

INTRODUCTION TO DATA MINING

Week01

What is Data Mining?

- Data mining
 - ▣ Data mining is the process of discovering patterns, relationships, anomalies, or useful insights from large datasets
 - ▣ It involves statistical techniques, database management, and machine learning algorithms to extract meaningful information
- Key objectives of data mining
 - ▣ Identify hidden patterns and trends in data
 - ▣ Support decision-making through predictive and descriptive analytics
 - ▣ Improve business intelligence and automation

What is Data Mining?

Data Mining is
Discovering patterns or
extracting information

from data

to utilize results
for further use

The diagram illustrates the components of Data Mining. It features three horizontal yellow bars. The first bar contains the text 'Discovering patterns or extracting information'. The second bar contains 'from data'. The third bar contains 'to utilize results for further use'. To the right of these bars, three curly braces group the text into categories: a brace for 'Discovering patterns or extracting information' is labeled 'Action'; a brace for 'from data' is labeled 'Resource'; and a brace for 'to utilize results for further use' is labeled 'Purpose'.

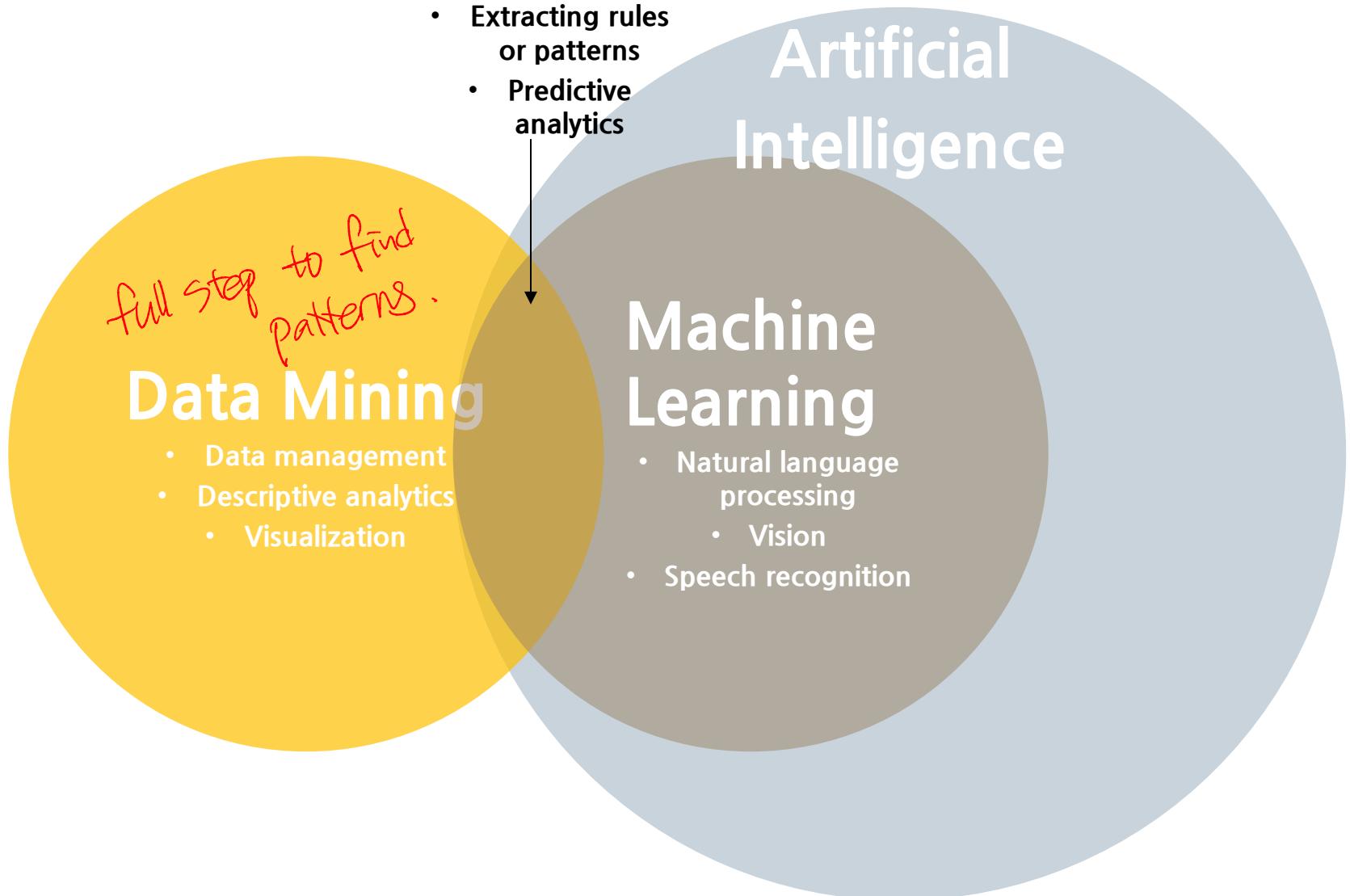
Are Machine Learning and Data Mining Different?

- Machine learning
 - ▣ Machine learning (ML) is a subset of AI that enables computers to learn patterns from data and make predictions without explicit programming.
 - ▣ Unlike traditional rule-based systems, ML algorithms improve performance over time as they process more data.
*find rule automatically
from data.*
- Key objectives of machine learning
 - ▣ Train models to recognize patterns in data
 - ▣ Make accurate predictions for unseen data
 - ▣ Automate decision-making based on learned patterns

What is Artificial Intelligence?

- Artificial intelligence
 - ▣ Artificial intelligence (AI) is a broad field of computer science that aims to create intelligent systems capable of mimicking human cognition, including learning, reasoning, and problem-solving.
 - ▣ Machine learning and data mining are part of AI, but AI encompasses more than just these techniques.
- Key objectives of artificial intelligence
 - ▣ Develop intelligent systems that can make autonomous decisions
 - ▣ Simulate human cognitive abilities such as reasoning, perception, and language processing
 - ▣ Enable automation across various industries

What is Data Mining?



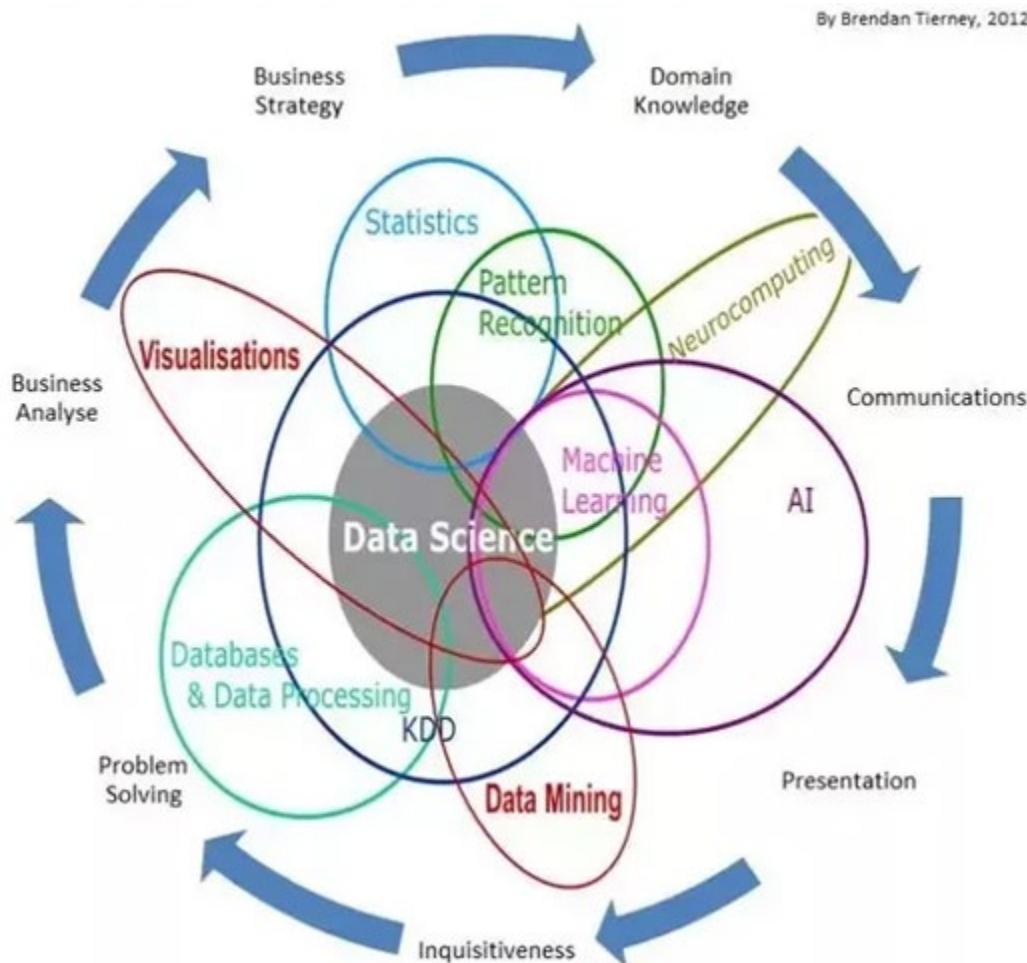
Data Mining vs. Machine Learning vs. Artificial Intelligence

□ Comparison

Feature	Data Mining	Machine Learning	Artificial Intelligence
Definition	Extracting patterns and knowledge from data	Training models to learn from data and make predictions	Creating intelligent systems that mimic human cognition
Main Goal	Discover hidden relationships in data	Enable computers to learn and adapt without explicit programming	Automate reasoning, learning, and problem-solving
Methods Used	Statistical analysis, rule-based systems, machine learning	Supervised, unsupervised, and reinforcement learning	Machine learning, Neural networks (deep learning), expert systems
Relationship	Uses ML techniques to analyze data	A subset of AI focused on learning patterns	The broadest concept, encompassing ML and data mining

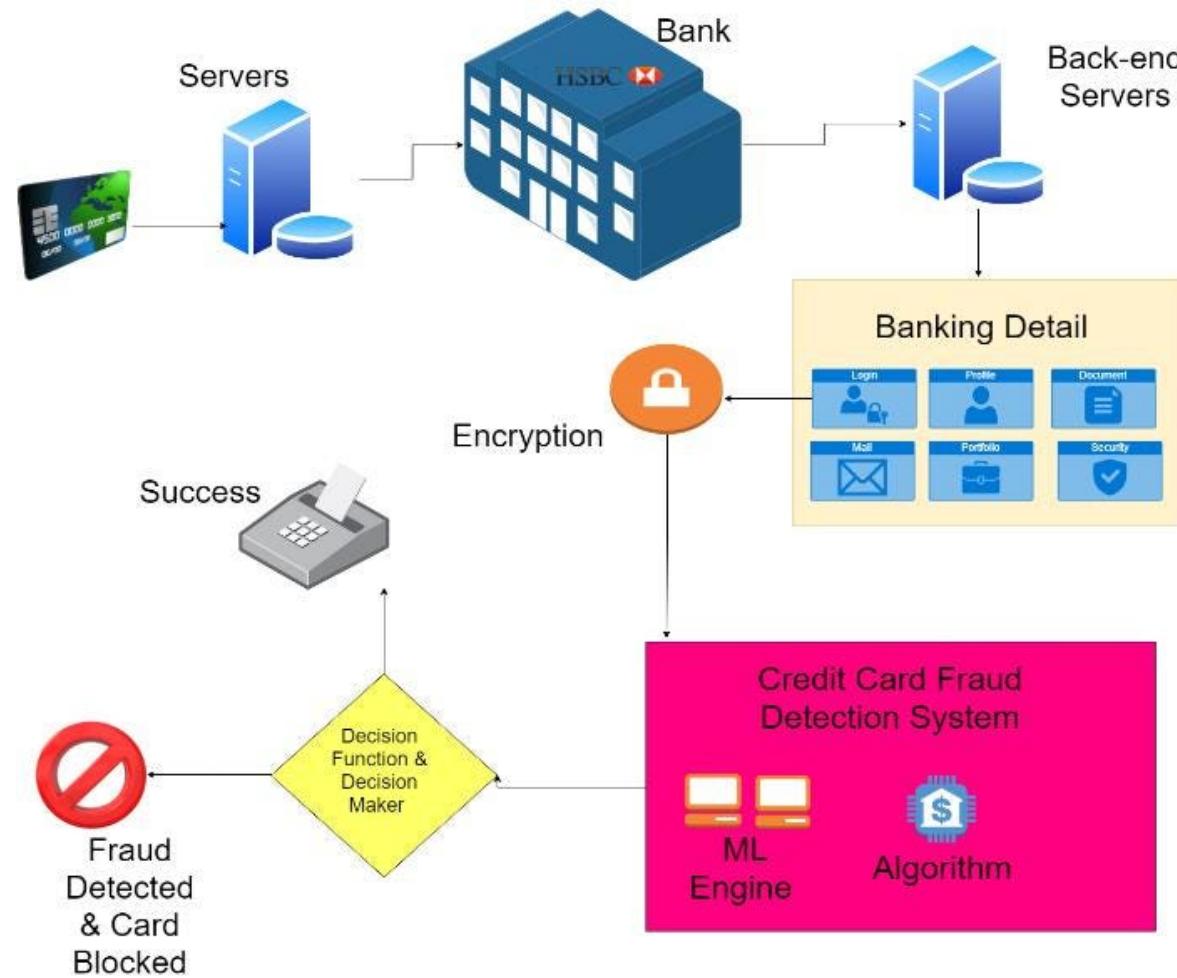
Data Mining, Part of Data Science

- Data Science is Multidisciplinary



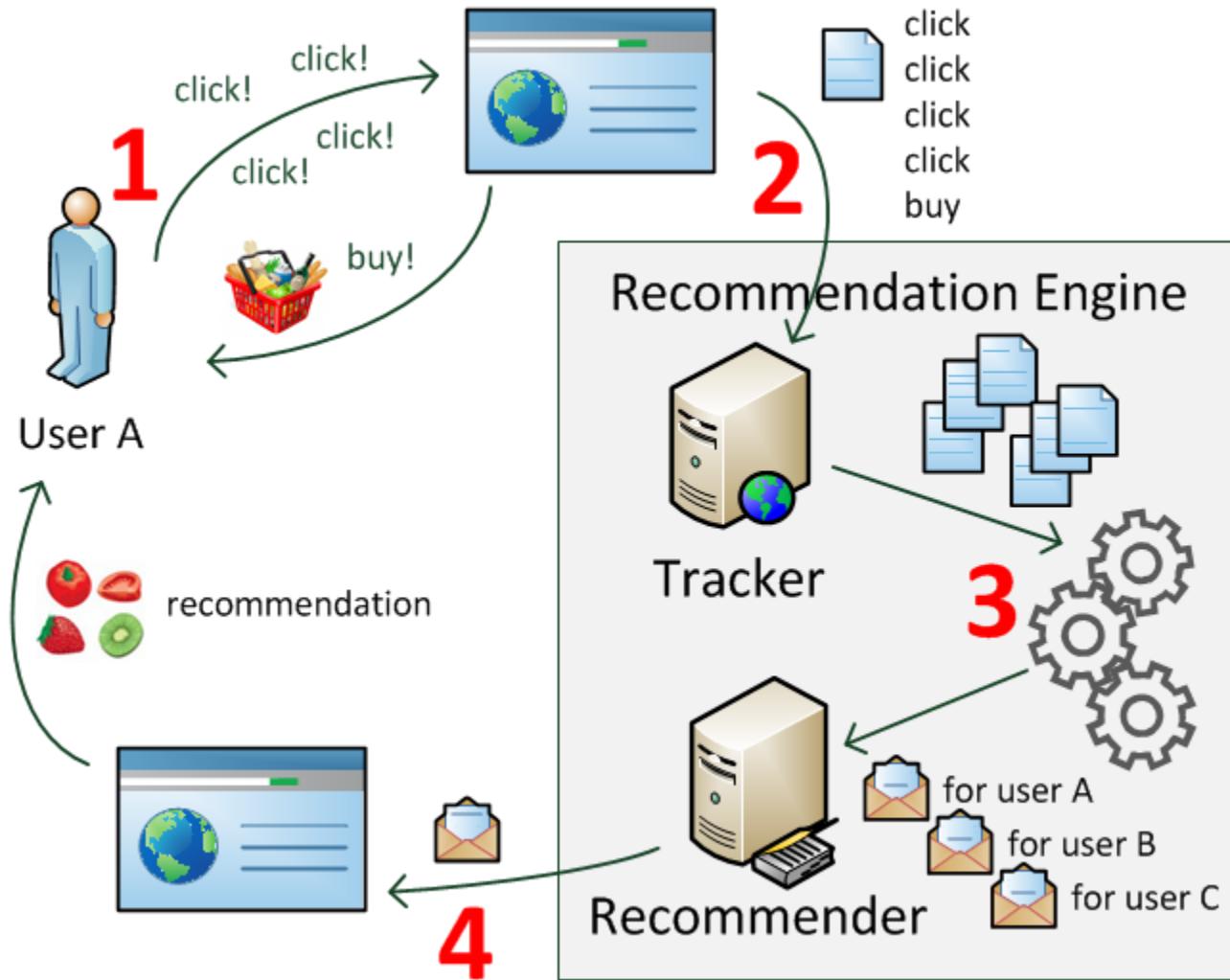
Application Areas of Data Mining

□ Fraud detection systems



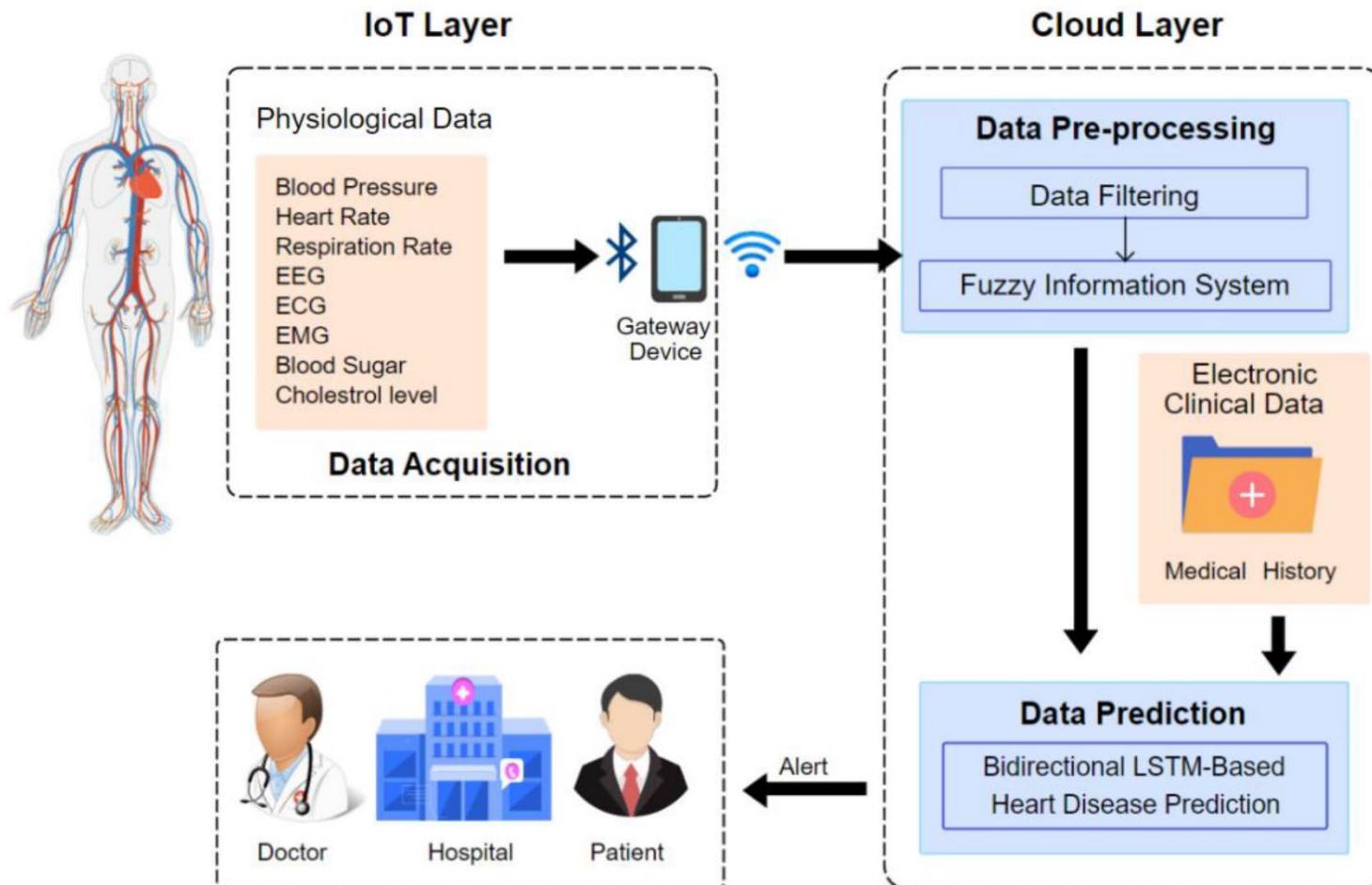
Application Areas of Data Mining

- Recommendation systems



Application Areas of Data Mining

- Healthcare monitoring systems

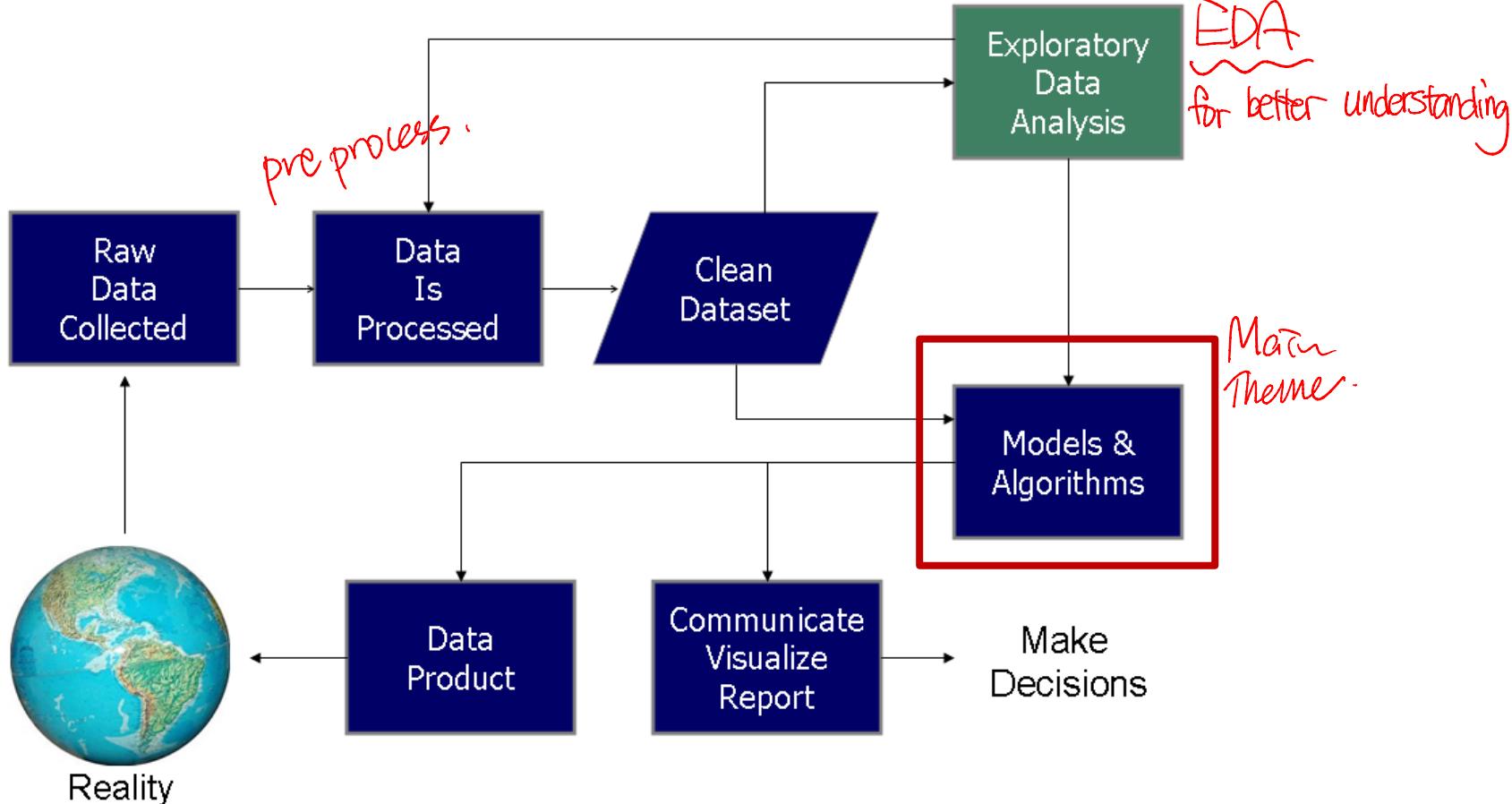


Application Areas of Data Mining

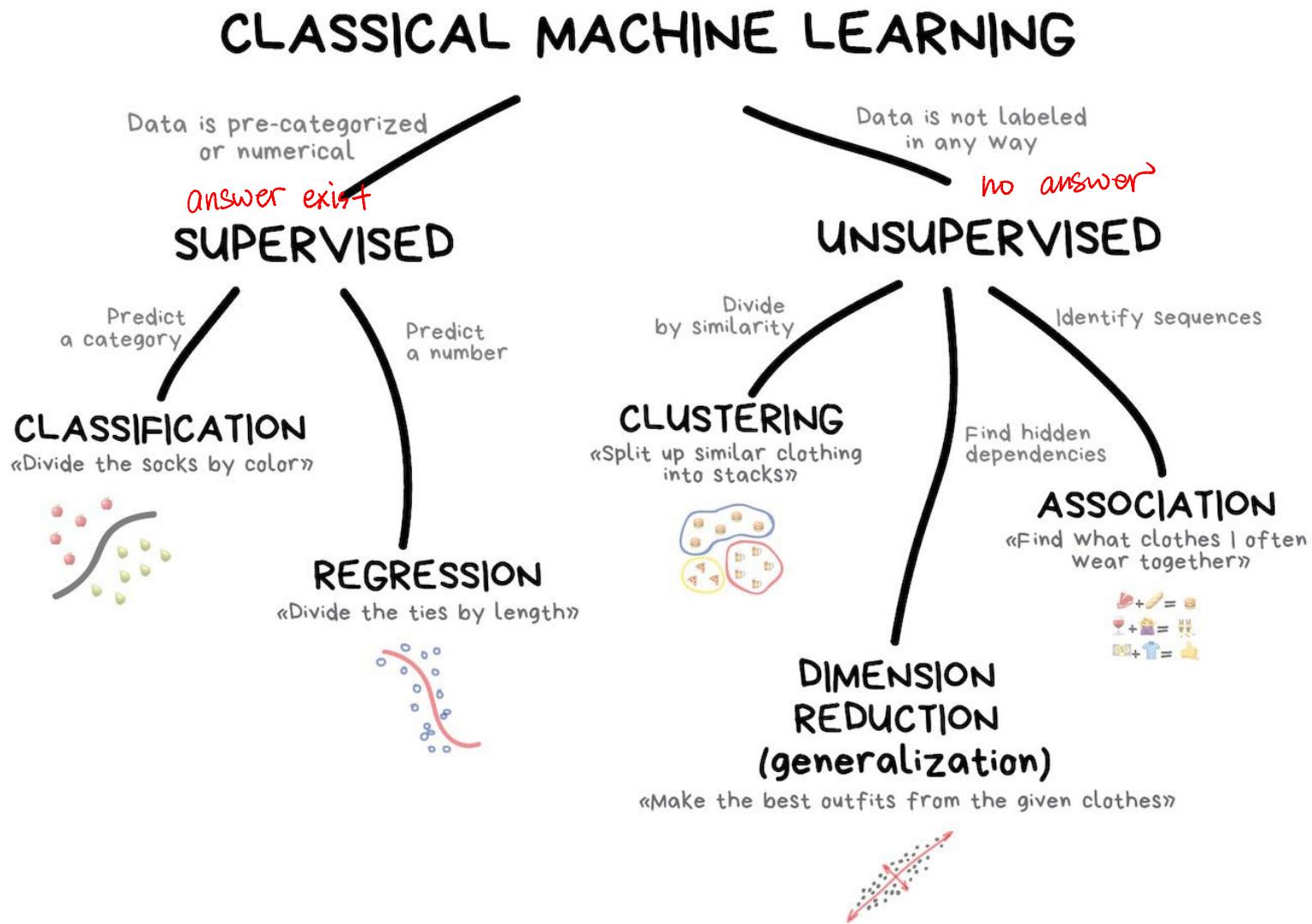
□ Smart factory



What We Will Learn in This Class



Topics Covered in This Class



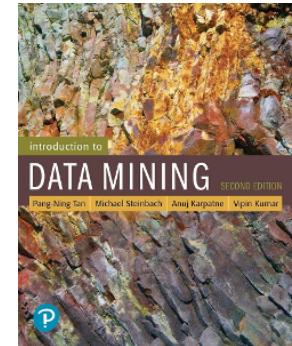
Topics Covered in This Class

- Supervised learning
 - ▣ Regression
 - Linear regression
 - Nearest neighbor methods
 - Decision tree
 - ▣ Classification
 - Logistic regression
 - Naïve Bayes
 - Nearest neighbor methods
 - Decision tree
- Unsupervised learning
 - ▣ Clustering
 - k -means
 - Hierarchical clustering
 - ▣ Dimension reduction
 - Principal component analysis (PCA)
 - ▣ Association rule mining

Recommended Data Mining Textbooks

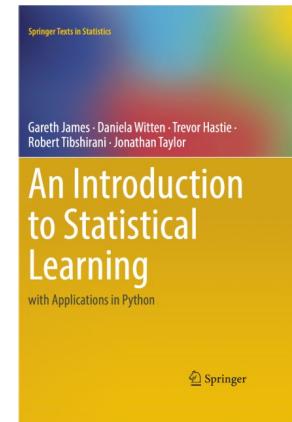
1. Introduction to Data Mining, 2nd edition

- <https://www-users.cse.umn.edu/~kumar001/dmbook/index.php>



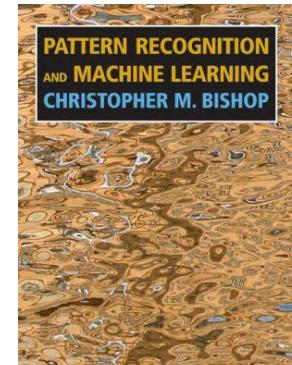
2. An Introduction to Statistical Learning: with Applications in Python

- <https://www.statlearning.com/>



3. Pattern Recognition and Machine Learning

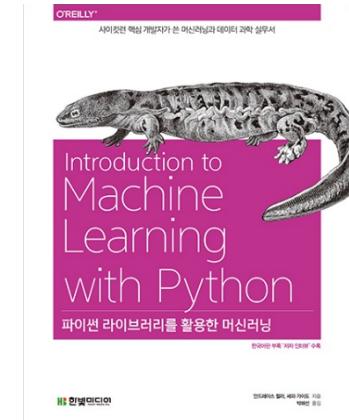
- <https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/>



Recommended Data Mining Textbooks

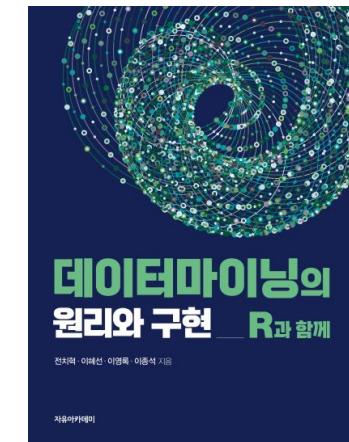
4. 파이썬 라이브러리를 활용한 머신러닝

- https://www.hanbit.co.kr/store/books/look.php?p_code=B6119391002



5. 데이터마이닝의 원리와 구현 : R과 함께

- <http://www.freeaca.com/new/book/MainBookView.aspx?bookid=31311&page=1&ca1=3&ca2=54&sword=&stype=>

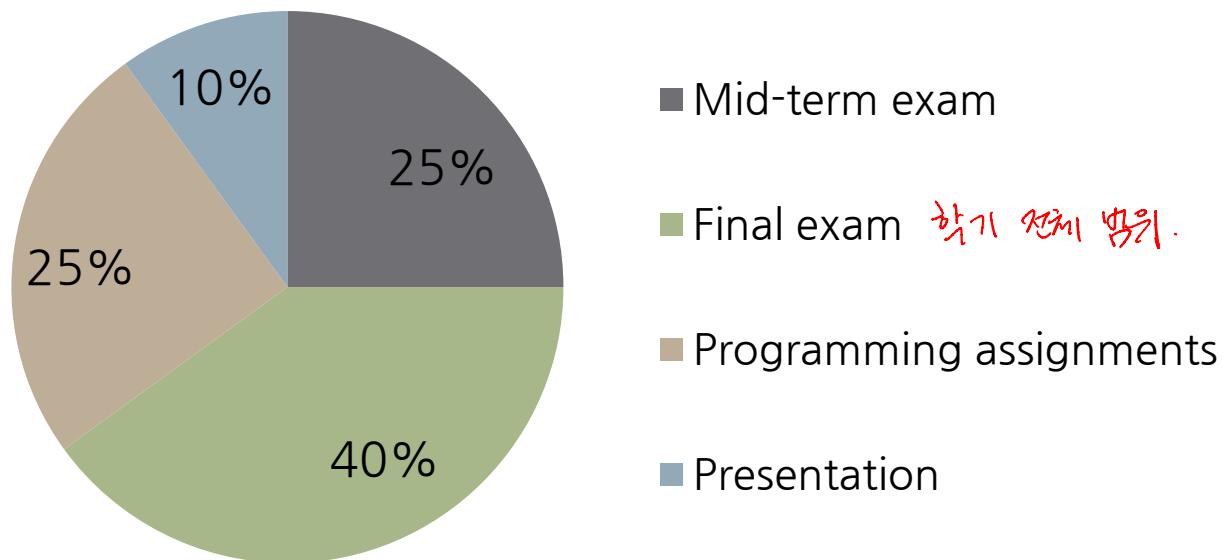


Principals of Lecture

- Understand main goal and basic principles of each data mining techniques
 - ▣ Why is an algorithm proposed?
 - ▣ What is a key point?
- Deliver principles as easy as possible without mathematics
- Understand detailed process of each data mining techniques
 - ▣ How are an algorithm working?
- Explain process step by step
- ※ Some equations will be introduced for explanation
- Exercise what you learned during lectures
 - ▣ Main programming language: Python
 - ▣ Confirm algorithms studied during lectures through programming exercises

Principals of Lecture: Assessment

- Course assessment
 - ▣ Exams will be held two times: mid-term and final exams
 - Final exam will cover the whole lectures
 - ▣ Programming assignments related with lectures
 - ▣ Team presentation: Case study
 - Topic proposal will be presented on the **9th** week
 - The final result will be presented on the **15th** week
 - Each team consist of 2~3 students (random)



Principals of Lecture: Assessment

- Exams
 - ▣ Assess the theoretical knowledge learned in class
 - Must understand principles and process of the data mining algorithms covered in class
 - ▣ No multiple choice questions
 - ▣ Can use a scientific calculator
 - ▣ **Schedule**
 - Mid-term exam: 8th Week, 4/24 (in the evening)
 - Final exam: 14th Week, 6/5 (in the evening)

Principals of Lecture: Assessment

- Team presentation
 - ▣ Case study using data mining
 - The purpose of data analysis
 - What is the problem?
 - Method
 - How did they solve the problem through data mining?
 - Result
 - What kinds of implication could be derived from the results of data analysis?

Schedule

Week	Date	Contents	Remarks
1	3/6	Introduction	
2	3/13	Background of data mining	
3	3/20	Explanatory data analysis (EDA)	
4	3/28	Linear regression: Theory Part 1 & Exercise	
5	4/3	Linear regression: Theory Part 2 & Exercise	
6	4/10	Logistic regression: Theory & Exercise	
7	4/17	Naïve Bayes classifier: Theory & Exercise	
8	4/24	Mid-term exam (in the evening)	
9	5/1	Nearest neighbor algorithm: Theory & Exercise Presentation: Case study topic proposal	<i>supervised</i>
10	5/8	Decision tree: Theory & Exercise	
11	5/15	Clustering: Theory & Exercise	
12	5/22	Dimensionality reduction: Theory & Exercise	
13	5/29	Association rule mining: Theory & Exercise	
14	6/5	Final exam (in the evening)	
15	6/12	Presentation: Case study	

Q & A

- If you want to ask a question related with lectures for data mining algorithms outside of class, please use the Q&A board of the e-class
 - ▣ Your question may be helpful to other students
 - Share your questions with other students
 - ▣ Do not ask individual questions by e-mail

Python: Installation

Installation

□ Python

- ▣ Visit <https://www.python.org/downloads/> and download Python installation file depending on your OS(Windows, Linux/UNIX, Mac OS X) and which version you want to install
 - This slide assumes that OS is Windows
- ▣ There are two stable versions of Python: 3.X, 2.X
 - Two versions are a little bit different, but different features do not matter in this course



The screenshot shows the Python website homepage with a dark blue header. The header includes the Python logo, a "Donate" button, a search bar, a "GO" button, and a "Socialize" button. Below the header is a navigation menu with tabs: About, Downloads, Documentation, Community, Success Stories, News, and Events. The "Events" tab is highlighted with a blue background. The main content area has a dark background. On the left, there is a code editor window displaying Python code to generate a Fibonacci series up to n=1000. On the right, there is a section titled "Functions Defined" with text about defining functions in Python, a "More about defining functions in Python 3" link, and a navigation bar with pages 1 through 5.

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
    print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

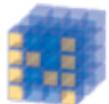
1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [»» Learn More](#)

Installation Useful Packages



- SciPy
 - Python-based ecosystem of open-source software for mathematics, science, and engineering
 - <http://www.scipy.org/>



NumPy
Base N-dimensional array package



SciPy library
Fundamental library for scientific computing



Matplotlib
Comprehensive 2D Plotting



IPython
Enhanced Interactive Console



Sympy
Symbolic mathematics



pandas
Data structures & analysis

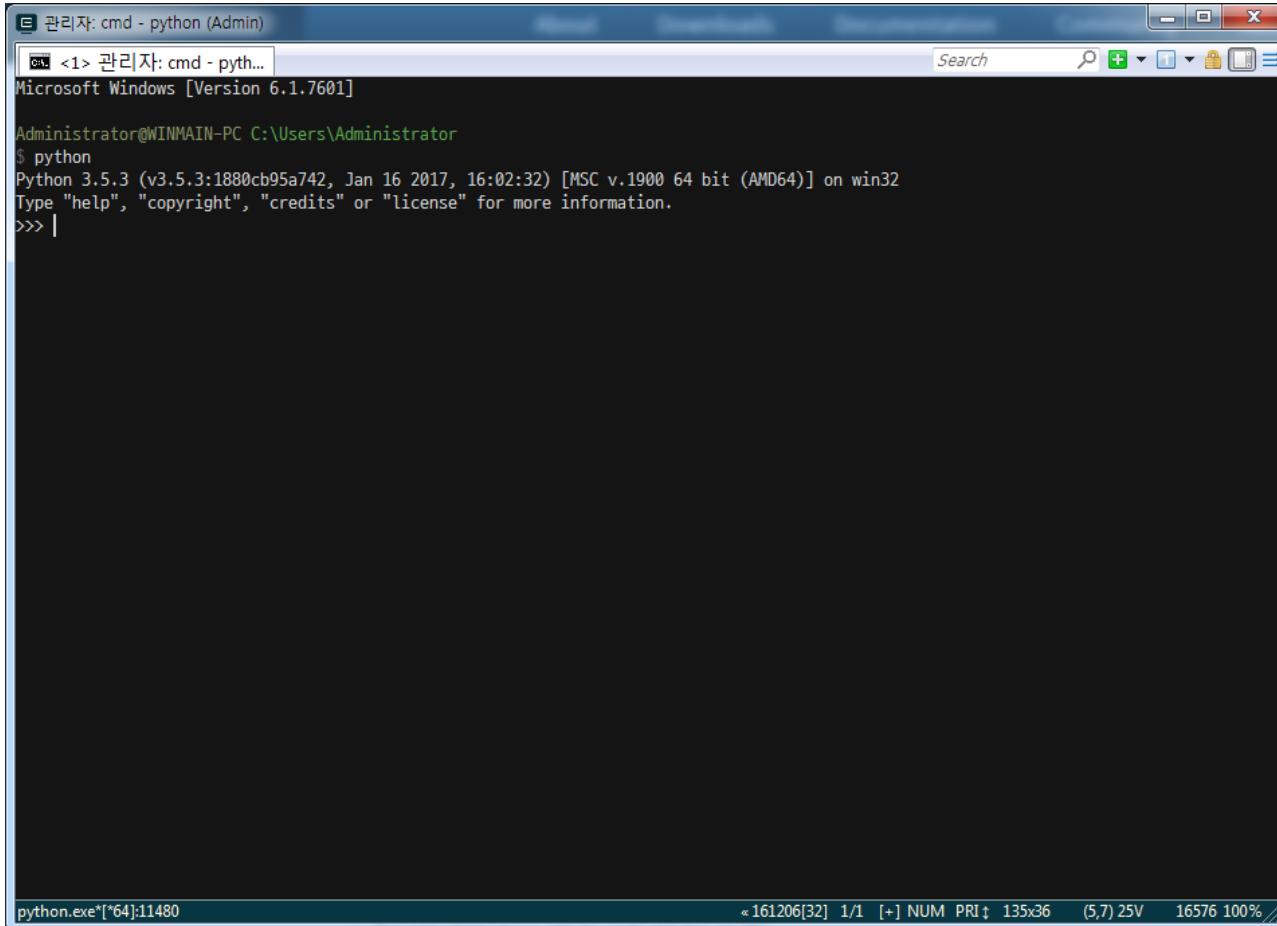
[Core packages]

Installation Useful Packages

- sci-kit learn
 - ▣ Free software machine learning library for the Python programming language
 - Simple and efficient tools for predictive data analysis
 - Built on Numpy, Scipy, and matplotlib
 - ▣ <https://scikit-learn.org/stable/index.html>

Start Python

- To start python, just type python at cmd prompt
 - ▣ Python is script language



The screenshot shows a Windows Command Prompt window titled "관리자: cmd - python (Admin)". The title bar includes the text "관리자: cmd - pyth..." and "Microsoft Windows [Version 6.1.7601]". The window content displays the Python 3.5.3 startup message:

```
Administrator@WINMAIN-PC C:\Users\Administrator
$ python
Python 3.5.3 (v3.5.3:1880cb95a742, Jan 16 2017, 16:02:32) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

The taskbar at the bottom shows the process name "python.exe[*64]:11480".

Start Python

- There are many Python IDEs(Integrated Development Environment)
 - ▣ However, notepad is also used for writing Python scripts
 - ▣ If you want to use better IDE than notepad
 - <http://pedrokroger.net/choosing-best-python-ide/>
 - ▣ There is also default IDE installed with Python

The image shows two windows side-by-side. The left window is titled "Python 3.4.3 Shell" and contains a terminal interface with the following text:

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (In tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

The right window is titled "getAddress.py - C:\Documents and Settings\Admin\My Documents\Personal\getAddress.py" and contains a Python script file with the following code:

```
#!/usr/bin/python27
# -*- coding: utf-8 -*-

import urllib, urllib2
from bs4 import BeautifulSoup
from unidecode import unidecode
import math
import csv
import datetime
import sys
import os.path
import shutil
import json
import xlrd

reload(sys)
sys.setdefaultencoding('utf-8')

def read_url(url):
    try:
        opener = urllib.FancyURLopener({})
        f = opener.open(url)
        document = f.read()
        opener.close()
        urllib.urlcleanup()
    except:
        pass
```

Python: Easy Installation

Scientific Python distributions

- The easiest way to install the packages of the SciPy stack is to download one of these Python distributions, which includes all the key packages
 - Anaconda: A free distribution for the SciPy stack. Supports Linux, Windows and Mac.
 - Enthought Canopy: The free and commercial versions include the core SciPy stack packages. Supports Linux, Windows and Mac.
 - Python(x,y): A free distribution including the SciPy stack, based around the Spyder IDE. Windows only.
 - WinPython: A free distribution including the SciPy stack. Windows only.
 - Pyzo: A free distribution based on Anaconda and the IEP interactive development environment. Supports Linux, Windows and Mac.

Scientific Python distributions

- Anaconda
 - ▣ URL: <https://www.anaconda.com/distribution/>
- WinPython
 - ▣ URL : <http://winpython.github.io/>
 - <https://sourceforge.net/projects/winpython/>

Scientific Python distributions

- Spyder
 - ▣ The Scientific PYthon Development EnviRonment

The screenshot shows the Spyder IDE interface with several windows open:

- Project explorer:** Shows the file structure of the current project, including files like `temp.py`, `interpolation.py`, and `__init__.py`.
- Editor:** Displays the code for `temp.py`. The code generates data for an ascending spiral in 3-space, performs spline calculations, and plots the results.
- Variable explorer:** A table showing variables and their values, such as `array_int8` (int8), `array_uint32` (uint32), `bars` (container.BarContainer), `df` (DataFrame), `filename` (str), `list_test` (list), `nrows` (int), `r` (float64), `radii` (float64), `region` (tuple), `rgb` (float64), `series` (Series), and `test_none` (NoneType).
- IPython console:** Shows the execution history of code snippets, including the generation of a 3D surface plot and a 2D polar plot.
- 3D Surface Plot:** A 3D surface plot generated by the code in `temp.py`, showing a complex, multi-peaked surface.
- 2D Polar Plot:** A 2D polar plot showing data points, likely from the `bars` variable.

Python: Short Tutorial

Variable Types

□ List

- A list contains items separated by commas and enclosed within square brackets ([])

```
list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
```

```
tinylist = [123, 'john']
```

```
print list      # Prints complete list
```

```
print list[0]    # Prints first element of the list
```

```
print list[1:3]  # Prints elements starting from 2nd till 3rd
```

```
print list[2:]   # Prints elements starting from 3rd element
```

```
print tinylist * 2 # Prints list two times
```

```
print list + tinylist # Prints concatenated lists
```

Variable Types

- Tuples
 - ▣ A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses
 - ▣ The main differences between lists and tuples are
 - Lists are enclosed in brackets ([]) and their elements and size can be changed
 - Tuples are enclosed in parentheses (()) and cannot be updated (read-only)

```
tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )
tinytuple = (123, 'john')
```

```
print tuple      # Prints complete list
print tuple[0]    # Prints first element of the list
print tuple[1:3]   # Prints elements starting from 2nd till 3rd
print tuple[2:]    # Prints elements starting from 3rd element
print tinytuple * 2 # Prints list two times
print tuple + tinytuple # Prints concatenated lists
```

Variable Types

- Dictionary
 - ▣ They work like associative arrays or hashes found in Perl and consist of key-value pairs
 - ▣ Dictionaries are enclosed by curly braces ({}) and values can be assigned and accessed using square braces ([])

```
dict = {}  
dict['one'] = "This is one"  
dict[2]    = "This is two"
```

```
tinydict = {'name': 'john','code':6734, 'dept': 'sales'}
```

```
print dict['one']      # Prints value for 'one' key  
print dict[2]          # Prints value for 2 key  
print tinydict         # Prints complete dictionary  
print tinydict.keys()  # Prints all the keys  
print tinydict.values() # Prints all the values
```

Data Conversion

Function	Description
int(x [,base])	Converts x to an integer. base specifies the base if x is a string.
long(x [,base])	Converts x to a long integer. base specifies the base if x is a string
float(x)	Converts x to a floating-point number.
complex(real [,imag])	Creates a complex number.
str(x)	Converts object x to a string representation.
repr(x)	Converts object x to an expression string.
eval(str)	Evaluates a string and returns an object.
tuple(s)	Converts s to a tuple.
list(s)	Converts s to a list.
set(s)	Converts s to a set.
dict(d)	Creates a dictionary. d must be a sequence of (key,value) tuples.
frozenset(s)	Converts s to a frozen set.
chr(x)	Converts an integer to a character.
unichr(x)	Converts an integer to a Unicode character.
ord(x)	Converts a single character to its integer value.
hex(x)	Converts an integer to a hexadecimal string.
oct(x)	Converts an integer to an octal string.

Basic Operation

Operator	Description	Example
+ Addition	Adds values on either side of the operator.	$a + b = 30$
- Subtraction	Subtracts right hand operand from left hand operand.	$a - b = -10$
* Multiplication	Multiplies values on either side of the operator	$a * b = 200$
/ Division	Divides left hand operand by right hand operand	$b / a = 2$
% Modulus	Divides left hand operand by right hand operand and returns remainder	$b \% a = 0$
** Exponent	Performs exponential (power) calculation on operators	$a^{**}b = 10 \text{ to the power } 20$
//	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed.	$9//2 = 4 \text{ and } 9.0//2.0 = 4.0$

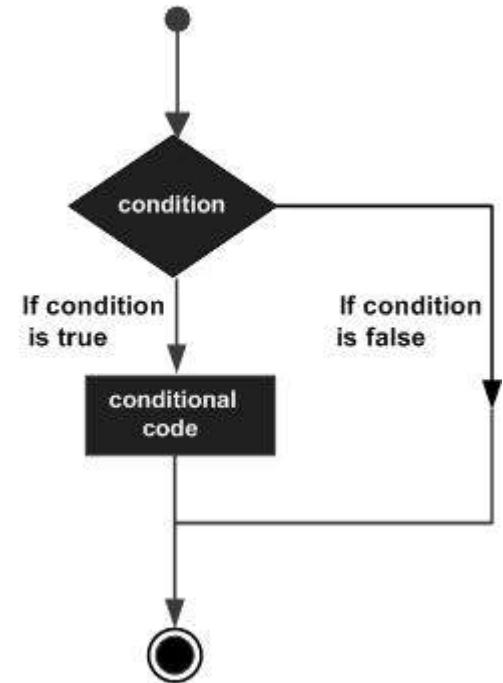
Comparison Operators

Operator	Description	Example
<code>==</code>	If the values of two operands are equal, then the condition becomes true.	$(a == b)$ is not true.
<code>!=</code>	If values of two operands are not equal, then condition becomes true.	
<code>◊</code>	If values of two operands are not equal, then condition becomes true.	$(a ◊ b)$ is true. This is similar to <code>!=</code> operator.
<code>></code>	If the value of left operand is greater than the value of right operand, then condition becomes true.	$(a > b)$ is not true.
<code><</code>	If the value of left operand is less than the value of right operand, then condition becomes true.	$(a < b)$ is true.
<code>>=</code>	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	$(a >= b)$ is not true.
<code><=</code>	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	$(a <= b)$ is true.

Decision Making

- Decision making is anticipation of conditions occurring while execution of the program and specifying actions taken according to the conditions

```
var = 100
if ( var == 100 ):
    print("Value of expression is 100")
else:
    print("Value of expression is not 100")
```



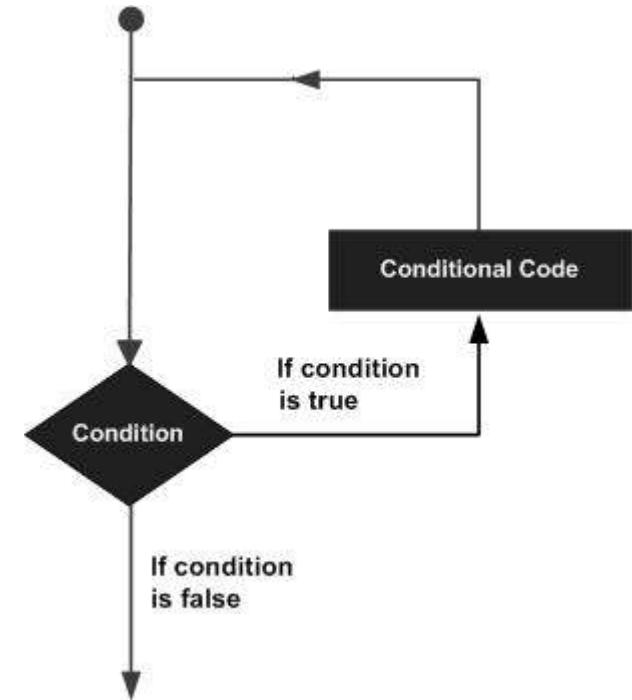
Loop

- A loop statement allows us to execute a statement or group of statements multiple times

```
primes = [2, 3, 5, 7]
for prime in primes:
    print(prime)
```

```
for x in range(5): # or range(5)
    print(x)
```

```
count = 0
while count < 5:
    print(count)
    count += 1 # This is the same as count = count + 1
```



Loop

- Loop control statements change execution from its normal sequence

Control Statement	Description
break statement	Terminates the loop statement and transfers execution to the statement immediately following the loop.
continue statement	Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.
pass statement	The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.

```
count = 0
while True:
    print(count)
    count += 1
    if count >= 5:
        break
```

List comprehensions

- Python supports a concept called “list comprehensions” used to construct lists in a very natural, easy way

```
>>> A=[x**2 for x in range(10)]
>>> print(A)
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
>>> B = [x for x in S if x % 2 == 0]
>>> print(B)
[0, 4, 16, 36, 64]
>>> C = [x+3 for x in A]
>>> print(C)
[3, 4, 7, 12, 19, 28, 39, 52, 67, 84]
>>> D = [x+3 if x%2==0 else x for x in A]
>>> print(D)
[3, 1, 7, 9, 19, 25, 39, 49, 67, 81]
```

List comprehensions

- $A=[x**2 \text{ for } x \text{ in range}(10)]$
 - ▣ $\text{range}(10)$ creates list whose elements are from zero to nine
 $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$
 - ▣ for x in $\text{range}(10)$: loop for elements in $\text{range}(10)$
 - x represents each element in $\text{range}(10)$
 - ▣ $x**2$ for x in $\text{range}(10)$: for every element in $\text{range}(10)$, calculate x^2
 - Results are stored in A as list

Index of Python

□ Python list

index	0	1	2	3	4	5	6
negative index	-7	-6	-5	-4	-3	-2	-1
	8	7	5	13	75	65	11

```
>>>A=[8,7,5,13,75,65,11]
```

```
>>>A[0]
```

```
8
```

```
>>>A[3]
```

```
13
```

```
>>>A[-1]
```

```
11
```

```
>>>A[1:4]
```

```
[7,5,13]
```

```
>>>A[:3]
```

```
[8,7,5]
```

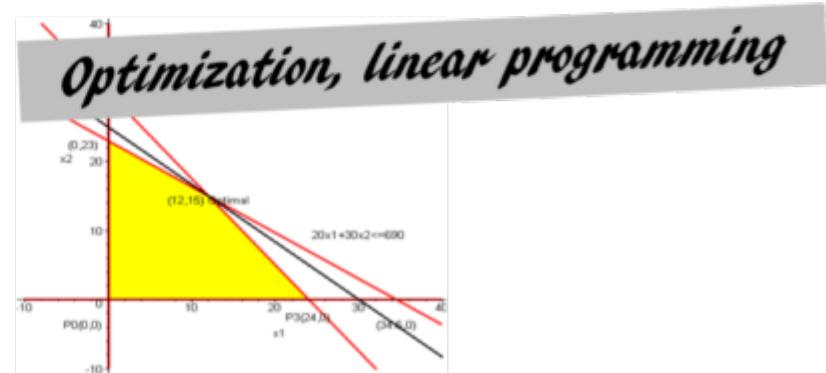
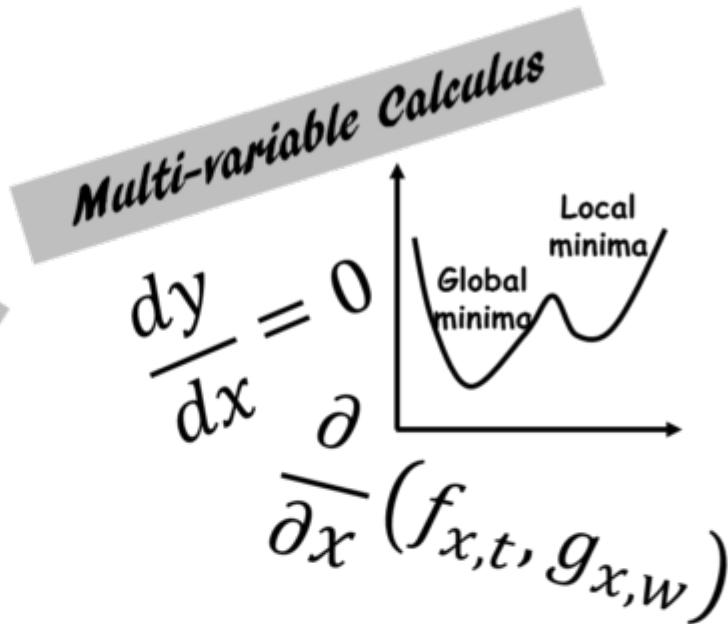
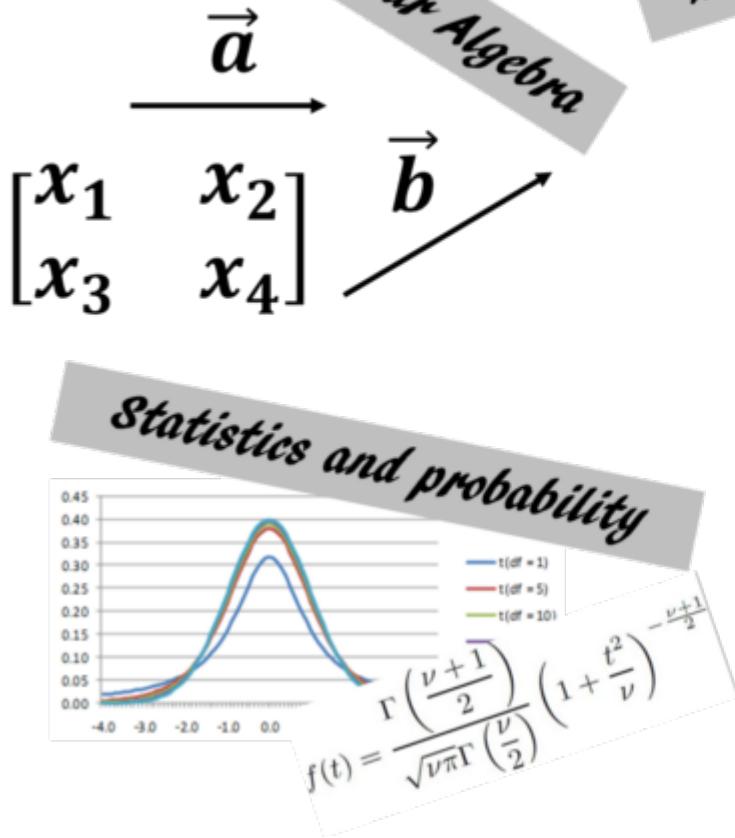
```
>>>A[4:]
```

```
[75,65,11]
```

BACKGROUND OF DATA MINING

Week02

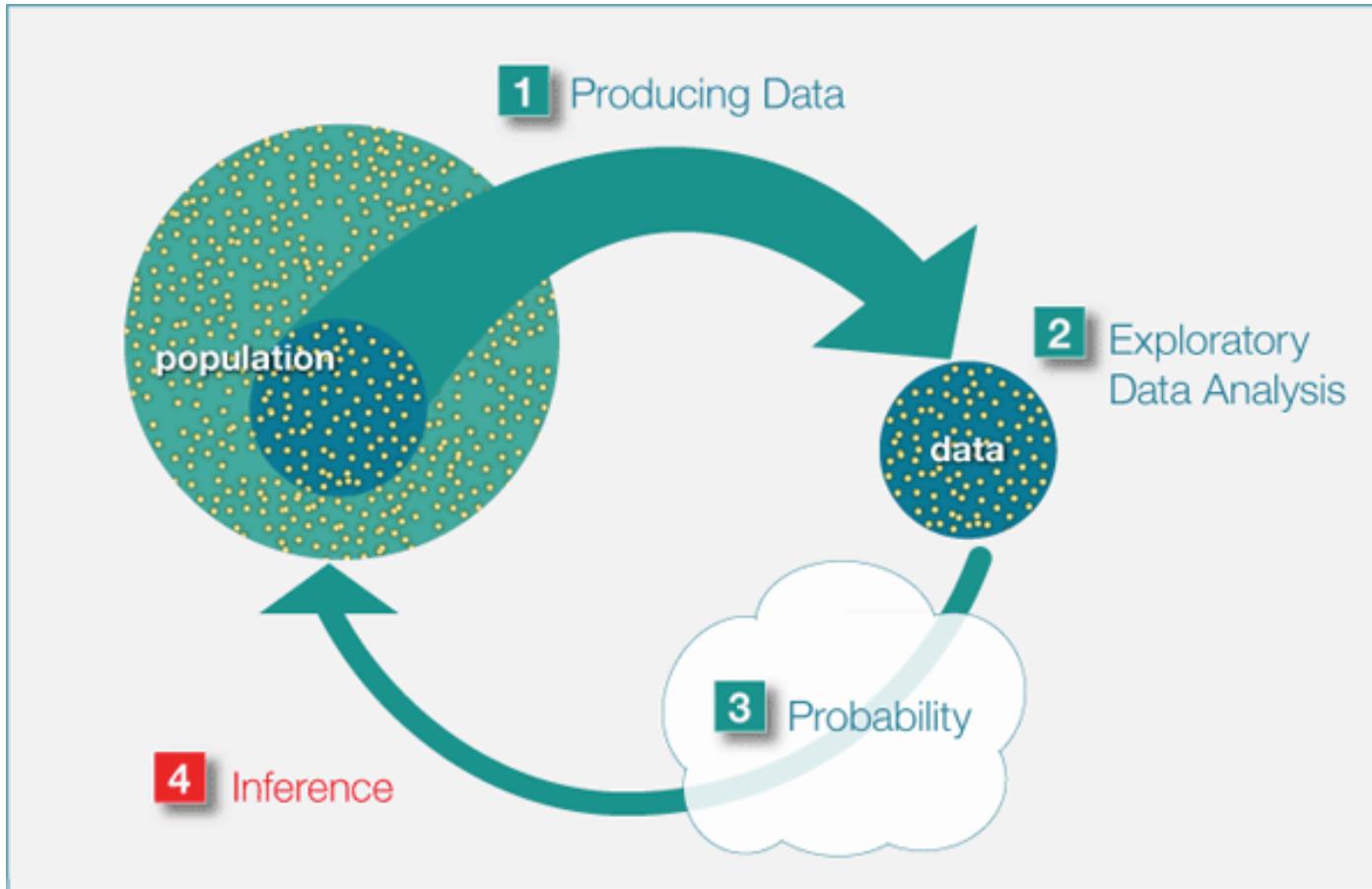
Essential Math for Data Mining



Essential Math for Data Mining: Statistics

- Two main branches of statistics
 - ▣ Descriptive statistics *characteristic, property . explore data*
 - Describe the basic features of data
 - Data summaries and descriptive statistics, central tendency, variance, covariance, correlation
 - ▣ Inferential statistics *estimate population based on sample-*
 - Deduce properties of an underlying distribution of probability
- Probability
 - ▣ Sampling, measurement, error, random number generation
 - ▣ Basic probability: basic idea, expectation, probability calculus, Bayes theorem, conditional probability
 - ▣ Probability distribution functions—uniform, normal, binomial, chi-square, student's t-distribution, Central limit theorem

Essential Math for Data Mining: Statistics



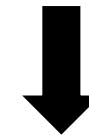
Essential Math for Data Mining: Linear Algebra

- Linear algebra
 - ▣ The study of linear sets of equations and their transformation properties
 - ▣ Concern linear equations, linear functions and their representations in vector spaces and through matrices
 - ▣ Used in most areas of science and engineering, because it allows modeling many natural phenomena, and efficiently computing with such models

$$\begin{aligned}3x + 5y &= 7 \\x - 2y &= 6\end{aligned}$$



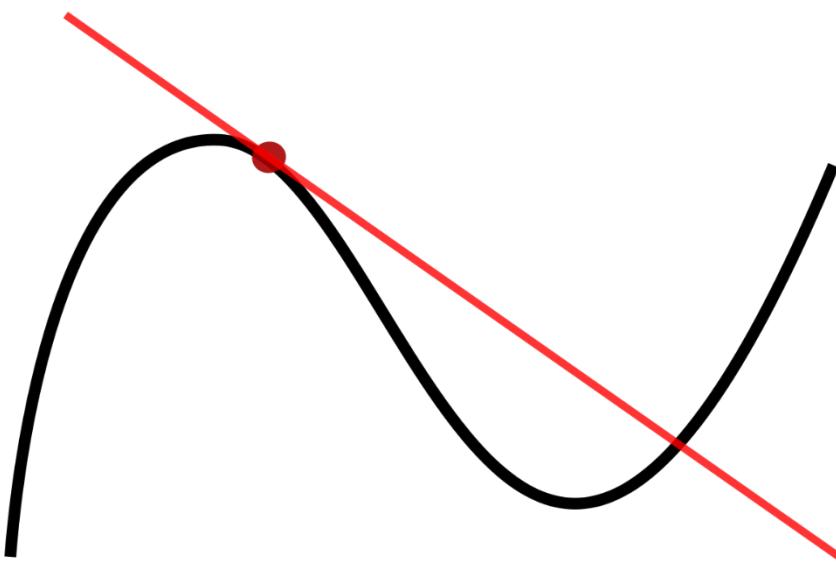
$$\begin{bmatrix} 3 & 5 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7 \\ 6 \end{bmatrix}$$



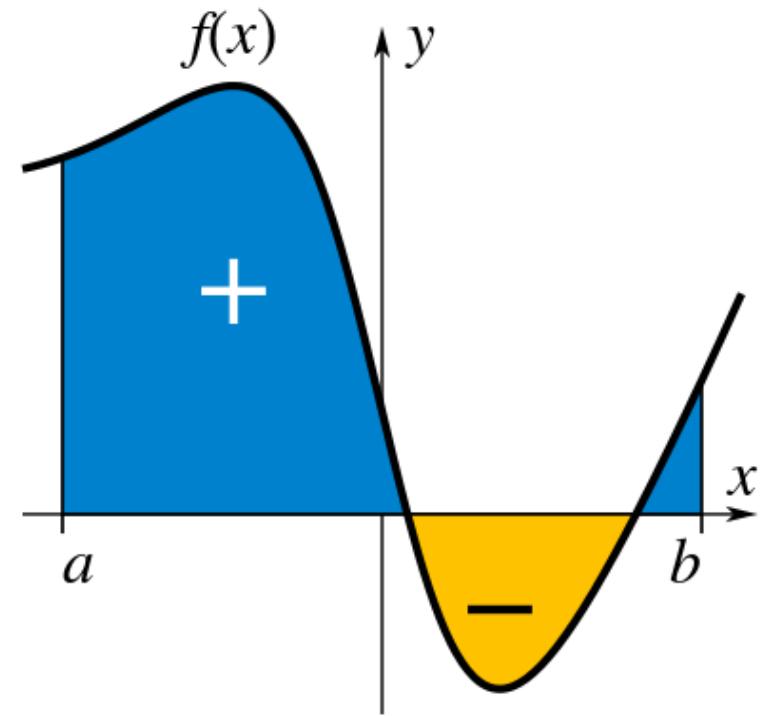
$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 1 & -2 \end{bmatrix}^{-1} \begin{bmatrix} 7 \\ 6 \end{bmatrix}$$

Essential Math for Data Mining: Calculus

- Calculus
 - ▣ Branch of mathematics concerned with the calculation of instantaneous rates of change (differential calculus) and the summation of infinitely many small factors to determine some whole (integral calculus)



❖ differential calculus



integral calculus

Essential Math for Data Mining: Optimization

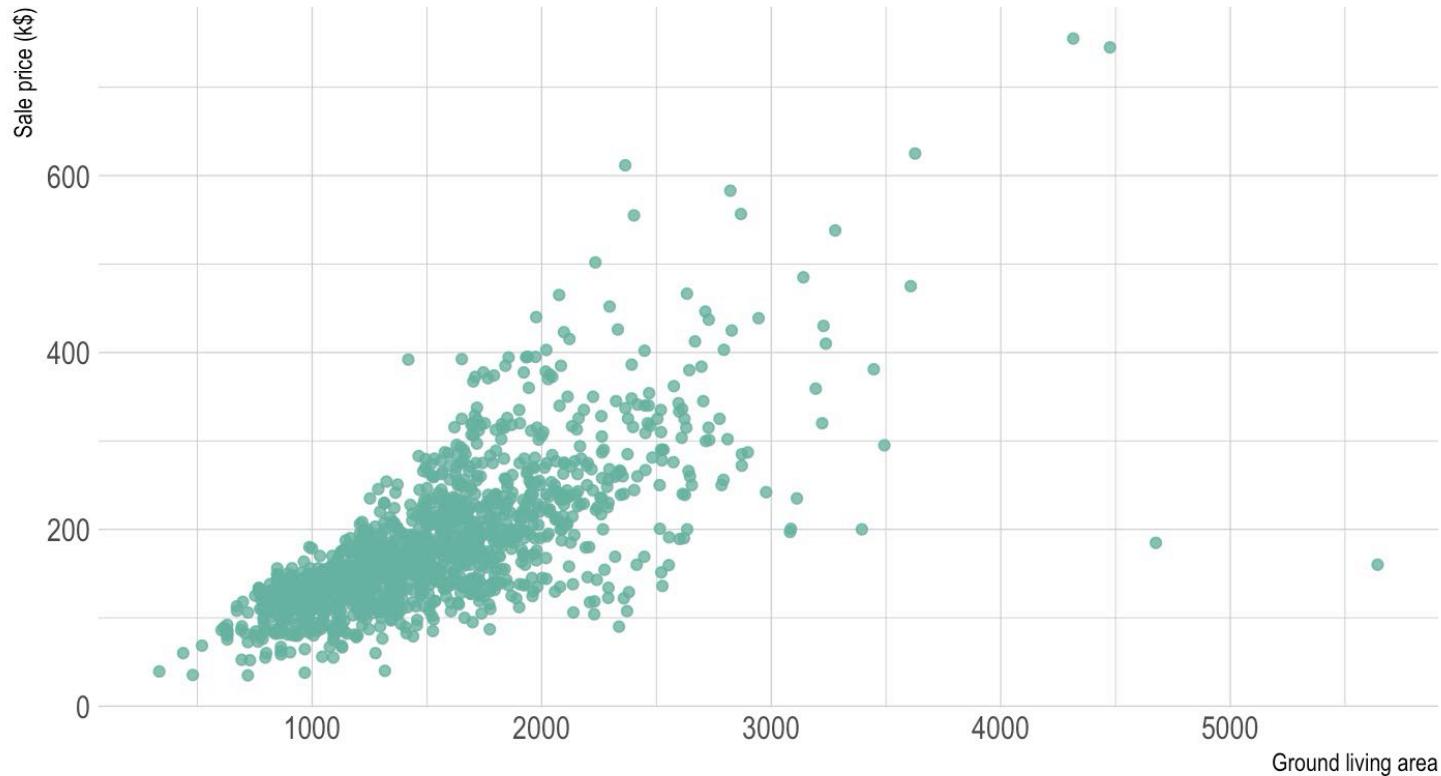
- Optimization
 - ▣ Collection of mathematical principles and methods used for solving optimization problems
 - ▣ Optimization problem is the problem of finding the best solution from all feasible solutions
 - In the simplest case, an optimization problem consists of maximizing or minimizing a real function

Statistics

- A vast set of tools for understanding data

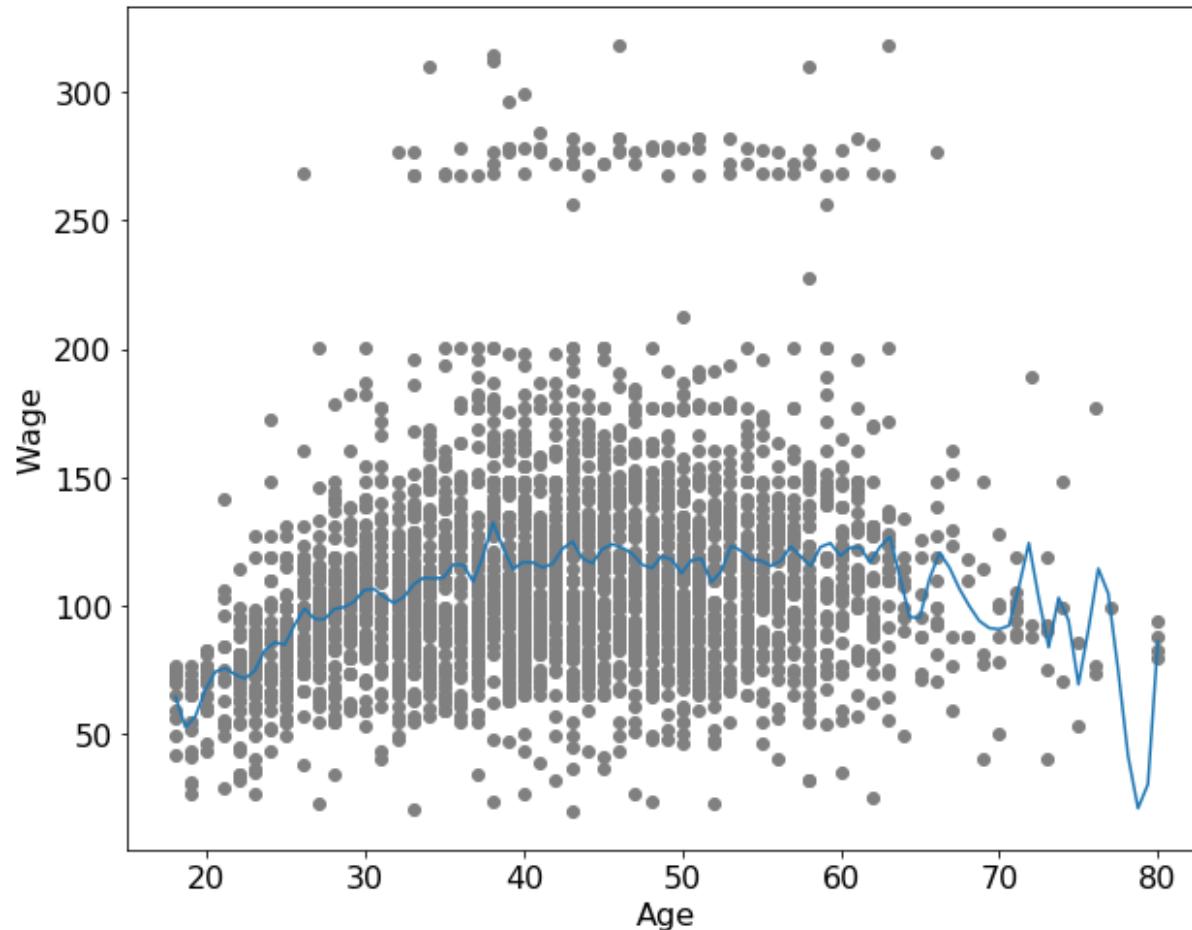
visualize
→ qualitative.

Ground living area partially explains sale price of apartments



Statistics

- A vast set of tools for understanding data

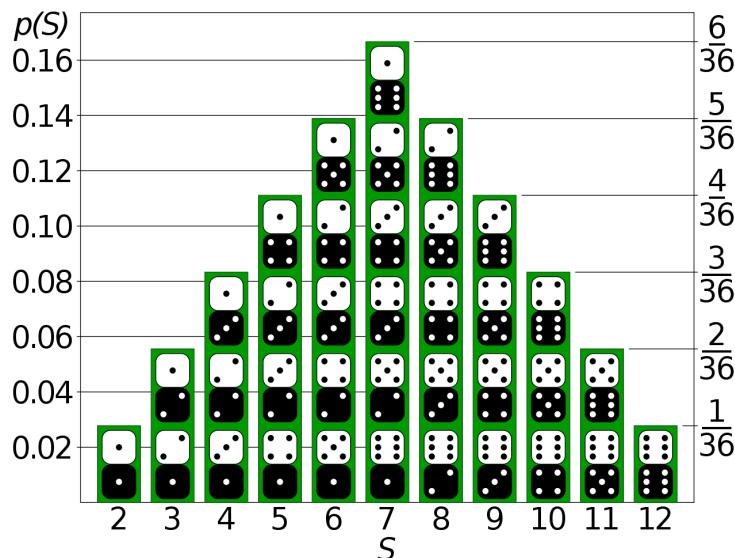


Statistics

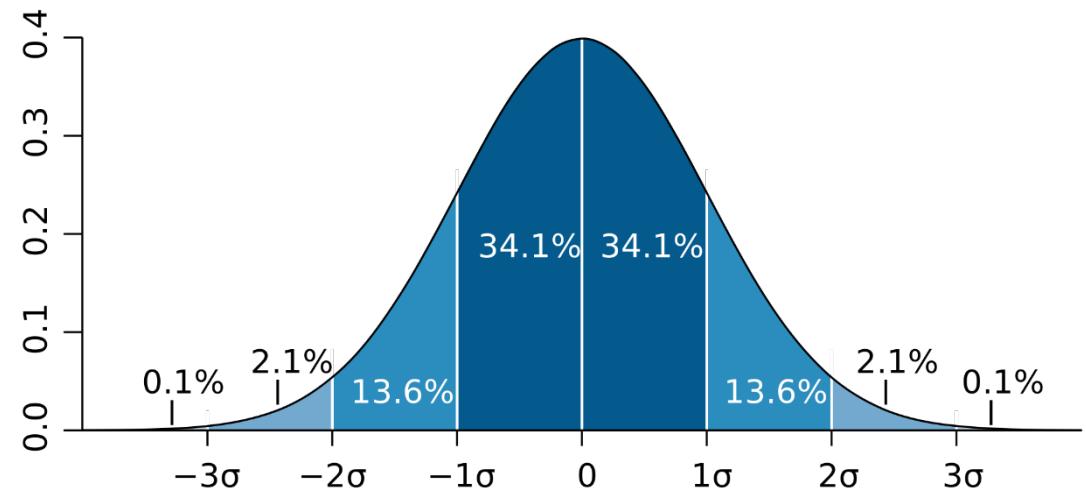
- Descriptive statistics
 - ▣ A **summary statistic** that quantitatively describes or summarizes features of a collection of information
 - ▣ Univariate
 - Mean, Median, Mode *平均数*
 - Variance, standard deviation, Percentile
 - Skewness, kurtosis *偏度* *峰度*
 - ▣ Bivariate or multivariate
 - Cross-tabulations and contingency tables
 - Graphical representation via scatterplots
 - Quantitative measures of dependence (covariance, correlation)

Statistics

- Probability distribution
 - ▣ A mathematical function that provides the probabilities of occurrence of different possible outcomes
 - The probabilities of occurrence of the specific observations



Discrete random variable →
probability mass function



Continuous random variable →
probability density function

Statistics: Discrete Probability Distributions

- Bernoulli distribution *... Yes/No, tossing coin*
 - ▣ The discrete probability distribution of a random variable which takes the value 1 with probability p and the value 0 with probability $q = 1 - p$
$$\Pr(X = 1) = p = 1 - \Pr(X = 0) = 1 - q$$
 - ▣ A special case of the binomial distribution where a single trial is conducted (so n would be 1 for such a binomial distribution)
 - ▣ Probability mass function
$$f(X = k; p) = p^k (1 - p)^{1-k}$$
- Binomial distribution *이항 분포*
 - ▣ The discrete probability distribution of the number of successes in a sequence of n independent experiments, each asking a yes-no question, and each with its own Boolean-valued outcome: success (with probability p) or failure (with probability $q = 1 - p$).
 - ▣ Probability mass function
$$f(X = k; n, p) = \Pr(X = k) = \frac{n!}{k!(n - k)!} p^k (1 - p)^{n-k}$$
$$n C_k p^k (1-p)^{n-k}$$

$$f(X = k; n, p) = \Pr(X = k) = \frac{n!}{k!(n - k)!} p^k (1 - p)^{n-k}$$
$$n C_k p^k (1-p)^{n-k}$$

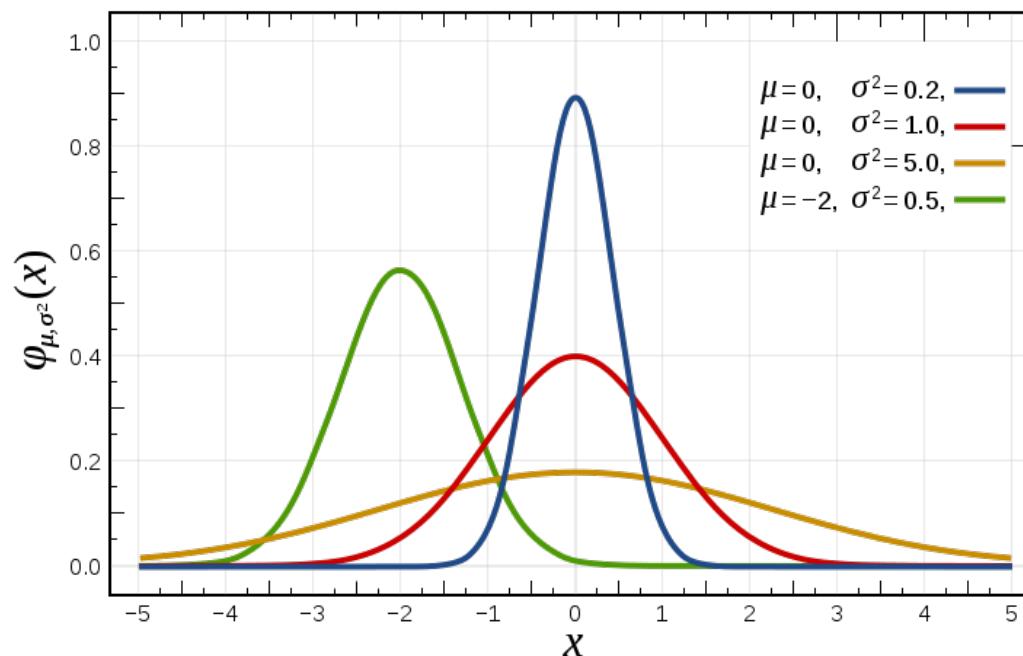
Statistics: Continuous Distributions

- Normal (Gaussian) distribution
 - Very common continuous probability distribution
 - Bell-shaped

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

■ μ : mean

■ σ : standard deviation



Statistics: Continuous Distributions

- Student's t -distribution (t -distribution)
 - ▣ Continuous probability distributions that arises when estimating the mean of a normally distributed population in situations where the **sample size is small** and **population standard deviation is unknown**
- Let X_1, \dots, X_n be independent and identically distributed (iid) as $N(\mu, \sigma^2)$
 - ▣ Sample mean

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

- ▣ Sample variance

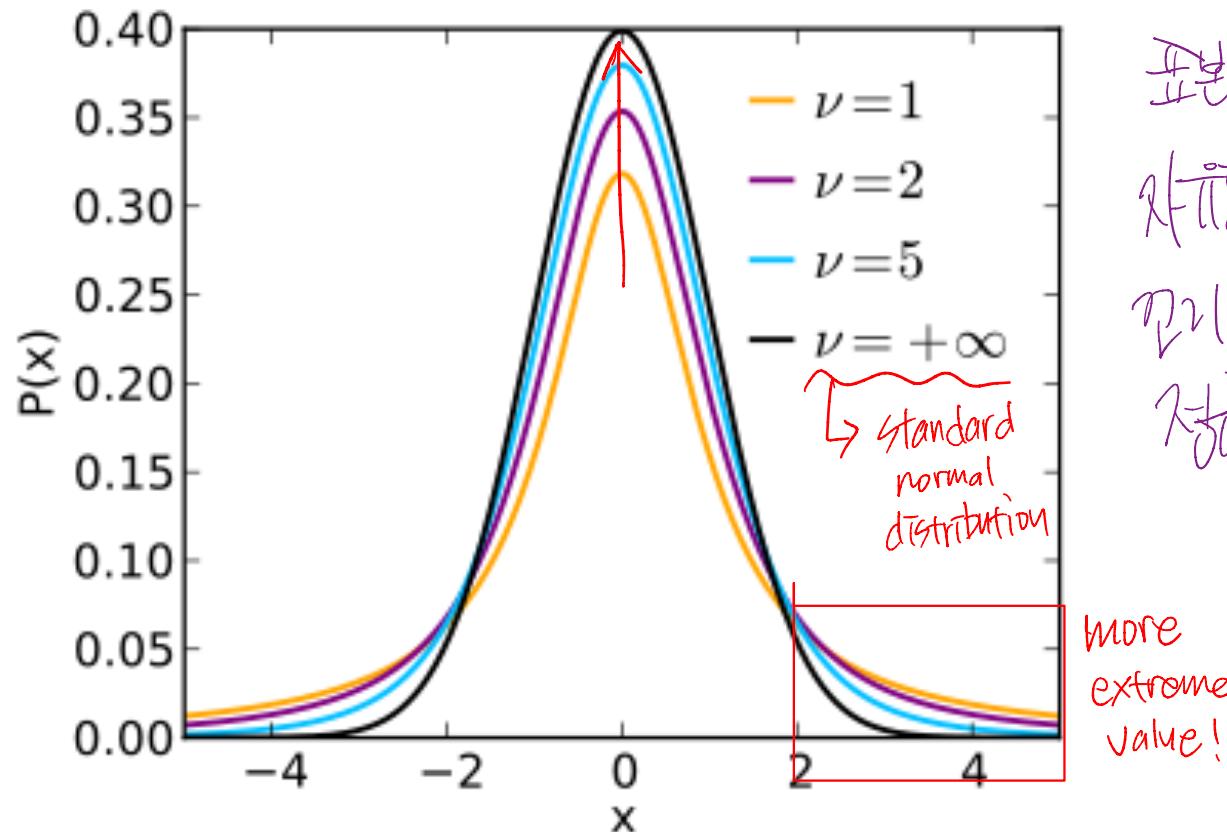
$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

- ▣ The random variable $\frac{\bar{X}-\mu}{\sigma/\sqrt{n}}$ has a standard normal distribution
- ▣ The random variable $\frac{\bar{X}-\mu}{S/\sqrt{n}}$ has a Student's t -distribution with **$n-1$ degrees of freedom**

모집단의 분산 모를 때!

Statistics: Continuous Distributions

- The probability density function of t -distribution with varying degree of freedom



표본 수 증가
자연 분포
균일 분포
정규분포에 가까워지기

Statistics: Student's t -distribution

- Probability density function

$$f(x; \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

- ν : degree of freedom
- Γ : gamma function

$$\Gamma(n) = (n-1)! \quad \text{if } n \text{ is positive integer}$$

$$\Gamma(z) = \int_0^{-\infty} x^{z-1} e^{-x} dx$$

Statistics: Continuous Distributions

□ Chi-squared distribution (χ^2)

자수분포
제곱분산

- The distribution of a sum of the squares of k independent standard normal random variables

□ Let X_1, \dots, X_k be independent, standard normal random variables

$$Y = \sum_{i=1}^k X_i^2$$

is distributed according to the chi-squared distribution with k degrees of freedom

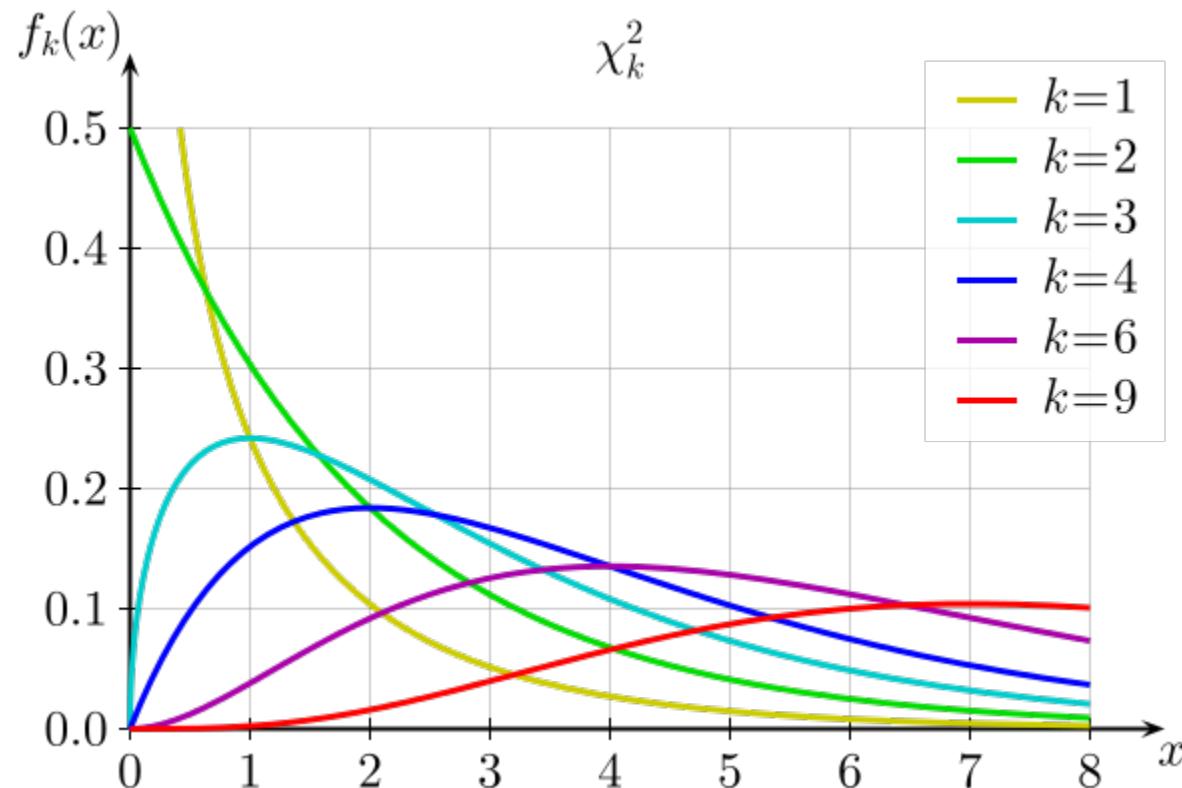
$$Y \sim \chi^2(k) \text{ or } Y \sim \chi_k^2$$

□ Probability density function

$$f(x; k) = \frac{1}{2^{k/2}\Gamma(k/2)} x^{\frac{k}{2}-1} e^{-\frac{x}{2}}$$

Statistics: Continuous Distributions

- The probability density function of chi-squared distribution with varying degree of freedom



Statistics: Continuous Distributions

□ **F-distribution**

- A random variate of the F -distribution with parameters d_1 and d_2 arises as the ratio of two appropriately scaled chi-squared variates
- Let X_1 and X_2 be two independent random variables and $X_1 \sim \chi^2(d_1)$ and $X_2 \sim \chi^2(d_2)$

$$Y = \frac{X_1/d_1}{X_2/d_2}$$

is distributed according to the F -distribution with d_1 and d_2 degrees of freedom

$$Y \sim F(d_1, d_2)$$

□ Probability density function

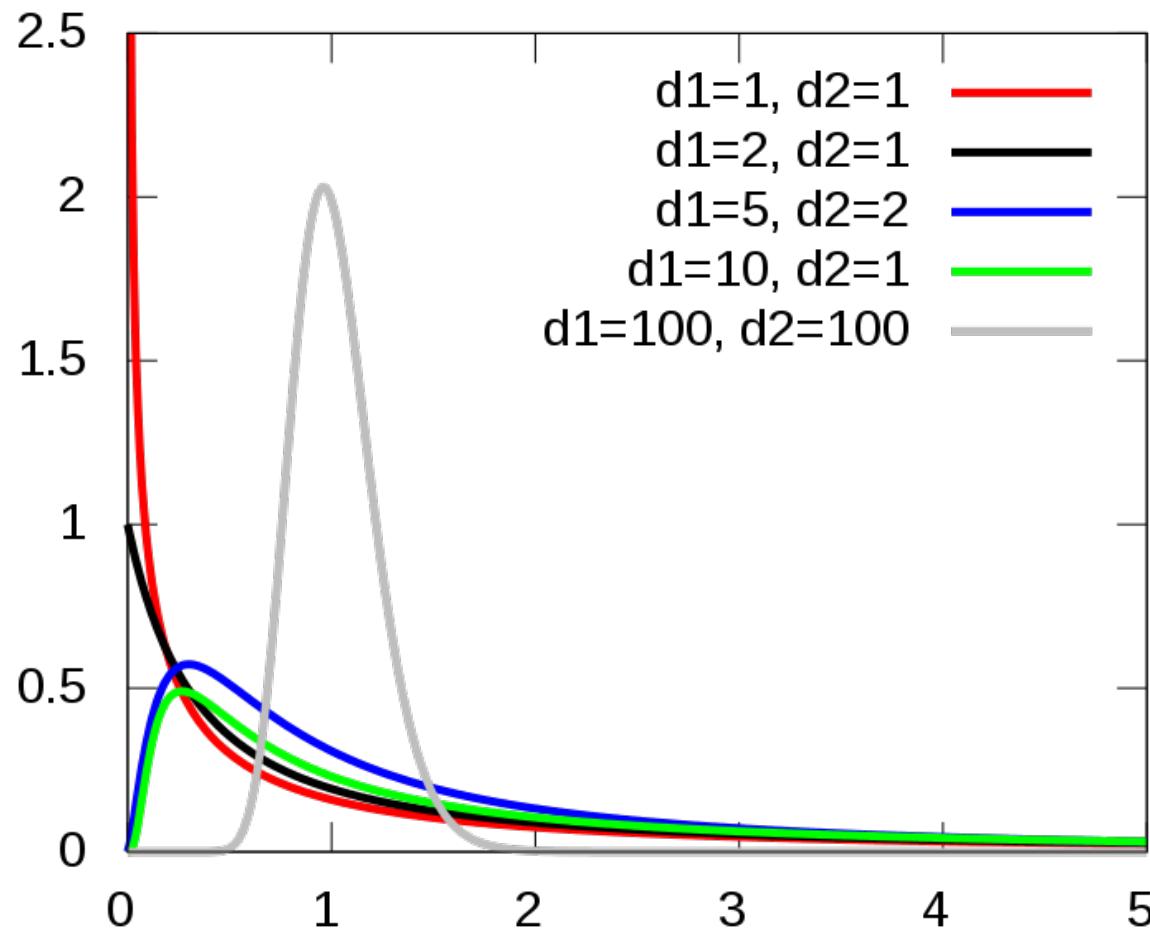
$$f(x; d_1, d_2) = \frac{\sqrt{\frac{(d_1 x)^{d_1} d_2^{d_2}}{(d_1 x + d_2)^{d_1 + d_2}}}}{x B\left(\frac{d_1}{2}, \frac{d_2}{2}\right)}$$

- B: beta function

$$B(x, y) = \int_0^1 t^{x-1} (1-t)^{y-1} dt$$

Statistics: Continuous Distributions

- The probability density function of F -distribution with varying degree of freedom



Linear Algebra

- Linear Algebra
 - ▣ Basic properties of matrix and vectors—scalar multiplication, linear transformation, transpose, conjugate, rank, determinant
 - ▣ Inner and outer products, matrix multiplication rule and various algorithms, matrix inverse
 - ▣ Special matrices—square matrix, identity matrix, triangular matrix, idea about sparse and dense matrix, unit vectors, symmetric matrix, Hermitian, skew-Hermitian and unitary matrices
 - ▣ Gaussian/Gauss-Jordan elimination, solving $Ax=b$ linear system of equation
 - ▣ Matrix factorization and decomposition
 - ▣ Vector space, basis, span, orthogonality, orthonormality, linear least square
 - ▣ Eigenvalues, eigenvectors, and diagonalization, singular value decomposition (SVD)

Linear Algebra

- Linear algebra is the study of vectors and linear functions
 - ▣ Scalar
 - A scalar is a number
 - ▣ Vector
 - A vector is a list of numbers
 - ▣ Matrix
 - A matrix is also a collection of numbers
 - The difference is that a matrix is a table of numbers rather than a list
- ▣ $\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$
- ▣ $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
- ▣ Linear equation
 - $$a_1x_1 + \cdots + a_nx_n = b$$
- ▣ Linear function
 - $$(x_1, \dots, x_n) \mapsto a_1x_1 + \cdots + a_nx_n$$

Linear Algebra

- Vectors

- ▣ Addition

$$\mathbf{v} + \mathbf{w}$$

- Example

$$\mathbf{v} + \mathbf{w} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

- ▣ Linear combination

$$a\mathbf{v} + b\mathbf{w}$$

- Example

$$3\mathbf{v} + 4\mathbf{w} = 3 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + 4 \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 15 \\ 22 \end{bmatrix}$$

Linear Algebra

□ Vectors

▣ Transpose

- column vector \leftrightarrow row vector
- Example

$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \rightarrow \mathbf{v}^T = [1 \quad 2]$$

▣ Dot product, inner product

$$[a_1, a_2, \dots, a_n] \cdot [b_1, b_2, \dots, b_n] \\ = a_1b_1 + a_2b_2 + \dots + a_nb_n$$

- Example

$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \\ \mathbf{v} \cdot \mathbf{w} = (1)(3) + (2)(4) = 9$$

▣ Length

$$\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$$

- Example

$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \|\mathbf{v}\| = \sqrt{(1)(1) + (2)(2)} = \sqrt{5}$$

Linear Algebra

- Matrix

- ▣ Addition

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{bmatrix}$$

- ▣ Multiplication

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

- ▣ Linear equation

$$A\mathbf{x} = \mathbf{b}$$

- Example

$$\begin{array}{rcl} x_1 & = & b_1 \\ -x_1 + x_2 & = & b_2 \\ -x_2 + x_3 & = & b_3 \end{array}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Linear Algebra

□ Matrix

□ Inverse matrix

- An n -by- n square matrix, A is called invertible (or nonsingular) if there exists an n -by- n square matrix, B such that

$$AB = BA = I_n$$

where I_n denotes the n -by- n identity matrix which is a square matrix with ones on the main diagonal and zeros elsewhere

- B is the inverse of A (A^{-1})
- If A has no inverse, A is singular or non-invertible
- Example

$$A = \begin{bmatrix} -1 & \frac{3}{2} \\ 1 & -1 \end{bmatrix}, A^{-1} = \begin{bmatrix} 2 & 3 \\ 2 & 2 \end{bmatrix}$$

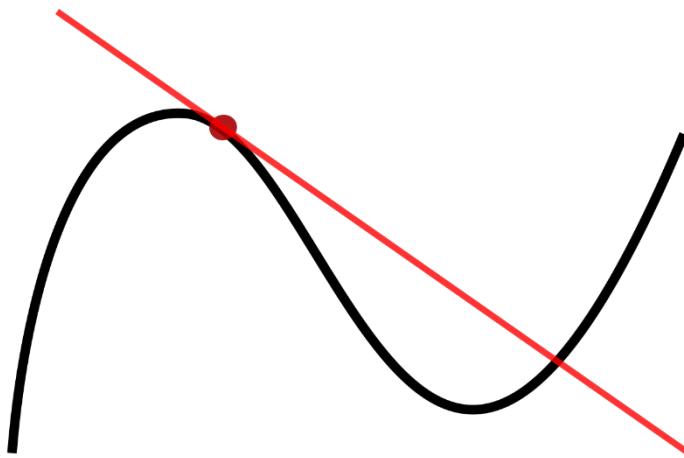
□ Solution of a linear equation

$$\mathbf{x} = A^{-1}\mathbf{b}$$

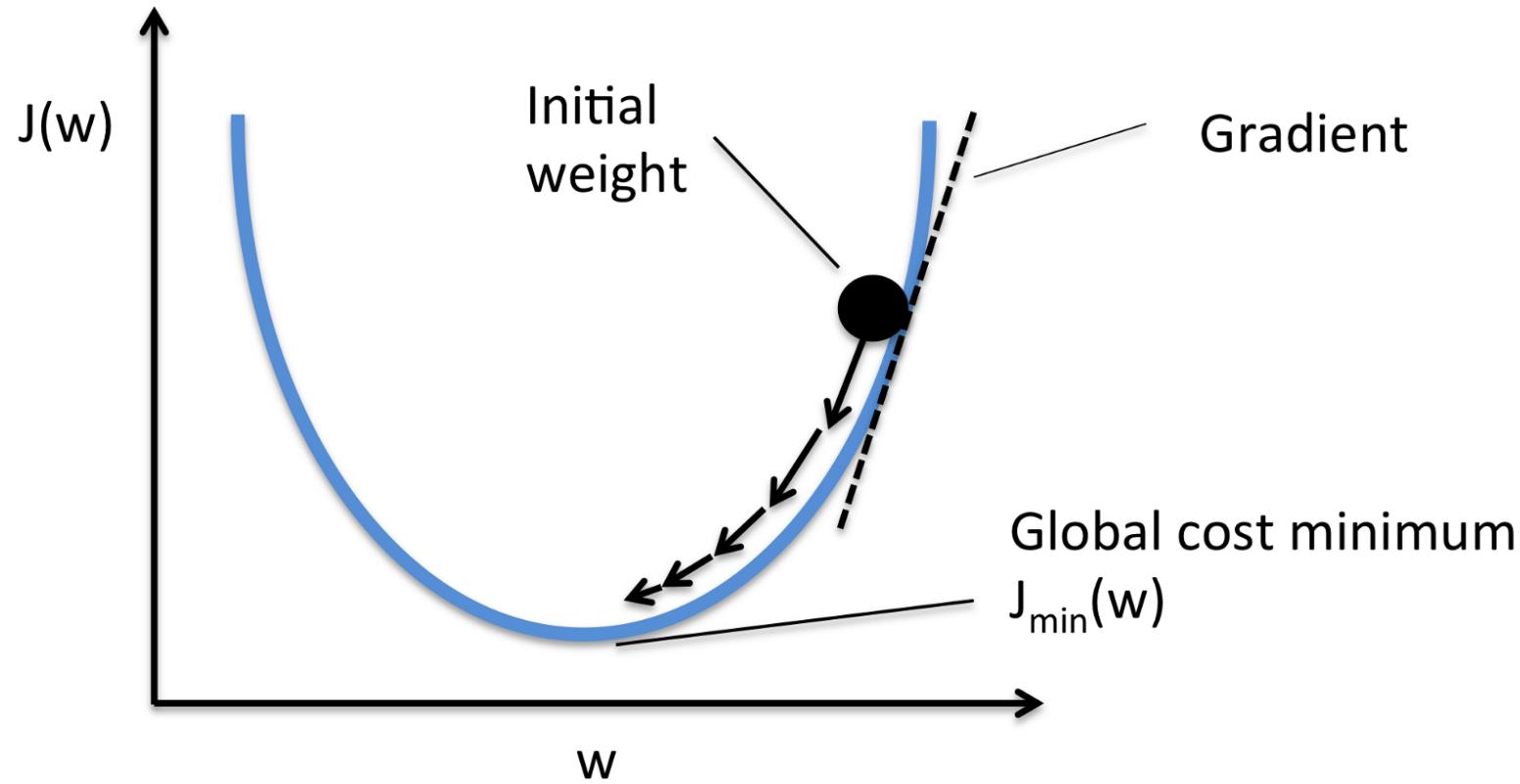
Calculus

- Derivative *기울기*
 - A function of a real variable measures the sensitivity to change of the function value (output value) with respect to a change in its argument (input value)

$$\frac{dy}{dx}$$



Calculus

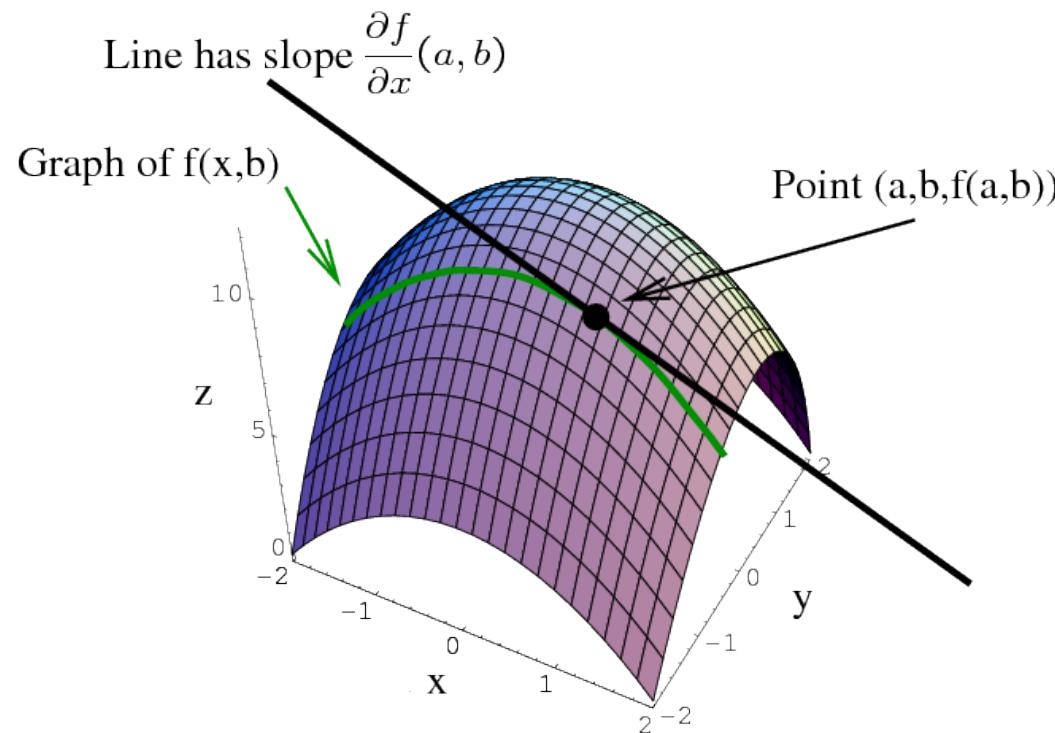


https://en.wikipedia.org/wiki/File:Gradient_Descent_in_2D.webm

Calculus

- Partial derivative *부분 미분*
 - A function of several variables is its derivative with respect to one of those variables, with the others held constant

$$\frac{\partial f}{\partial x}$$

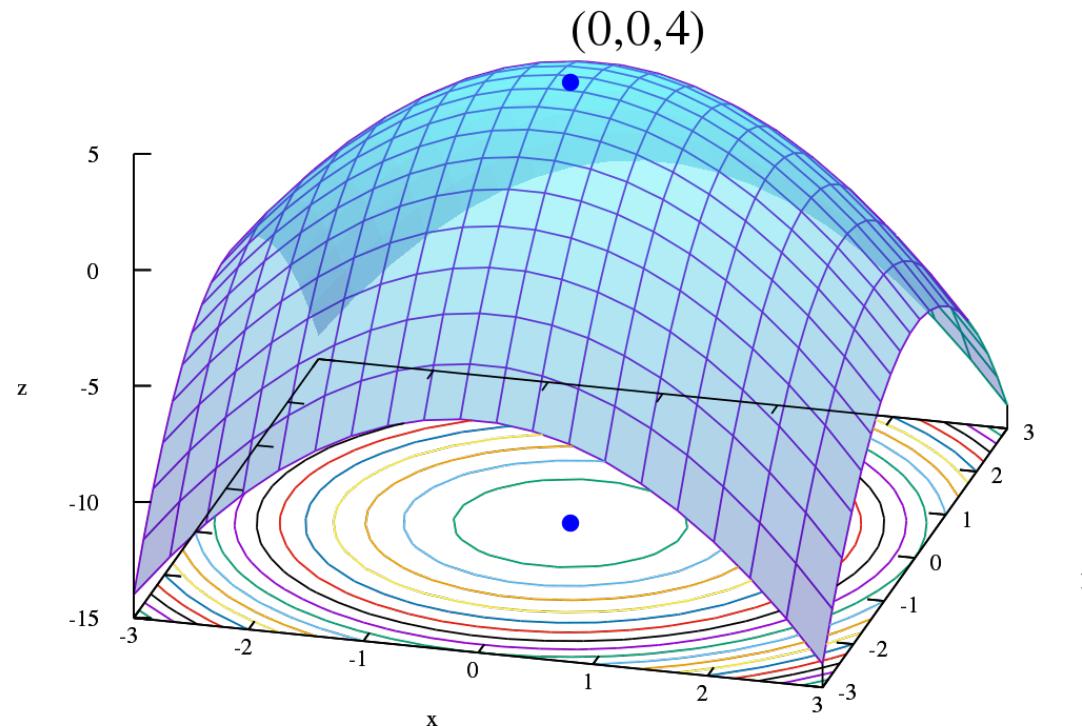


Optimization

- Optimization
 - ▣ Basics of optimization —how to formulate the problem
 - ▣ Linear programming, simplex algorithm
 - ▣ Integer programming
 - ▣ Constraint programming, knapsack problem
 - ▣ Randomized optimization techniques—hill climbing, simulated annealing, Genetic algorithms

Optimization

- Optimization problem
 - ▣ Maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function



Optimization

Example

- For materials, the manufacturer has 750 m² of cotton textile and 1,000 m² of polyester. Every pair of pants (1 unit) needs 1 m² of cotton and 2 m² of polyester. Every jacket needs 1.5 m² of cotton and 1 m² of polyester.
- The price of the pants is fixed at \$50 and the jacket, \$40.
- What is the number of pants and jackets that the manufacturer must give to the stores so that these items obtain a maximum sale?**

- Variables to be determined

$$x = \text{number of pants}$$

$$y = \text{number of jackets}$$

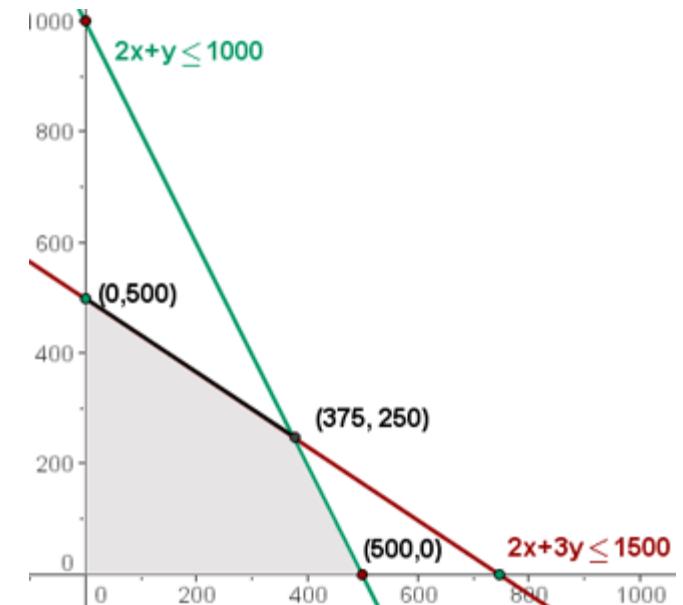
- Objective function

$$f(x, y) = 50x + 40y$$

- Constraints

$$x + 1.5y \leq 750$$

$$2x + y \leq 1000$$

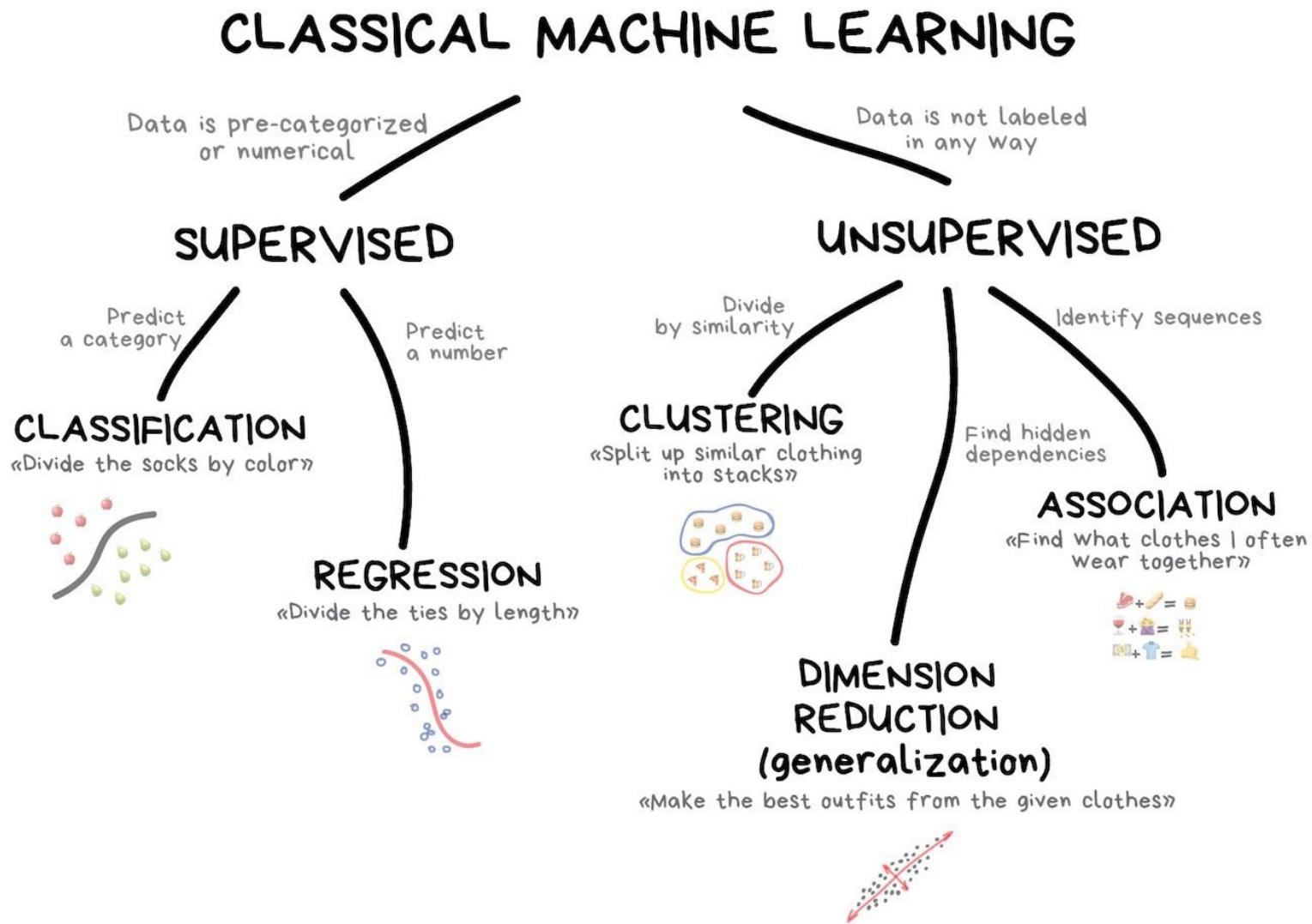


Optimization

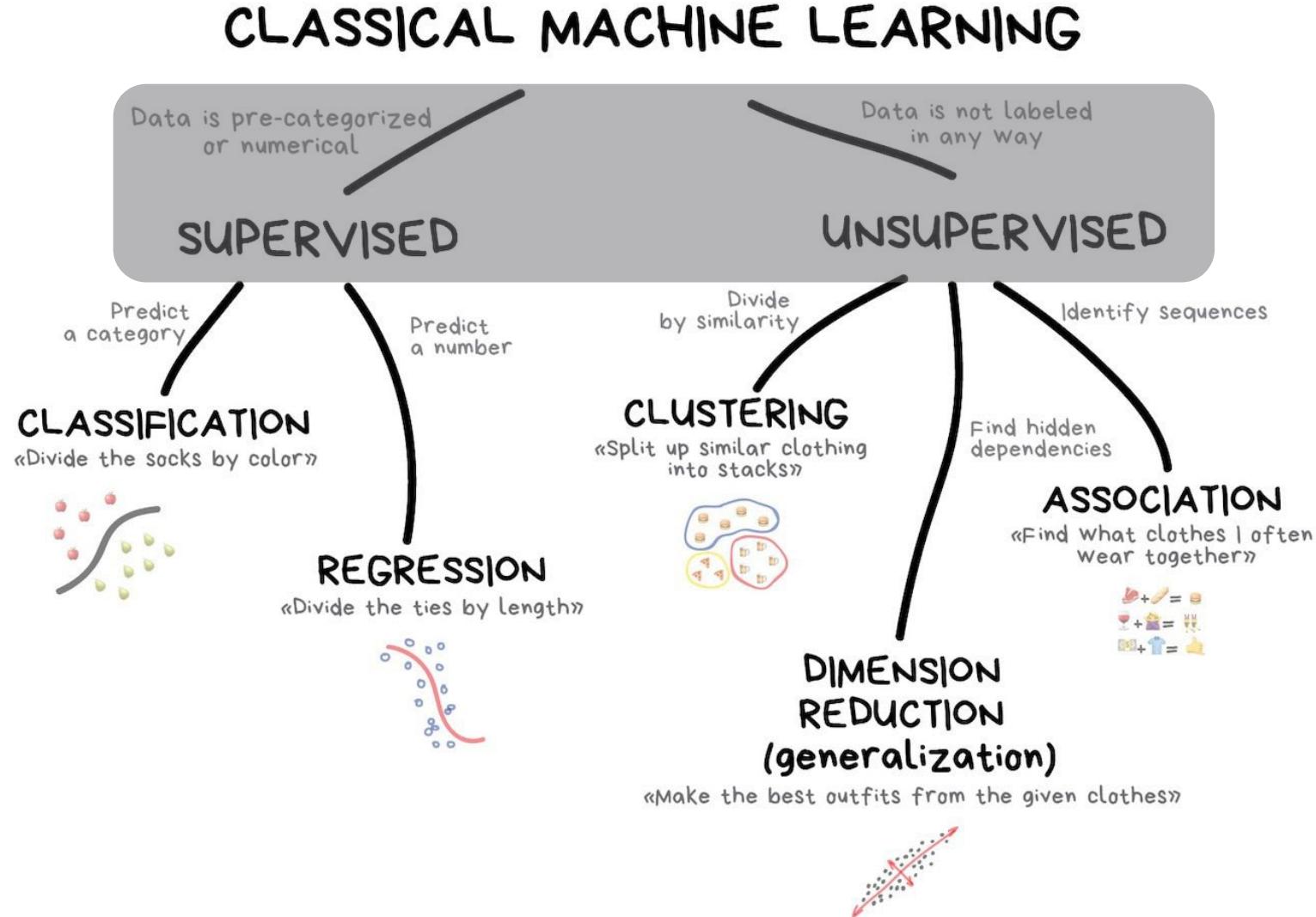
- Why are optimization algorithms important for data analysis?
 - ▣ One of fundamental data analysis tasks is to seek a function that approximately maps \mathbf{x}_i to y_i for each observation, i
$$y = f(\mathbf{x})$$
 - ▣ The process of finding f based on data is called learning or training
 - During learning, optimization algorithms provide a tool to find the most appropriate f

Basic Terminologies

Topics Covered in This Class



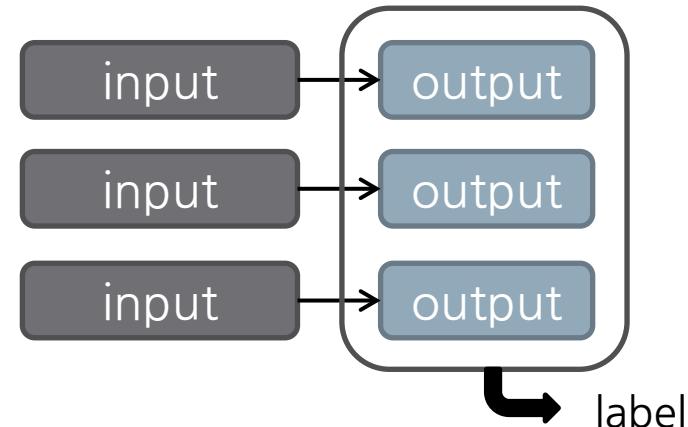
Topics Covered in This Class



Types of Learning

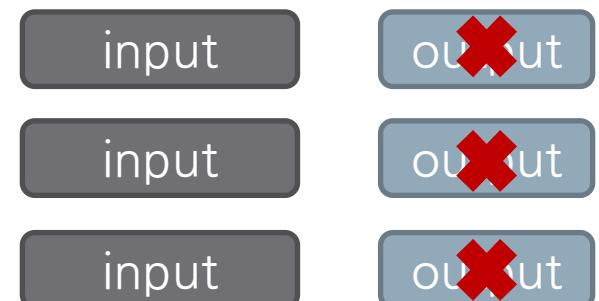
□ Supervised learning

- We have knowledge of output
 - We call such data **labeled**
 - **We know answer**
- Goal
 - Estimate output for unlabeled input

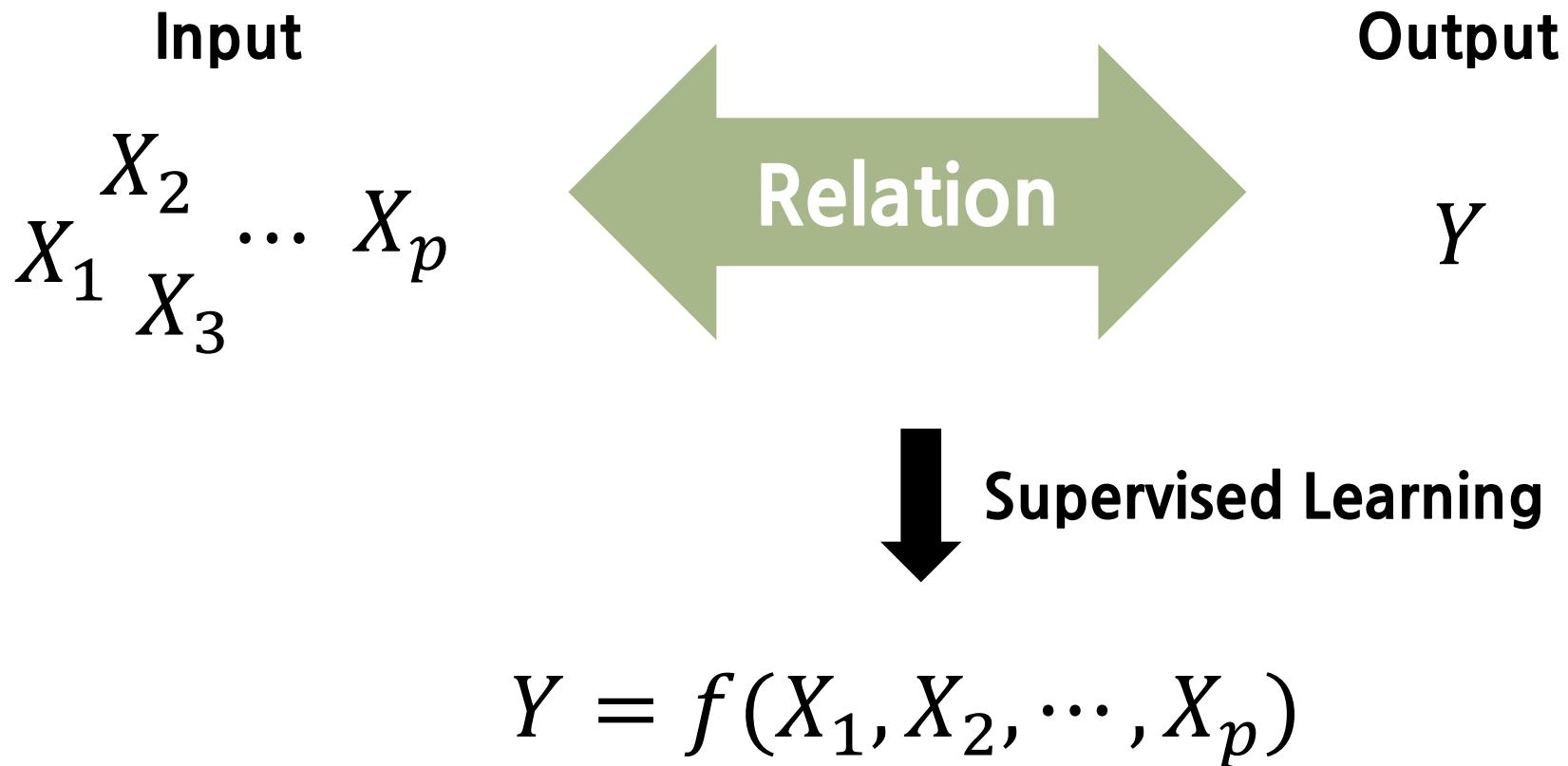


□ Unsupervised learning

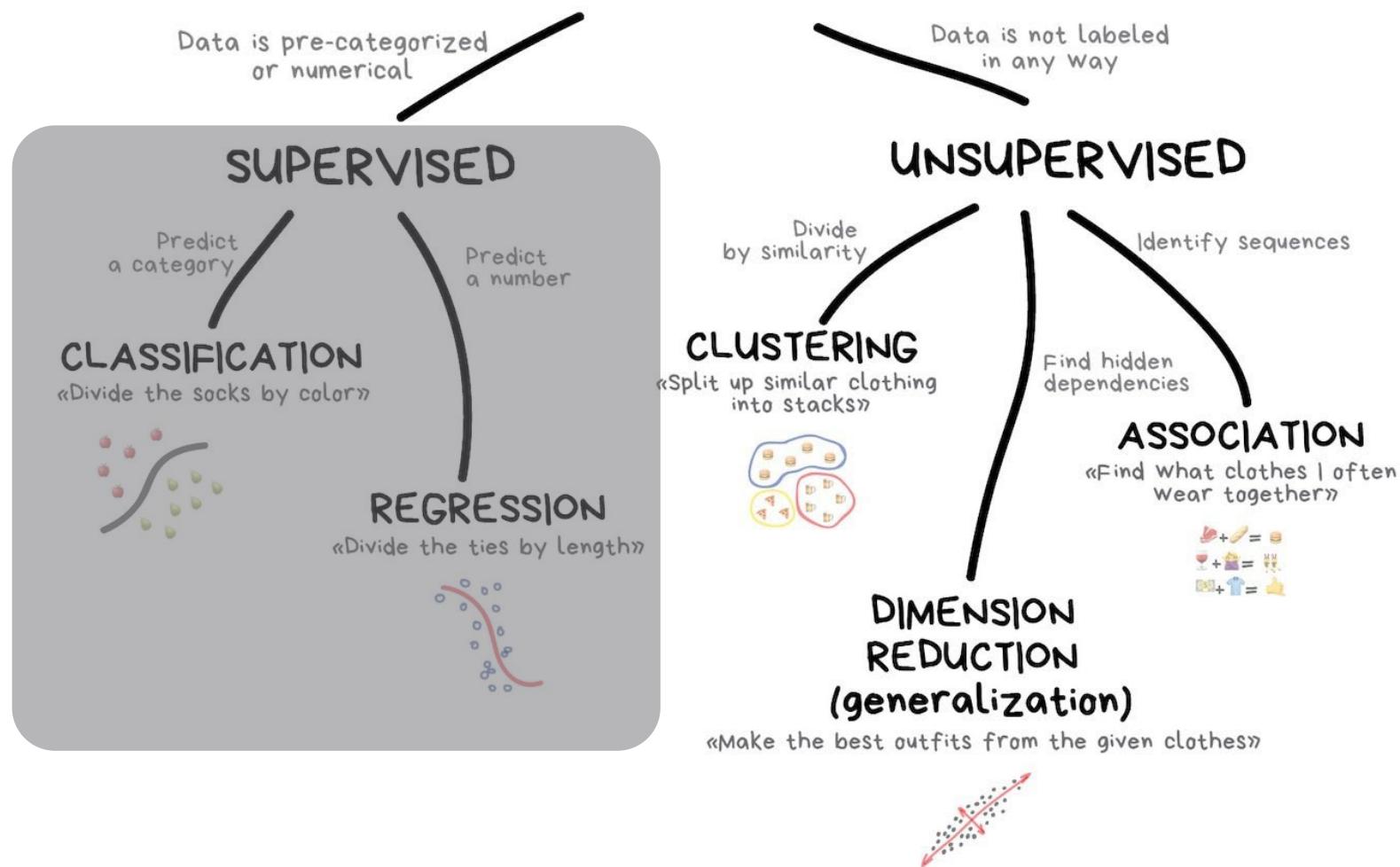
- **No output**
 - We call such data **unlabeled**
- Goal
 - Find patterns, groups, or relation



Supervised learning



Topics Covered in This Class



Data for Data Mining: Structured Data

- Example of data set
 - The input data set is usually expressed as a set of independent instances

Structured data.

instance,
sample,
example

Outlook	Temperature(°F)	Humidity(%)	Windy	Play Time(min)
Sunny	85	85	false	5
Sunny	80	90	true	0
Rainy	70	96	false	40
Rainy	68	80	false	65
Sunny	72	95	false	0
Sunny	69	70	false	70
Rainy	75	80	true	45

variable,
attribute,
feature

Types of Data

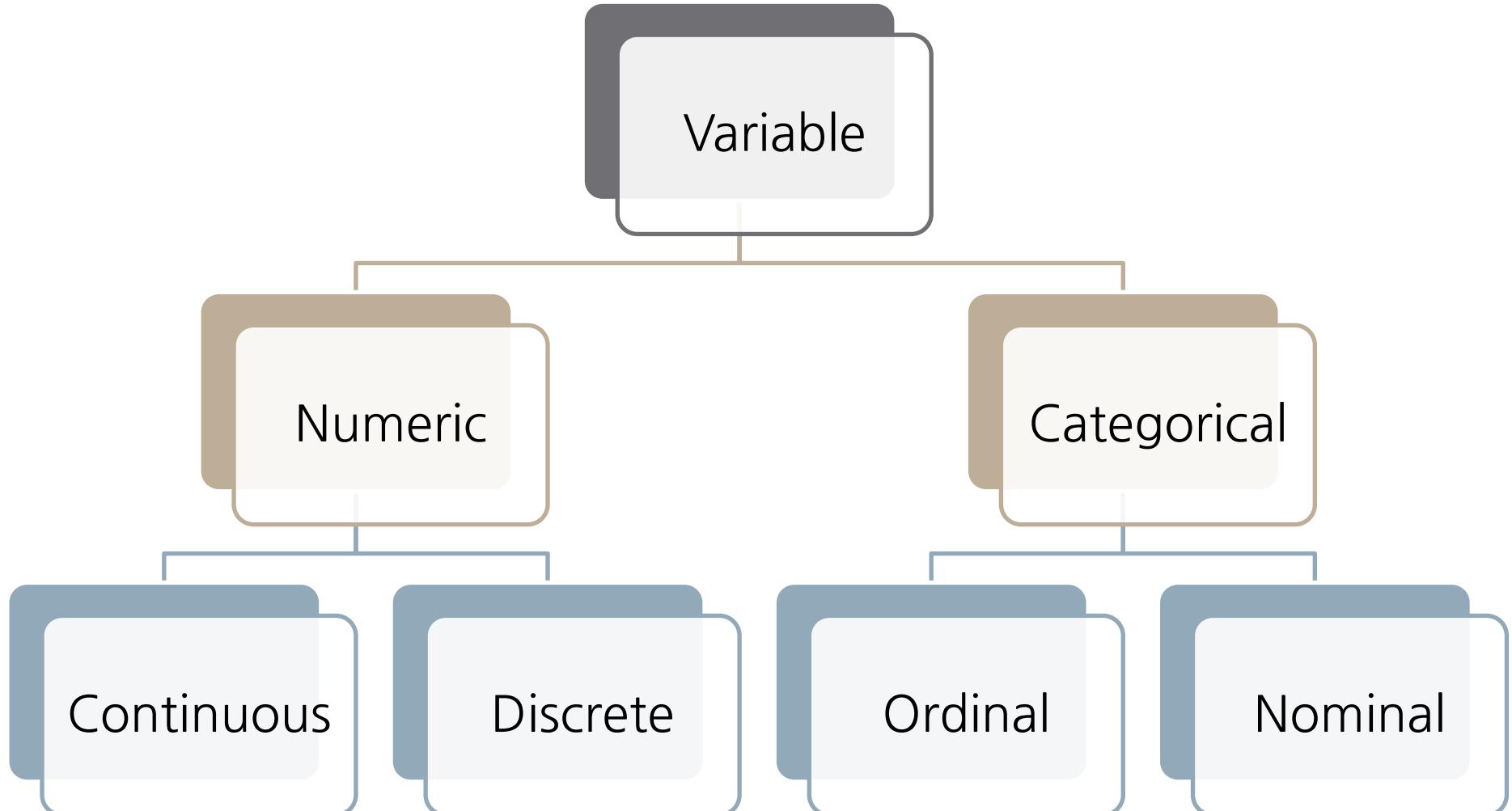
□ Structured

- Values of variable reside in a fixed field
- Examples
 - Numeric
 - Date
 - Restricted terms: (male, female), (Mr., Ms., Mrs.)
 - Address

□ Unstructured

- Values of variable do not reside in a fixed field
- Examples
 - Documents
 - Webpages
 - Images
 - Videos

Structured Data: Types of Variables



Structured Data: Types of Variables

□ Numeric (Quantitative)

- A broad category that includes any variable that can be counted, or has a numerical

Continuous

- A variable with infinite number of values
- Example
 - Many numeric variables: temperature, weight, height, pressure and etc.

Discrete

- A variable that can only take on a certain number of values or have a countable number of values between any two values
- Example
 - The number of cars in a parking lot
 - the number of flaws or defects

Structured Data: Types of Variables

□ Categorical

- A variable that contains a finite number of categories or distinct groups

□ Nominal

- A Variable that has two or more categories, but there is no intrinsic ordering to the categories.

■ Example

- (Male, Female), (Class 1, Class 2, Class 3), (Red, Yellow, Green)

□ Ordinal

- Similar to a nominal variable, but the difference between the two is that there is a clear ordering of the variables.

■ Example

- Score: A+, A, A-, B+, B, B-, C+, C, C-, D, F

- Size: S, M, L, XL, XXL

Example: The Input to a Data Mining

- Example of data set

num-of-doors	body-style	wheel-base	length	make
2	convertible	88.6	168.8	Audi
2	convertible	88.6	168.8	BMW
2	hatchback	94.5	171.2	Chevrolet
4	sedan	99.8	176.6	BMW
4	sedan	99.4	176.6	Audi
2	sedan	99.8	177.3	Audi
4	wagon	105.8	192.7	Chevrolet

Types:

Discrete

Nominal

Continuous

Continuous

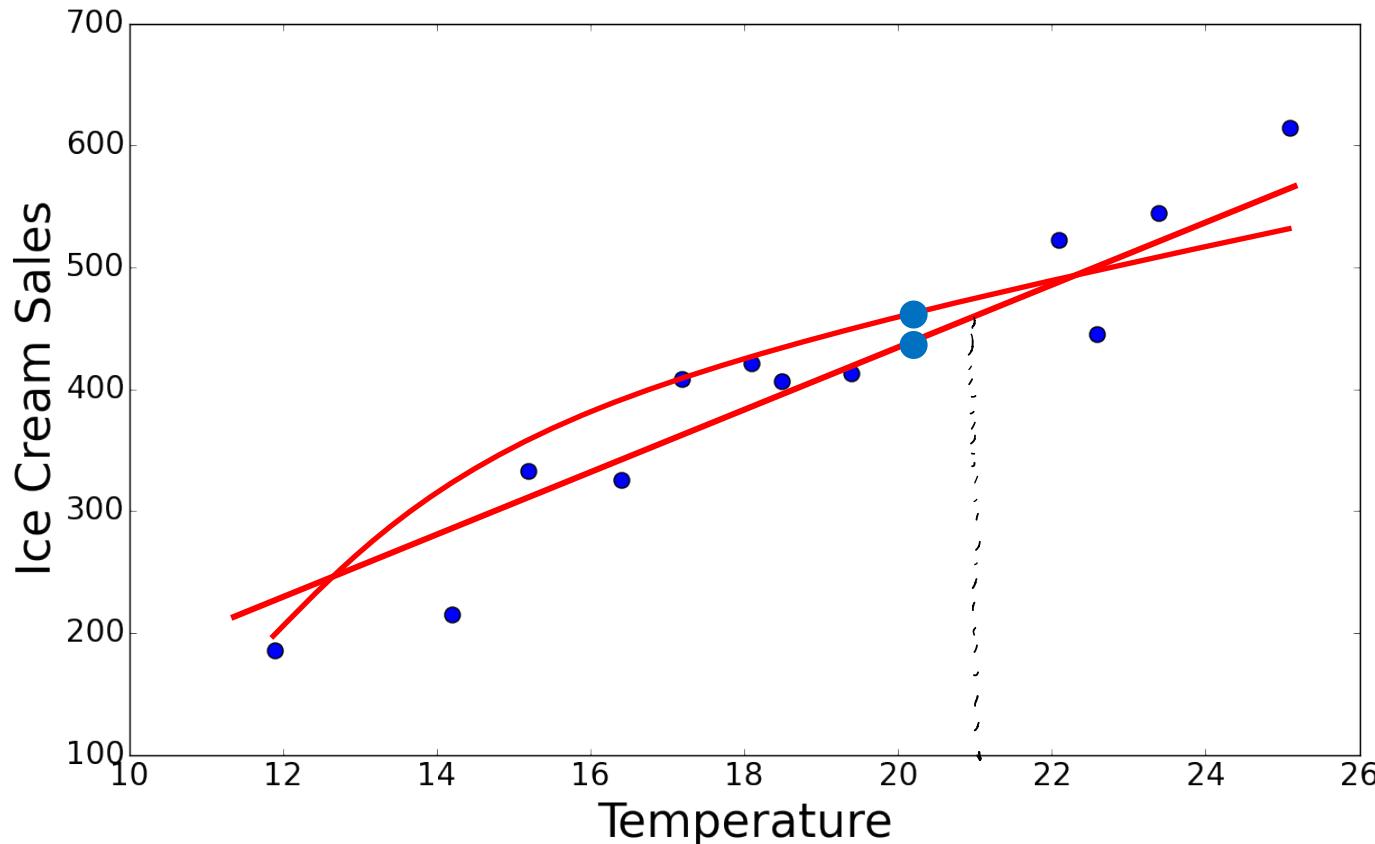
Nominal

Supervised Learning: Regression

eff

- Temperature vs. Ice Cream Sales
 - How about 21°C?

for continuous

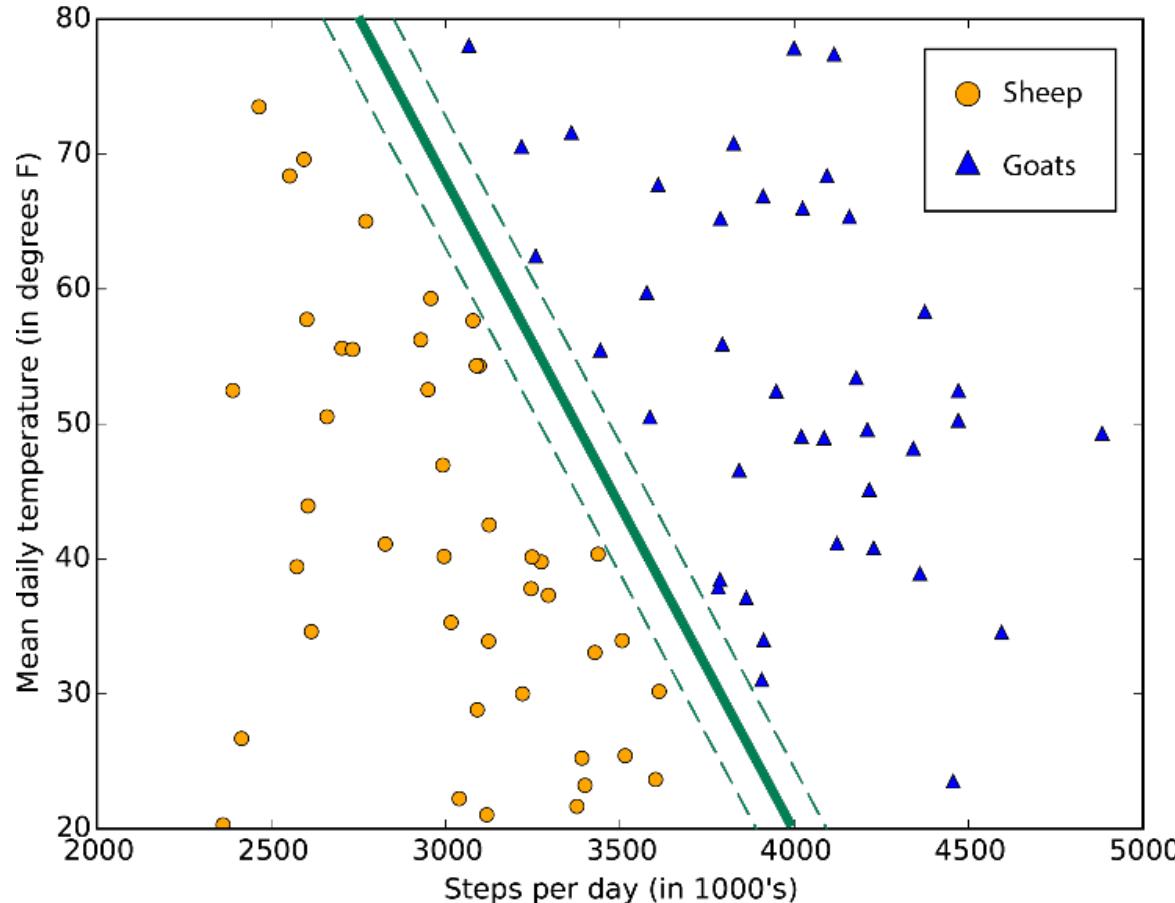


Supervised Learning: Classification

1d2
11.

- Which one is a sheep?

for discrete value · yes/no



Question

- Suppose you are working on weather prediction, and you would like to predict whether or not it will be raining at 5pm tomorrow. Y/N
You want to use a learning algorithm for this. Would you treat this as a classification or a regression problem?
 - ① Regression
 - ② Classification
- Suppose you are working on stock market prediction, and you would like to predict the price of the specific stock tomorrow (measured in dollars).
You want to use a learning algorithm for this. Would you treat this as a classification or a regression problem?
 - ① Regression
 - ② Classification

Overfitting vs. Underfitting

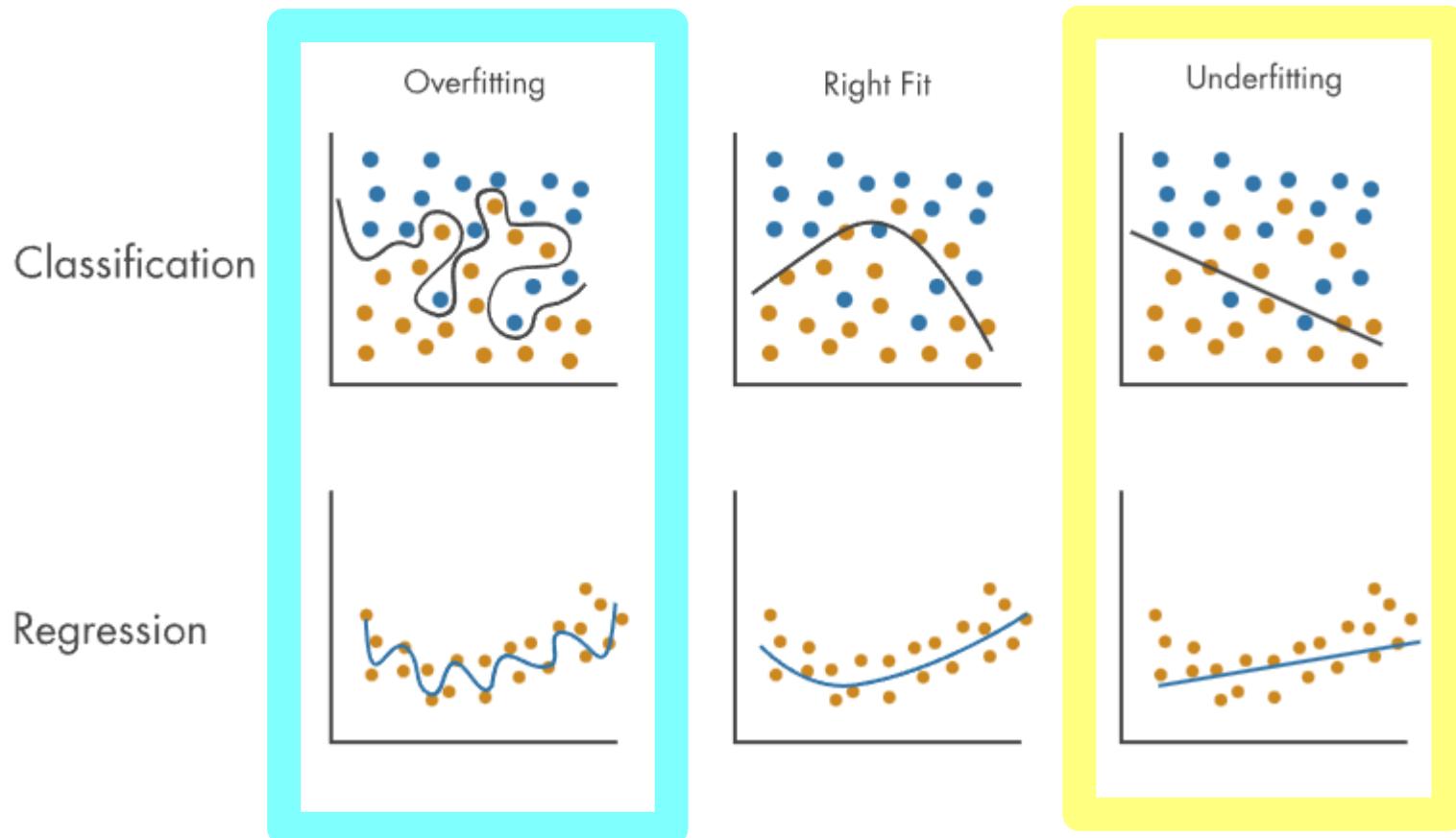
□ Overfitting

- Overfitting is a machine learning problem that occurs when a model is too closely aligned to training data, causing it to perform poorly on new data.
- How it happens
 - The model is too complex
 - The training data is too small or contains irrelevant information
 - The model memorizes subtle patterns in the training data
- Why it's a problem
 - An overfit model can't generalize well to new data
 - It can give inaccurate predictions
 - It can't perform well for all types of new data

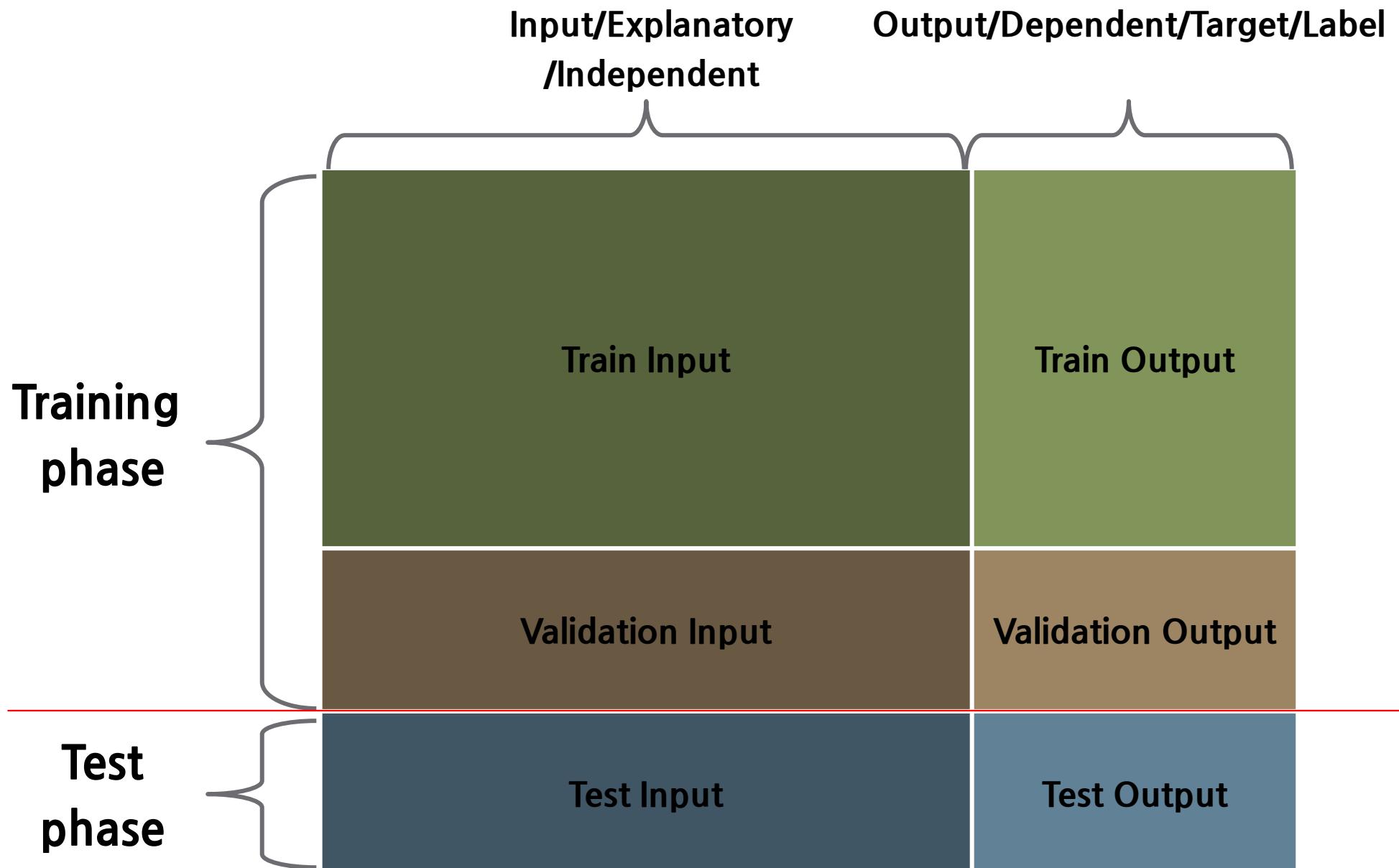
□ Underfitting

- Underfitting occurs when a machine learning model is too simple to capture the underlying patterns in data.

Overfitting vs. Underfitting



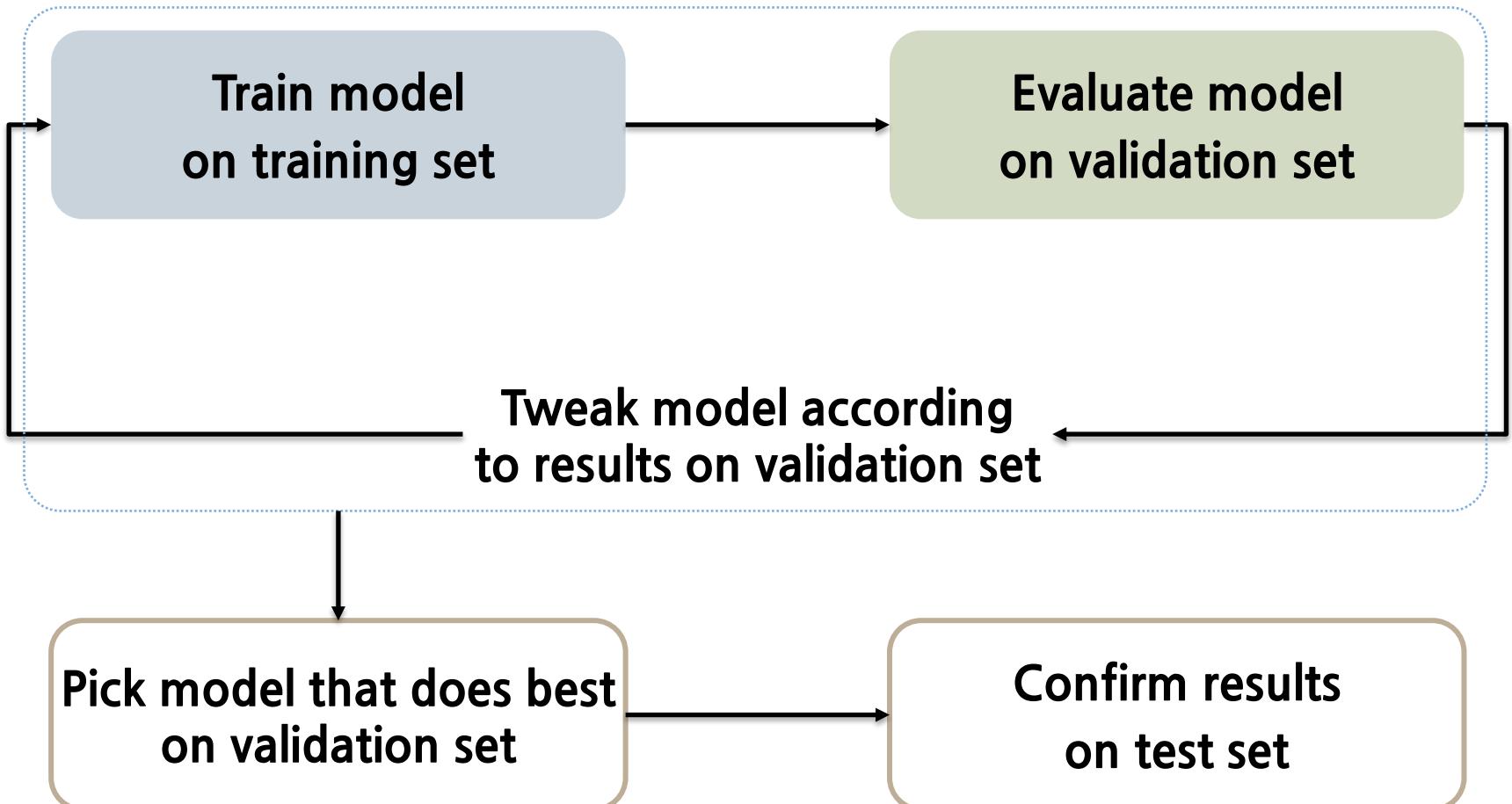
Data Partition



Data Partition

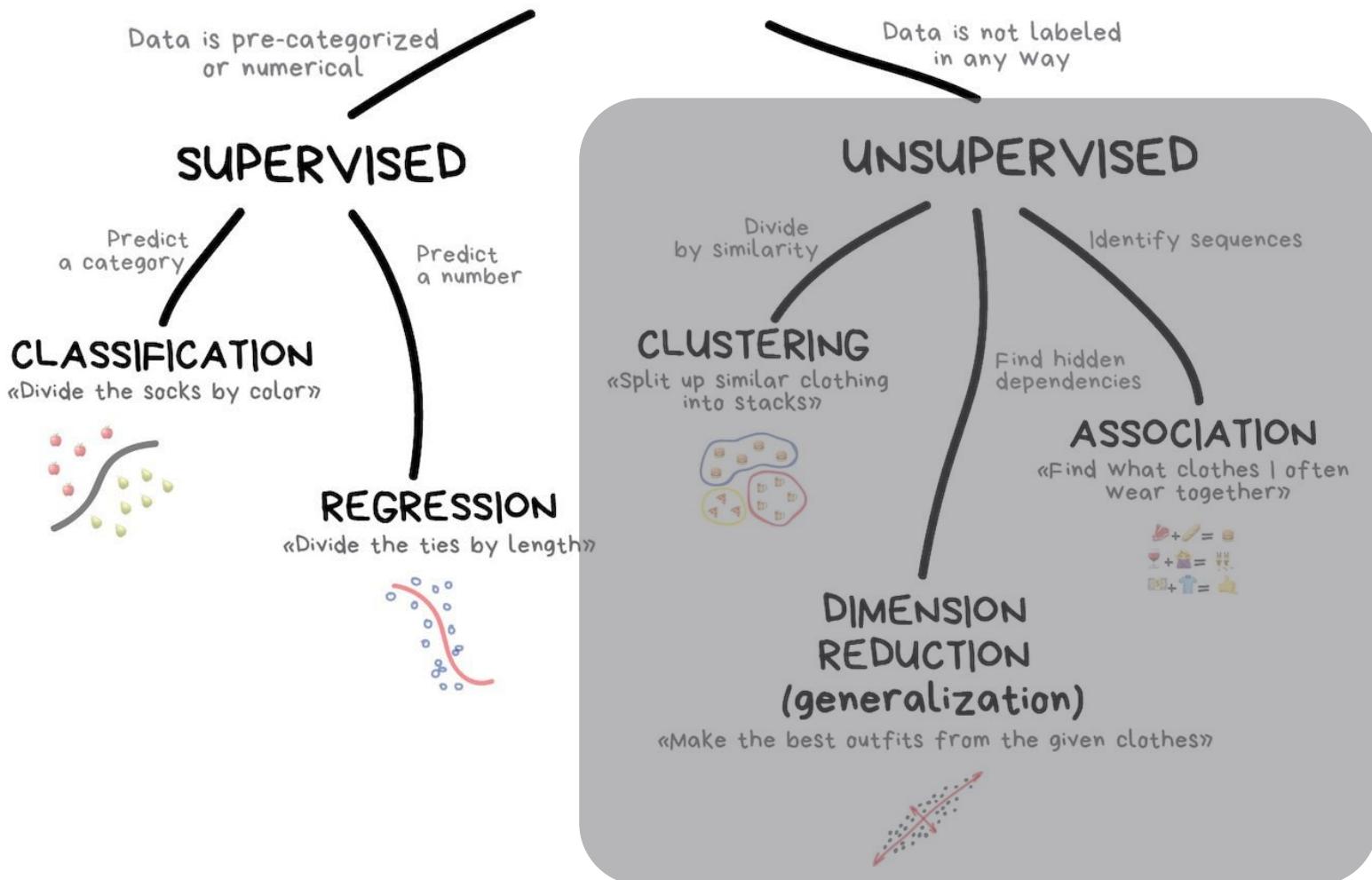
- Training set
 - ▣ Purpose: The training set is used to train the model. It contains the labeled examples the model will learn from. During the training phase, the model's parameters are adjusted based on the data in the training set.
- Validation set
 - ▣ Purpose: The validation set is used to tune hyperparameters and evaluate the model during training. It helps in selecting the best version of the model.
- Test set
 - ▣ Purpose: The test set is used to evaluate the model's final performance after training and validation. This set simulates new, unseen data, giving an unbiased estimate of how the model will perform in a real-world scenario.

Process of Supervised Learning with Partitioned Data



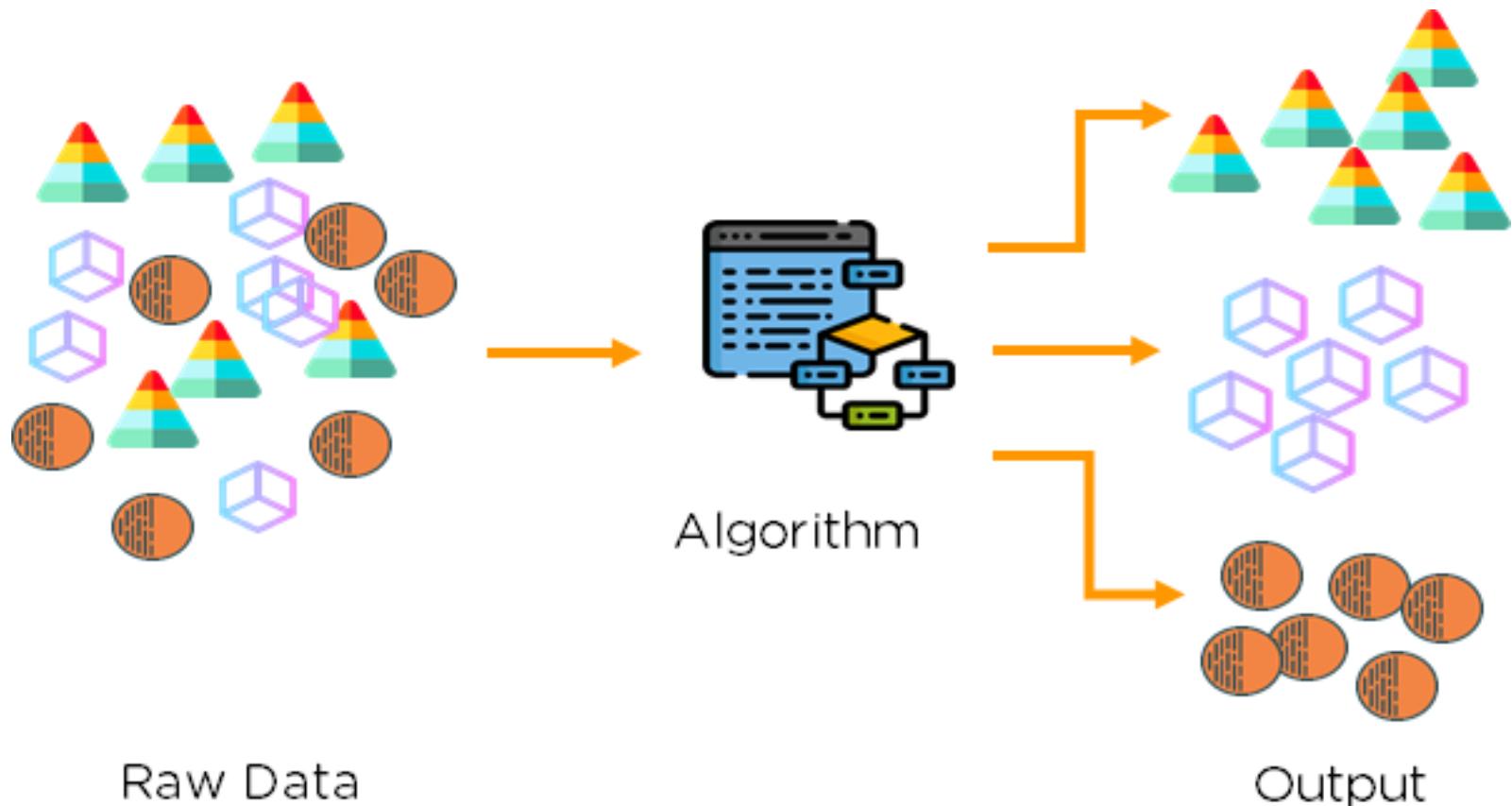
Topics Covered in This Class

CLASSICAL MACHINE LEARNING



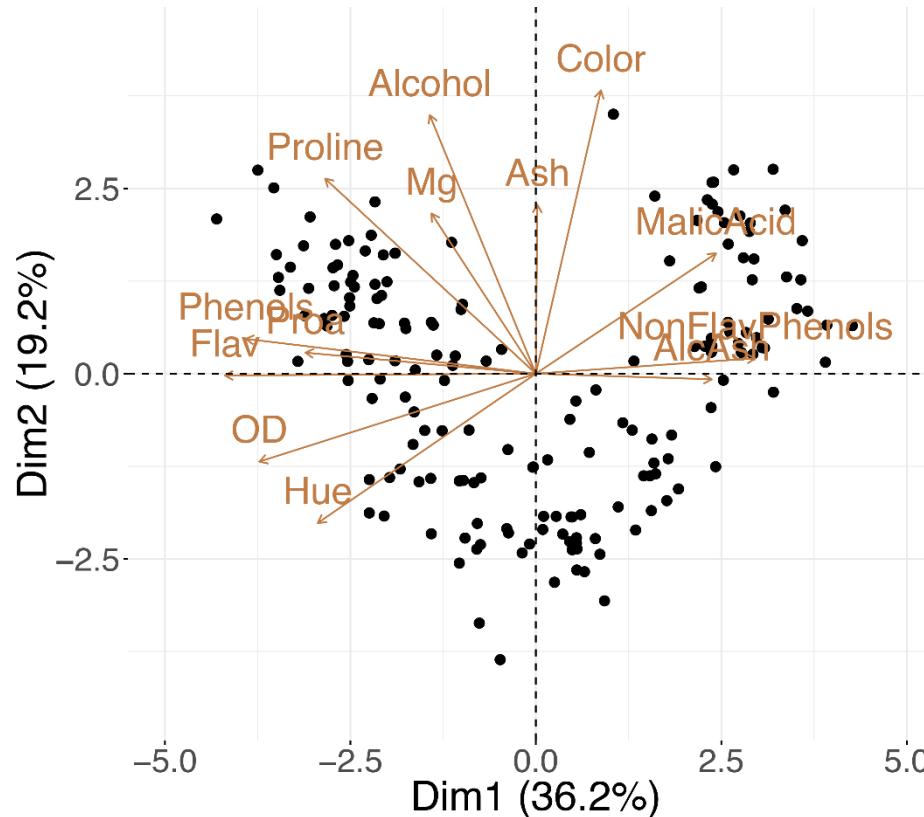
Unsupervised Learning: Clustering

- Grouping data points
 - How to determine which **group** does each data belongs to?



Unsupervised Learning: Dimensionality Reduction

- Dimensionality reduction
 - The process of reducing the number of random variables under consideration by obtaining a set of principal variables
 - High dimension → Low dimension



Unsupervised Learning: Association Rule Mining

similar with Search algorithm

- Find useful information from transactions

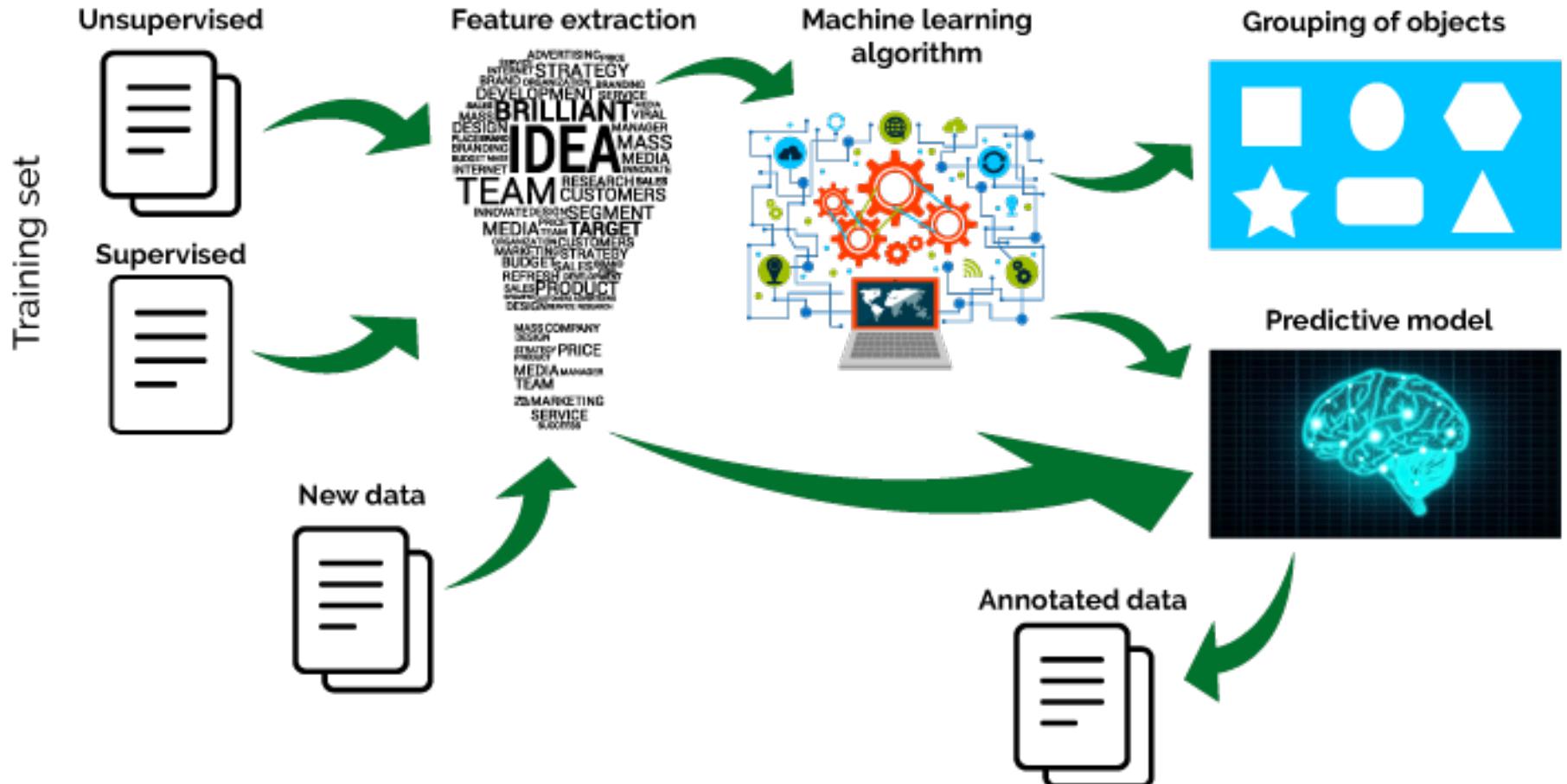
Datetime	Customer	Items
2015-07-15 14:03	1	orange juice, banana
2015-07-15 16:20	2	orange juice, milk
2015-07-16 10:14	3	detergent, banana, orange juice
2015-07-25 19:34	2	milk, bread, soda
2015-07-29 09:41	4	detergent, window cleaner
2015-08-01 20:55	1	bread, milk

- One of useful information is information like “If item A then item B”
 - This information is called association rule
- Find pair of items that are more likely to be purchased together based on transactions

Question

- Of the following examples, which would you address using an unsupervised learning algorithm? (Find all that apply.)
 - ① Given email labeled as spam/not spam, learn a spam filter. *supervised classify*
 - ② Given a set of news articles found on the web, group them into set of articles about the same story. *unsupervised clustering*
 - ③ Given a database of customer data, automatically discover market segments and group customers into different market segments. *unsupervised clustering*
 - ④ Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not. *supervised classification*

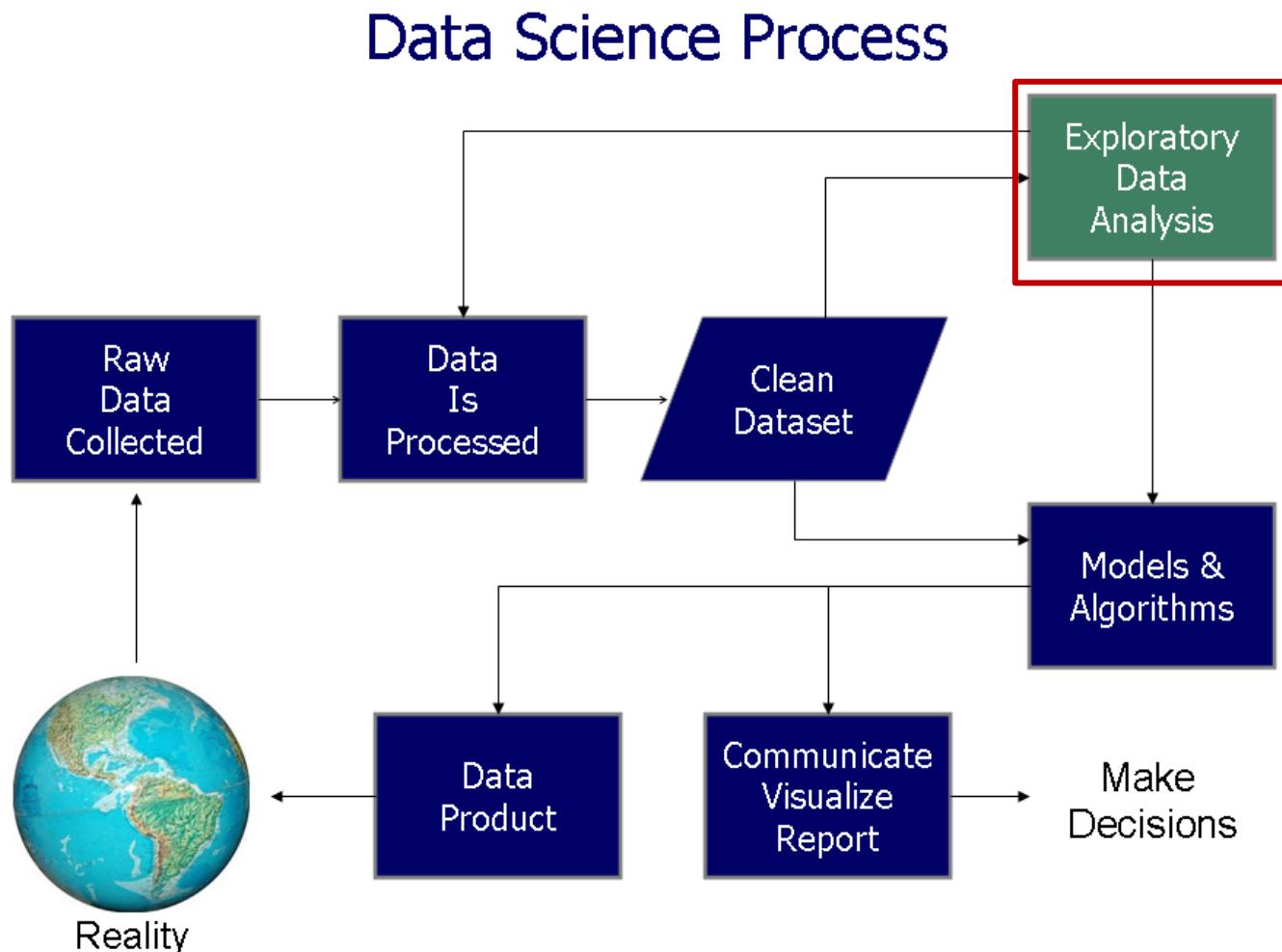
Overall Description



EXPLANATORY DATA ANALYSIS

Week03

Data Science Process



Explanatory Data Analysis

What is Exploratory Data Analysis (EDA)?

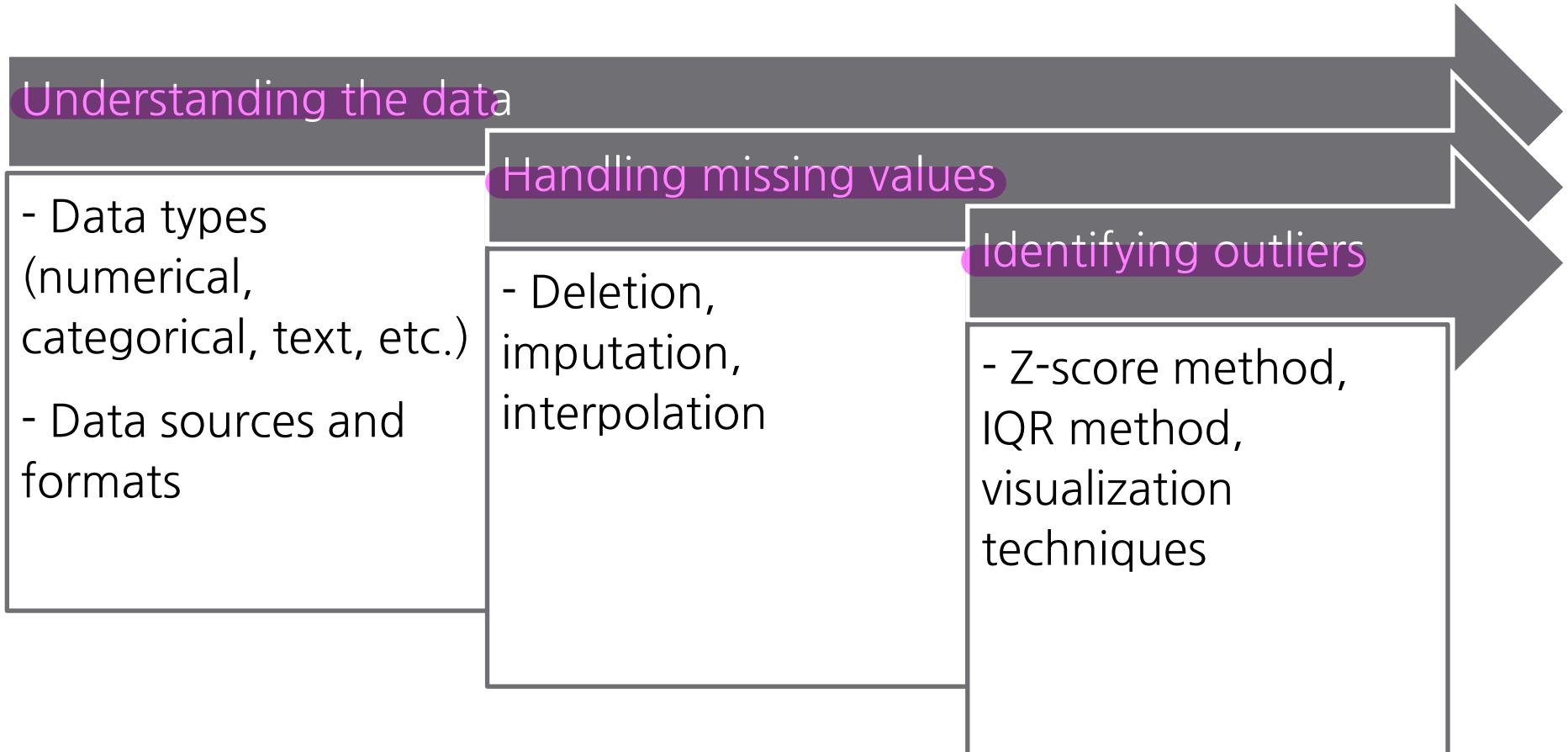
- Definition
 - ▣ EDA is an approach to analyzing data sets to summarize their main characteristics, often with visual methods
- Purpose
 - ▣ Identify patterns, detect anomalies, test hypotheses, and check assumptions with summary statistics and graphical representations
- Key Techniques
 - ▣ Data visualization, summary statistics, handling missing values, detecting outliers

Importance of EDA

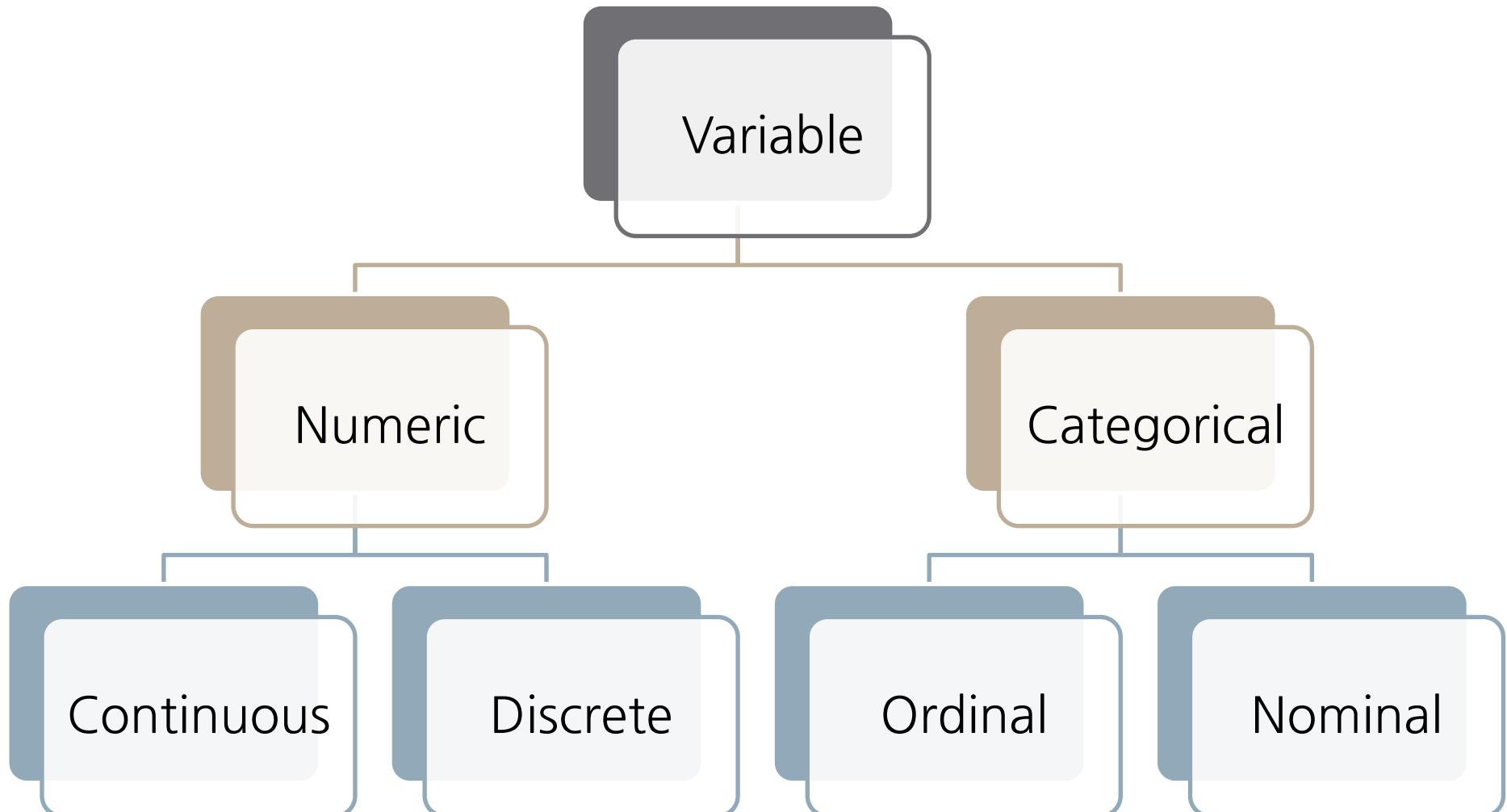
- Understanding the Data
 - Identify data distributions and relationships
 - Detect missing values and anomalies
- Data Preprocessing
 - Helps in feature selection and engineering
 - Assists in choosing appropriate machine learning models
- Insights & Decision Making
 - Provides actionable insights before model building
 - Supports better decision-making in data-driven projects.

Data Transformation?

Steps in EDA



Understanding Data: Types of Variables



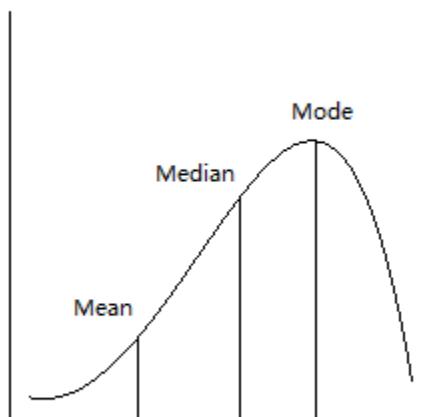
Understanding Data: Summary Statistics

- Measures of central tendency

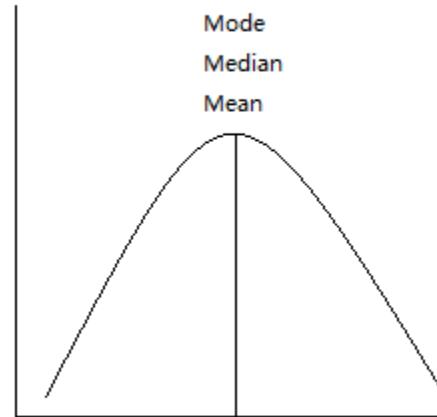
- Mean: The average value of a dataset

$$\mu = \frac{\sum_i x_i}{n}$$

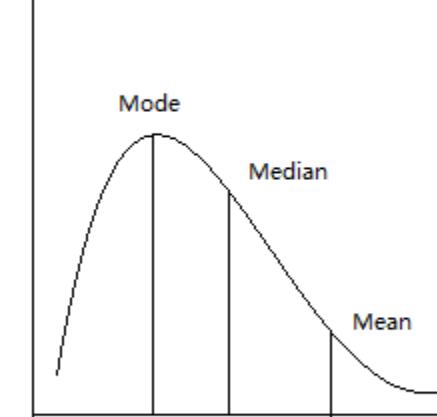
- Median: The middle value in a sorted dataset. If the number of observations (n) is odd, the median is the middle value. If n is even, the median is the average of the two middle values
 - Mode: The most frequently occurring value in the dataset



Left skew



Normal Distribution



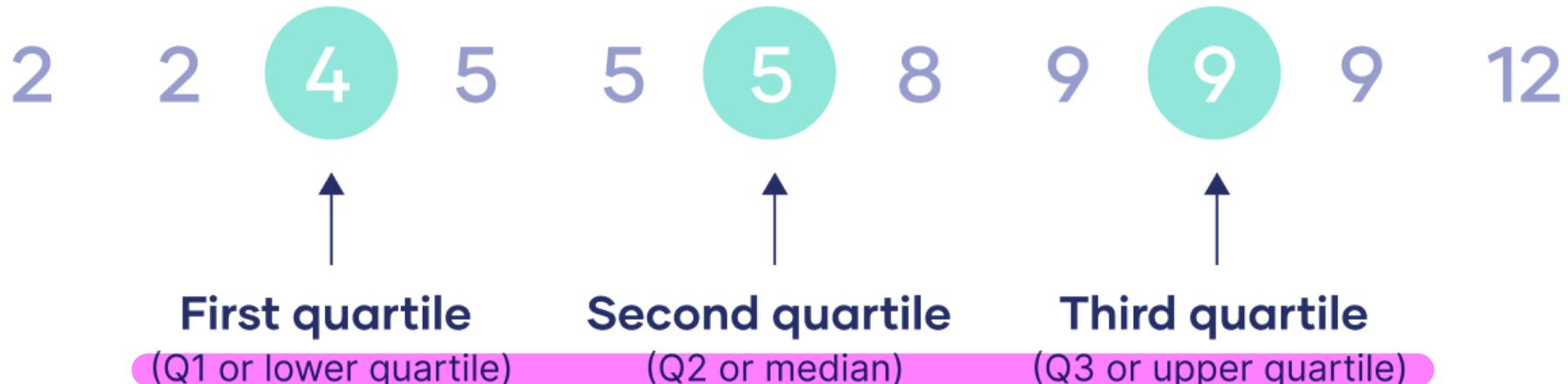
Right skew

Understanding Data: Summary Statistics

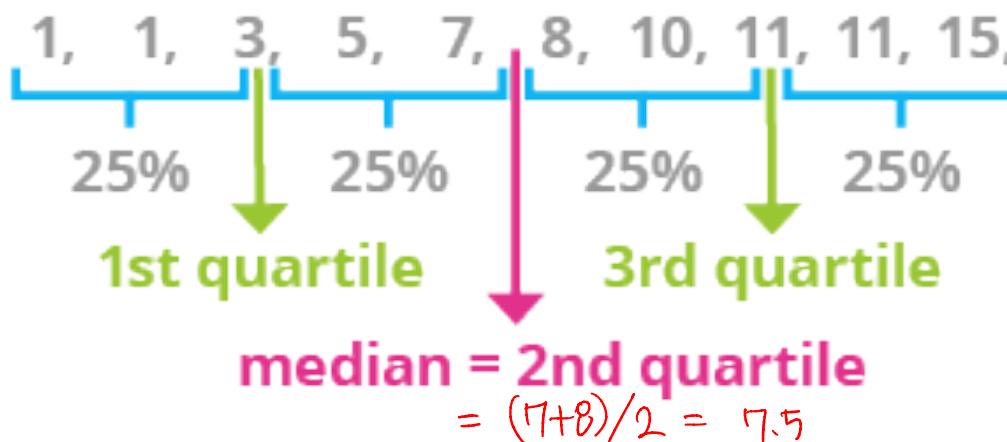
- Measures of spread
 - ▣ Variance: Measures the average squared deviation from the mean
 - Population
 - Sample
$$\sigma^2 = \frac{\sum_i (x_i - \mu)^2}{n}$$
 - ▣ Standard Deviation: The square root of the variance, indicating dispersion
 - Population
 - Sample
$$\sigma = \sqrt{\frac{\sum_i (x_i - \mu)^2}{n}}$$
 - ▣ Range: The difference between the maximum and minimum values in the dataset
$$Range = \max(x) - \min(x)$$
 - ▣ Inter-quartile range (IQR): Measures the spread of the middle 50% of data
$$IQR = Q_3 - Q_1$$

Understanding Data: Summary Statistics

odd case



even case



Understanding Data: Summary Statistics

- Skewness *entz*

- Skewness measures the symmetry of a distribution

- Population

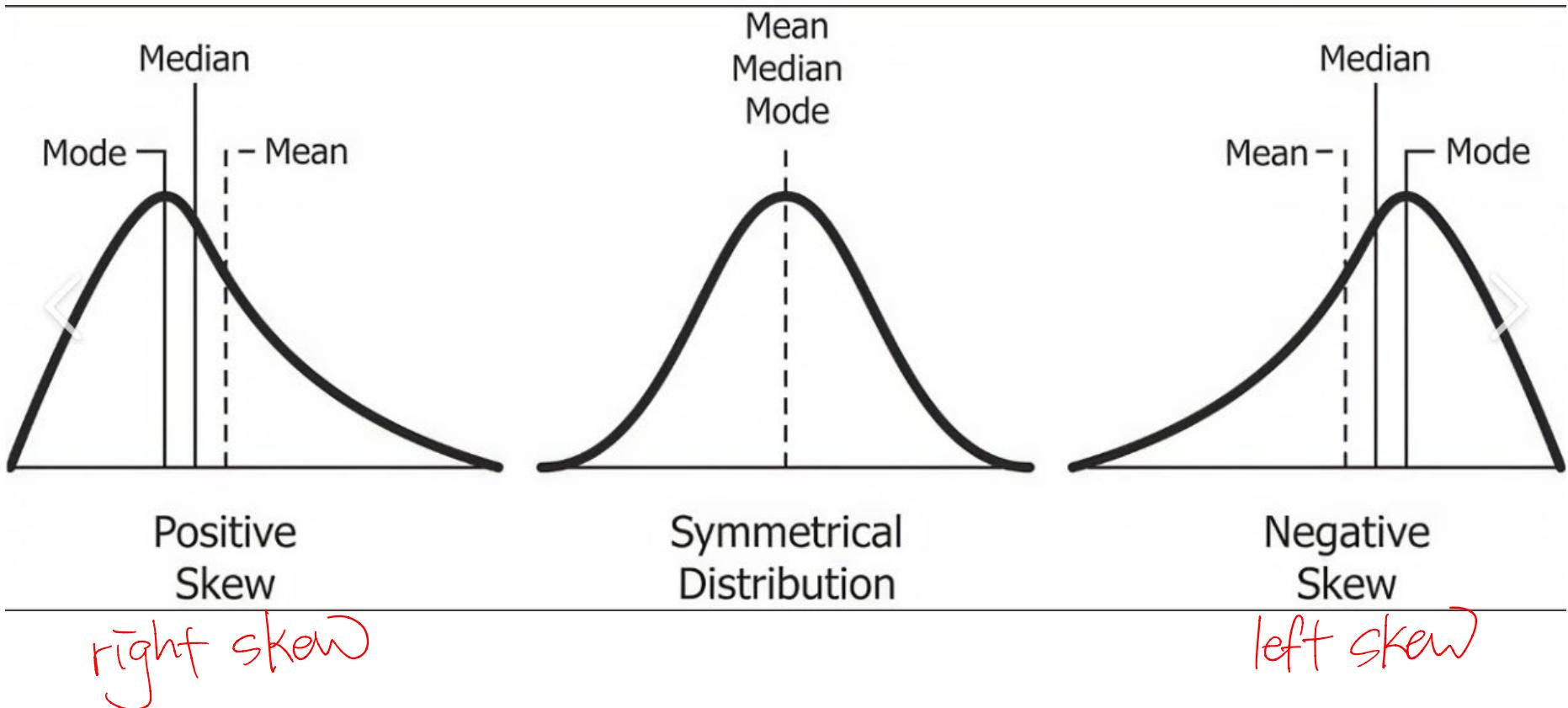
$$\gamma_1 = \mathbb{E} \left[\left(\frac{X - \mu}{\sigma} \right)^3 \right]$$

- Sample

$$b_1 = \frac{m_3}{s^3} = \frac{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}{\left[\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right]^{3/2}}$$

- Positive skewness: The tail extends to the right (more values are on the left side of the mean).
 - Negative skewness: The tail extends to the left (more values are on the right side of the mean)
 - A skewness value of zero indicates a symmetric distribution

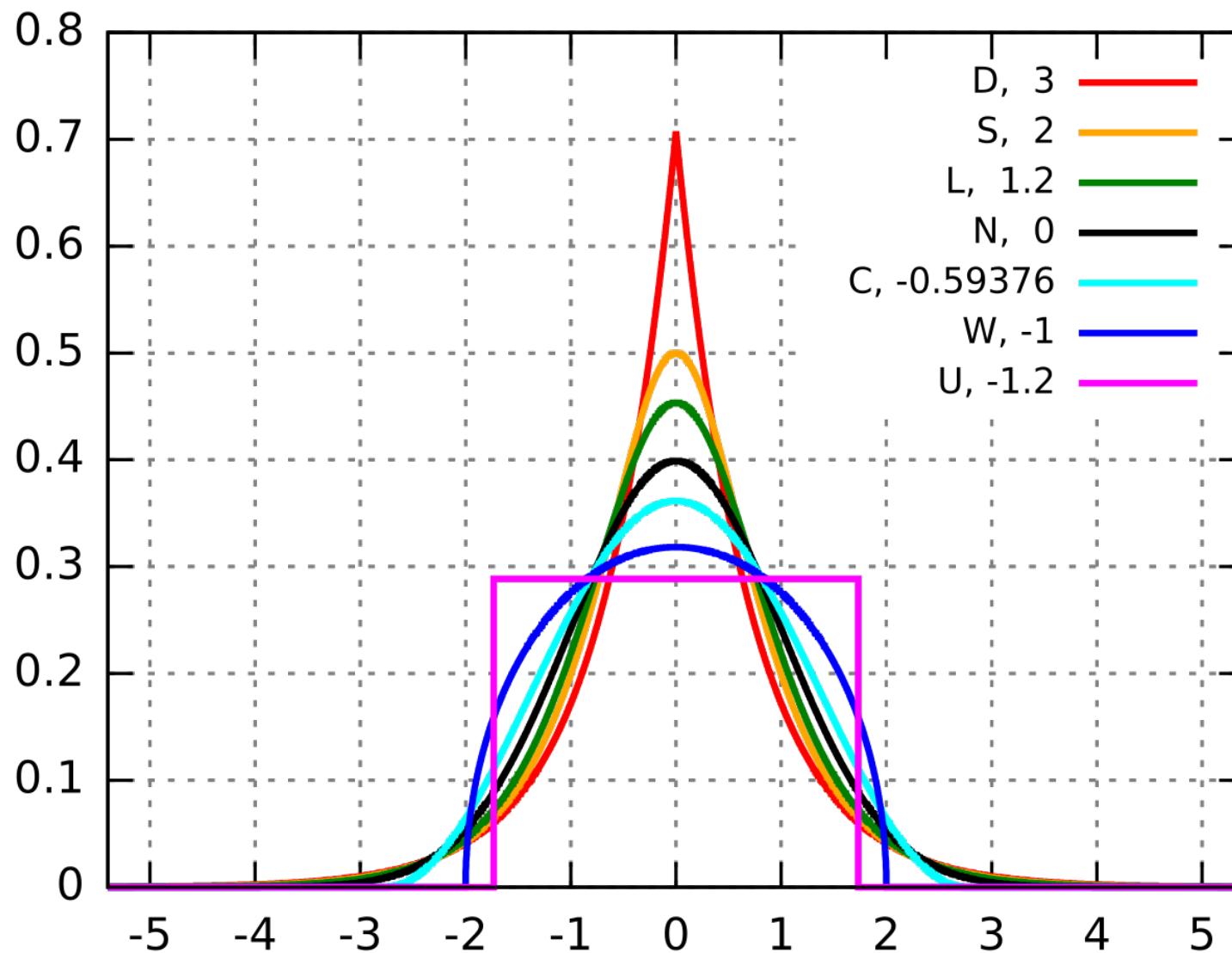
Understanding Data: Summary Statistics



Understanding Data: Summary Statistics

- Kurtosis *K_{UZ}*
 - ▣ Kurtosis measures the “tailedness” or “peakedness” of a distribution compared to a normal distribution
 - Population
 - Sample
 - Excess kurtosis (Fisher kurtosis)
$$Kurt[X] = \mathbb{E} \left[\left(\frac{X - \mu}{\sigma} \right)^4 \right]$$
$$\kappa = \frac{m_4}{s^4}$$
 - Positive kurtosis (leptokurtic): The distribution has heavier tails and a sharper peak than a normal distribution
 - Negative kurtosis (platykurtic): The distribution has lighter tails and a flatter peak than a normal distribution
 - A kurtosis value of 3 (or 0 for excess kurtosis) indicates a distribution similar to a normal distribution (mesokurtic)

Understanding Data: Summary Statistics



Understanding Data: Summary Statistics

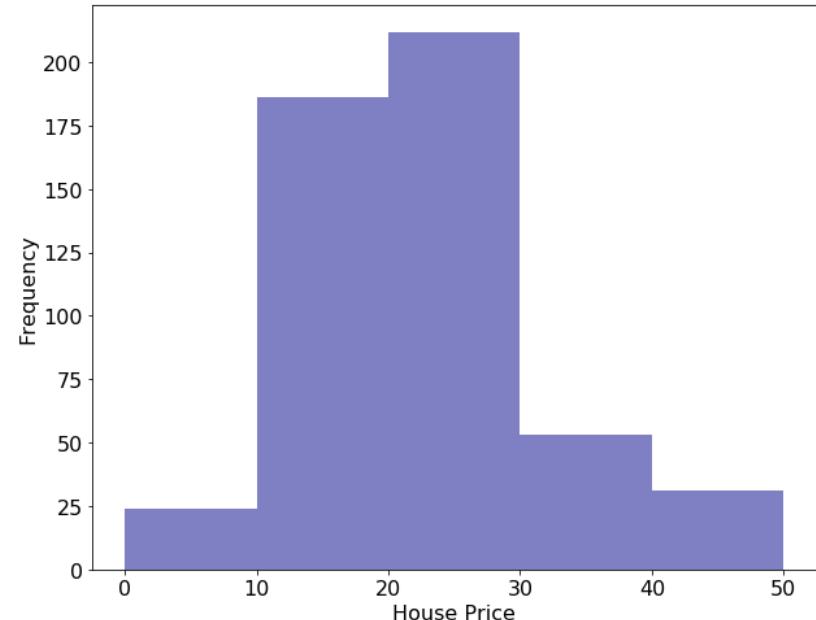
- Data distribution
 - ▣ Histograms: A graphical representation of the distribution of numerical data. It divides the dataset into bins and counts the number of observations within each bin, showing how data values are distributed across different ranges

24.0, 21.6, 34.7, 33.4, 36.2, 28.7, 22.9,
27.1, 16.5, 18.9, 15.0, 18.9, 21.7, 20.4,
18.2, 19.9, 23.1, 17.5, 20.2, 18.2, 13.6,
19.6, 15.2, 14.5, 15.6, 13.9, 16.6, 14.8
... ...



Bin	Count
0 to 10	24
10 to 20	186
20 to 30	212
30 to 40	53
40 to 50	31

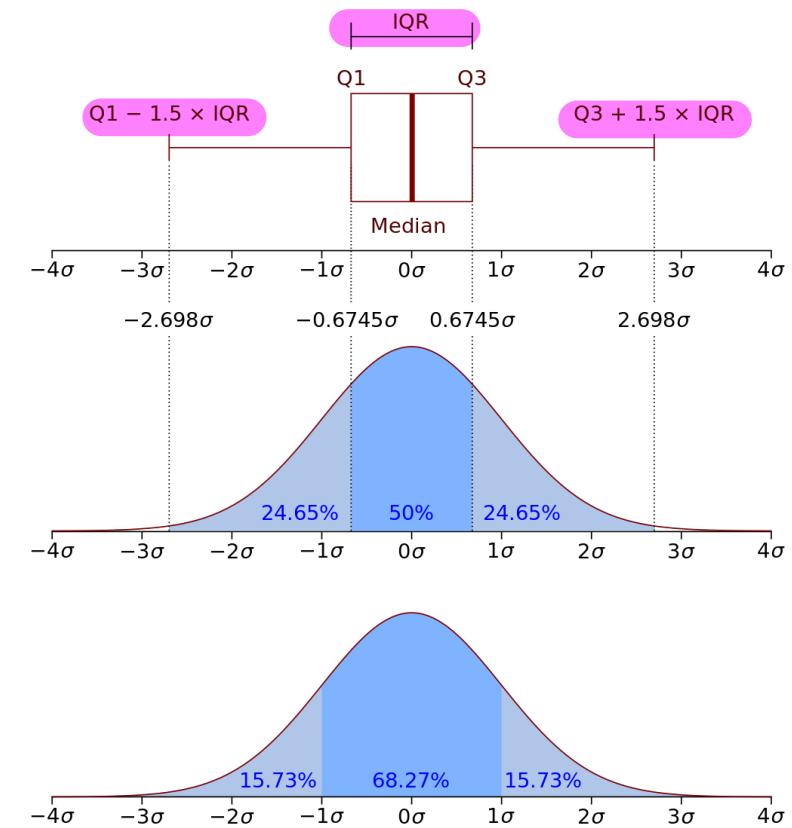
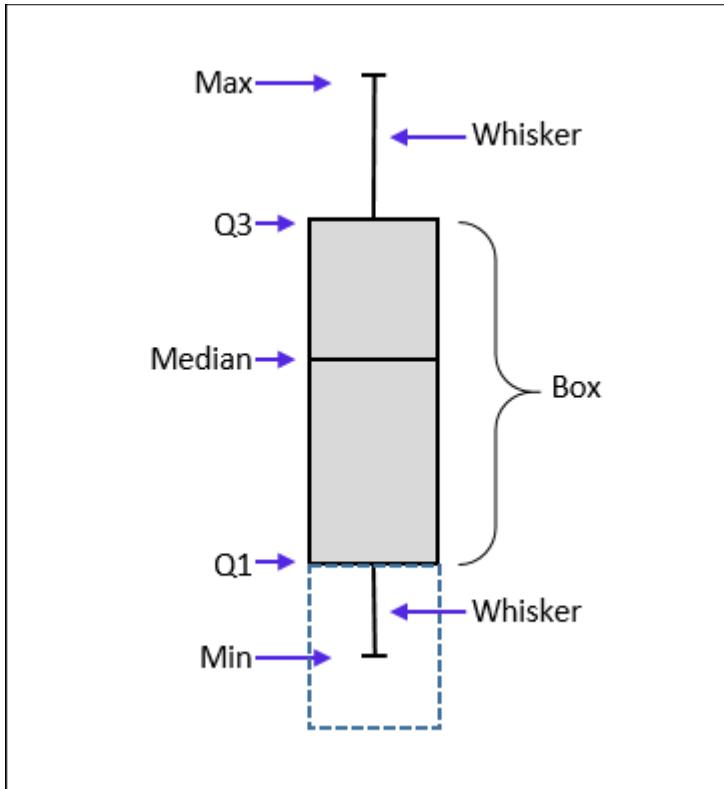
Histogram



Understanding Data: Summary Statistics

□ Data distribution

- Box plot: A visualization that summarizes the distribution of a dataset using five key summary statistics: minimum, first quartile (Q1), median (Q2), third quartile (Q3), and maximum. It helps identify outliers and the spread of the data



Understanding Data: Summary Statistics

□ Correlations between variables

- ▣ Pearson correlation coefficient: Measures the linear relationship between two continuous variables

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}}$$

- ▣ Spearman rank correlation: Measures the strength and direction of the monotonic relationship between two variables \Rightarrow categorical variables (ordinal)

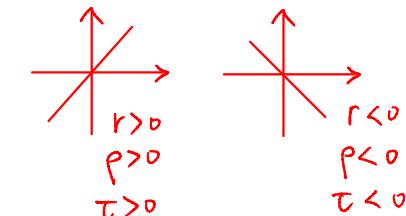
$$\rho = 1 - \frac{6 \sum_i d_i^2}{n(n^2 - 1)}$$

- d_i : the difference between the ranks of corresponding values

- ▣ Kendall's tau: Measures the ordinal association between two variables

$$\tau = \frac{C - D}{\frac{1}{2} n(n - 1)}$$

- C : the number of concordant pairs
- D : the number of discordant pairs



Understanding Data: Summary Statistics

□ Spearman rank correlation

$$\rho = 1 - \frac{6 \sum_i d_i^2}{n(n^2 - 1)}$$

ascending order

X	Y
10	40
20	30
30	50
40	20
50	60

Calculate ranks
and
differences in
ranks



X	Rank(X)	Y	Rank(Y)	d_i	d_i^2
10	1	40	3	-2	4
20	2	30	2	0	0
30	3	50	4	-1	1
40	4	20	1	3	9
50	5	60	5	0	0

Understanding Data: Summary Statistics

- Kendall's tau

$$\tau = \frac{C - D}{\frac{1}{2} n(n-1)} = 0.2$$

$$\# \text{ pairs} = 5C_2 = \frac{5 \cdot 4}{2} = 10$$

X	Y
10	40
20	30
30	50
40	20
50	60



Pairs	Relationship	Pairs	Relationship
(10,40) & (20,30)	Discordant	(20,30) & (40,20)	Discordant
(10,40) & (30,50)	Concordant	(20,30) & (50,60)	Concordant
(10,40) & (40,20)	Discordant	(30,50) & (40,20)	Discordant
(10,40) & (50,60)	Concordant	(30,50) & (50,60)	Concordant
(20,30) & (30,50)	Concordant	(40,20) & (50,60)	Concordant

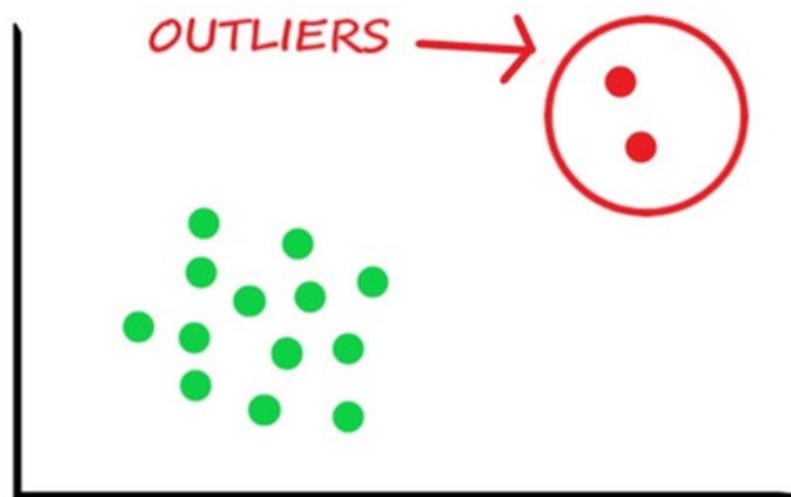
Handling Missing Values

- Methods to handle missing values
 - ▣ **Deletion:** Remove missing data points if they are few and do not significantly impact the dataset
 - ▣ Mean/Median/Mode **Imputation:** Replace missing values with statistical measures such as mean, median or mode
 - ▣ k -Nearest Neighbors (k NN) imputation: Use similar data points to estimate missing values
 - ▣ Regression imputation: Predict missing values based on relationships between variables
 - ▣ Multiple Imputation: Generate multiple datasets with estimated values and combine results

Identifying outliers

□ Outliers

- Outliers are data points that are significantly different from the rest of the data
- They can be unusually high or low compared to other observations in a dataset



Identifying outliers

- Characteristics of Outliers
 - ▣ Extreme values: Outliers fall far outside the normal range of data
 - ▣ Unexpected behavior: They may arise from errors, but they can also be valid data points
 - ▣ Impact on analysis: They can distort statistical measures such as mean, standard deviation, and regression models
- Types of outliers
 - ▣ Point outliers: A single data point that deviates significantly
 - Example: A person with an extremely high income compared to a group
 - ▣ Contextual outliers: Data points that are considered outliers only in certain contexts or subgroups
 - Example: A summer day temperature of 30°C in a region with a typical summer range of 20-25°C.
 - ▣ Collective outliers: A collection of data points that, together, show unusual behavior.
 - Example: A set of consecutive days with temperatures unusually high for a region.

Identifying outliers

- Why are outliers important?
 - ▣ Affect mean and standard deviation
 - Outliers can skew the mean and inflate the standard deviation, leading to misleading conclusions
 - ▣ Modeling issues
 - Outliers can affect the performance of machine learning models (e.g., linear regression) by making predictions less accurate
 - ▣ Data Integrity
 - Outliers may indicate data entry errors, measurement mistakes, or special cases that need further investigation

Identifying outliers

- Importance of removing outliers
 - ▣ Improved accuracy
 - Removing outliers can lead to more accurate models, ensuring that predictions are not disproportionately influenced by extreme values
 - ▣ Better data representation
 - The central tendency of the data (mean, median) becomes more representative of the general population
 - ▣ Enhanced model performance
 - Machine learning algorithms can perform better when outliers are removed, reducing noise and improving the fit of the model

Identifying outliers

assume normality

- Statistical method using Z-scores to identify outliers
 - ▣ Z-score measures how many standard deviations a data point is away from the mean. It is a standardized way of identifying outliers in normally distributed data
$$Z = \frac{X - \mu}{\sigma}$$
 - ▣ Typically, if a Z-score is greater than 3 or less than -3, the data point is considered an outlier (assuming a normal distribution)
 - This corresponds to values that are more than 3 standard deviations away from the mean

Calculate the
mean and
standard deviation

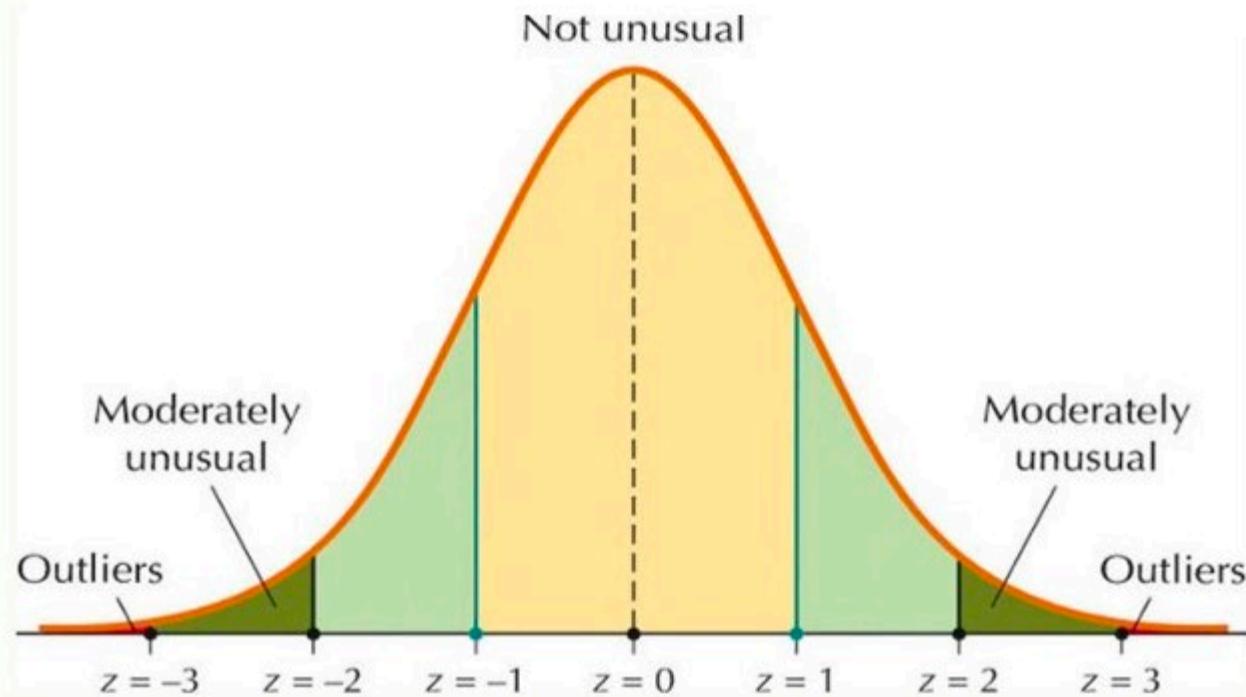
Calculate Z-scores
for each data
point

Set a threshold for
identifying outliers

Remove outliers

Identifying outliers

Detecting Outliers with z-Scores

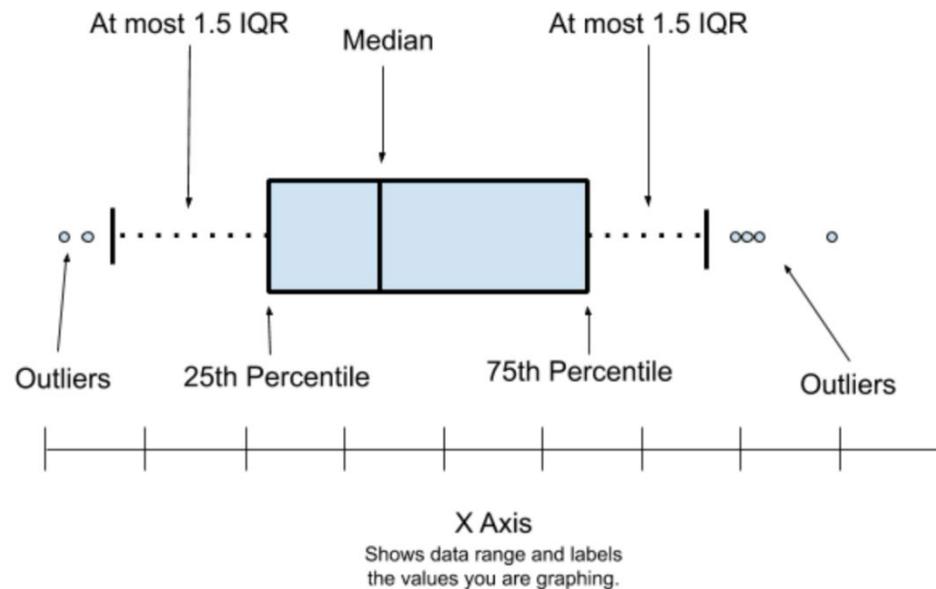


Identifying outliers

No normality assumption

- Statistical method using IQR to identify outliers
 - ▣ Calculate the lower and upper fences for outliers using IQR
 - Lower inner fence
$$\text{Lower inner fence} = Q_1 - 1.5IQR$$
 - Upper inner fence
$$\text{Upper inner fence} = Q_3 + 1.5IQR$$
 - Lower outer fence
$$\text{Lower outer fence} = Q_1 - 3IQR$$
 - Upper outer fence
$$\text{Upper outer fence} = Q_3 + 3IQR$$
 - ▣ Points beyond the inner fences in either direction are mild outliers; points beyond the outer fences in either direction are extreme outliers

Identifying outliers



Visualization: Why is Visualization Important in EDA?

1. Understanding Data Distributions

- ❑ Helps in identifying patterns, trends, and underlying structures in the data.
- ❑ Enables detection of skewness, multimodal distributions, and irregularities.

2. Identifying Outliers and Missing Data

- ❑ Visual tools like boxplots and scatter plots can highlight outliers.
- ❑ Heatmaps can reveal missing values in a dataset.

3. Discovering Relationships Between Variables

- ❑ Scatter plots and correlation heatmaps help identify strong or weak relationships.
- ❑ Pair plots allow simultaneous visualization of multiple variable interactions.

4. Enhancing Interpretability

- ❑ Complex numerical summaries can be difficult to interpret.
- ❑ Visualizations simplify insights and make data-driven decisions easier.

5. Guiding Feature Engineering

- ❑ Helps determine transformations like log-scaling, binning, or normalization.
- ❑ Aids in selecting important features for modeling.

Visualization: Types of Plots

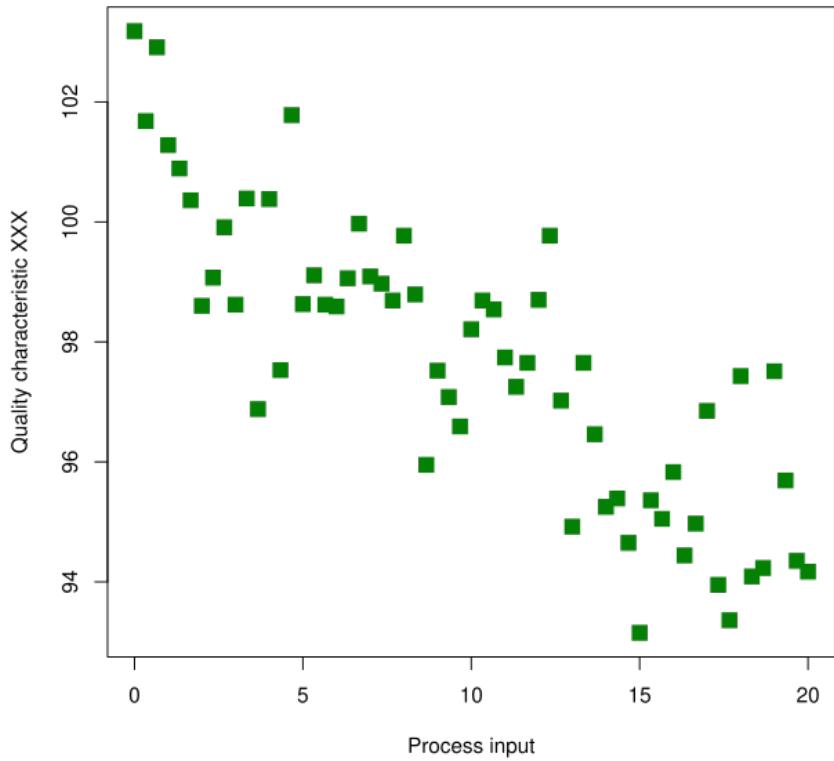
- Histograms
 - ▣ Use when: You want to analyze the distribution of a single continuous variable
 - ▣ Example: Examining the frequency of customer purchase amounts in an e-commerce dataset
 - ▣ Key insights: Shape of the distribution (normal, skewed, multimodal), outliers, and spread
- Boxplots
 - ▣ Use when: You need to compare distributions across categories or detect outliers
 - ▣ Example: Comparing salaries across different job roles in a company
 - ▣ Key insights: Median, quartiles, spread, and presence of outliers

Visualization: Types of Plots

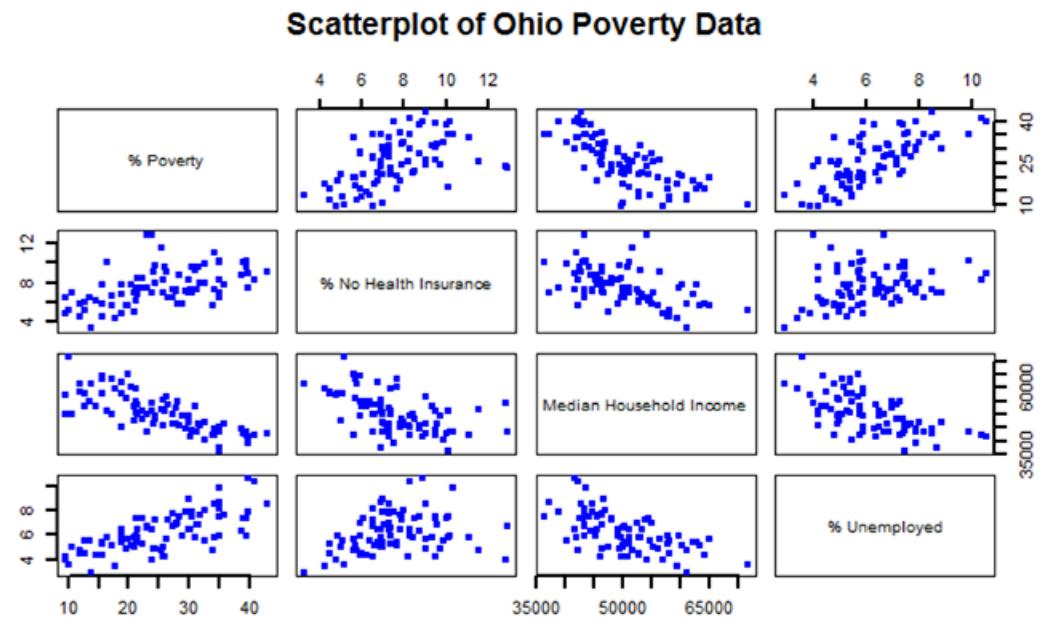
- Scatter plots
 - ▣ Use when: You want to examine the relationship between two numerical variables
 - ▣ Example: Checking if there is a correlation between advertising spend and sales revenue
 - ▣ Key insights: Strength and direction of relationships (linear, non-linear, or no correlation)
- Pair plots (Pairwise scatter plots)
 - ▣ Use when: You need to visualize relationships between multiple numerical variables in a dataset
 - ▣ Example: Analyzing relationships between weight, height, age, and cholesterol levels in a health study
 - ▣ Key insights: Trends, clusters, and correlations across multiple features

Visualization: Types of Plots

Scatterplot for quality characteristic XXX



Scatter plot



Pair plot

Feature Engineering and Transformation

□ Feature Engineering

- ❑ The process of creating new features from raw data to improve model performance
- ❑ Involves domain knowledge to extract meaningful information from raw data
- ❑ Examples: Extracting "day of the week" from a timestamp, creating interaction terms, or encoding categorical variables.

□ Feature Transformation

- ❑ Techniques used to modify features to make them more suitable for analysis and modeling
- ❑ Helps normalize, standardize, or reduce skewness in the data
- ❑ Examples: Log transformation to reduce skewness, min-max scaling, and principal component analysis (PCA) for dimensionality reduction.

Feature Engineering and Transformation

- Encoding categorical variables
 - ▣ One-Hot Encoding (OHE) *categorical variable*

- Converts categorical variables into a series of binary columns
 - Suitable for nominal (unordered) categories
 - Example: Category=[Red, Blue, Green]

Color	X_{Red}	X_{Blue}	X_{Green}
Red	1	0	0
Blue	0	1	0
Green	0	0	1

Feature Engineering and Transformation

- Encoding categorical variables
 - ▣ Label encoding
 - Assigns a unique integer to each category without considering any ranking.
 - Suitable for nominal categorical data where categories have no natural order.
 - Can sometimes be problematic if a machine learning model incorrectly interprets the numerical values as having an order.
 - Example: Category=[Red, Blue, Green]
Encoding: Red→0, Blue→1, Green→2
 - ▣ Ordinal encoding
 - Used for ordinal categorical data where categories have a clear order or ranking
 - Assigns numbers to categories based on their position in the order (e.g., "low" = 1, "medium" = 2, "high" = 3)
 - May introduce misleading information if the order between categories is not well-defined

Feature Engineering and Transformation

- Scaling and normalization
 - ▣ Min-max scaling: Rescales data to a fixed range [0,1]
- ▣ Standardization (Z-score normalization): Transforms data to have zero mean and unit variance

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- ▣ Robust scaling: Uses median and IQR to reduce the effect of outliers

$$X' = \frac{X - \text{median}(X)}{\text{IQR}(X)}$$

Feature Engineering and Transformation

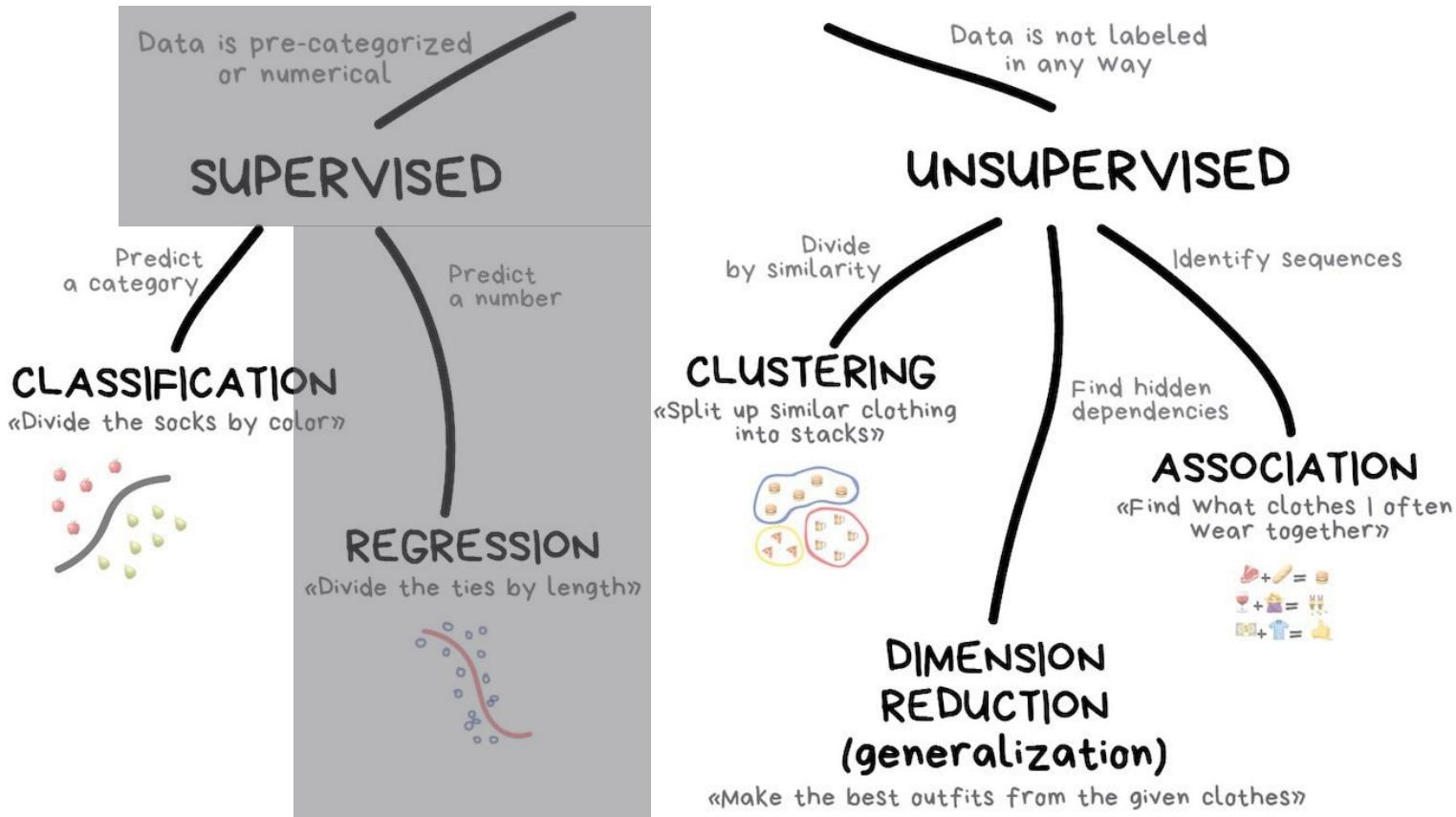
- Purposes of scaling and normalization
 - ▣ Ensuring **fair comparisons** between features
 - Different features in a dataset may have different units and scales (e.g., age in years vs. income in dollars). Scaling ensures that all features contribute equally to analysis and modeling
 - ▣ **Improving convergence** in machine learning models
 - Many machine learning algorithms (e.g., gradient descent-based models like logistic regression and neural networks) perform better when input data is normalized, leading to faster convergence and better optimization
 - ▣ Enhancing the **interpretability of data**
 - Some models (e.g., distance-based models like K-Nearest Neighbors and clustering algorithms) rely on numerical distances. Features with larger scales can dominate, leading to biased results. Scaling ensures that no single feature disproportionately influences outcomes
 - ▣ Reducing the **impact of outliers**
 - Robust scaling techniques (e.g., Robust Scaler using median and interquartile range) mitigate the influence of extreme values, improving model stability.
 - ▣ Facilitating **data visualization**
 - When visualizing multiple features together (e.g., in scatter plots or heatmaps), unscaled features can distort patterns. Scaling helps provide clearer insights.

LINEAR REGRESSION

Week04

Topics Covered in This Class

CLASSICAL MACHINE LEARNING

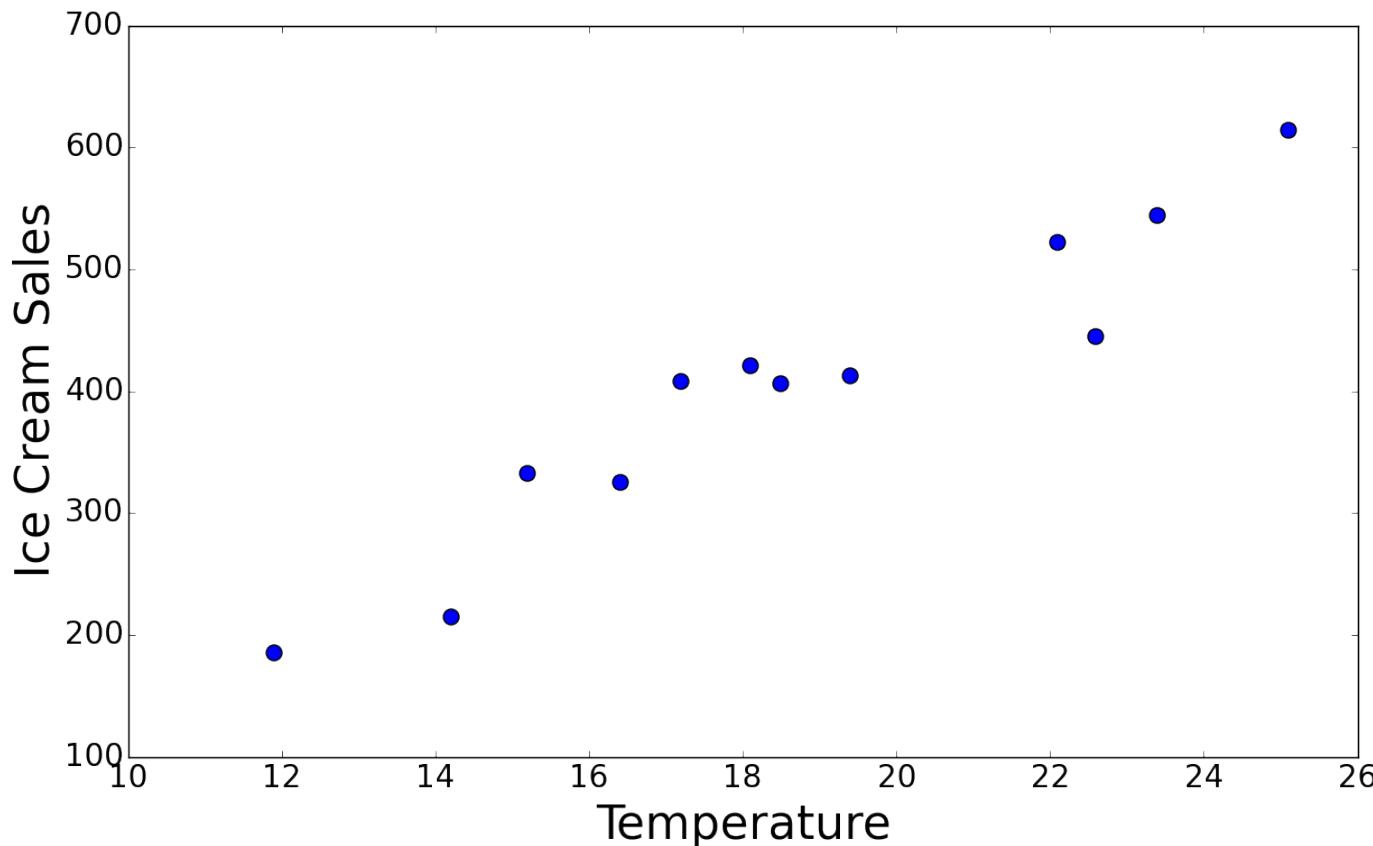


Linear Regression

Supervised Learning: Regression

- Prediction ice cream sales over given temperature

$$\text{ice cream sales} = f(\text{temperature})$$



Temperature (°C)	Ice Cream Sales (\$)
14.2	215
16.4	325
11.9	185
15.2	332
18.5	406
22.1	522
19.4	412
25.1	614
23.4	544
18.1	421
22.6	445
17.2	408
12.0	185
14.5	215
15.0	332
16.5	325
17.5	406
18.0	412
18.5	421
19.0	445
21.5	544
22.0	550
22.5	555
23.0	560
23.5	565
24.0	570
24.5	575
25.0	580
25.5	585
26.0	590

Linear Regression

- Linear regression
 - ▣ Based on the assumption that
the relationship between a scalar dependent variable y and explanatory(independent) variables X is linear
 - ▣ $X = [x_1, x_2, x_3, \dots, x_n]$
Explanatory variables: print run(x_1), page number(x_2)

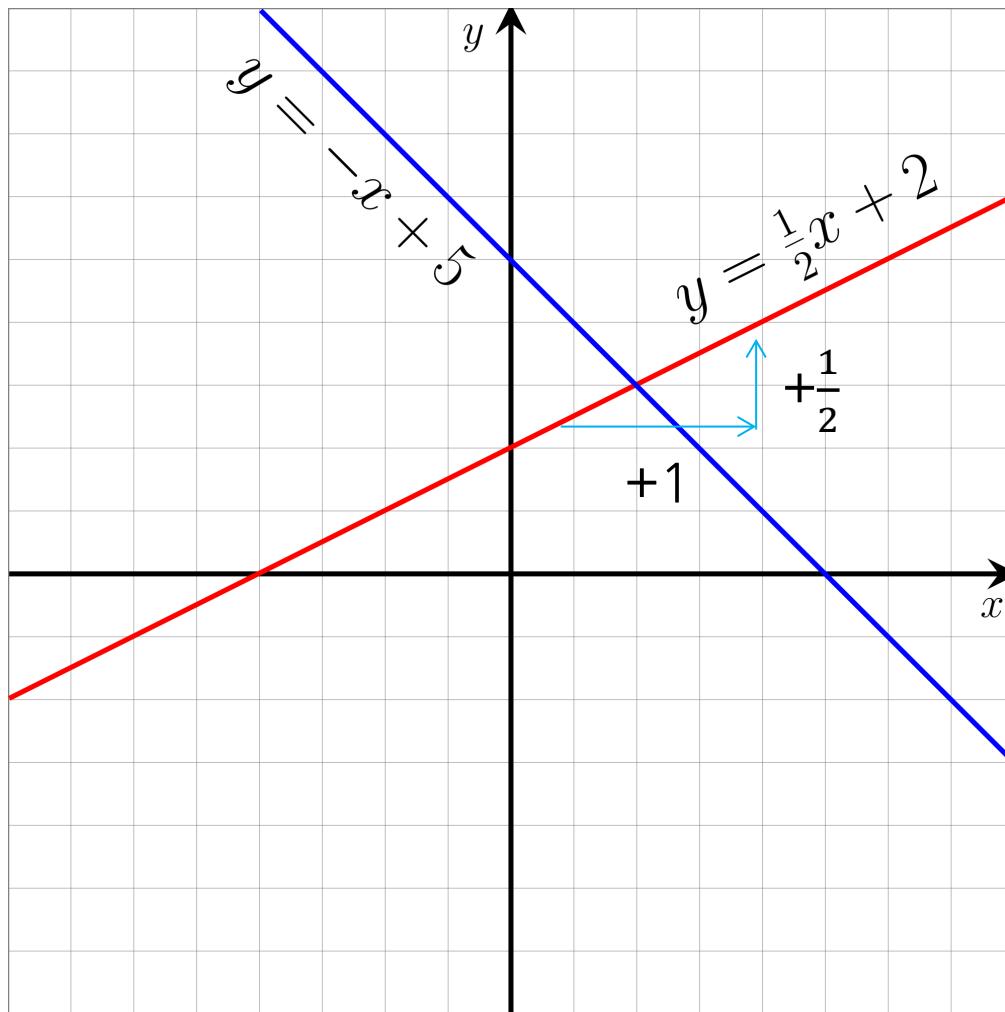
x_1	x_2
2800	22
2670	14
2800	37
2784	15
2800	38

→ X

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$
$$\epsilon \sim N(0, \sigma^2)$$

* Linear function

- You studied linear function when you are high school student!



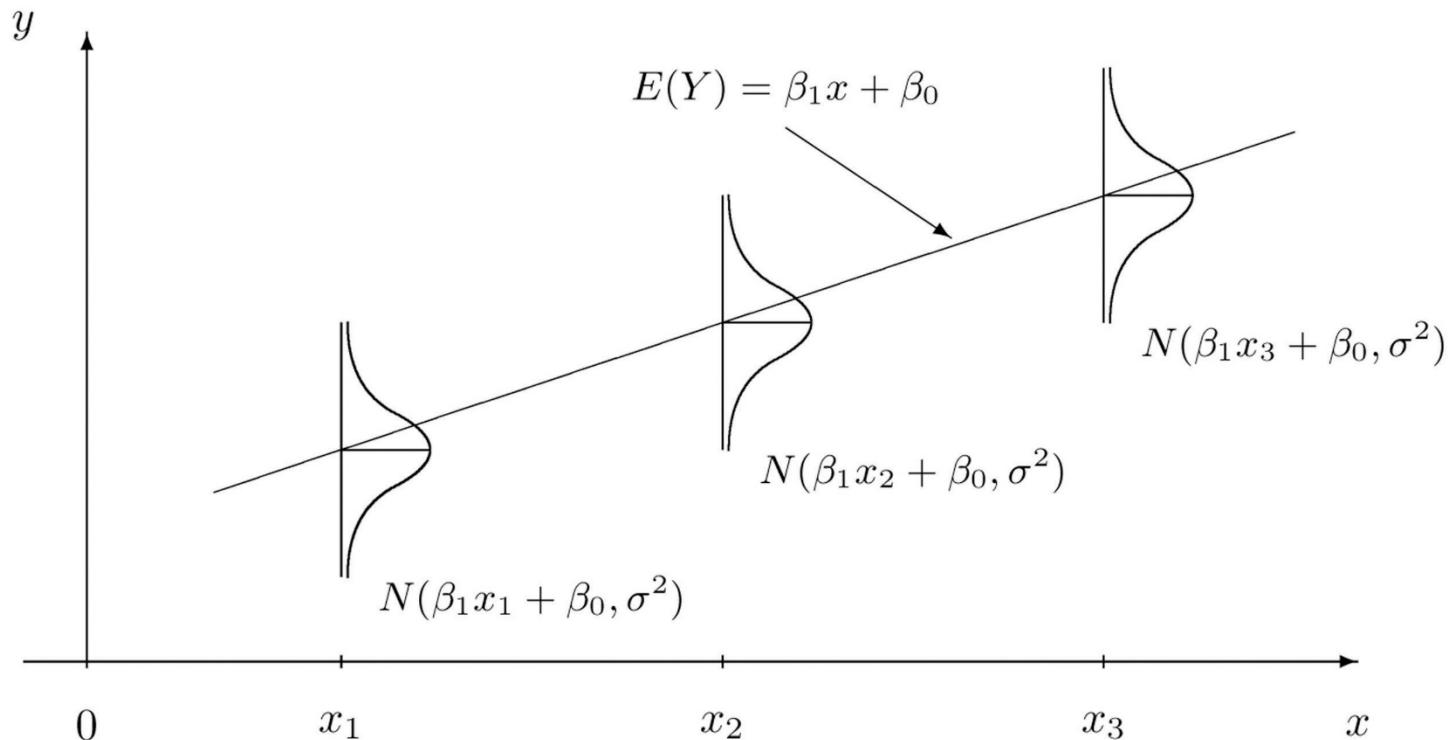
$$y = \frac{1}{2}x + 5$$

slope
→ coefficient

intercept

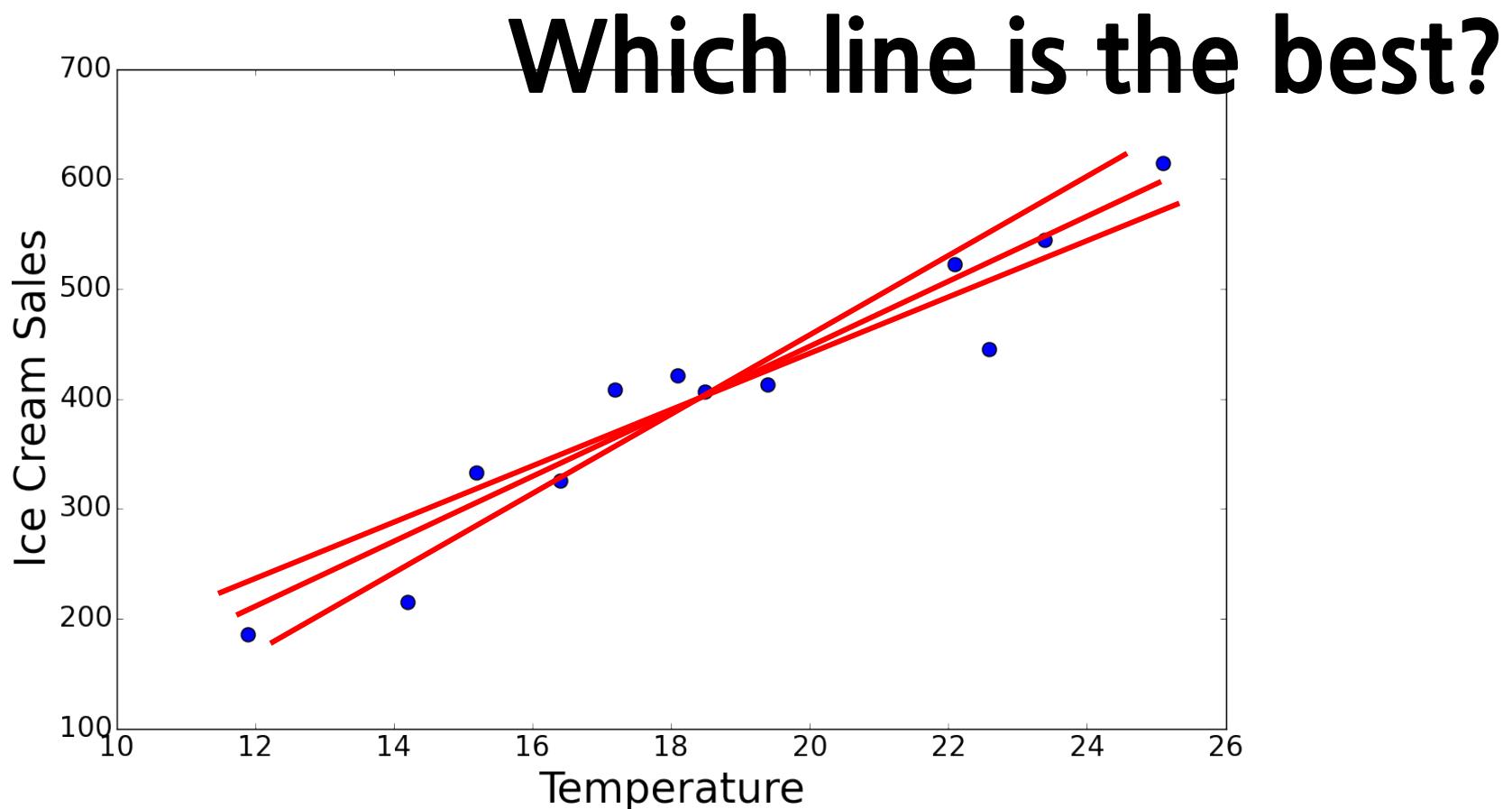
Main Assumptions of Linear Regression

- Linear regression analysis makes several key assumptions
 - ▣ Linear relationship
 - ▣ Homoscedasticity
 - ▣ Normality
 - ▣ No or little multicollinearity



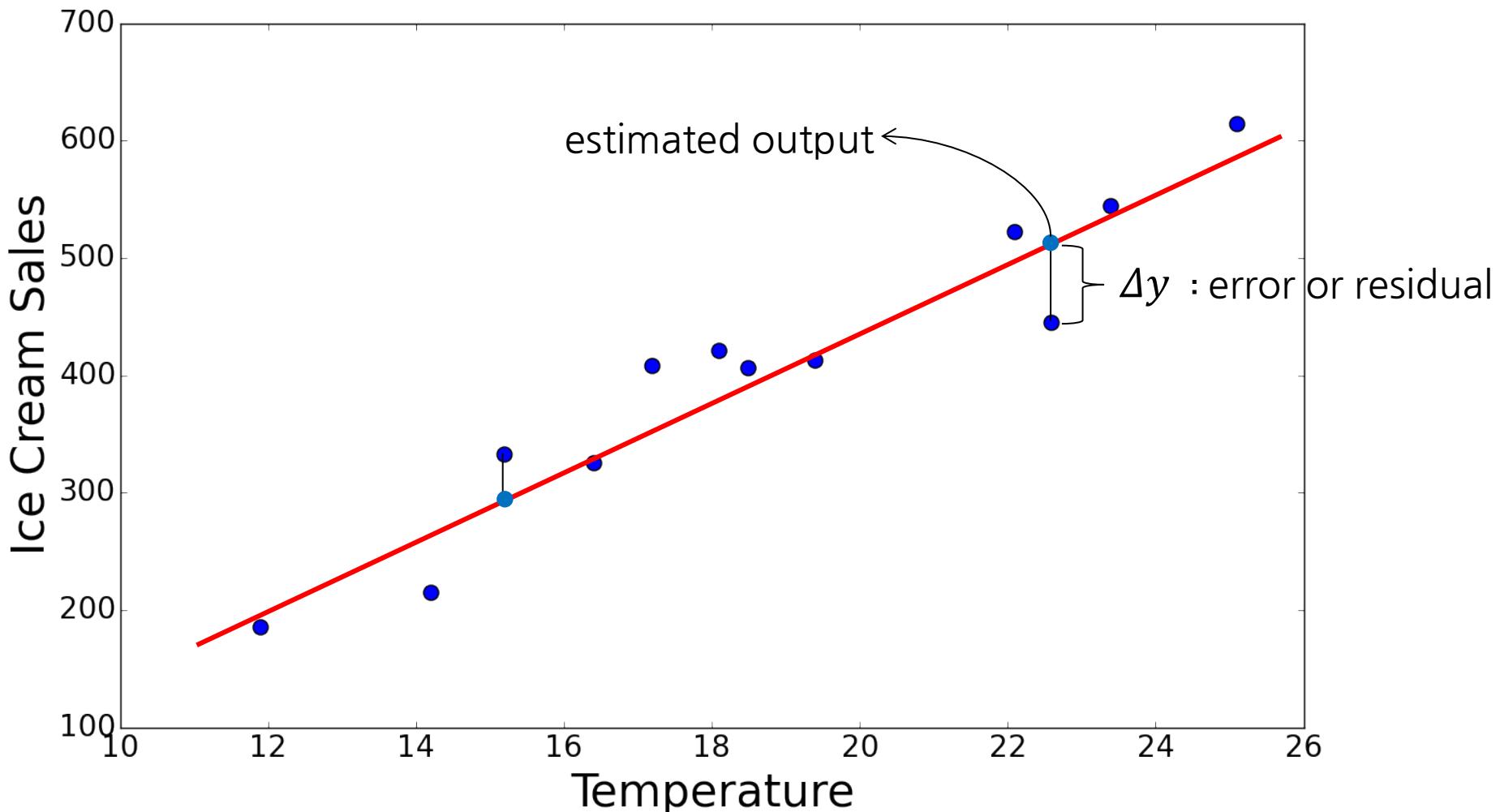
How to Determine Relationship between x and y

- Need a criterion



Least Square Method

- Minimize summation of squared error
 - ▣ Squared error=(estimated output-real output)²=error²



Least Square Method

- Minimize summation of squared error

$$\sum_i (y_i - \hat{y}_i)^2 \quad \text{GSE}$$

Sum for all data points in train set

estimated output

real output

- In the simple case: Only one independent variable
 - Estimated output $\hat{y}_i = \beta_0 + \beta_1 x_i$

$$\min \sum_i (y_i - \beta_0 - \beta_1 x_i)^2$$

Unknown values

Known values

※ Summary for Notation

- Hat, ($\hat{}$)
 - ▣ Represents estimation
 - ▣ β_1 is unknown true value, $\hat{\beta}_1$ is estimation for β_1 through model learning
 - ▣ y_i is known output value of i -th sample, \hat{y}_i is estimated output by learned model
- Bar, ($\bar{}$)
 - ▣ Represents sample mean
 - ▣ Arithmetic average of the observed values of variable

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

- ▣ If the number of input variables is more than one, elements of sample mean vector consist of average of each variable

$$\bar{\mathbf{x}} = \left(\frac{\sum_{i=1}^n x_{1i}}{n}, \frac{\sum_{i=1}^n x_{2i}}{n}, \dots, \frac{\sum_{i=1}^n x_{pi}}{n} \right) = \frac{\sum_{i=1}^n \mathbf{x}_i}{n}$$

- Bold character usually represents vector

Least Square Method

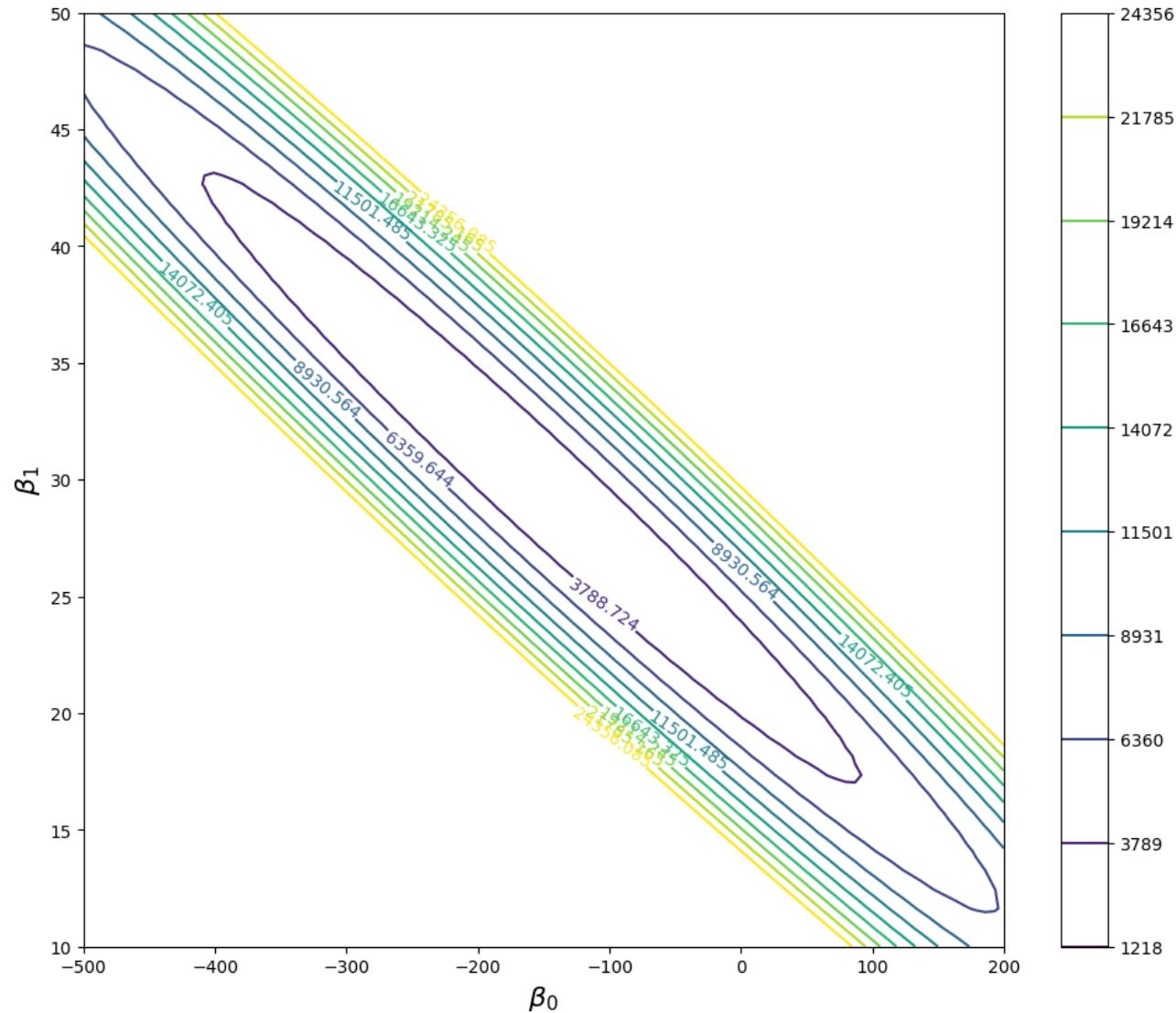
- Which one is better?

Temperature (°C)	Ice Cream Sales (\$)	$\beta_1 = 20, \beta_0 = -80$		$\beta_1 = 30, \beta_0 = -160$	
		Estimated Sales	Squared error	Estimated Sales	Squared error
14.2	y_i	215	\hat{y}_i	204	$(y_i - \hat{y}_i)^2$
16.4		325		248	5929
11.9		185		158	729
15.2		332		224	11664
18.5		406		290	13456
22.1		522		362	25600
19.4		412		308	10816
25.1		614		422	36864
23.4		544		388	24336
18.1		421		282	19321
22.6		445		372	5329
17.2		408		264	20736
sum				174901	14594

better

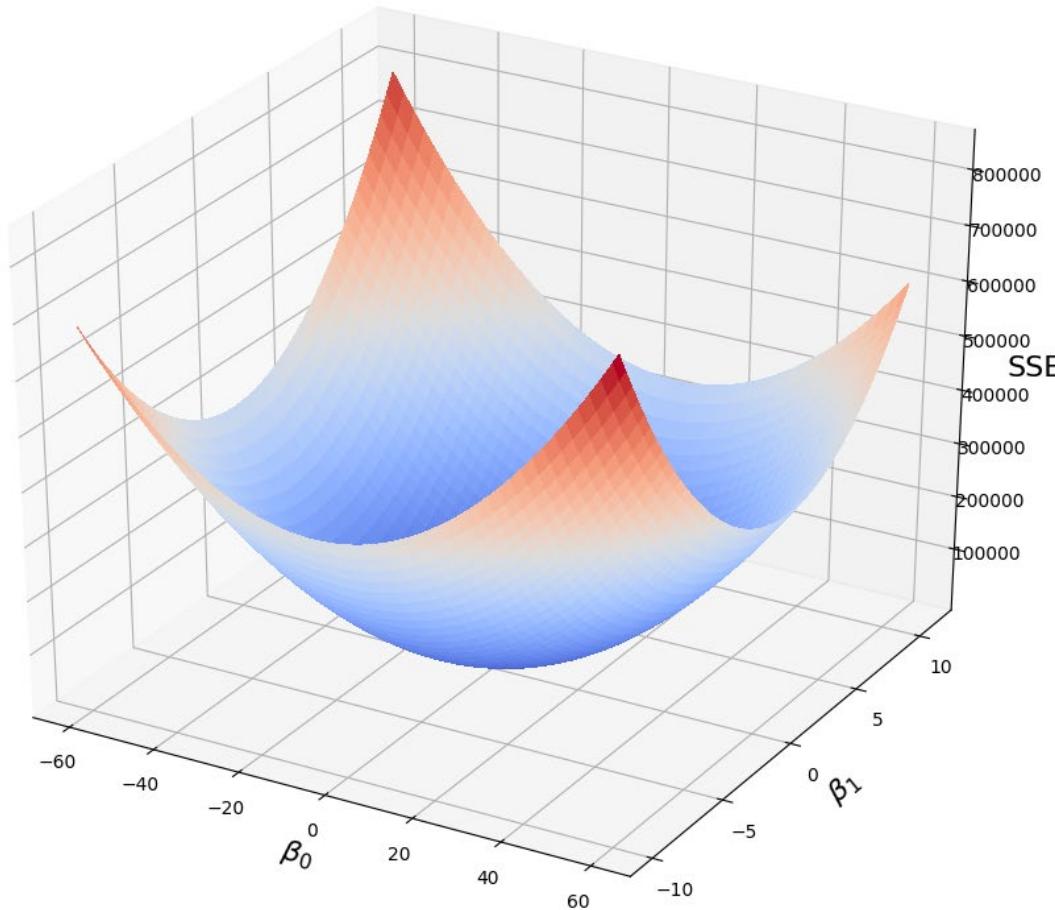
Least Square Method

- Summation of squared error with different β_0 and β_1



Least Square Method

- Summation of squared error with different β_0 and β_1 for simulated data from $y = x + 1$



Optimization for Linear Regression

- Variables to be determined

$$\beta_0, \beta_1$$

- Objective function

$$\min f(\beta_0, \beta_1) = \min \sum_i (y_i - \beta_0 - \beta_1 x_i)^2$$

- Constraints

- No constraint

Optimization for Linear Regression

- Solution

- ▣ Calculate partial derivatives with respect to β_0, β_1

$$\frac{\partial f(\beta_0, \beta_1)}{\partial \beta_0} = \sum_{i=1}^n -2(y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\frac{\partial f(\beta_0, \beta_1)}{\partial \beta_1} = \sum_{i=1}^n -2x_i(y_i - \beta_0 - \beta_1 x_i) = 0$$

- ▣ Solve linear equations

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Multiple Input Variables

- More than one input variable
 - Want to predict consumption of petrol

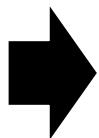
Petrol Tax(\$)	Average Income (\$)	Paved Highways (miles)	Proportion of population with driver's license	Consumption of petrol (M of gallons)
9	3571	1976	0.525	541
9	4092	1250	0.572	524
9	3865	1586	0.58	561
7.5	4870	2351	0.529	414
8	4399	431	0.544	410
10	5342	1333	0.571	457
8	5319	11868	0.451	344
8	5126	2138	0.553	467
8	4447	8577	0.529	464
7	4512	8507	0.552	498
...

Multiple Input Variables

- Estimation based on petrol tax, average income, length of paved highways, proportion of population with driver's license

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \epsilon$$

- y = consumption of petrol
- x_1 = petrol tax
- x_2 = average income
- x_3 = length of paved highways
- x_4 = proportion of population with driver's license
- ϵ is random error which follows Gaussian distribution with 0 mean, σ^2 variance



$$\min \sum_i (y_i - \hat{y}_i)^2$$

Same as the simple case!

Optimization for Linear Regression: Multivariate

- multivariate linear regression

$$\min f(\beta_0, \dots, \beta_p) = \min \sum_i (y_i - \beta_0 - \beta_1 x_{1i} - \dots - \beta_p x_{pi})^2$$

- Estimated parameters are obtained by setting partial derivatives zero

$$\left. \begin{aligned} \frac{\partial f(\beta_0, \dots, \beta_p)}{\partial \beta_0} &= \sum_{i=1}^n -2(y_i - \beta_0 - \beta_1 x_{1i} - \dots - \beta_p x_{pi}) = 0 \\ \frac{\partial f(\beta_0, \dots, \beta_p)}{\partial \beta_1} &= \sum_{i=1}^n -2x_{1i}(y_i - \beta_0 - \beta_1 x_{1i} - \dots - \beta_p x_{pi}) = 0 \\ &\vdots \\ \frac{\partial f(\beta_0, \dots, \beta_p)}{\partial \beta_p} &= \sum_{i=1}^n -2x_{pi}(y_i - \beta_0 - \beta_1 x_{1i} - \dots - \beta_p x_{pi}) = 0 \end{aligned} \right\}$$

Multiple Input Variables

- Matrix approach to multiple regression model

$$y_1 = \beta_0 \cdot 1 + \beta_1 x_{11} + \beta_2 x_{21} + \cdots + \beta_p x_{k1} \quad n \text{ samples, } p \text{ input variables}$$

$$\begin{aligned} y_2 &= \beta_0 \cdot 1 + \beta_1 x_{12} + \beta_2 x_{22} + \cdots + \beta_p x_{k2} \\ &\vdots \end{aligned}$$

$$y_n = \beta_0 \cdot 1 + \beta_1 x_{1n} + \beta_2 x_{2n} + \cdots + \beta_p x_{kn}$$



$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{p1} \\ 1 & x_{12} & x_{22} & \cdots & x_{p2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{pn} \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{p1} \\ 1 & x_{12} & x_{22} & \cdots & x_{p2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{pn} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$



$$= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

$$\boldsymbol{\epsilon} \sim MVN(0, \sigma^2 \mathbf{I})$$



$$E = \|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2$$

Optimization for Linear Regression: Multivariate

$$\min E = \min \|y - X\beta\|^2$$

- Solution is obtained by setting $\frac{\partial E}{\partial \beta} = 0$

$$\frac{\partial (\mathbf{x} - A\mathbf{s})^T W (\mathbf{x} - A\mathbf{s})}{\partial \mathbf{s}} = -2A^T W (\mathbf{x} - A\mathbf{s})$$

$X=Y$ $S=\beta$
 $A=X$ $W=I$



$$\frac{\partial (\mathbf{y} - X\beta)^T (\mathbf{y} - X\beta)}{\partial \beta} = -2X^T (\mathbf{y} - X\beta) = \mathbf{0}$$



$$\underline{X^T X \beta - X^T y = 0} \quad \begin{aligned} X &= n \times (p+1) \\ X^T &= (p+1) \times n \end{aligned}$$

- Reference

- Matrix Cookbook

- <https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf>

Estimation of Regression Coefficients: Multivariate

- Use least square methods as same as simple linear regression

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- \mathbf{X}^T : transpose matrix of \mathbf{X}
- \mathbf{X}^{-1} : inverse matrix of \mathbf{X}

$$\hat{\mathbf{y}} = \mathbf{X} \hat{\beta} = \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \mathbf{H} \mathbf{y}$$

- Residual(error) terms

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{H}) \mathbf{y} = [e_1 \ e_2 \ \cdots \ e_n]^T$$

- Covariance

$$\begin{aligned} \text{Cov}[\mathbf{e}] &= \text{Cov}[\mathbf{y} - \hat{\mathbf{y}}] = \text{Cov}[(\mathbf{I} - \mathbf{H}) \mathbf{y}] = \text{Cov}[(\mathbf{I} - \mathbf{H})(\mathbf{X} \beta + \epsilon)] \\ &= \text{Cov}[(\mathbf{I} - \mathbf{H}) \epsilon] = (\mathbf{I} - \mathbf{H}) \text{Cov}[\epsilon] (\mathbf{I} - \mathbf{H})^T = \sigma^2 (\mathbf{I} - \mathbf{H}) (\mathbf{I} - \mathbf{H})^T = \sigma^2 (\mathbf{I} - \mathbf{H}) \end{aligned}$$

- SSE

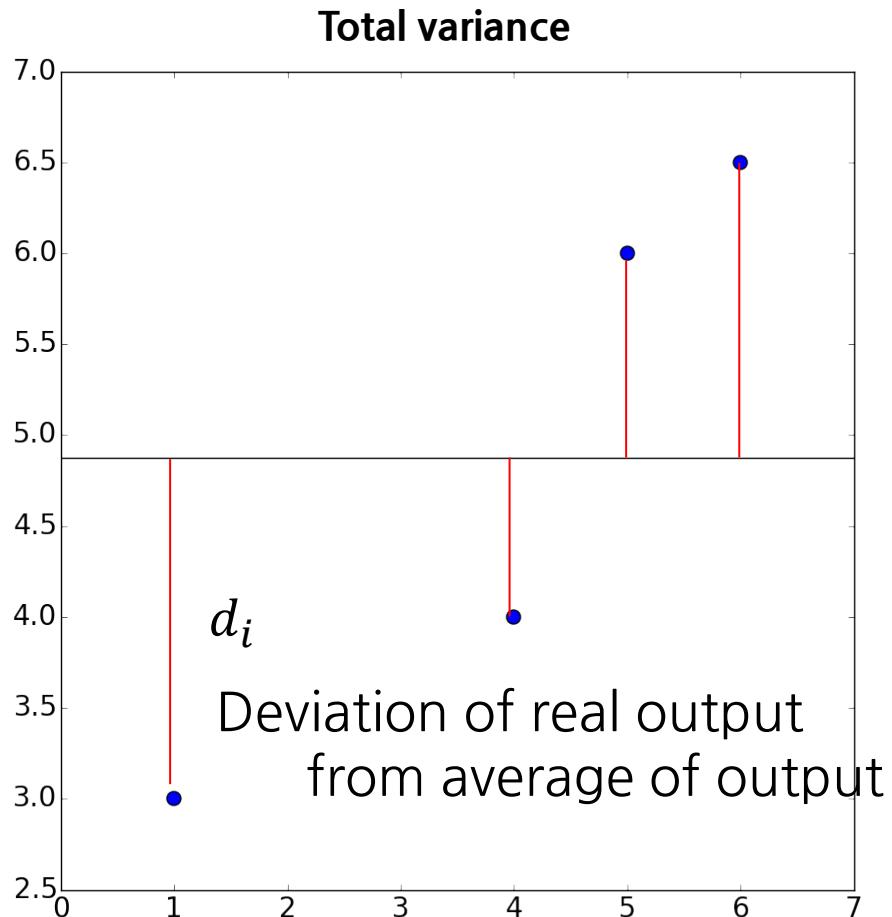
$$SSE = \sum_i (y_i - \hat{y}_i)^2 = \sum_i e_i^2 = \mathbf{e}^T \mathbf{e}$$

Is the Regression Model Significant?

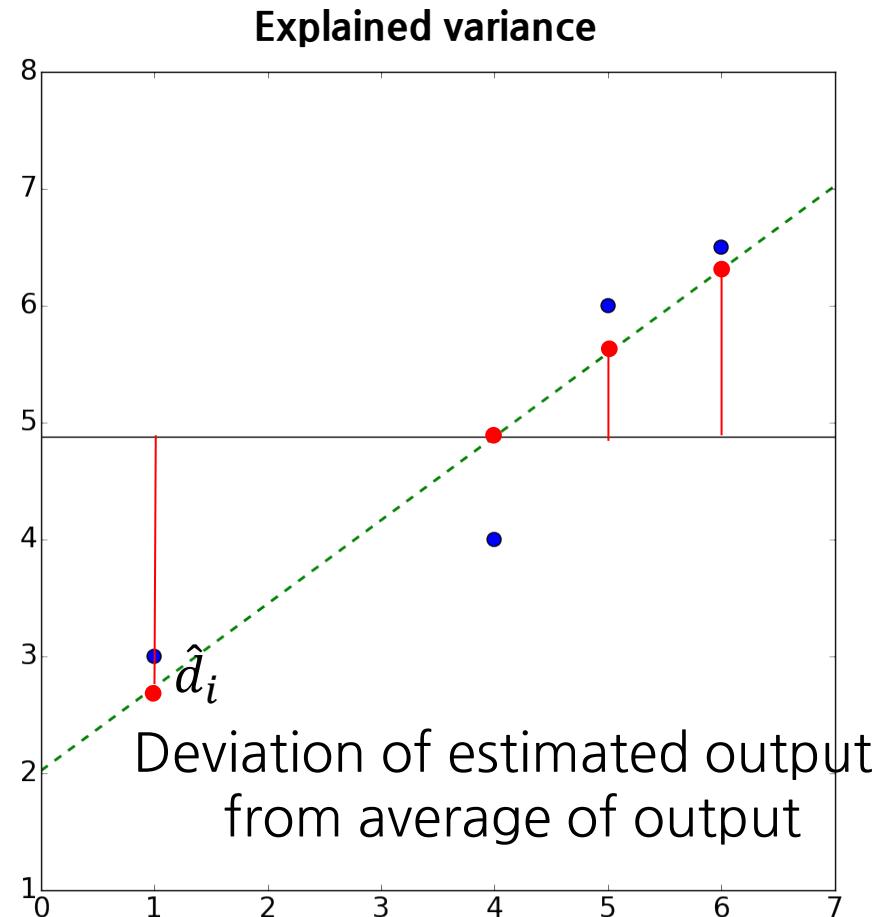
- Modeling learning is not the end of the analysis
 - ▣ Check overall significance in regression models
 - Whether the regression model is overall significant for predicting a target
 - ▣ Check significance of regression coefficients
 - Whether the specific variable is significant for predicting a target
- In the case of simple linear regression, testing overall significance of the model is the same as testing significance of regression coefficients
 - ▣ Because only one explanatory variable is used

Sum of Square

- Explained variance(SSR) and Total variance(SST)



$$\sum_i d_i^2$$



$$\sum_i \hat{d}_i^2$$

Sum of Square

Chi-squared distribution

- Total variance: the total sum of squares

$$SST = \sum_i (y_i - \bar{y})^2$$

- Explained variance: the regression sum of squares, also called the explained sum of squares

$$SSR = \sum_i (\hat{y}_i - \bar{y})^2$$

- Residual variance: the sum of squares of residuals, also called the residual sum of squares

$$SSE = \sum_i (y_i - \hat{y}_i)^2$$

- Relationship among three values

$$SST = SSR + SSE$$

Test of Model Significance

전체 회귀 모형 유의성 검증 or 분산 차이

- F -test for general regression models

- ▣ Hypothesis

$$H_0: \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

$$H_1: \text{not all } \beta_i (i = 1, 2, \dots, p) \text{ equal zero}$$

- ▣ Test statistic

$$\frac{(MSR_p)}{(MSE_{n-p-1})}$$

$$F^* = MSR/MSE$$

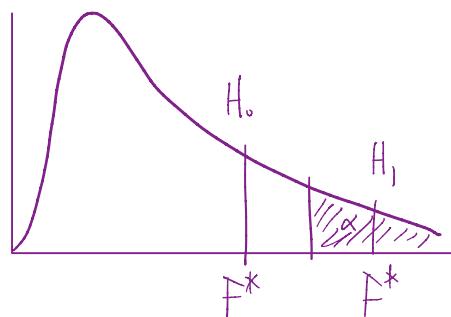
- F follows F -distribution with $(p, n - p - 1)$ degree of freedom

- ▣ Decision rule

If $F^* \leq F(1 - \alpha; p, n - p - 1)$, conclude H_0

If $F^* > F(1 - \alpha; p, n - p - 1)$, conclude H_1

- α : significance level



⌘ Statistical Test

- A statistical test provides a mechanism for making quantitative decisions about a process or processes
 - ▣ The intent is to determine whether there is enough evidence to "reject" a conjecture or hypothesis about the process
 - ▣ The procedure is based on how likely it would be for a set of observations to occur if the null hypothesis were true
- Null hypothesis
 - ▣ A general statement or default position that there is no relationship between two measured phenomena, or no association among groups
- Alternative hypothesis
 - ▣ It is the hypothesis used in hypothesis testing that is contrary to the null hypothesis

⌘ Statistical Test

- Steps in testing for statistical significance

State the research hypothesis



State the null hypothesis & alternative hypothesis



Select a probability of error level (alpha level)



Select and compute the test for statistical significance



Interpret the results

※ Statistical Test

두 집단의 평균이 유의하게 다른가?,

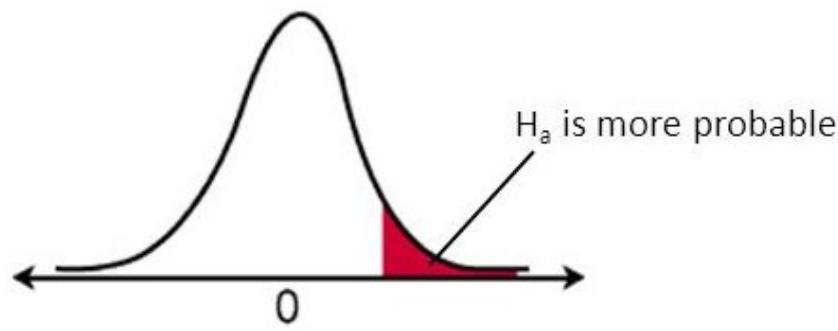
- Consider 20 first year resident female doctors drawn at random from one area
 - ▣ resting systolic blood pressures measured using an electronic sphygmomanometer
 - Sample mean = 130.05
 - ▣ Research hypothesis is that a resting systolic blood pressure of 120 mm Hg is predicted as the population mean
 - ▣ Null Hypothesis

$$H_0: \mu = 120$$

$$H_1: \mu \neq 120$$

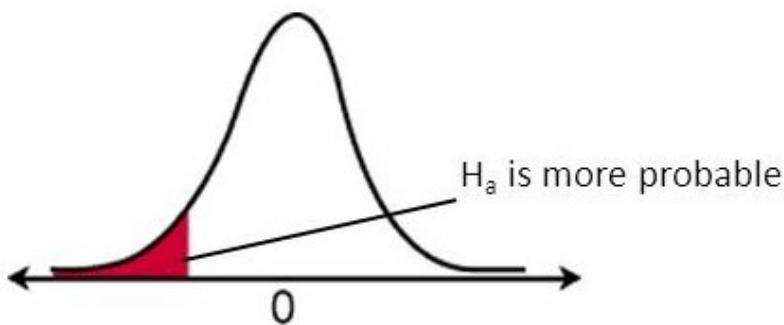
- ▣ Set significance level as 0.05
- ▣ Determine test statistics and underlying distribution
 - $t = \frac{\bar{x} - \mu}{\sqrt{s^2/n}}$
 - t follows t -distribution with the degree of freedom as $n - 1$

* Statistical Test



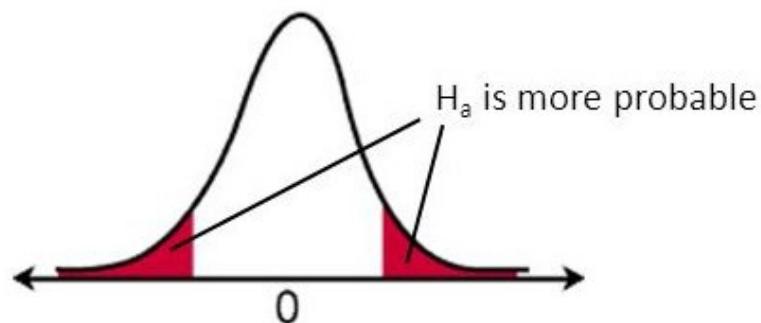
Right-tail test

$$H_a: \mu > \text{value}$$



Left-tail test

$$H_a: \mu < \text{value}$$



Two-tail test

$$H_a: \mu \neq \text{value}$$

Test of Model Significance

- ANOVA table for multiple regression model with p input variables

Factor	Sum of square	Degree of freedom	Mean square	F-value	p-value
Model	SSR	p	$MSR = SSR/p$	$F_0 = MSR/MSE$	$P\{F_{p,n-p-1} > F_0\}$
Residual	SSE	$n - p - 1$	$MSE = SSE/(n - p - 1)$		
Total	SST	$n - 1$			

- Analysis of Variance (ANOVA)

Degree of Freedom

- The number of degrees of freedom is the number of values in the final calculation of a statistic that are free to vary
 - ▣ The number of independent ways by which a dynamic system can move, without violating any constraint imposed on it
- Sample variance

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$$

- ▣ The reason that denominator is $n - 1$ is that degree of freedom of sample mean, \bar{y} is $n - 1$
- ▣ Another reason is that in the case of that denominator is $n - 1$, S^2 is unbiased estimator of variance of population
- Mean squared error for simple linear regression

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

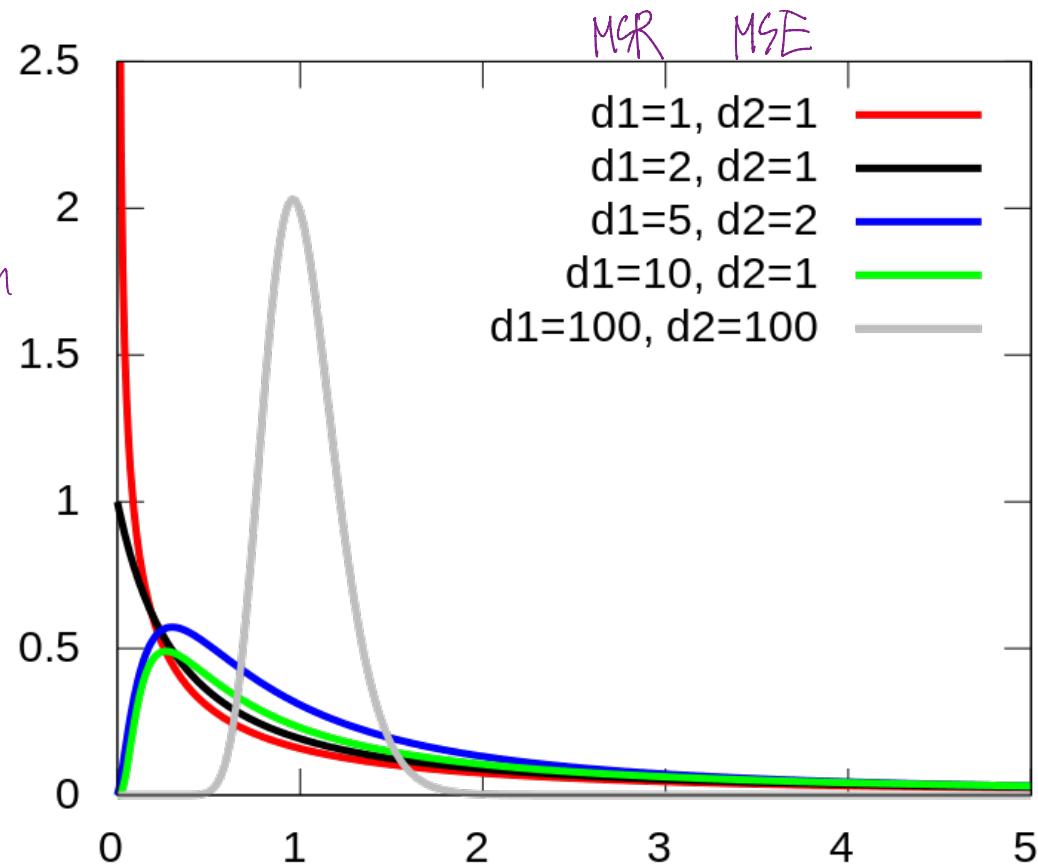
$P=1$

- ▣ The reason that denominator is $n - 2$ is that \hat{y}_i is calculate from $\hat{\beta}_0 + \hat{\beta}_1 x_i$ and it depends on two estimators $\hat{\beta}_0, \hat{\beta}_1 \rightarrow$ Decrease two degrees of freedom

* F distribution

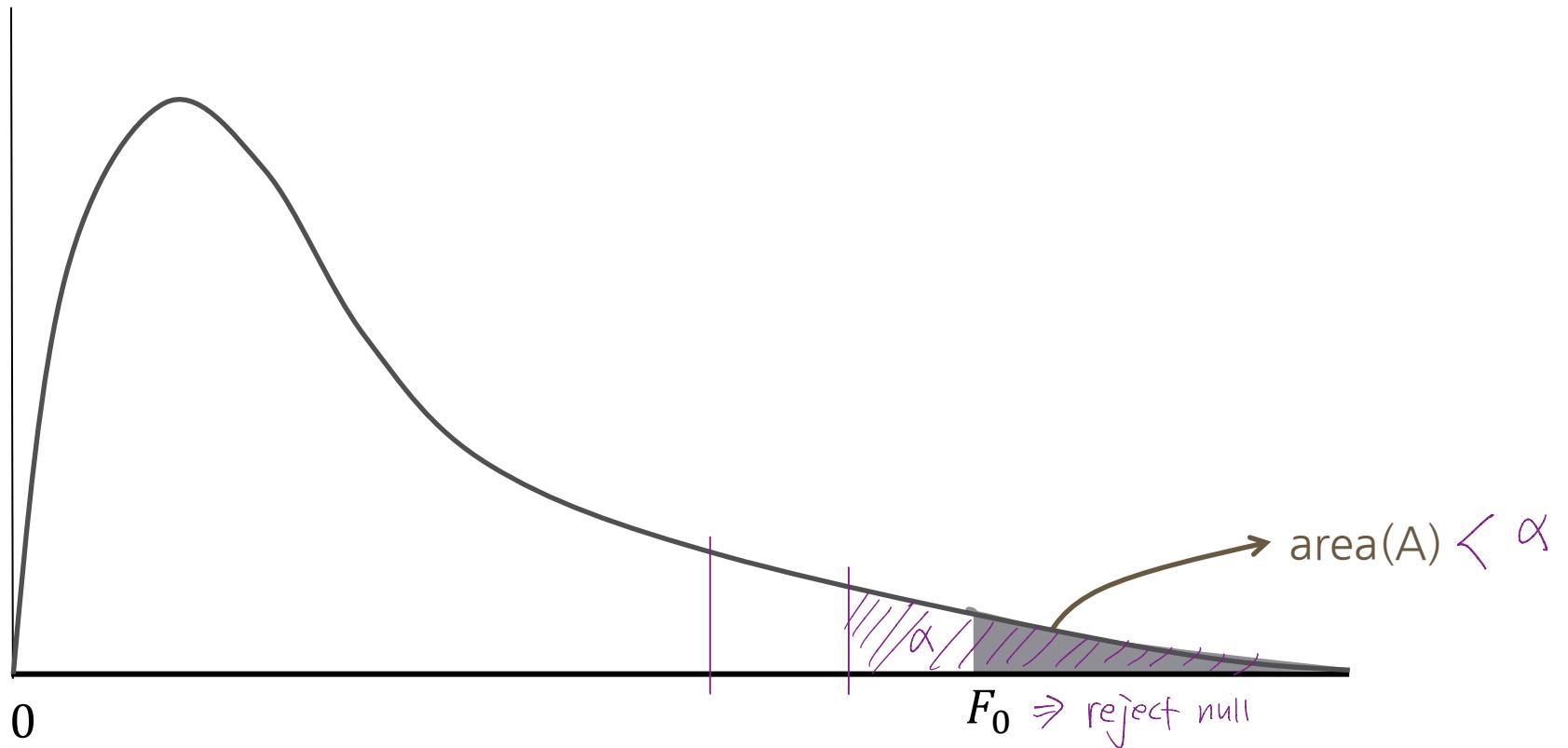
- F statistics follows F distribution with $(p, n - p - 1)$ degree of freedom
 - Probability density function of F distribution with different parameters
 - F distribution is determined by two parameters

df 4
Normal Distribution



Test of Model Significance $\Rightarrow F$

- If (area under density function from F_0 to ∞) $< \alpha$
→ Reject null hypothesis → not all $\beta_i (i = 1, 2, \dots, p)$ equal zero
 - ▣ α is significance level
 - ▣ significance level is usually set to 0.1, 0.05, or 0.01
 - The higher significance level, the higher probability to reject null hypothesis



※ Table for Distributions

F-Distribution, Continued
Upper 0.01 Critical Points

		$F_{0.01}(r_1, r_2)$								
		r_1								
r_2		10	15	20	25	30	40	60	120	∞
1	6055.9	6157.3	6208.7	6239.8	6260.7	6286.8	6313.0	6339.4	6365.9	
2	99.40	99.43	99.45	99.46	99.47	99.47	99.48	99.49	99.50	
3	27.23	26.87	26.69	26.58	26.50	26.41	26.32	26.22	26.13	
4	14.55	14.20	14.02	13.91	13.84	13.75	13.65	13.56	13.46	
5	10.05	9.72	9.55	9.45	9.38	9.29	9.20	9.11	9.02	
6	7.87	7.56	7.40	7.30	7.23	7.14	7.06	6.97	6.88	
7	6.62	6.31	6.16	6.06	5.99	5.91	5.82	5.74	5.65	
8	5.81	5.52	5.36	5.26	5.20	5.12	5.03	4.95	4.86	
9	5.26	4.96	4.81	4.71	4.65	4.57	4.48	4.40	4.31	
10	4.85	4.56	4.41	4.31	4.25	4.17	4.08	4.00	3.91	
11	4.54	4.25	4.10	4.01	3.94	3.86	3.78	3.69	3.60	
12	4.30	4.01	3.86	3.76	3.70	3.62	3.54	3.45	3.36	
13	4.10	3.82	3.66	3.57	3.51	3.43	3.34	3.25	3.17	
14	3.94	3.66	3.51	3.41	3.35	3.27	3.18	3.09	3.00	
15	3.80	3.52	3.37	3.28	3.21	3.13	3.05	2.96	2.87	
16	3.69	3.41	3.26	3.16	3.10	3.02	2.93	2.84	2.75	
17	3.59	3.31	3.16	3.07	3.00	2.92	2.83	2.75	2.65	
18	3.51	3.23	3.08	2.98	2.92	2.84	2.75	2.66	2.57	
19	3.43	3.15	3.00	2.91	2.84	2.76	2.67	2.58	2.49	
20	3.37	3.09	2.94	2.84	2.78	2.69	2.61	2.52	2.42	
21	3.31	3.03	2.88	2.79	2.72	2.64	2.55	2.46	2.36	
22	3.26	2.98	2.83	2.73	2.67	2.58	2.50	2.40	2.31	
23	3.21	2.93	2.78	2.69	2.62	2.54	2.45	2.35	2.26	
24	3.17	2.89	2.74	2.64	2.58	2.49	2.40	2.31	2.21	
25	3.13	2.85	2.70	2.60	2.54	2.45	2.36	2.27	2.17	
26	3.09	2.81	2.66	2.57	2.50	2.42	2.33	2.23	2.13	
27	3.06	2.78	2.63	2.54	2.47	2.38	2.29	2.20	2.10	
28	3.03	2.75	2.60	2.51	2.44	2.35	2.26	2.17	2.06	
29	3.00	2.73	2.57	2.48	2.41	2.33	2.23	2.14	2.03	
30	2.98	2.70	2.55	2.45	2.39	2.30	2.21	2.11	2.01	
40	2.80	2.52	2.37	2.27	2.20	2.11	2.02	1.92	1.80	
60	2.63	2.35	2.20	2.10	2.03	1.94	1.84	1.73	1.60	
120	2.47	2.19	2.03	1.93	1.86	1.76	1.66	1.53	1.38	
∞	2.32	2.04	1.88	1.77	1.70	1.59	1.47	1.32	1.00	

LINEAR REGRESSION

Week05

Linear Regression

Is the Regression Model Significant?

- Modeling learning is not the end of the analysis
 - ▣ Check overall significance in regression models
 - Whether the regression model is overall significant for predicting a target
 - ▣ Check significance of regression coefficients
 - Whether the specific variable is significant for predicting a target
- In the case of simple linear regression, testing overall significance of the model is the same as testing significance of regression coefficients
 - ▣ Because only one explanatory variable is used

F-test		모든 자료가 예측에 유의한가
T-test		개별 변수가 예측에 유의한가

Test Concerning Regression Coefficients

- Test for $\beta_j (j = 0, 1, 2, \dots, p)$

- Hypothesis

$$H_0: \beta_j = 0$$

$$H_1: \beta_j \neq 0$$

- Test statistic

$$t_j = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)}$$

- $se^2(\hat{\beta}) = MSE(\mathbf{X}^T \mathbf{X})^{-1} \rightarrow se^2(\hat{\beta}_j) = [MSE(\mathbf{X}^T \mathbf{X})^{-1}]_{j,j}$

- Decision rule

If $|t_j| \leq t\left(1 - \frac{\alpha}{2}; n - p - 1\right)$, conclude H_0

If $|t_j| > t\left(1 - \frac{\alpha}{2}; n - p - 1\right)$, conclude H_1

Test Concerning Regression Coefficients

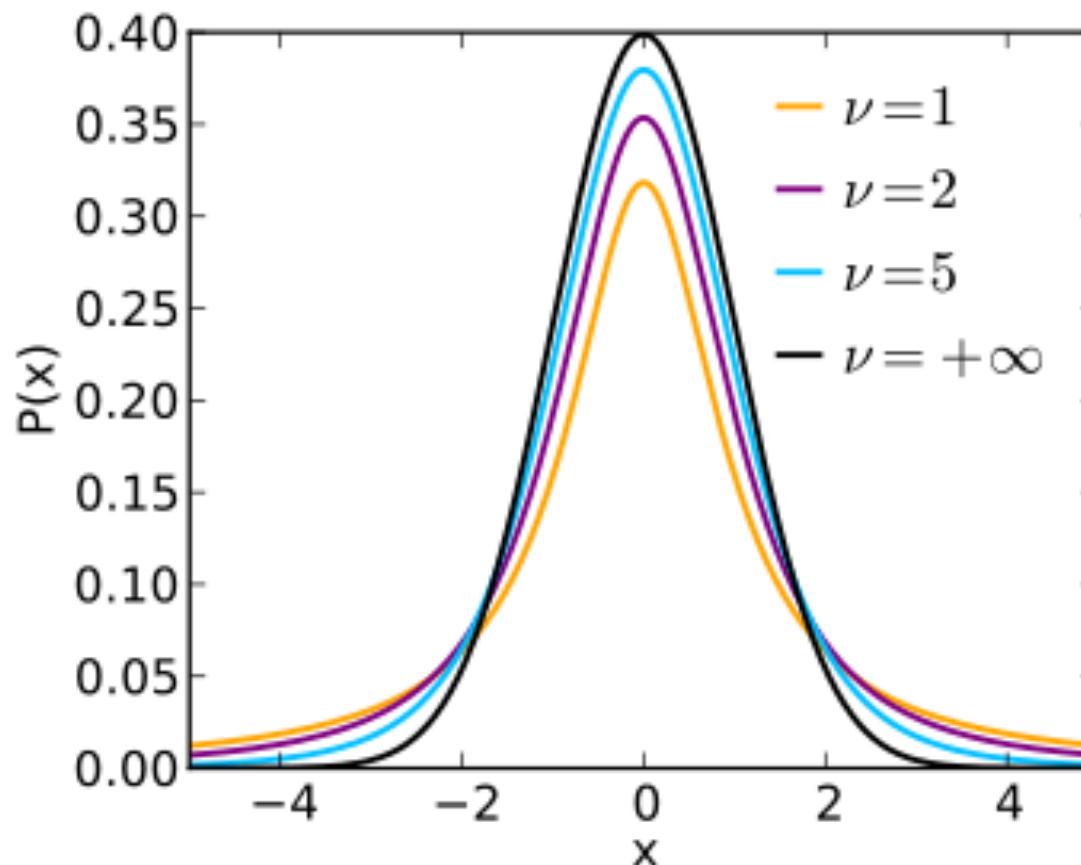
- $se^2(\hat{\beta}_i)$
 - ▣ Ex) two input variables

$$(X^T X)^{-1} = \begin{bmatrix} \beta_0 & \beta_1 & \beta_2 \\ x_{00} & & \\ & x_{11} & \\ & & x_{22} \end{bmatrix}$$

- $se^2(\hat{\beta}_0) = MSE \cdot x_{00} \rightarrow se(\hat{\beta}_0) = \sqrt{MSE \cdot x_{00}}$
- $se^2(\hat{\beta}_1) = MSE \cdot x_{11} \rightarrow se(\hat{\beta}_1) = \sqrt{MSE \cdot x_{11}}$
- $se^2(\hat{\beta}_2) = MSE \cdot x_{22} \rightarrow se(\hat{\beta}_2) = \sqrt{MSE \cdot x_{22}}$

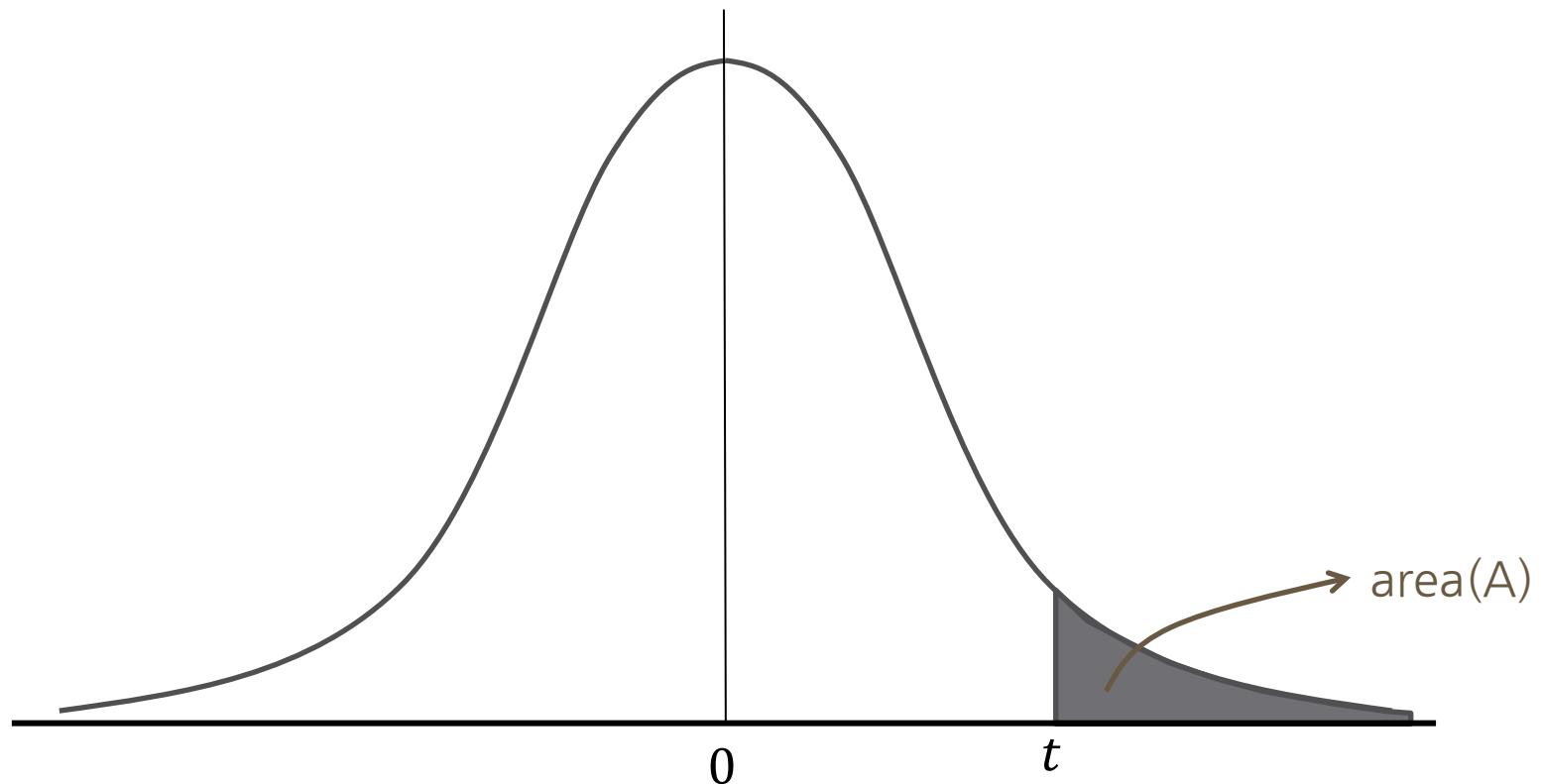
Test Concerning Regression Coefficients

- Test statistics of t -test follows student's t distribution with $n - p - 1$ degree of freedom
 - ▣ Probability density function of student's t distribution with different parameters(degree of freedom)

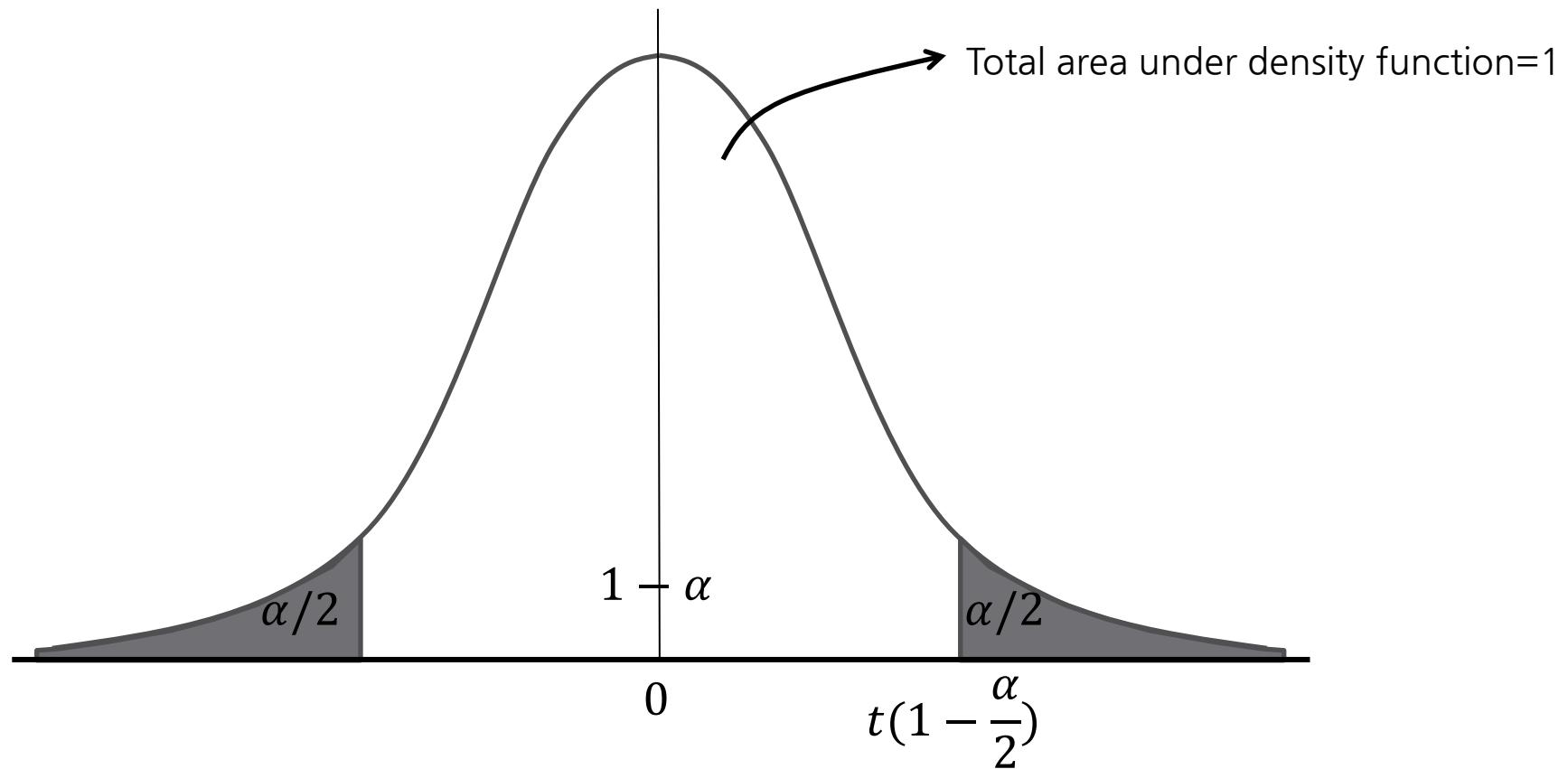


Test Concerning Regression Coefficients

- If $(\text{area under density function from } |t| \text{ to } \infty) < \frac{\alpha}{2}$
→ Reject null hypothesis → β_i is not zero
 - ▣ α is significance value
 - ▣ significance level is usually set to 0.1, 0.05
 - The higher significance level, the higher probability to reject null hypothesis



Test Concerning Regression Coefficients



Test Concerning Regression Coefficients

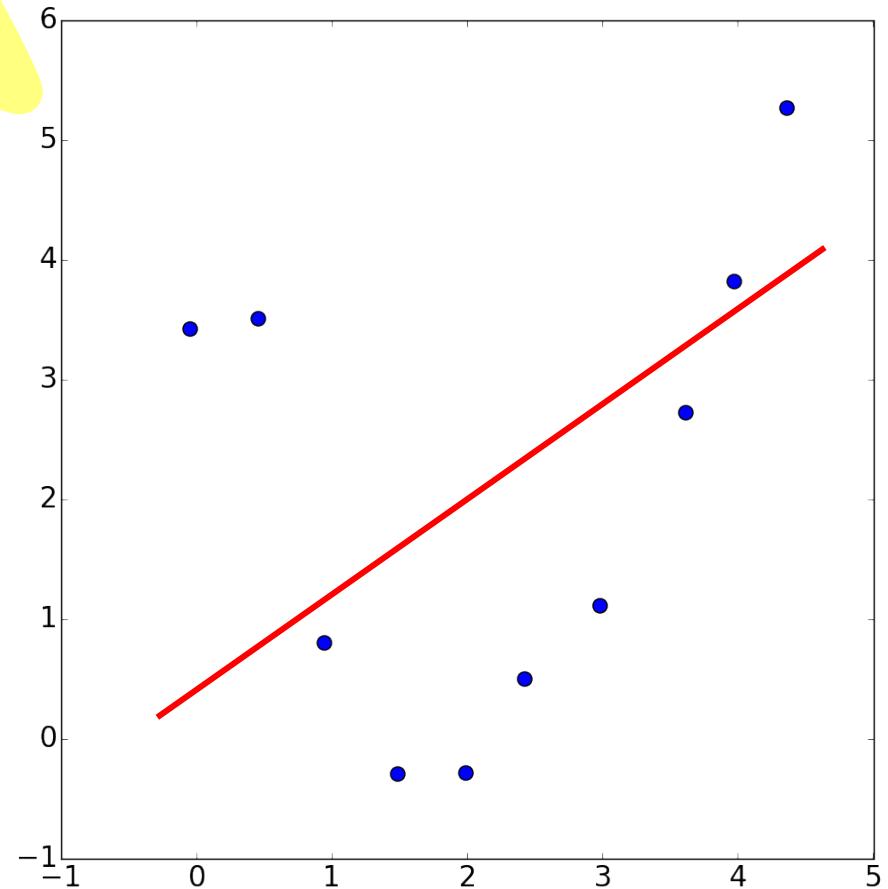
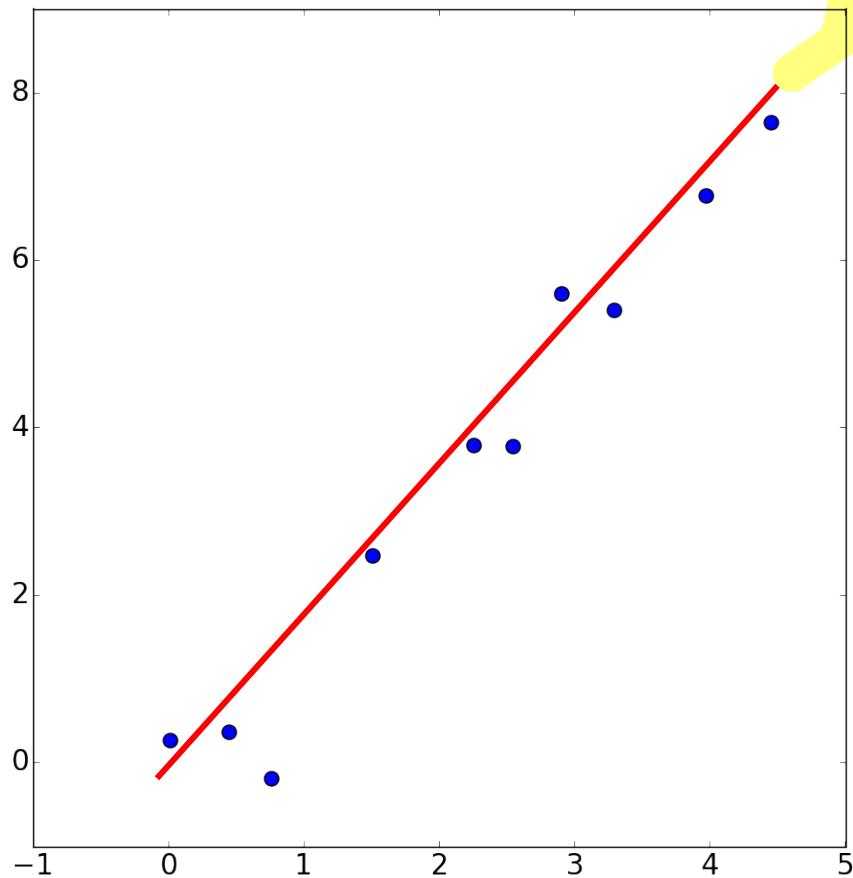
- How to calculate area?
 - ▣ Don't worry. There is pre-calculated table!

Student t-Table									
Alpha	0.250	0.200	0.150	0.100	0.050	0.025	0.010	0.005	0.0005
df									
1	1.000	1.376	1.963	3.078	6.314	12.706	31.821	63.656	636.578
2	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	31.600
3	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	12.924
4	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	8.610
5	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	6.869
6	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.959
7	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	5.408
8	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	5.041
9	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.781
10	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.587
11	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.437
12	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	4.318
13	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	4.221
14	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	4.140
15	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	4.073
16	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	4.015
17	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.965
18	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.922
19	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.883
20	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.850

t value that area is 0.25 with 20 degree of freedom

Goodness-of-fit

- How to measure quantitatively performance of fitted models?
 - Calculate goodness-of-fit



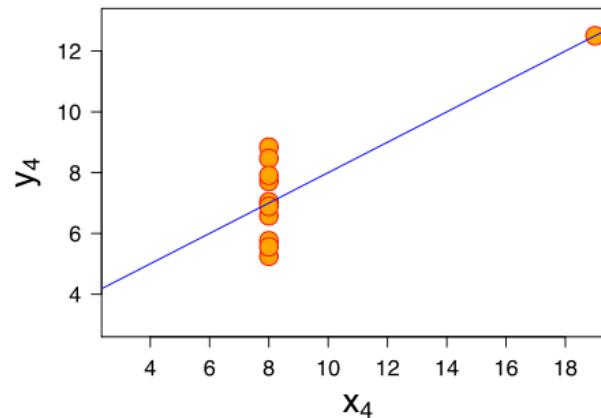
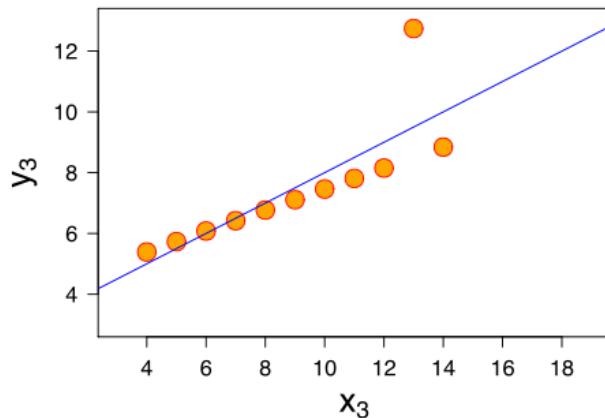
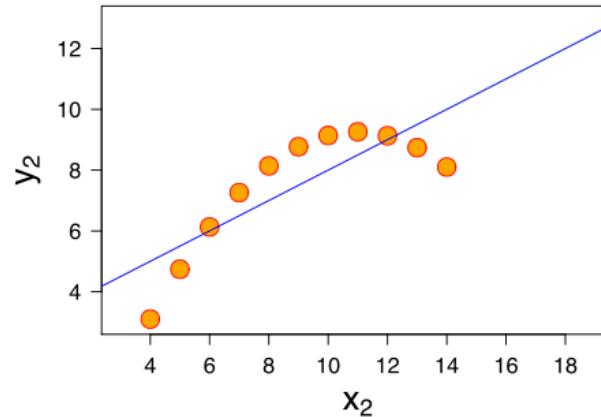
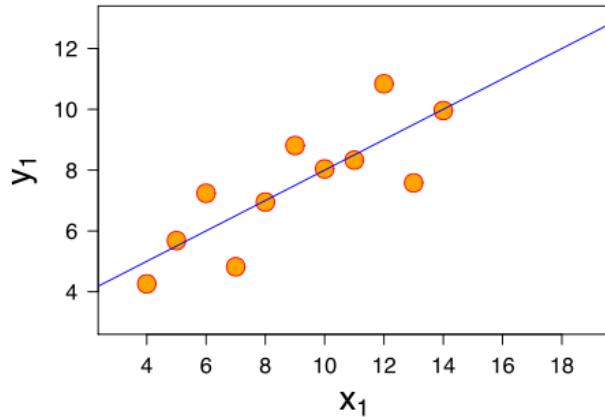
R^2

- Statistical measures for goodness-of-fit
 - R^2 ($0 \leq R^2 \leq 1$)

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

R^2 is NOT All-around Player

- Anscombe's quartet
 - ▣ The same linear regression line but are themselves very different.



Adjusted R^2

- Adding more input variables to the regression model increases R^2 and never reduce it
 - ▣ Tend to add more input variables to the model

Is always right to add more variables?

- Adjusted R^2

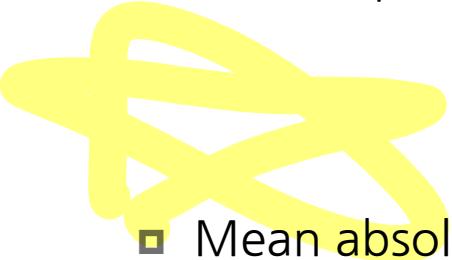
$$R^2_{adj} = 1 - \frac{\frac{SSE}{n-p-1}}{\frac{SST}{n-1}} = 1 - \left(\frac{n-1}{n-p-1} \right) (1 - R^2)$$

Depend on the number of input variables 

- ▣ Penalty on the number of input variable by $n - p - 1$
- ▣ Adjusted R^2 may actually become smaller when another input variable is introduced into the model

Performance metrics

- Functions to measure regression performance
 - ▣ Mean squared error


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

- ▣ Mean absolute error

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n}$$

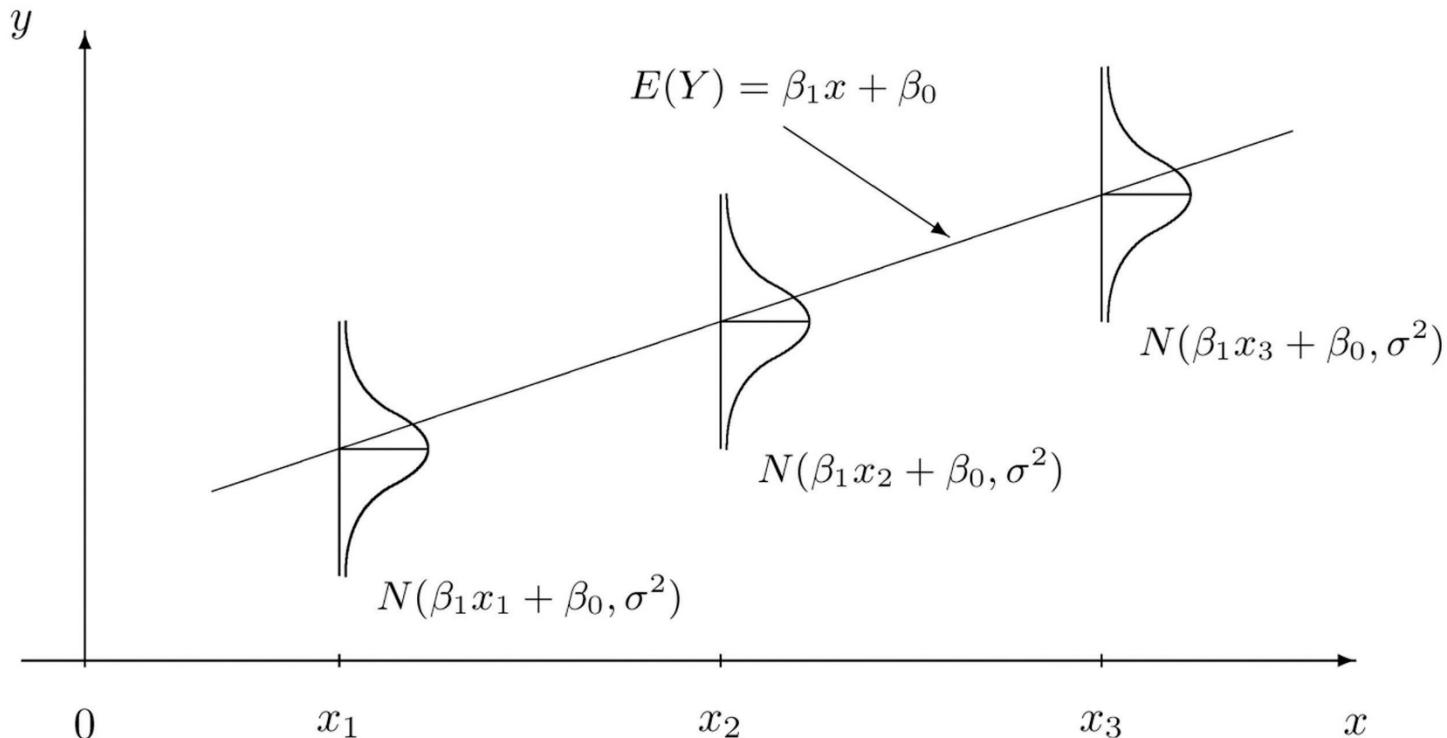
- ▣ Median absolute error

- robust to outliers

$$MedAE = \text{median}(|y_1 - \hat{y}_1|, \dots, |y_n - \hat{y}_n|)$$

Check Appropriateness of Linear Regression

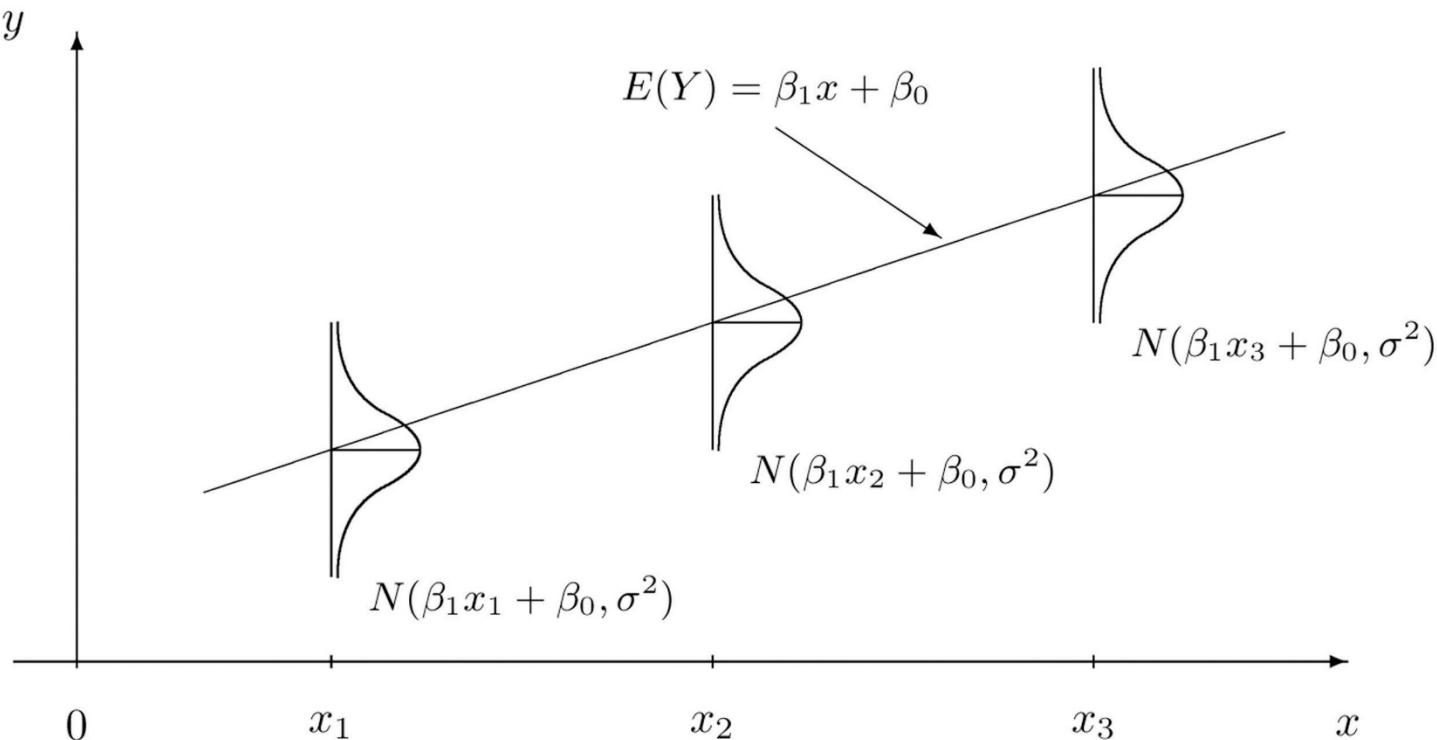
- Do you remember main assumptions of linear regression?



Main Assumption of Linear Regression

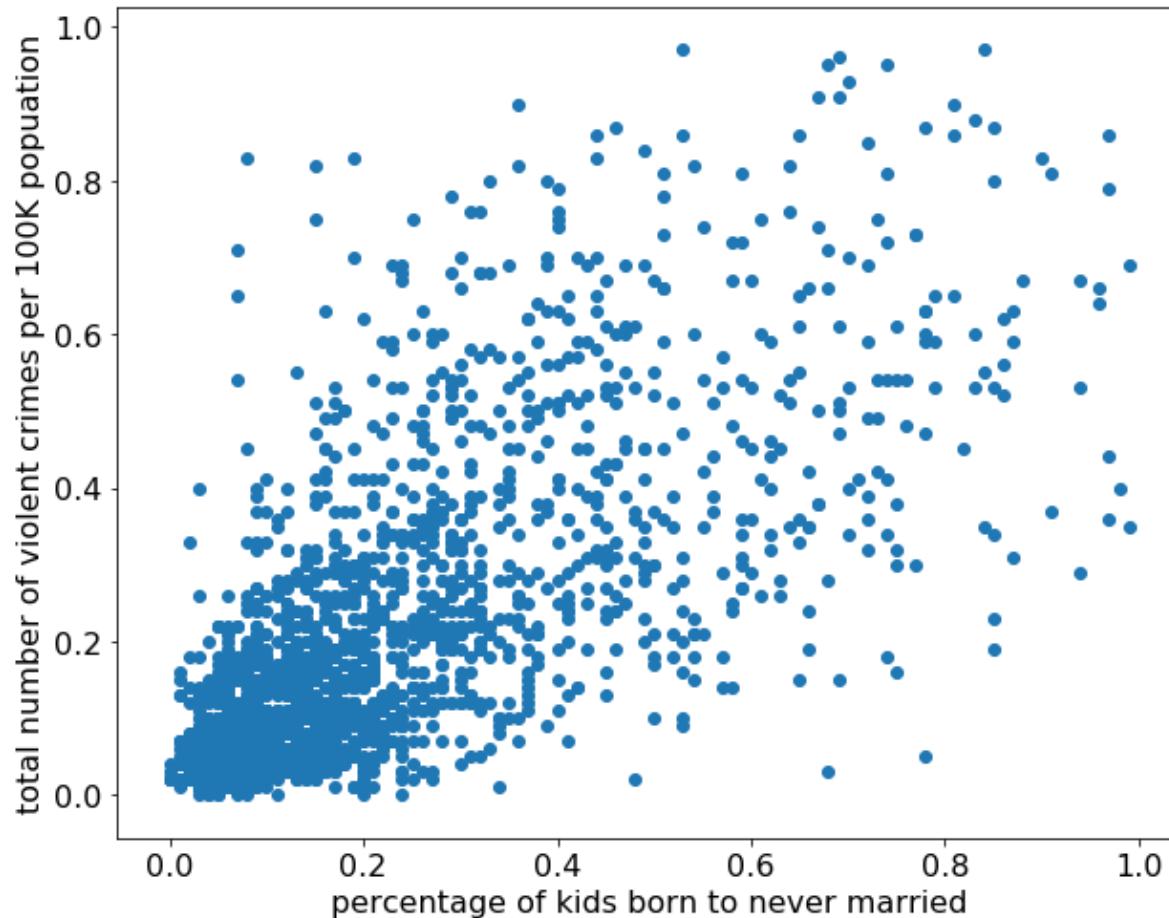
- Linear regression analysis makes several key assumptions

- Linear relationship
- Homoscedasticity
- Normality
- No or little multicollinearity



Check Appropriateness of Linear Regression

- **Linear relationship**
 - ▣ Check relationships between input variables and a responsive variable



Relationships between Input Variables

- If some of input variables are highly correlated, regression coefficients are unstable

	1	2	3	4	5	6	7	8	9	10
x_1	98	120	140	195	181	128	107	106	88	77
x_2	24	35	36	51	45	30	29	24	22	19
x_3	21	11	31	42	57	82	67	13	55	36

- Correlation matrix

$$corr = \begin{bmatrix} 1.00 & 0.98 & 0.17 \\ 0.98 & 1.00 & 0.11 \\ 0.17 & 0.11 & 1.00 \end{bmatrix}$$

- x_1 and x_2 are highly correlated

※ Covariance

- Variance of a random variable X is the expected value of the squared deviation from the mean ($\mu = \mathbb{E}[X]$)

$$Var(X) = \mathbb{E}[(X - \mu)^2]$$

- Sample variance is calculated by

$$s_x^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{(n - 1)}$$

- Covariance is a measure of how much two random variables change together

$$Cov(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$$

- Variance is the covariance of a random variable with itself

$$Var(X) = Cov(X, X)$$

- Sample covariance is calculated by

$$q_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n - 1}$$

※ Correlation

- Any statistical relationship, whether causal or not, between two random variables or bivariate data
 - ▣ Pearson's correlation coefficient
 - The most popular correlation

$$\rho_{X,Y} = \text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

- Sample correlation coefficient

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)s_x s_y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum_{i=1}^n (x_i - \bar{x})^2)(\sum_{i=1}^n (y_i - \bar{y})^2)}}$$

Relationships between Input Variables

- Two difference cases

	1	2	3	4	5	6	7	8	9	10
y_1	295	310	404	567	574	532	442	283	366	285
y_2	282	311	402	581	573	523	446	277	374	274

- Output values of two cases are quite similar
- Regression coefficient for y_1 and y_2

$$\text{Case 1: } [\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3] = [2.16 \quad 0.14 \quad 2.88]$$
$$\text{Case 2: } [\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3] = [1.73 \quad 2.18 \quad 2.97]$$

- Because x_1 and x_2 are highly correlated, explained variance by x_2 is also explained by $x_1 \rightarrow$ Coefficient of x_2 is quite unstable

Why This Situation Happens

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- To estimate regression coefficients, inverse matrix of $\mathbf{X}^T \mathbf{X}$ should be calculated
 - Ill-conditioned matrices
 - If a small change in the coefficient matrix results in a large change in the solution, the coefficient matrix is called ill-conditioned
- $$\begin{cases} x + y = 2 \\ x + 1.001y = 2 \end{cases} \quad \text{and} \quad \begin{cases} x + y = 2 \\ x + 1.001y = 2.001 \end{cases}$$
- Left: $x = 2, y = 0$
 - Right: $x = 1, y = 1$

$\begin{bmatrix} 1 & 1 \\ 1 & 1.001 \end{bmatrix}$ is ill-conditioned

Variance Inflation Factor



- Variance inflation factor(VIF) quantifies the severity of multicollinearity in a least square method

[Multicollinearity]

A Phenomenon in which two or more input variables in a multiple regression model are highly correlated

→ In this situation the coefficient estimates of the multiple regression may change erratically in response to small changes in the model or the data

- Variance of estimated coefficients for j – th input variable

$$var(\hat{\beta}_j) = se^2(\hat{\beta}_j) = [MSE(\mathbf{X}^T \mathbf{X})^{-1}]_{j,j} = \frac{MSE}{(n - 1)se^2(x_j)} \frac{1}{1 - R_j^2}$$

- R_j^2 is the R^2 for the regression of the x_j on the other input variables

- VIF

$$\frac{1}{1 - R_j^2}$$

Variance Inflation Factor

- Calculate VIF
 - ▣ Step 1) Apply least square method to regression problem that i -th input variable is regressed by the remained input variables

$$x_i = \alpha_1 x_1 + \cdots + \alpha_{i-1} x_{i-1} + \alpha_{i+1} x_{i+1} + \cdots + \alpha_p x_p + \alpha_0 + \epsilon$$

- ▣ Step 2) Calculate R^2 for above regression problem and set the value as R_i^2
- ▣ Step 3) Calculate VIF from R_i^2

$$VIF = \frac{1}{1 - R_i^2}$$

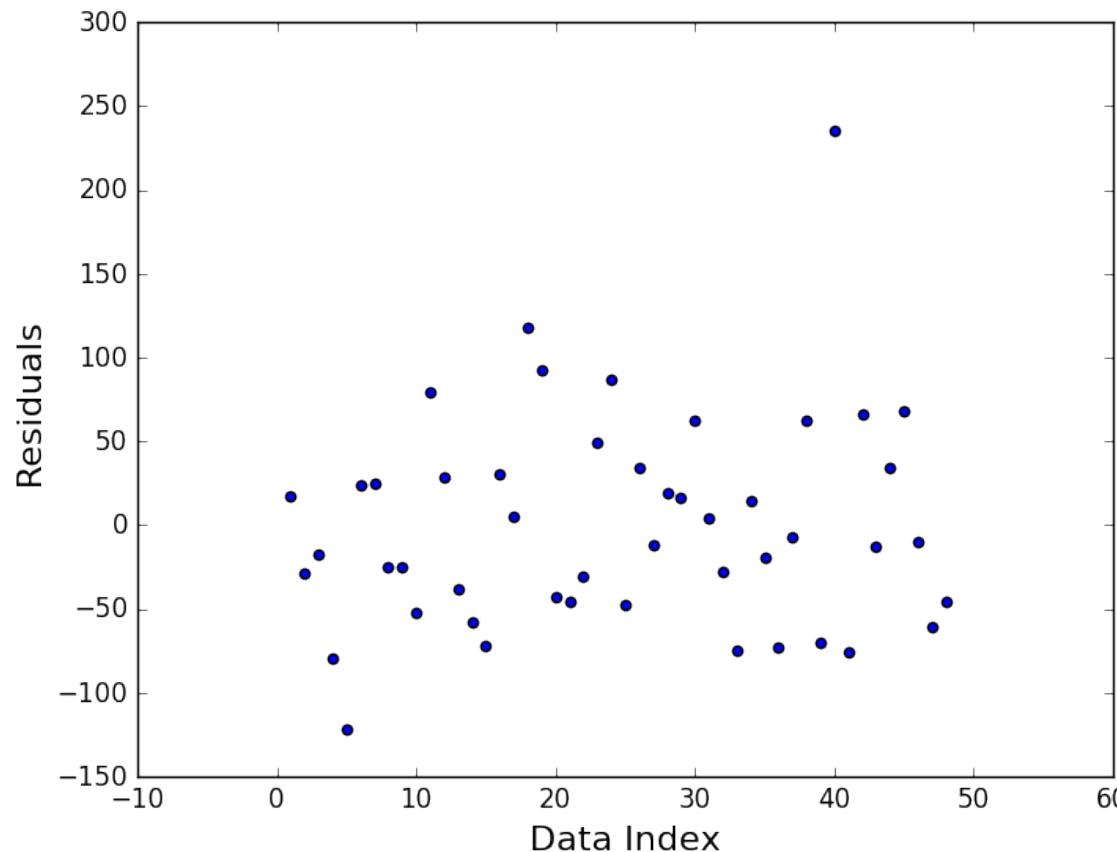
- A rule of thumb is that if $VIF(\hat{\beta}_i) > 10$ then multicollinearity is high
 - ▣ In this case, do not use x_i as explanatory variable to estimate output

Check Appropriateness of Linear Regression

□ Normality

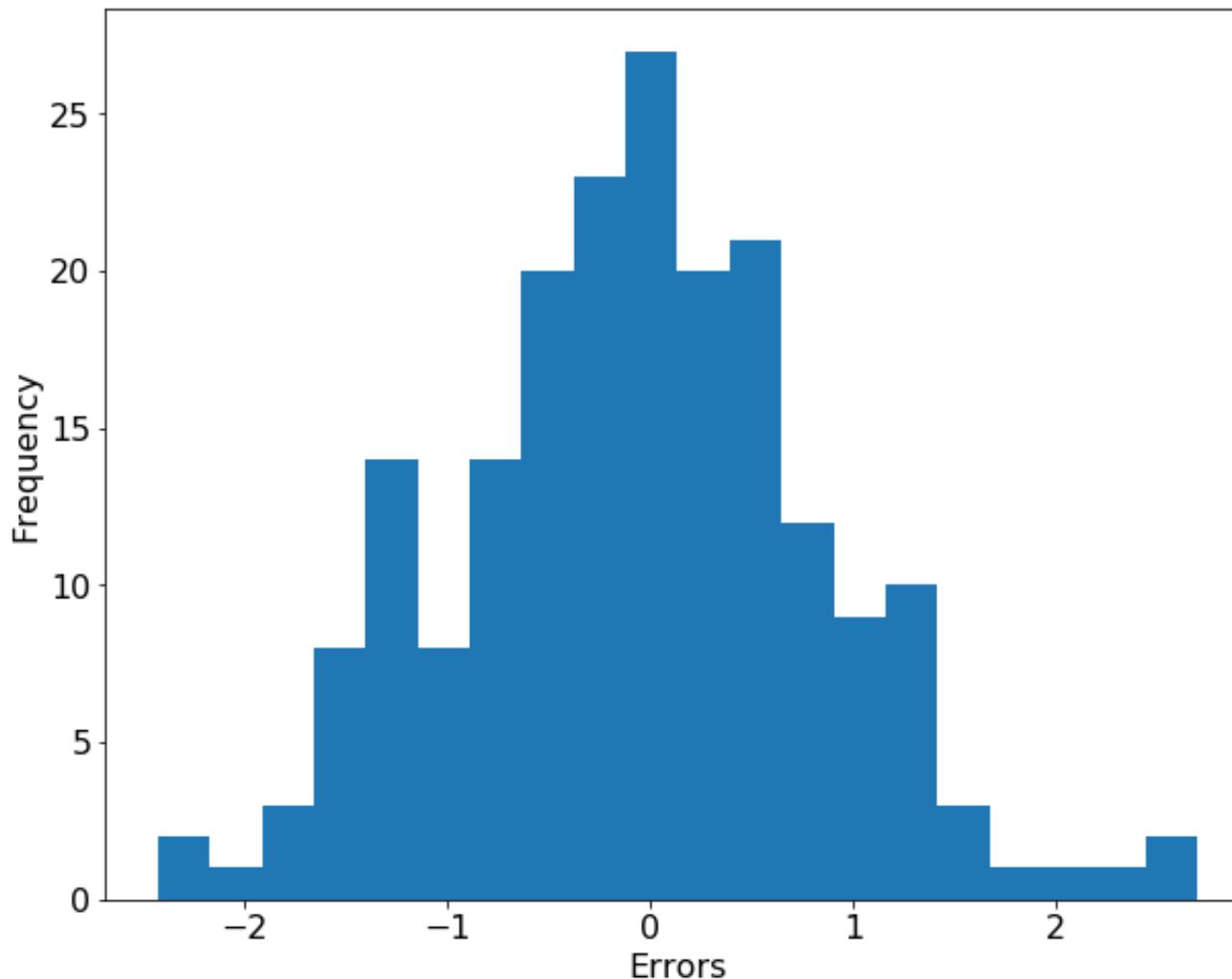
- Errors should follow normal distribution
- Calculate errors (residuals) and check normality

$$e_i = y_i - \hat{y}_i$$



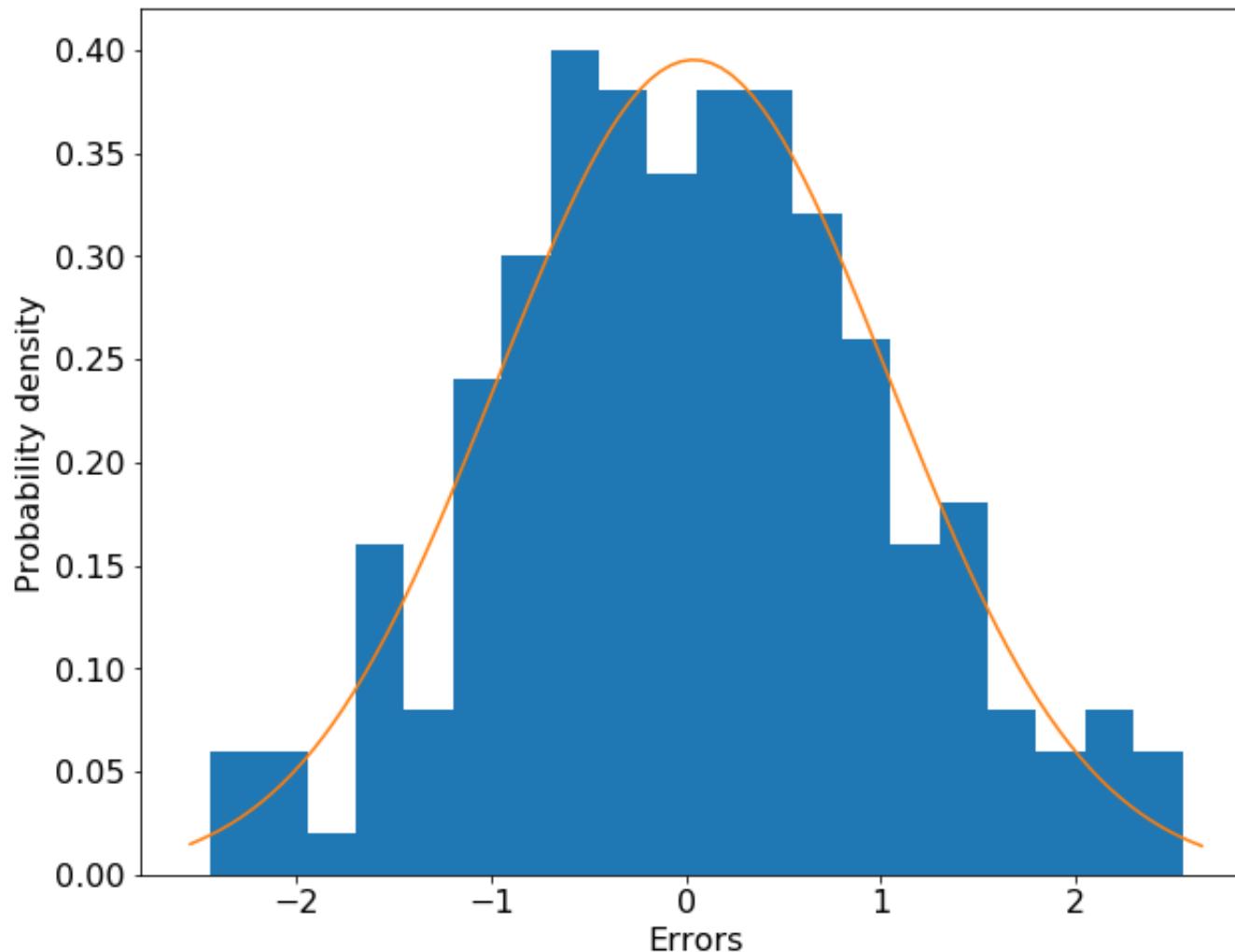
Check Appropriateness of Linear Regression

- Histogram



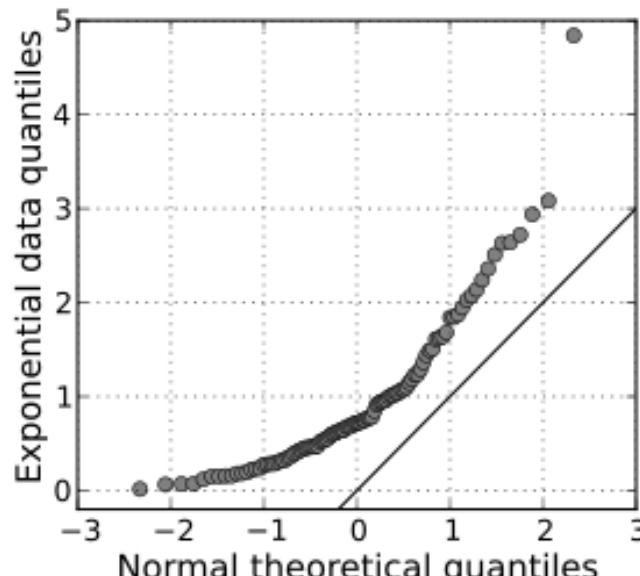
Check Appropriateness of Linear Regression

- Histogram

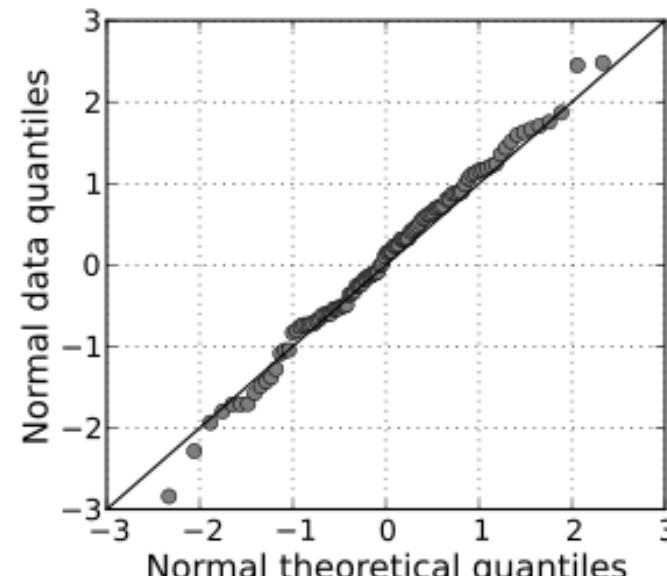


Check Appropriateness of Linear Regression

- Q-Q plot *Normality*
 - A probability plot, which is a graphical method for **comparing two probability distributions** by plotting their quantiles against each other
 - Quantiles are cutpoints dividing a set of observations into equal sized groups
 - q -Quantiles are values that partition a finite set of values into q subsets of (nearly) equal sizes
 - Median is 2-quartile, 0.5 quantile and 50 percentile

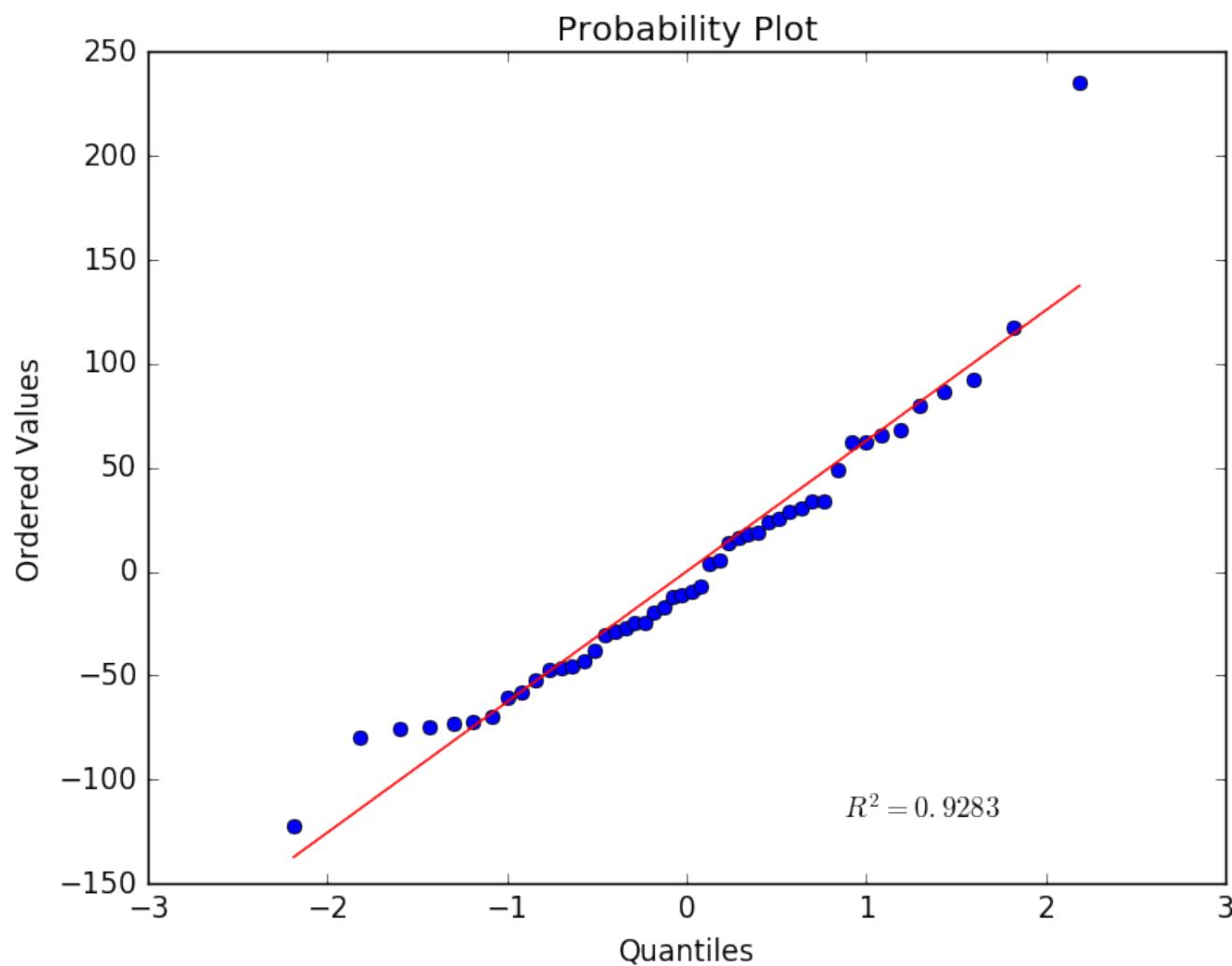


$X \sim Exp(1)$



$X \sim N(0,1)$

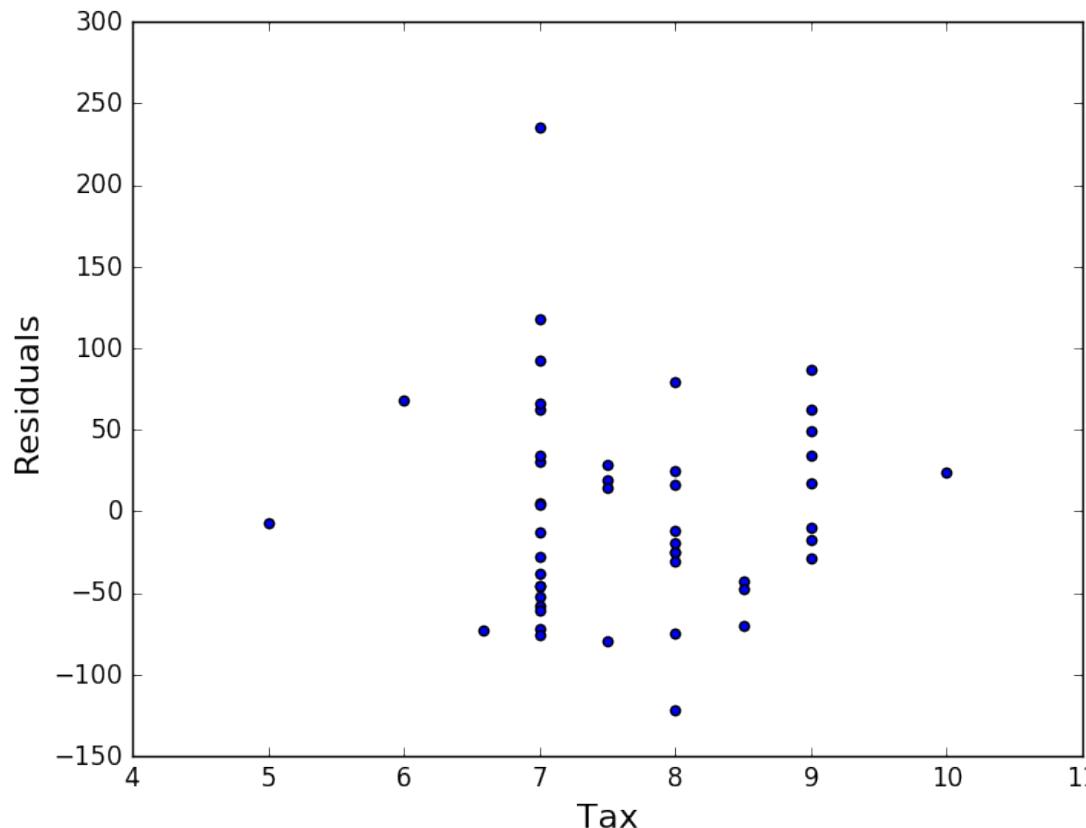
Q-Q Plot



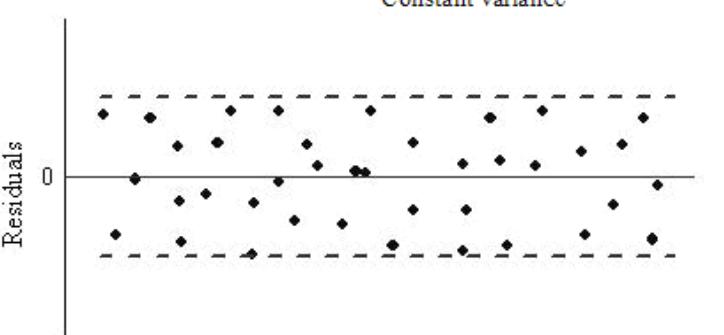
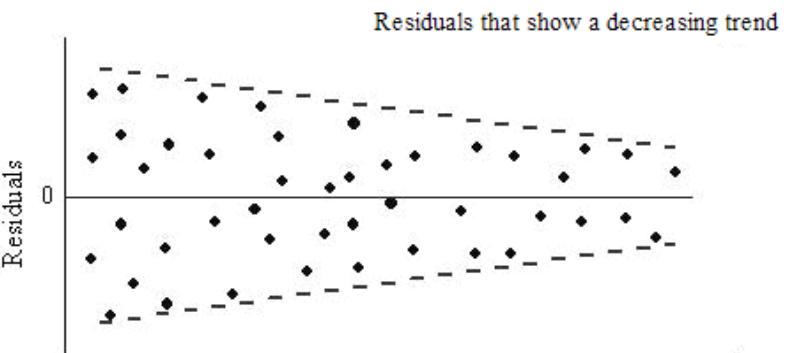
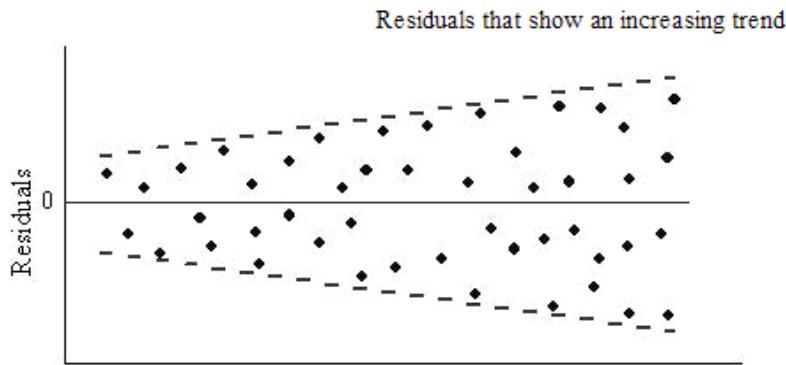
Check Appropriateness of Linear Regression

□ Homoscedasticity↔Heteroscedasticity

- Check whether all random variables in the sequence or vector have the same finite variance

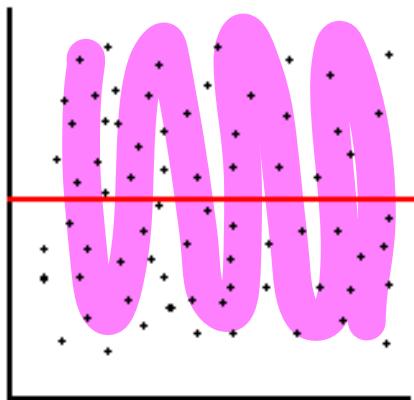


Check Appropriateness of Linear Regression

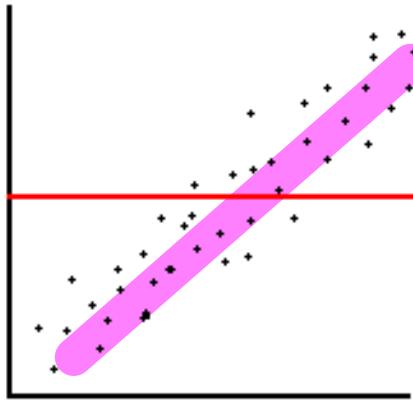


Check Appropriateness of Linear Regression

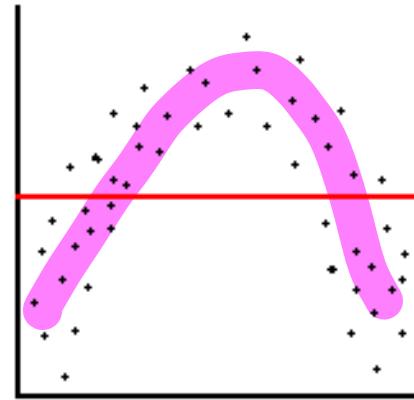
- Residual plot



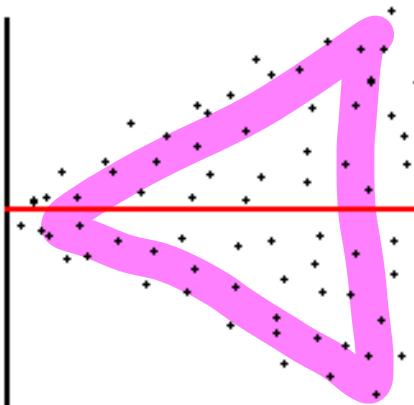
(a) Unbiased and Homoscedastic



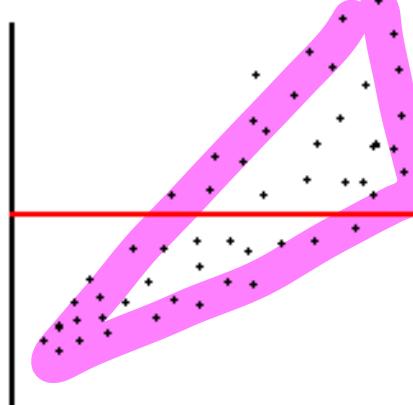
(b) Biased and Homoscedastic



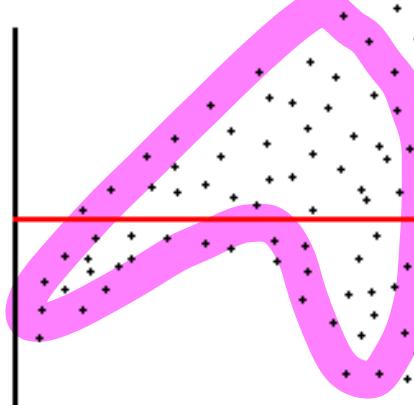
(c) Biased and Homoscedastic



(d) Unbiased and Heteroscedastic



(e) Biased and Heteroscedastic

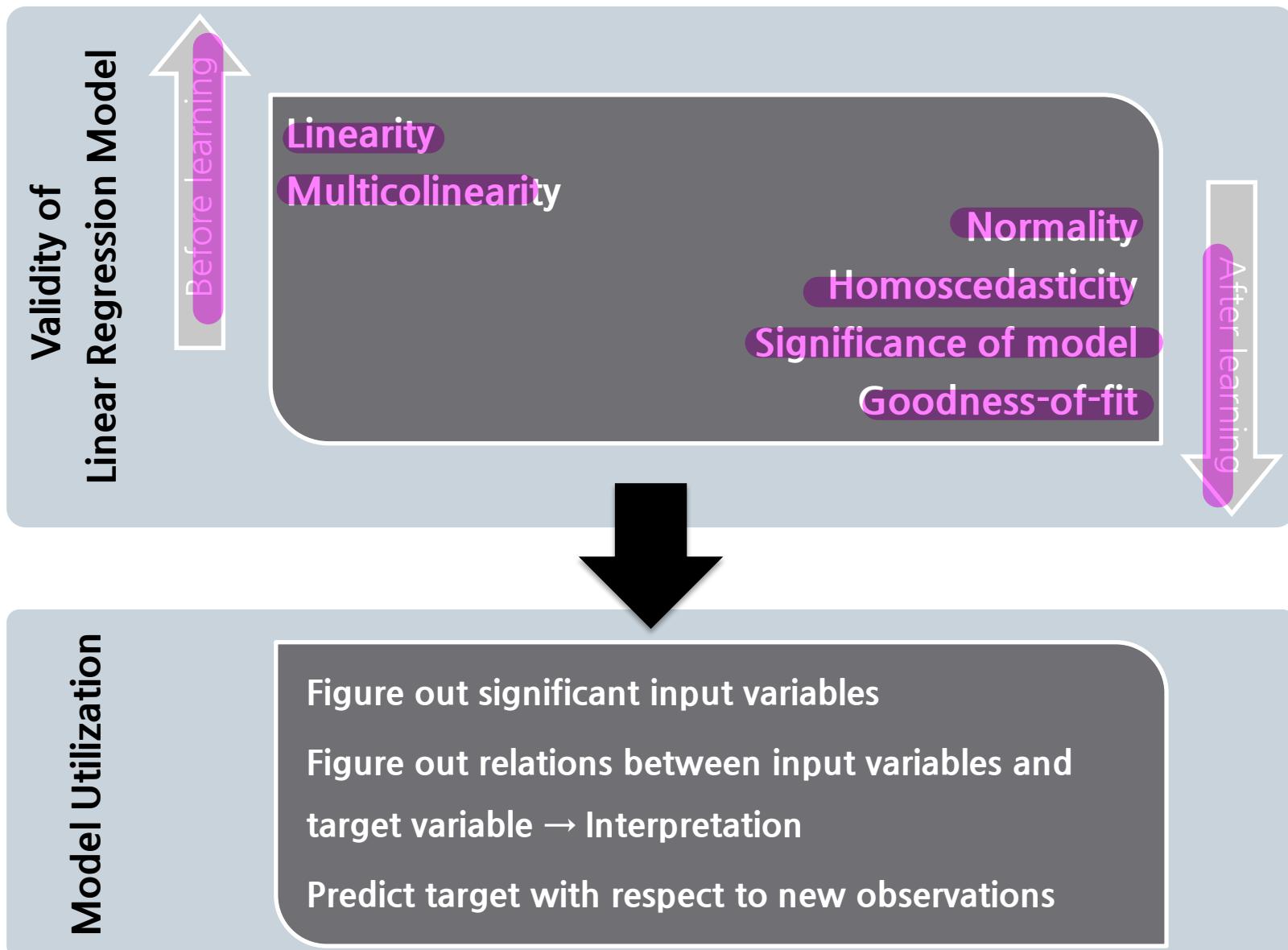


(f) Biased and Heteroscedastic

Interpretation & Prediction

- If the fitted regression model is appropriate and significant you can use the model for future use
 - ▣ Linear regression models have strength in interpretation
 - Each coefficient explain relationship between each explanatory variable and the target variable
 - ▣ Based on the fitted model, predict the target on test samples

Overall Process for Linear Regression



Feature Scaling

- Predict consumption of petrol
 - ▣ Linear model by least square method

$$y = -34.8x_1 - 0.0666x_2 - 0.002x_3 + 1336x_4 + 377.3$$

Petrol Tax(\$)	Average Income (\$)	Paved Highways (miles)	Proportion of population with driver's license	Consumption of petrol (M of gallons)
9	3571	1976	0.525	541
9	4092	1250	0.572	524
9	3865	1586	0.58	561
7.5	4870	2351	0.529	414
...

How about changing scale of variable?

Feature Scaling

- Change unit of paved highways from mile to cm

$$1 \text{ mile} = 160934.4 \text{ cm}$$

Petrol Tax(\$)	Average Income (\$)	Paved Highways (cm)	Proportion of population with driver's license	Consumption of petrol (M of gallons)
9	3571	31683974.4	0.525	541
9	4092	20043000	0.572	524
9	3865	25430558.4	0.58	561
7.5	4870	37696874.4	0.529	414
...

- Linear regression on new data

$$y = -34.8x_1 - 0.0666x_2 - 1.5 \times 10^{-7}x_3 + 1336x_4 + 377.3$$

Feature Scaling

- Scale change only affects on the changed variable
 - ▣ Coefficients of other variables are not changed
 - ▣ If variable x is replaced with ax , coefficient of x , β by linear regression is changed to β/a
 - ▣ If scale of certain variable is too large, coefficient of the variable might be too small
→ It is better to change scale

Variable Transformation

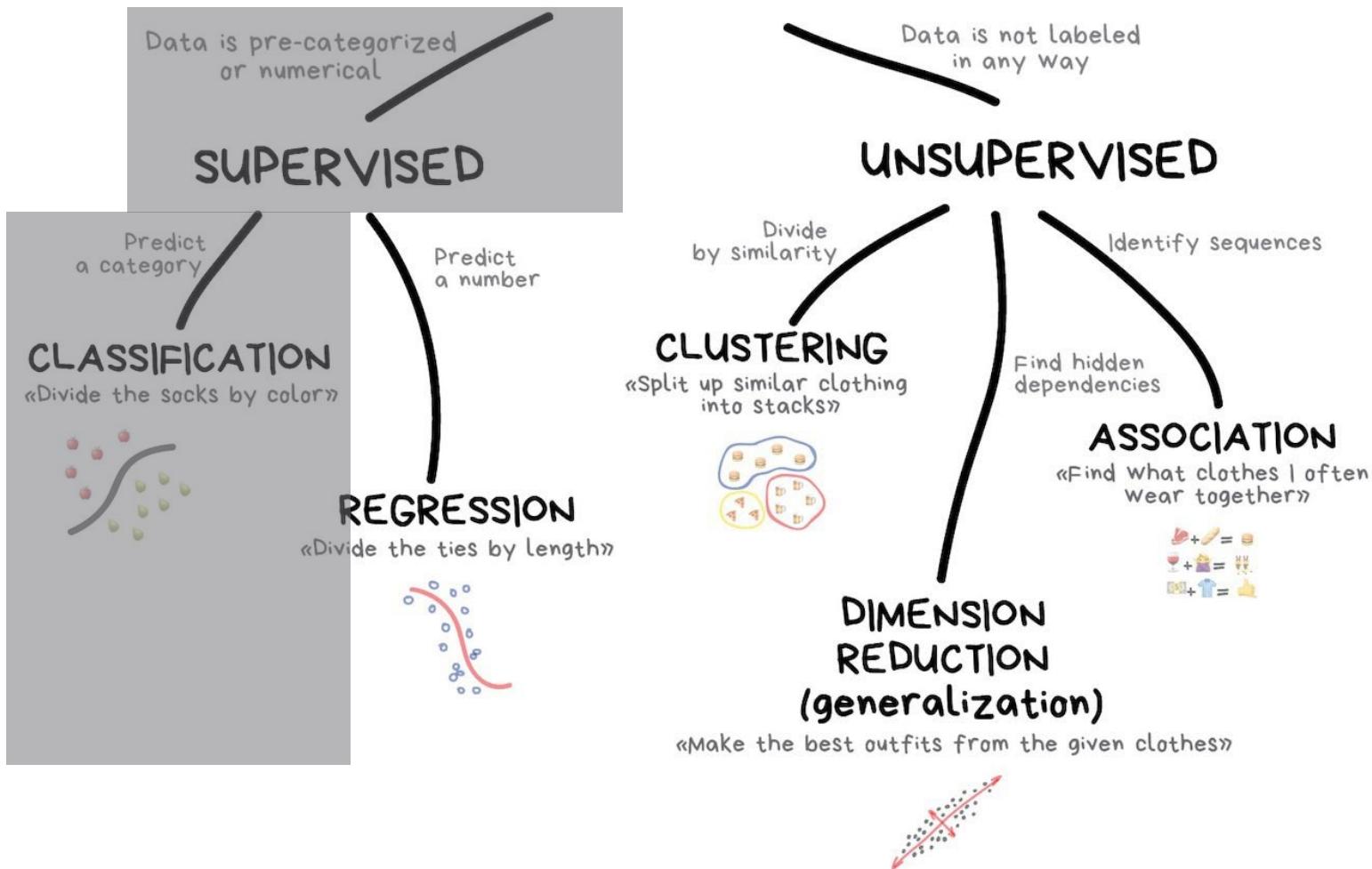
- Linear regression algorithm is quite simple, but it can be extended using transformation
 - $x \rightarrow x^2$
 - $x \rightarrow \log x$
 - $x \rightarrow \sqrt{x}$

LOGISTIC REGRESSION

Week06

Topics Covered in This Class

CLASSICAL MACHINE LEARNING



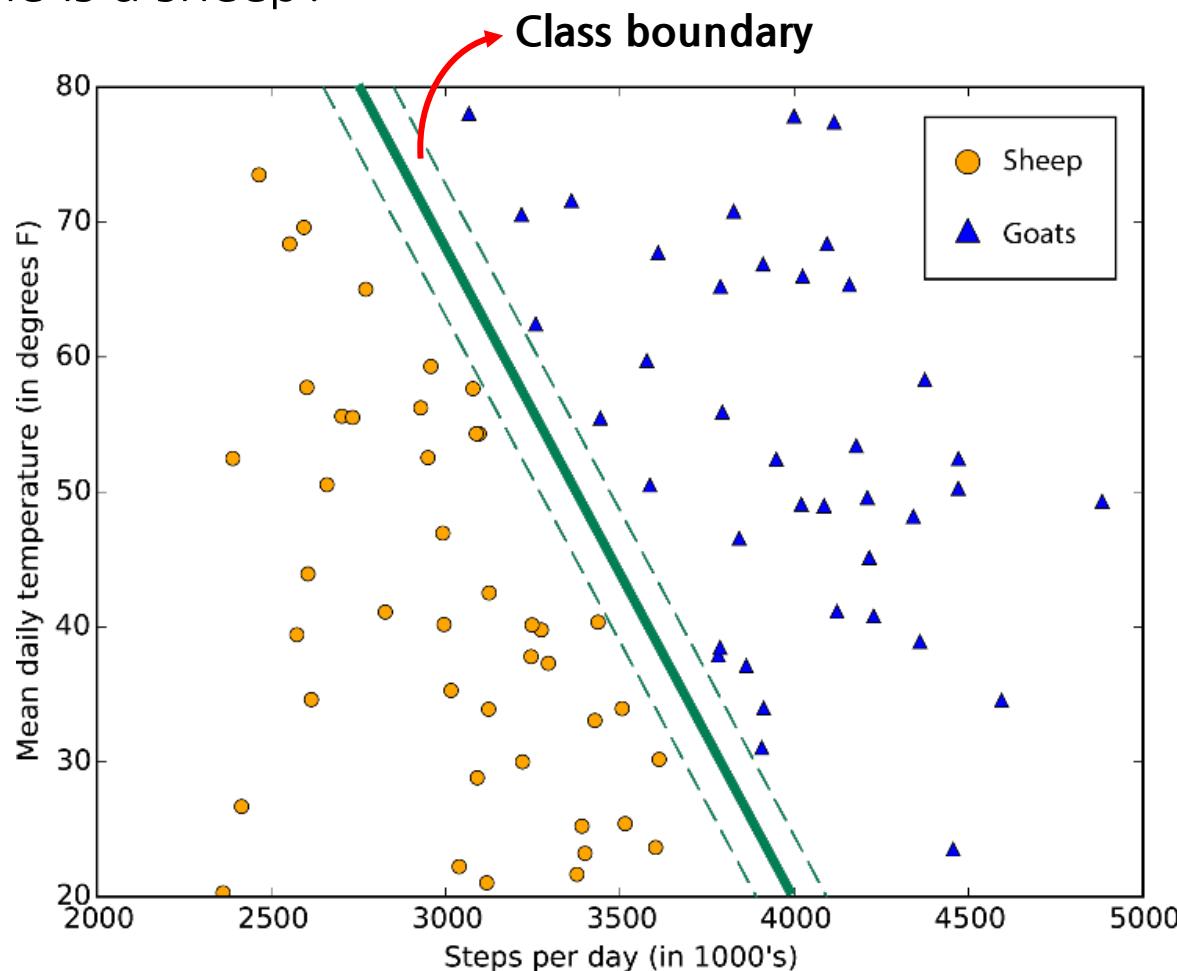
Logistic Regression

Supervised: Classification

- Classification problem
 - ▣ Output is categorical variable
 - Spam/Non Spam
 - Male/Female
 - Long/Medium/Short
 - O/X
 - Binary classification
 - ▣ The number of categories is 2
 - ▣ Generally, these two categories are denoted as 0 and 1
 - 0 and 1 are not integer in this case
 - Multi-class classification
 - ▣ More than two classes
- $$y \in \{0,1\}$$
- $$y \in \{1,2,\dots,C\}, \quad C > 2$$

Supervised: Classification

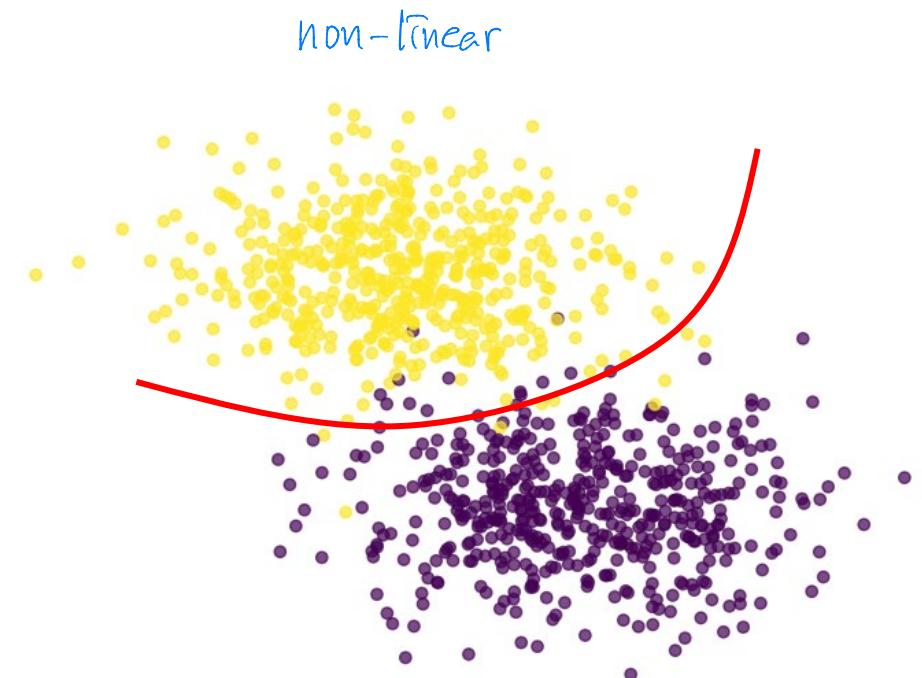
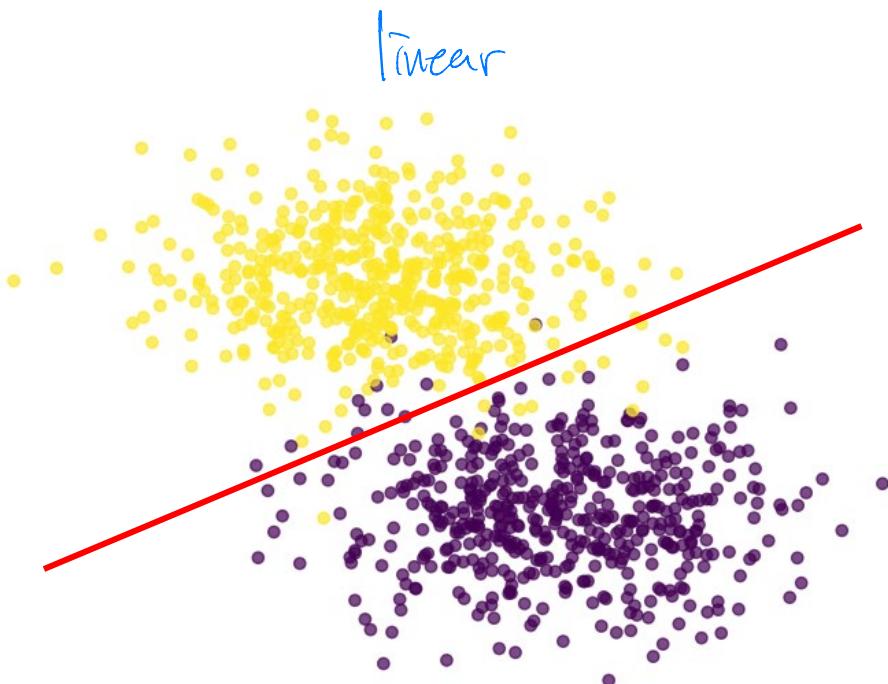
- Which one is a sheep?



The Decision Boundary of Classifiers

- Decision boundary

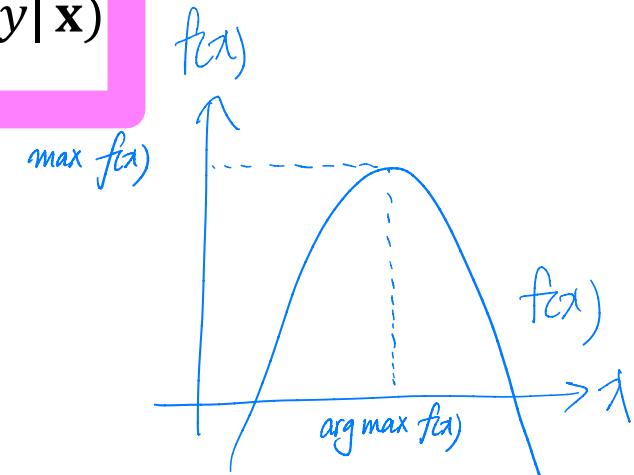
$$y = f(X), \quad y \in \{1, 2, \dots, C\}$$



Types of Classifiers

- A classifier is a function that assigns to a sample, \mathbf{x} a class label \hat{y}
$$\hat{y} = f(\mathbf{x})$$
- A **probabilistic classifier** obtains **conditional distributions** $\Pr(Y|\mathbf{x})$, meaning that for a given $\mathbf{x} \in \mathcal{X}$, they assign probabilities to all $y \in Y$
 - Hard classification

$$\hat{y} = \arg \max_y \Pr(Y = y | \mathbf{x})$$



Logistic Regression

- Logistic regression
 - ▣ Regression model where the dependent variable is categorical
 - ▣ The probabilities of possible outcomes are modeled using explanatory variables

$$f(x) = P(Y|X)$$

- $0 \leq f(x) \leq 1$

How can we ensure that $f(x)$ remains within [0,1]?

$$P(Y=0|X) + P(Y=1|X) = 1$$

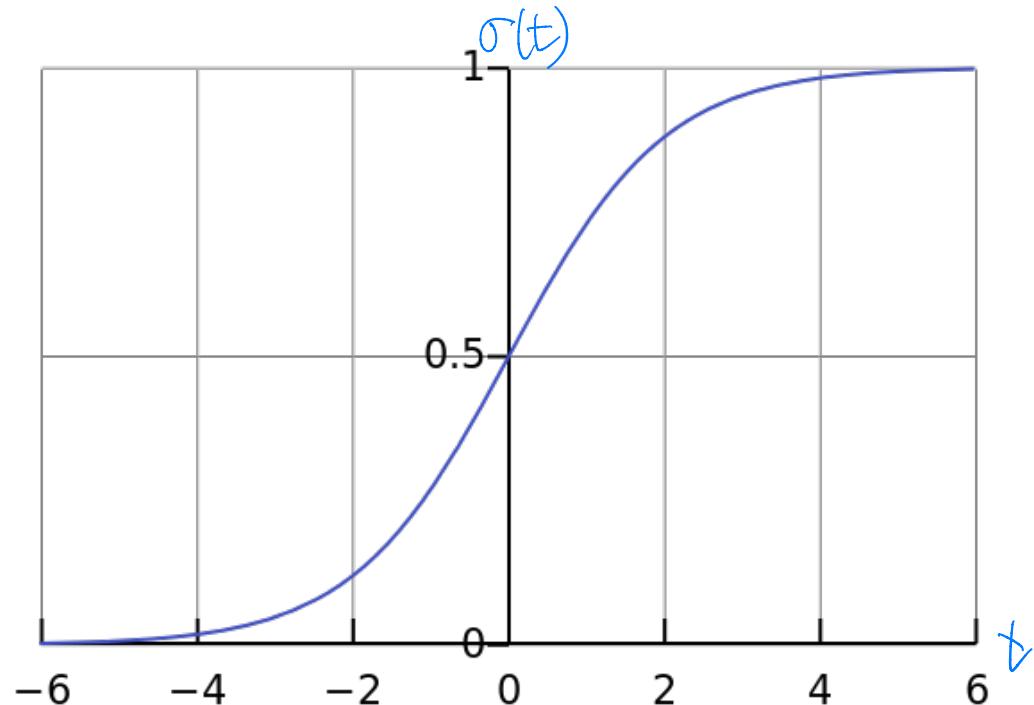
Logistic Regression: Logistic function

- Logistic function is the function that can take an input with any value from negative to positive infinity, whereas the output always takes values between zero and one

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

$$\lim_{t \rightarrow -\infty} \sigma(t) = 0$$

$$\lim_{t \rightarrow \infty} \sigma(t) = 1$$



- In logistic regression, t is determined by explanatory variables

Logistic Regression

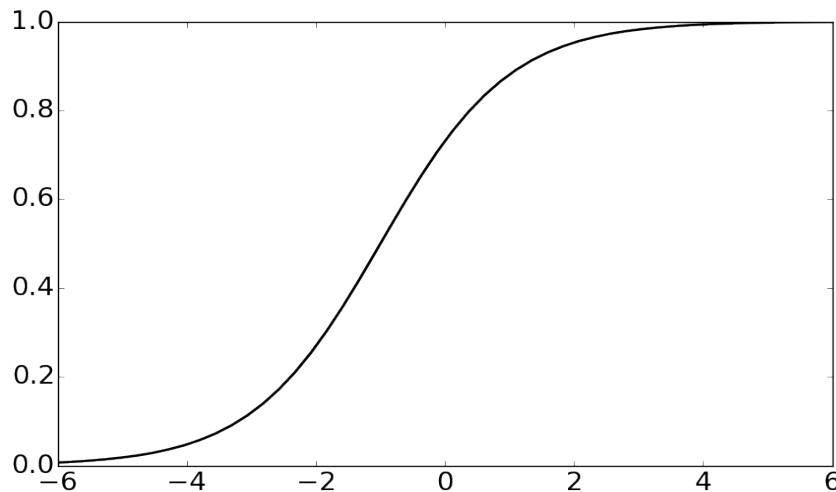
- t is determined by linear combination of explanatory variables

$$t = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \quad (-\infty, \infty)$$

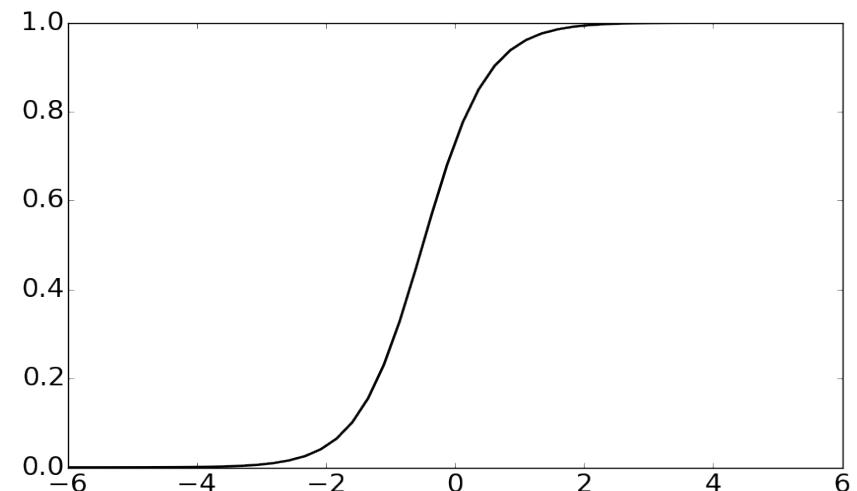


$$f(x) = P(Y = 1 | \mathbf{x}) = \frac{1}{1 + e^{-\beta_0 - \beta_1 x_1 - \beta_2 x_2 - \cdots - \beta_p x_p}} \quad [0, 1]$$

$$t = 1 + x$$

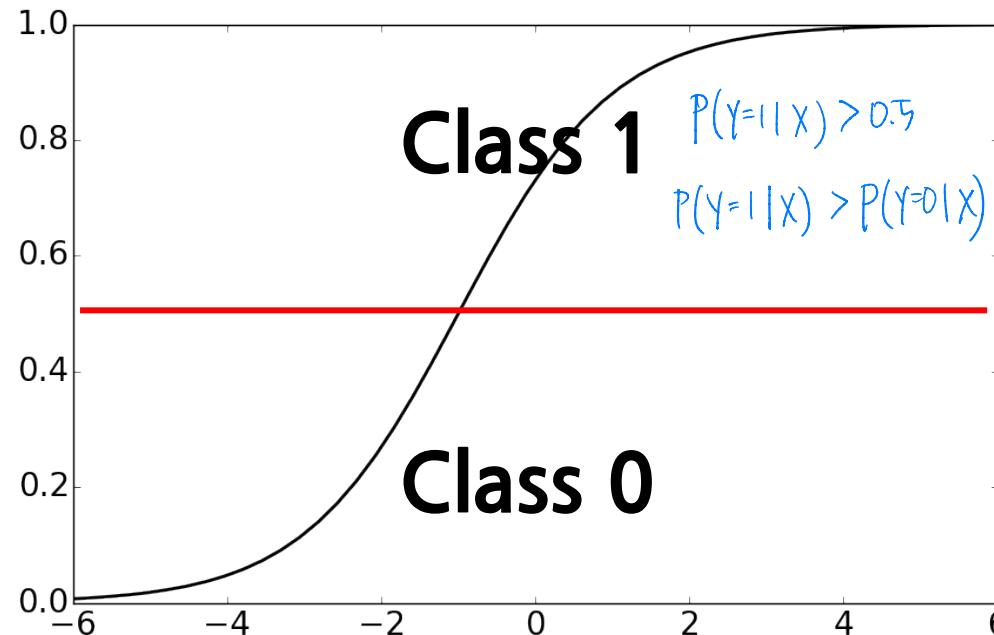


$$t = 1 + 2x$$



Logistic Regression

- Determine class
 - ▣ Set class boundary
 - Without any prior knowledge about class, set 0.5



- If you have some knowledge about class distribution, class boundary can be determined based on the knowledge

Logistic Regression: Parameter Estimation

$$f(x) = P(Y = 1|\mathbf{x}) = \frac{1}{1 + e^{-\beta_0 - \beta_1 x_1 - \beta_2 x_2 - \dots - \beta_p x_p}}$$

- Unknown parameters
 - $\beta_0, \beta_1, \dots, \beta_p$
- Logistic regression should estimate $\beta_0, \beta_1, \dots, \beta_p$ based on the given observations

Maximum Likelihood Estimation



- Maximum likelihood estimation (MLE)
 - ▣ Method of estimating the parameters of statistical model
 - ▣ Given a statistical model, maximize likelihood
- Example of maximum likelihood estimation
 - ▣ Suppose that data set $D = \{x_1, x_2, \dots, x_n\}$ consists of n independent and identically distributed(iid) samples coming from a distribution with an unknown probability density function $f(x)$
 - ▣ Assume $f(x)$ belongs to a certain type of distributions with parameters θ
 - ▣ Joint density function for all observations

$$f(x_1, x_2, \dots, x_n | \theta) = f(x_1 | \theta) \times f(x_2 | \theta) \times \cdots \times f(x_n | \theta)$$

because x_i is iid sample

- ▣ Likelihood

$$\mathcal{L}(\theta; x_1, x_2, \dots, x_n) = f(x_1, x_2, \dots, x_n | \theta) = \prod_{i=1}^n f(x_i | \theta)$$

Likelihood

- Imagine the situation that a ball is drawn from the bag consisting of three blue balls and five white balls with replacement
 - Drawing is repeated five times and output is color of ball

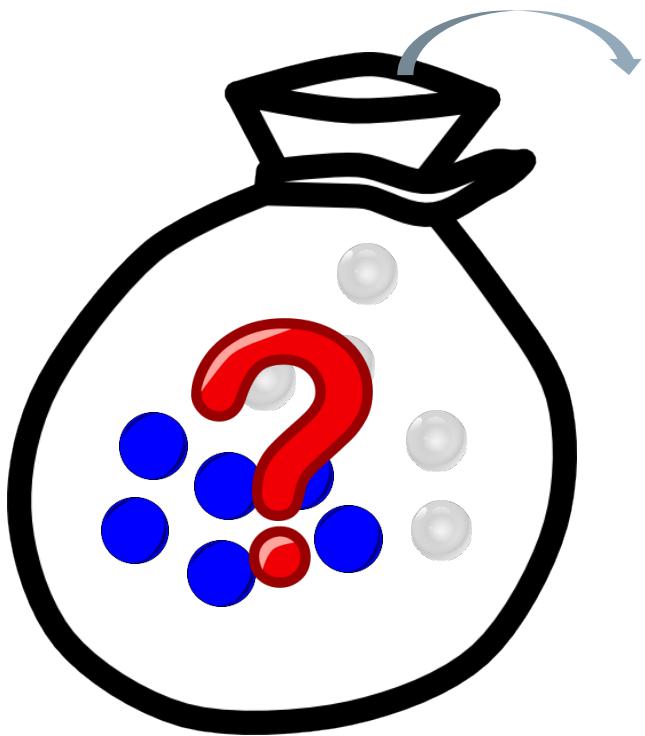
	1	2	3	4	5
Case 1	blue	white	blue	white	white
Case 2	blue	blue	blue	blue	blue

Which case is more probable?

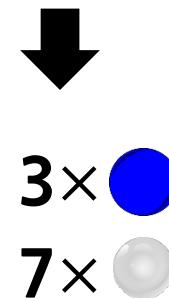


Likelihood represents how much probable is observed data samples given statistical model

Example of Likelihood Function



Sampling with replacement



- Want to estimate p_{blue} and p_{white} based on the sampling result

Example of Likelihood Function

- There are only two outputs → Bernoulli distribution
- Bernoulli distribution: the probability distribution of a random variable which takes the value 1 with success probability of p and the value 0 with failure probability of $q = 1 - p$
 - For random variable following Bernoulli distribution,
$$p(X = 1) = 1 - p(X = 0) = p = 1 - q$$
 - Probability mass function over possible outcomes y

$$f(y; p) = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{if } y = 0 \end{cases}$$

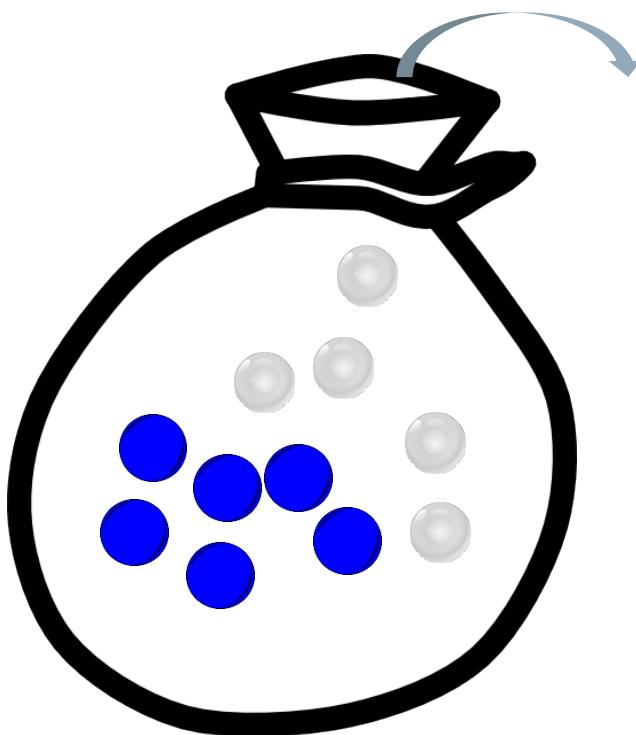
- This can also be expressed as

$$f(y; p) = p^y(1 - p)^{1-y} \quad \text{for } y \in \{1, 0\}$$

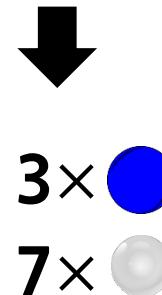
- For Bernoulli distribution, p is θ
 - In this example, assume that blue ball is 1

$$\begin{aligned} p &= p_{blue} \\ 1 - p &= p_{white} \end{aligned}$$

Example of Likelihood Function



Sampling with replacement



- Likelihood function

- If blue ball, $f(1; p) = p$
 - If white ball, $f(0; p) = 1 - p$

$$\mathcal{L} = \prod_{i=1}^{10} f(y_i; p) = p^3(1-p)^7$$

- Maximize \mathcal{L} with respect to p

$$\arg \max_y p^3(1-p)^7$$

$$\Rightarrow \arg \max_y \ln(p^3(1-p)^7) \\ = 3\ln p + 7\ln(1-p)$$

$$\frac{d \ln \mathcal{L}}{dp} = 0, \quad \hat{p} = \frac{3}{10}$$

Example of Likelihood Function

- 1D data samples from Gaussian distribution with $\sigma = 1$

$$f(x; \theta) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\theta)^2}{2}}$$

	1	2	3	4	5
x	2.61	3.73	2.80	4.29	3.12

- Likelihood function is function of parameter θ

$$\mathcal{L}(\theta; \mathbf{x}) = \prod_{i=1}^5 \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i-\theta)^2}{2}}$$

- If $\theta = 2$, $\mathcal{L}(2) \approx 0.33 \times 0.09 \times 0.29 \times 0.03 \times 0.21 = 0.0000542619$

	1	2	3	4	5
x	2.61	3.73	2.80	4.29	3.12
$f(x; \theta)$	0.33	0.09	0.29	0.03	0.21

- Maximum likelihood estimation is method to find parameter to maximize likelihood function with given data samples

Maximum Likelihood Estimation

- Compare likelihood with different parameters
 - ▣ If $\theta = 2, \mathcal{L}(2) \approx 0.33 \times 0.09 \times 0.29 \times 0.03 \times 0.21 = 0.0000542619$

	1	2	3	4	5
x	2.61	3.73	2.80	4.29	3.12
$f(x; \theta)$	0.33	0.09	0.29	0.03	0.21

- ▣ If $\theta = 3, \mathcal{L}(3) \approx 0.37 \times 0.31 \times 0.39 \times 0.17 \times 0.40 = 0.003041844$

	1	2	3	4	5
x	2.61	3.73	2.80	4.29	3.12
$f(x; \theta)$	0.37	0.31	0.39	0.17	0.40

- ▣ If $\theta = 4, \mathcal{L}(4) \approx 0.15 \times 0.38 \times 0.19 \times 0.38 \times 0.27 = 0.001111158$

	1	2	3	4	5
x	2.61	3.73	2.80	4.29	3.12
$f(x; \theta)$	0.15	0.38	0.19	0.38	0.27

Solve Optimization Problem

- Likelihood function

$$\begin{aligned}\mathcal{L}(\theta; \mathbf{x}) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}} e^{-\frac{(x_i-\theta)^2}{2}} \\ &= \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(-\frac{\sum_{i=1}^n (x_i-\theta)^2}{2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi}}\right)^n \exp\left(-\frac{\sum_{i=1}^n (\theta^2 - 2x_i\theta + x_i^2)}{2}\right) \\ &\propto \exp\left(-\sum_{i=1}^n (\theta^2 - 2x_i\theta + x_i^2)\right)\end{aligned}$$

- When $\sum_{i=1}^n (\theta^2 - 2x_i\theta + x_i^2)$ is minimum, $\mathcal{L}(\theta; \mathbf{x})$ is maximized

$$\frac{d}{d\theta} n\theta^2 - 2(\sum_{i=1}^n x_i)\theta + \sum_{i=1}^n x_i^2 = 2n\theta - 2\sum x_i = 0. \quad \theta = \frac{\sum x_i}{n} = \bar{x}$$

- Second order equation of $\theta \rightarrow$ There is a solution to minimize equation
- Example
 - <https://www.geogebra.org/m/zOmGcvXq>

※ Gaussian (Normal) Distribution

- The Gaussian distribution is a continuous probability distribution
 - ▣ probability density function

$$\mathcal{N}(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- μ : mean or expectation of the distribution
- σ : standard deviation

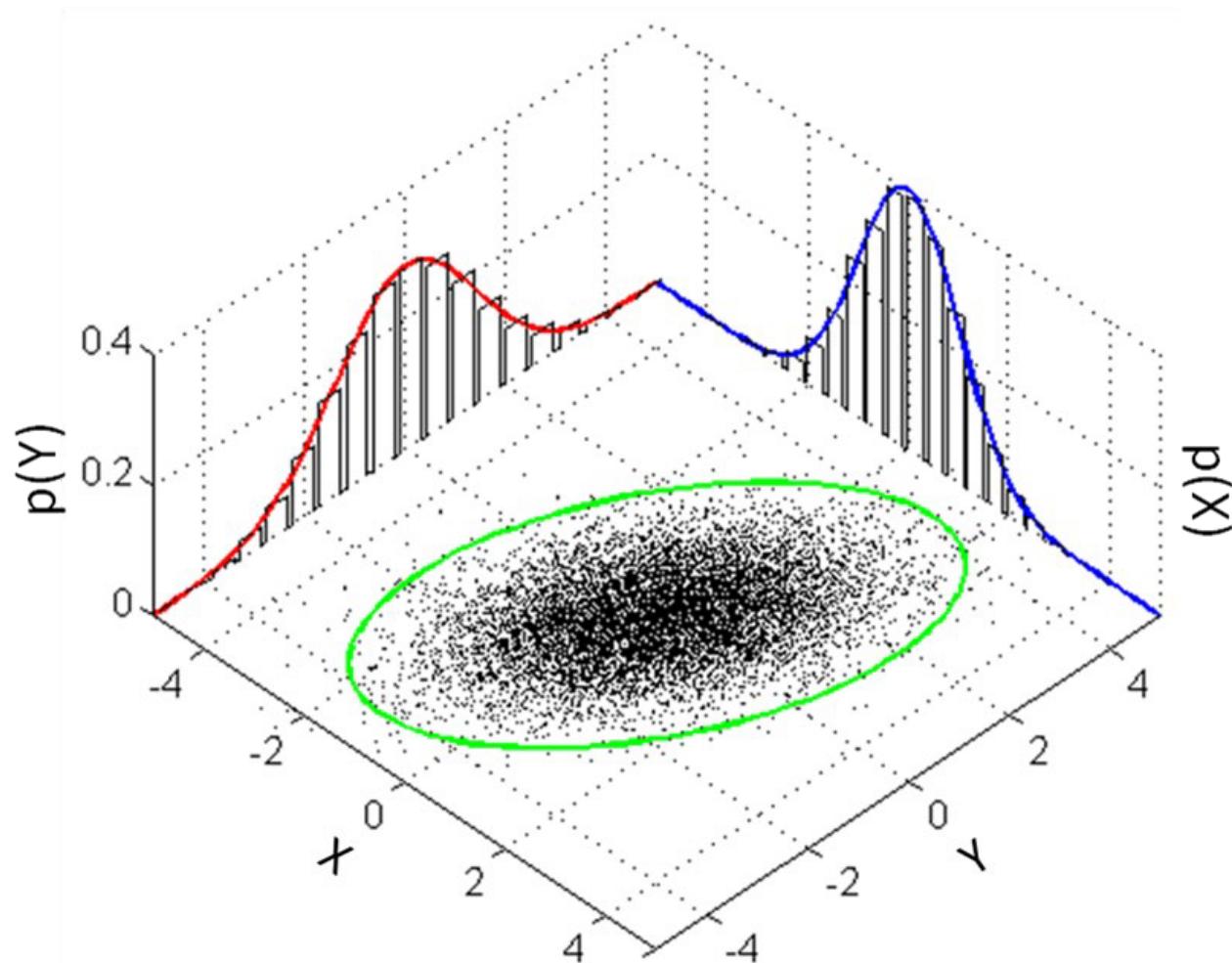
- ▣ When $\mu = 0$ and $\sigma = 1$, the distribution is called the standard normal distribution
- Multivariate normal distribution is a generalization of the 1D normal distribution
 - ▣ probability density function

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \left(\frac{1}{(2\pi)^p |\boldsymbol{\Sigma}|} \right)^{1/2} e^{-\frac{1}{2} (\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

- p : dimensionality
- $\boldsymbol{\mu}$: mean vector
- $\boldsymbol{\Sigma}$: covariance matrix

※ Gaussian (Normal) Distribution

- Two dimensional normal distribution



How to Find Parameters for Logistic Regression?

- Output is 0 or 1 → Output follows Bernoulli distribution with parameter p
- Each sample has different p depending on input
 $y_i \sim \text{Bernoulli}(P_i)$
 - ▣ P_i is the probability that output value is 1
 - ▣ Assume that all samples from the same Bernoulli distribution

$$f(y_i) = P\{Y = y_i\} = P_i^{y_i} (1 - P_i)^{1-y_i}, \quad y_i \in \{0,1\}$$

- Likelihood function of logistic regression model

$$\mathcal{L} = \prod_{i=1}^n f(y_i) = \prod_{i=1}^n P_i^{y_i} (1 - P_i)^{1-y_i}$$

$$\square P_i = \frac{1}{1 + e^{-\beta_0 - \beta_1 x_1 - \beta_2 x_2 - \dots - \beta_p x_p}}$$

Issues with Using the Likelihood Function Directly

1. Numerical Stability
 - ▣ The likelihood function involves a product of probabilities, which can become extremely small for large datasets
 - ▣ Floating-point precision issues can arise, leading to numerical underflow
2. Computational Simplicity
 - ▣ The product operation in $\mathcal{L}(\theta)$ makes differentiation complex when optimizing θ .

Log-Likelihood Function

- Likelihood function

$$\mathcal{L}(\theta; x_1, x_2, \dots, x_n) = f(x_1, x_2, \dots, x_n | \theta) = \prod_{i=1}^n f(x_i | \theta)$$

- Log-likelihood function

$$\log \mathcal{L}(\theta) = \log \prod_{i=1}^n f(x_i | \theta) = \sum_{i=1}^n \log f(x_i | \theta)$$

- Log-likelihood function for logistic regression

$$\log \mathcal{L} = \sum_{i=1}^n y_i \log P_i + \sum_{i=1}^n (1 - y_i) \log(1 - P_i)$$

- Find parameters $\beta_0, \beta_1, \dots, \beta_p$ to maximize $\log \mathcal{L}$

Advantages of Log-Likelihood

1. **Avoids numerical underflow**

- Logarithm converts products into sums, keeping values within a manageable range.

2. **Easier differentiation**

$$\frac{d}{dt} f(t)$$

- Differentiating a sum is simpler than differentiating a product, making gradient-based optimization easier.

3. **Log-convexity and optimization benefits**

- Many likelihood functions become convex in log-space, simplifying optimization.

Odds and Odds Ratio

- Odds reflect the likelihood that the event will take place
 - In gambling, odds represent the ratio between the amounts staked by parties to a wager or bet
 - In logistic regression, odds represent the ratio between $P(y = 1)$ and $P(y = 0)$

$$\frac{P(Wins)}{P(Losses)}$$

$$Odds = \frac{P}{1-P}$$

$\cancel{P \rightarrow 1} \quad Odds = \infty, \quad \cancel{P \rightarrow 0} \quad Odds = 0$

$$\text{odds} = \frac{P(y = 1)}{P(y = 0)} = \frac{\frac{1}{1 + e^{-\beta_0 - \beta_1 x_1 - \beta_2 x_2 - \dots - \beta_p x_p}}}{1 - \frac{1}{1 + e^{-\beta_0 - \beta_1 x_1 - \beta_2 x_2 - \dots - \beta_p x_p}}} = \exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p)$$

- Odds ratio is the ratio between odds when unit increment of a variable

$$\text{odds ratio} = \frac{\text{odds when input is } x_1 = x + 1}{\text{odds when input is } x_1 = x} = \frac{\exp(\beta_0 + \beta_1(x + 1) + \beta_2 x_2 + \dots + \beta_p x_p)}{\exp(\beta_0 + \beta_1 x + \beta_2 x_2 + \dots + \beta_p x_p)} = e^{\beta_1}$$

- Odd increases e^{β_1} times for every 1-unit increase in x_1

Logistic Regression: Odds

- A logistic model is one where the log-odds of the probability of an event is a linear combination of independent or predictor variables (binary case)
- Logistic model

$$\ln(\text{odds}) = \ln\left(\frac{P(y = 1)}{P(y = 0)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

- Let $P = P(y = 1)$

$$\frac{P(y = 1)}{P(y = 0)} = \frac{P}{1 - P}$$

$$g(P) = \ln\left(\frac{P}{1 - P}\right) = \underbrace{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p}_{\text{linear predictor}}$$

Link function

Logistic Regression

- Link function
 - ▣ A link function connects the linear predictor (i.e., a linear combination of input variables) to the expected value of the dependent variable
 - ▣ The choice of the link function determines how we transform probabilities into a form suitable for regression modeling

Logistic Regression: Link Function

□ Logit function

- ❑ The logit function is the **most commonly used link function** in logistic regression
- ❑ It transforms probabilities $P(P = P(Y = 1|X))$ into log-odds:

$$g(P) = \log\left(\frac{P}{1 - P}\right)$$

- ❑ It ensures symmetry and interpretability in terms of odds
- ❑ Logistic regression with this function estimates the log-odds as a linear function of predictors

Logistic Regression: Link Function

symmetric ✓

□ Probit function

input $[0,1]$ output $(-\infty, \infty)$

- The probit function uses the inverse cumulative distribution function (CDF) of the standard normal distribution

$$g(P) = \Phi^{-1}(P)$$

where Φ^{-1} is the inverse of the standard normal CDF

- It is similar to the logit function but assumes a normal distribution of the underlying latent variable
- Probit regression is used in **econometrics and psychometrics** when normality assumptions are more appropriate

Logistic Regression: Link Function

ASYMMETRY

- Complementary Log-Log (cloglog) function
 - ▣ The cloglog function is asymmetric and models extreme event probabilities
$$g(P) = \log(-\log(1 - P))$$
 - ▣ It is useful when **probabilities near 1 are more common than those near 0** (e.g., survival analysis, infectious disease modeling)
 - ▣ Unlike the logit and probit functions, it does not have symmetry around $P = 0.5$
- Log-Log and Negative Log-Log functions
 - ▣ Log-log function
$$g(P) = -\log(-\log P)$$
 - Used when small probabilities dominate the data
 - ▣ Negative log-log function
$$g(P) = \log(-\log P)$$
 - Similar to cloglog but in the opposite direction

Logistic Regression for Multi-class Classification

- Logistic function only can be used in binary classification
- For K classes, $P(y_i = k)$ is the probability that i th data point belong to class k ($k \in \{1,2,3, \dots, K\}$)
 - It is reasonable to select class k whose probability is the highest

Multinomial Logistic Regression

- Multinomial logistic regression assumes that log ratio between probabilities of two different classes is linear
 - ▣ Log linear model

$$\ln p(y_i = 1) = \boldsymbol{\beta}_1 \cdot \mathbf{x}_i - \ln Z$$

$$\ln p(y_i = 2) = \boldsymbol{\beta}_2 \cdot \mathbf{x}_i - \ln Z$$

⋮

$$\ln p(y_i = K) = \boldsymbol{\beta}_K \cdot \mathbf{x}_i - \ln Z$$

- $\mathbf{x}_i = (1, x_{i1}, x_{i2}, \dots, x_{ip})$

- $\boldsymbol{\beta}_k = (\beta_{k0}, \beta_{k1}, \beta_{k2}, \dots, \beta_{kp})$

- $\boldsymbol{\beta}_k \cdot \mathbf{x}_i = \beta_{k0} + \beta_{k1}x_{i1} + \dots + \beta_{kp}x_{ip}$



$$\mathcal{Z} \left(p(y_i = k) = \frac{1}{Z} e^{\boldsymbol{\beta}_k \cdot \mathbf{x}_i} \right) = 1.$$

$$Z = \sum_{k=1}^K e^{\boldsymbol{\beta}_k \cdot \mathbf{x}_i}$$

※ Multinomial distribution

- Multinomial distribution is a generalization of the binomial distribution

- Binomial distribution with parameters n and p is the discrete probability distribution of the number of successes in a sequence of n independent yes/no experiments with success probability p

$$p(k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} = {}^n C_k p^k (1-p)^{n-k}$$

- Example of binomial distribution is the distribution of the number of head when flipping a coin n times (in this case, $p = 0.5$)

- Probability that k times head occur among n trials

$$p(k) = \frac{n!}{k!(n-k)!} 0.5^k 0.5^{n-k} = \frac{n!}{k!(n-k)!} 0.5^n$$

- In multinomial distribution, possible outcome is more than two and each outcome has its own probability to occur, (p_1, \dots, p_d)

- $p_1 + \dots + p_d = 1$
 - d is the number of possible outcomes
 - $n_x = \sum_{i=1}^d x_i$

$$p(\mathbf{x} = (x_1, x_2, \dots, x_d)) = \frac{n_{\mathbf{x}}!}{x_1! \cdots x_d!} p_1^{x_1} \cdots p_d^{x_d}$$

Likelihood Function

- Likelihood function

$$\mathcal{L} = \prod_{i=1}^n \prod_{k=1}^K P_{ik}^{v_{ik}} , \quad v_{ik} = \begin{cases} 1, & y_i = k \\ 0, & y_i \neq k \end{cases}$$

- $P_{ik} = p(y_i = k)$

- Log likelihood function

$$\log \mathcal{L} = \sum_{i=1}^n \sum_{k=1}^K v_{ik} \log P_{ik}$$

- Through maximum likelihood estimation, determine β_k as the same as in binary logistic regression

Multiclass Classification Using Binary Classifiers

- There are other ways to get multiclass classifiers by combining binary classifiers
 - For multiclass classification commonly used approach is to construct K separate binary classifiers $K > H \text{ class} \Rightarrow K \text{ 개 } \text{이진 분류기 } \text{생성}$
 - Each model is trained using the data from class C_k as the positive examples and the data from the remaining $K - 1$ classes as the negative examples

$$y(\mathbf{x}) = \max_k y_k(\mathbf{x})$$

→ One-versus-the rest approach

각 클래스에 대한 분류기

각각 데이터에 대해

각각 클래스에 대한 학습

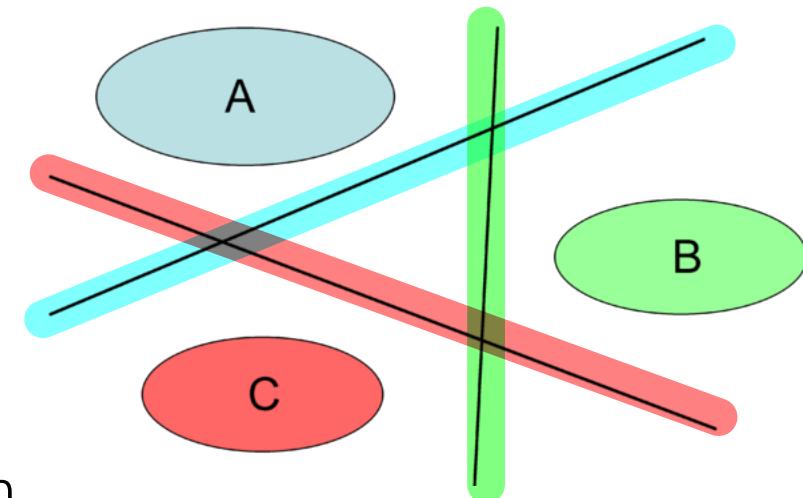
각각 확률(가능성)을 계산

결국은 분류기의 확률로 표시

(A) vs (B C)

(B) vs (A C)

(C) vs (A B)



- Problems of one-versus-the rest approach

- Because each classifier was trained on different task, there is no guarantee that the real-values quantities $y_k(\mathbf{x})$ will have appropriate scales
- Imbalance of data on training

한 클래스는 적고 나머지는 많아서 모델 편향

각 분류기 고유 기준으로 학습. 출력값 스케일 차이
최적화 간 차이 비교 어렵음

Multiclass Classification Using Binary Classifiers

- Another approach is to train $K(K - 1)/2$ different 2-class classifiers on all possible pairs of classes

kC_2

A-B A-C B-C

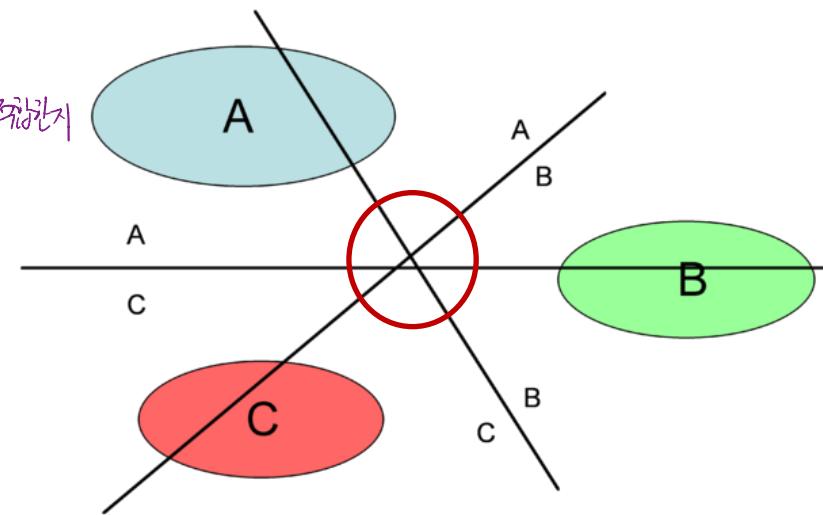
- Classify test points according to which class has the highest number of votes

→ one-versus-one approach

$$\begin{array}{ccc} A - B & A - C & B - C \\ \downarrow & \downarrow & \downarrow \\ ? & ? & ? \end{array}$$

vote

두 클래스 중 어떤 클래스에 더 적합한지
판정하여 그 결과를 반환.
가장 많은 득표 클래스로 판정



- Problems of one-versus-one approach

- It can lead to ambiguities in the resulting classification
- For large K , it requires significantly more training time

NAÏVE BAYES CLASSIFIER

Week07

Naïve Bayes Classifier

Review: Types of Classifiers

- A classifier is a function that assigns to a sample, \mathbf{x} a class label \hat{y}
$$\hat{y} = f(\mathbf{x})$$
- A probabilistic classifier obtains conditional distributions $\Pr(Y|\mathbf{x})$, meaning that for a given $\mathbf{x} \in \mathcal{X}$, they assign probabilities to all $y \in Y$
 - Hard classification
$$\hat{y} = \arg \max_y \Pr(Y = y | \mathbf{x})$$
- Logistic regression

$$f(x) = P(Y = 1 | \mathbf{x}) = \frac{1}{1 + e^{-\beta_0 - \beta_1 x_1 - \beta_2 x_2 - \dots - \beta_p x_p}}$$

Any other ways to model $P(Y|X)$?

Naïve Bayes Classifier

- Naïve Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong independence assumption between features

- **Bayes' theorem**

$$P(A|B) = \frac{\text{prior} \ P(A) \text{ likelihood} \ P(B|A)}{P(B)}$$

- A and B are events
- $P(A)$ and $P(B)$ are the probabilities of A and B without regard to each other
- $P(A|B)$, a conditional probability, is the probability of A given that B is true
- $P(B|A)$, is the probability of B given that A is true



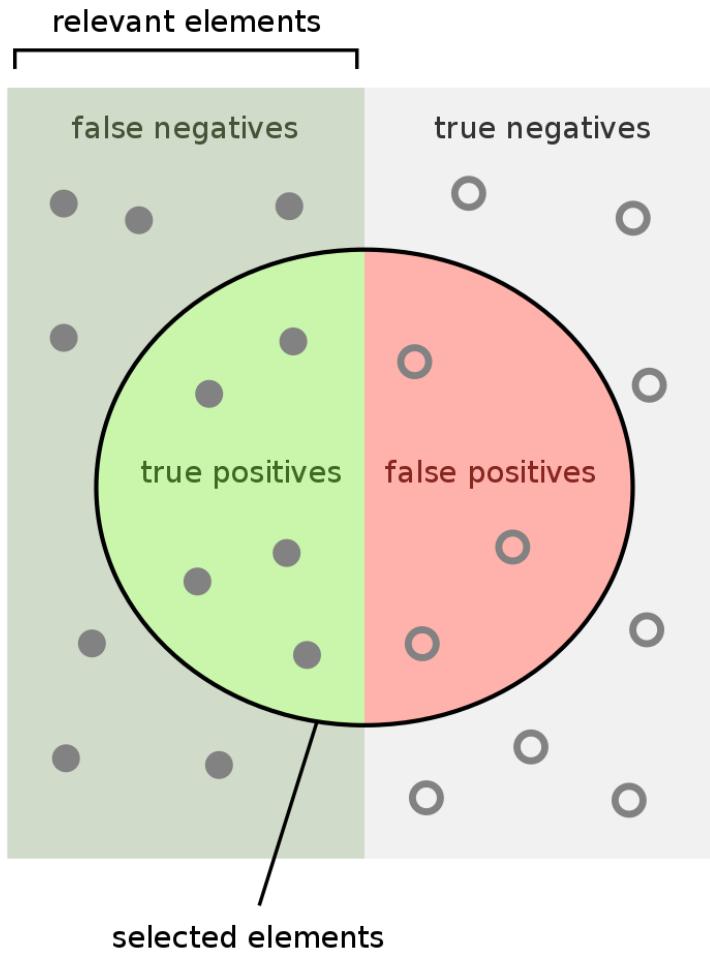
Example of Bayes' Theorem

- Suppose a drug test is 99% sensitive and 99% specific
 - ▣ 99% sensitive=99% true positive over real positive $P(\text{positive} | \text{User}) = 0.99$
 - ▣ 99% specific=99% true negative over real negative $P(\text{negative} | \text{Not-user}) = 0.99$

Real Decision	Positive	Negative
Positive	True positive	False positive (Type I error)
Negative	False negative (Type II error)	True negative

- Suppose that 0.5% of people are users of the drug

* Sensitivity and Specificity



How many relevant items are selected?
e.g. How many sick people are correctly identified as having the condition.

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

실제 양성 중
양성이 맞았는가.

How many negative selected elements are truly negative?
e.g. How many healthy people are identified as not having the condition.

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Negative}}$$

실제 음성 중
음성이 맞았는가.

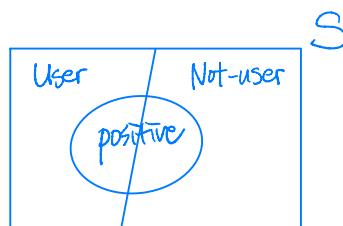
Example of Bayes' Theorem

- If a randomly selected individual tests positive, what is the probability he or she is a user of drug?

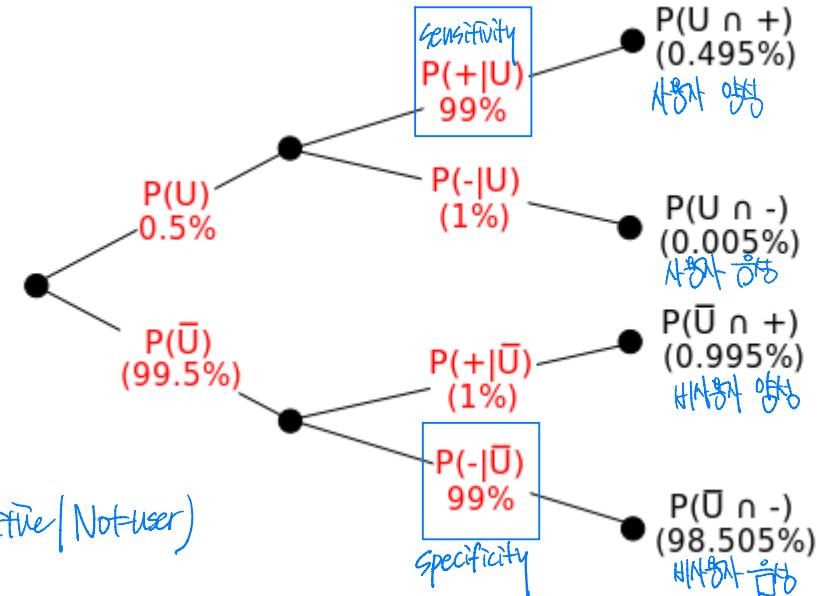
- This problem is to calculate $P(U|+)$

+ means positive drug test

U represents user, \bar{U} represents non-user



$$P(\text{positive}) = P(\text{User}) P(\text{positive} | \text{User}) + P(\text{Not-user}) P(\text{positive} | \text{Not-user})$$



$$\begin{aligned}
 P(U|+) &= \frac{P(U)P(+|U)}{P(+)} = \frac{P(U)P(+|U)}{P(U)P(+|U) + P(\bar{U})P(+|\bar{U})} \\
 &= \frac{0.005 \times 0.99}{0.005 \times 0.99 + 0.995 \times 0.01} \approx 33.2\%
 \end{aligned}$$

law of total probability

$$\therefore P(A) = \sum_{b \in B} P(b)P(A|b)$$

Other example :

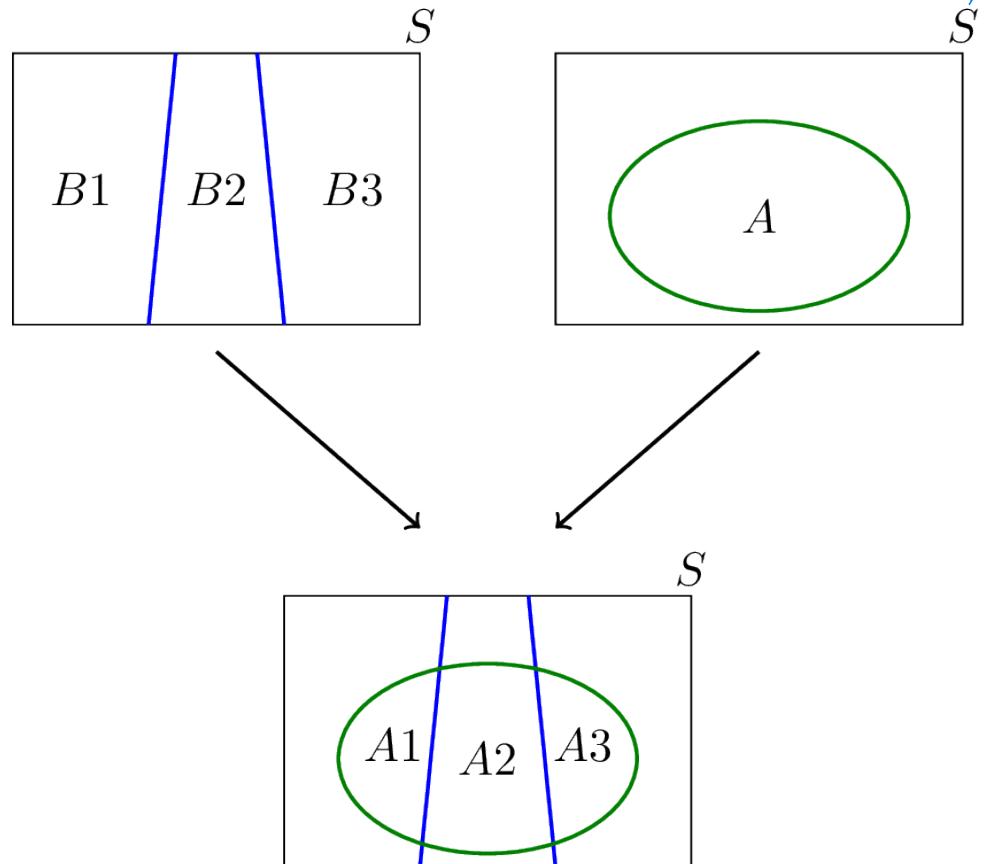
$$\left. \begin{array}{l} 95\% \text{ SENSITIVE} \\ 90\% \text{ SPECIFIC} \\ 1\% \text{ USER} \end{array} \right\} P(\text{User} | \text{positive}) = \frac{P(\text{User}) P(\text{positive} | \text{user})}{P(\text{positive})} = \frac{0.01 \times 0.95}{0.01 \times 0.95 + 0.99 \times 0.10} \approx 0.0876, 8.76\%$$

0.0876은 8.76% 만이 실제 사용자

$$\left. \begin{array}{l} P(\text{User}) \\ = 0.01 \\ P(\text{Non-user}) \\ = 0.99 \end{array} \right\} \begin{array}{l} P(\text{Positive} | \text{user}) \\ = 0.95 \\ P(\text{Negative} | \text{user}) \\ = 0.05 \end{array}$$
$$P(\text{positive} | \text{Non-user}) = 0.10$$
$$P(\text{Negative} | \text{Non-user}) = 0.90$$

※ Law of Total Probability

$$P(A) = \sum_{b \in B} P(b)P(A|b) = P(B_1)P(A|B_1) + P(B_2)P(A|B_2) + P(B_3)P(A|B_3)$$



Naïve Bayes Classifier

- Conditional probability model for Naïve Bayes classifier
 - ▣ Naïve Bayes classifier calculates following probability for every class

$$p(C_k | x_1, \dots, x_p) = p(C_k | \mathbf{x})$$

- x_i represents each feature (independent variable)
- k represents k -th class and classifier assigns output class with the maximum probability

- ▣ Re-formulation using Bayes' theorem

$$p(C_k | \mathbf{x}) = \frac{p(C_k) p(\mathbf{x} | C_k)}{p(\mathbf{x})}$$

Handwritten annotations in blue:

- Top right: C_k 의 확률
- Top right: C_k 의 확률
- Middle left: C_k 의 확률
- Middle left: \mathbf{x} 의 확률
- Bottom left: C_k 의 확률
- Bottom left: \mathbf{x} 의 확률
- Middle right: $p(C_k)$ prior
- Middle right: $p(\mathbf{x} | C_k)$ likelihood
- Bottom right: $p(\mathbf{x})$ evidence

- This equation is also written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

Naïve Bayes Classifier

- Naïve Bayes Classifier

$$p(C_k|\mathbf{x}) = \frac{p(C_k)p(\mathbf{x}|C_k)}{p(\mathbf{x})}$$

- ▣ Denominator $p(\mathbf{x})$ does not depend on class 가장 학점이 높은 출판사가!

$$\operatorname{argmax}_k p(C_k|\mathbf{x}) = \operatorname{argmax}_k p(C_k)p(\mathbf{x}|C_k)$$

- $p(C_k)p(\mathbf{x}|C_k)$ is equivalent to the joint probability $p(C_k, x_1, \dots, x_p)$

- ▣ Using chain rule $p(C_k, x_1, \dots, x_p)$ can be written as follows

$$\begin{aligned} p(C_k, x_1, \dots, x_p) &= p(C_k)p(x_1, \dots, x_p|C_k) = p(C_k)p(x_1|C_k)p(x_2, \dots, x_p|C_k, x_1) \\ &= p(C_k)p(x_1|C_k)p(x_2|C_k, x_1) \cdots p(x_p|C_k, x_1, \dots, x_{p-1}) \end{aligned}$$

- ▣ Naïve Bayes classifier assumes conditional independence of each feature

$$\begin{aligned} p(x_i|C_k, x_j) &= p(x_i|C_k) \\ p(x_i|C_k, x_j, x_l) &= p(x_i|C_k) \end{aligned}$$

※ Chain Rule

- Chain rule permits the calculation of any member of the joint distribution of a set of random variables using only conditional probabilities

$$P(A_n, \dots, A_1) = P(A_n | A_{n-1}, \dots, A_1) \cdot P(A_{n-1}, \dots, A_1)$$

- Repeating this process with each final term creates the product

$$P\left(\bigcap_{k=1}^n A_k\right) = \prod_{k=1}^n P\left(A_k \mid \bigcap_{j=1}^{k-1} A_j\right)$$

Naïve Bayes Classifier

□ Naïve Bayes Classifier

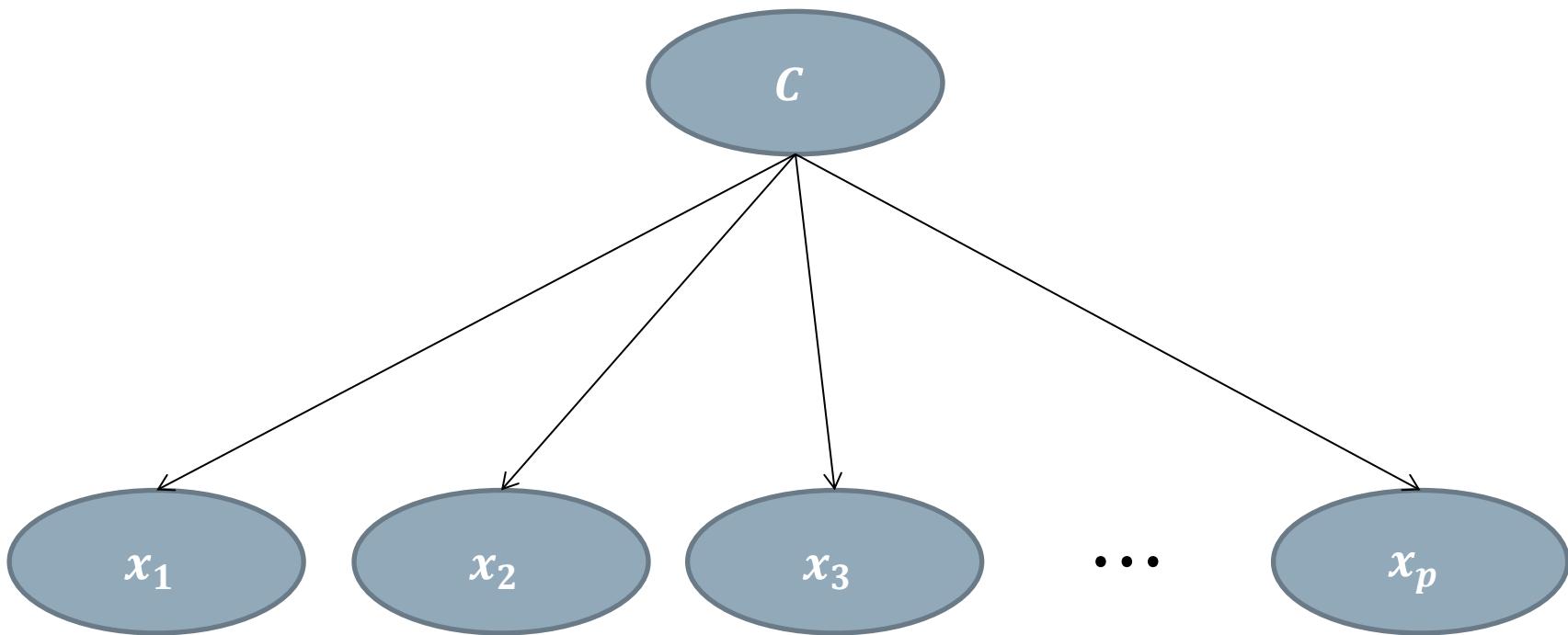
$$\begin{aligned} & p(C_k)p(x_1|C_k)p(x_2|C_k, x_1) \cdots p(x_p|C_k, x_1, \dots, x_{p-1}) \\ &= p(C_k)p(x_1|C_k)p(x_2|C_k) \cdots p(x_p|C_k) = p(C_k) \prod_{i=1}^p p(x_i|C_k) \\ &\therefore p(C_k)p(\mathbf{x}|C_k) = p(C_k) \prod_{i=1}^p p(x_i|C_k) \end{aligned}$$

□ Decision function of Naïve Bayes classifier

$$\hat{y} = \operatorname{argmax}_{k \in \{1, \dots, K\}} p(C_k) \prod_{i=1}^p p(x_i|C_k)$$

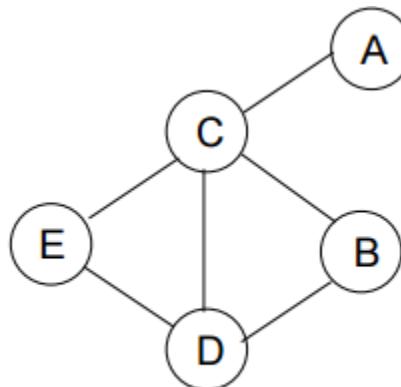
Naïve Bayes Classifier

- Graphical model representation of Naïve Bayes

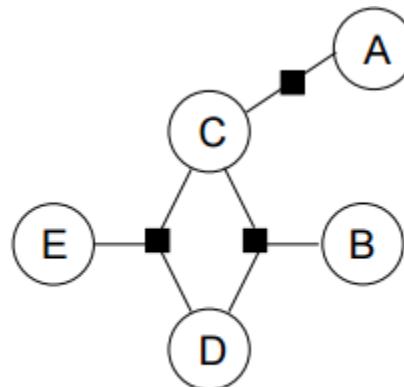


* Graphical Model

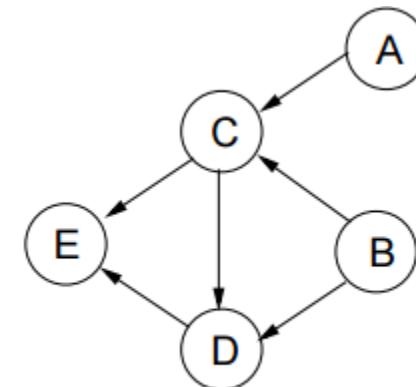
- In graphical model, each node represents a random variable and the edge express probabilistic relationships between these variables



Undirected Graph



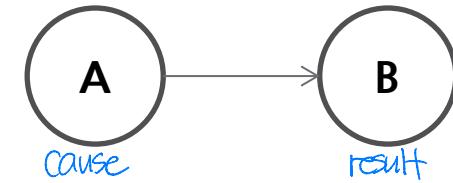
Factor Graph



Bayesian Network

- Bayesian network is a specific form of graphical model
 - Directed acyclic graphs
 - If two nodes A and B are connected by edge and direction of edge heads to B, it means that the state of A affects on probability of B

$$p(B, A) \neq p(B)p(A)$$
$$p(B|A) \neq p(B)$$



$A \& B \Rightarrow$ not independent

Naïve Bayes Classifier: Advantages and Limitation

- Advantages of Naïve Bayes
 - ▣ Simple and fast to train
 - ▣ Works well with small datasets
 - ▣ Handles missing data well
 - ▣ Requires less training data compared to other classifiers

- Limitations of Naïve Bayes
 - ▣ Independence assumption is often unrealistic
 - ▣ Struggles with correlated features
 - ▣ Zero probability problem (solved using Laplace smoothing)

Naïve Bayes Classifier: Parameter Estimation

- Parameter estimation and event models
 - ▣ A class' prior setting: by assuming equiprobable classes ($p(C_k) = 1/K$) or by calculating an estimate for the class probability from the training set ($p(C_k) = n_k/n$)
 - ▣ Select appropriate probability distribution for $p(x_i|C_k)$
 - For continuous variables, Gaussian distribution is the common choice
 - For discrete variables, multinomial distribution is the common choice
 - ▣ After setting, probabilistic model for naïve Bayes classifier, parameters of distributions are estimated using training data
 - Calculate $\tilde{p}(C_{y_j}|\mathbf{x}) = (C_{y_j}) \prod_{i=1}^p p(x_i|C_{y_j})$ for j -th sample
 - Calculate $\tilde{p}(\mathbf{C}|\mathbf{X}) = \prod_{j=1}^n \tilde{p}(C_{y_j}|\mathbf{x})$ and maximize this probability

$\frac{Y}{2}$	$\frac{E}{2}$	$\frac{N}{2}$
Sunny	Hot	N
S	H	N
Cloud	H	Y
Rain	Low	Y
R	Mid	Y
R	H	N
C	M	Y
S	L	Y

$$P(\frac{E}{2} = Y | R, H) = P(Y) \cdot P(R|Y) \cdot P(H|Y) = \frac{5}{8} \times \frac{2}{5} \times \frac{1}{5} = \frac{1}{20} = 0.05$$

$$P(\frac{E}{2} = N | R, H) = P(N) \cdot P(R|N) \cdot P(H|N) = \frac{3}{8} \times \frac{1}{3} \times \frac{2}{3} = \frac{1}{8} = 0.125$$

$$P(Y) = \frac{5}{8} \quad P(R|Y) = \frac{2}{5} \quad P(H|Y) = \frac{1}{5}$$

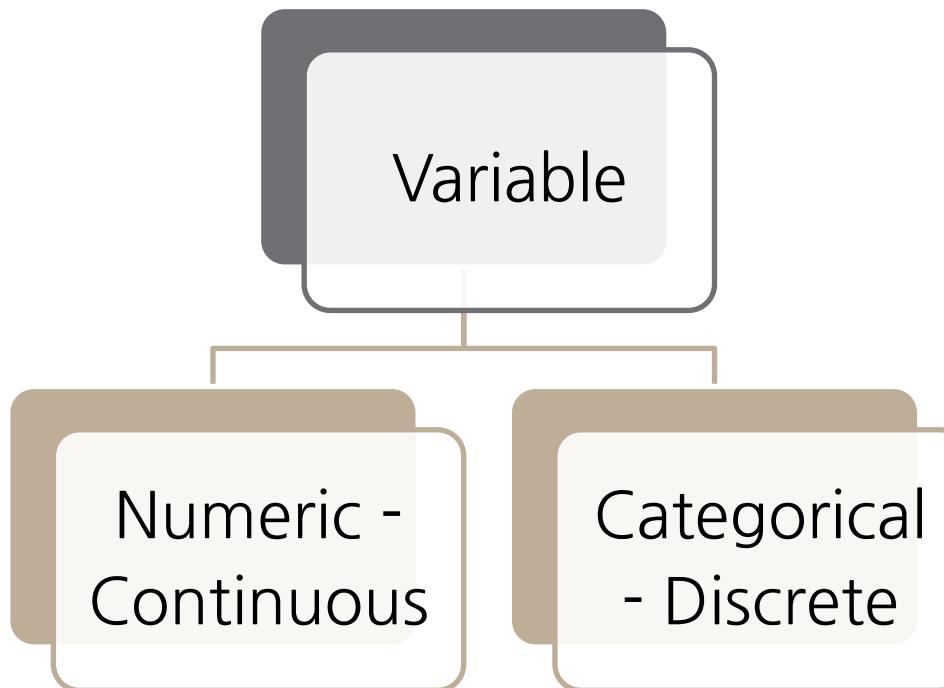
$$P(N) = \frac{3}{8} \quad P(R|N) = \frac{1}{3} \quad P(H|N) = \frac{2}{3}$$

$$P(Y|R, H) = 0.05 < P(N|R, H) = 0.125$$

\therefore H가 더우면 온도 X

Naïve Bayes Classifier

- Determine $p(x_i|C_k)$
 - ▣ The probability functions depend on the type of variables
 - Probability distributions for discrete random variables: Binomial, Multinomial, Geometric, ...
 - Probability distributions for continuous random variables: Gaussian(Normal), χ^2 , beta, F , t , ...





Bernoulli Naïve Bayes

- Bernoulli naïve Bayes
 - ▣ In the multivariate Bernoulli event model, features are independent booleans (binary variables) describing inputs
 - Ex) Each variable only takes values 0 or 1
 - ▣ Each x_i is a boolean expressing the occurrence of event

$$p(\mathbf{x}|C_k) = \prod_{i=1}^d p_{ki}^{x_i} (1 - p_{ki})^{1-x_i}$$

- d is the number of input features $x_i \in \{0, 1\}$
- p_{ki} is the probability that x_i is 1(true) for class k

※ Bernoulli distribution

- The probability distribution of a random variable which takes the value 1 with success probability of p and the value 0 with failure probability of $q = 1 - p$

- For random variable following Bernoulli distribution,

$$p(X = 1) = 1 - p(X = 0) = p = 1 - q$$

- Probability mass function over possible outcomes y

$$f(y; p) = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{if } y = 0 \end{cases}$$

- This can also be expressed as

$$f(y; p) = p^y(1 - p)^{1-y} \quad \text{for } y \in \{1, 0\}$$

- Expected value of a Bernoulli random variable X

$$\mathbb{E}[X] = p$$

- Variance of a Bernoulli random variable

$$\text{Var}[X] = p(1 - p)$$

Bernoulli Naïve Bayes: Estimation of Parameters

- Maximum likelihood estimates for Bernoulli Naïve Bayes model
 - ▣ Each input variable can take values 0 or 1
 - ▣ There exist n samples with d features
 - ▣ Prior probability $p(C_k)$

$$p(C_k) = \frac{n_k}{n}$$

- n_k is the number of samples that y_i belongs to class k
- ▣ Likelihood $p(\mathbf{x}|C_k)$

$$p(\mathbf{x}|C_k) = \prod_{i=1}^d p_{ki}^{x_i} (1 - p_{ki})^{1-x_i}$$

- ▣ Posterior probability $p(C_k|\mathbf{x})$

$$p(C_k|\mathbf{x}) \propto p(C_k)p(\mathbf{x}|C_k)$$

$$\frac{n_k}{n} \times \prod_{i=1}^d p_{ki}^{x_i} (1 - p_{ki})^{1-x_i}$$

Bernoulli Naïve Bayes: Estimation of Parameters

- Maximum likelihood estimates for Bernoulli Naïve Bayes model

$$L = \prod_{j=1}^n p(C_{y_j}) p(\mathbf{x}_j | C_{y_j}) = \prod_{j=1}^n \frac{n_{y_j}}{n} \left(\prod_{i=1}^d p_{y_j i}^{x_{ji}} (1 - p_{y_j i})^{1-x_{ji}} \right)$$

$$\log L = \sum_{j=1}^n \log \frac{n_{y_j}}{n} + \sum_{j=1}^n \sum_{i=1}^d (x_{ji} \log p_{y_j i} + (1 - x_{ji}) \log (1 - p_{y_j i}))$$

- n is the total number of data samples
- n_{y_j} is the number of data samples belong to class y_j ($y_j \in \{1, 2, \dots, k\}$)
- x_{ji} is the i -th input variable's value for j -th data sample

- Parameters to be estimates

- For each class k , probability to occur 1 for each feature i , p_{ki}

Bernoulli Naïve Bayes: Estimation of Parameters

- Maximum likelihood estimates for Bernoulli Naïve Bayes model
 - ▣ To obtain optimal p_{ki} , set $\frac{\partial \log L}{\partial p_{ki}} = 0$

$$\begin{aligned}\frac{\partial \log L}{\partial p_{ki}} &= \sum_{j \in \{m: y_m=k\}} \left\{ \frac{x_{ji}}{p_{ki}} - \frac{1-x_{ji}}{1-p_{ki}} \right\} \\ &= \frac{n_{k1}}{p_{ki}} - \frac{n_{k0}}{1-p_{ki}} = 0\end{aligned}$$

- $n_{k1} = |\{m: x_{mi} = 1, y_m = k\}|$ is the number of data samples in set of $\{m: x_{mi} = 1, y_m = k\}$
- $n_{k0} = |\{m: x_{mi} = 0, y_m = k\}|$ is the number of data samples in set of $\{m: x_{mi} = 0, y_m = k\}$
- $\{m: x_{mi} = 1, y_m = k\}$ is a set that contains every sample with $x_i = 1$ in class k
- $\{m: x_{mi} = 0, y_m = k\}$ is a set that contains every sample with $x_i = 0$ in class k
- $|\{m: x_{mi} = 1, y_m = k\}| + |\{m: x_{mi} = 0, y_m = k\}| = n_k$

$$p_{ki} = \frac{n_{ki}}{n_k}$$

이미지에서
1가지 이상 비율
} 전부 확률과
동일하다.

Bernoulli Naïve Bayes: Estimation of Parameters

x	y
1	0
1	0
1	0
0	0
0	1
0	1
1	1



$$p(x = 0|y = 0) = p_{00} = \frac{1}{4}$$

$$p(x = 1|y = 0) = p_{01} = \frac{3}{4}$$



$$p(x = 0|y = 1) = p_{10} = \frac{2}{3}$$

$$p(x = 1|y = 1) = p_{11} = \frac{1}{3}$$

Categorical Naïve Bayes

- Discrete random variables with more than two outcomes

$$P(\mathbf{x} = (x_1, x_2, \dots, x_m)) = \prod_{j=1}^m p_j^{x_j}$$

- m outcomes
- x_j is a binary indicator variable: 1 when j -th outcome is observed; otherwise 0
- p_j is probability of j -th outcome ($\sum_{j=1}^m p_m = 1$)

$$p(\mathbf{X}|C_k) = \prod_{i=1}^n \prod_{j=1}^m p_{kj}^{x_j}$$

$$\log p(\mathbf{X}|C_k) = \sum_{i=1}^n \sum_{j=1}^m x_j \log p_{kj}$$
$$p_{ki} = \frac{n_{kj}}{n_k}$$

Categorical Naïve Bayes

- Discrete random variables with more than two outcomes

x	y
High	0
Mid	0
High	0
Low	0
High	0
Low	0
High	0
High	0

$$p(x = \text{High}|y = 0) = p_{0,\text{High}} = \frac{5}{8}$$

$$p(x = \text{Mid}|y = 0) = p_{0,\text{Mid}} = \frac{1}{8}$$

$$p(x = \text{Low}|y = 0) = p_{0,\text{Low}} = \frac{2}{8}$$

Parameter Smoothing

- Smoothing techniques for parameter estimation
 - ▣ Smoothing techniques helps prevent zero probabilities when a category is not observed in the training data
 - ▣ Laplace smoothing (additive smoothing)
$$p_{kj} = \frac{n_{kj} + \alpha}{n_k + \alpha m}$$
- α : smoothing parameter, commonly $\alpha = 1$

Multinomial Naïve Bayes

- Multinomial naïve Bayes
 - ▣ With a multinomial event model, samples represent the frequencies with which certain events have been generated by a multinomial (p_1, \dots, p_d)
 - p_i is the probability that event i occurs
 - m is the number of features in input data

$$p(\mathbf{x}|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_{ki}^{x_i}$$

- $\mathbf{x} = (x_1, x_2, \dots, x_d)$ represents each sample and it can be seen as a histogram with x_i counting the number of times event i was observed in a particular instance
- ▣ This is the event model typically used for document classification
 - With events representing the occurrence of a word in a single document
→ bag of words representation



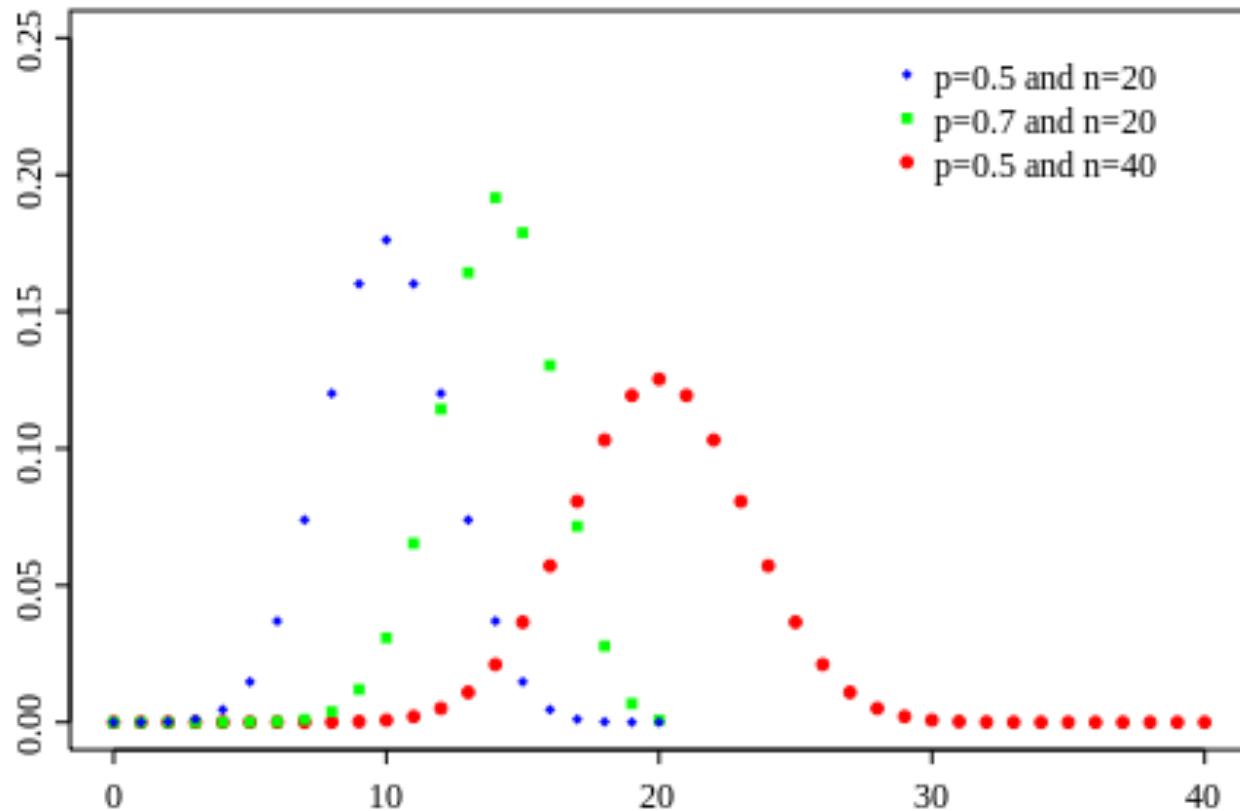
※ Multinomial distribution

- Multinomial distribution is a generalization of the binomial distribution
 - Binomial distribution with parameters n and p is the discrete probability distribution of the number of successes in a sequence of n independent yes/no experiments with success probability p
$$p(k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$
 - Example of binomial distribution is the distribution of the number of head when flipping a coin n times (in this case, $p = 0.5$)
 - Probability that k times head occur among n trials
$$p(k) = \frac{n!}{k!(n-k)!} 0.5^k 0.5^{n-k} = \frac{n!}{k!(n-k)!} 0.5^n$$
 - In multinomial distribution, possible outcome is more than two and each outcome has its own probability to occur, (p_1, \dots, p_d)
 - $p_1 + \dots + p_d = 1$
 - d is the number of possible outcomes
 - $n_x = \sum_{i=1}^d x_i$

$$p(\mathbf{x} = (x_1, x_2, \dots, x_d)) = \frac{n_{\mathbf{x}}!}{x_1! \cdots x_d!} p_1^{x_1} \cdots p_d^{x_d}$$

※ Multinomial distribution

□ Binomial distribution



Multinomial Naïve Bayes: Application

- Relies on very simple representation of document
 - ▣ Bag of words

I simply loved this film but was shocked by the bad reviews that people gave it. To this I say to them: You seriously misunderstood the meaning of it. Although I won't reveal any real details about the meaning because I think that you should try and understand it yourself. The movie was terrific and simply breathless the whole time. I felt awestruck about how the life of one man could be so changed after an experience that Hanks went through. I say that every element of the film was perfect. And for those of you who hate Wilson, you have to understand about how human he really was to Chuck. I was amazed on how well this movie was made and think that everybody should have an experience that should cause you to take stock of your life

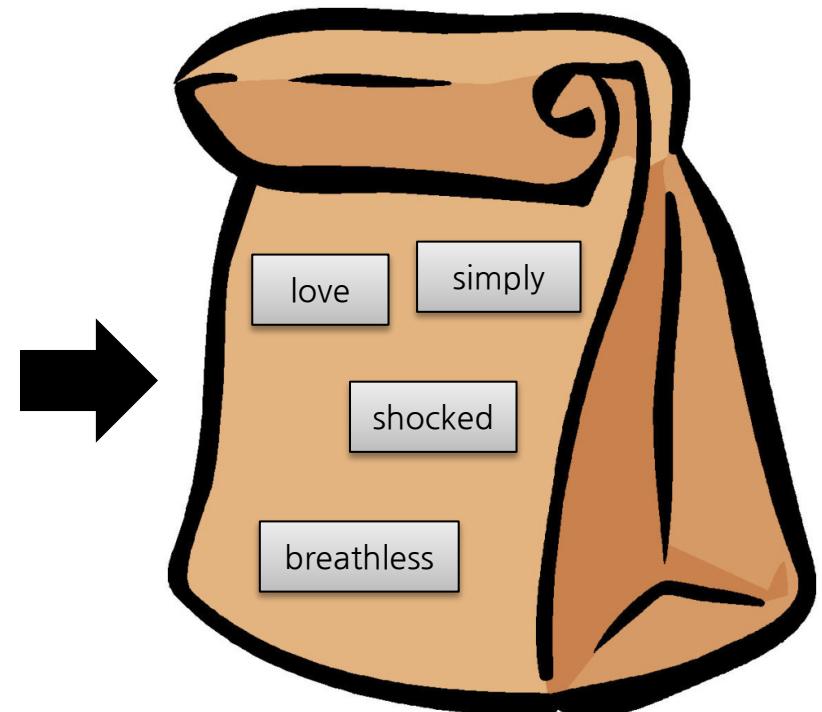
$f($

) = C

Multinomial Naïve Bayes: Application

- The bag of words representation

I simply loved this film but was shocked by the bad reviews that people gave it. To this I say to them: You seriously misunderstood the meaning of it. Although I won't reveal any real details about the meaning because I think that you should try and understand it yourself. The movie was terrific and simply breathless the whole time. I felt awestruck about how the life of one man could be so changed after an experience that Hanks went through. I say that every element of the film was perfect. And for those of you who hate Wilson, you have to understand about how human he really was to Chuck. I was amazed on how well this movie was made and think that everybody should have an experience that should cause you to take stock of your life



Multinomial Naïve Bayes: Application

- The bag of words representation
 - ▣ It is possible to select a subset of words

Word	Frequency
love	1
great	2
recommend	1
simply	1
happy	1
bad	2
:	:

$f() = C$

Multinomial Naïve Bayes: Application

- The bag of words representation
 - ▣ It is possible to select a subset of words

Word	Frequency
love	1
great	2
recommend	1
old	0
simply	1
happy	1
bad	2
dog	0
:	:

$f($

$) = C$

Multinomial Naïve Bayes: Application

- Revisit naïve Bayes

$$\hat{y} = \operatorname{argmax}_{k \in \{1, \dots, K\}} p(C_k) \prod_{i=1}^p p(x_i | C_k)$$

- Each i th column represent the specific word
- C_k is the possible output class
 - Ex.) Spam filter: spam or non-spam
- The basic idea that uses Naïve Bayes for text classification
 - Each class has the different distribution of words
 - Spam mails usually contain “click below”, “free dvd”, “great offer” and etc.

Multinomial Naïve Bayes: Estimation of Parameters

- Maximum likelihood estimator with Laplace smoothing

$$p_{ki} = \frac{\sum_{j \in C_k} x_{ji} + \alpha}{\sum_{j \in C_k} \sum_{l=1}^d x_{jl} + \alpha d}$$

- p_{ki} : The probability of occurrence of i -th variable for class k
- d : The number of input variables
- x_{ji} : The occurrence of i -th variable for sample j

Spam	free	click	w̄in
A	2	1	0
B	1	0	1
C	0	1	1
	3	2	2

$$d=3$$

$$\alpha = 1$$

$$P(\text{free} | \text{Spam}) = \frac{3}{7} \Rightarrow \frac{3+1}{7+3 \times 1} = 0.4$$

$$P(\text{w̄in} | \text{Spam}) = \frac{2}{7} \Rightarrow \frac{2+1}{7+3 \times 1} = 0.3$$

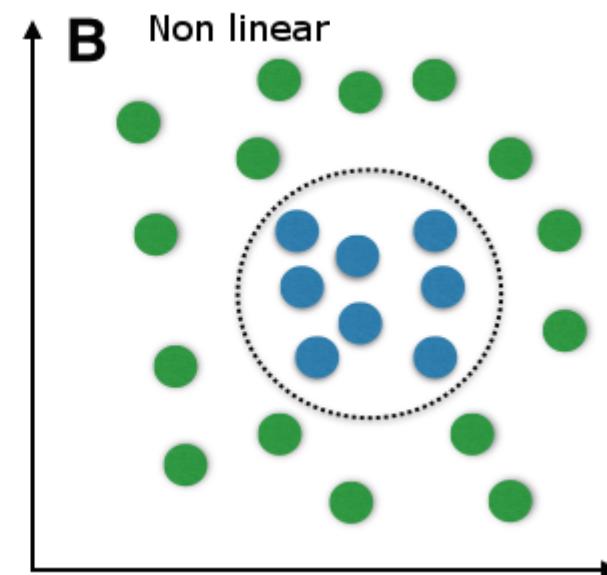
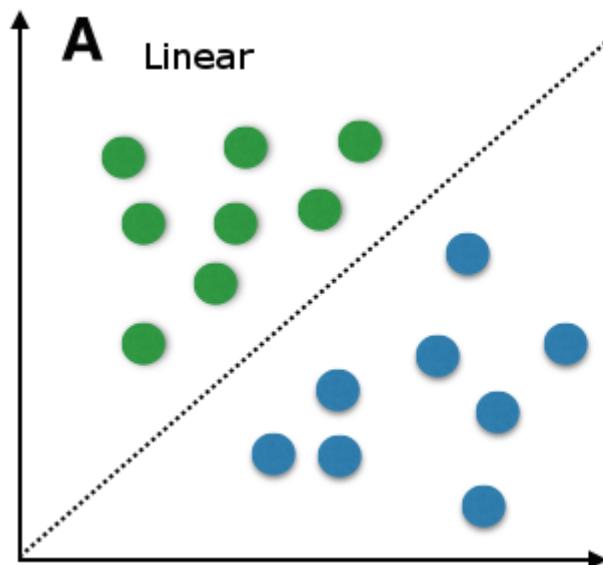
Multinomial Naïve Bayes

- Multinomial naïve Bayes
 - ▣ The multinomial naive Bayes classifier becomes a linear classifier when expressed in log-space

$$\log p(C_k | \mathbf{x}) \propto \log p(C_k) \prod_i p_{k_i}^{x_i} = \log p(C_k) + \sum_{i=1}^n x_i \log p_{ki} = b + \mathbf{w}_k^T \mathbf{x}$$

- $b = \log p(C_k)$
- $w_{ki} = \log p_{ki}$
- $\frac{(\sum_i x_i)!}{\prod_i x_i!}$ term only depends on \mathbf{x} and does not depend on class

* Decision Boundary



Complement Naïve Bayes

- Complement naïve Bayes
 - ▣ Complement naïve Bayes is an adaptation of the standard multinomial naïve Bayes algorithm that is particularly suited for imbalanced data sets
 - ▣ Unlike multinomial naïve Bayes, it uses word occurrence frequencies from the complement of each class

$$p_{ki} = \frac{\sum_{j \notin C_k} x_{ji} + \alpha}{\sum_{j \notin C_k} \sum_{l=1}^d x_{jl} + \alpha d}$$

Spam vs Ham

⇒ Spam 예측 시 Ham 틀림 사용
Ham 예측 시 Spam 틀림 사용

Gaussian Naïve Bayes

- Gaussian naïve Bayes
 - ▣ When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution

$$p(x = v | C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-u_k)^2}{2\sigma_k^2}}$$

$$p(C_k) \prod_{i=1}^p p(x_i | C_k) = p(C_k) \prod_{i=1}^p \frac{1}{\sqrt{2\pi\sigma_{ki}^2}} e^{-\frac{(v_i-u_{ki})^2}{2\sigma_{ki}^2}}$$

여러개의 변수 \Rightarrow 각 변수에서 평균과 분산이 다른데 고려

Naïve Bayes Classifier: Variations

- Semi-naïve Bayes classifier
 - ▣ A type of Bayesian classifier that relaxes the strict independence assumption of the Naive Bayes classifier, allowing for limited dependencies between features, while still maintaining computational efficiency and simplicity
 - It assumes that correlations exist only within disjoint subsets of features, meaning that features within a subset can depend on each other, but not across subsets
$$\begin{aligned}P(\mathbf{x}|C) &= P(x_1, x_2, \dots, x_n|C) \\&= P(x_1|C)P(x_2|C) \cdot \dots \cdot P(x_k|C) \cdot P(x_{k+1}, x_{k+2}, \dots, x_n|C)\end{aligned}$$
 - x_1, x_2, \dots, x_k are the features assumed to be conditionally independent
 - $x_{k+1}, x_{k+2}, \dots, x_n$ are the features that are allowed to have dependencies

0118 521
27 718

일부 특성 X, 다른!

Naïve Bayes Classifier: Variations

- Hidden naïve Bayes classifier
 - ▣ The Hidden naïve Bayes classifier is an extension of the standard naïve Bayes classifier, which introduces the concept of latent or hidden variables into the model
 - Hidden variables are assumed to influence the observed features but are not directly observed or measured

$$P(C|\mathbf{x}) = \sum_{h \in H} P(C|h) \prod_i P(x_i|C,h)$$

where h represents the hidden variables

- ▣ This model is especially useful when we suspect that there are unobserved or hidden factors influencing the outcome, and these factors should be taken into account for better classification performance

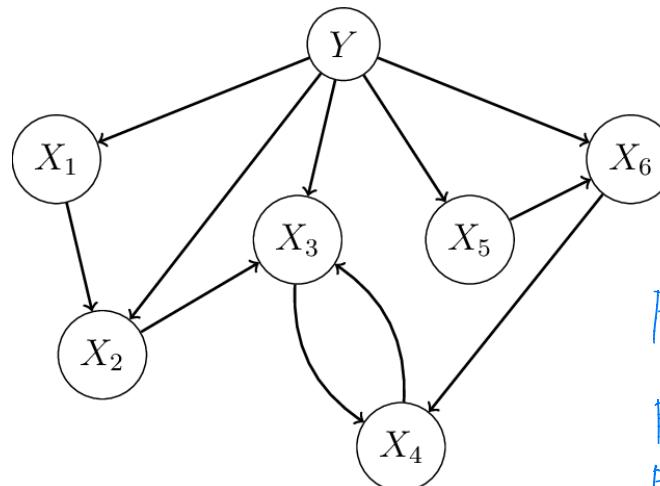
간접 feature 와 숨겨진 변수 존재 가능.

Naïve Bayes Classifier: Variations

- Bayesian network classifier
 - ▣ A Bayesian network classifier a probabilistic classifier that uses a **Bayesian network** to model the relationships between variables
 - ▣ It allows dependencies between features by using a probabilistic graphical model

$$P(\mathbf{x}) = \prod_{i=1} P(x_i | pa_i)$$

- pa_i represents the set of parent nodes for x_i
- ▣ It requires structure learning which can be computationally expensive, especially for large datasets with many features



Study sleep
 \ /
 intelligence
 |
 pass

$$P(i = \text{high} \mid \text{study} = \text{yes}, \text{sleep} = \text{no.}) = 0.7$$

$$P(\text{pass} = \text{yes} \mid i = \text{high}) = 0.95$$

$$P(\text{pass} = \text{yes} \mid \text{study} = \text{yes}, \text{sleep} = \text{no.}) = 0.7 \times 0.95$$