파이썬 데이터 분석 3종 세트
statsmodels, scikit-learn, theano
...

김도형
@drjoelkim
https://datascienceschool.net

@drjoelkim
Trade Informatix
https://datascienceschool.net

- 증권 분석 및 최적 집행 시스템 개발
- 금융 공학 / 데이터 분석 / 머신 러닝
- 컨설팅 및 교육

People Asks ...

[이러이러한] 데이터가 있습니다.

머신러닝으로
[저러저러한] 것을 알(할) 수 있나요?

대부분의 문제는 ...

- 다른 사람들이 이미 풀었거나
  하지만 당신이 원하는 답은 아닐 수 있습니다.

- 다른 사람들이 포기했습니다.
  또는 계속 연구하고 있습니다.

R 과 Python은 분석 도구의 바다 sea of tools

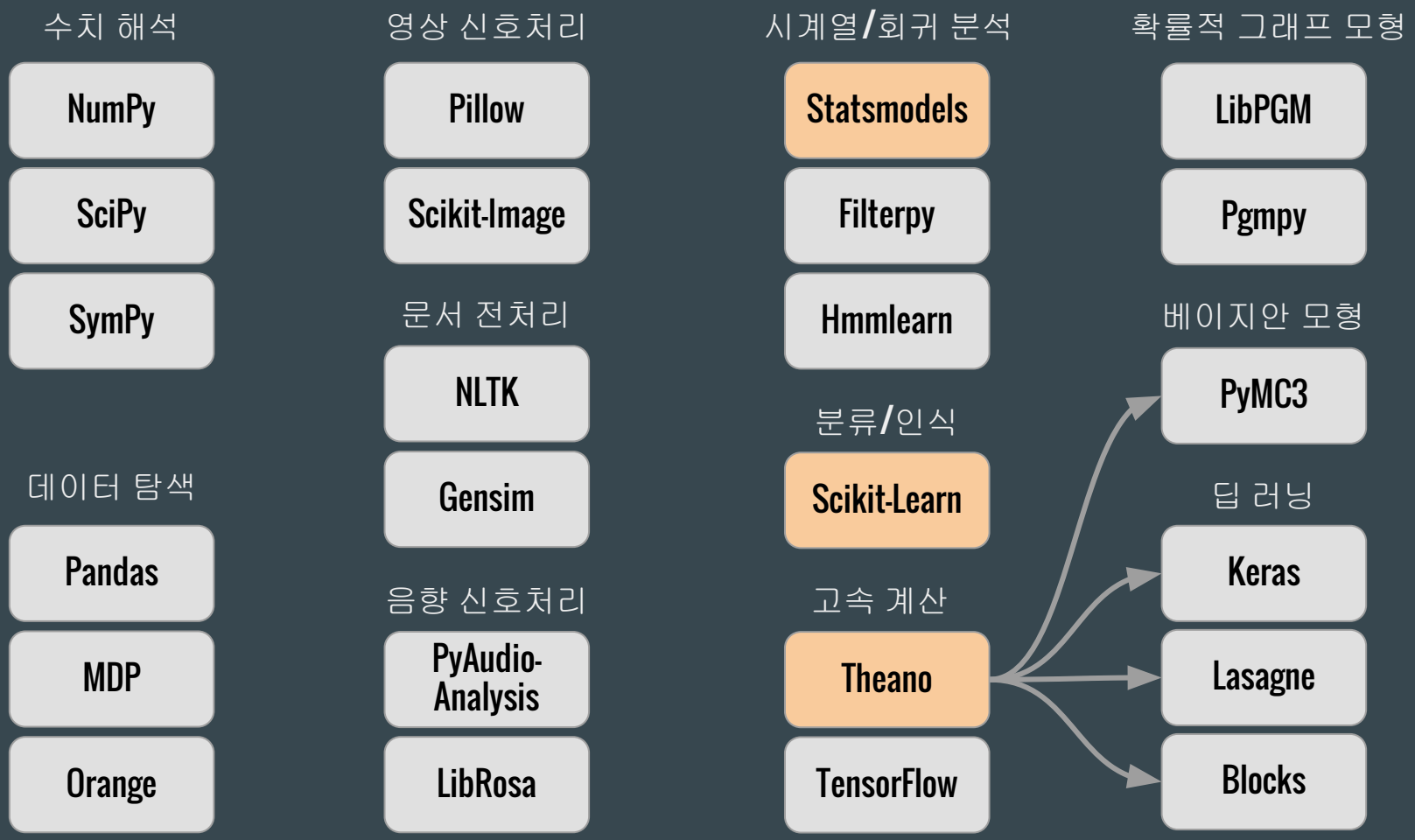
- 하지만 너무 넓습니다.
  지도guidemap가 필요합니다.

- 내가 어디로 가고 싶은지를 알아야 합니다.
  문제 유형을 파악하세요.

- 머신 러닝은 데이터 분석의 일부일 뿐입니다.
  가장 적절한 도구를 선택하세요.

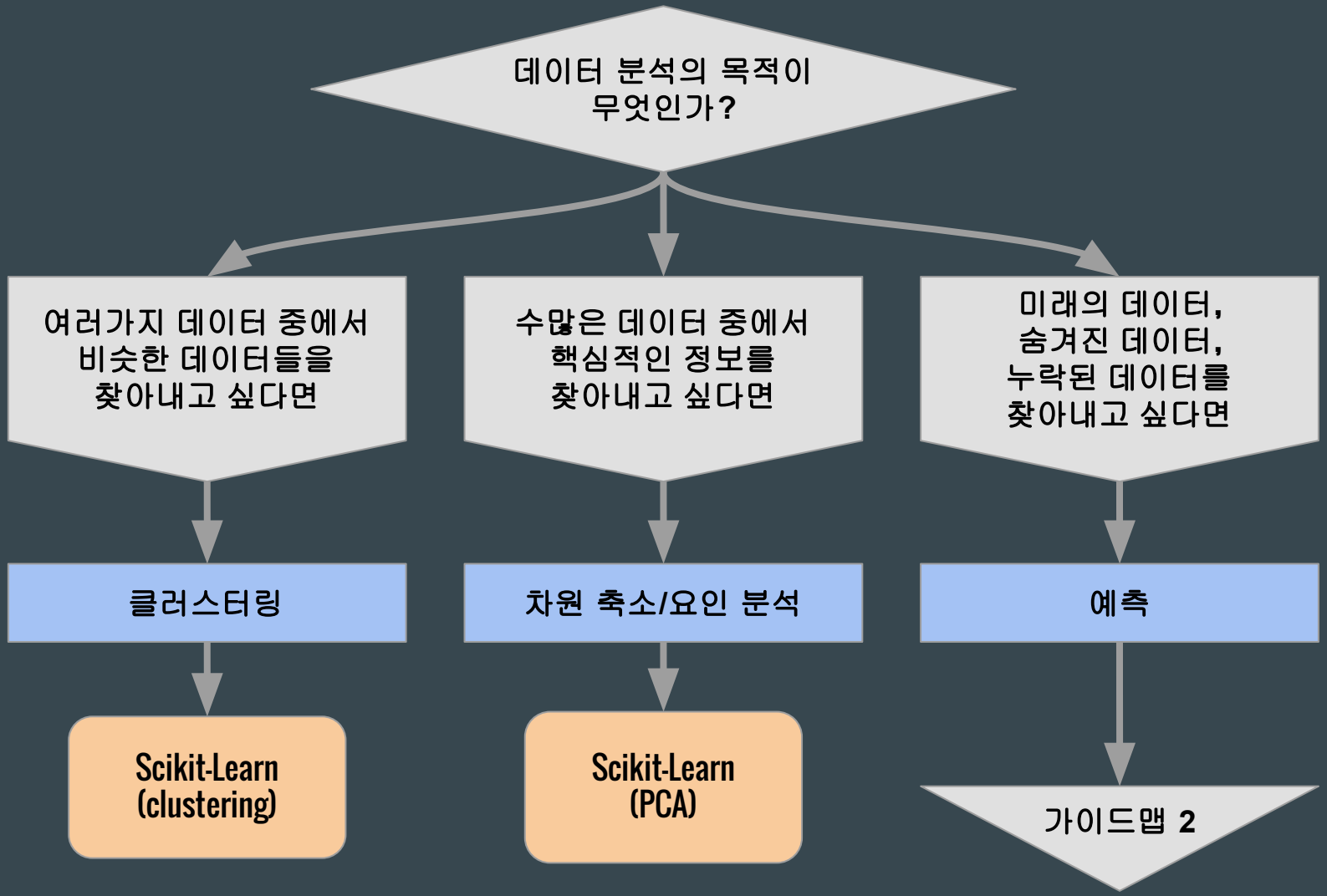
발표 내용의 간략한 소개

- 데이터 분석 문제의 5가지 기본적인 유형을 소개합니다.
  - 클러스터링
  - 차원축소
  - 시계열 예측
  - 회귀 분석
  - 분류
- 문제 유형에 따라 어떤 파이썬 패키지를 사용할 수 있는지를 제시합니다.
  - 패키지 선택 가이드맵
- 널리 사용되는 3가지 데이터 분석용 패키지를 소개합니다.
  - Scikit-learn
  - Statsmodels
  - Theano/Tensorflow
- 알고리즘 또는 패키지 코딩 방법을 자세히 소개하지는 않습니다.


수치 해석
NumPy
SciPy
SymPy
데이터 탐색
Pandas
MDP
Orange
영상 신호처리
Pillow
Scikit-Image
문서 전처리
NLTK
Gensim
음향 신호처리
PyAudio-Analysis
LibRosa
시계열/회귀 분석
Statsmodels
Filterpy
Hmmlearn
분류/인식
Scikit-Learn
고속 계산
Theano
TensorFlow
확률적 그래프 모형
LibPGM
Pgmpy
베이지안 모형
PyMC3
딥 러닝
Keras
Lasagne
Blocks

데이터 분석 패키지 맵

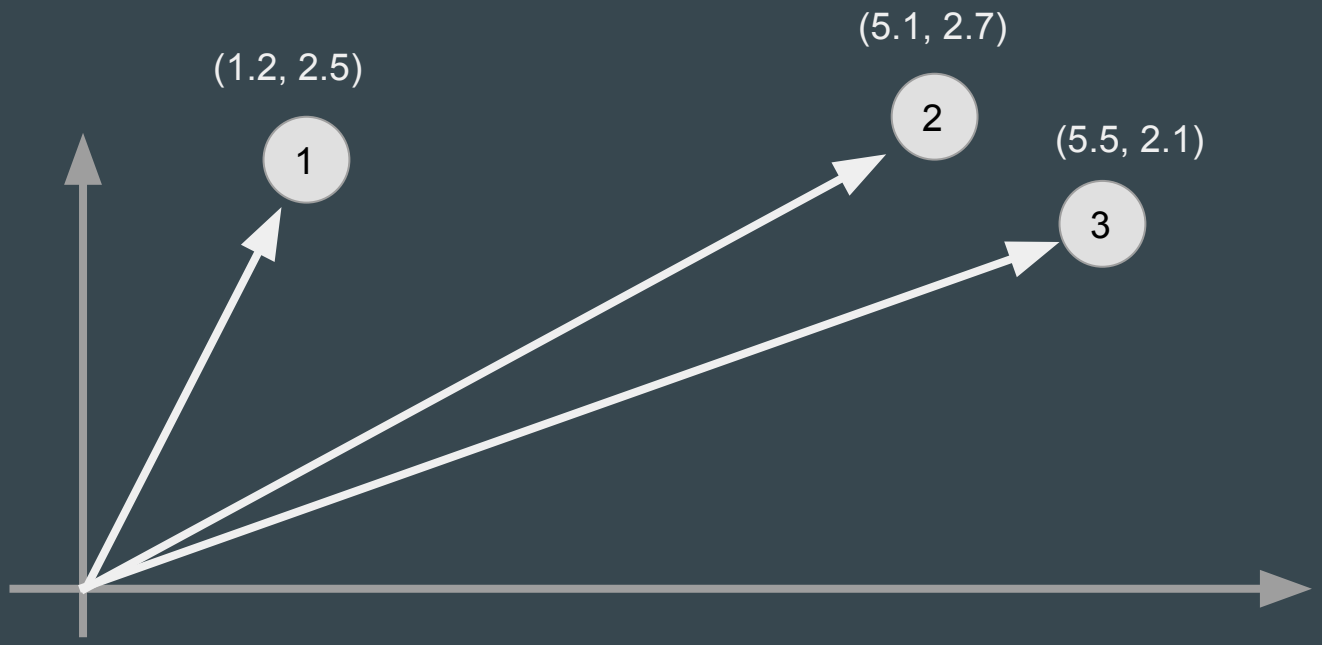
데이터 분석에는 **목적**이 있다


데이터 분석의 목적이
무엇인가?
여러가지 데이터 중에서
비슷한 데이터들을
찾아내고 싶다면
수많은 데이터 중에서
핵심적인 정보를
찾아내고 싶다면
미래의 데이터,
숨겨진 데이터,
누락된 데이터를
찾아내고 싶다면
클러스터링
차원 축소/요인 분석
예측
Scikit-Learn
(clustering)
Scikit-Learn
(PCA)
가이드맵 2

데이터 분석 가이드맵 1

클러스터링 Clustering

- 데이터는 벡터 공간상에서 하나의 점 또는 벡터로 표시할 수 있습니다.
- 데이터 혹은 데이터 묶음(cluster) 사이의 거리를 계산할 수 있습니다.
- 거리가 가까운 데이터는 유사한 데이터라고 볼 수 있습니다.



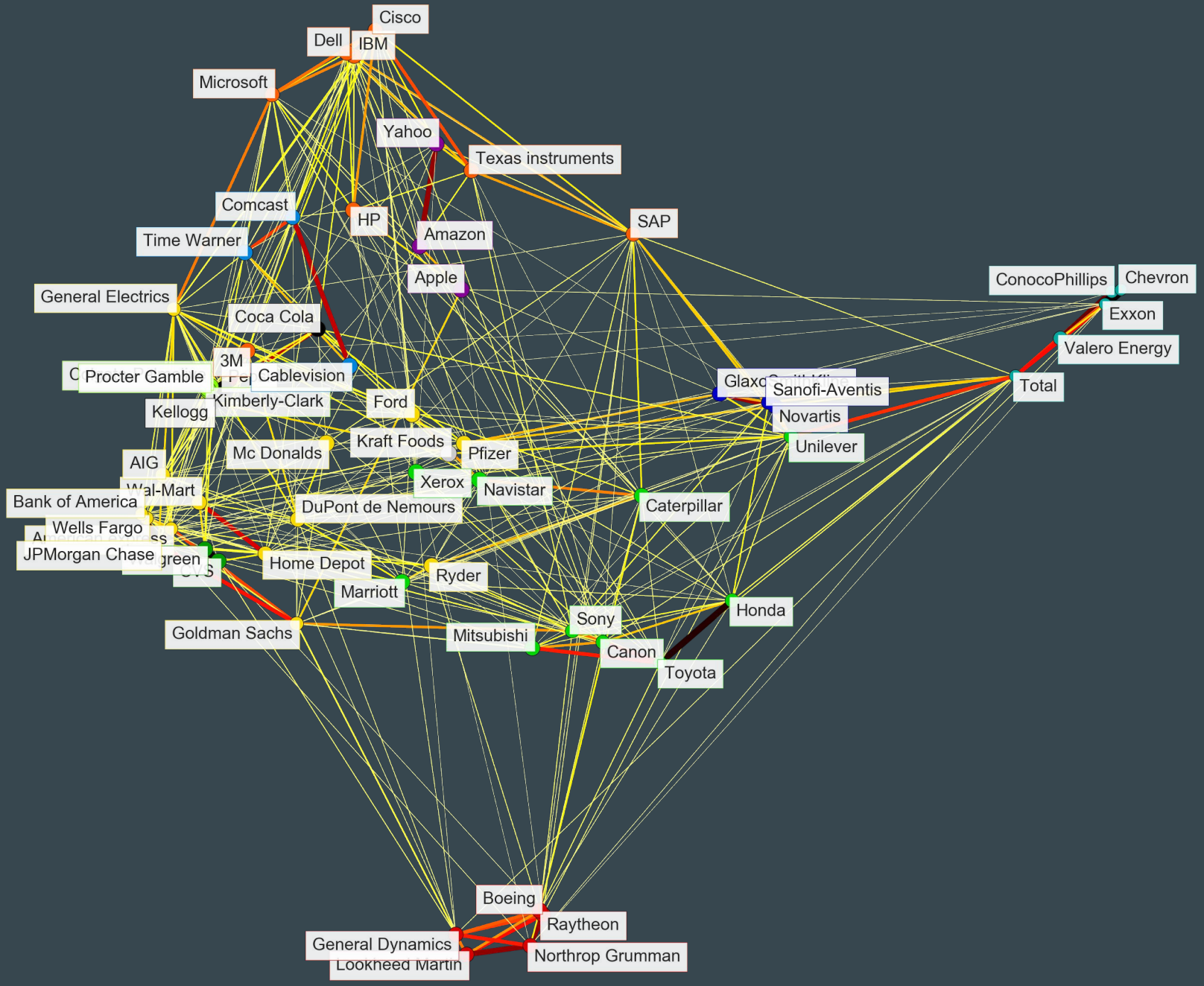
클러스터링의 예:
일중 가격의 움직임이 유사한 주식 찾기

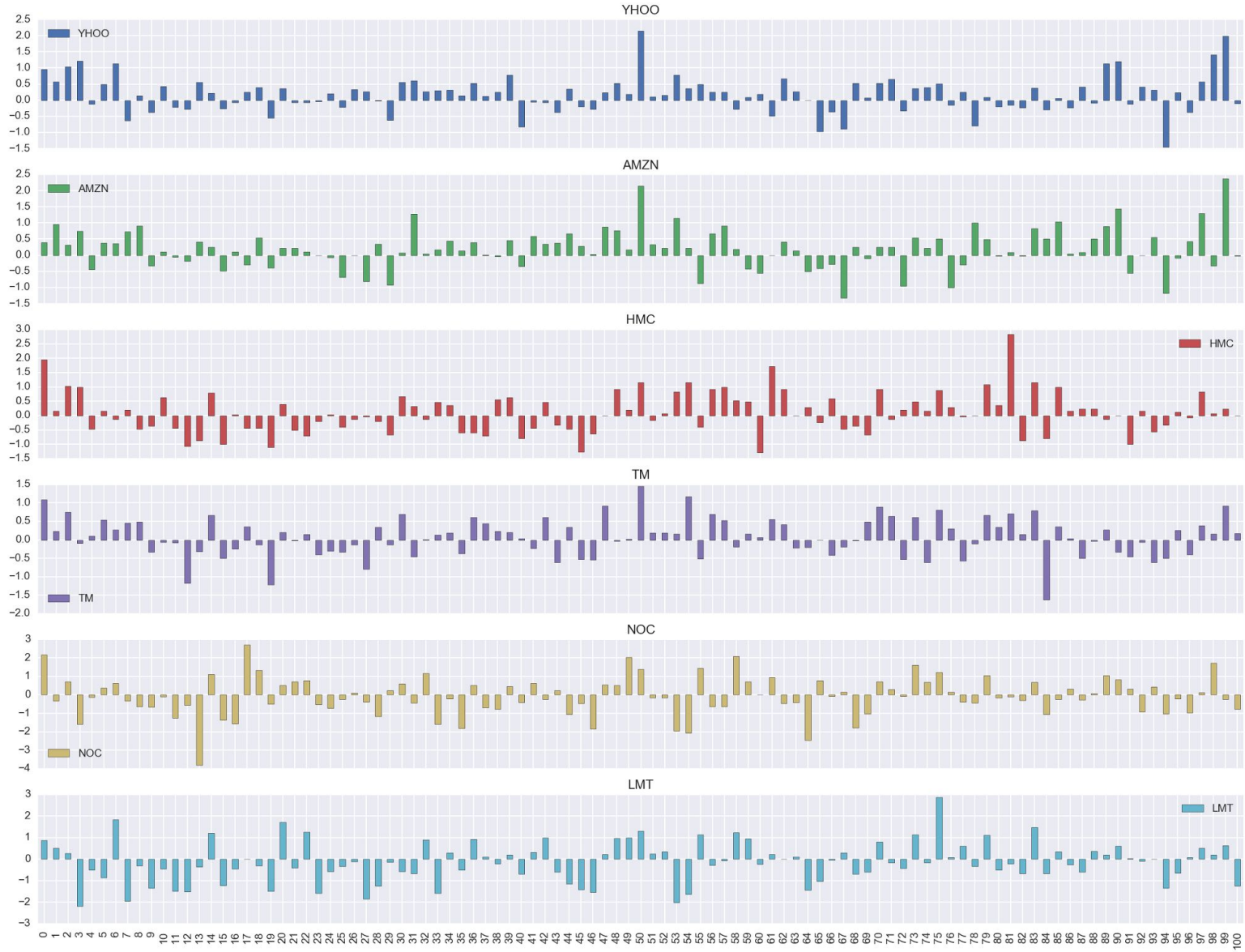
- http://scikit-learn.org/stable/auto_examples/applications/plot_stock_market.html
- 미국 대형주 60 종목 중 2003~2008까지 시가와 종가의 차이 움직임이 닮아 있는 종목들을 찾습니다.

코드

```
# 전처리 - 데이터 준비
X # standardized open - close


import sklearn


edge_model = sklearn.covariance.GraphLassoCV().fit(X)
_, labels = sklearn.cluster.affinity_propagation(edge_model.covariance_)

# 후처리 - 시각화
```

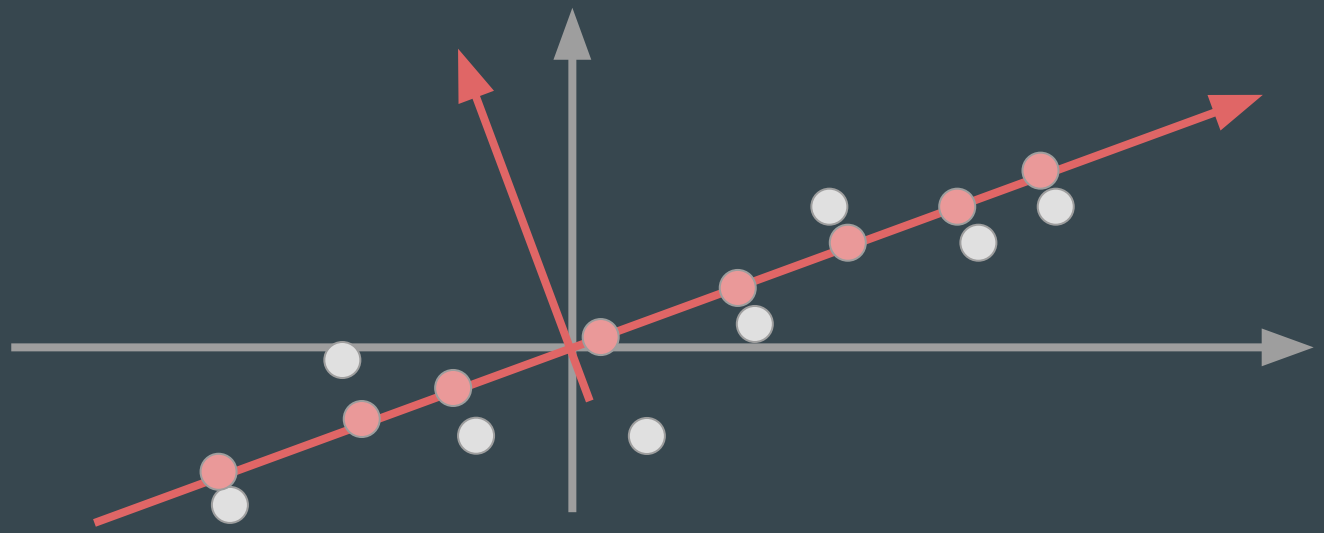
결과


Cisco
Dell
IBM
Microsoft
Yahoo
Texas instruments
Comcast
HP
Amazon
SAP
Time Warner
Apple
ConocoPhillips
Chevron
General Electrics
Coca Cola
Exxon
Valero Energy
3M
Procter Gamble
Cablevision
Sanofi-Aventis
Total
Kimberly-Clark
Ford
Kellogg
Novartis
Kraft Foods
Unilever
Mc Donalds
Pfizer
AIG
Xerox
Navistar
Bank of America
DuPont de Nemours
Caterpillar
Wells Fargo
JPMorgan Chase
Home Depot
Ryder
Marriott
Honda
Sony
Goldman Sachs
Mitsubishi
Canon
Toyota
Boeing
Raytheon
General Dynamics
Northrop Grumman


YHOO
AMZN
HMC
TM
NOC
LMT

주성분 분석 PCA(Principal Component Analysis)

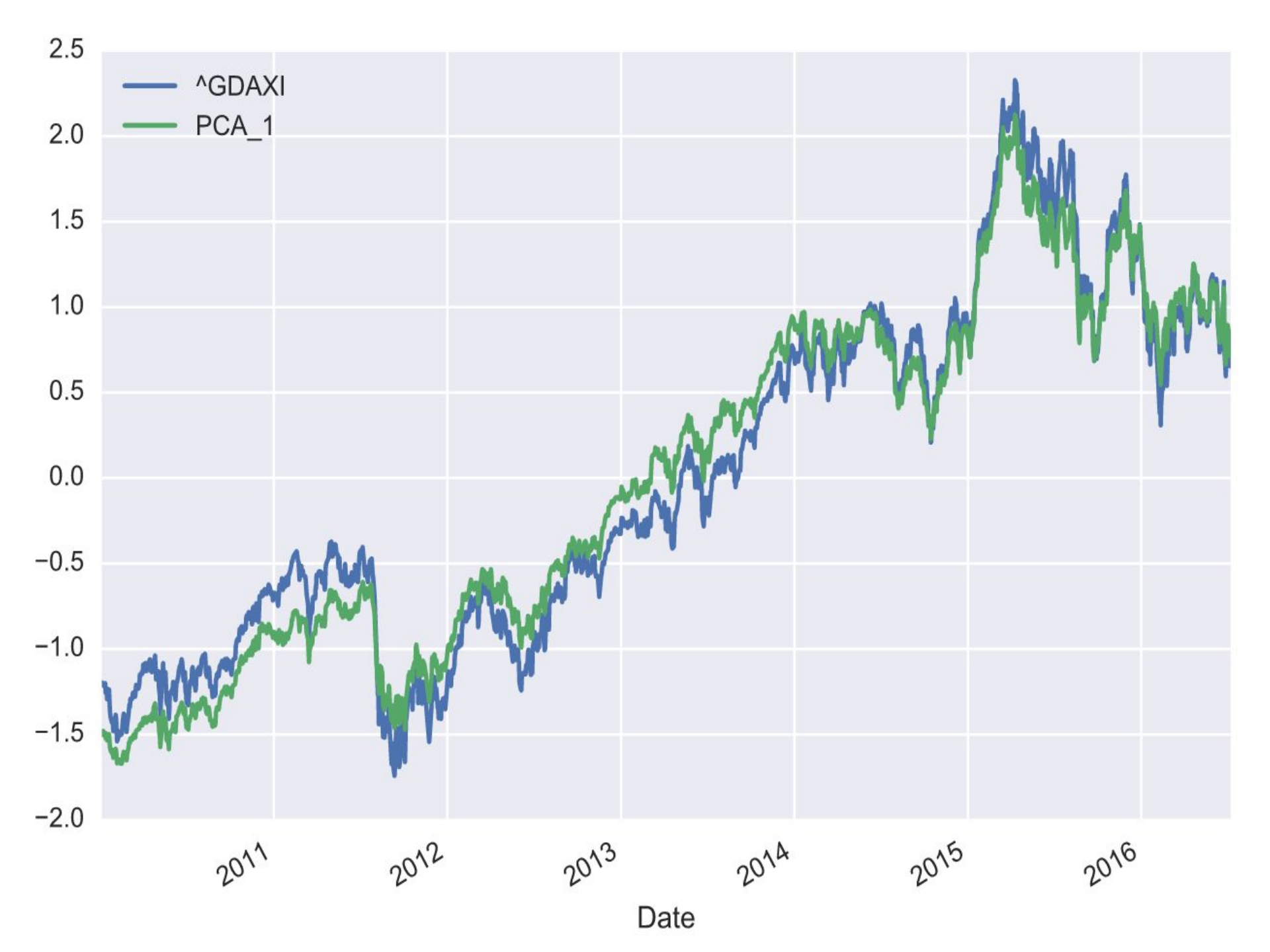
- 다차원 데이터는 좌표를 변환 할 수 있습니다.
- 데이터가 특정한 움직임만 보이는 경우에는 주된 움직임을 포착하는 좌표 변환을 할 수 있습니다.
- 좌표 변환 후에는 작은 움직임을 나타내는 좌표는 생략할 수도 있습니다.

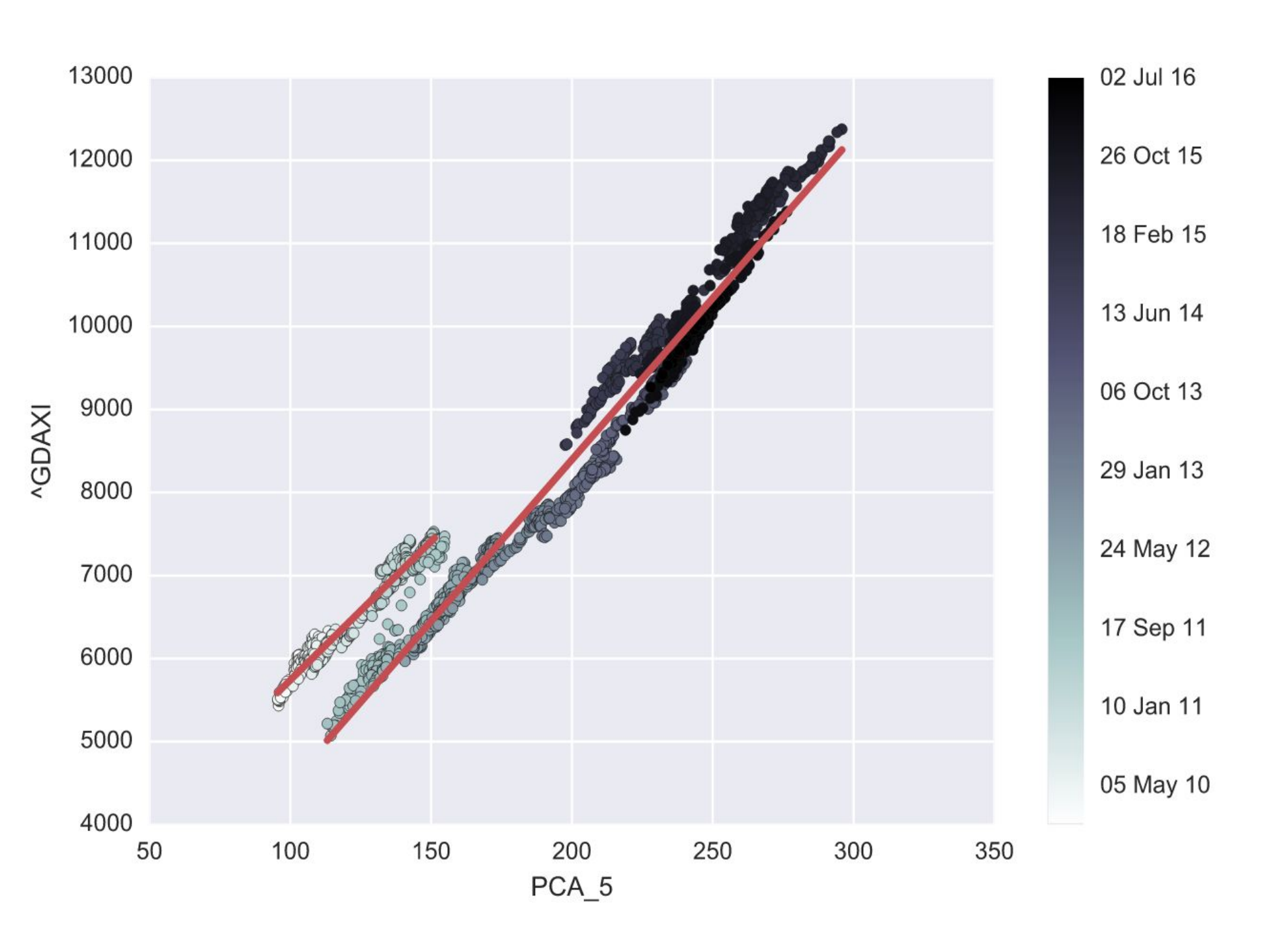


주성분 분석의 예:
주식 시장의 지수 성분 찾기

- 파이썬을 활용한 금융 분석(한빛 미디어), 11장, pp.403~410
- 독일 대형주 30 종목의 움직임을 분석하여 가장 주된 주가 움직임을 찾는다.

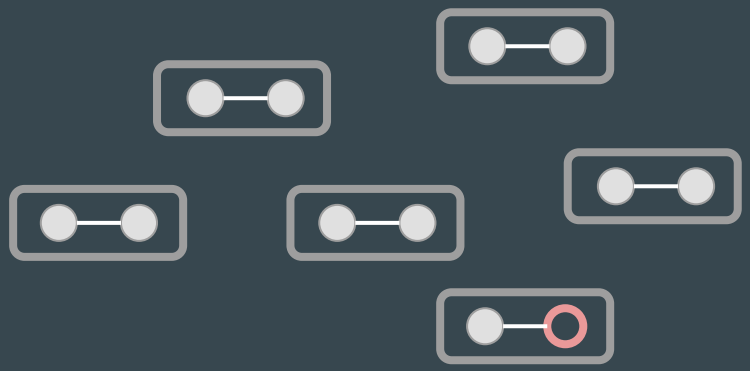
```
# 전처리 : 데이터 준비
data # 30종목의 주가 데이터 (pandas DataFrame)

import sklearn
pca = sklearn.decomposition.KernelPCA(n_components=1).fit(data)
PCA_1 = pca.transform(data)

# 후처리 : 시각화
```

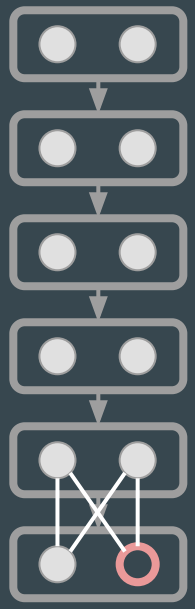

^GDAXI
PCA_1
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
2011
2012
2013
2014
2015
2016
Date


13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
^GDAXI
50
100
150
200
250
300
350
PCA_5
02 Jul 16
26 Oct 15
18 Feb 15
13 Jun 14
06 Oct 13
29 Jan 13
24 May 12
17 Sep 11
10 Jan 11
05 May 10

데이터 간에는 **순서**가 있을 수 있다.

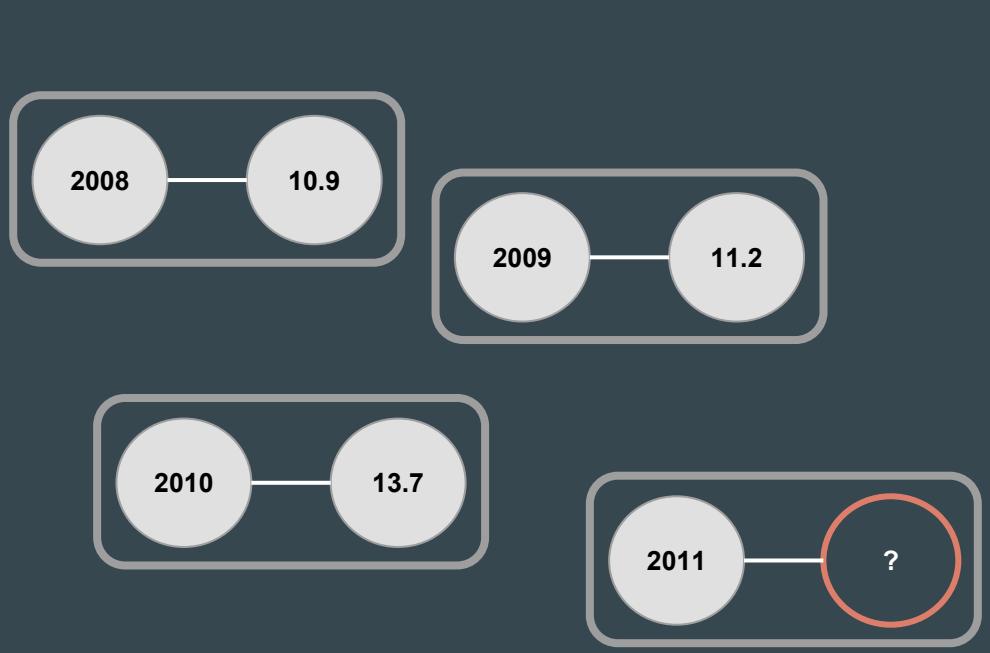
시계열 데이터는 순서가 있는 데이터

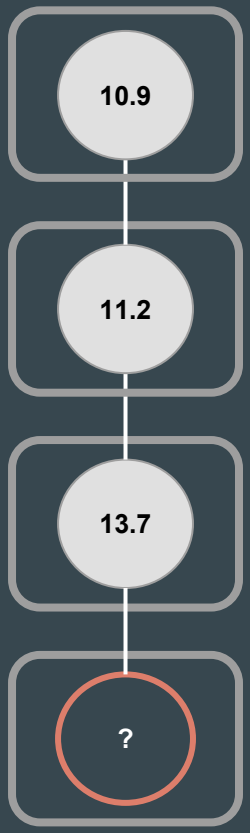


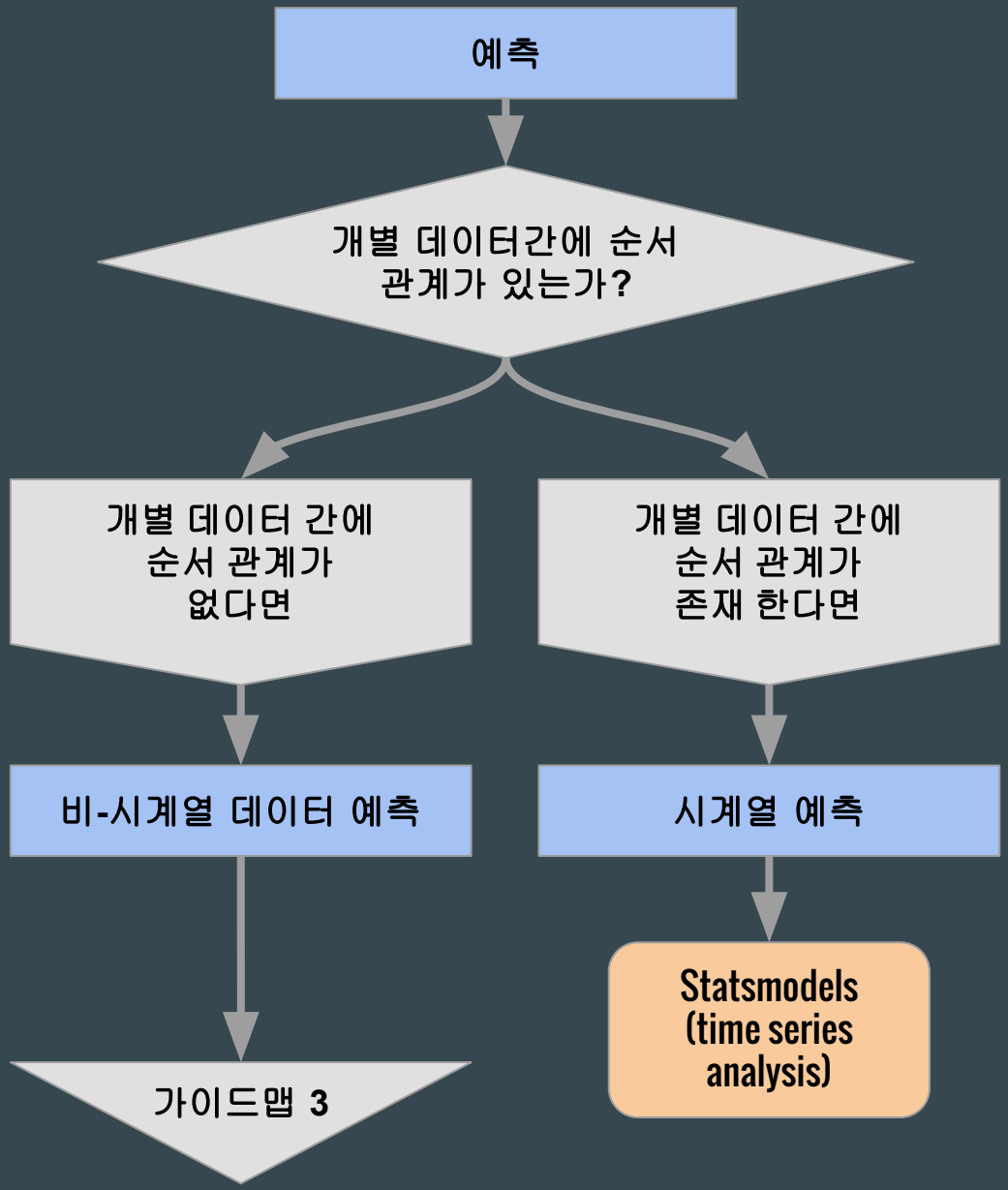


주의:

시간 정보를 이용하면 시계열 분석이 **아니다**






예측
개별 데이터간에 순서
관계가 있는가?
개별 데이터 간에
순서 관계가
없다면
개별 데이터 간에
순서 관계가
존재 한다면
비-시계열 데이터 예측
시계열 예측
Statsmodels
(time series
analysis)
가이드맵 3

데이터 분석 가이드맵 2

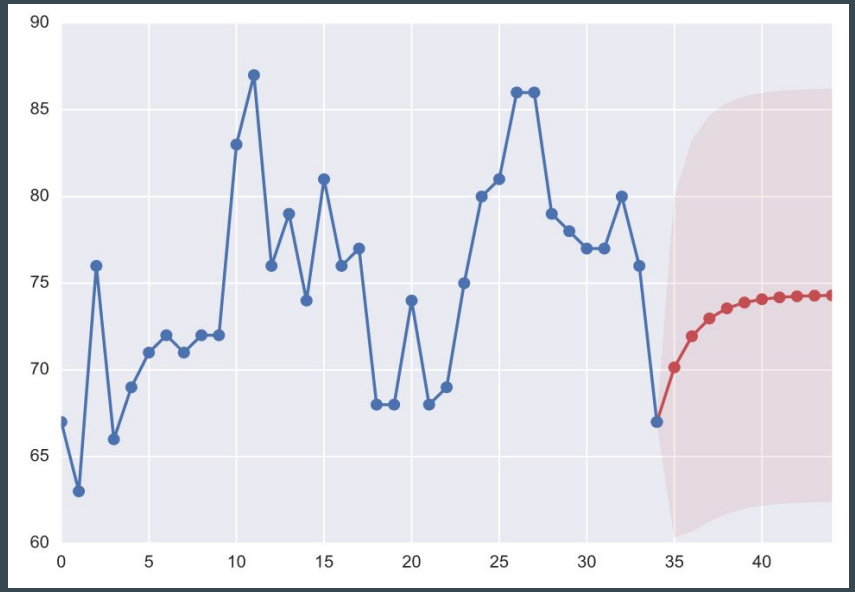
statsmodels.tsa 에서 지원하는 시계열 모형

- ARMA (Auto-Regressive Moving Average)
  - 단일 정상(stationary) 시계열
- ARIMA (Auto-Regressive Integrated Moving Average)
  - 단일 비정상(non-stationary) 시계열
- SARIMA (Seasonal Auto-Regressive Integrated Moving Average)
  - 계절성을 가지는 시계열
- VARMA (Vector Auto-Regressive Moving Average)
  - 함께 움직이는 복수개의 정상(stationary) 시계열
- Kalman filter, State space model
  - 보이지 않는 상태 변수(states)에 의해 움직이는 시계열
- Structural unobserved components model
  - 특정한 모형을 따르는 상태 변수의 합으로 이루어진 시계열
- Dynamic factor model
  - 보이지 않는 요인 시계열의 합으로 이루어진 시계열

ARMA 모형 시계열 예측의 예

- 화학 공정 Quality 예측. Cryer & Chan (2008)
- https://www.datascienceschool.net/view-notebook/4cdb363d9cac4a1381ed0ac7498e3e1c/

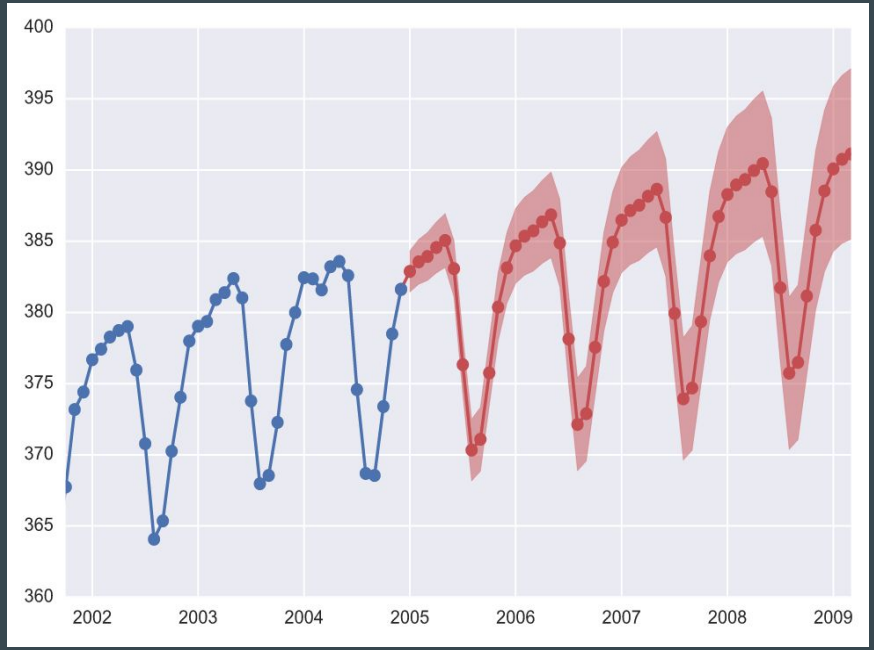
```
# 전처리 - 데이터 준비
df  # 시계열 데이터 (panda DataFrame)


import statsmodels.api as sm

m = sm.tsa.ARMA(df, (1, 0))
r = m.fit()
fred, se, confint = r.forecast(10)

# 후처리 - 시각화
```



SARIMA 모형 시계열 예측의 예

- 대기중 CO2 수준 예측. Cryer & Chan (2008)
- https://www.datascienceschool.net/view-notebook/8c4f6ad9487149ca872374bbbf098e5f

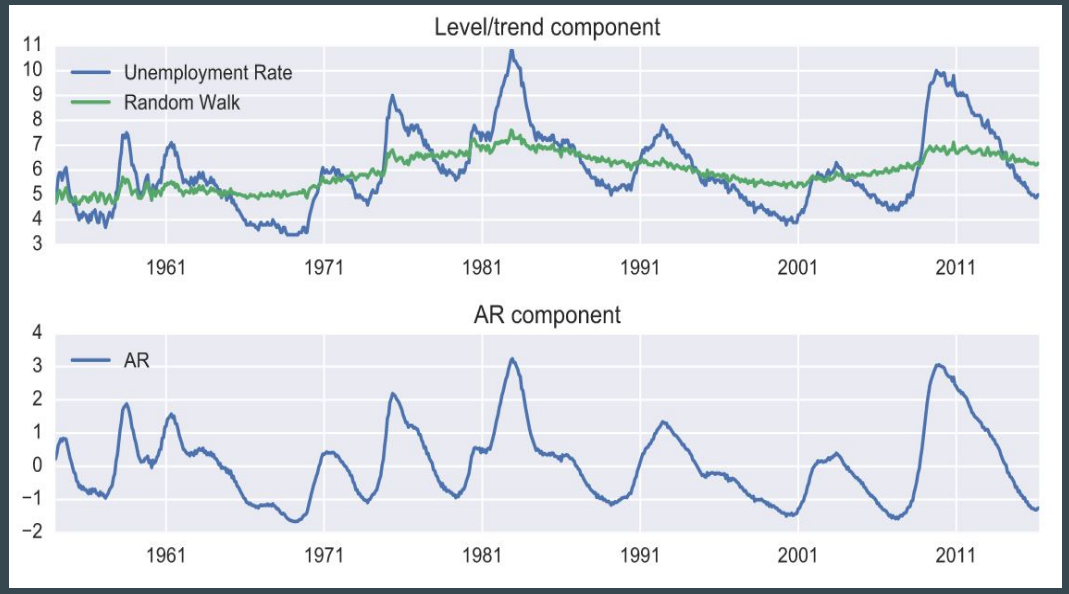
```
# 전처리 - 데이터 준비
df  # 시계열 데이터 (panda DataFrame)


import statsmodels.api as sm

m = sm.tsa.SARIMAX(df, order=(0,1,1),
                   seasonal_order=(0,1,1,12))
r = m.fit()
pred = r.get_prediction(start=len(df),
                        end=len(df)+50)

# 후처리 - 시각화
```



구조화 모형 시계열 분리의 예

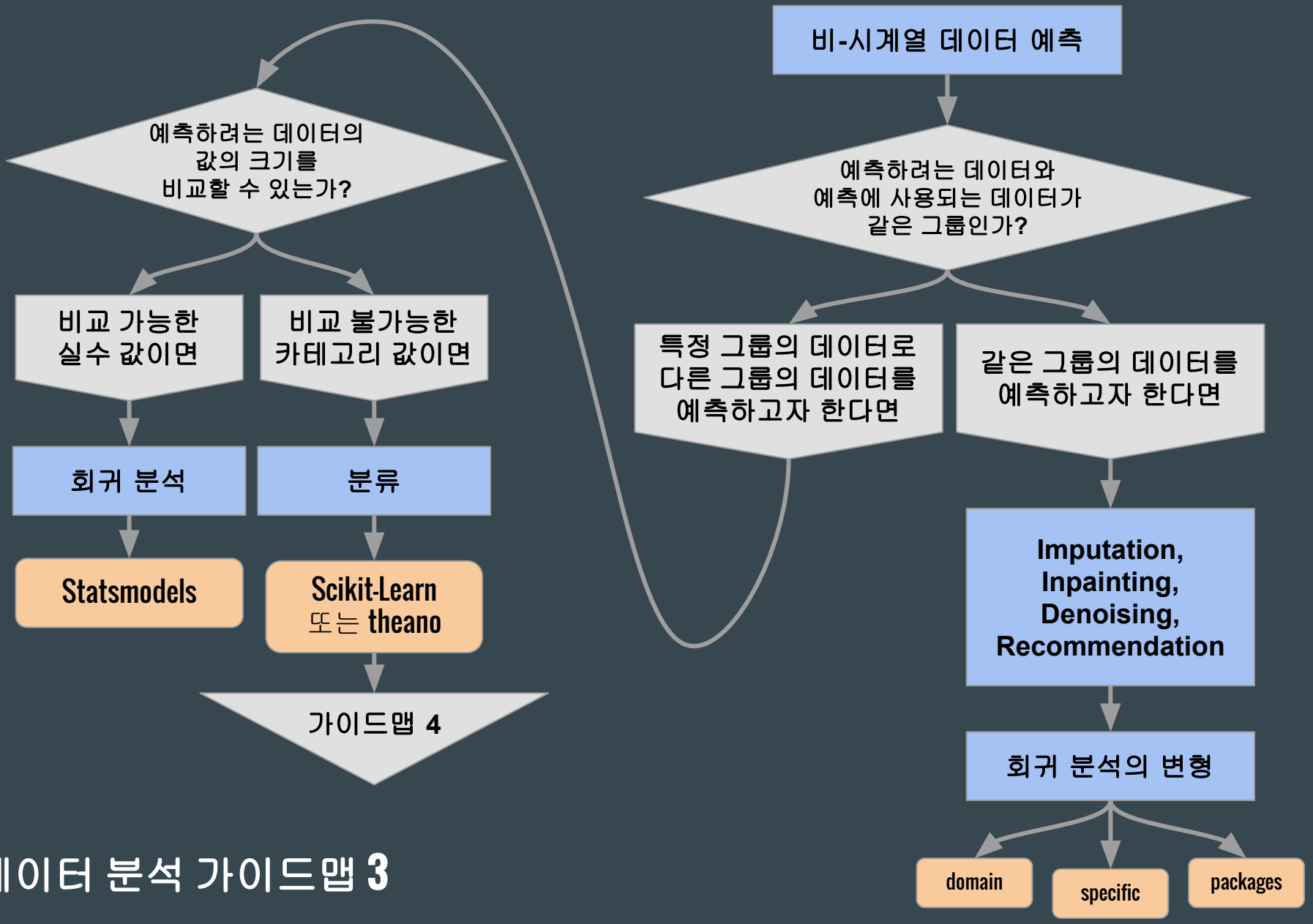
- 미국 실업률 주기 분석. Harvey and Jaeger (1993)
- http://www.statsmodels.org/dev/examples/notebooks/generated/statespace_cycles.html

```
# 전처리 - 데이터 준비
df  # 시계열 데이터 (panda DataFrame)


import statsmodels.api as sm

m = sm.tsa.UnobservedComponents(
        df, 'rwalk', autoregressive=4)
r = m.fit()

# 후처리 - 시각화
```



데이터의 **속성**에 따라
분석 방법이 달라진다.

어떤 **그룹**에 속하는가?
값의 **크기**가 있는가?


비-시계열 데이터 예측
예측하려는 데이터와
예측에 사용되는 데이터가
같은 그룹인가?
특정 그룹의 데이터로
다른 그룹의 데이터를
예측하고자 한다면
같은 그룹의 데이터를
예측하고자 한다면
Imputation,
Inpainting,
Denoising,
Recommendation
회귀 분석의 변형
domain
specific
packages
예측하려는 데이터의
값의 크기를
비교할 수 있는가?
비교 가능한
실수 값이면
비교 불가능한
카테고리 값이면
회귀 분석
분류
Statsmodels
Scikit-Learn
또는 theano
가이드맵 4

데이터 분석 가이드맵 3

회귀 분석
Regression Analysis

vs

분류
Classification

실수 값

크기를 알 수 있고
크기의 비교가 가능한 값
이산(서로 떨어진) 여부는 관계 없습니다!

- 부동산 가격
- 주가
- 매출
- 평가(별점, 학점)

카테고리 값

서로 떨어진 값이면서
크기의 비교가 불가능한 값

- 사람의 이름
- 사물의 명칭
- 장르
- 감성 (Sentiment)

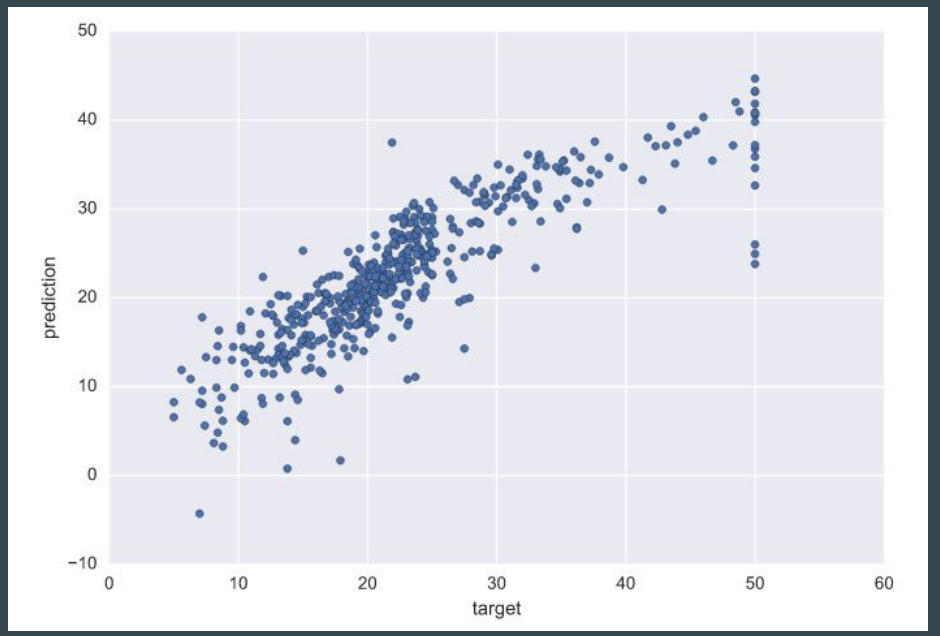
회귀 분석의 예: Boston House Price

- https://archive.ics.uci.edu/ml/datasets/Housing
- 보스턴 주택 가격 예측
- Feature
  - CRIM: 범죄율
  - INDUS: 비소매상업지역 면적 비율
  - NOX: 일산화질소 농도
  - RM: 주택당 방 수
  - LSTAT: 인구 중 하위 계층 비율
  - B: 인구 중 흑인 비율
  - PTRATIO: 학생/교사 비율
  - ZN: 25,000 평방피트를 초과 거주지역 비율
  - CHAS: 찰스강의 경계에 위치한 경우는 1, 아니면 0
  - AGE: 1940년 이전에 건축된 소유주택의 비율
  - RAD: 방사형 고속도로까지의 거리
  - DIS: 직업센터의 거리
  - TAX: 재산세율

분석 코드

- OLS (Ordinary Least Square) 방법 사용
- https://www.datascienceschool.net/view-notebook/58269d7f52bd49879965cdc4721da42d/

```
# 전처리 - 데이터 X, y 준비

import statsmodels.api as sm
m = sm.OLS(y, X)
r = m.fit()

# 후처리 - 리포트
print(r.summary())
```



Results

Dep. Variable:	MEDV	R-squared:	0.741
Model:	OLS	Adj. R-squared:	0.734
Method:	Least Squares	F-statistic:	108.1
Date:	Fri, 03 Jun 2016	Prob (F-statistic):	6.95e-135
Time:	11:46:38	Log-Likelihood:	-1498.8
No. Observations:	506	AIC:	3026.
Df Residuals:	492	BIC:	3085.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	36.4911	5.104	7.149	0.000	26.462	46.520
CRIM	-0.1072	0.033	-3.276	0.001	-0.171	-0.043
ZN	0.0464	0.014	3.380	0.001	0.019	0.073
INDUS	0.0209	0.061	0.339	0.735	-0.100	0.142
CHAS	2.6886	0.862	3.120	0.002	0.996	4.381
NOX	-17.7958	3.821	-4.658	0.000	-25.302	-10.289
RM	3.8048	0.418	9.102	0.000	2.983	4.626
AGE	0.0008	0.013	0.057	0.955	-0.025	0.027
DIS	-1.4758	0.199	-7.398	0.000	-1.868	-1.084
RAD	0.3057	0.066	4.608	0.000	0.175	0.436
TAX	-0.0123	0.004	-3.278	0.001	-0.020	-0.005
PTRATIO	-0.9535	0.131	-7.287	0.000	-1.211	-0.696
B	0.0094	0.003	3.500	0.001	0.004	0.015
LSTAT	-0.5255	0.051	-10.366	0.000	-0.625	-0.426

Omnibus:	178.029	Durbin-Watson:	1.078
Prob(Omnibus):	0.000	Jarque-Bera (JB):	782.015
Skew:	1.521	Prob(JB):	1.54e-170
Kurtosis:	8.276	Cond. No.	1.51e+04

분류의 예: 20 Newsgroups

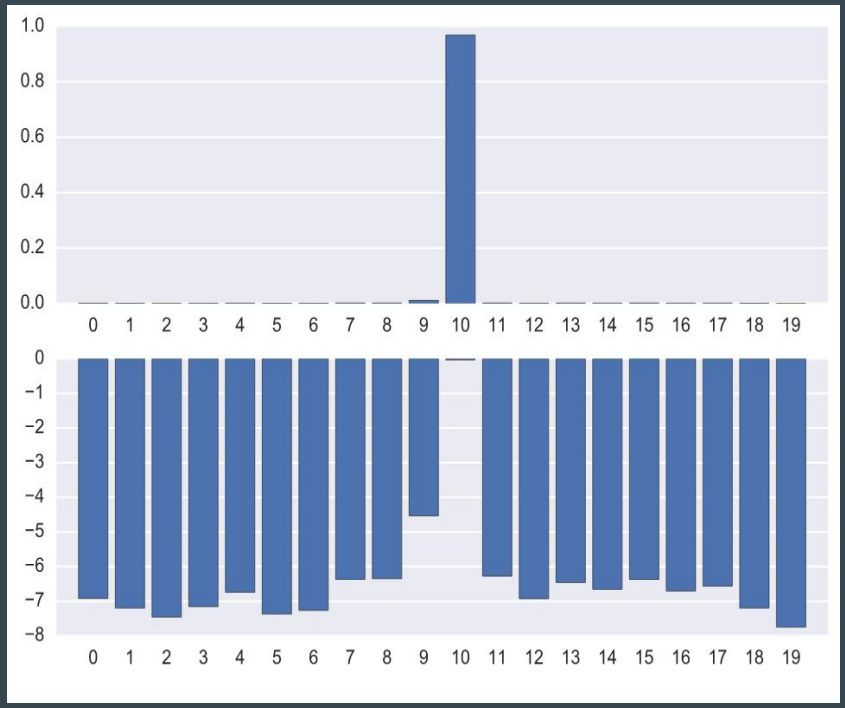
- http://qwone.com/~jason/20Newsgroups/
- 문서가 속하는 뉴스 그룹 예측
  - Comp.graphics
  - Comp.os.ms-windows.misc
  - Comp.sys.ibm.pc.hardware
  - Comp.sys.mac.hardware
  - Comp.windows.x
  - Rec.autos
  - Rec.motorcycles
  - Rec.sport.baseball
  - Rec.sport.hockey
  - Sci.crypt
  - Sci.electronics
  - Sci.med
  - Sci.space
  - Misc.forsale
  - Talk.politics.misc
  - Talk.politics.guns
  - Talk.politics.mideast
  - Talk.religion.misc
  - Alt.atheism
  - Soc.religion.christian

```
From: Mamatha Devineni Ratnam
<mr47+@andrew.cmu.edu>
Subject: Pens fans reactions
Organization: Post Office, Carnegie Mellon,
Pittsburgh, PA
Lines: 12
NNTP-Posting-Host: po4.andrew.cmu.edu


I am sure some bashers of Pens fans are pretty
confused about the lack of any kind of posts about the
recent Pens massacre of the Devils. Actually, I am
bit puzzled too and a bit relieved. However, I am
going to put an end to non-PIttsburghers' relief with
a bit of praise for the Pens. Man, they are killing
those Devils worse than I thought. Jagr just showed
you why he is much better than his regular season
stats. He is also a lot fo fun to watch in the
playoffs. Bowman should let JAgr have a lot of fun in
the next couple of games since the Pens are going to
beat the pulp out of Jersey anyway. I was very
disappointed not to see the Islanders lose the final
regular season game.

PENS RULE!!!
```

분석 코드

- Naive Bayesian 방법 사용
- https://www.datascienceschool.net/view-notebook/58269d7f52bd49879965cdc4721da42d/

```
# 전처리 - 데이터 feature, target 준비


from sklearn.feature_extraction.text \
    import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline

model = Pipeline([
    ('vect', TfidfVectorizer(stop_words="english")),
    ('nb', MultinomialNB()),
])
model.fit(feature, target)

# 후처리 - 시각화
```



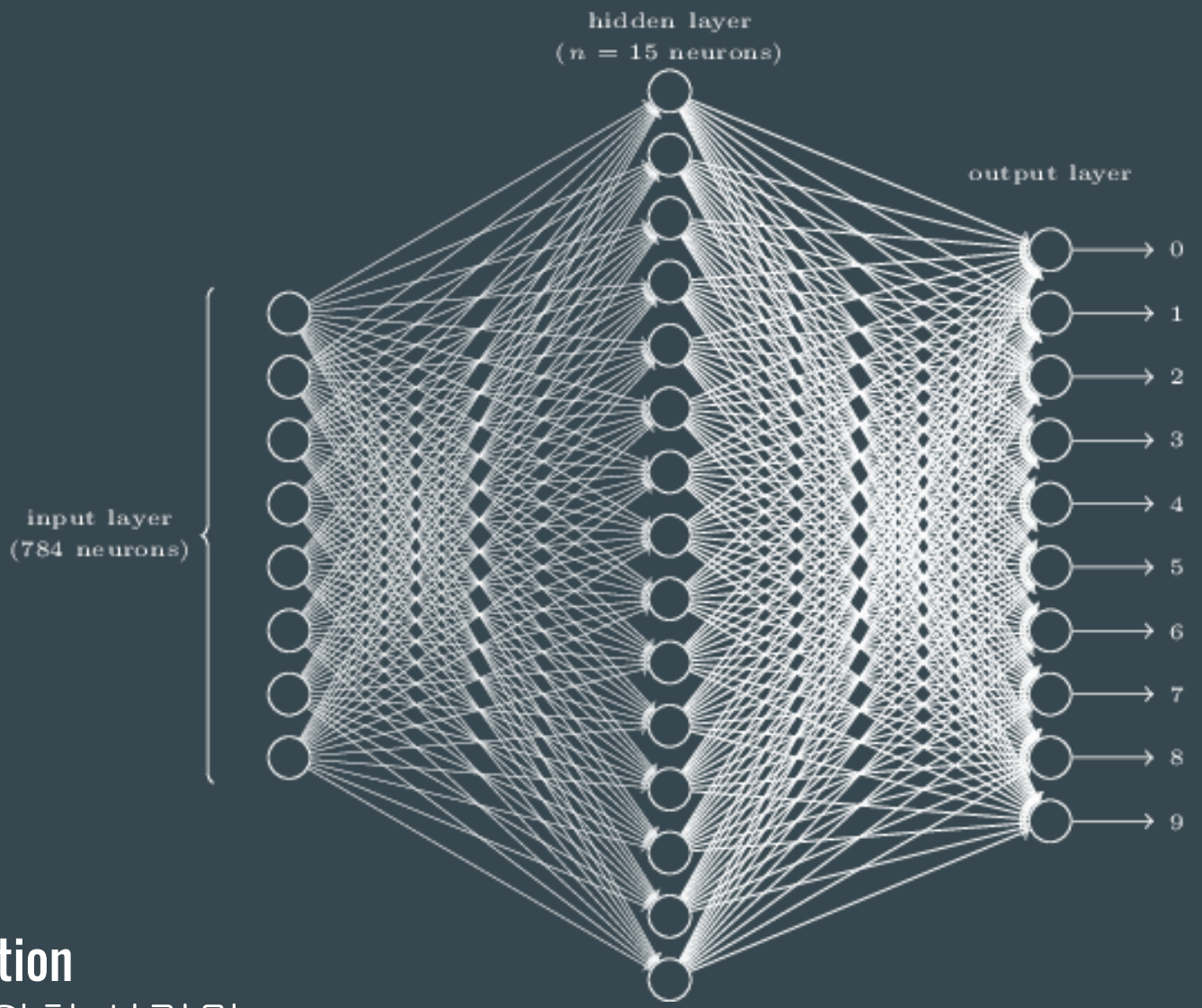
분류 **모형**을 선택한다.


분류
어떤 모형(알고리즘)을 사용하는가?
Logistic Regression
Naive Bayesian
Support Vector Machine
Decision Tree
Random Forest
AdaBoost
....
Neural Network
Convolutional Neural Network
Recurrent Neural Network
Long Short-Term Memory
Restricted Boltzmann Machine
Deep Belief Network
....
Shallow learning
단순한 목적함수 형상
적은 수의 파라미터
적은 계산량
Deep learning
복잡한 목적함수 형상
많은 수의 파라미터
많은 계산량
Scikit-Learn
Theano / TensorFlow
Keras
Lasagne
Blocks

데이터 분석 가이드맵 4

딥러닝 Deep Learning

- 모형의 자유도와 파라미터의 수를 극대화한 데이터 분석 모형입니다.
- 현실 시계의 복잡한 데이터 관계를 모형화하는데 뛰어난 성능을 보입니다.
- 그러나 계산량이 많고 파라미터 값의 계산 시간이 오래 걸립니다.



MNIST Digit Recognition

필기 숫자 인식을 위한 신경망

http://neuralnetworksanddeeplearning.com/chap1.html#a_simple_network_to_classify_handwritten_digits

Theano : Math Expression Compiler

- Theano는 대량/고속의 수학 연산용 파이썬 라이브러리입니다.
- GPU 용 코드를 자동으로 생성합니다.
- 수식의 심볼릭 그래프(Symbolic Graph)를 이용하여 수식 계산을 최적화합니다.
- 함수의 편미분을 심볼릭 방식으로 계산합니다.
- 그러나 프로그래밍이 복잡하고 확장이 어렵습니다.
- 그래서 일반적으로 Theano 를 기반으로 한 고수준 패키지를 많이 사용합니다.

Theano 코드의 예

https://github.com/mnielsen/neural-networks-and-deep-learning/blob/master/src/network3.py

```
class ConvPoolLayer(object):

    def __init__(self, filter_shape, image_shape, poolsize=(2, 2), activation_fn=sigmoid):
        self.filter_shape = filter_shape
        self.image_shape = image_shape
        self.poolsize = poolsize
        self.activation_fn=activation_fn
        n_out = (filter_shape[0]*np.prod(filter_shape[2:])/np.prod(poolsize))
        self.w = theano.shared(np.asarray(np.random.normal(loc=0, scale=np.sqrt(1.0/n_out), size=filter_shape),
                               dtype=theano.config.floatX), borrow=True)
        self.b = theano.shared(np.asarray(np.random.normal(loc=0, scale=1.0, size=(filter_shape[0],)),
                               dtype=theano.config.floatX), borrow=True)
        self.params = [self.w, self.b]

    def set_inpt(self, inpt, inpt_dropout, mini_batch_size):
        self.inpt = inpt.reshape(self.image_shape)
        conv_out = conv.conv2d(input=self.inpt, filters=self.w, filter_shape=self.filter_shape,
                               image_shape=self.image_shape)
        pooled_out = downsample.max_pool_2d(input=conv_out, ds=self.poolsize, ignore_border=True)
        self.output = self.activation_fn(pooled_out + self.b.dimshuffle('x', 0, 'x', 'x'))
        self.output_dropout = self.output
```

고수준 딥러닝 패키지의 예: Keras

- 신경망에 필요한 요소를 빌딩 블럭(building block)으로 구현하고 있습니다.
- 사용자는 레고 조립처럼 블럭을 연결하기만 하면 됩니다.

```
from keras.models import Sequential
from keras.layers.core import Dense, Dropout
from keras.optimizers import SGD


model = Sequential()
model.add(Dense(30, input_dim=784, activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(10, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adadelta', metrics=["accuracy"])
```

감사합니다.

Open Space Time
@210