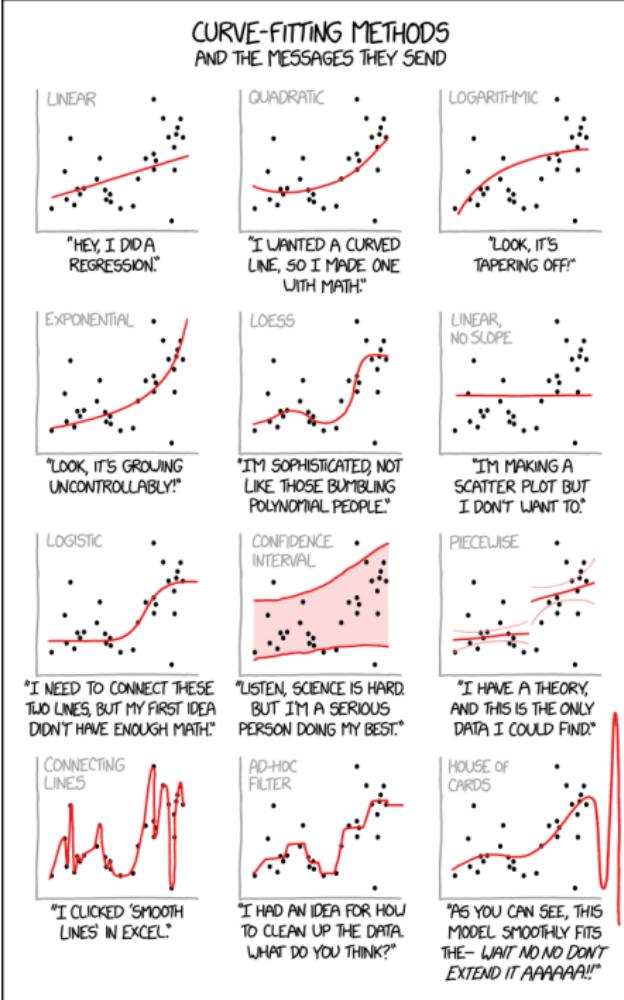


그림 출처: MathWorks

예측을 이용한 모형적합성 확인

예측적 추론

- **표본내 예측(in-sample prediction)** =
표본내 예측을 통해 추정된 종속변수를 실제 종속변수와 비교
- **표본외 예측(out-of-sample prediction)** =
교차타당성(cross-validation) 검사 등



머신러닝과 예측

예측이 중심이 되는 것이라면 **예측의 성과(performance)**를 어떻게 평가할까?

분류인지 예측인지에 따라 평가 방법이 달라짐

- 분류: 혼동행렬에서 비롯되는 여러 지표들
- 예측: 지난번에 배운 MSE, 조정된 R^2 , ...

분류(CLASSIFICATION)

- 특성 벡터(feature vector) X 및 질적 변수(qualitative variable) Y
- Y 가 \mathcal{C} 집합 내의 값을 가진다고 했을 때 함수 $C(X) \in \mathcal{C}$ 를 만드는 게 목표
- \mathcal{C} 의 각각의 카테고리에 속할 확률에 관심있는 경우가 많음
 - e.g., 이 이메일이 스팸일 확률은 얼마나 될까?

분류(CLASSIFICATION)

- 회귀분석을 쓴 후 $\hat{Y} > 0.5$ 인 경우 스팸이라고 분류하는 것도 가능
(linear discriminant analysis; $E(Y|X = x) = \Pr(Y = 1|X = x)$)
- 근데 Y 가 이항변수가 아니라면?
 - e.g., (미국에서) 투표 방법: 선거 당일 대면 투표, 우편 투표, 사전 투표
 - 대면 투표 = 1, 우편 투표 = 2, 사전 투표 = 3이라고 하면
양적 변수로 취급했을 때 우편 - 대면 = 사전 - 우편이라고 하는 셈!

분류의 성과평가: 혼동행렬

혼동행렬(confusion matrix) 또는 오차행렬 (이항변수의 경우)

예측된 결과	실제 결과	
	TRUE	FALSE
TRUE	참긍정 (true positive, TP)	거짓긍정 (false positive, FP)
FALSE	거짓부정 (false negative, FN)	참부정 (true negative, TN)

분류의 성과평가: 여러 지표들

정확도(Accuracy)

$$\frac{TP+TN}{TP+TN+FP+FN}$$

정밀도(Precision)

$$\frac{TP}{TP+FP}$$

또는 Positive predictive value (PPV)

민감도(Sensitivity)

$$\frac{TP}{TP+FN}$$

또는 재현율(Recall)

또는 True positive rate (TPR)

특이도(Specificity)

$$\frac{TN}{TN+FP}$$

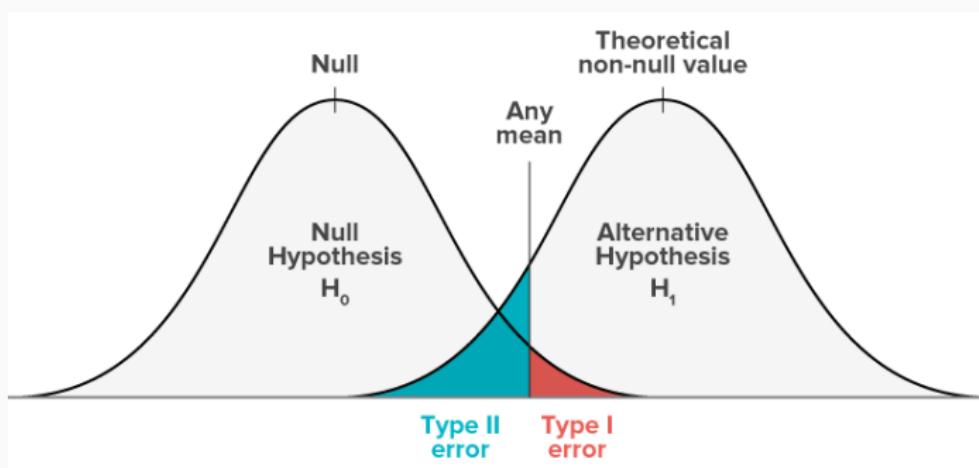
또는 True negative rate (TNR)

F1값 (precision과 recall의 조화 평균)

$$\frac{2\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$$

분류의 성과평가: TYPE I ERROR VS. TYPE II ERROR

	H_0	
	TRUE	FALSE
검정 결과	Accept	Type II Error
	Reject	Type I Error



출처: The Errors of A/B Testing: Your Conclusions Can Make Things Worse

분류의 성과평가: TYPE I ERROR VS. TYPE II ERROR

상황에 따라 한 쪽이 정말 중요한 문제일 수 있음. 예시:

- 실제로 암인데 암이 아니라고 판별하는 경우
- 실제로 범인이 아닌데 형을 집행하는 경우
- ...

분류의 성과평가: TYPE I ERROR VS. TYPE II ERROR

Type I: false positive

Positive

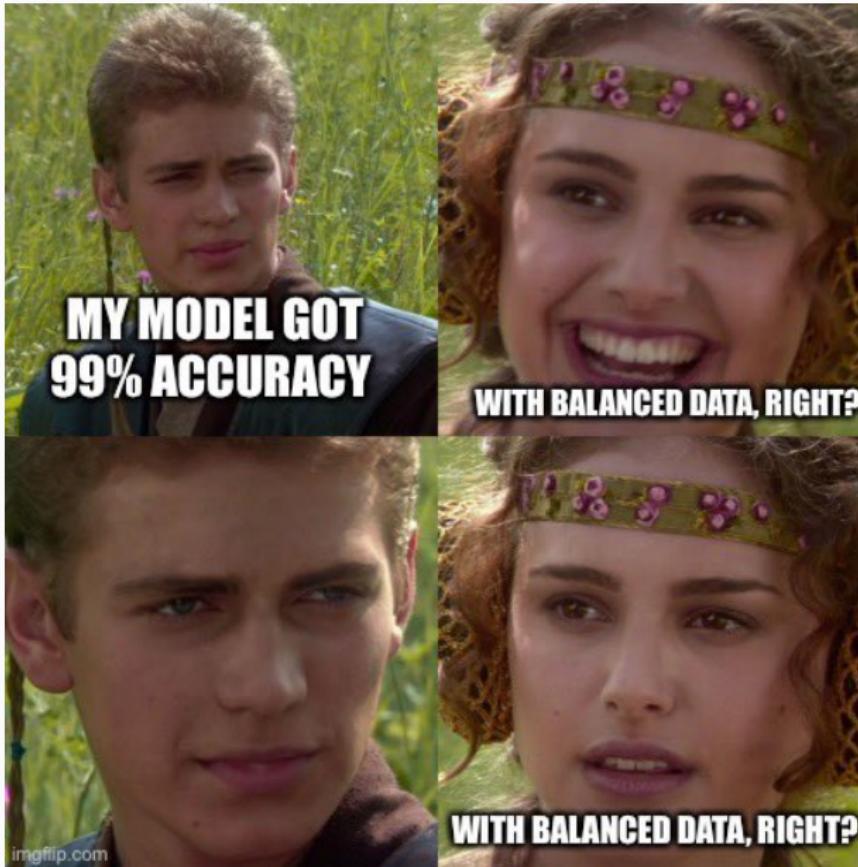
Type II: false negative

Inegative

언제 어떤 지표가 적절할까?

Accuracy: 클래스 불균형(class imbalance)가 있는 경우 썩 좋지 않을 수 있음

- 예시: **내전(civil war)의 시작**을 예측해보자 (Wang, Yu. 2019. “Comparing Random Forest with Logistic Regression for Predicting Class-Imbalanced Civil War Onset Data: A Comment.” Political Analysis 27(1): 107–10.)
- 7,000건+의 나라 × 연도 데이터 중 **고작 0.016%의 데이터가 positive class**
- “옛날에도 앞으로도 내전은 절대로 없다”고 예측하면 accuracy는 99.5% 이상 그런데 우리가 관심있는 것은 사실 positive class의 예측!



클래스 불균형과 DECISION THRESHOLD

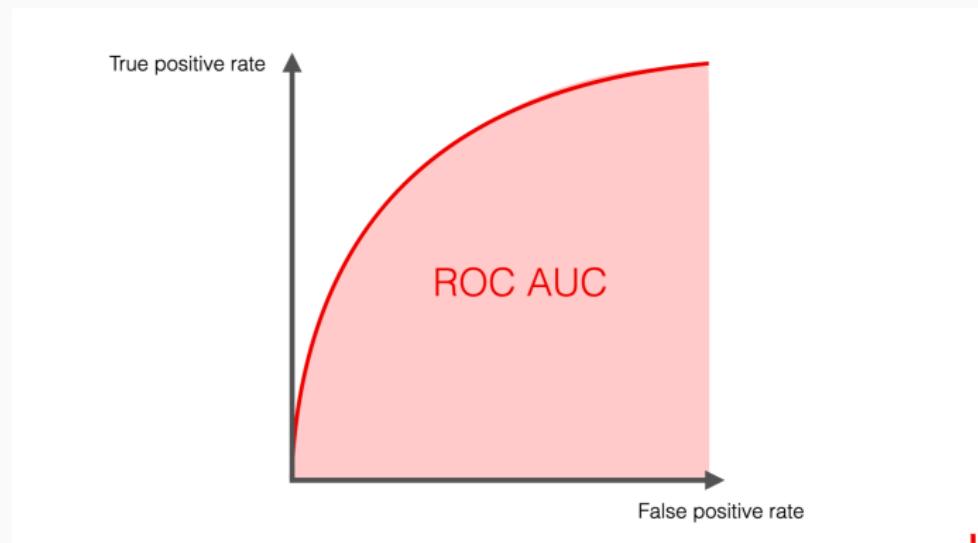
내전의 시작이라는 0-1 이항변수를 확률로 예측했을 때
보통 사용하는 decision threshold = 0.5

- 즉 0.5 이상이면 내전이 난다고 예측하고, 미만이면 안 난다고 예측
- 경우에 따라 decision threshold를 옮길 수도 있음
 - “20%만 되어도 내전이 난다고 예측해보자”
 - “80%나 되어야 내전이 난다고 예측해보자”

분류의 성과평가: AREA UNDER THE CURVE (AUC)

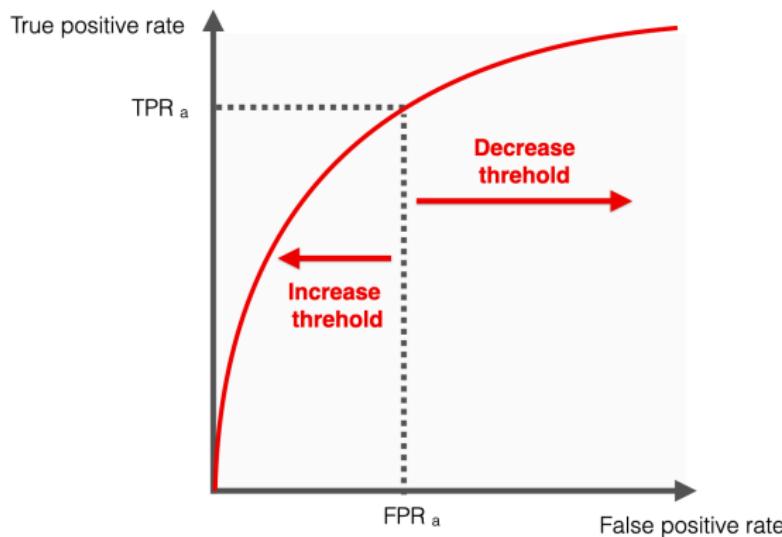
Area under the curve (AUC)

- 보통 receiver-operator characteristic (ROC) 곡선을 하나의 숫자로 표현한 지표
- (경우에 따라 precision-recall curve를 사용한 PRAUC도 있음)

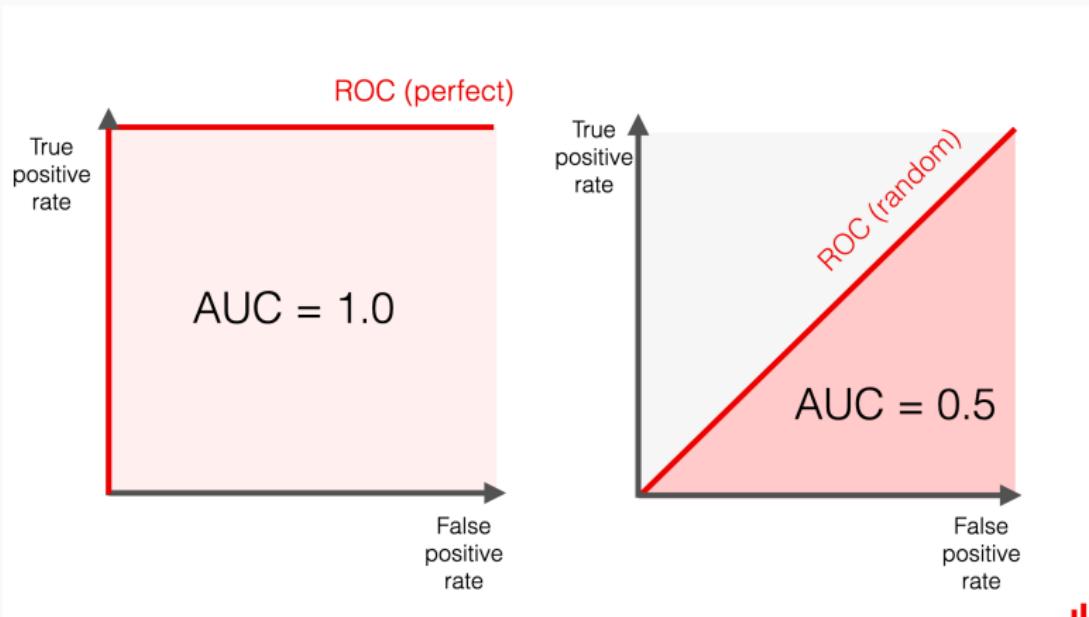


출처: [How to explain the ROC curve and ROC AUC score?](#)

분류의 성과평가: AREA UNDER THE CURVE (AUC)



분류의 성과평가: AREA UNDER THE CURVE (AUC)



지도 학습(Supervised Learning)

지도학습에서 표본외 예측

$Y = f(X) + \epsilon$ 이라고 할 때 데이터를 두 개로 쪼갬

- **훈련 데이터(training set)**

- X 도 Y 도 “아는(known)” 상태
- 이 데이터를 사용해서 X 와 Y 의 관계를 학습

- **테스트 데이터(test set)**

- X 만 “아는” 상태
- 앞서 훈련한 모형을 사용해서 X 를 집어넣고 Y 를 예측
- 예측의 성과를 평가할 수 있음

훈련 오차(TRAINING ERROR) vs. 검정 오차(TEST ERROR)

- **훈련 오차(Training error) =**

Training set에 통계적 학습법을 사용해서 쉽게 계산할 수 있음

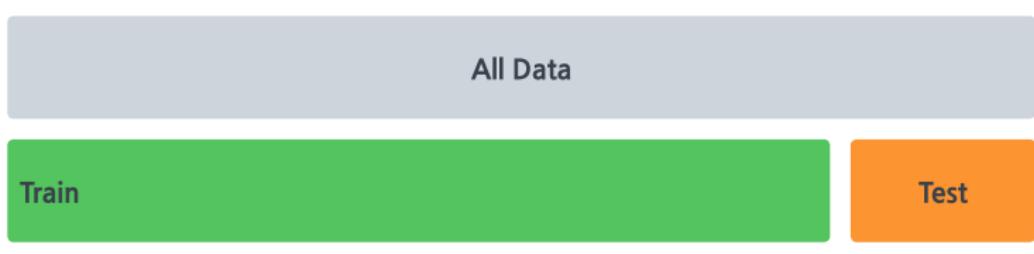
- **검정 오차(Test error) =**

Test set에 없는 새로운 관측치에 대해 예측했을 때 평균적으로 발생하는 오차

- 보통 둘은 상당히 다름

훈련 오차는 보통 검정 오차를 심각하게 과소평가(underestimate)할 수 있음

훈련-테스트 패러다임(TRAINING-TESTING PARADIGM)



- (Validation set approach라고 ISLR에서 지칭)
- 2장에서 배운 회귀분석 때는 회귀분석 + 표본내 예측이었기 때문에 Test set이 따로 존재하지 않음
- **Training set에만 모형을 적합시킴**
(이 모형 자체가 회귀분석일 수도 있음; 선형 계수 추정)
- Test set, 또는 hold-out set의 X 를 모형에 넣고 예측
- 80:20이나 50:50 split을 흔히 사용하지만 목적과 데이터의 성격에 따라 다름
- 테스트 데이터가 어떻게 선정되느냐에 따라서 편향(bias)이 심하게 발생할 수 있음

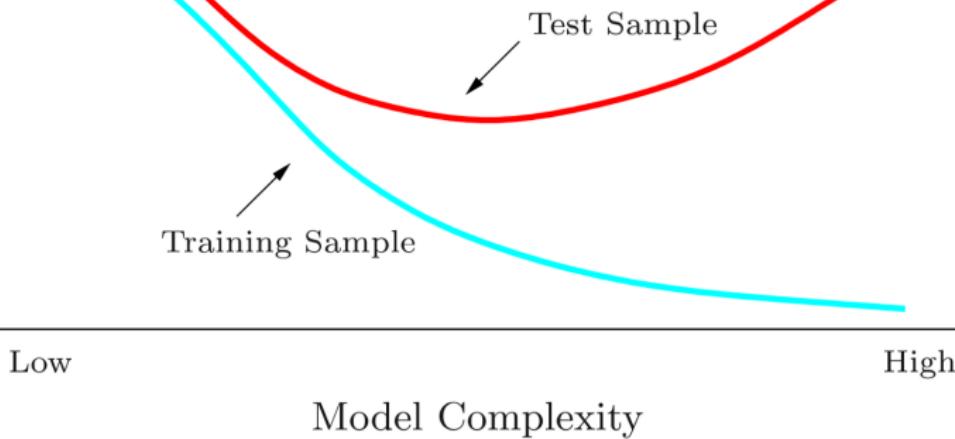
훈련-테스트 데이터 쪼개기

```
1 library(tidymodels)
2 library(tidyverse)
3 library(ranger)
4 data(diamonds)
5
6 ## For our purposes, remove the ordering from factors
7 diamonds$cut <- factor(diamonds$cut, ordered = FALSE)
8 diamonds$color <- factor(diamonds$color, ordered = FALSE)
9 diamonds$clarity <- factor(diamonds$clarity, ordered = FALSE)
10
11 ## Split to training-testing data
12 set.seed(123)
13 data_split <- initial_split(diamonds, prop = 0.8)
14 df_train <- training(data_split)
15 df_test <- testing(data_split)
16
17 nrow(diamonds)
18 nrow(df_train)
19 nrow(df_test)
```

Prediction Error

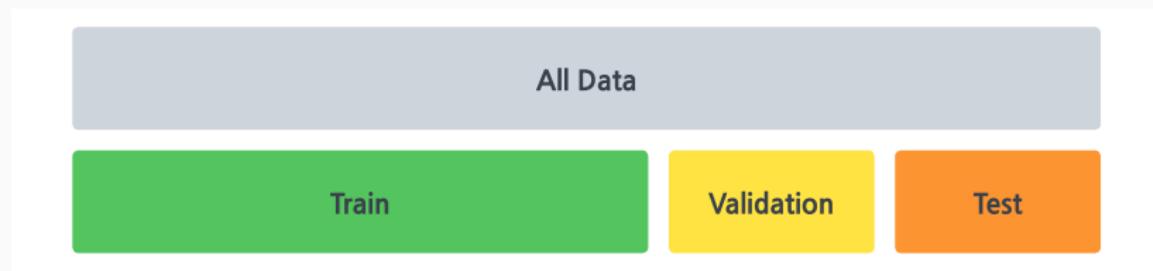
High Bias
Low Variance

Low Bias
High Variance



훈련-검증-테스트

모형 파라미터(parameter)를 Validation set을 이용해서 튜닝(tuning)



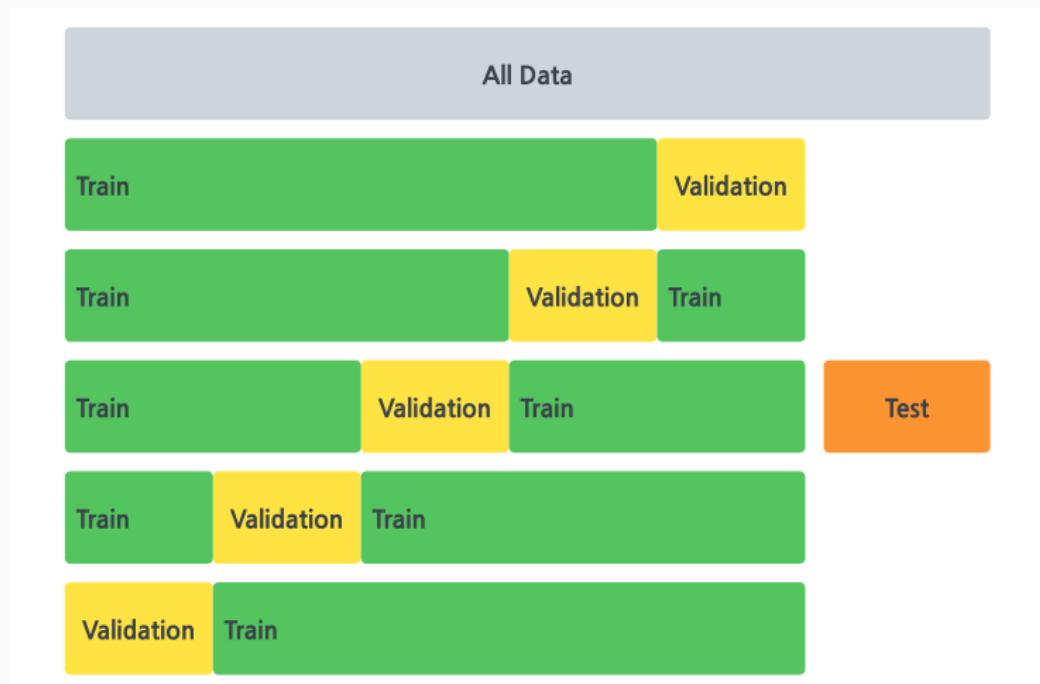
- 가끔 책이나 사람따라 test set과 validation set 용어를 섞어쓰는 경우가 있음
ISLR도 Chapter 5.1에 test set을 validation set이라고 지칭
- 우리는 ESL에서 쓴 것처럼 Train-validation-test를 구분할 예정
모형 선택(model selection)과 모형 평가(model assessment)는 다른 목적
- 문제 = 데이터가 많이 필요함
전체 데이터를 사용했을 때보다 훈련 오차가 과대평가될 수 있음

교차 검증(CROSS-VALIDATION)

데이터의 정보를 최대한 활용하기 위한 방안

- 데이터가 충분치 못할 때는 validation set을 따로 쓰기 어려울 수 있음
- 5-fold 또는 10-fold cross validation을 많이 쓰고
Leave-one-out cross-validation (LOOCV)도 자주 사용
- Test data를 제외하고 데이터를 K개로 쪼慨
- 그 중 1개를 Validation set으로 설정하고 나머지 K-1개는 Training set으로
- 모형을 튜닝하고 그 추정치들을 합치거나(combine) 모형을 선택
- Test set에 다시 테스트 (중요! Test set은 절대로 훈련에 사용되지 않음)
Hold-out set이라고도 함
- 계산의 양이 많아짐

교차 검증(CROSS-VALIDATION)



5-fold validation을 시각화

교차 검증(CROSS-VALIDATION)

- K-fold cross validation이라고 했을 때
쪼개진 데이터를 C_1, C_2, \dots, C_K 라고 하자
- k -파트에는 n_k 개의 관측치가 있음
- 만약 n 이 K 의 배수이면 $n_k = n/K$
- 다음을 계산

$$CV_{(K)} = \sum_{k=1}^K \frac{n_k}{n} MSE_k$$

이 때 $MSE_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$ 이고 \hat{y}_i 는 k 번째 데이터를 제끼고 계산했을 때 i 번째 관측치의 적합값

- 만약 $K = n$ 이라면 n-fold 또는 leave-One-Out Cross-Validation (LOOCV)
이는 편향은 적지만 분산이 큰 방법

회귀분석과 교차검증

```

1 model <- linear_reg(mode = "regression") %>%
2   fit(price ~ carat + cut + color + clarity, data = diamonds_train)
3 summary(model$fit)

```

결과 (일부 잘림)

```

1 Call :
2 stats :: lm(formula = price ~ carat + cut + color + clarity , data = data)
3

```

4 Residuals:

	Min	1Q	Median	3Q	Max
	-16834.6	-678.1	-195.8	467.3	10424.9

7

8 Coefficients :

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-7329.59	57.88	-126.63	<2e-16 ***
carat	8879.93	13.38	663.47	<2e-16 ***
cutGood	645.75	37.49	17.22	<2e-16 ***
cutVery Good	858.14	34.85	24.62	<2e-16 ***
cutPremium	878.95	34.47	25.50	<2e-16 ***
cutIdeal	1005.20	34.16	29.43	<2e-16 ***
colorE	-227.78	20.35	-11.19	<2e-16 ***
colorF	-310.05	20.62	-15.04	<2e-16 ***
colorG	-515.47	20.12	-25.62	<2e-16 ***
colorH	-975.23	21.42	-45.52	<2e-16 ***

지도학습에서 흔히 쓰는 모형

사실 가르쳐야 할 게 너무 많음!

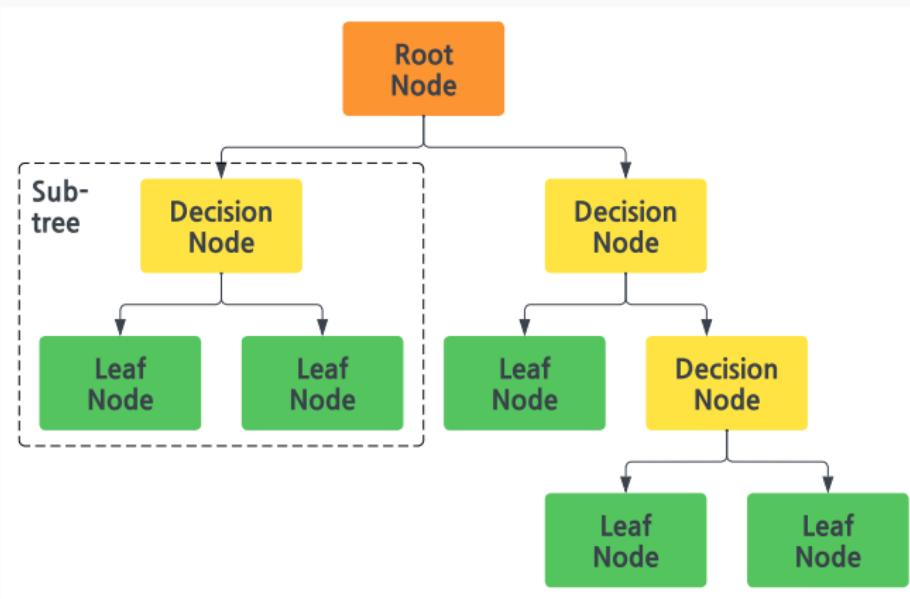
- 선형회귀 + 머신러닝 =
best subset selection, shrinkage approaches (LASSO, ridge), ...
- Support vector machine
- Linear discriminant analysis
- Gradient boosting
- Boosting

지도학습에서 흔히 쓰는 모형

(진도를 고려해서) 우선은 **트리 기반(tree-based) 통계적 학습 기법**을 학습할 예정

- 분류에도 예측에도 사용 가능
- 기본 = 예측 공간(predictor space)을 계층화(stratifying)하거나 세분화(segmenting)하는 것
- 예측 공간을 쪼개는 **분리 규칙(splitting rule)**들은 하나의 나무(tree)로 표현할 수 있음
- **의사결정 나무(decision tree)** 또는 **의사결정 트리**라고 표현

의사결정 나무(DECISION TREE)



Leaf node 대신 terminal node라고도 부름

의사결정 나무

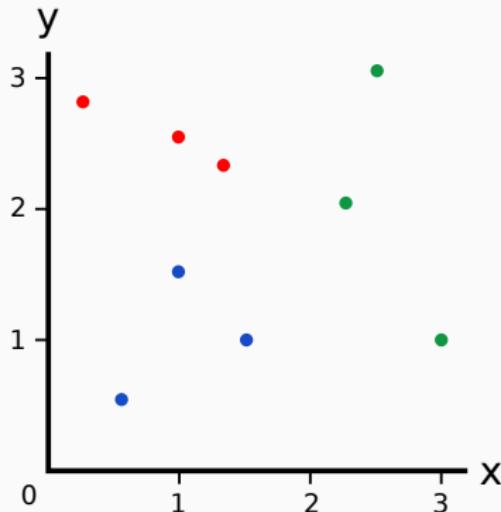
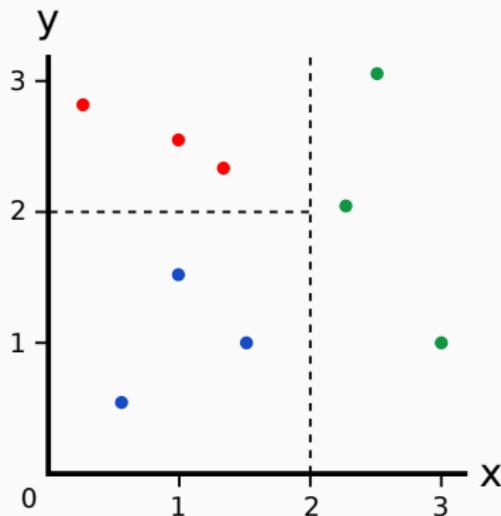


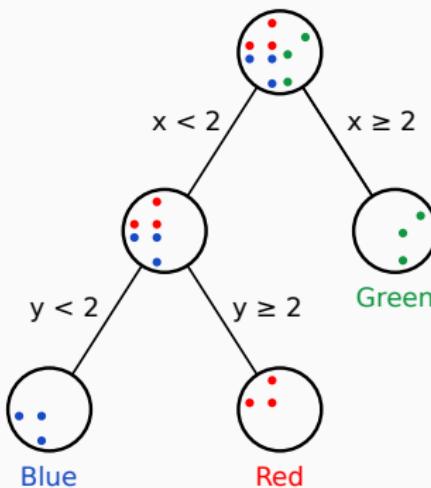
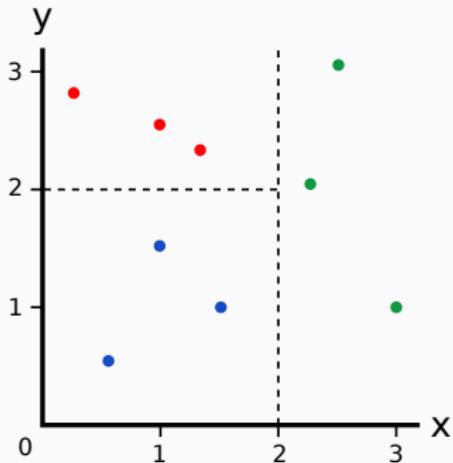
그림 출처: <https://victorzhou.com/intro-to-random-forests/>

의사결정 나무



- 빨간 점들이 있는 구역을 R_1 , 파란 점 구역을 R_2 , 초록 점 구역을 R_3 라고 하자
서로 겹치지 않지만 포괄적인 구역들(region)
- 해당 구역에 떨어지는 모든 점들에 대해 동일하게 예측

의사결정 나무



트리 모형 관련 용어

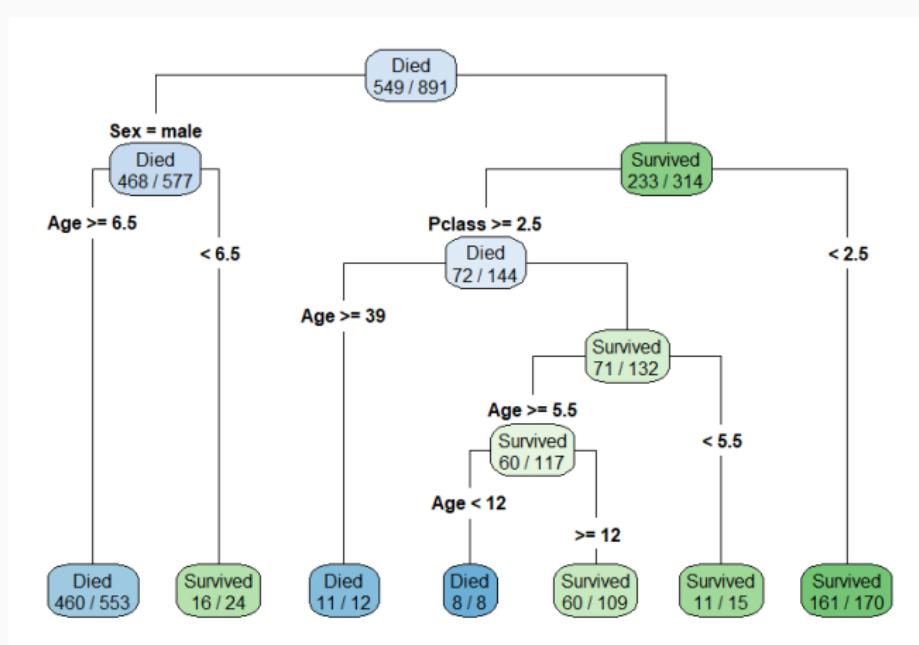
나무를 거꾸로 그리는 것

- 분리가 일어나는 부분은 internal node
최종적으로 남은 것들은 **terminal node**
- 조건부 확률이라고 생각하면 됨
- 가장 처음 시작하는 **root node** = 가장 데이터를 잘 분리하는 기준

의사결정 나무(DECISION TREE): 타이타닉 생존자 분류

```
1 library(tidymodels)
2 library(tidyverse)
3 library(titanic)
4 library(rpart.plot)
5
6 ## https://www.kaggle.com/c/titanic/data
7 titanic_train$Survived <- factor(
8   titanic_train$Survived, levels = c(0, 1),
9   labels = c("Died", "Survived"))
10 )
11 model <- decision_tree(mode = "classification") %>%
12   fit(Survived ~ Pclass + Sex + Age, data = titanic_train)
13
14 model$fit %>% rpart.plot(type = 4, extra = 2, roundint = FALSE)
15 predict(model, titanic_test)
16 predict(model, titanic_test, type = "prob")
```

의사결정 나무: 타이타닉 생존자 분류



스무고개를 하는 것,
또는 자식들이 동질성을 목표로 분가하는 것이라고 생각하면 쉬움

트리 기반 모형의 장단점

- 각 구역들(R_1, \dots, R_j)이 직사각형의 박스 모양을 띠게 됨
(비교적) 간단하고 해석이 유용
- 목표: RSS를 최소화**

$$\sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

이 때 \hat{y}_{R_j} 는 j 번째 박스 내 훈련용 관측치들의 평균적인 결과(response) 값

- 가장 최신 모형들보다는 예측 정확성이 떨어지긴 함
- 여러 트리를 모으면 예측 정확성이 극적으로 좋아지지만
대신 해석이 좀 더 어려워짐 e.g., 랜덤 포레스트

근데 대체 분할을 어떻게 하지?

Feature space의 모든 가능한 분할방법을 고려하기엔 계산이 불가능
그래서 대신 재귀적 분리(recursive binary splitting)

- **하향식(Top-down)**

- 트리의 맨 위에서 시작해서 연속적으로 예측공간을 분할
- 각 분할은 노드에서 뻗어나온 새로운 가지로 표현

- **탐욕(Greedy) 알고리즘**

- 미래에 더 나은 트리가 되는 것을 내다보지 않고
당장 지금 노드에서 최선인 형태로 데이터를 분리
- 따라서 결과적으로 최적의 트리가 아닐 수 있음

가지치기(PRUNING)

- 문제는 이런 과정에서 Training Data에 과적합시킬 가능성이 있음 → 막상 Test Data에서는 성과가 나쁠 수도
- 트리가 단순하면 약간의 편향을 감수하고 분산 및 해석이 개선될 수 있음
- 그렇다고 나무를 작게 키우는 것은 근시안적일 수 있음. 당장 RSS가 작게 줄어든다 해서 나중에 RSS 감소에 도움이 되지 않는다는 뜻은 아님

가지치기(PRUNING)

- 큰 트리(T_0)를 키운 후 가지치기를 해서 서브트리(subtree)를 만드는 전략이 더 효율적
- Cost complexity pruning** 또는 weakest link pruning
- 튜닝 파라미터 α 의 값마다

$$\sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

를 최소화하는 서브트리 $T \subset T_0$ 가 있음

- $|T| = T$ 트리의 터미널 노드의 수
- $R_m = m$ 번째 터미널 노드에 해당하는 예측 공간의 직사각형(rectangle)
- $\hat{y}_{R_m} = R_m$ 의 트레이닝 데이터의 종속변수의 평균
- $\alpha =$ 적합성과 복잡함 사이의 트레이드오프를 컨트롤
- CV를 통해 최적의 $\hat{\alpha}$ 를 선택 → 해당하는 서브트리 최종 선택

분류 나무(CLASSIFICATION TREES)

- 회귀 나무(regression tree)와 거의 비슷
- 양적 결과 대신 질적(qualitative) 결과를 예측
- 각 관측치는 그것이 속한 region의 트레이닝 데이터에서 가장 일반적으로 발생하는 클래스(class)에 속한다고 예측
- 분류와 회귀를 통틀어 **CART (classification and regression tree)**라고도 함

분류 나무(CLASSIFICATION TREES)

- 회귀때와는 달리 RSS를 사용할 수 없음
- Classification error rate를 사용할 수 있음

$$E = 1 - \max_k(\hat{p}_{mk})$$

- 가장 일반적인 클래스에 속하지 않는 training observation의 비율
- $\hat{p}_{mk} = m$ 번째 region에서 k 번째 클래스에 속하는 training obs.의 비율
- 하지만 tree-growing을 할 때는 classification error는 충분히 민감(sensitive)하지 않음
- 그래서 보통은 **지니 계수(Gini index)**
또는 **크로스 엔트로피(cross-entropy)**를 사용

지니 계수와 크로스 엔트로피

지니 계수

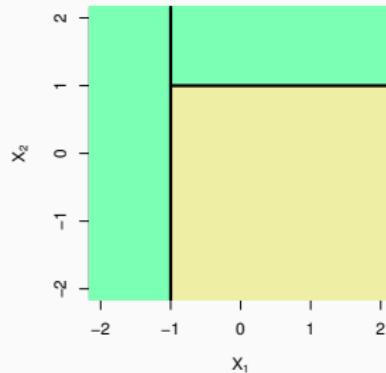
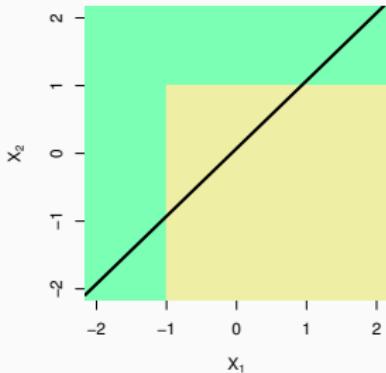
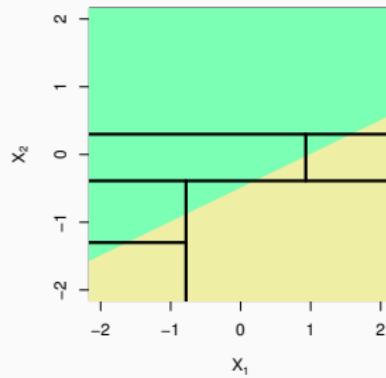
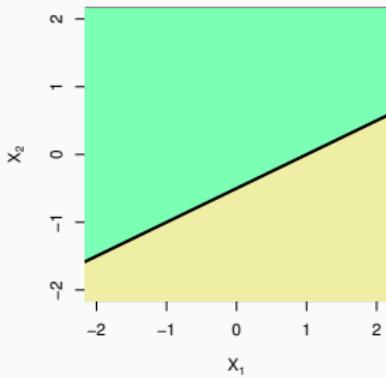
$$G = \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk})$$

- K 클래스 전체에 걸친 총 분산(total variance)의 측도
- \hat{p}_{mk} 가 전부 0이나 1에 가까우면 최소화될 수 있음
- 따라서 노드의 순도(purity)의 표현
값이 작다면 노드의 관측치들이 거의 한 가지 클래스에 속해있다는 이야기가 됨

지니 계수 대신 사용할 수 있는 크로스 엔트로피

$$D = - \sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk}$$

트리 vs. 선형 모형



트리 모형의 장단점

- (외외로) 선형 모형보다 사람들에게 설명이 쉬움
- 선형회귀보다 인간의 의사결정에 가깝다고 여겨질 수 있음
- 쉽게 시각화할 수 있고 전문가가 아니라도 해석이 쉬움
- 더미 변수를 생성하지 않고도 질적 변수를 다룰 수 있음
- 예측 정확도는 좀 떨어지지만 여러 트리를 종합해서 예측을 개선할 수 있음

배깅(BAGGING)

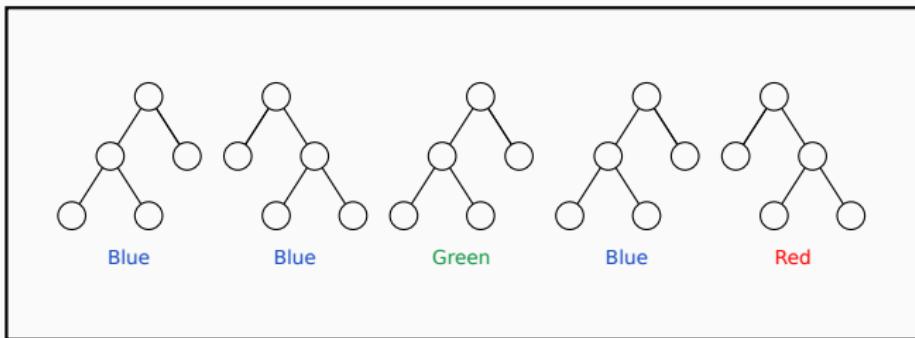
Bootstrap aggregation 또는 bagging

- **앙상블(ensemble)**의 일종. 즉 하나의 강력한 모델을 얻기 위해 단순한 “building block” 모델 여러 개를 결합하는 접근법
- Z_1, \dots, Z_n 이라는 독립적인 관측치들이 각각 분산이 σ^2 라고 하면 표본평균 \bar{Z} 의 분산은 σ^2/n 임. 이렇듯이 관측치의 평균을 내는 것은 분산을 줄이게 되어 있음
- **Booststrapping** = 하나의 training dataset에서 반복적으로 샘플을 추출하기 Random sampling with replacement
- B 개의 부트스트랩 훈련 데이터를 생성한 후 b 번째 데이터에 대해 x 에서의 예측인 $\hat{f}^{*b}(x)$ 를 계산, 최종적으로 평균

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^{*b}(x)$$

다수결 투표(MAJORITY VOTE)

Classification tree에 대해서는 B 개 나무가 각각 예측한 클래스를 기록하고
다수결 투표로 예측하는 클래스를 선택하게 함



Blue

BAGGING의 오차는 어떻게 계산하나?

- **부트스트래핑(bootstrapping)** = n 개의 관측치에 대해서 전체 관측치의 수, 즉 n 의 크기로 복원추출
- 한 개의 관측치가 한 번 샘플링에서 안 뽑힐 확률은

$$\left(1 - \frac{1}{n}\right)^n$$

그 극한은

$$\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n = \frac{1}{e}$$

한 번의 샘플에서 $1 - \frac{1}{e} \approx 0.632$, 즉 평균적으로 63.2%의 관측치를 사용

- **Out-of-bag (OOB) observations** = 주어진 bagged tree를 적합시키는데 사용되지 않은 나머지 $1/3$ 정도의 관측치
- 특정 관측치가 OOB였을 때의 예측의 평균을 구함
(즉 $B/3$ 개 정도의 predictions)

랜덤 포레스트(RANDOM FORESTS)

Random forests = Bagging에서 분산을 줄이는 방향으로 개선시킨 것

- 트리들의 상관성을 줄이면(decorrelate) 트리들의 평균을 구했을 때 분산이 줄어듦
- Decision tree를 만들 때 전체 p 개 예측변수를 사용하기보단 m 개 예측변수를 무작위로 뽑은 후 그 중에서만 다음 split을 고려
- 보통 $m \approx \sqrt{p}$ 를 사용
- (Forests라고 하기도 하고 forest라고 하기도 하고. 아무 상관 없음)
- 따로 결측치처리를 할 필요 없이 결측치(missing value)를 **하나의 범주로 다룰 수 있음**

(시간상 boosting은 생략)

랜덤 포레스트: 시청각 자료

<https://youtu.be/v6VJ2RO66Ag?feature=shared>

선형회귀, 의사결정나무, 랜덤 포레스트

```
1 ## parsnip::linear_reg
2 model_lm <- linear_reg(mode = "regression") %>%
3   fit(price ~ carat + cut + color + clarity, data = df_train)
4
5 ## parsnip::decision_tree
6 model_tree <- decision_tree(mode = "regression") %>%
7   fit(price ~ carat + cut + color + clarity, data = df_train)
8
9 ## parsnip::rand_forest
10 model_rf <- rand_forest(mode = "regression") %>%
11   fit(price ~ carat + cut + color + clarity, data = df_train)
12
13 ## 테스트데이터
14 out <- bind_cols(
15   predict(model_lm, new_data = df_test) %>% rename(pred_lm = .pred),
16   predict(model_tree, new_data = df_test) %>% rename(pred_tree = .pred),
17   predict(model_rf, new_data = df_test) %>% rename(pred_rf = .pred),
18   df_test
19 )
20
21 ## MSE
22 sum((out$price -out$pred_lm)^2)
23 sum((out$price -out$pred_tree)^2)
24 sum((out$price -out$pred_rf)^2)
```

모형 튜닝하기

튜닝(Tuning) =

모델 예측 향상을 위해 하이퍼 파라미터(hyperparameter)를 조정하기

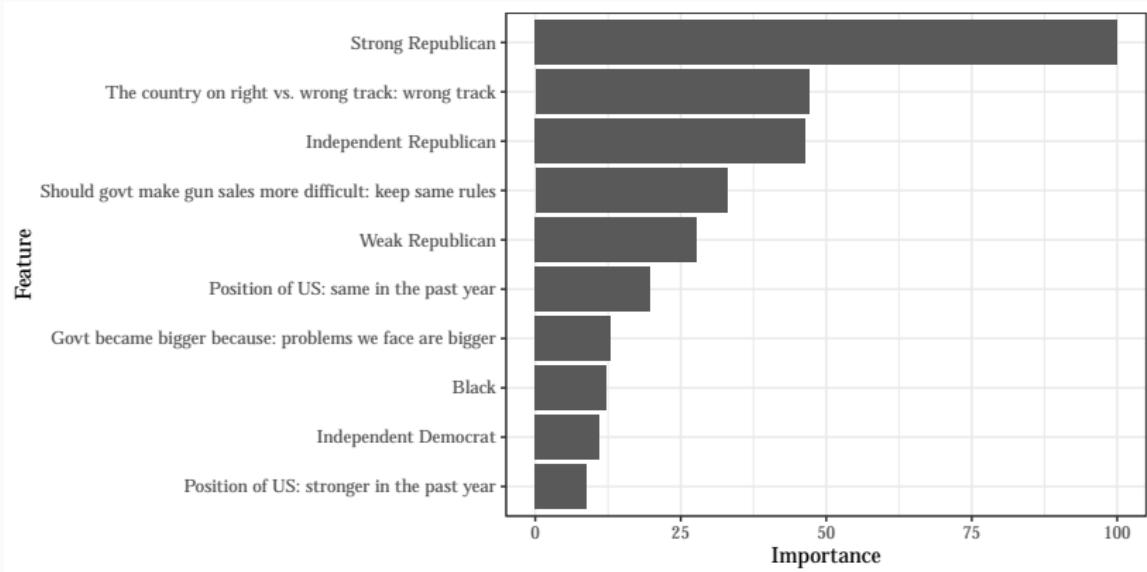
- m 의 계산 방식
- 나무를 몇개나 만들지 등등

(현실적으로 이게 크게 결과를 바꾸진 않음)

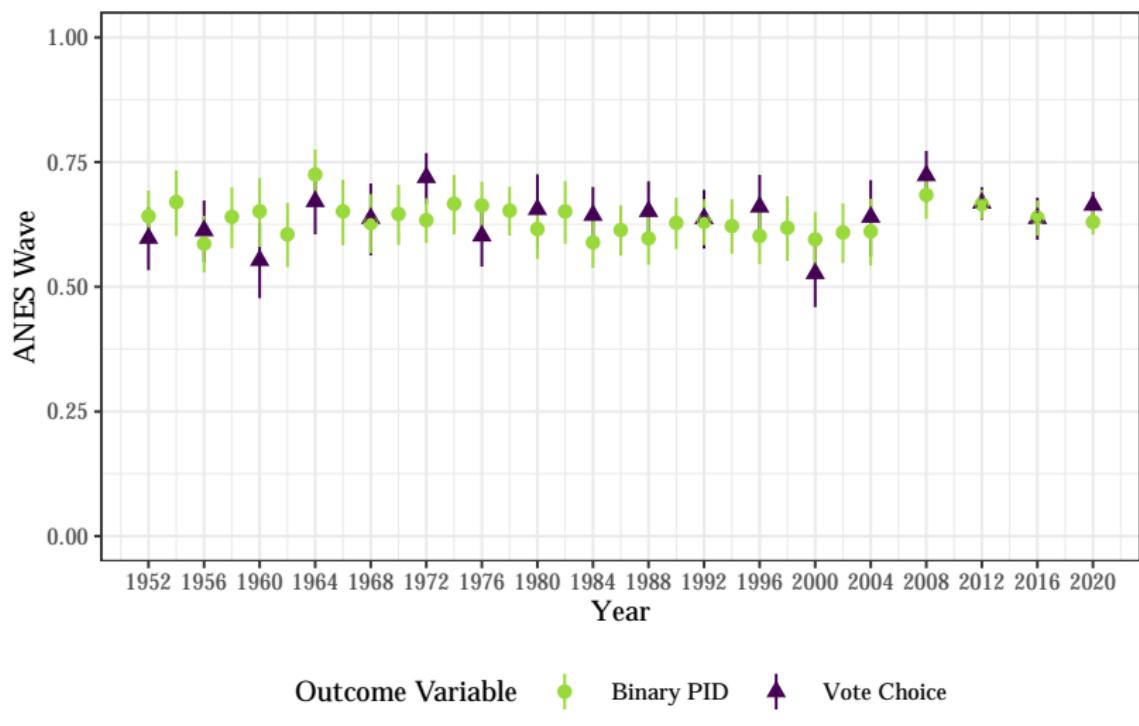
변수 중요도 측도(VARIABLE IMPORTANCE MEASURE)

- B 개 나무에 대해 평균적으로 어떤 설명변수를 사용해서 split했을 때 얼마나 RSS나 Gini index가 줄어드는가?
- 다른 방법으로는 한 개의 예측 변수를 일부러 무작위로 섞어서 (randomly shuffle 또는 **permute**) 한 후 예측 오차가 얼마나 커지는지 관측
- 선형 회귀처럼 해석할 수 없음
주의해서 해석해야 함

예시



2016 ANES Presidential Vote Choice Prediction (United States)



Kim, Seo-young Silvia, and Jan Zilinsky. 2022. “Division Does Not Imply Predictability: Demographics Continue to Reveal Little About Voting and Partisanship.” Political Behavior. First view online.

비지도 학습(Unsupervised Learning)

비지도 학습의 목적과 장단점

- 다시 한 번 말하지만 결과 변수가 있는 것이 아님
- **측도(measurement)에 대해 흥미로운 패턴을 찾는 것이 목적**
 - e.g., 정보 전달이 잘 되는 시각화 방식이 있는지, 서브그룹(subgroup)이나 클러스터를 찾을 수 있는지
e.g., 이 영화에 5점 만점에 5점을 준 사람들의 공통적인 특징은 무엇일까?
 - 지도학습에 비해 좀 더 주관적(subjective)일 가능성이 있지만 몹시 중요한 분야
 - Label data는 노동이 필요하기 때문에 unlabeled data를 구하는 것이 더 쉬움

주성분 분석(PRINCIPAL COMPONENT ANALYSIS, PCA)

- 차원(dimension) 축소 방법: 고차원 → 저차원
- 변수의 선형 조합(linear combination)들을 서로 상관성이 없도록 (uncorrelated) 만들면서도 최대한의 분산(maximal variance)을 보존
- 데이터 시각화에서도 사용됨

주성분 분석(PRINCIPAL COMPONENT ANALYSIS, PCA)

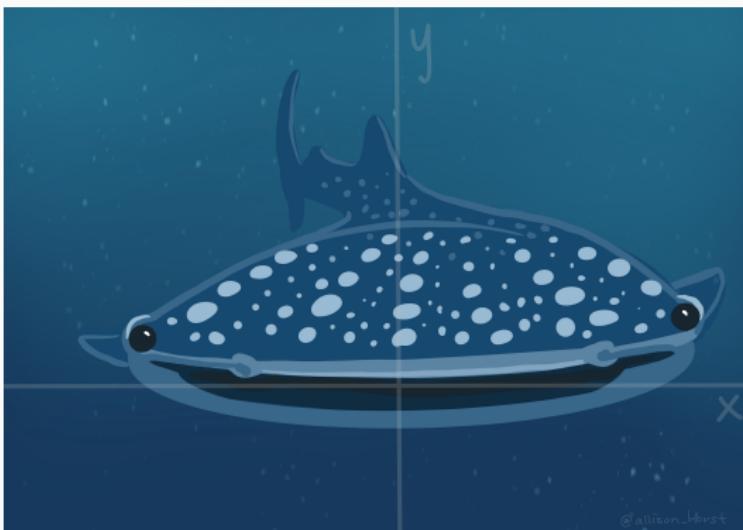


Image credit: [Allison Horst](#)

주성분 분석(PRINCIPAL COMPONENT ANALYSIS, PCA)



- 이 데이터의 분포를 가장 잘 설명할 수 있는 두 개의 벡터는?
- 최대한의 분산을 최소한의 변수로
- 축(axis)을 회전(rotate)시키는 것

주성분 분석(PRINCIPAL COMPONENT ANALYSIS, PCA)

- 첫 번째 주성분(first principal component)은 분산을 최대한 보존

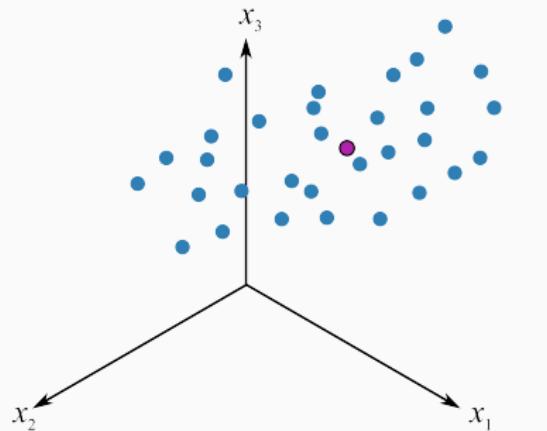
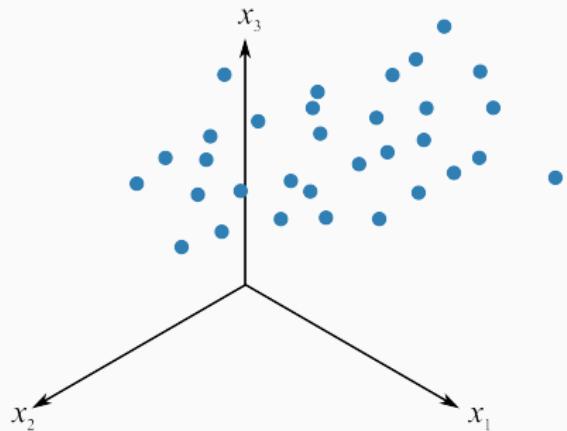
$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$$

- 이 때 선형 조합은 정규화(normalized)됨: $\sum_{j=1}^p \phi_{j1}^2 = 1$
- 계산법

$$\max_{\phi_1} \frac{1}{n} \sum_{i=1}^n \left(\sum_{j=1}^p \phi_{j1} X_{ij} \right)^2 \text{ subject to } \sum_{j=1}^p \phi_{j1}^2 = 1$$

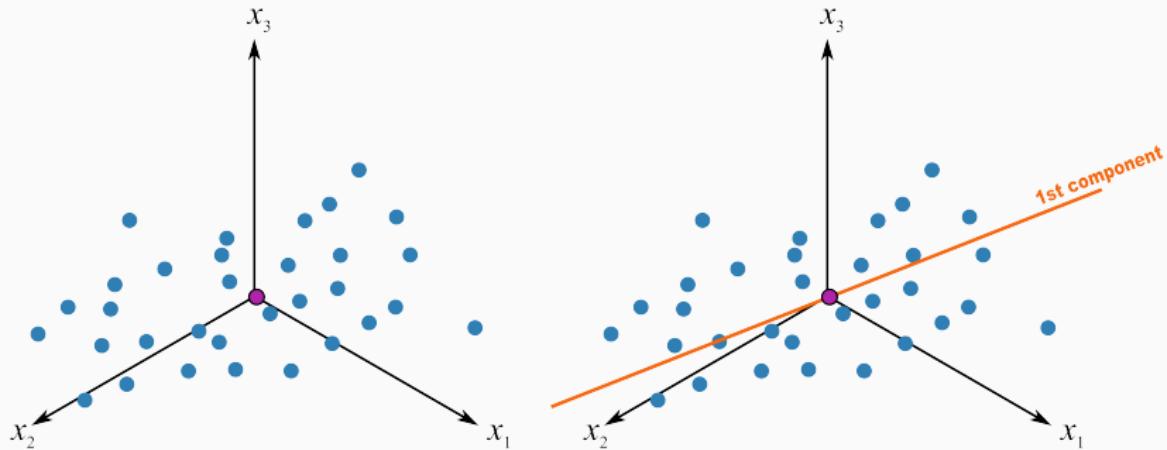
- $\phi_{11}, \dots, \phi_{p1}$ 을 첫 번째 주성분의 로딩/loading이라고 부름
주성분 로딩 벡터 $\phi_1 = (\phi_{11}, \phi_{21}, \dots, \phi_{p1})^T$
- 그 다음 주성분은 첫 번째 주성분과 상관성이 없어야 함: 직교(orthogonal)
모든 성분들은 서로 상관성이 없음(uncorrelated)

PCA 시각화

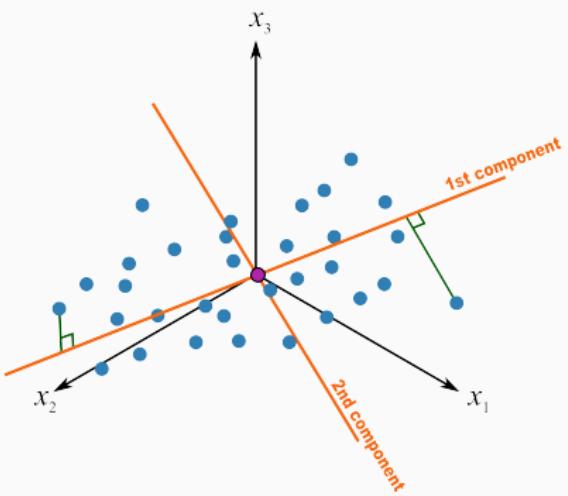
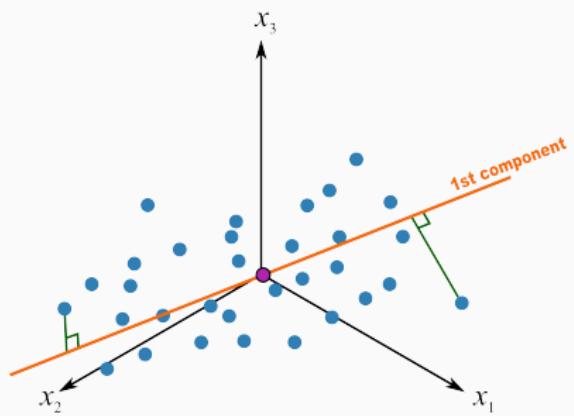


출처: Process Improvement Using Data

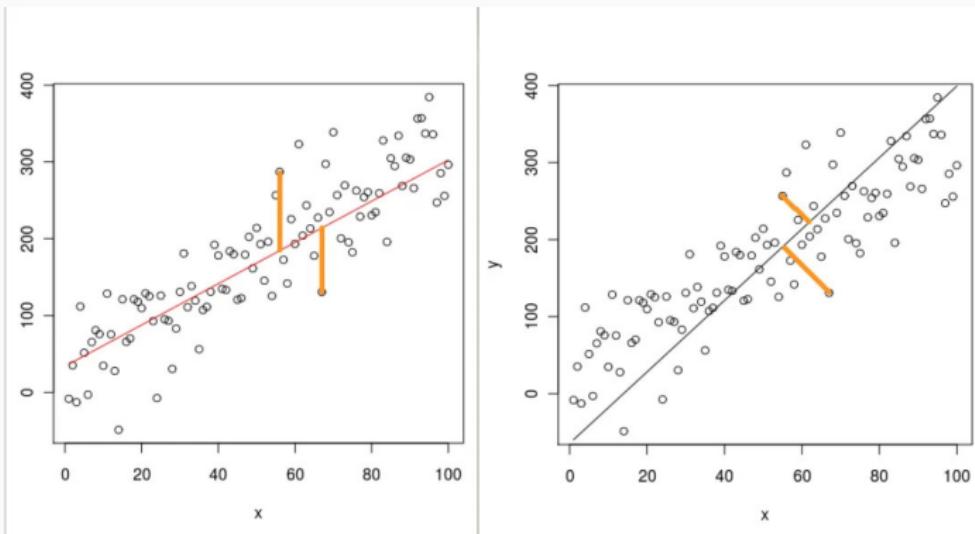
PCA 시각화



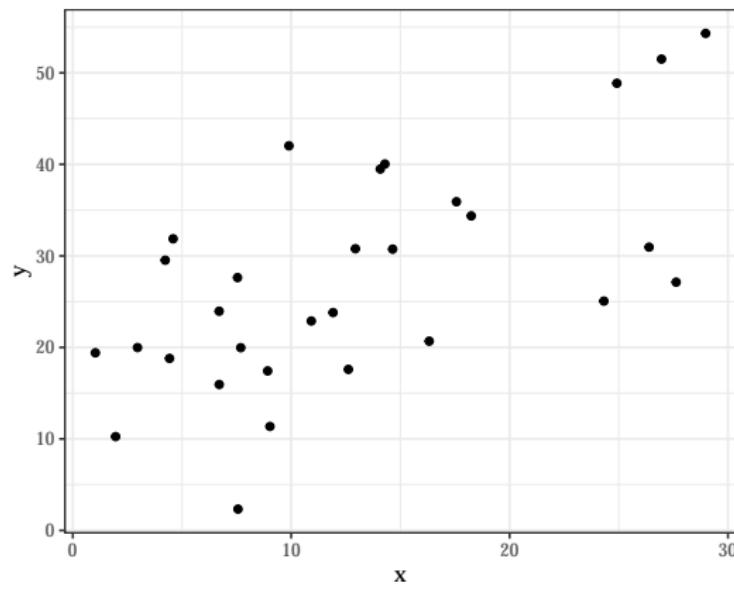
PCA 시각화



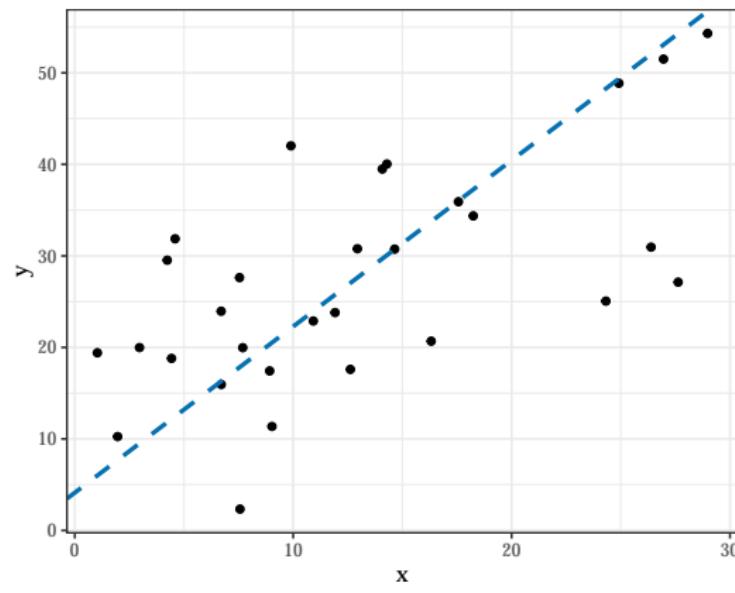
PCA REGRESSION VS. LEAST SQUARES



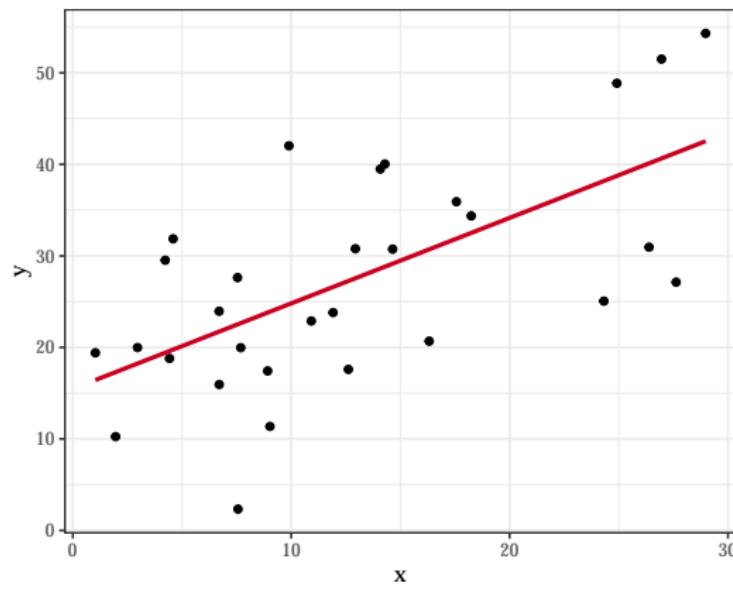
PCA REGRESSION VS. LEAST SQUARES



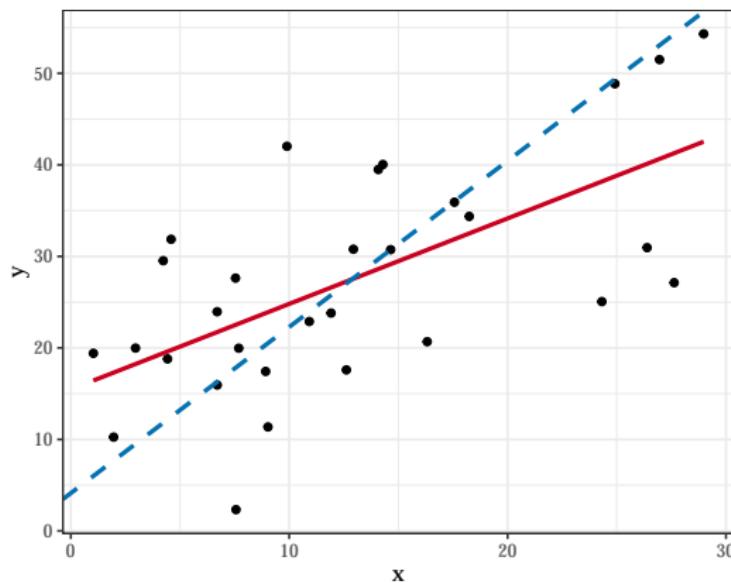
PCA REGRESSION VS. LEAST SQUARES



PCA REGRESSION VS. LEAST SQUARES



PCA REGRESSION VS. LEAST SQUARES



See also: https://twitter.com/page_eco/status/1094267994002391041

실습

Posit Cloud 참조

클러스터링(CLUSTERING), 또는 군집 분석

- 각 그룹 내의 관측치가 서로 유사하도록 데이터를 별개의 그룹으로 분할
- 두 개 관측치가 유사(similar)하거나 다르다(different)는 것은 무슨 뜻인지 먼저 명확하게 정의해야 함
- 예시: 특정 광고에 대해 더 효과적으로 반응하는 소비자들은 누구일까?
시장 세분화(market segmentation)

여러 방법이 있지만 이 수업에서는 간단하게

K-평균 군집분석(K-means clustering)만 배울 예정

K-MEANS CLUSTERING

- C_1, \dots, C_K 를 각 클러스터에 들어있는 관측치들의 인덱스(index)라고 하면
exhaustive and exclusive
 - $C_1 \cup C_2 \cup \dots \cup C_K = \{1, \dots, n\}$: 각 관측치는 K 클러스터 중 하나에 소속
 - $k \neq k'$ 일 때 $C_k \cap C_{k'} = \emptyset$ 즉 겹치지 않음
- i 번째 관측치가 k 번째 클러스터 소속이면 $i \in C_k$
- **클러스터 내 분산(within-cluster variation)**이 최소화되는 클러스터가 좋은 것
- 클러스터 C_k 에 대해 WCV(C_k)를 정의한 후

$$\min_{C_1, \dots, C_K} \sum_{k=1}^K \text{WCV}(C_k)$$

이 때 보통 WCV에는 유클리드 거리(Euclidean distance)를 사용

$$\text{WCV}(C_k) = \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2$$

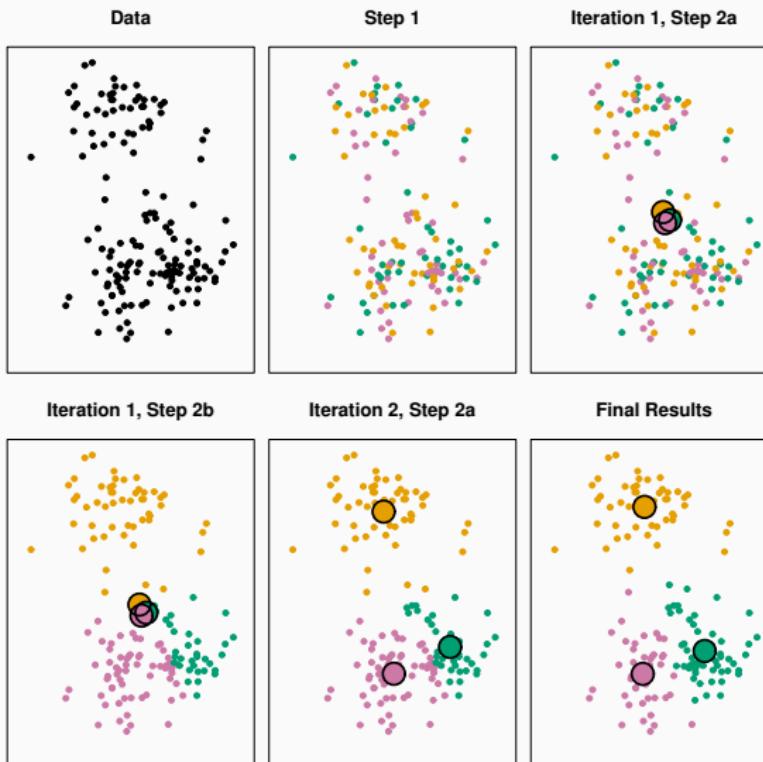
이 때 $|C_k|$ 는 k 번째 클러스터에 속한 관측치의 수

K-MEANS CLUSTERING

1. 일단 초기에는 무작위로 관측치에 1에서 K의 숫자를 부여 =
initial cluster assignment
2. 다음을 cluster assignment가 바뀌지 않을 때까지 반복
 - 2.1 K개 클러스터에 대해 클러스터의 **중심(centroid)**를 계산
(p 개 feature의 평균)
 - 2.2 가장 가까운 centroid에 따라 cluster를 변경

도심(centroid)는 반드시 데이터 포인트 중 하나일 필요는 없음

K-MEANS CLUSTERING



애니메이션

- <https://gallery.shinyapps.io/050-kmeans-example/>
- <http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html>

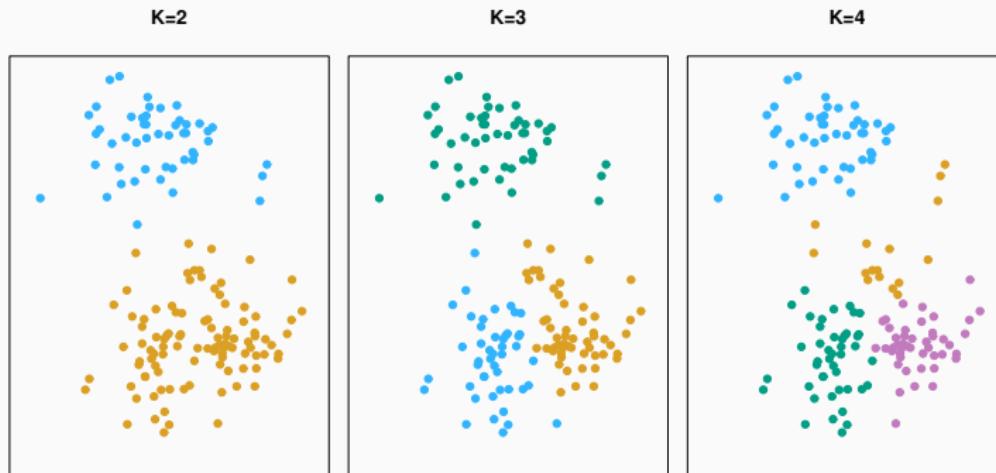
```
1 library(cluster)
2 library(factoextra)
3
4 data("iris")
5 head(iris)
6 features <- iris[, 1:4]
7
8 # Visualize
9 fviz_cluster(kmeans(features, centers = 3, nstart = 10), data = features)
```

주의해야 할 점

- 초기 할당에 따라 결과가 조금씩 달라질 수 있음
- 변수의 단위, 즉 스케일(scale)에 따라 결과가 달라질 수 있음

주의해야 할 점

미리 K 를 정해주어야 하는데, 몇 개의 주성분을 쓸지 어떻게 결정?
즉 최적의 군집수는 얼마인가를 정해야 하는데, 어떻게 정해야 할까?



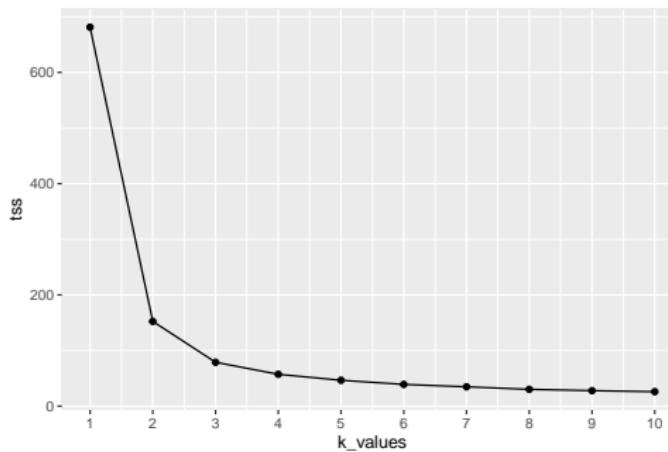
“정답” label이 없는 경우 답하기 쉽지 않음 + cross-validation 불가능

최적 군집수 정하기

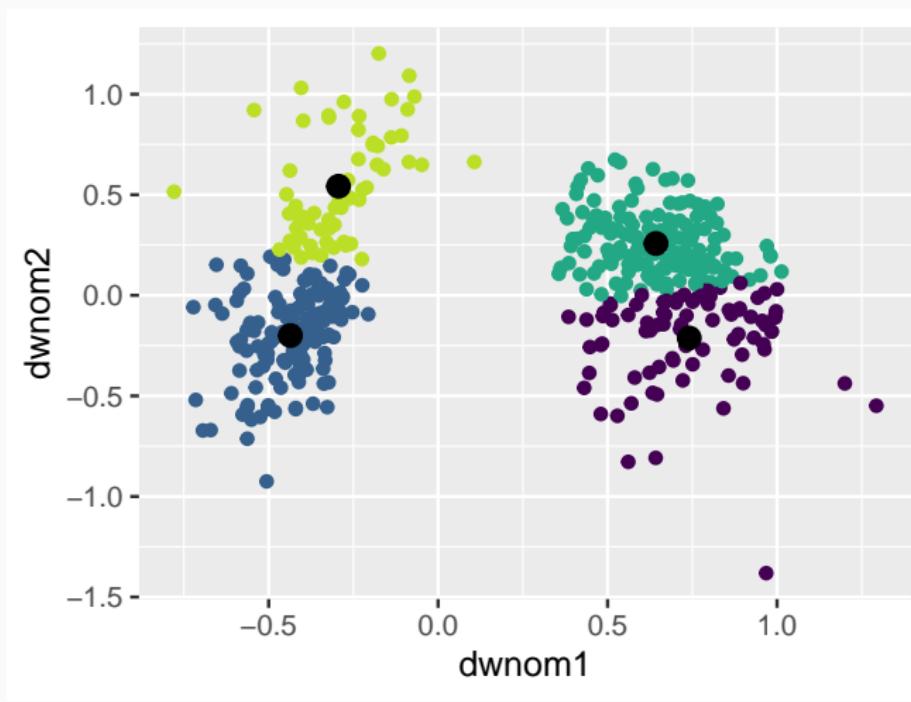
- Elbow method
- Gap statistics
- Silhouette score
- ...

ELBOW METHOD WITH A “SCREE PLOT”

```
1 # Perform k-means clustering for a range of k values
2 k_values <- 1:10
3 distortions <- numeric(length(k_values))
4
5 for (k in k_values) {
6   km <- kmeans(features, centers = k, nstart = 10)
7   distortions[k] <- km$tot.withinss
8 }
```



예시: IDEOLOGICAL CLUSTERS IN U.S. LEGISLATORS



코드: Posit Cloud

IMAGE COMPRESSION

<https://towardsdatascience.com/clear-and-visual-explanation-of-the-k-means-algorithm-applied-to-image-compression-b7fdc547e410>