

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

데이터의 전반적인 특성을 보존하면서 데이터의 변수 수를 줄이는 방법

크기가 10행 7열인 dataframe의 차원은 (10x7)

변수 3개를 제거하면 10행 4열로 바뀜. 이때의 차원은 (10x4)

방법	종류
특성 선택	가장 중요한 특성들만 선택하여 기존의 데이터를 표현
특성 추출	기존 특성들을 사용하여 새로운 특성들을 만들어내는 방법

주성분 분석(PCA)

데이터의 가장 큰 분산을 가진 방향으로 차원을 축소하여 데이터를 표현하는 방법

이를 통해 데이터를 가장 잘 설명하는 주요 특성들을 찾을 수 있다.

데이터의 복잡성을 줄여주어 다양한 분야에서 활용되며, 머신러닝, 패턴 인식, 시각화, 데이터 압축 등 다양한 분야에서 중요한 기술로 사용

#01. 작업준비

패키지 참조

scikit-learn, pca 패키지의 설치가 필요하다

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

```
# 주성분 분석용 패키지
from sklearn.decomposition import PCA
# 주성분 분석 결과를 DataFrame으로 확인할
from pca import pca
# 표준화 처리 패키지
from sklearn.preprocessing import StandardScaler

from pandas import read_excel, DataFrame

from matplotlib import pyplot as plt
import seaborn as sb

import sys
import os
sys.path.append(os.path.dirname(os.path.abspath(__file__)))
from helper import my_ols
```

데이터 가져오기

```
df = read_excel("https://data.hossam.lk/data/medv.xlsx")
df.drop('CAT. MEDV', axis=1, inplace=True)
df.head()
```

	CRIM	ZN	INDUS	CHAS	NOX
0	0.00632	18.0	2.31	0	0.538
1	0.02731	0.0	7.07	0	0.469
2	0.02729	0.0	7.07	0	0.469
3	0.03237	0.0	2.18	0	0.458
4	0.06905	0.0	2.18	0	0.458

#02. 데이터 전처리

독립변수 컬럼만 추출

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

```
x_train = df.drop("MEDV", axis=1)
x_train
```

	CRIM	ZN	INDUS	CHAS	NOX
0	0.00632	18.0	2.31	0	0.538
1	0.02731	0.0	7.07	0	0.468
2	0.02729	0.0	7.07	0	0.468
3	0.03237	0.0	2.18	0	0.458
4	0.06905	0.0	2.18	0	0.458
...
501	0.06263	0.0	11.93	0	0.578
502	0.04527	0.0	11.93	0	0.578
503	0.06076	0.0	11.93	0	0.578
504	0.10959	0.0	11.93	0	0.578
505	0.04741	0.0	11.93	0	0.578

506 rows × 13 columns

추출된 독립변수를 표준화

```
scaler = StandardScaler()
x_train_std = scaler.fit_transform(x_train)
x_train_std
```

```
array([[ -0.41978194,  0.28482986, -1.20261148,
         0.44105193, -1.0755623 ],
       [ -0.41733926, -0.48772236, -0.57807661,
         0.44105193, -0.49243937],
       [ -0.41734159, -0.48772236, -0.57807661,
         0.39642699, -1.2087274 ],
       ...,
       [ -0.41344658, -0.48772236,  0.13420439,
         0.44105193, -0.98304761],
```

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

```
[[-0.40776407, -0.48772236, 0.1
  0.4032249, -0.86530163],
 [-0.41500016, -0.48772236, 0.1
  0.44105193, -0.66905833]])
```

표준화 결과를 데이터프레임으로 재구성

```
std_df = DataFrame(x_train_std, columns=std_df.head())
```

	CRIM	ZN	INDUS	...
0	-0.419782	0.284830	-1.287909	-0.27
1	-0.417339	-0.487722	-0.593381	-0.27
2	-0.417342	-0.487722	-0.593381	-0.27
3	-0.416750	-0.487722	-1.306878	-0.27
4	-0.412482	-0.487722	-1.306878	-0.27

#02. Sklearn을 사용한 PCA 분석

```
# 주성분 분석 객체 생성 (n_components: 주
model = PCA(n_components=5)
fit = model.fit_transform(std_df)
fit
```

```
array([[ -2.09829747,  0.77311275,  0.3
        [-1.45725167,  0.59198521, -0.6
        [-2.07459756,  0.5996394 ,  0.1
        ... ,
        [-0.31236047,  1.15524644, -0.4
        [-0.27051907,  1.04136158, -0.5
        [-0.12580322,  0.76197805, -1.2
```

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

결과 확인

`sklearn`의 PCA 분석은 기존의 변수들을 토대로 `n_component` 수 만큼의 새로운 변수를 생성한다.

대부분 머신러닝의 학습 데이터를 생성하는 용도

```
pca_df = DataFrame(fit)
pca_df
```

	0	1	2	
0	-2.098297	0.773113	0.342943	-0
1	-1.457252	0.591985	-0.695199	-0
2	-2.074598	0.599639	0.167122	-0
3	-2.611504	-0.006871	-0.100284	-0
4	-2.458185	0.097712	-0.075348	-0
...
501	-0.314968	0.724285	-0.860896	-0
502	-0.110513	0.759308	-1.255979	-0
503	-0.312360	1.155246	-0.408598	-0
504	-0.270519	1.041362	-0.585454	-0
505	-0.125803	0.761978	-1.294882	-0

506 rows × 5 columns

#03. pca 패키지를 사용한 분석 (추천)

```
# 주성분 분석의 대상 컬럼 수를 독립변수의 총 수로 지정
model = pca(n_components=len(std_df.columns))
# 표준화 결과를 활용하여 주성분 분석 수행
```

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

```
fit = model.fit_transform(std_df)
fit
```

```
[pca] >Extracting column labels from data
[pca] >Extracting row labels from data
[pca] >The PCA reduction is performed
[pca] >Fit using PCA.
[pca] >Compute loadings and PCs.
[pca] >Compute explained variance.
[pca] >Outlier detection using Hotelling's test
[pca] >Multiple test correction applied
[pca] >Outlier detection using SPE/Dmod
```

```
{'loadings':
      CRIM      ZN
PC1    0.250951 -0.256315  0.346672
PC2   -0.315252 -0.323313  0.112493
PC3    0.246566  0.295858 -0.015946
PC4    0.061771  0.128712  0.017146
PC5    0.082157  0.320617 -0.007811
PC6   -0.219660 -0.323388 -0.076138
PC7    0.777607 -0.274996 -0.339576
PC8    0.153350 -0.402680  0.173932
PC9    0.260390  0.358137  0.644416
PC10   0.019369  0.267527 -0.363532
PC11  -0.109644  0.262756 -0.303169
PC12   0.086761 -0.071425 -0.113200
PC13   0.045952 -0.080919 -0.251077
```

```
      DIS      RAD      TAX
PC1  -0.321544  0.319793  0.338469
PC2  -0.349070 -0.271521 -0.239454
PC3  -0.049736  0.287255  0.220744
PC4   0.215436  0.132350  0.103335
PC5   0.098592 -0.204132 -0.130461
PC6   0.023439 -0.143194 -0.192934
PC7  -0.103900 -0.137943 -0.314887
PC8  -0.121812  0.080358  0.082774
PC9  -0.153291 -0.470891 -0.176563
PC10 -0.171213  0.021909 -0.035168
PC11 -0.695693  0.036544 -0.104836
PC12  0.390941 -0.107026 -0.215191
PC13 -0.018299 -0.633490  0.720233
```

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

```

'PC':
PC1      PC2
0  -2.098297  0.773113  0.342943 -0.
1  -1.457252  0.591985 -0.695199 -0.
2  -2.074598  0.599639  0.167122 -0.
3  -2.611504 -0.006871 -0.100284 -0.
4  -2.458185  0.097712 -0.075348 -0.
..      ...      ...      ...
501 -0.314968  0.724285 -0.860896 -0.
502 -0.110513  0.759308 -1.255979 -0.
503 -0.312360  1.155246 -0.408598 -0.
504 -0.270519  1.041362 -0.585454 -0.
505 -0.125803  0.761978 -1.294882 -0.

PC8      PC9      PC10
0  0.295832 -0.424937  0.640206 -0.
1 -0.223670 -0.166962  0.084236 -0.
2  0.105166  0.069775 -0.180380 -0.
3  0.255941 -0.342246  0.045901 -0.
4 -0.134524 -0.417668 -0.140880 -0.
..      ...      ...      ...
501 -0.249896  0.877036  0.183086  0.
502 -0.146502  0.853628  0.631847  0.
503 -0.638660  0.981032  0.589670  0.
504 -0.579344  0.936755  0.594610  0.
505 -0.133382  0.854689  0.823404  0.

[506 rows x 13 columns],
'explained_var': array([0.47129606, 0.8578876 , 0.89906884, 0.9295
0.98209137, 0.99511467, 1.
'variance_ratio': array([0.47129606,
0.05056978, 0.04118124, 0.0304
0.0143088 , 0.01302331, 0.0048
'model': PCA(n_components=13),
'scaler': None,
'pcp': 1.0,
'topfeat':
PC  feature  loading
0  PC1  INDUS  0.346672  best
1  PC2  CHAS  0.454829  best
2  PC3  RM  0.593961  best
3  PC4  CHAS  0.815941  best
4  PC5  PTRATIO -0.584002  best
5  PC6  B -0.803455  best
6  PC7  CRIM  0.777607  best
7  PC8  AGE -0.600823  best

```

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

```

8      PC9      INDUS    0.644416    best
9      PC10     LSTAT   -0.600711    best
10     PC11      DIS    -0.695693    best
11     PC12     NOX     0.804323    best
12     PC13     TAX     0.720233    best
13     PC8      ZN     -0.402680    weak
14     PC13     RAD    -0.633490    weak,
'outliers':          y_proba      p_raw
0      0.999977    0.930684    16.223114
1      0.999977    0.992463    11.756468
2      0.999977    0.952796    15.239485
3      0.999977    0.865636    18.272488
4      0.999977    0.882745    17.805197
..          ...          ...          ...
501    0.999977    0.979935    13.414755
502    0.999977    0.982736    13.134686
503    0.999977    0.905283    17.124208
504    0.999977    0.940977    15.796442
505    0.999977    0.981559    13.256252

[506 rows x 6 columns],
'outliers_params': {'paramT2': (0.0,
'paramSPE': (array([ 4.45131415e-16,
array([[6.13898120e+00, 1.75595512e-14,
[1.75595512e-14, 1.43611329e-14])

```

생성된 주성분에 사용된 필드 확인

```

topfit = fit['topfeat']
topfit

```

	PC	feature	loading	type
0	PC1	INDUS	0.346672	best
1	PC2	CHAS	0.454829	best
2	PC3	RM	0.593961	best
3	PC4	CHAS	0.815941	best
4	PC5	PTRATIO	-0.584002	best
5	PC6	B	-0.803455	best

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

	PC	feature	loading	type
6	PC7	CRIM	0.777607	best
7	PC8	AGE	-0.600823	best
8	PC9	INDUS	0.644416	best
9	PC10	LSTAT	-0.600711	best
10	PC11	DIS	-0.695693	best
11	PC12	NOX	0.804323	best
12	PC13	TAX	0.720233	best
13	PC8	ZN	-0.402680	weak
14	PC13	RAD	-0.633490	weak

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

```
best = topfit.query("type='best'")
feature = list(set(list(best['feature'])))
```

```
['TAX',
 'B',
 'CHAS',
 'PTRATIO',
 'CRIM',
 'NOX',
 'RM',
 'INDUS',
 'LSTAT',
 'AGE',
 'DIS']
```

```
ols = my_ols(df, "MEDV", feature)
```

주성분 분석 (PCA 분석)

ols.summary

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

OLS Regression Results

Dep. Variable:	MEDV	R-squared:	0.72
Model:	OLS	Adj. R-squared:	0.71
Method:	Least Squares	F-statistic:	118.
Date:	Wed, 26 Jul 2023	Prob (F-statistic):	9.42e-131
Time:	11:22:15	Log-Likelihood:	-151
No. Observations:	506	AIC:	3051
Df Residuals:	494	BIC:	3102
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P> t
Intercept	30.5585	5.020	6.087	0.000
TAX	0.0029	0.002	1.240	0.216
B	0.0086	0.003	3.112	0.002
CHAS	3.1229	0.880	3.548	0.000
PTRATIO	-0.9655	0.124	-7.774	0.000
CRIM	-0.0619	0.032	-1.905	0.057
NOX	-15.8300	3.880	-4.079	0.000
RM	4.2847	0.420	10.206	0.000
INDUS	-0.0722	0.061	-1.194	0.233
LSTAT	-0.5091	0.052	-9.786	0.000

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

AGE	-0.0083	0.013	-0.616	0.538
DIS	-1.2542	0.187	-6.698	0.000

Omnibus:	192.416	Durbin-Watson:	1.043
Prob(Omnibus):	0.000	Jarque-Bera (JB):	944.67
Skew:	1.617	Prob(JB):	7.36e-206
Kurtosis:	8.861	Cond. No.	1.45e+

Notes:

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

[2] The condition number is large, 1.45e+04. This might indicate that there are strong multicollinearity or other numerical problems.

ols.table

		B	표준 오차	β	
종속 변수	독립 변수				
MEDV	TAX	0.0029	0.002	0	1
	B	0.0086	0.003	0	3
	CHAS	3.1229	0.880	0	3
	PTRATIO	-0.9655	0.124	0	-7
	CRIM	-0.0619	0.032	0	-7
	NOX	-15.8300	3.880	0	-4

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

		B	표준 오차	β	
종속 변수	독립변수				
	RM	4.2847	0.420	0	1
	INDUS	-0.0722	0.061	0	-
	LSTAT	-0.5091	0.052	0	-9
	AGE	-0.0083	0.013	0	-0
	DIS	-1.2542	0.187	0	-6

결과 비교하기

```
실제집값 = df["MEDV"]
실제집값
```

```
0      24.0
1      21.6
2      34.7
3      33.4
4      36.2
...
501     22.4
502     20.6
503     23.9
504     22.0
505     11.9
Name: MEDV, Length: 506, dtype: float64
```

```
예측집값 = ols.fit.predict(df.filter(fe
예측집값
```

```
0      31.319245
1      25.506274
2      31.493779
```

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

```
3      29.616826
4      28.986126
...
501    24.133272
502    22.626843
503    28.076938
504    26.573324
505    22.545271
Length: 506, dtype: float64
```

```
result_df = DataFrame({
    "실제집값":실제집값,
    "예측집값":예측집값
})
result_df
```

	실제집값	예측집값
0	24.0	31.319245
1	21.6	25.506274
2	34.7	31.493779
3	33.4	29.616826
4	36.2	28.986126
...
501	22.4	24.133272
502	20.6	22.626843
503	23.9	28.076938
504	22.0	26.573324
505	11.9	22.545271

506 rows × 2 columns

```
plt.rcParams["font.family"] = 'AppleGothic'
plt.rcParams["font.size"] = 12
plt.rcParams["figure.figsize"] = (20, 10)
plt.rcParams["axes.unicode_minus"] = True
```

주성분 분석 (PCA 분석)

#01. 주성분 분석 개요

차원 축소 (Dimensionality Reduction)

주성분 분석(PCA)

#01. 작업준비

패키지 참조

데이터 가져오기

#02. 데이터 전처리

독립변수 컬럼만 추출

추출된 독립변수를 표준화

표준화 결과를 데이터프레임으로 재구성

#02. Sklearn을 사용한 PCA 분석

결과 확인

#03. pca 패키지를 사용한 분석 (추천)

생성된 주성분에 사용된 필드 확인

#04. 주성분 분석 결과를 토대로 회귀분석 수행

원본 데이터프레임으로 분석

결과 비교하기

```
sb.lineplot(data=result_df.filter(['실
plt.show()
plt.close()
```

