

1 데이터분석 단계(Data Analysis Cycle)

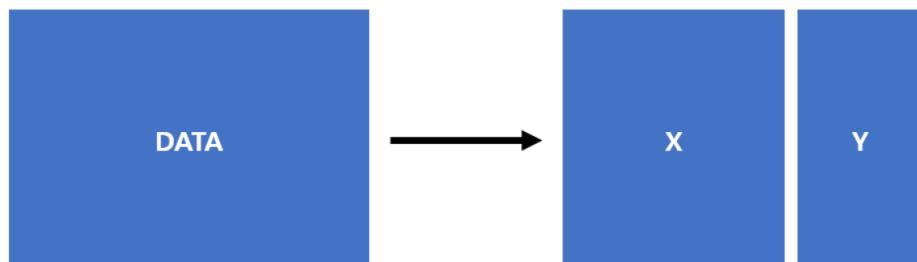
[Open in Colab](#)

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

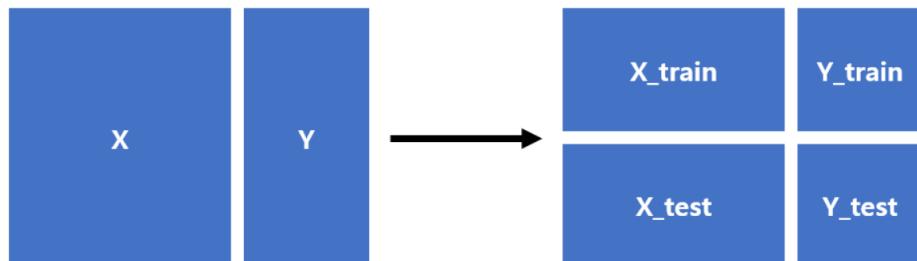
100	T50	횟수	111	TPU	...
few	Gds	Hvi	Rew	Fa	...
Fre	CT	QTP	D	합	...
'1'	1	23	22	NaN	43
76	NaN	43	32	1	8
'Hi'	NaN	NaN	NaN	NaN	87
23	98	NaN	64	46	NaN
c	90	'WW'	24	'KK'	4
t	NaN	2	NaN	NaN	6
64	NaN	90	'IU'	4	76

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

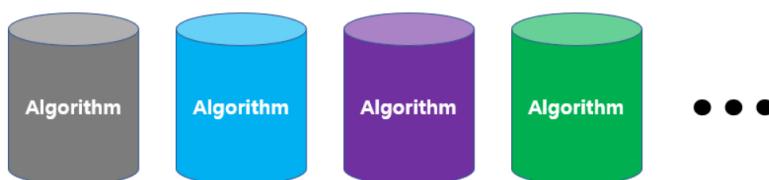
✓ 데이터 분할: (1) 목표/종속변수 Y와 설명/독립변수 X설정



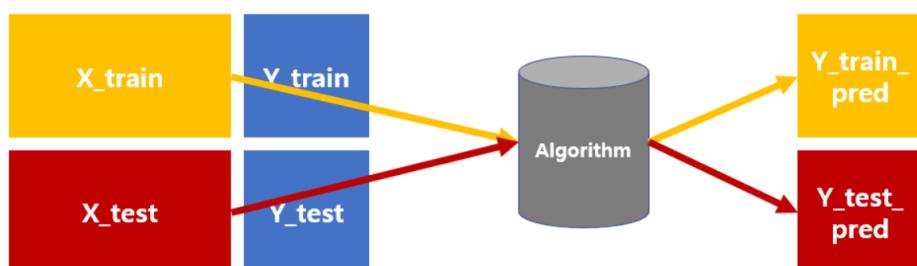
✓ 데이터 분할: (2) 학습데이터 Train과 예측 데이터 Test로 분할



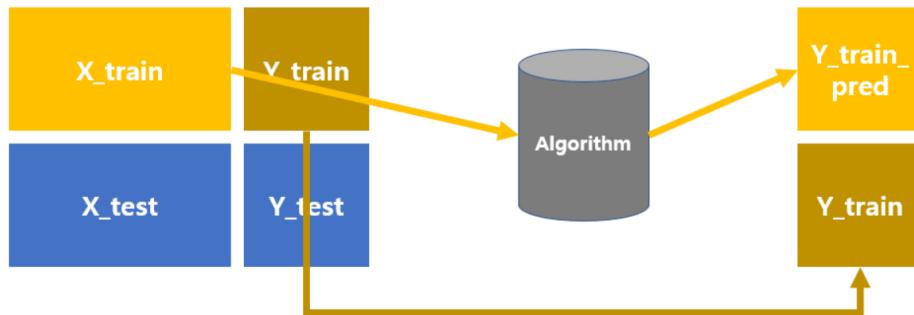
✓ 모델링: (3) 분석 목적에 맞는 알고리즘(Base & Advanced) 후보들 준비



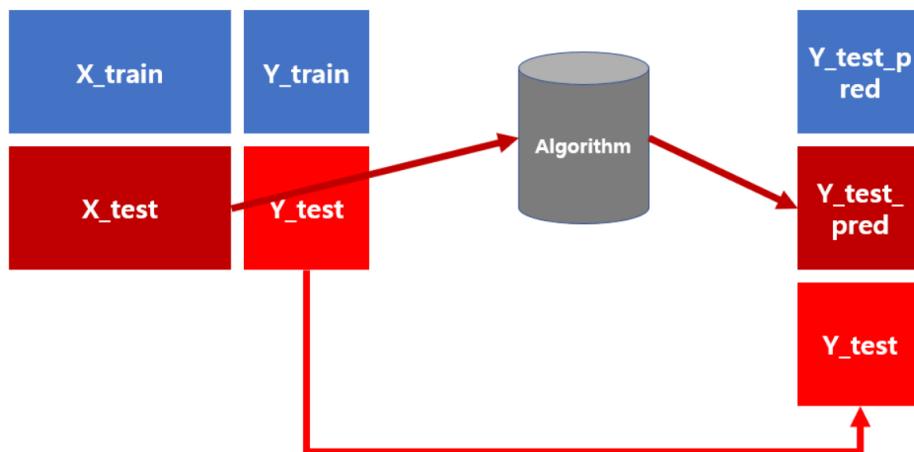
✓ 모델링 & 학습: (4) 알고리즘 평가를 위해 Train/Test의 예측값 추정



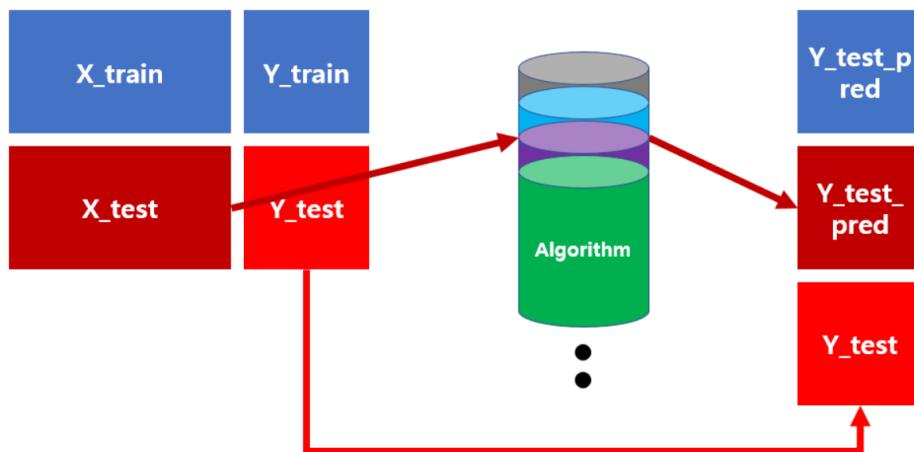
✓ 평가: (5) 학습(Train)이 잘 되었는지 알고리즘 성능검증



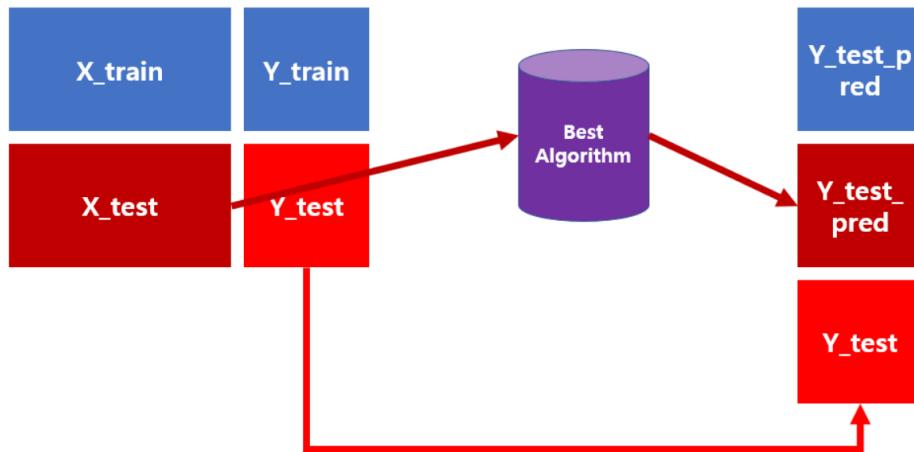
✓ 평가: (6) 예측(Test)이 잘 되었는지 알고리즘 성능검증

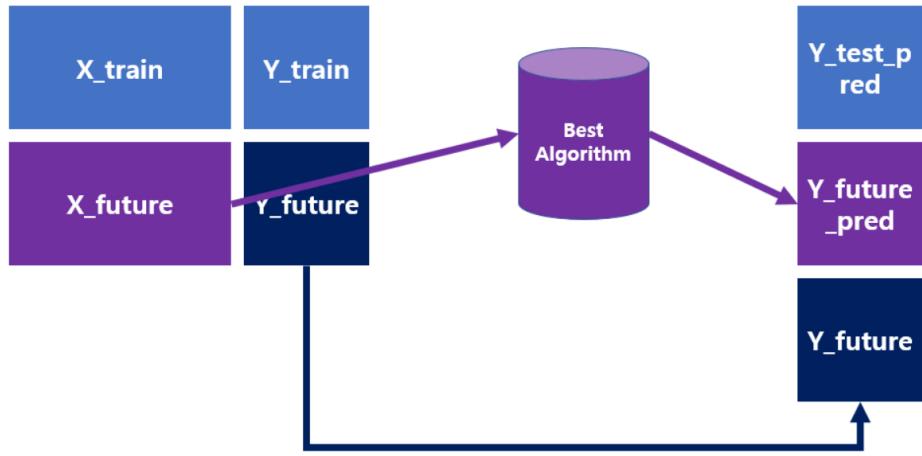


✓ 최적 알고리즘 선택: (7) 알고리즘을 변경하여 위 과정 반복 후



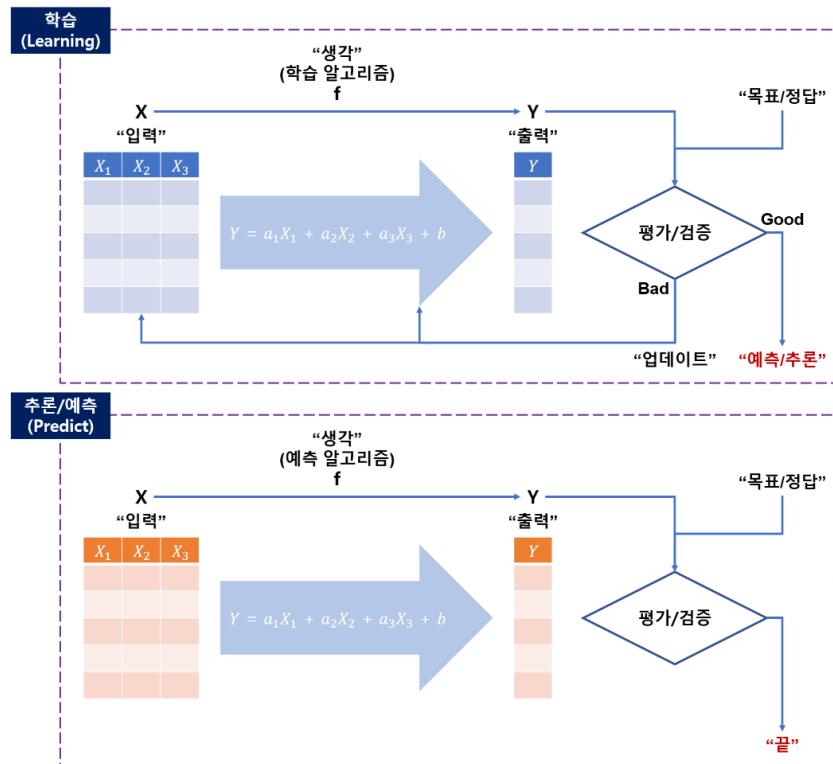
✓ 최적 알고리즘 선택: (7) 최고 성능의 알고리즘 선택





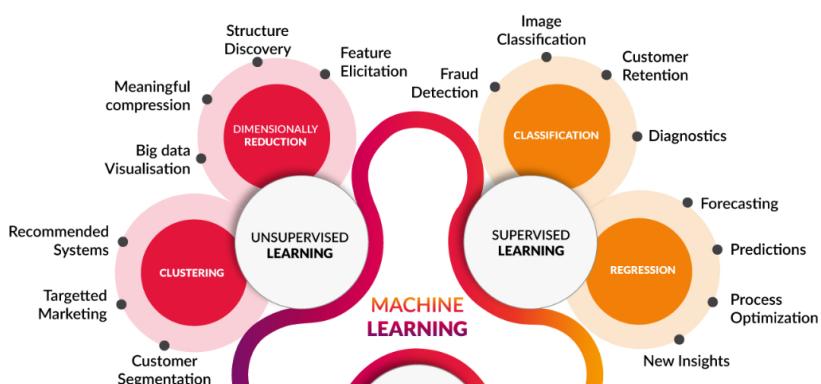
2 지도학습(Supervised) 알고리즘: 회귀분석

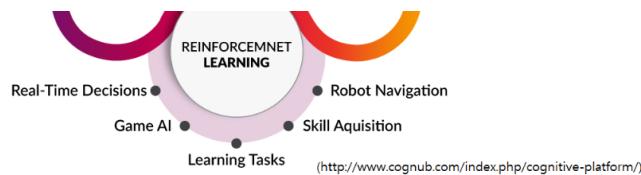
- 데이터분석 과정: 학습 + 추론/예측



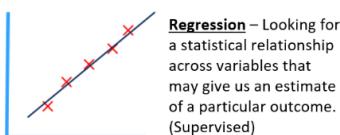
- 회귀분석: 지도학습 알고리즘 중 예측을 위해 사용되는 가장 기본(Baseline) 알고리즘

“연속형 출력(Y, 종속변수)에 영향을 주는 입력(X, 독립변수)과의 관계를 정량적으로 추론/추정하여 미래 값을 예측하는 알고리즘”

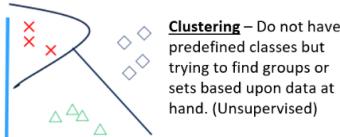




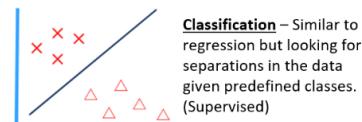
- How much is the stock of Samsung Electronics tomorrow?



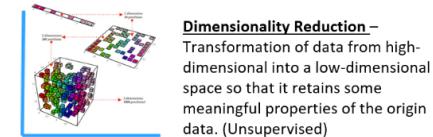
- Are Samsung Electronics and Naver similar business companies?



- Will Samsung Electronics' stocks rise or fall tomorrow?



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



Regression Algorithms	Instance-based Algorithms	Regularization Algorithms	Decision Tree Algorithms	Bayesian Algorithms	Artificial Neural Network Algorithms
Ordinary Least Squares Regression (OLSR)	k-Nearest Neighbor (kNN)	Ridge Regression	Classification and Regression Tree (CART)	Naive Bayes	Perceptron
Linear Regression	Learning Vector Quantization (LVQ)	Least Absolute Shrinkage and Selection Operator (LASSO)	Iterative Dichotomiser 3 (ID3)	Gaussian Naive Bayes	Back-Propagation
Logistic Regression	Self-Organizing Map (SOM)	Elastic Net	C4.5 and C5.0 (different versions of a powerful approach)	Multinomial Naive Bayes	Hopfield Network
Stepwise Regression	Locally Weighted Learning (LWL)	Least-Angle Regression (LARS)	Chi-squared Automatic Interaction Detection (CHAID)	Averaged One-Dependence Estimators (AODE)	Radial Basis Function Network (RBFN)
Multivariate Adaptive Regression Splines (MARS)	-	-	Decision Stump	Bayesian Belief Network (BBN)	-
Locally Estimated Scatterplot Smoothing (LOESS)	-	-	M5	Bayesian Network (BN)	-
-	-	-	Conditional Decision Trees	-	-

- Target Algorithm:

- Linear Regression (Simple/Multiple/Multivariate)
- Polynomial Regression
- Stepwise Regression
- Ridge Regression
- Lasso Regression
- ElasticNet Regression
- Bayesian Linear Regression
- Quantile Regression
- Decision Tree Regression
- Random Forest Regression
- Support Vector Regression

3 예제 데이터셋(Dataset)

3.1 statsmodels 모듈 사용 데이터셋

```
# 라이브러리 불러오기
import statsmodels.api as sm

• 대기중 CO2농도 데이터:
data = sm.datasets.get_rdataset("CO2", package="datasets")

• 황체형성 호르몬(Luteinizing Hormone)의 수치 데이터:

```

```

data = sm.datasets.get_rdataset("Ih")
• 1974~1979년 사이의 영국의 흐름기 질환 사망자 수 데이터:
data = sm.datasets.get_rdataset("deaths", "MASS")
• 1949~1960년 사이의 국제 항공 운송인원 데이터:
data = sm.datasets.get_rdataset("AirPassengers")
• 미국의 강수량 데이터:
data = sm.datasets.get_rdataset("precip")
• 타이타닉호의 탑승자들에 대한 데이터:
data = sm.datasets.get_rdataset("Titanic", package="datasets")

```

- **data**가 포함하는 정보:

- **package**: 데이터를 제공하는 R 패키지 이름
- **title**: 데이터 이름
- **data**: 데이터를 담고 있는 데이터프레임
- **__doc__**: 데이터에 대한 설명 문자열(R 패키지의 내용 기준)

3.2 sklearn 모듈 사용 데이터셋

1) 패키지에 포함된 데이터(`load` 명령어)

```

# 라이브러리 불러오기
from sklearn.datasets import load_boston

• load_boston: 회귀용 보스턴 집 값
raw = load_boston()
print(raw.DESCR)
print(raw.keys())
print(raw.data.shape, raw.target.shape)

• load_diabetes: 회귀용 당뇨병 자료
• load_linnerud: 회귀용 linnerud 자료
• load_iris: 분류용 붓꽃(iris) 자료
• load_digits: 분류용 숫자(digit) 필기 이미지 자료
• load_wine: 분류용 포도주(wine) 등급 자료
• load_breast_cancer: 분류용 유방암(breast cancer) 진단 자료

```

2) 인터넷에서 다운로드할 수 있는 데이터(`fetch` 명령어)

```

# 라이브러리 불러오기
from sklearn.datasets import fetch_california_housing

• fetch_california_housing: 회귀용 캘리포니아 집 값
raw = fetch_california_housing()
print(raw.DESCR)
print(raw.keys())
print(raw.data.shape, raw.target.shape)

• fetch_covtype: 회귀용 토지 조사 자료
• fetch_20newsgroups: 뉴스 그룹 텍스트 자료
• fetch_olivetti_faces: 얼굴 이미지 자료
• fetch_lfw_people: 유명인 얼굴 이미지 자료
• fetch_lfw_pairs: 유명인 얼굴 이미지 자료
• fetch_rcv1: 로이터 뉴스 말뭉치
• fetch_kddcup99: Kddcup 99 Tcp dump 자료

```

3) 확률분포를 사용한 가상 데이터(`make` 명령어)

```

# 라이브러리 불러오기
from sklearn.datasets import make_regression

• make_regression: 회귀용 가상 데이터
X, y, c = make_regression(n_samples=100, n_features=10, n_targets=1, bias=0, noise=0, coef=True, random_state=0)

• make_classification: 분류용 가상 데이터 생성
• make_blobs: 클러스터링용 가상 데이터 생성

```

4) `load/fetch` 명령어 데이터에서 `raw`가 포함하는 정보: Bunch라는 클래스 객체 형식으로 생성

- **data**: (필수) 독립 변수 ndarray 배열
- **target**: (필수) 종속 변수 ndarray 배열
- **feature_names**: (옵션) 독립 변수 이름 리스트

- target_names : (옵션) 종속 변수 이름 리스트
- DESC : (옵션) 자료에 대한 설명

3.3 회귀문제 예시

```
In [1]: # Regression
import statsmodels.api as sm
data = sm.datasets.get_rdataset("CO2", package="datasets")
print(data.title)
print(data.__doc__)
print(data.keys())
display(data.data)
executed in 1.80s, finished 03:31:46 2022-04-22

Carbon Dioxide Uptake in Grass Plants
.. container:: 

    === =====
    CO2 R Documentation
    === =====

    .. rubric:: Carbon Dioxide Uptake in Grass Plants
    :name: carbon-dioxide-uptake-in-grass-plants

    .. rubric:: Description
    :name: description

    The ``CO2`` data frame has 84 rows and 5 columns of data from an
    experiment on the cold tolerance of the grass species *Echinochloa
    crus-galli*.

    .. rubric:: Usage
    :name: usage

    ::

    CO2

    .. rubric:: Format
    :name: format

    An object of class
    ``c("nfnGroupedData", "nfGroupedData", "groupedData", "data.frame")``
    containing the following columns:

    Plant
    an ordered factor with levels ``Qn1`` < ``Qn2`` < ``Qn3`` < ... <
    ``Mc1`` giving a unique identifier for each plant.

    Type
    a factor with levels ``Quebec`` ``Mississippi`` giving the origin
    of the plant

    Treatment
    a factor with levels ``nonchilled`` ``chilled``

    conc
    a numeric vector of ambient carbon dioxide concentrations (mL/L).

    uptake
    a numeric vector of carbon dioxide uptake rates
    (``#mu#mbox{mol}/m^2`` sec).

    .. rubric:: Details
    :name: details

    The ``CO_2`` uptake of six plants from Quebec and six plants from
    Mississippi was measured at several levels of ambient ``CO_2``
    concentration. Half the plants of each type were chilled overnight
    before the experiment was conducted.

    This dataset was originally part of package ``nlme``, and that has
    methods (including for ``[`` , ``as.data.frame`` , ``plot`` and
    ``print``) for its grouped-data classes.

    .. rubric:: Source
    :name: source

    Potvin, C., Lechowicz, M. J. and Tardif, S. (1990) "The statistical
    analysis of ecophysiological response curves obtained from
    experiments involving repeated measures", *Ecology*, **71**,
    1389-1400.

    Pinheiro, J. C. and Bates, D. M. (2000) *Mixed-effects Models in S
    and S-PLUS*, Springer.

    .. rubric:: Examples
    :name: examples

    ::

        require(stats); require(graphics)

        coplot(uptake ~ conc | Plant, data = CO2, show.given = FALSE, type = "b")
        ## fit the data for the first plant
```

```

fml <- nls(uptake ~ SSasymp(conc, Asym, lrc, c0),
           data = CO2, subset = Plant == "Qn1")
summary(fml)
## fit each plant separately
fmlist <- list()
for (pp in levels(CO2$Plant)) {
  fmlist[[pp]] <- nls(uptake ~ SSasymp(conc, Asym, lrc, c0),
                      data = CO2, subset = Plant == pp)
}
## check the coefficients by plant
print(sapply(fmlist, coef), digits = 3)

dict_keys(['data', '__doc__', 'package', 'title', 'from_cache'])

```

	Plant	Type	Treatment	conc	uptake
0	Qn1	Quebec	nonchilled	95	16.0
1	Qn1	Quebec	nonchilled	175	30.4
2	Qn1	Quebec	nonchilled	250	34.8
3	Qn1	Quebec	nonchilled	350	37.2
4	Qn1	Quebec	nonchilled	500	35.3
...
79	Mc3	Mississippi	chilled	250	17.9
80	Mc3	Mississippi	chilled	350	17.9
81	Mc3	Mississippi	chilled	500	17.9
82	Mc3	Mississippi	chilled	675	18.9
83	Mc3	Mississippi	chilled	1000	19.9

84 rows × 5 columns

```

In [2]: # Regression
from sklearn.datasets import load_boston
raw = load_boston()
print(raw.DESCR)
print(raw.keys())
print(raw.data.shape, raw.target.shape)
executed in 119ms, finished 03:31:47 2022-04-22
... _boston_dataset:

Boston house prices dataset
-----
**Data Set Characteristics:**
:Number of Instances: 506
:Number of Attributes: 13 numeric/categorical predictive. Median Value (attribute 14) is usually the target.
:Attribute Information (in order):
 - CRIM per capita crime rate by town
 - ZN proportion of residential land zoned for lots over 25,000 sq.ft.
 - INDUS proportion of non-retail business acres per town
 - CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
 - NOX nitric oxides concentration (parts per 10 million)
 - RM average number of rooms per dwelling
 - AGE proportion of owner-occupied units built prior to 1940
 - DIS weighted distances to five Boston employment centres
 - RAD index of accessibility to radial highways
 - TAX full-value property-tax rate per $10,000
 - PTRATIO pupil-teacher ratio by town
 - B 1000(Bk - 0.63)^2 where Bk is the proportion of black people by town
 - LSTAT % lower status of the population
 - MEDV Median value of owner-occupied homes in $1000's

:Missing Attribute Values: None
:Creator: Harrison, D. and Rubinfeld, D.L.

This is a copy of UCI ML housing dataset.
https://archive.ics.uci.edu/ml/machine-learning-databases/housing/

```

This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University.

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic prices and the demand for clean air', J. Environ. Economics & Management, vol.5, 81-102, 1978. Used in Belsley, Kuh & Welsch, 'Regression diagnostics ...', Wiley, 1980. N.B. Various transformations are used in the table on pages 244-261 of the latter.

The Boston house-price data has been used in many machine learning papers that address regression problems.

... topic:: References

- Belsley, Kuh & Welsch, 'Regression diagnostics: Identifying Influential Data and Sources of Collinearity', Wiley, 1980. 244-261.
- Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on the Tenth International Conference on Machine Learning, 236-243, University of Massachusetts, Amherst. Morgan Kaufmann.

```

dict_keys(['data', 'target', 'feature_names', 'DESCR', 'filename', 'data_module'])
(506, 13) (506,)

```

C:\Users\KKK\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Function load_boston is deprecated; `load_bos

ton` is deprecated in 1.0 and will be removed in 1.2.

The Boston housing prices dataset has an ethical problem. You can refer to the documentation of this function for further details.

The scikit-learn maintainers therefore strongly discourage the use of this dataset unless the purpose of the code is to study and educate about ethical issues in data science and machine learning.

In this special case, you can fetch the dataset from the original source::

```
import pandas as pd
import numpy as np

data_url = "http://lib.stat.cmu.edu/datasets/boston"
raw_df = pd.read_csv(data_url, sep="#$+", skiprows=22, header=None)
data = np.hstack([raw_df.values[:, 2:], raw_df.values[1:, 2]])
target = raw_df.values[1:, 2]
```

Alternative datasets include the California housing dataset (i.e. :func:`~sklearn.datasets.fetch_california_housing`) and the Ames housing dataset. You can load the datasets as follows::

```
from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()

for the California housing dataset and:

from sklearn.datasets import fetch_openml
housing = fetch_openml(name="house_prices", as_frame=True)

for the Ames housing dataset.
warnings.warn(msg, category=FutureWarning)
```

```
In [3]: # Regression
from sklearn.datasets import fetch_california_housing
raw = fetch_california_housing()
print(raw.DESCR)
print(raw.keys())
print(raw.data.shape, raw.target.shape)
executed in 29ms, finished 03:31:47 2022-04-22
```

... _california_housing_dataset:

California Housing dataset

Data Set Characteristics:

:Number of Instances: 20640

:Number of Attributes: 8 numeric, predictive attributes and the target

:Attribute Information:

- MedInc median income in block group
- HouseAge median house age in block group
- AveRooms average number of rooms per household
- AveBedrms average number of bedrooms per household
- Population block group population
- AveOccup average number of household members
- Latitude block group latitude
- Longitude block group longitude

:Missing Attribute Values: None

This dataset was obtained from the StatLib repository.

https://www.dcc.fc.up.pt/~itorgo/Regression/cal_housing.html

The target variable is the median house value for California districts, expressed in hundreds of thousands of dollars (\$100,000).

This dataset was derived from the 1990 U.S. census, using one row per census block group. A block group is the smallest geographical unit for which the U.S. Census Bureau publishes sample data (a block group typically has a population of 600 to 3,000 people).

An household is a group of people residing within a home. Since the average number of rooms and bedrooms in this dataset are provided per household, these columns may take surprisingly large values for block groups with few households and many empty houses, such as vacation resorts.

It can be downloaded/loaded using the :func:`sklearn.datasets.fetch_california_housing` function.

... topic:: References

- Pace, R. Kelley and Ronald Barry, Sparse Spatial Autoregressions, Statistics and Probability Letters, 33 (1997) 291-297

```
dict_keys(['data', 'target', 'frame', 'target_names', 'feature_names', 'DESCR'])
(20640, 8) (20640,)
```

```
In [4]: # Regression
```

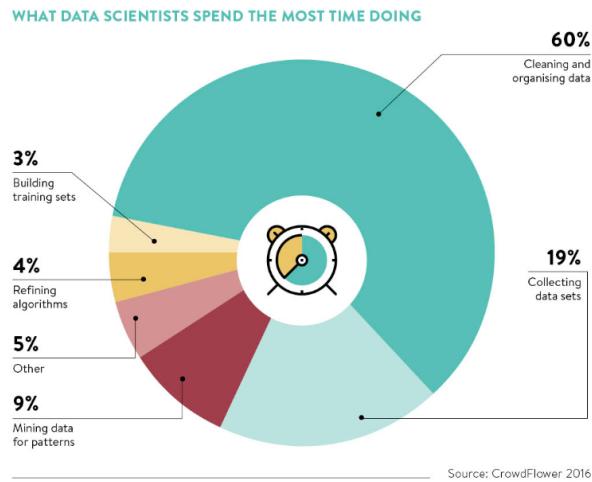
```
from sklearn.datasets import make_regression
X, y, c = make_regression(n_samples=100, n_features=10, n_targets=1, bias=0, noise=0, coef=True, random_state=0)
print(X.shape, y.shape, c.shape)
```

executed in 14ms. finished 03:31:47 2022-04-22

4 전처리 방향(Preprocessing)

- 목표:

- 대량으로 수집된 데이터는 그대로 활용 어려움
- 잘 못 수집/처리 된 데이터는 엉뚱한 결과를 발생
- 알고리즘이 학습이 가능한 형태로 데이터를 정리



일반적인 전처리 필요항목:

- 데이터 결합
- 결측값 처리
- 이상치 처리
- 자료형 변환
- 데이터 분리
- 데이터 변환
- 스케일 조정

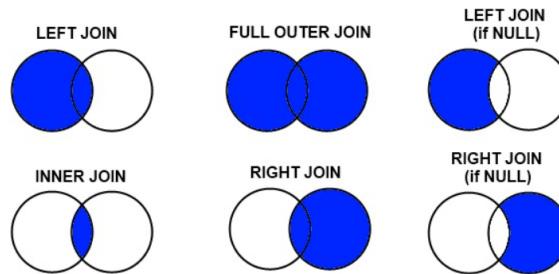
4.1 데이터 결합(Data Integration)

- 목표: 여러개로 구분된 데이터 이거나 블록데이터로 확장 할 경우 결합

- 중복 데이터 제거
- 의미는 같으나 단위나 이름의 표현이 다른 경우 일치 필요

```
import pandas as pd
pd.merge(베이스 데이터프레임, 결합할 데이터프레임, how= , on= , ...)
```

- how : left, right, inner, outer (2개의 데이터프레임을 어떤 방식으로 결합할지 결정)
- on : 2개의 데이터프레임 결합을 위한 key 설정
- left_on : key 이름이 서로 다를 경우 베이스 데이터프레임의 key 설정
- right_on : key 이름이 서로 다를 경우 결합할 데이터프레임의 key 설정

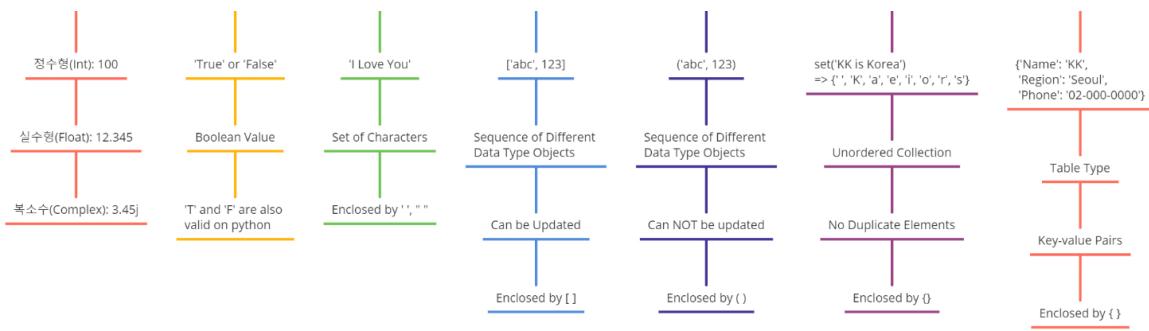


4.2 자료형 변환(Type)

- 목표: 각 변수의 특성을 확인하고 범주형과 연속형에 맞도록 변경

Python Data Types





- 사람의 데이터 분류:

- 데이터 관계에 따라: $Y = f(X)$

대분류	의미/예시
독립변수(Independent Variable)	다른 변수에 영향을 미치는 변수 (X)
종속변수(Dependent Variable)	다른 변수에 의해 영향을 받는 변수 (Y)

- 데이터 특성에 따라:

대분류	소분류	의미/예시
질적변수(Qualitative Variable)	-	내부 값이 특정 범주(Category)로 분류된 변수(색상, 성별, 종교)
	명목형 변수(Nominal Variable)	값이 순위가 존재하지 않는 경우(혈액형)
	순위형 변수(Ordinal Variable)	값이 순위가 존재하는 경우(성적)
양적변수(Quantitative Variable)	-	내부 값이 다양한 숫자 분포로 구성된 변수(기, 몸무게, 소득)
	이산형 변수(Discrete Variable)	값이 셀 수 있는 경우(정수)
	연속형 변수(Continuous Variable)	값이 셀 수 없는 경우(실수)

- 컴퓨터의 데이터 분류:

대분류	소분류	컴퓨터의 분류1	컴퓨터의 분류2
질적변수(Qualitative Variable)	-	-	범주형
	명목형 변수(Nominal Variable)	문자	범주형
	순위형 변수(Ordinal Variable)	숫자	범주형
양적변수(Quantitative Variable)	-	-	연속형
	이산형 변수(Discrete Variable)	숫자	연속형
	연속형 변수(Continuous Variable)	숫자	연속형

4.3 결측값 처리(Missing Value)

Column names									
	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

© GeeksforGeeks

(<https://www.geeksforgeeks.org/creating-a-pandas-dataframe/>)

- 목표: 결측값이란 값이 비어있는 것 (NaN)을 의미하며, 알고리즘 작동을 어렵게 하고 작동이 되어도 해석의 왜곡 가능성 존재하기 때문에 처리 필요

- 삭제: 결측값이 발생한 모든 변수(Column)를 삭제하거나 일부(Row)를 삭제

```
df.dropna(axis=0) # 행 삭제  
df.dropna(axis=1) # 열 삭제
```

- 대체: 결측값을 제외한 값들의 통계량으로 결측값을 대체

- 중심 통계량
 - 분포 기반 랜덤 추출
- ```
df.fillna(df.mean()) # 평균치로 대체
df.fillna(df.median()) # 중앙값으로 대체
df.fillna(df.mode()) # 최빈값으로 대체
```

- 예측: 별도 분석을 통해 결측값을 예측하여 삽입

- Interpolation
- Regression Imputation
- EM Algorithm

```
df.interpolate(method='linear') # 선형방식으로 삽입
df.interpolate(method='time') # 인덱스 날짜고려 선형방식으로 삽입
df.interpolate(method='spline') # 비선형방식으로 삽입
df.interpolate(method='polynomial') # 비선형 다항식으로 삽입
```

| 결측치 비율    | 처리 방향              |
|-----------|--------------------|
| 10% 미만    | 삭제 또는 통계량기반 대체     |
| 10% ~ 30% | 모델링기반 예측           |
| 30% 이상    | 변수의 완전성/신뢰성 문제로 삭제 |

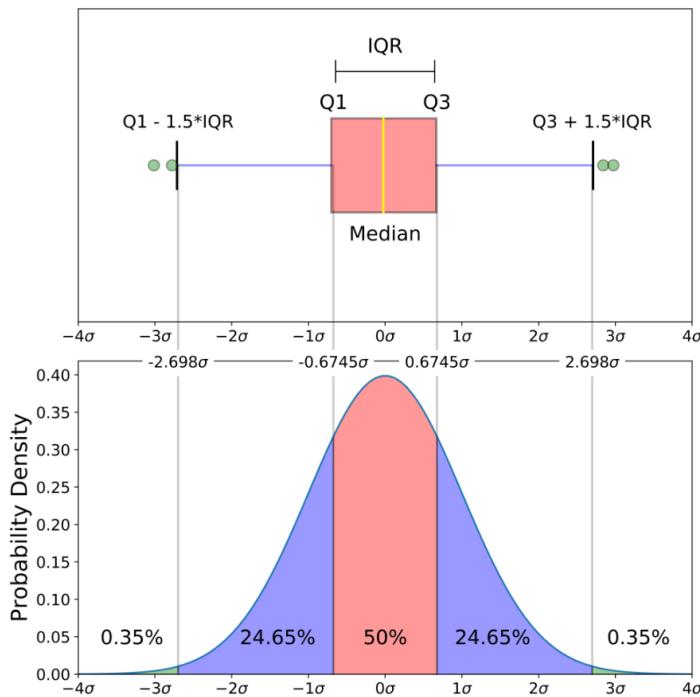
(Hair et al., 2016)

## 4.4 이상치 처리(Outlier)

- 목표: 일반적인 데이터와 동떨어진 관측치로 분석 결과를 왜곡할 가능성을 줄이는 것

### 1) 검출:

- 전통적으로 분포(Boxplot / Histogram / Scatter Plot 등)의 중심에서 벗어난 값을 지칭
- 빅데이터 시대에서는 이상치의 존재 유무? 논란이 존재



### 2) 처리:

- 삭제: Human Error 등은 보통 삭제 처리
- 대체: 스몰데이터의 경우 삭제시 데이터의 양이 적어지기에 다른 값으로 대체
- 예측: 별도 분석을 통해 이상치 대신 예측값으로 반영
- 변수화: 이상치를 변수화하여 유의성을 판별
- 별도 분석: 이상치 포함 분석과 이상치 미포함 분석을 병행 진행

## 4.5 데이터 분리(Data Split)

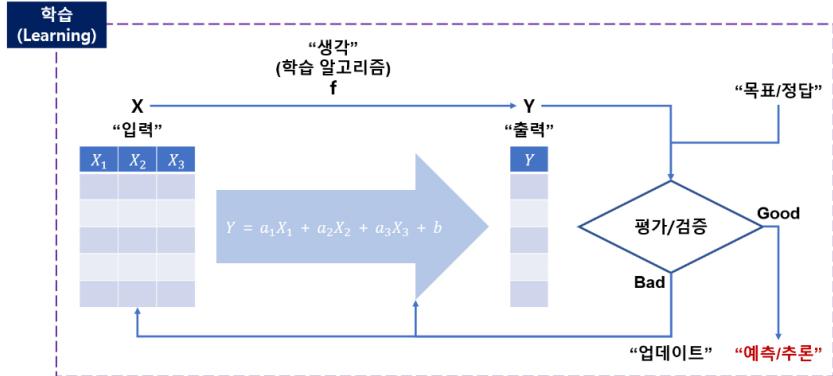
- 배경:

### (1) 독립변수와 종속변수 구분

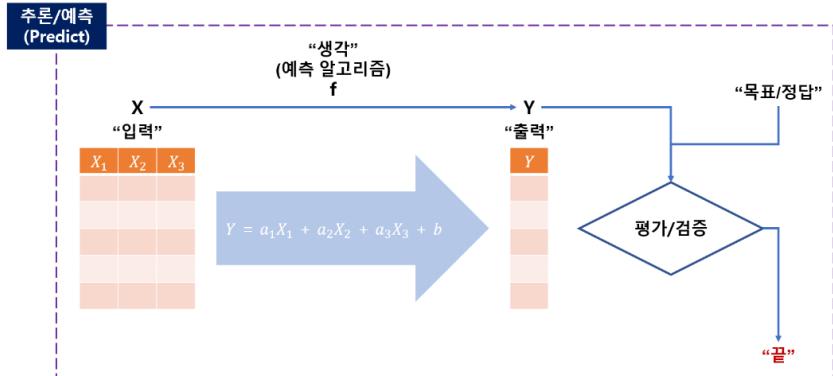
| 대분류                        | 의미/예시                   |
|----------------------------|-------------------------|
| 독립변수(Independent Variable) | 다른 변수에 영향을 미치는 변수 (X)   |
| 종속변수(Dependent Variable)   | 다른 변수에 의해 영향을 받는 변수 (Y) |

### (2) 과거/현재와 미래 기간 구분: 과거/현재의 상황을 분석하고, 미래를 예측 할 수 있는 환경 구축

▪ **Training Period:** 과거/현재의 상황을 분석



▪ **Testing Period:** 미래를 예측 할 수 있는 환경



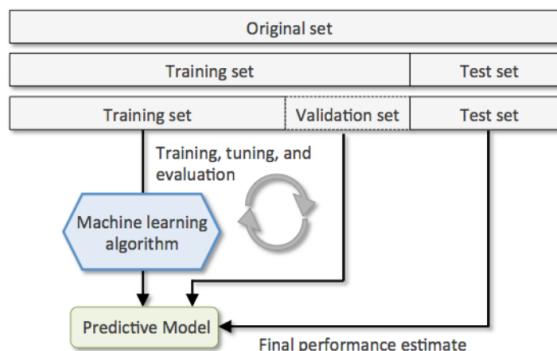
1) 간단한 방법(Holdout Validation):

- 훈련셋(Training set): 일반적으로 전체 데이터의 70% 사용
- 테스트셋(Testing set): 일반적으로 전체 데이터의 30% 사용

2) 일반적 방법(Simple Validation):

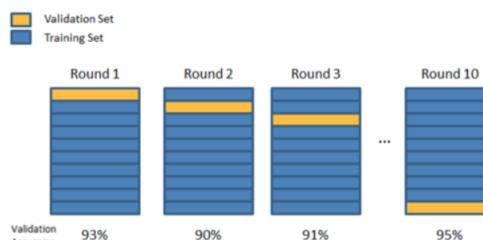
- 훈련셋(Training set): 일반적으로 전체 데이터의 60%를 사용
- 검증셋(Validation set):
  - 개발셋이라고도 하며, 일반적으로 전체 데이터의 20%를 사용함
  - 훈련된 여러가지 모델들의 성능을 테스트하는데 사용되며 모델 선택의 기준이 됨

- 테스트셋(Testing set): 일반적으로 전체 데이터의 20%를 사용하여 최종 모델의 정확성을 확인하는 목적으로 사용됨



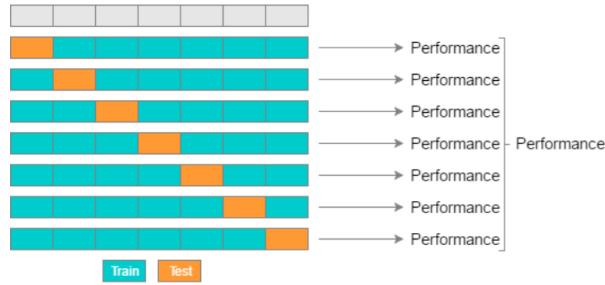
3) K 교차검사(K-fold Cross Validation):

- (1) 훈련셋을 복원없이  $K$  개로 분리한 후,  $K-1$  는 하위훈련셋으로 나머지 1개는 검증셋으로 사용함
- (2) 검증셋과 하위훈련셋을 번갈아가면서  $K$ 번 반복하여 각 모델별로  $K$  개의 성능 추정치를 계산
- (3)  $K$ 개의 성능 추정치 평균을 최종 모델 성능 기준으로 사용

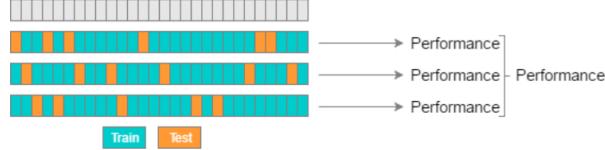


## 4) K-fold vs. Random-subsamples vs. Leave-one-out vs. Leave-p-out

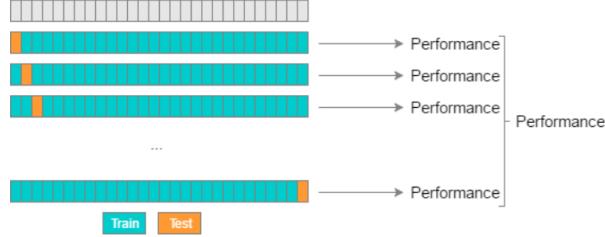
## • K-fold



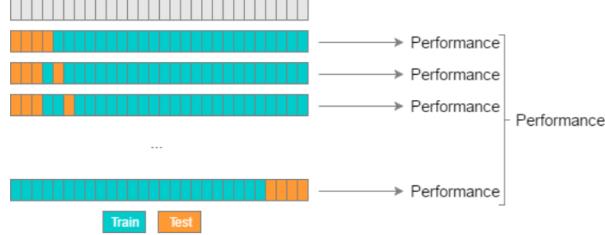
## • Random-subsamples



## • Leave-one-out



## • Leave-p-out



## 4.6 데이터 변환(Categorical Features)

- 목표: 컴퓨터와 알고리즘이 이해하도록 숫자 형태로 변환

| 대분류                         | 소분류 | 컴퓨터의 분류1 | 컴퓨터의 분류2 |
|-----------------------------|-----|----------|----------|
| 질적변수(Qualitative Variable)  | -   | -        | 범주형      |
| 명목형 변수(Nominal Variable)    | 문자  | 문자       | 범주형      |
| 순위형 변수(Ordinal Variable)    | 숫자  | 숫자       | 범주형      |
| 양적변수(Quantitative Variable) | -   | -        | 연속형      |
| 이산형 변수(Discrete Variable)   | 숫자  | 숫자       | 연속형      |
| 연속형 변수(Continuous Variable) | 숫자  | 숫자       | 연속형      |

- 문자형: 일반적으로 유한개면 범주형 숫자, 무한개면 연속형 숫자로 변환
- 숫자형: 해석 목적으로 따라 범주형→연속형 또는 연속형→범주형으로 변환
- 연속형 변수들은 대부분 알고리즘에서 자동으로 처리됨
- 기계학습(Machine Learning)은 범주형 데이터를 처리하는데서 출발

## 1) Binning(구간화): 연속형 변수를 범주형 변수로 변환

- 숫자로 구성된 연속형 값이 넓을 경우 그룹을 지어 이해도를 높임
- 변수의 선형적 특성 이외에 비선형적 특성을 반영

## 2) Label Encoding: 범주형 변수의 값들을 숫자 값(레이블)로 변경

"기존 변수" → "변환 변수"

| 기존 변수 | 변환 변수 |
|-------|-------|
| 계절    | 계절    |
| 여름    | 1     |
| 봄     | 0     |
| 봄     | 0     |
| 겨울    | 3     |
| 가을    | 2     |
| 가을    | 2     |

3) Dummy Variable(가변수,  $D_i$ ): 범주형 변수를 0 또는 1값을 가진 하나 이상의 새로운 변수로 변경(One-hot Encoding)

- 생성법: 계절변수 가 봄/여름/가을/겨울이라는 값을 포함하는 경우, 계절\_봄, 계절\_여름, 계절\_가을, 계절\_겨울 총 4개의 변수를 생성

- (1) 범주형 변수의 독립 값을 확인 (봄/여름/가을/겨울)
- (2) 독립 값의 갯수만큼 더미변수를 생성 ( $D_1 =$ 봄,  $D_2 =$ 여름,  $D_3 =$ 가을,  $D_4 =$ 겨울)

더미변수의 갯수는 최대 1개까지 줄일 수 있음

- (3) 각 더미변수들의 값은 변수의 정의와 같으면 1이고 나머지는 0으로 채움

“기존 변수”

| 계절 |
|----|
| 여름 |
| 봄  |
| 봄  |
| 겨울 |
| 가을 |
| 가을 |
| 여름 |

“가변수”

| 계절_봄 | 계절_여름 | 계절_가을 | 계절_겨울 |
|------|-------|-------|-------|
| 0    | 1     | 0     | 0     |
| 1    | 0     | 0     | 0     |
| 1    | 0     | 0     | 0     |
| 0    | 0     | 0     | 1     |
| 0    | 0     | 1     | 0     |
| 0    | 0     | 1     | 0     |
| 0    | 1     | 0     | 0     |

## 4.7 스케일 조정(Scaling)

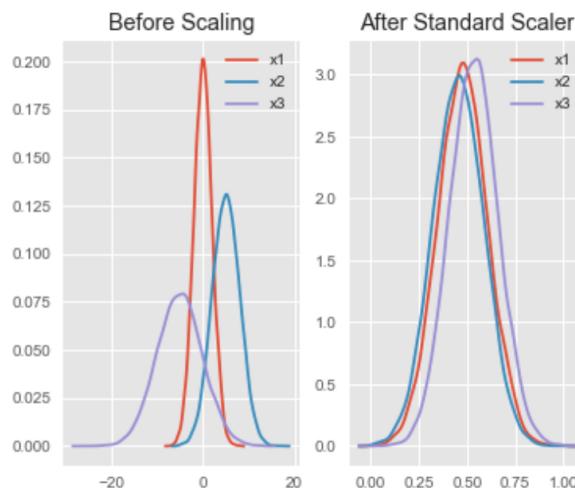
- 목적: 변수들의 크기를 일정하게 맞추어 크기 때문에 영향이 높은 현상을 회피
- 수학적: 독립 변수의 공분산 행렬 조건수(Condition Number)를 감소 시켜 최적화 안정성 및 수렴 속도 향상
- 컴퓨터적: PC 메모리를 고려하여 오버플로우(Overflow)나 언더플로우(Underflow)를 줄여줌

### 1) Standard Scaler:

$$\frac{X_{it} - E(X_i)}{SD(X_i)}$$

- 기본 스케일로 평균을 제외하고 표준편차를 나누어 변환
- 각 변수(Feature)가 정규분포를 따른다는 가정 아래에 정규분포가 아닐 시 최선이 아닐 수 있음

```
sklearn.preprocessing.StandardScaler().fit()
sklearn.preprocessing.StandardScaler().transform()
sklearn.preprocessing.StandardScaler().fit_transform()
```



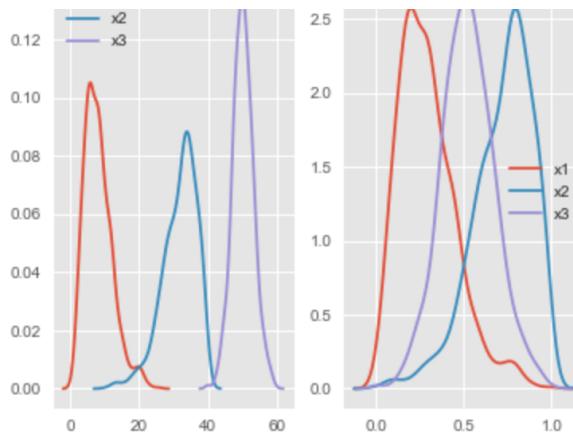
### 2) Min-Max Scaler:

$$\frac{X_{it} - \min(X_i)}{\max(X_i) - \min(X_i)}$$

- 가장 많이 활용되는 방식으로 최소~최대 값이 0~1 또는 -1~1 사이의 값으로 변환
- 각 변수(Feature)가 정규분포가 아니거나 표준편차가 매우 작을 때 효과적

```
sklearn.preprocessing.MinMaxScaler().fit()
sklearn.preprocessing.MinMaxScaler().transform()
sklearn.preprocessing.MinMaxScaler().fit_transform()
```



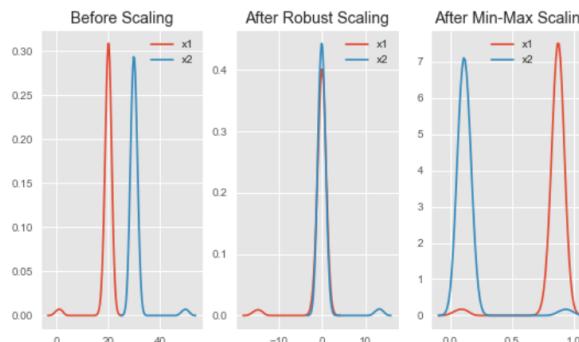


### 3) Robust Scaler:

$$\frac{X_{it} - Q_1(X_i)}{Q_3(X_i) - Q_1(X_i)}$$

- 최소-최대 스케일러와 유사하지만 최소/최대 대신에 IQR(Interquartile Range) 중 25%값/75%값을 사용하여 변환
- 이상치(Outlier)에 영향을 최소화하였기에 이상치가 있는 데이터에 효과적이고 적은 데이터에도 효과적인 편

```
sklearn.preprocessing.RobustScaler().fit()
sklearn.preprocessing.RobustScaler().transform()
sklearn.preprocessing.RobustScaler().fit_transform()
```

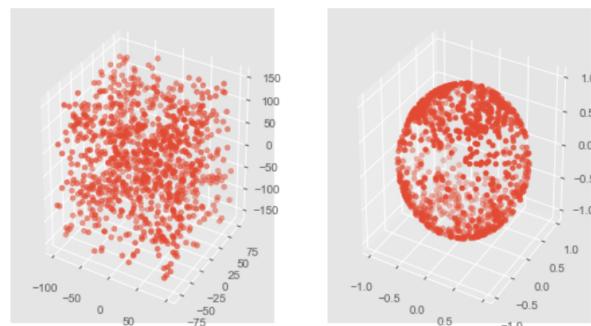


### 4) Normalizer:

$$\frac{X_{it}}{\sqrt{X_i^2 + X_j^2 + \dots + X_k^2}}$$

- 각 변수(Feature)를 전체 n 개 모든 변수들의 크기들로 나누어서 변환(by Cartesian Coordinates)
- 각 변수들의 값은 원점으로부터 반지름 1만큼 떨어진 범위 내로 변환

```
sklearn.preprocessing.Normalizer().fit()
sklearn.preprocessing.Normalizer().transform()
sklearn.preprocessing.Normalizer().fit_transform()
```

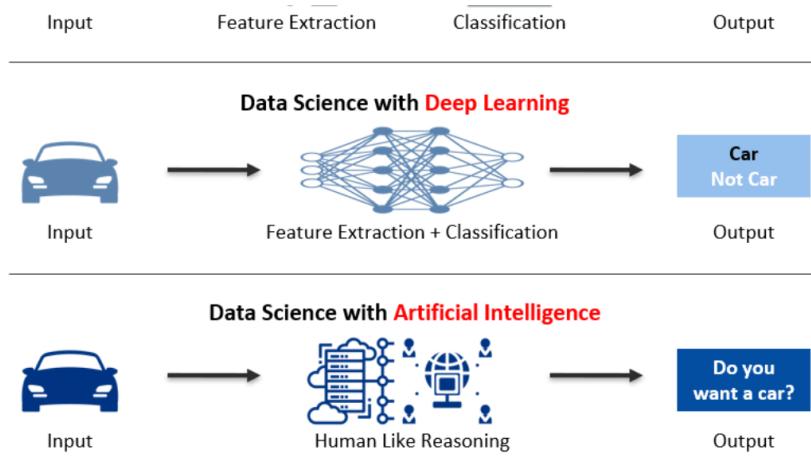


## 4.8 참고

- 데이터 과학자들은 보통 수동/자동 변수 처리 및 변환(Feature Engineering)에 익숙하지만, 새로운 변수를 생성하는 것은 분석에서 가장 중요하고 시간이 많이 걸리는 작업 중 하나이며, 머신러닝과 딥러닝의 발전으로 점차 자동화 중

### Data Science with Machine Learning





"변수 생성시 주의할 점!"

- 1. 미래의 실제 종속변수 예측값이 어떤 독립/종속변수의 Feature Engineering에 의해 효과가 있을지 단정할 수 없음
- 2. 독립변수의 예측값을 Feature Engineering을 통해 생성될 수 있지만 이는 종속변수의 예측에 오류증가를 야기할 수 있음

## 5 함수세팅 및 추정 방향(Modeling): Linear Regression

### 5.1 머신러닝의 배경과 등장

"인공지능(Artificial Intelligence)은 여러 의미를 포괄하지만, 일반적으로 기계학습(Machine Learning) + 딥러닝(Deep Learning)을 의미"

#### 1) 배경: 기존 사람과 프로그래밍 접근방식 한계

- 예시: 주어진 사진에서 고양이와 강아지를 판단하기



(<https://codong.tistory.com/37?category=952287>)

- 사람: 높은 정확도로 직관적으로 분류 가능
- 프로그래밍: 수많은 특징을 규칙으로 사람이 모두 작성하는 것은 거의 불가능

```
def prediction(이미지파일):
 if 눈코귀가 있을 때:
 if 근데 강아지는 아닐 때:
 if 털이 있고 꼬리 있을 때:
 if 다른 동물이 아닐 때:
 ...
 어제 하누 ...
 return 결과
```

- 프로그래밍 고도화: 특징규칙들을 수학적으로 고도화 해왔지만 한계

#### 2) 머신러닝의 등장: 사람이 규칙을 일일이 정의 <<< 머신러닝으로 규칙을 탐색

"데이터에 규칙을 적용시켜 결과를 찾는 것 이 아니라 데이터와 결과를 학습하여 규칙을 찾는 것 으로, 기존 프로그래밍 방식의 한계를 해결하게 된 새로운 계기"

### 일반 프로그래밍



## 인공지능(머신러닝+딥러닝)



- 학습(Learning): 주어진 데이터를 기계/컴퓨터에 학습시켜 규칙성을 찾는 과정
- 일반적으로 과거데이터를 학습하기 때문에 훈련(Training)이라고도 하며, 이렇게 발견된 규칙성으로 새로운 미래데이터에 적용하여 정답을 추정(Testing)
- 예를들어, 구글 번역기는 사람이 직접 규칙을 정의한 것이 아니라, 딥러닝이 스스로 규칙을 찾아 높은 성능으로 번역 수행

• 예시:



"인공지능, 기계학습 그리고 딥러닝" 강의자료 15p 일부 변경, <https://www.slideshare.net/JinwonLee9/ss-70446412>

## 5.2 선형회귀분석 작동방식으로 머신러닝 이해

"선형회귀분석을 포함하여 머신러닝과 딥러닝 등의 모든 알고리즘은 큰 틀에서 작동방식이 동일."

"머신러닝과 딥러닝의 작동방식을 이해하기 위해 가장 기초 예측 알고리즘인 선형회귀분석(Linear Regression) 작동방식부터"

"선형회귀분석을 포함한 대부분의 알고리즘은 큰 틀에서 3가지 도구를 사용하여 작동"

1) 방정식(Equation) = 함수(Function) = 가설(Hypothesis)

2) 비용함수(Cost Function)

3) 옵티마이저(Optimizer)

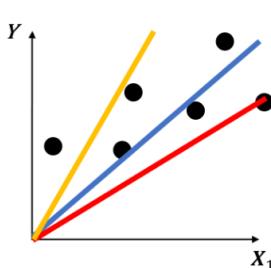
0) 선형회귀분석(Linear Regression): 어떤 변수 값에 따라 다른 변수 값이 영향을 받는 관계성을 분석

- X(독립변수): 다른 변수 값을 변하게 하는 변수
- Y(종속변수): 변수 X에 의해서 종속되어 변하는 변수
- 선형회귀분석: 1개 이상의 독립변수 X와 종속변수 Y의 관계를 모델링

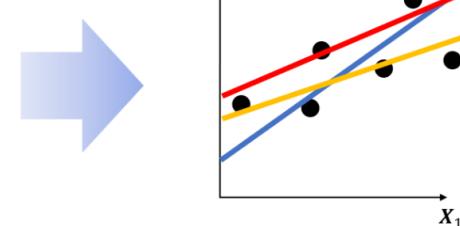
| 관계로직                        | 관계식                                          |
|-----------------------------|----------------------------------------------|
| $Y$ 는 $X$ 와 같다              | $Y = X$                                      |
| $Y$ 는 초기값과 $X$ 들의 비율의 합과 같다 | $Y = w_0 + w_1X_1 + w_2X_2 + \dots + w_kX_k$ |

- 편향(Bias) = Y절편(Y-intercept) = 초기값 =  $w_0$
- 가중치(Weight) = 기울기 = 비율 =  $w_1, w_2, \dots, w_k$

✓ 편향/상수변수가 없을 때



✓ 편향/상수변수가 있을 때  
: 자유도 및 정확성 증가



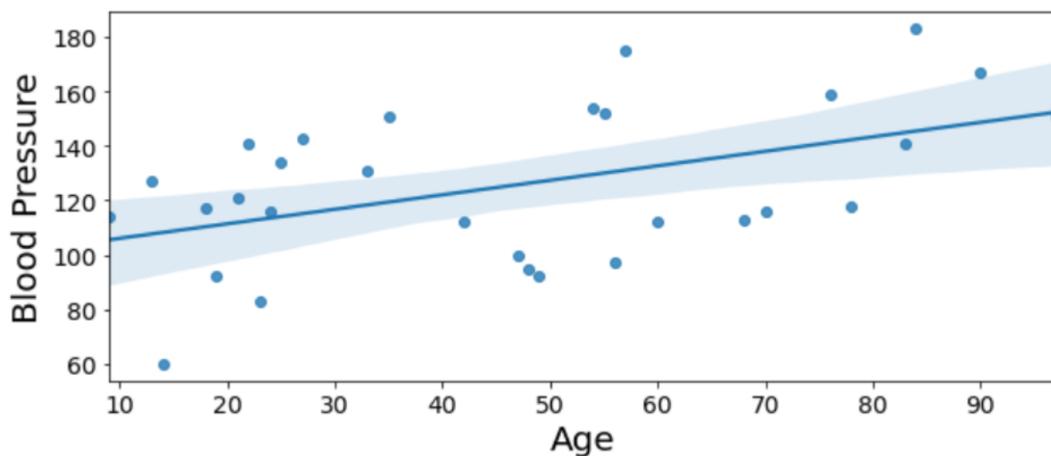
1) 방정식(Equation) = 함수(Function) = 가설(Hypothesis)

- **회귀문제:** 나이에 따라서 혈압이 어떤 관련? / 공부 시간에 따라 성적은 어떤 관련?
- **가설(Hypothesis):** 머신러닝에서는 이런 관련성을 표현한 관계식

| 종류                         | 가설: $H(X)$                                   |
|----------------------------|----------------------------------------------|
| Multiple Linear Regression | $Y = w_0 + w_1X_1 + w_2X_2 + \dots + w_kX_k$ |
| Simple Linear Regression   | $Y = w_0 + w_1X_1$                           |

- 예시: 나이에 따라서 혈압이 어떤 관련?

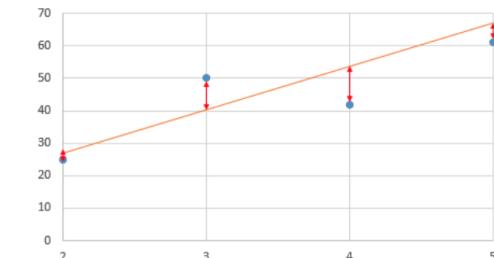
| Age | Blood Pressure |
|-----|----------------|
| 0   | 33             |
| 1   | 18             |
| 2   | 57             |
| 3   | 70             |
| 4   | 60             |
|     | 131            |
|     | 117            |
|     | 175            |
|     | 116            |
|     | 112            |



⇒ "선형회귀분석은 주어진 데이터로부터  $Y$ 와  $X$ 의 관계를 가장 잘 나타내는 직선을 찾는 것 또는 가설을 검증하는 것 또는 가중치와 편향을 추정하는 것"

## 2) 비용함수(Cost Function): 어떻게 가설을 검증? + 가중치와 편향을 추정?

- **아이디어:** 데이터를 사용하여 가설로부터 얻은 예측값과 실제값의 차이를 최소화하는 방향으로 가설 또는 가중치와 편향을 업데이트
- **오차(Error):** 예측값과 실제값의 차이



(임의 직선/가설)가중치에 따른 오차의 크기, <https://wikidocs.net/21670>)

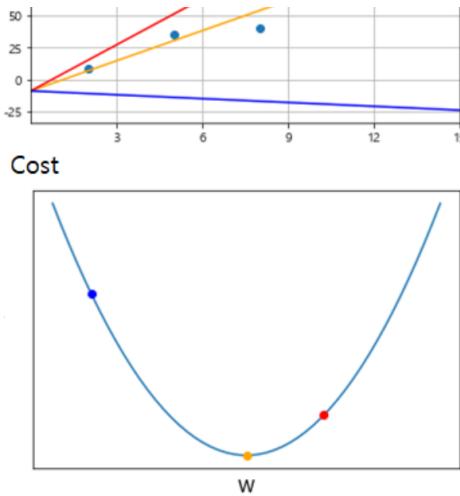
- **오차를 반영한 관계식:** 비용함수(Cost Function) = 손실함수(Loss Function)

- 단순히 오차를 반영하는 것 뿐만 아니라 오차를 줄이는 일에 최적화 된 식
- 다양한 문제들마다 적합한 비용함수들이 있을 수 있음
- 회귀문제의 경우 주로 평균 제곱 오차(Mean Squared Error, MSE)가 사용

$$\text{Cost Function} = \sum_{i=1}^m \left[ \sum_{j=0}^k (Y_i - w_j X_j)^2 \right]$$

- 오차의 크기를 측정하기 위해 오차의 부호와 무관하도록 제곱하여 더하여 절대적 크기 추정
- 오차가 클수록 비용함수가 커지기 때문에, 결과적으로 비용함수가 최소화되는 가중치와 편향을 추정하면  $Y$ 와  $X$ 의 관계를 가장 잘 나타내는 직선





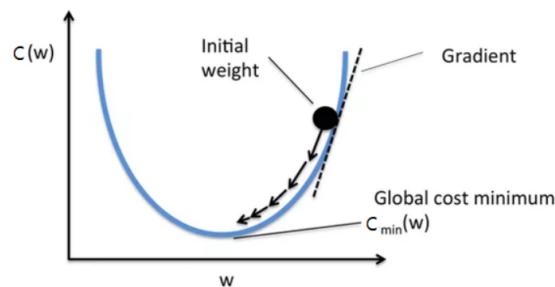
(임의 직선/가설/가중치에 따른 오차의 크기 변화, <https://realblack0.github.io/2020/03/27/linear-regression.html>)

⇒ "선형회귀분석은 주어진 데이터로부터  $Y$ 와  $X$ 의 관계를 가장 잘 나타내는 가중치와 편향을 추정하기 위해 비용함수를 최소화하면 모든 데이터에 위치적으로 가장 가까운 직선이 추정됨"

### 3) 옵티마이저(Optimizer) : 어떻게 비용함수를 최소화?

- 선형회귀분석을 포함한 수많은 머신러닝과 딥러닝 알고리즘은 결국 비용함수를 최소화하는 작업
- 비용함수를 최소화시키는 알고리즘을 최적화 알고리즘 = 옵티마이저(Optimizer)
- 학습(Learning): 데이터로 비용함수를 추정하며 최적화 또는 옵티마이저를 통해 적절한 가중치와 편향을 찾아내는 과정
- 경사하강법(Gradient Descent): 가장 기본적인 옵티마이저

- 비용함수가 가장 최소값을 갖게 하는 가중치  $W$ 를 찾는 알고리즘
- 임의의 기울기(Gradient) = 미분값  $W_0$ 에서 시작하여 기울기를 낮추다가 0이 될 때까지 가중치 업데이트 하기에 Gradient Descent



- 수학적 추정 과정: 틀린 초기 가중치를 여러번 반복해서 업데이트

$$\begin{aligned}
 W_1 &:= W_0 - \alpha \frac{\partial}{\partial w} [\text{Cost Function}] \\
 &= W_0 - \alpha \frac{\partial C(W)}{\partial W} \Big|_{W=W_0} \\
 W_2 &:= W_1 - \alpha \frac{\partial C(W)}{\partial W} \Big|_{W=W_1} \\
 W_3 &:= W_2 - \alpha \frac{\partial C(W)}{\partial W} \Big|_{W=W_2} \\
 &\vdots \\
 W &:= W - \alpha \frac{\partial C(W)}{\partial W}
 \end{aligned}$$

- $\alpha$ : 가중치를 업데이트하는 속도로 학습률(Learning Rate)이라고 하며, 속도가 빠르면 정확성이 낮아질 수 있고 속도가 느리면 오래걸리기 예상되는 수치 필요

"기울기가 가장 작은/0인 곳을 한번에/수학적으로 안찾고, 왜 결국 틀린 가중치를 여러번 반복 해서 찾으며 업데이트하나요?"

- 수학적으로 비용함수를 미분하고 0인 지점을 찾으면 되지만 모든 경우 가능하지 않음
- 현실문제에 적합한 비용함수는 알수가 없는 면이고 창의적인 영역 + 기울기(미분)를 꼭 계산할 수 있지 않음
- 사람은 반복이 귀찮지만 컴퓨터/기계는 반복이 쉽고 틀린 가중치를 반복적으로 업데이트하는 것 이 완벽한 정답은 아니지만 정답에 가까운 근사치이며 정확성을 별도 추정하여 신뢰도 의사결정

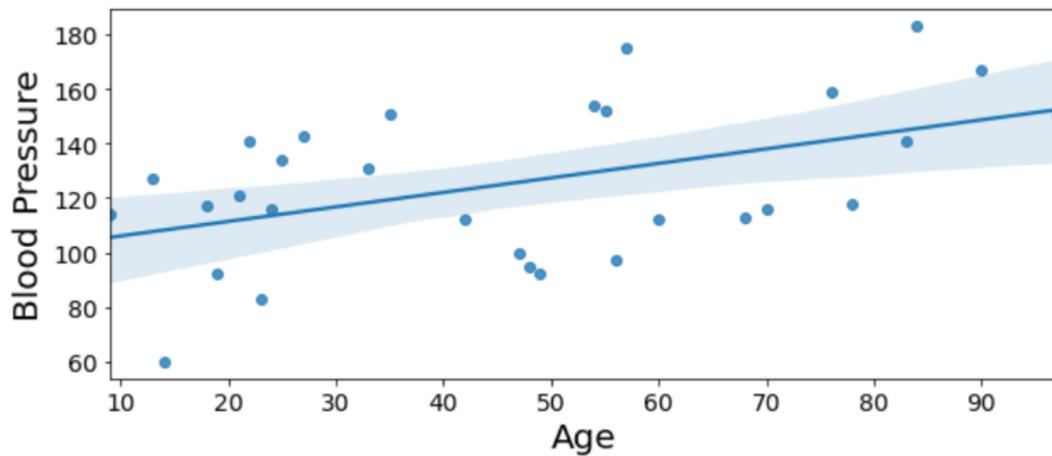
## 5.3 회귀문제 해결을 위한 가설 및 비용함수

### 1) 알고리즘 함수세팅:

$$Y \approx \hat{Y} = f(X_1, X_2, \dots, X_k) = w_0 + w_1 X_1 + w_2 X_2 + \dots + w_k X_k = [w_0 \ w_1 \ w_2 \ \dots \ w_k] \begin{bmatrix} 1 \\ X_1 \\ X_2 \\ \vdots \\ X_k \end{bmatrix}$$

$$= [1 \ X_1 \ X_2 \ \dots \ X_k] \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix} = \begin{bmatrix} 1 & X_{11} & X_{21} & \dots & X_{k1} \\ 1 & X_{12} & X_{22} & \dots & X_{k2} \\ \vdots & & & & \\ 1 & X_{1t} & X_{2t} & \dots & X_{kt} \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix} = XW = WX$$

|   | Age | Blood Pressure |
|---|-----|----------------|
| 0 | 33  | 131            |
| 1 | 18  | 117            |
| 2 | 57  | 175            |
| 3 | 70  | 116            |
| 4 | 60  | 112            |



2) 함수 추정을 위한 비용함수: 나의 주장 기반 알고리즘의 예측값 ( $\hat{Y}$ )과 실제 데이터 ( $Y$ )의 차이를 평가하는 함수

- 손실함수(Loss Function): 하나의 데이터(Single Row)에서 예측값과 정답의 차이를 평가
- 비용함수(Cost Function): 모든 데이터에서 예측값과 정답의 차이를 평가

$$Y - \hat{Y} = Y - WX = \text{residual} = \text{cost}$$

$$= \sum_{i=1}^m \left[ \sum_{j=1}^k (Y_i - w_j X_j) \right]$$

- 회귀분석은 여러가지의 비용함수 중 최소제곱법/최소자승법을 사용
- 최소제곱법/최소자승법을 최소로 하는 직선을 추정하여 계수(coefficient)를 결정

$$\hat{W} = \arg \min_W \sum_{i=1}^m \left[ \sum_{j=1}^k (Y_i - w_j X_j)^2 \right]$$

## 5.4 결정론적 모형(Deterministic Model): 수학적 모형

"잔차제곱합(Residual Sum of Squares)을 최소로하는  $W$ 를 추정"

1) 잔차벡터(Residual Vector):

$$\epsilon = Y - \hat{Y} = Y - XW$$

2) 잔차제곱합(Residual Sum of Squares):

$$RSS = \epsilon^T \epsilon = (Y - XW)^T (Y - XW)$$

$$= Y^T Y - 2Y^T XW + W^T X^T XW$$

3) 잔차제곱합의 기울기 추정: 그레디언트(미분, 기울기)

$$\frac{dRSS}{dW} = -2X^T Y + 2X^T XW$$

4) 잔차제곱합이 최소가 되는 계수는 그레디언트가 0이 되는 곳: 최적화 알고리즘의 작동원리

$$\frac{dRSS}{dW} = 0$$

5) 최적화 실행:

$$dRSS = \dots$$

$$\frac{dW}{dW} = -2X^T Y + 2X^T X W = 0$$

$$X^T X W = X^T Y$$

6) 추정 계수:

$$W = (X^T X)^{-1} X^T Y$$

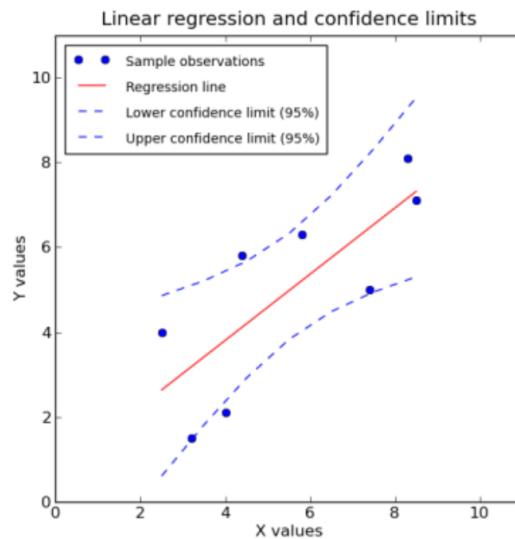
• 특징:

- $X^T X$  행렬의 역행렬이 존재해야 해 추정/존재 가능
- 역행렬이 미존재
  - =  $X$ 가 서로 독립이 아님
  - =  $X$ 가 Full Rank가 아님
  - =  $X^T X$ 가 양의 정부호(Positive Definite)가 아님

## 5.5 확률론적 모형(Probabilistic Model): 통계적 모형

"증속변수의 발생가능성을 최대(최소)로하는  $W$ 를 추정"

- 필요성: 결정론적 방식은 데이터의 확률적 가정이 없기에 1회성으로 가중치를 추정(점추정) 하나, 이 반복추정으로 가중치가 발생할 범위(구간추정)는 알 수 없음



- 예시: 집값에 대한 범죄율 영향력(가중치)이  $-1.08$  이라면, 범죄율이 높은 곳은 집값이 떨어진다 결론 내릴 수 있을까?

- $-1.08$ 는 1회성 결과일 뿐 오차가 준재
- 만약 오차가  $0.1$ 이라면 실제 영향력의 추정 범위(신뢰구간)는  $-1.08 \pm 0.1$  ( $-1.18 \sim -0.98$ )이기에 범죄율이 높은 곳은 집값이 떨어진다 결론 가능
- 만약 오차가  $2$ 라면 실제 영향력의 추정 절위(신뢰구간)는  $-1.08 \pm 2$  ( $-3.08 \sim 0.92$ )이기에 범죄율이 높은 곳은 집값이 떨어질 수도 오를 수도 있다 결론 가능

0) 실제  $Y$  추정값의 확률론적 표현:

Main Equation

$$\begin{aligned} Y &\approx \hat{Y} = f(X_1, X_2, \dots, X_k) \\ &= w_0 + w_1 X_1 + w_2 X_2 + \dots + w_k X_k \\ &= E(Y|X_1, X_2, \dots, X_k) \\ &\sim \mathcal{N}(XW, \sigma^2) \\ Pr(Y | X, \theta) &= \mathcal{N}(y | XW, \sigma^2) \end{aligned}$$

1) 실제  $Y$  값의 추정가능성(Likelihood):

$$\begin{aligned} Pr(Y_i | X_i, \theta) &= \prod_{i=1}^N \mathcal{N}(Y_i | X_i w_i, \sigma^2) \\ &= \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(Y_i - X_i w_i)^2}{2\sigma^2} \right\} \end{aligned}$$

2) 추정가능성의 더하기 표시 변환을 위한 Log함수 적용(Log-Likelihood):

$$\begin{aligned} LL &= \log Pr(Y_i | X_i, \theta) \\ &= \log \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(Y_i - X_i w_i)^2}{2\sigma^2} \right\} \\ &= -\frac{1}{2\sigma^2} \sum_{i=1}^N (Y_i - X_i w_i)^2 - \frac{N}{2} \log 2\pi\sigma^2 \\ LL(\text{Matrix Form}) &= -C_1 (Y - XW)^T (y - XW) - C_0 \end{aligned}$$

$$= -C_1(W^T X^T X W - 2Y^T X W + Y^T Y) - C_0$$

$$\text{where } C_1 = \frac{1}{2\sigma^2}, C_0 = \frac{N}{2} \log 2\pi\sigma^2$$

3) Log-Likelihood의 음수화 및 그레디언트가 0이 되는 곳:

$$-\frac{d}{dW} \text{LL} = C_1 (2X^T X W - 2X^T Y) = 0$$

$$W = (X^T X)^{-1} X^T Y$$

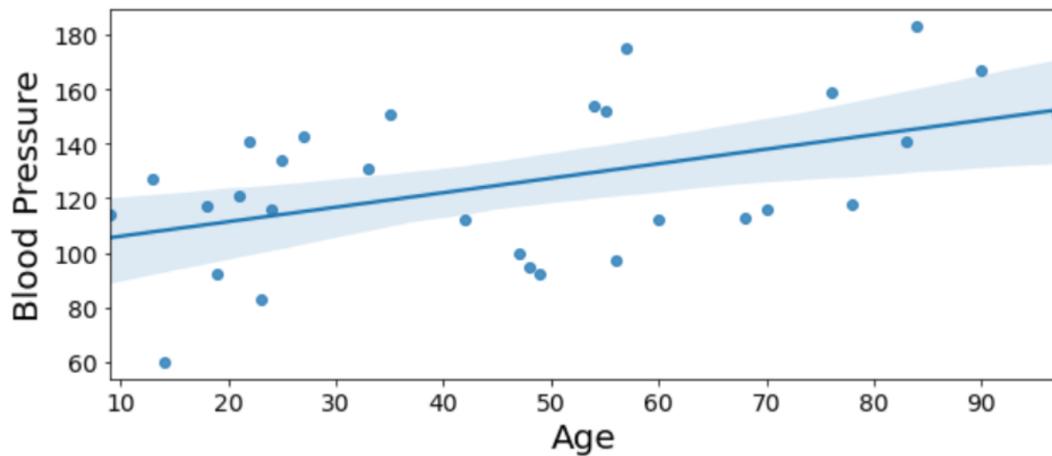
• 특징:

- $X^T X$  행렬의 역행렬이 존재해야 해 추정/존재 가능
- 역행렬이 미존재
  - =  $X$ 가 서로 독립이 아님
  - =  $X$ 가 Full Rank가 아님
  - =  $X^T X$ 가 양의 정부호(Positive Definite)가 아님

## 5.6 수학통계적 vs 머신러닝기계

Age Blood Pressure

|   |    |     |
|---|----|-----|
| 0 | 33 | 131 |
| 1 | 18 | 117 |
| 2 | 57 | 175 |
| 3 | 70 | 116 |
| 4 | 60 | 112 |



```
In [5]: # 분석 라이브러리 불러오기
import warnings
warnings.filterwarnings('ignore') # 'always'
import numpy as np
import pandas as pd
import statsmodels.api as sm
```

```
In [6]: # 예제 데이터
df = pd.read_csv(r'./Data/MedicalCheckup/hme.csv')
Y = df[['BloodPressure_Max']]
X = df[['Age']]
X.loc[:, 'Constant'] = 1
X = X[['Constant', 'Age']]
pd.concat([Y, X], axis=1)
```

```
Out[6]:
```

|     | BloodPressure_Max | Constant | Age |
|-----|-------------------|----------|-----|
| 0   | 118               | 1        | 8   |
| 1   | 134               | 1        | 50  |
| 2   | 129               | 1        | 55  |
| 3   | 137               | 1        | 30  |
| 4   | 92                | 1        | 45  |
| ... | ...               | ...      | ... |
| 995 | 117               | 1        | 60  |
| 996 | 122               | 1        | 60  |
| 997 | 104               | 1        | 55  |
| 998 | 133               | 1        | 70  |
| 999 | 121               | 1        | 30  |

1000 rows × 3 columns

```
In [7]: # 수학적 계산을 위한 가중치 계산
관계식이 복잡해진다면? 비용함수가 물려진다면? 일반화 가능한가?
매번 손으로 기계 대신 미분을 할 수 있는가?
weight = np.matmul(np.matmul(np.linalg.inv(np.matmul(X.T, X)), X.T), Y)
weight.columns = ['weight']
weight.index = ['w0', 'w1']
weight
```

```
Out[7]: weight
```

|    |            |
|----|------------|
| w0 | 108.671945 |
| w1 | 0.259873   |

```
In [8]: # 머신러닝 기계로 가중치 계산
model = sm.OLS(Y, X).fit()
model.summary()
```

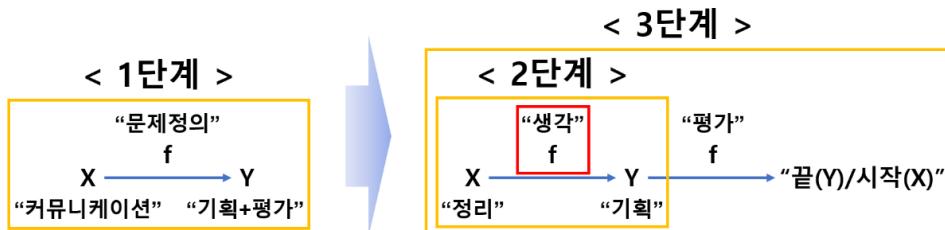
```
Out[8]: OLS Regression Results
```

| Dep. Variable:    | BloodPressure_Max | R-squared:          | 0.070    |       |         |         |
|-------------------|-------------------|---------------------|----------|-------|---------|---------|
| Model:            | OLS               | Adj. R-squared:     | 0.069    |       |         |         |
| Method:           | Least Squares     | F-statistic:        | 75.08    |       |         |         |
| Date:             | Sun, 18 Sep 2022  | Prob (F-statistic): | 1.80e-17 |       |         |         |
| Time:             | 00:03:24          | Log-Likelihood:     | -4014.1  |       |         |         |
| No. Observations: | 1000              | AIC:                | 8032.    |       |         |         |
| Df Residuals:     | 998               | BIC:                | 8042.    |       |         |         |
| Df Model:         | 1                 |                     |          |       |         |         |
| Covariance Type:  | nonrobust         |                     |          |       |         |         |
|                   | coef              | std err             | t        | P> t  | [0.025  | 0.975]  |
| Constant          | 108.6719          | 1.638               | 66.343   | 0.000 | 105.458 | 111.886 |
| Age               | 0.2599            | 0.030               | 8.665    | 0.000 | 0.201   | 0.319   |
| Omnibus:          | 51.299            | Durbin-Watson:      | 1.942    |       |         |         |
| Prob(Omnibus):    | 0.000             | Jarque-Bera (JB):   | 64.091   |       |         |         |
| Skew:             | 0.497             | Prob(JB):           | 1.21e-14 |       |         |         |
| Kurtosis:         | 3.742             | Cond. No.           | 211.     |       |         |         |

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

## 6 검증지표 방향(Evaluation Metrics)



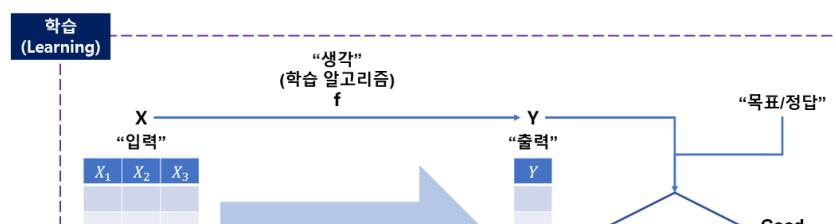
“문제해결 검증지표 와 알고리즘 검증지표 는 같을 수 있으나 대부분은 다른 편”

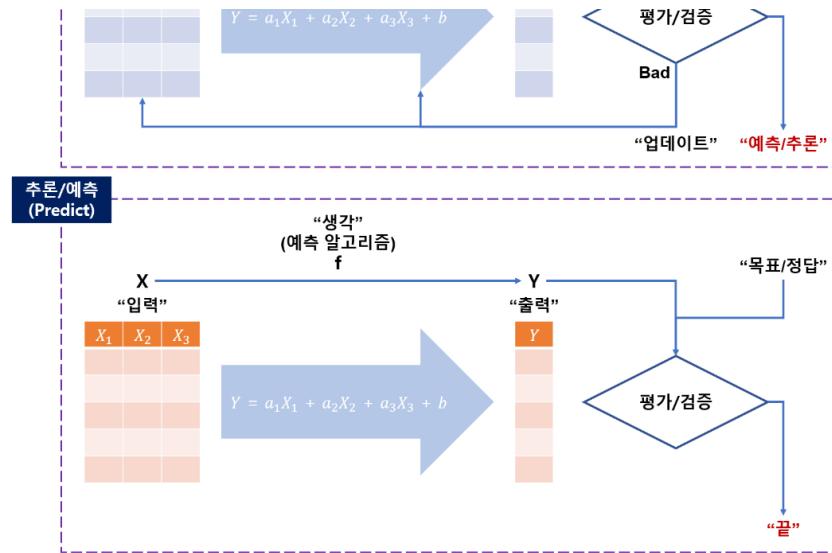
(1) 문제해결 검증지표: 실제 문제를 잘 해결하는지 평가 (3단계)

(2) 알고리즘 검증지표: 데이터의 패턴이 잘 추출되고 예측의 정확성을 평가 (2단계)

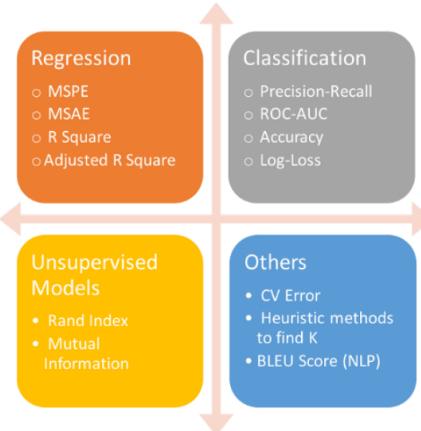
- 알고리즘 성능이 좋은 것은 문제해결이 가능한 것은 다르기 때문에 문제해결 지표와 알고리즘 지표는 대부분은 다른 편
- 알고리즘 검증지표는 없어도 되지만 문제해결 검증지표는 반드시 필요
- (이론적) 알고리즘들은 일반적으로 특정 알고리즘 검증지표를 향상시키는 방향으로 개발됨

### 6.1 대표적인 검증지표





### 1) 문제별 종류:



#### • Statistical Metrics: Correlation

- 입력(Input): 무한대 ~ 무한대 범위의 연속형 값
- 출력(Output): 이론적으로 -1 ~ 1 범위의 연속형 값

#### • Regression Metrics: MSE, MSPE, RMSE, RMSLE, MAE, MAPE, MPE, R^2, Adjusted R^2, ... (Y의 범위가 무한대가 가능한 연속형일 때)

- 입력(Input): 무한대 ~ 무한대 범위의 연속형 값
- 출력(Output): 이론적으로 0 ~ 무한대 범위의 연속형 값

#### • Classification Metrics: Log Loss, Cross-entropy, ROC, AUC, Gini, Confusion Matrix, Accuracy, Precision, Recall, F1-score, Classification Report, KS Statistic, Concordant-Discordant Ratio, (ARI, NMI, AMI), ... (Y가 2개 또는 그 이상개수의 이산형일 때)

- 입력(Input): 무한대 ~ 무한대 범위의 연속형 값
- 출력(Output): 알고리즘 종류에 따라 출력이 달라질 수 있음
  - 확률(Probability): 0 ~ 1 범위의 연속형 값 (Logistic Regression, Random Forest, Gradient Boosting, Adaboost, ...)
  - 집단(Class): 0 또는 1의 이산형 값 (SVM, KNN, ...)

#### • Clustering: Dunn Index, Silhouette, ...

#### • Ranking Metrics: Gain, Lift, MRR, DCG, NDCG, ...

#### • Computer Vision Metrics: PSNR, SSIM, IoU, ...

#### • NLP Metrics: Perplexity, BLEU score, ...

#### • Deep Learning Related Metrics: Inception score, Frechet Inception distance, ...

#### • Real Problem: ???

### 2) 검증지표 성능의 종류: 데이터/분석은 높은 정확도를 낳거나 높은 에러를 발생시킴

- 높은정확도(High Accuracy): 과거 패턴이 미래에도 그대로 유지가 된다면 예측 정확도가 높아짐

- 높은에러(High Error): 패턴이 점차적으로 또는 갑자기 변경되면 예측값은 실제값에서 크게 벗어날 수 있음

- Black Swan: 일어날 것 같지 않은 일이 일어나는 현상

- **White Swan:** 과거 경험들로 충분히 예상되는 위기지만 대응책이 없고 반복될 현상
- **Gray Swan:** 과거 경험들로 충분히 예상되지만 발생되면 충격이 지속되는 현상

## 6.2 회귀분석 검증지표 및 해석하기

1) 예측문제 검증지표: MSE, MSPE, RMSE, RMSLE, MAE, MAPE, MPE, R^2, Adjusted R^2, ... (Y의 범위가 무한대가 가능한 연속형일때)

The diagram shows two regression models and their corresponding error metrics:

- Linear Regression: Single Variable**:  $\hat{y} = \beta_0 + \beta_1 x + \epsilon$ . Labels: Predicted output, Coefficients, Input, Error.
- Linear Regression: Multiple Variables**:  $\hat{y} = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \epsilon$ . Labels: Divide by the total number of data points, Predicted output value, Actual output value, The absolute value of the residual.
- MAE (Mean Absolute Error)**:  $MAE = \frac{1}{n} \sum \left| y - \hat{y} \right|$ . Labels: Sum of, The absolute value of the residual.
- MSE (Mean Squared Error)**:  $MSE = \frac{1}{n} \sum \left( y - \hat{y} \right)^2$ . Labels: The square of the difference between actual and predicted.
- MAP E (Mean Absolute Percentage Error)**:  $MAP E = \frac{100\%}{n} \sum \left| \frac{\hat{y} - y}{y} \right|$ . Labels: Multiplying by 100% converts to percentage, The residual, Each residual is scaled against the actual value.
- MPE (Mean Percentage Error)**:  $MPE = \frac{100\%}{n} \sum \left( \frac{\hat{y} - y}{y} \right)$ .

- 사용 예시: [Comparison of Algorithm Performance Metrics](#)

2) 회귀분석 알고리즘 적합/추정 후 결과지표: 추정성능(빨간색) + 잔차진단(보라색)

OLS Regression Results

| Dep. Variable:             | count            | R-squared:          | 0.390     |       |         |         |
|----------------------------|------------------|---------------------|-----------|-------|---------|---------|
| Model:                     | OLS              | Adj. R-squared:     | 0.390     |       |         |         |
| Method:                    | Least Squares    | F-statistic:        | 593.6     |       |         |         |
| Date:                      | Sat, 16 Feb 2019 | Prob (F-statistic): | 0.00      |       |         |         |
| Time:                      | 02:13:49         | Log-Likelihood:     | -81290.   |       |         |         |
| No. Observations:          | 13003            | AIC:                | 1.626e+05 |       |         |         |
| Df Residuals:              | 12988            | BIC:                | 1.627e+05 |       |         |         |
| Df Model:                  | 14               |                     |           |       |         |         |
| Covariance Type: nonrobust |                  |                     |           |       |         |         |
|                            | coef             | std err             | t         | P> t  | [0.025  | 0.975]  |
| season                     | 12.3801          | 2.305               | 5.371     | 0.000 | 7.862   | 16.899  |
| holiday                    | -21.4220         | 8.249               | -2.597    | 0.009 | -37.592 | -5.252  |
| workingday                 | 0.2884           | 3.991               | 0.072     | 0.942 | -7.535  | 8.112   |
| weather                    | -7.1833          | 1.938               | -3.706    | 0.000 | -10.982 | -3.384  |
| temp                       | 1.8075           | 1.184               | 1.526     | 0.127 | -0.514  | 4.129   |
| atemp                      | 4.5848           | 1.083               | 4.235     | 0.000 | 2.463   | 6.707   |
| humidity                   | -1.6066          | 0.069               | -23.305   | 0.000 | -1.742  | -1.471  |
| windspeed                  | 0.4438           | 0.148               | 3.001     | 0.003 | 0.154   | 0.734   |
| Year                       | -0.0033          | 0.004               | -0.742    | 0.458 | -0.012  | 0.005   |
| Quater                     | -32.9994         | 4.584               | -7.199    | 0.000 | -41.985 | -24.014 |
| Quater_ver2                | 20.8358          | 0.662               | 31.487    | 0.000 | 19.539  | 22.133  |
| Month                      | 5.5936           | 1.376               | 4.066     | 0.000 | 2.897   | 8.290   |
| Day                        | -0.2107          | 0.126               | -1.667    | 0.096 | -0.458  | 0.037   |
| Hour                       | 6.8639           | 0.169               | 40.695    | 0.000 | 6.533   | 7.195   |
| DavofWeek                  | 0.6025           | 0.915               | 0.658     | 0.510 | -1.191  | 2.396   |

|                |          |                   |          |
|----------------|----------|-------------------|----------|
| Omnibus:       | 2395.271 | Durbin-Watson:    | 0.552    |
| Prob(Omnibus): | 0.000    | Jarque-Bera (JB): | 4484.367 |
| Skew:          | 1.146    | Prob(JB):         | 0.00     |
| Kurtosis:      | 4.739    | Cond. No.         | 1.57e+04 |

#### Warnings:

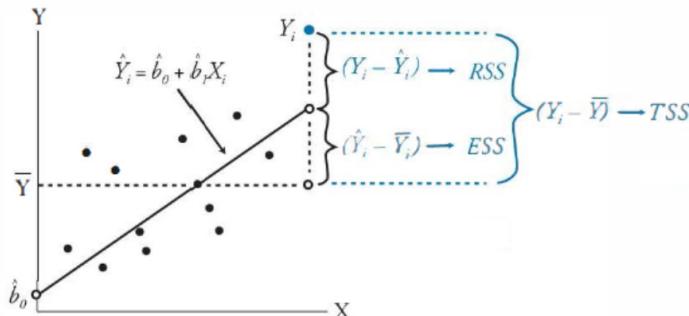
- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 1.57e+04. This might indicate that there are strong multicollinearity or other numerical problems.

#### 2-1) R-squared( $R^2$ ): 추정된 모형이 데이터에 잘 적합된 정도, $(-\infty, 1]$

" TSS = ESS + RSS "

- **TSS(Total Sum of Squares):** 실제 종속변수  $Y$ 의 움직임 범위
- **ESS(Explained Sum of Squares):** 예측된 종속변수  $\hat{Y}$ 의 움직임 범위
- **RSS(Residual Sum of Squares):** 잔차  $e$ 의 움직임 범위

- "예측값의 움직임 범위 << 실제 움직임 범위보다 클 수 없음"
- "예측 성능이 좋을수록 예측값의 움직임 범위는 실제 움직임 범위와 비슷해짐"

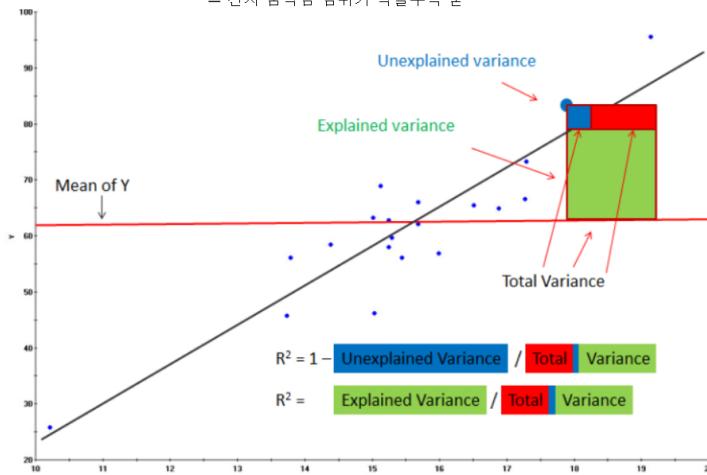


$$R^2 = \frac{ESS}{TSS} = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

= 종속변수 움직임 범위 대비 예측변수 움직임 범위도 비슷할수록 굳

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

= 잔차 움직임 범위가 작을수록 굳



#### 2-2) t-검정: 추정계수가 t분포의 움직임을 보이기 때문에, t분포 기반 독립변수와 종속변수 간의 영향력 정도 의 사결정

- 추정계수의 분포:

Main Equation

$$\hat{W} = (X^T X)^{-1} X^T Y$$

$$= (X^T X)^{-1} X^T (XW + e)$$

$$= W + (X^T X)^{-1} X^T e$$

Expectation

$$E(\hat{W}) = E(W + (X^T X)^{-1} X^T e)$$

$$= W + (X^T X)^{-1} X^T E(e)$$

$$= W$$

Variance

$$\text{Var}(\hat{W}_i) = (\text{Cov}(\hat{W}))_{ii} \quad (i = 0, \dots, K-1)$$

Covariance

$$\text{Cov}(\hat{W}) = F((\hat{W} - W)(\hat{W} - W)^T)$$

$$\begin{aligned}
& E((X^T X)^{-1} X^T e) = E((X^T X)^{-1} X^T e e^T X (X^T X)^{-1}) \\
& = E((X^T X)^{-1} X^T e e^T X (X^T X)^{-1}) \\
& = (X^T X)^{-1} X^T E(e e^T) X (X^T X)^{-1} \\
& = (X^T X)^{-1} X^T (\sigma^2 I) X (X^T X)^{-1} \\
& = \sigma^2 (X^T X)^{-1}
\end{aligned}$$

Standard Deviation  $\sqrt{\text{Var}(\hat{W}_i)} \approx se_{\hat{W}_i} = \sqrt{\sigma^2 ((X^T X)^{-1})_{ii}} \quad (i = 0, \dots, K-1)$

Asymptotic  $\frac{\hat{W}_i - W_i}{se_{\hat{W}_i}} \sim t_{N-K} \quad (i = 0, \dots, K-1)$

- 검정통계량(t-통계량) 의미:

$$t = \frac{\hat{W}_i - W_i}{se_{\hat{W}_i}} = \frac{\hat{W}_i - 0}{se_{\hat{W}_i}} = \frac{\hat{W}_i}{se_{\hat{W}_i}}$$

- $t$  값이 작으면, 독립변수와 종속변수의 상관성이 없다
- $t$  값이 크면, 독립변수와 종속변수의 상관성이 있다

- 의사결정:

#### (1) 가설확인:

| 종류                                             | 해석                         |
|------------------------------------------------|----------------------------|
| 대중주장<br>(귀무가설, Null Hypothesis, $H_0$ )        | 독립변수와 종속변수의 상관관계(선형관계)가 없다 |
| 나의주장<br>(대립가설, Alternative Hypothesis, $H_1$ ) | 독립변수와 종속변수의 상관관계(선형관계)가 있다 |

#### (2) 유의수준 설정 및 유의확률 확인

- 유의수준: 5% (0.05) 분석자가 알아서 결정
- 유의확률(p-value): 컴퓨터가 알아서 주정

#### (3) 의사결정

| 기준                            | 의사결정   | 해석                                                 |
|-------------------------------|--------|----------------------------------------------------|
| p-value $\geq$ 유의수준(ex. 0.05) | 대중주장 참 | 독립변수와 종속변수의 상관관계(선형관계)가 없다<br>분석한 변수는 모델링에 영향력이 없다 |
| p-value $<$ 유의수준(ex. 0.05)    | 나의주장 참 | 독립변수와 종속변수의 상관관계(선형관계)가 있다<br>분석한 변수는 모델링에 영향력이 있다 |

2-3) F검정: 전체 계수에 대한  $ESS/RSS$ 가 F분포의 움직임을 보이기 때문에, F분포 기반 독립변수와 종속변수 간의 알고리즘 신뢰성 의사결정

- 필요성:

- 변수의 갯수와 크기가 커지면 잔차제곱합(Residual Sum of Square)이 무조건 감소
- 분산 분석(Analysis of Variance(ANOVA))은 종속변수의 분산과 모든 독립변수의 분산간의 관계를 사용하여 F분포 기준 알고리즘 성능 평가

- 검정통계량(F-통계량): 분산분석표(ANOVA Table)를 통해 쉽게 계산되며,  $T$ 는 데이터의 갯수,  $K$ 는 변수의 갯수

$$\frac{ESS}{K-1} \div \frac{RSS}{T-K} \sim F(K-1, T-K)$$

| Source     | Degree of Freedom | Sum of Square     | Mean Square                    | F test-statistics                   | p-value   |
|------------|-------------------|-------------------|--------------------------------|-------------------------------------|-----------|
| Estimation | $K-1$             | $ESS$             | $\sigma_Y^2 = \frac{ESS}{K-1}$ | $F = \frac{\sigma_Y^2}{\sigma_e^2}$ | $p-value$ |
| Residual   | $T-K$             | $RSS$             | $\sigma_e^2 = \frac{RSS}{T-K}$ |                                     |           |
| Total      | $T-1$             | $TSS$             | $\sigma_T^2 = \frac{TSS}{T-1}$ |                                     |           |
| $R^2$      |                   | $\frac{ESS}{TSS}$ |                                |                                     |           |

- 의사결정:

#### (1) 가설확인:

| 종류                                             | 해석                                                          |
|------------------------------------------------|-------------------------------------------------------------|
| 대중주장<br>(귀무가설, Null Hypothesis, $H_0$ )        | 모형은 아무 효과가 없다<br>$W_0 = W_1 = \dots = W_{K-1} = 0$          |
| 나의주장<br>(대립가설, Alternative Hypothesis, $H_1$ ) | 모형은 효과가 있다<br>$W_0 \neq W_1 \neq \dots \neq W_{K-1} \neq 0$ |

#### (2) 유의수준 설정 및 유의확률 확인

- 유의수준: 5% (0.05) 분석가가 알아서 결정
- 유의확률(p-value): 컴퓨터가 알아서 추정

### (3) 의사결정

| 기준                            | 의사결정   | 해석                                        |
|-------------------------------|--------|-------------------------------------------|
| p-value $\geq$ 유의수준(ex. 0.05) | 대중주장 참 | 분석한 모델링은 효과가 없다<br>모델은 데이터 패턴을 잘 추정하지 못한다 |
| p-value < 유의수준(ex. 0.05)      | 나의주장 참 | 분석한 모델링은 효과가 있다<br>모델은 데이터 패턴을 잘 추정한다     |

2-4) 정보량기준(Information Criterion): 회귀분석 외에도 다양한 알고리즘에 활용, 값이 작을수록 좋은 모형결과 (Likelihood는 클수록 좋은 모형결과)

#### • AIC(Akaike Information Criterion)

: 모형과 데이터 확률분포의 Kullback-Leibler 수준을 가장 크게하기 위한 시도

$$AIC = -2\log(L) + 2K$$

(L: likelihood, K: 추정할 파라미터의 수(column수))

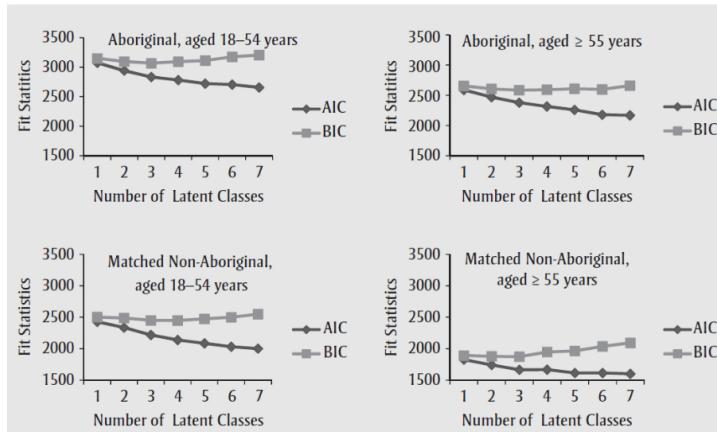
#### • BIC(Bayesian Information Criterion)

: 데이터가 exponential family라는 가정하에 데이터에서 모형의 likelihood를 측정하기 위한 값에서 유도

$$BIC = -2\log(L) + K\log(T)$$

(L: likelihood, K: 추정할 파라미터의 수(column수), T: 데이터의 수(row수))

#### • 사용 예시:



## 7 잔차진단 방향(Residual Diagnostics)

- 회귀분석 알고리즘 적합/추정 후 결과지표: 추정성능(빨간색) + 잔차진단(보라색)

| OLS Regression Results     |                  |                 |          |                     |           |        |
|----------------------------|------------------|-----------------|----------|---------------------|-----------|--------|
| Dep. Variable:             | count            | R-squared:      | 0.390    | Adj. R-squared:     | 0.390     |        |
| Model:                     | OLS              | F-statistic:    | 593.6    | Prob (F-statistic): | 0.00      |        |
| Method:                    | Least Squares    | Log-Likelihood: | -81290.  | AIC:                | 1.626e+05 |        |
| Date:                      | Sat, 16 Feb 2019 | Time:           | 02:13:49 | BIC:                | 1.627e+05 |        |
| No. Observations:          | 13003            | Df Residuals:   | 12988    | Df Model:           | 14        |        |
| Covariance Type: nonrobust |                  |                 |          |                     |           |        |
|                            | coef             | std err         | t        | P> t                | [0.025    | 0.975] |
| season                     | 12.3801          | 2.305           | 5.371    | 0.000               | 7.862     | 16.899 |
| holiday                    | -21.4220         | 8.249           | -2.597   | 0.009               | -37.592   | -5.252 |
| workingday                 | 0.2884           | 3.991           | 0.072    | 0.942               | -7.535    | 8.112  |
| weather                    | -7.1833          | 1.938           | -3.706   | 0.000               | -10.982   | -3.384 |
| temp                       | 1.8075           | 1.184           | 1.526    | 0.127               | -0.514    | 4.129  |
| atemp                      | 4.5848           | 1.083           | 4.235    | 0.000               | 2.463     | 6.707  |
| humidity                   | -1.6066          | 0.069           | -23.305  | 0.000               | -1.742    | -1.471 |
| windspeed                  | 0.4438           | 0.148           | 3.001    | 0.003               | 0.154     | 0.734  |

|             |          |       |        |       |         |         |
|-------------|----------|-------|--------|-------|---------|---------|
| Year        | -0.0033  | 0.004 | -0.742 | 0.458 | -0.012  | 0.005   |
| Quater      | -32.9994 | 4.584 | -7.199 | 0.000 | -41.985 | -24.014 |
| Quater_ver2 | 20.8358  | 0.662 | 31.487 | 0.000 | 19.539  | 22.133  |
| Month       | 5.5936   | 1.376 | 4.066  | 0.000 | 2.897   | 8.290   |
| Day         | -0.2107  | 0.126 | -1.667 | 0.096 | -0.458  | 0.037   |
| Hour        | 6.8639   | 0.169 | 40.695 | 0.000 | 6.533   | 7.195   |
| DayofWeek   | 0.6025   | 0.915 | 0.658  | 0.510 | -1.191  | 2.396   |

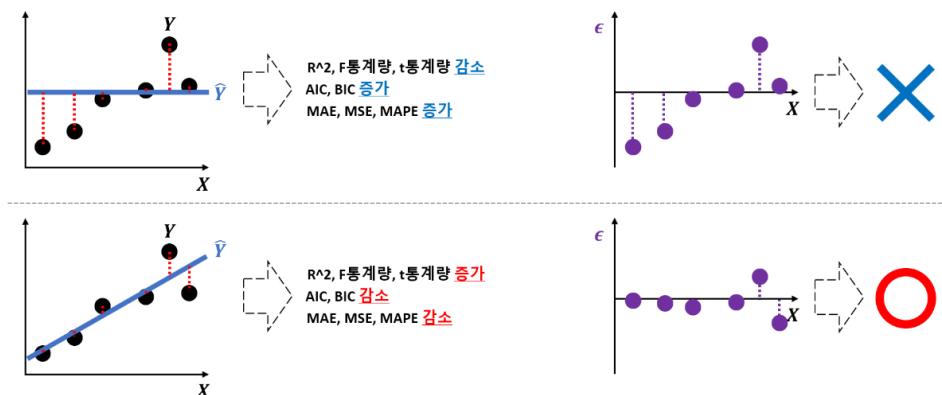
|                |          |                   |          |
|----------------|----------|-------------------|----------|
| Omnibus:       | 2395.271 | Durbin-Watson:    | 0.552    |
| Prob(Omnibus): | 0.000    | Jarque-Bera (JB): | 4484.367 |
| Skew:          | 1.146    | Prob(JB):         | 0.00     |
| Kurtosis:      | 4.739    | Cond. No.         | 1.57e+04 |

Warnings:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 1.57e+04. This might indicate that there are strong multicollinearity or other numerical problems.

0) 검증지표 vs 잔차진단: Y수치 비교가 가능한 예측문제에서 Y라벨이 있는 지도학습에 적용 가능

✓ 검증지표(Evaluation Metrics) : 실제  $Y$ 와 예측  $\hat{Y}$ 이 얼마나 유사한지 측정      ↪ 상호작용      ✓ 잔차진단(Residual Diagnostics) :  $Y - \hat{Y} = \epsilon$  에서 잔차에 남은 패턴이 없는지 측정



1) 잔차진단의 2가지 목적:

"예측 성능도 중요하지만(추정성능), 추정/분석 이후 데이터의 패턴이 모델링에 잘 반영되었는지 (잔차진단) 평가하는 것도 중요"

(1) 추가할만한 데이터 전처리 또는 다른 모델링의 대안 파악

- 잔차에 남아있는 패턴을 전처리 단계에서 추가 반영 가능
- 잔차의 남은 패턴으로 다른 분석 알고리즘 고려 가능

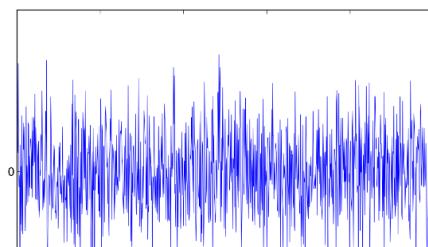
(2) 분석 시작은 여러분들이 시작했지만, 분석 종료는 잔차 진단이 알려줌

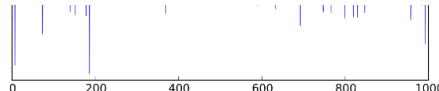
- 잔차의 남은 패턴이 없다는 것은 모델링이 데이터의 패턴을 최대한 반영 의미
- 모델링으로 더이상 할 수 있는 것들이 없으니 분석을 마무리 해도 됨을 의미

2) 잔차진단의 목표: 잔차가 백색잡음 과 얼마나 유사한지 측정

- 모델링에 데이터의 패턴이 잘 반영 되었다면, 추정 후의 잔차에는 아무 패턴도 없어야 함
- 잔차에 아무런 패턴도 남아있지 않은 경우 백색잡음의 형태로 분포
- 잔차 분석/진단을 통해 잔차가 백색잡음(White Noise)라면 역으로 모델링에서 데이터의 패턴을 잘 반영하여 성능이 좋음을 의미

3) 백색잡음: 잔차가 백색잡음이 아니라면 모델링으로 개선의 여지가 있음을 의미





(1) 잔차들은 정규분포이고, (unbiased) 평균 0이고 일정한 분산을 가져야 함:

$$\{\epsilon_t : t = \dots, -1, 0, 1, \dots\} \sim N(0, \sigma_{\epsilon_t}^2)$$

where  $\epsilon_t \sim \text{i.i.d}(independent and identically distributed)$

$$\epsilon_t = Y_t - \hat{Y}_t$$

$$E(\epsilon_t) = 0$$

$$Var(\epsilon_t) = \sigma_{\epsilon_t}^2$$

$$Cov(\epsilon_s, \epsilon_k) = 0 \text{ for different times!}(s \neq k)$$

(2) 잔차들이 시간의 흐름에 따라 상관성이 없어야 함: 자기상관함수(Autocorrelation Function, ACF)=0 확인

- 공분산(Covariance):

$$Cov(Y_s, Y_k) = E[(Y_s - E(Y_s))(Y_k - E(Y_k))] = \gamma_{s,k}$$

- 자기상관함수(Autocorrelation Function):

$$Corr(Y_s, Y_k) = \frac{Cov(Y_s, Y_k)}{\sqrt{Var(Y_s)Var(Y_k)}} = \frac{\gamma_{s,k}}{\sqrt{\gamma_s \gamma_k}}$$

• 편자기상관함수(Partial Autocorrelation Function):  $s$ 와  $k$ 사이의 상관성을 제거한 자기상관함수

$$Corr[(Y_s - \hat{Y}_s, Y_{s-t} - \hat{Y}_{s-t})] \text{ for } 1 < t < k$$

## 7.1 정규분포 테스트(Normality Test)

- Shapiro-Wilk test:

(1) 가설확인:

| 종류                                             | 해석                |
|------------------------------------------------|-------------------|
| 대중주장<br>(귀무가설, Null Hypothesis, $H_0$ )        | 데이터는 정규분포 형태이다    |
| 나의주장<br>(대립가설, Alternative Hypothesis, $H_1$ ) | 데이터는 정규분포 형태가 아니다 |

(2) 유의수준 설정 및 유의확률 확인

- 유의수준: 5% (0.05) 분석자가 알아서 결정
- 유의확률(p-value): 컴퓨터가 알아서 추정

(3) 의사결정

| 기준                            | 의사결정   | 해석                          |
|-------------------------------|--------|-----------------------------|
| p-value $\geq$ 유의수준(ex. 0.05) | 대중주장 참 | 내가 수집/분석한 데이터는 정규분포 형태이다    |
| p-value $<$ 유의수준(ex. 0.05)    | 나의주장 참 | 내가 수집/분석한 데이터는 정규분포 형태가 아니다 |

- Kolmogorov-Smirnov test:

- 가설확인: Shapiro-Wilk와 동일

- Lilliefors test:

- 가설확인: Shapiro-Wilk와 동일

- Anderson-Darling test:

- 가설확인: Shapiro-Wilk와 동일

- Jarque-Bera test:

- 가설확인: Shapiro-Wilk와 동일

- Pearson's chi-squared test:

- 가설확인: Shapiro-Wilk와 동일

- D'Agostino's K-squared test:

- 가설확인: Shapiro-Wilk와 동일

- 예시:

| Nr | Stock | Ret.    | Var.   | Skew.   | Shapiro-Wilk Statistic | p-value | Sig |
|----|-------|---------|--------|---------|------------------------|---------|-----|
| 1  | BRL   | 0.0838  | 0.7274 | 0.1648  | 0.9870                 | 0.4506  |     |
| 2  | AIR   | 0.1027  | 0.3334 | 0.0285  | 0.9952                 | 0.9818  |     |
| 3  | CGE   | 0.0773  | 0.5935 | 0.0043  | 0.9849                 | 0.3277  |     |
| 4  | MIDI  | 0.0585  | 0.8904 | 0.6243  | 0.9509                 | 0.0011  | *** |
| 5  | BYG   | 0.0120  | 0.7989 | 0.3171  | 0.9716                 | 0.0323  | **  |
| 6  | CAN   | 0.0950  | 0.5194 | -0.2843 | 0.9596                 | 0.0043  | *** |
| 7  | CGS   | -0.0010 | 0.9000 | 0.2884  | 0.9868                 | 0.4380  |     |
| 8  | CRFR  | 0.2060  | 0.4531 | -0.0190 | 0.9949                 | 0.9734  |     |
| 9  | CSO   | 0.0431  | 0.7997 | -0.1314 | 0.9878                 | 0.5057  |     |
| 10 | CCF   | 0.0832  | 0.5431 | -0.1101 | 0.9715                 | 0.0314  | **  |
| 11 | BSN   | 0.0725  | 0.2887 | -0.0115 | 0.9948                 | 0.9708  |     |
| 12 | ORAF  | 0.1771  | 0.4724 | 0.1226  | 0.9877                 | 0.5026  |     |
| 13 | LFG   | 0.0388  | 0.6573 | -0.2074 | 0.9841                 | 0.2874  |     |
| 14 | TVMU  | 0.1510  | 0.5270 | 0.0000  | 0.9452                 | 0.0005  | *** |

|    |      | 0.1517  | 0.5270 | -0.0407 | 0.2454 | 0.0003     |
|----|------|---------|--------|---------|--------|------------|
| 15 | MCL  | 0.0407  | 1.0336 | -0.1582 | 0.9879 | 0.5154     |
| 16 | PGT  | 0.0556  | 0.6676 | -0.0318 | 0.9929 | 0.8888     |
| 17 | PRNT | 0.1214  | 0.8311 | 0.0796  | 0.9892 | 0.6161     |
| 18 | GOB  | 0.0487  | 0.5312 | -0.1419 | 0.9833 | 0.2497     |
| 19 | SQAF | 0.1159  | 0.5402 | -0.0050 | 0.9845 | 0.3036     |
| 20 | QTAF | 0.1320  | 1.1093 | 0.0371  | 0.9862 | 0.3990     |
| 21 | SGE  | 0.0693  | 0.5728 | 0.0561  | 0.9923 | 0.8510     |
| 22 | SDX  | 0.1729  | 0.6073 | -0.0621 | 0.9715 | 0.0314 **  |
| 23 | LE   | 0.0651  | 0.7052 | -0.0953 | 0.9812 | 0.1755     |
| 24 | CSF  | -0.0087 | 0.8090 | 0.1477  | 0.9909 | 0.7505     |
| 25 | CFP  | 0.1374  | 0.4624 | 0.0670  | 0.9935 | 0.9205     |
| 26 | VAL  | 0.1219  | 0.7774 | -0.6872 | 0.9470 | 0.0006 *** |
| 27 | EXAF | 0.0806  | 0.5344 | -0.0987 | 0.9866 | 0.4282     |

$H_0$ : Distribution of returns for stock X is normally distributed

\* Rejection at 10% level, \*\*, 5%, \*\*\* 1%

## 7.2 등분산성 테스트(Homoscedasticity Test)

- [Goldfeld–Quandt test:](#)

(1) 가설확인:

| 종류                                             | 해석                                                    |
|------------------------------------------------|-------------------------------------------------------|
| 대중주장<br>(귀무가설, Null Hypothesis, $H_0$ )        | 데이터는 Homoscedasticity 상태다<br>(등분산이다)                  |
| 나의주장<br>(대립가설, Alternative Hypothesis, $H_1$ ) | 데이터는 Heteroscedasticity 상태다<br>(등분산이 아니다 / 일산하는 분산이다) |

(2) 유의수준 설정 및 유의확률 확인

- 유의수준: 5% (0.05) 분석가가 알아서 결정
- 유의확률(p-value): 컴퓨터가 알아서 주정

(3) 의사결정

| 기준                            | 의사결정   | 해석                      |
|-------------------------------|--------|-------------------------|
| p-value $\geq$ 유의수준(ex. 0.05) | 대중주장 참 | 내가 수집/분석한 데이터는 등분산 이다   |
| p-value < 유의수준(ex. 0.05)      | 나의주장 참 | 내가 수집/분석한 데이터는 등분산이 아니다 |

- [Breusch–Pagan test:](#)

- 가설확인: Goldfeld–Quandt와 동일

- [Bartlett's test:](#)

- 가설확인: Goldfeld–Quandt와 동일

- 예시:

| Predictor                    | Mean Squared Residual | Breusch-Pagan Statistic |
|------------------------------|-----------------------|-------------------------|
| Household type               |                       | 600.67**                |
| Dual heads                   | 1.2373                |                         |
| Male headed                  | 0.8644                |                         |
| Female headed                | 0.5725                |                         |
| Labor force status           |                       | 25.51**                 |
| Not in the labor force       | 0.7968                |                         |
| Retired                      | 1.0859                |                         |
| In the labor force           | 0.9896                |                         |
| Education                    |                       | 2308.37**               |
| Less than high school degree | 0.3593                |                         |
| High school degree           | 0.6140                |                         |
| Some college                 | 0.9362                |                         |
| Bachelor's degree            | 1.4422                |                         |
| Postgraduate degree          | 2.1336                |                         |

\*\* $p < .01$ .

Table 6: Heteroscedasticity diagnostics for restaurants food sales data

| Test            | Without outliers   |         | With outliers      |         |
|-----------------|--------------------|---------|--------------------|---------|
|                 | Value of statistic | p-value | Value of statistic | p-value |
| Goldfeld-Quandt | 4.03671            | 0.019   | 1.074              | 0.4563  |
| Breusch-Pagan   | 3.1787             | 0.0746  | 0.3799             | 0.5376  |
| White           | 4.3575             | 0.0368  | 0.0963             | 0.7562  |
| MGQ             | 4.9917             | 0.0090  | 10.4566            | 0.0005  |

## 7.3 자기상관 테스트(Autocorrelation Test)

- [Ljung–Box test:](#)

(1) 가설확인:

| 종류                                             | 예석                                    |
|------------------------------------------------|---------------------------------------|
| 대중주장<br>(귀무가설, Null Hypothesis, $H_0$ )        | 데이터는 Autocorrelation은 0이다( 존재하지 않는다 ) |
| 나의주장<br>(대립가설, Alternative Hypothesis, $H_1$ ) | 데이터는 Autocorrelation은 0이 아니다( 존재한다 )  |

## (2) 유의수준 설정 및 유의확률 확인

- 유의수준: 5% (0.05) 분석가가 알아서 결정
- 유의확률(p-value): 컴퓨터가 알아서 추정

## (3) 의사결정

| 기준                            | 의사결정   | 해석                                       |
|-------------------------------|--------|------------------------------------------|
| p-value $\geq$ 유의수준(ex. 0.05) | 대중주장 참 | 내가 수집/분석한 데이터는 Autocorrelation은 존재하지 않는다 |
| p-value < 유의수준(ex. 0.05)      | 나의주장 참 | 내가 수집/분석한 데이터는 Autocorrelation은 존재한다     |

### Portmanteau test:

- 가설확인: Ljung-Box와 동일
- Breusch-Godfrey test:
  - 가설확인: Ljung-Box와 동일
- Durbin-Watson statistic:

### (1) 가설확인:

| 종류                                             | 해석                                    |
|------------------------------------------------|---------------------------------------|
| 대중주장<br>(귀무가설, Null Hypothesis, $H_0$ )        | 데이터는 Autocorrelation은 0이다( 존재하지 않는다 ) |
| 나의주장<br>(대립가설, Alternative Hypothesis, $H_1$ ) | 데이터는 Autocorrelation은 0이 아니다( 존재한다 )  |

## (2) 유의수준 설정 및 유의확률 확인

- 유의수준: 5% (0.05) 분석가가 알아서 결정
- 유의확률(p-value): 컴퓨터가 알아서 추정

## (3) 의사결정

| 기준        | 의사결정   | 해석                                                                                                                       |
|-----------|--------|--------------------------------------------------------------------------------------------------------------------------|
| 2 근방      | 대중주장 참 | 내가 수집/분석한 데이터는 Autocorrelation은 존재하지 않는다                                                                                 |
| 0 또는 4 근방 | 나의주장 참 | 내가 수집/분석한 데이터는 Autocorrelation은 존재한다<br>- 0: 양(Positive)의 Autocorrelation 존재한다<br>- 4: 음(Negative)의 Autocorrelation 존재한다 |

### • 예시:

| Model                         | Residual analysis                             | AIC    |
|-------------------------------|-----------------------------------------------|--------|
| ARIMA(1, 1, 0): one AR term   | Ljung-Box test:<br>$\chi^2 = 0.005, p = 0.94$ | 419.80 |
| ARIMA(0, 1, 1): one MA term   | Ljung-Box test:<br>$\chi^2 = 0.01, p = 0.92$  | 423.84 |
| ARIMA(1, 1, 1): a mixed model | Ljung-Box test:<br>$\chi^2 = 0.02, p = 0.89$  | 425.84 |
| ARIMA(2, 1, 0): two AR terms  | Ljung-Box test:<br>$\chi^2 = 0.61, p = 0.43$  | 448.79 |
| ARIMA(0, 1, 2): two MA terms  | Ljung-Box test:<br>$\chi^2 = 0.02, p = 0.89$  | 425.84 |

ACF plots for all models showed that <5% of autocorrelations reached statistical significance.

| Dependent variable EC |             |            |             |          |
|-----------------------|-------------|------------|-------------|----------|
| Variable              | Coefficient | Std. Error | t-Statistic | Prob.    |
| Intercept             | 0.986898*   | 0.151686   | 6.506195    | 0        |
| FA                    | 0.47757*    | 0.027709   | 17.23522    | 0        |
| R-squared             |             |            |             | 0.891909 |
| Adjusted R-squared    |             |            |             | 0.970055 |
| F-statistic           | 297.0529    | Probabil-  |             | 0        |
| Durbin-Watson stat    |             | ity        |             | 0.844436 |

(\* Significance at 1% level.