

3

Selection

Your time is limited, so don't waste it living someone else's life.

–Steve Jobs

개요

개념

- 경험이나 이론에 의해 목표변수(Y)에 영향을 미칠 것 같은 예측변수를 선택하고 선형 회귀모형을 설정한다. => 데이터를 수집 후 회귀모형의 회귀계수를 추정하고(OLS) 분산분석의 F-검정에 의해 모형의 유의성을 검정한다. ($H_0: \beta_1 = \beta_2 = \dots = \beta_p = 0$, 모든 설명변수는 유의하지 않다) => F-검정(분산분석) 결과 귀무가설이 기각되면 t-검정을 이용하여 회귀계수(설명변수)의 유의성 검정을 한다.
- 오컴의 면도날(Occam's Razor) : 경제성의 원리(Principle of economy), 절약성의 원리(Principle of parsimony) - 모든 상황이 동일하다면 가장 간단한 방법이 최선이다. 즉, 회귀모형이 동일한 정보를 준다면(설정 모형이 예측변수의 변동을 설명하는 비율) 적은 예측변수 모형이 최선의 모형이다.
- 유의하지 않은 설명변수를 하나씩 제외하면서 모든 변수가 유의할 때까지 계속한다. 어떤 설명변수를 제외할 것인가? 가장 유의하지 않은 설명변수, 즉 t-검정통계량 값이 가장 작은 것(p-value가 가장 큰 것)을 제외한다.

모형 유의성 검정 - F 검정

- 귀무가설 : 모든 설명변수가 유의하지 않음 - 선택되면 분석 종료하고 귀무가설이 기각되어 모형 내 설정한 설명변수 중 적어도 하나라도 유의하면 다음 단계로 넘어간다.

개별 설명변수 - t 검정

- 유의하지 않은 순서대로 하나씩 제외함 : 유의확률이 가장 크거나 t-통계량이 가장 작은 순서대로 모형 내 모든 설명변수가 유의할 때까지 반복 제거함

Comment

다중공선성 분석과 순서 문제

대부분의 교과서에서는 다중공선성 진단을 먼저 설명하고 있으나 어느 단계를 먼저해도 동일 결과를 얻는 경우가 대부분이므로 간편 작업(변수선택 과정을 먼저 거치면 다중공선성 진단을 하면 진단 변수의 수 줄어듦)을 위하여 변수선택을 먼저하는 것이 적절하다.

반드시 필요한가?

예측이 목적이라면 설명력이 전혀 없는(목표변수와 상관관계수 0) 예측변수는 필드에서는 존재하지 않으므로 미미하지만 목표변수를 설명하므로 선택, 제외할 필요는 없다. 그러나 너무 많은 변수가 존재하면 과적합 문제가 발생하므로 데이터 개수에 비해 변수의 개수가 많아지는 경우 차원 축소 측면에서 변수선택은 필요하다.

통계량 이용

$$\text{결정계수 Determination Coefficient } R_p^2 = \frac{SSR_p}{SST} = 1 - \frac{SSE_p}{SST}$$

- R_p^2 는 모형내의 p 개 설명변수들의 종속변수 (총)변동에 대한 설명력의 정도를 나타내는 수치이므로 변수선택의 지표가 된다.
- 결정계수의 크기가 70%이면 설명변수 수가 충분함, 80% 이상이면 매우 충분하다고 판단함 - 현재 선택된 설명변수만으로 종속변수 변동을 예측할(설명) 수 있음
- 설명변수의 개수가 같은 경우 어떤 변수 그룹이 설명력이 높은가를 쉽게 알아보는 사용할 수 있으나 검정 통계량은 존재하지 않는 단점이 있다.
- 그러나 설명변수 개수가 증가할 때 마다 결정계수는 항상 증가하므로 변수의 개수가 다른 경우에는 수정 결정 계수를 사용하는 것이 좋다.

$$\text{수정결정계수 } R_{adj}^2 = 1 - \frac{SSE_p/(n - p - 1)}{SST/n - 1}$$

- 수정(adjusted) 결정계수는 결정계수의 문제점(유의하지 않은 설명변수가 삽입되어도 항상 증가)을 해결하기 위하여 계산되는 척도
- 그러나 설명변수의 증가로 줄어드는 SSE의 값과 오차 자유도의 감소가 상쇄되므로 설명력의 지표로 좋은 것은 아니다. 그리고 유의성을 검정할 검정통계량이 존재하지 않는다.
- 수정결정계수는 설명변수 군과 개수가 서로 다른 회귀모형의 설명력을 비교할 때 사용된다.
- 종속변수의 예측 값을 높이는 모형을 선택하는 것이 분석 목적이라면 이 방법을 이용하여 최적 모형을 찾으면 됨

$$\text{Mallow } C_p = \frac{SSE_p}{MSE_F} - n + 2(p + 1)$$

- Mallow C_p 값이 설명변수 개수와 근사한 경우 좋은 회귀 모형으로 판단한다.
- 여전히 이 방법에서도 각 설명변수의 유의성에 대한 검정을 실시한 것은 아니다.
- (이론적 근거) $MSE(F) = \sigma^2$ 이고 만약 모형이 적합하다면 $C_p \approx (p + 1)$ Mallow C_p 는 변수의 개수 +1(절편)와 같다 .

$$PRESS_p = \sum (y_i - \hat{y}_{(i)})^2$$

예측잔차자승합(Prediction REsidual Sum of Squares)

- $\hat{y}_{(i)}$: i -번째 관측치를 제외한 후 모형을 추정한 후 구한 종속변수 y_i 의 적합값
- i -번째 관측치 제외 잔차 = $(y_i - \hat{y}_{(i)})$
- 잔차자승합 PRESS 값이 적을수록 좋은 모형

Comment

- (수정)결정계수, Mallow , PRESS 통계량 이용 방법은 어떤 변수 군들이 좋은지에 대한 지표만을 제공할 뿐 변수의 유의성은 검정되지 않았다.
- 일반적으로 이 방법들을 이용할 때는 수정 결정 계수와 Mallow C_p 에 의해 좋다고 간주되는 변수 군들을 몇 개 선택하고 각 변수 군에 대해 PRESS 값을 계산하여 최적 변수 군을 선택하면 된다.
- PRESS는 전혀 다른 변수 군을 비교할 때(변수들이 모두 유의한 경우) 사용되므로 널리 사용된다.
- 서로 다른 변수 군의 적합 정도 비교 시 사용된다. AIC, SBC 모두 작을수록 적합도가 높다.

AIC(Akaike Information Criteria)

$$AIC = n * \ln(SSE/n) + 2(p - 1)$$

SBC(Schwarz's Bayesian criterion)

$$SBC = n * \ln(SSE/n) + (p - 1)\ln(n)$$

유의성 검정

추가 자승합

Full Model

모형 : $\underline{y}_{n \times 1} = X_{n \times (p+1)} \underline{b}_{(p+1) \times 1} + \underline{e}_{n \times 1}$, 설명변수 벡터 $\underline{x} = (x_1, x_2, \dots, x_p)'$

총변동 SST_F . 모형변동 $SSR_F(df = p)$, 오차변동 $SSE_F(df = n - p - 1)$

$$\text{오차분산}(\sigma^2) \text{ 추정량} = MSE_F = \frac{SSE_F}{n - p - 1}$$

Reduced Model

귀무가설 : 일부 설명변수가 유의하지 않다. 유의하지 않은 설명변수 개수 = $p - k$

모형 : $\underline{y}_{n \times 1} = X_{n \times (k+1)} \underline{b}_{(k+1) \times 1} + \underline{e}_{n \times 1}$, 설명변수 벡터 $\underline{x} = (x_1, x_2, \dots, x_k)'$, $k < p$

총변동 SST_R . 모형변동 $SSR_R(df = k)$, 오차변동 $SSE_R(df = n - k - 1)$

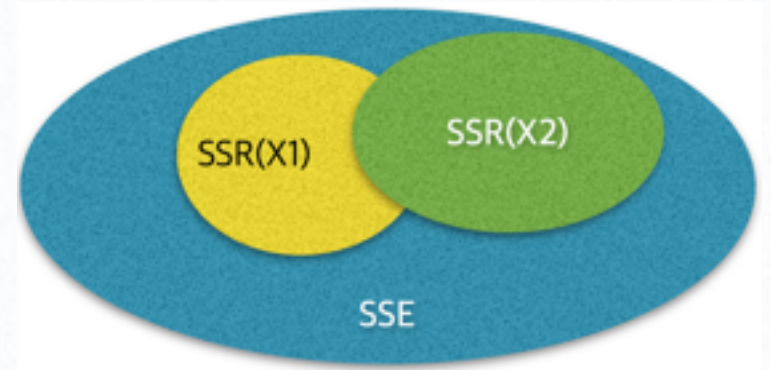
검정통계량

$$\frac{SSR_F - SSR_R}{MSE_F} = \frac{SSE_R - SSE_F}{MSE_F} \sim F(p - k, n - p - 1)$$

- 분자의 자유도는 귀무가설에 의해 줄어든 모수의 개수가 된다.
- 즉 full 모형의 설명변동과 reduced 모형의 설명변동의 차이가 유의적인가를 검정하여 귀무가설을 기각, 채택하게 된다.

추가 자승합(Extra Sum of Squares) 정의

- $Y \leftarrow \text{Reg}(X_1, X_2)$ 모형변동 $SSR(X_1, X_2)$, 오차변동 $SSE(X_1, X_2)$
- $Y \leftarrow \text{Reg}(X_1)$ 모형변동 $SSR(X_1)$, 오차변동 $SSE(X_1)$



예측변수 X_1 이 목표변수 Y 의 총변동을 설명하고 남은 변동 중 X_2 예측변수가 설명하는 변동

$$SSR(X_2|X_1) = SSR(X_1, X_2) - SSR(X_1) = SSE(X_1) - SSE(X_1, X_2)$$

회귀모형 활용 방안

Sequential SS, Type I

회귀모형 삽입 예측모형 순서 대로 : $SSR(X_1|\mu)$, $SSR(X_2|\mu, X_1)$, $SSR(X_3|\mu, X_1, X_2)$

- 삽입 순서에 따른 설명변동의 차이가 있음
- 통제변수 모형에 주로 사용된다.

Type II: hierarchical or partially sequential

다른 예측변수가 고정인 경우 해당 변수의 변동 : $SSR(X_1|\mu, X_2, X_3, \dots, X_p)$, $SSR(X_2|\mu, X_1, X_3, \dots, X_p)$

- 유의 변수 선택에 사용된다.
- 교차항이 있을 경우에는 Type III와 상이하다.

Type III: marginal or orthogonal , partial

다른 예측변수가 고정인 경우 해당 변수의 변동 : $SSR(X_1|\mu, X_2, X_3, \dots, X_p)$, $SSR(X_2|\mu, X_1, X_3, \dots, X_p)$

검정방법

귀무가설 : 예측변수 X_k 는 유의하지 않다. $H_0; \beta_k = 0$ [Reduced Model]

- 대립가설 : 설정된 모형이 적절하다. [Full Model]
- 검정통계량 : $TS = \frac{(SSE_{reduced} - SSE_{full})/df}{MSE_{full}} \sim F(1, n - p - 1)$
- 분모의 자유도는 귀무가설에 설정되어 축소된 모수의 개수 - 여기서는 1임
- 분자의 자유도가 1인 경우 $F(1, n) \sim t^2(n)$ 이므로 개별 회귀계수에 대한 t-검정과 동일하다.

귀무가설 : 예측변수 X_1, X_2 모두는 유의하지 않다. $H_0; \beta_1 = \beta_2 = 0$ [Reduced Model]

- 대립가설 : 설정된 모형이 적절하다. [Full Model]
- 검정통계량 : $TS = \frac{(SSE_{reduced} - SSE_{full})/df}{MSE_{full}} \sim F(2, n - p - 1)$

부분 결정계수 Coefficient of Partial Determination

예측변수 X_1 에 의해 설명되고 남은 목표변수 변동을 (X_2, X_3) 두 예측변수가 설명하는 변동 비는 다음과 같이 정의하고 이를 부분 결정계수라 한다.

$$R_{Y.2,3|1}^2 = \frac{SSR(X_2, X_3 | X_1) = SSR(X_1, X_2, X_3) - SSR(X_1)}{SSE(X_1)} = \frac{SSE(X_1) - SSE(X_1, X_2, X_3)}{SSE(X_1)}$$

고전적 변수선택 방법

Backward 후진제거

모든 설명변수를 고려한 모형에서 유의하지 않은 설명변수를 하나씩 제거하는 방법이다.

- 고려된 설명변수를 모두 삽입한 후 설명변수 중 가장 유의하지 않은(유의확률이 가장 큰) 설명변수를 제외
- 가장 유의하지 않다는 것은 다음 검정 결과 유의하지 않고 F값이 가장 작은 설명변수를 의미한다.

$$\frac{SSE_R(\text{no } X_k) - SSE_F}{MSE_F} \sim F(1, n - p - 1)$$

- 모든 설명변수가 유의할 때까지 위의 과정을 반복한다.

전진삽입 forward

고려된 설명변수 중 설명력($SSR(X_k)$)이 가장 높고(유의확률이 가장 작음) 설명력이 유의하면 변수를 선택한다.

$$\max \frac{SSR(X_k)}{MSE(x_k)} \sim F(1, n - 2)$$

이미 선택된 설명변수(X_k)의 설명 부분을 제외하고 남은 종속변수 변동 중 가장 많이 설명하고 ($SSR(X_m | X_k)$ 이 가장 큰) 그 설명력이 유의한 경우 X_m 을 2번째로 선택한다.

$$\max \frac{SSR(X_k, X_m) - SSR(X_k)}{MSE(x_k, x_m)} \sim F(1, n - 3)$$

유의한 설명변수가 없을 때까지 2)의 작업을 반복한다.

stepwise 단계삽입

단계 삽입(stepwise)은 Forward 방법과 유사하지만 한 번 선택된 설명 변수에 대해서는 유의성 검정을 다시 실시한다는 점이 다르다.

- 고려된 설명변수 중 설명력($SSR(X_k)$)이 가장 높고(유의확률이 가장 작음) 설명력이 유의하면 변수를 선택한다. = 첫단계는 전진삽입 방법과 동일함

- 이미 선택된 설명변수(X_k)의 설명 부분을 제외하고 남은 종속변수 변동 중 가장 많이 설명하고 ($SSR(X_m|X_k)$ 이 가장 큰) 그 설명력이 유의한 경우 X_m 을 2번째로 선택한다. = 2번째 단계까지도 동일하다.

$$\frac{SSR(X_k, X_m) - SSR(X_k)}{MSE(x_k, x_m)} \sim F(1, n - 3)$$

- 새롭게 선택된 변수(X_m)를 이용하여 기존 선택된 설명변수()의 유의성을 검증하여 유의하지 않으면 X_k 을 제거한다.

$$\frac{SSR(X_k, X_m) - SSR(X_m)}{MSE(x_k, x_m)} \sim F(1, n - 3)$$

- 유의한 설명변수가 존재하지 않을 때까지 위의 과정을 반복한다.

최적 변수선택 방법

다른 설명 변수가 주어진 경우 각 설명 변수의 설명력을 볼 수 있다는 점에서 어떤 학자들은 변수의 수가 많지 않으면 후진제거 방법을 많은 경우는 전진삽입 방법이 적합한 방법이라 한다. 통계 소프트웨어가 발달하기 전에는 계산이 복잡한 단계삽입(Stepwise) 방법이 선호되지 않았으나 이제는 이 방법이 가장 선호된다.

설명변수 개수가 10개를 넘지 않는 경우는 수작업(t-검정)이 가장 적절하다. 이유는 분석자가 자신의 지식을 이용하여 해석이 용이하거나 더 좋다고 생각하는 변수를 선택 할 수 있기 때문이다.

파이썬 코드

목표변수 Y, 예측변수 X 지정

정규화 변환 데이터 사용하였음 smsa_clean(페이지 37)

```
import statsmodels.api as sm
y=smsa_clean['mortality_index']
X=smsa_clean.drop(columns=['mortality_index','southern'])
```

scikit-learn 모듈은 유의확률에 근거한 변수선택방법은 머신러닝 측면에서는 선호하지 않는다. 하여, 코딩이 필요한 작업이다.

후진제거

유의수준 10%에서 유의한 예측변수를 후진제거 방법으로 선택하는 코드임

```
#Backward Elimination
sig_level=0.1; cols = list(X.columns); pmax = 1
while (len(cols)>0):
    p= []
    X_1 = X[cols]
    X_1 = sm.add_constant(X_1)
    model = sm.OLS(y,X_1).fit()
    p = pd.Series(model.pvalues.values[1:],index = cols)
    pmax = max(p)
    feature_with_p_max = p.idxmax()
    if(pmax>sig_level):
        cols.remove(feature_with_p_max)
    else:
        break
selected_features_BE = cols
print(selected_features_BE)
```

```
['jan_temp', 'rainfall', 'education', 'non_white_ratio_sqrt', 'NOxPot_log']
```


전진삽입

```
import statsmodels.formula.api as smf
def forward_selected(data, response):
    remaining = set(data.columns)
    remaining.remove(response)
    selected = []
    current_score, best_new_score = 0.0, 0.0
    while remaining and current_score == best_new_score:
        scores_with_candidates = []
        for candidate in remaining:
            formula = "{} ~ {} + 1".format(response, ' + '.join(selected + [candidate]))
            score = smf.ols(formula, data).fit().rsquared_adj
            scores_with_candidates.append((score, candidate))
        scores_with_candidates.sort()
        best_new_score, best_candidate = scores_with_candidates.pop()
        if current_score < best_new_score:
            remaining.remove(best_candidate)
            selected.append(best_candidate)
            current_score = best_new_score
    formula = "{} ~ {} + 1".format(response, ' + '.join(selected))
    print('selected variables :',selected,'n')
    model = smf.ols(formula, data).fit()
    return model
```

```
df=smsa_clean.drop(columns=['southern'])
model = forward_selected(df, 'mortality_index')
print(model.model.formula,'\n'); print(model.rsquared_adj)
```

```
➞ selected variables : ['non_white_ratio_sqrt', 'education', 'jan_temp', 'NOxPot_log'
mortality_index ~ non_white_ratio_sqrt + education + jan_temp + NOxPot_log + rainfa
0.7297682020409422
```

빅데이터 방법

최종 회귀모형 선택 방법

고전적 접근

- 서로 설명변수가 고려된 모형 중 최적 모형을 선택하고자 할 때는 MAE(Mean Absolute Error), MSE(Mean Squared Error), RMSE(Root Mean Squared Error) 나 수정결정계수 값을 비교한다.

$$MAE = \sum |y_i - \hat{y}_i|/n, \quad MSE = \sum (y_i - \hat{y}_i)^2/n, \quad RMSE = \sqrt{MSE}$$

- 새로운 자료(표본크기 n^*) 수집하고 모형의 예측력을 다음 방법에 의해 계산하여 각 모형을 비교한다. Mean Square Prediction Error 예측오차 평균자승합

$$MSPR = \frac{\sum_i^{n^*} (y_i - \hat{y}_i)^2}{n^*}$$

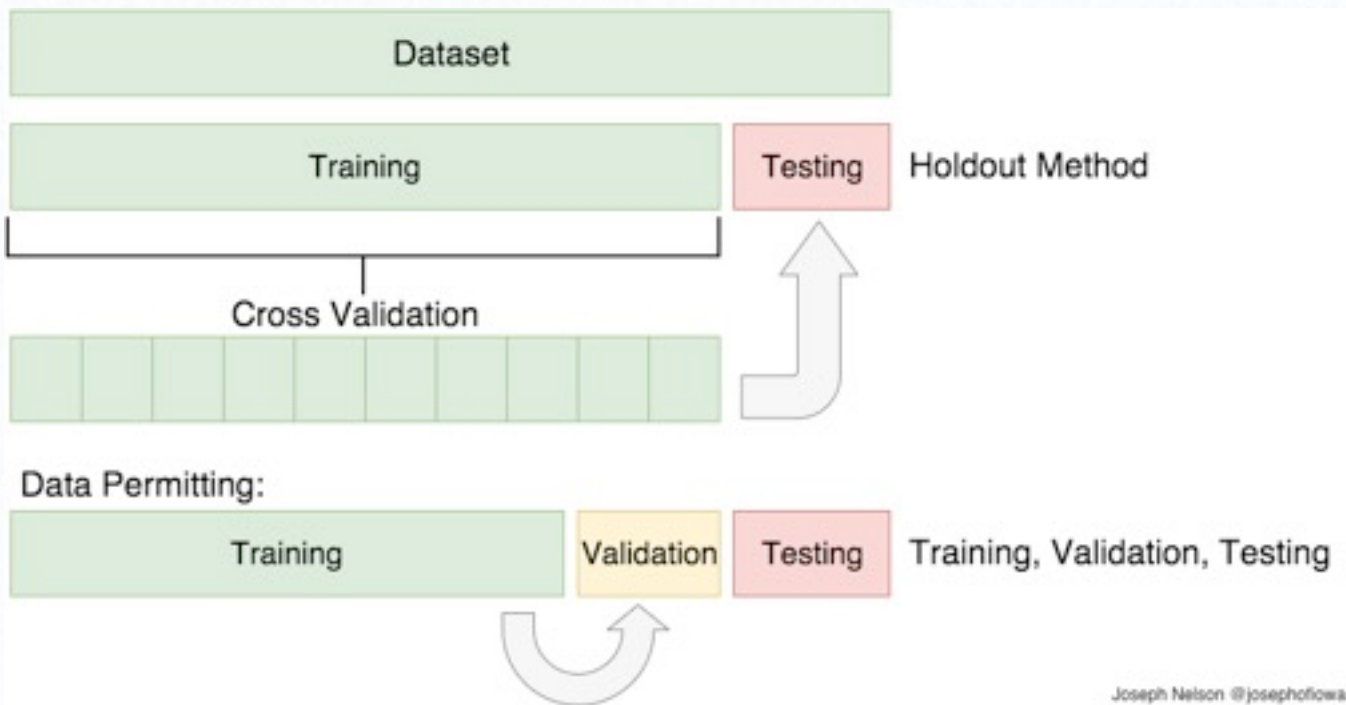
- 새로운 자료로 회귀 모형을 추정하였을 때 이전 자료에서 추정된 회귀 모형과 유사하면 추정된 회귀 모형은 좋다고 판단할 수 있다. 새로운 자료 수집이 불가능한 경우에는 데이터를 splitting하여 모형추정 데이터와 예측 데이터로 나누어 분석한다.
- 수정결정계수나 PRESS, AIC, SBC에 의한 최적 회귀모형 선택은 설명변수가 서로 다른 그룹을 비교할 때 사용되는 통계량이다.

빅데이터 접근

train and test split

대용량 빅데이터의 경우에는 데이터 세트를 2분화 하여 모형을 추정하는 데이터(훈련 데이터 train dataset), 추정 모형을 이용하여 예측 정확도를 판단하는데(평가) 활용되는 데이터(검증 test dataset)로 활용한다. 일반적으로 8:2 혹은 7:3으로 나눈다.

이는 예측모형의 과적합과 underfitting 문제를 해결하기 위하여 머신러닝 기법에서 제안된 방법이다. 이 방법은 전혀 새롭지 않은 개념이다. 이전 데이터마이닝에서 데이터를 3단계, train, validation, test 데이터로 나눈 것과 유사하다.

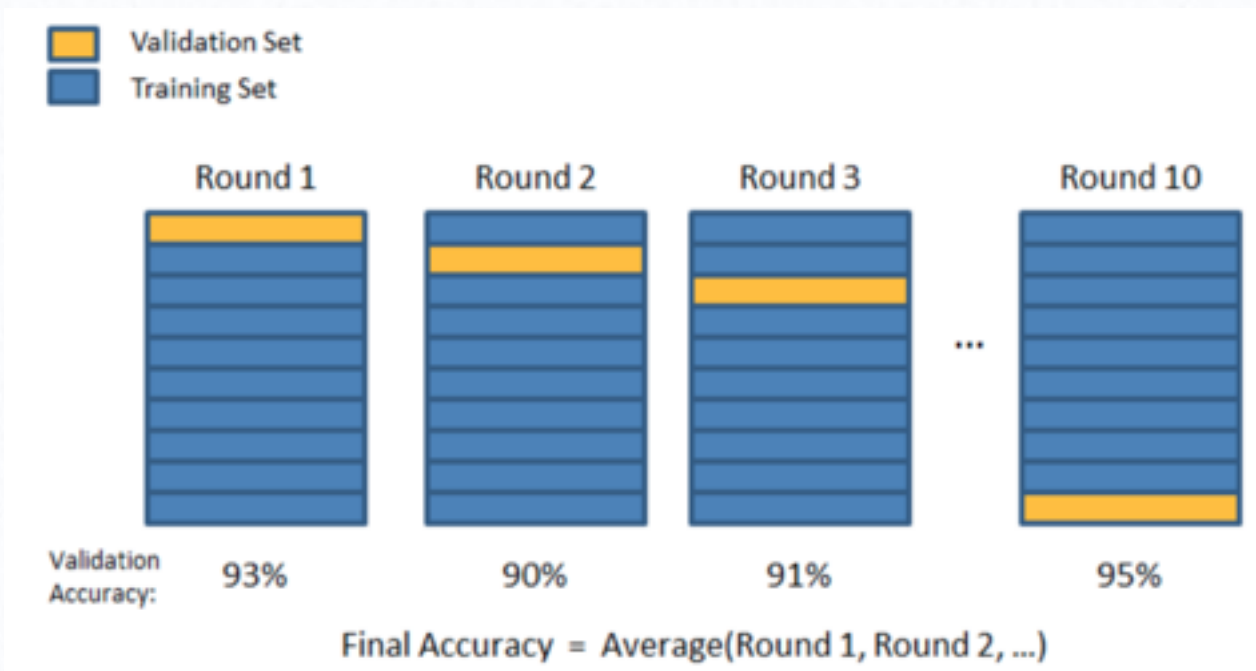


cross validation 필요 이유

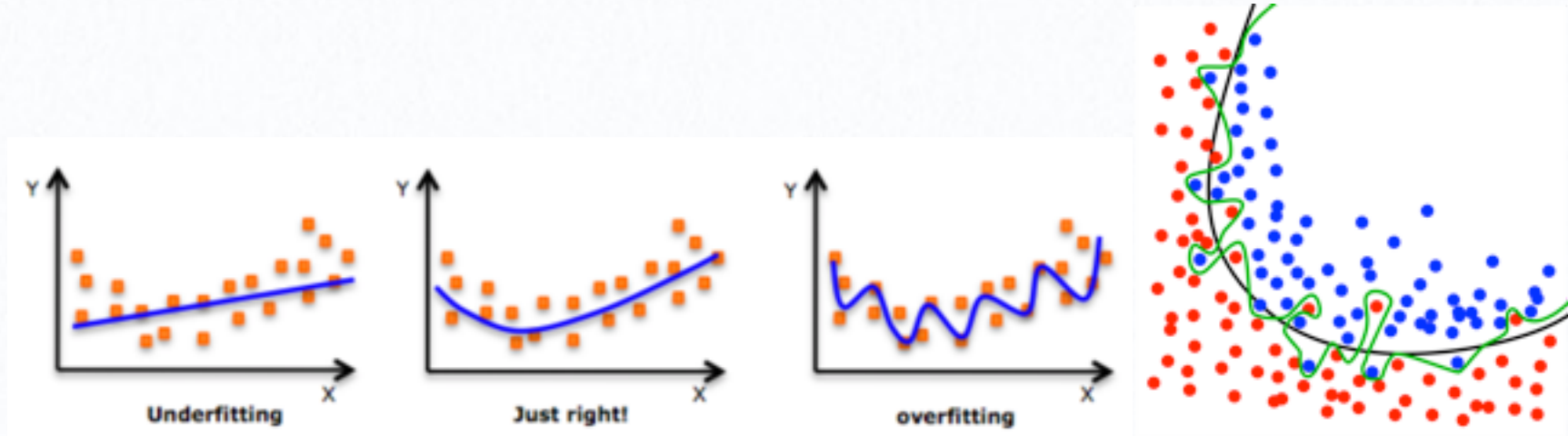
훈련 데이터와 검증 데이터의 스플릿이 무작위가 아닌 경우 문제가 발생한다. 훈련 데이터가 특정 변인에 의해 왜곡되어 있다면 과적합 문제가 발생한다. 예를 들어 우연히 소득수준이 높은 집단이 훈련 데이터에 많이 포함되어 있다면?

K-Folds Cross Validation

- K-Folds Cross Validation에서 훈련 데이터를 k 개의 다른 부분 집합으로 나눈다. k-1 부분 집합을 사용하여 데이터를 훈련시키고 마지막 부분 집합을 테스트 데이터로 하여 검증하게 된다. 이런 과정을 k번 거쳐 계산된 정확도 평균을 훈련 데이터 정확도로 하게 되며 그런 다음 검증 데이터 세트에 대해 검증한다.



과적합과 과소적합



overfitting

과적합은 우리가 추정한 모형이“너무 잘 훈련”되었고 현재는 훈련 데이터 세트를 가장 잘 설명하고 있다는 것을 의미한다. 위의 오른쪽 두 추정 모형을 보면 과적합으로 인하여 데이터에 범용(향후 검증 데이터에 적용 가능한지? 혹은 일반 상황에도 적용 가능한지?) 적합한 모형인지 이 훈련 데이터에만 적합한지 알 수 없다는 사실이다.

관측치 수에 비해 너무 많은 특징(변수)가 존재하는 경우 발생하며 추정 모형은 훈련 데이터에서 매우 정확하지만 훈련되지 않은 또는 새로운 데이터에서는 정확하지 않을 수 있다.

이 추정모형의 결과를 일반화하고 다른 데이터, 즉 궁극적으로 수행하려는 작업에 대해 추론 할 수 없음을 의미한다. 기본적으로 과적합이 발생하면 모델은 데이터의 변수 간 실제 관계 대신 학습 데이터의 "노이즈"를 학습하거나 설명하게 되므로 이 노이즈가 포함되어 있지 않은 (검증)데이터 세트에는 적용할 수 없게 된다.

underfitting

과적합과 대조적으로, 모델이 과소적합되면 모델이 훈련 데이터에 적합하지 않으므로 데이터의 추세를 놓치게 되며 또한 추정 모형을 새 데이터로 일반화 할 수 없음을 의미한다.

이는 매우 간단한 모형 (충분한 예측 변수 / 독립 변수가 아님)의 결과로 선형이 아닌 데이터에 선형 모델 (예 : 선형 회귀)을 적용 할 때도 발생할 수 있다. 이 모델은 예측 능력이 좋지 않을 것이며 훈련 데이터에서 다른 데이터로 일반화 할 수 없음을 알 수 있다.

L1 and L2 Regularization

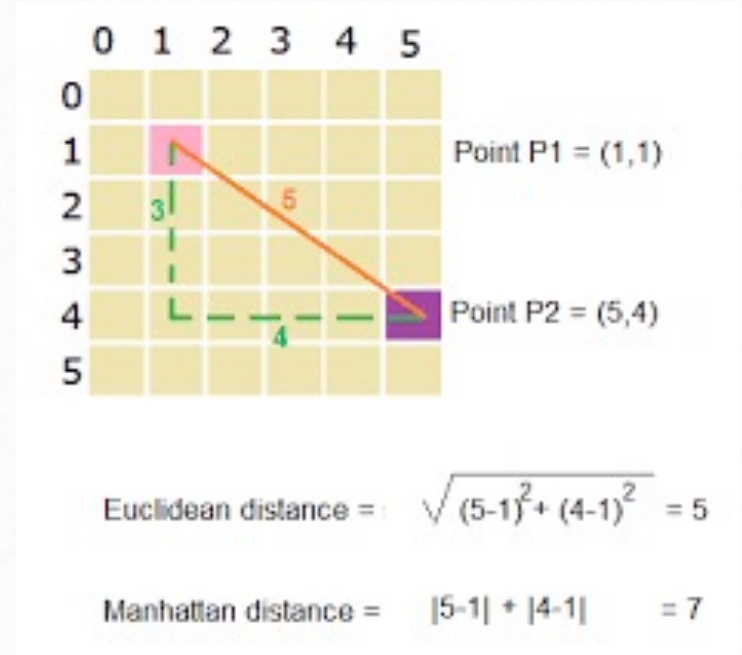
과적합 문제를 해결하기 위한 전처리 작업에 해당된다.

Norm

벡터의 크기 혹은 두 벡터의 거리를 측정하는 함수이다.
 p =놈의 차수를 의미하며 $p=1$ 이면 L_1 놈, $p=2$ 이면 L_2 놈이라 한다.

- L_1 놈 : 맨하탄(Manhattan) 거리이다.
- L_2 놈 : 유클리디안(Euclidean) 거리이다.

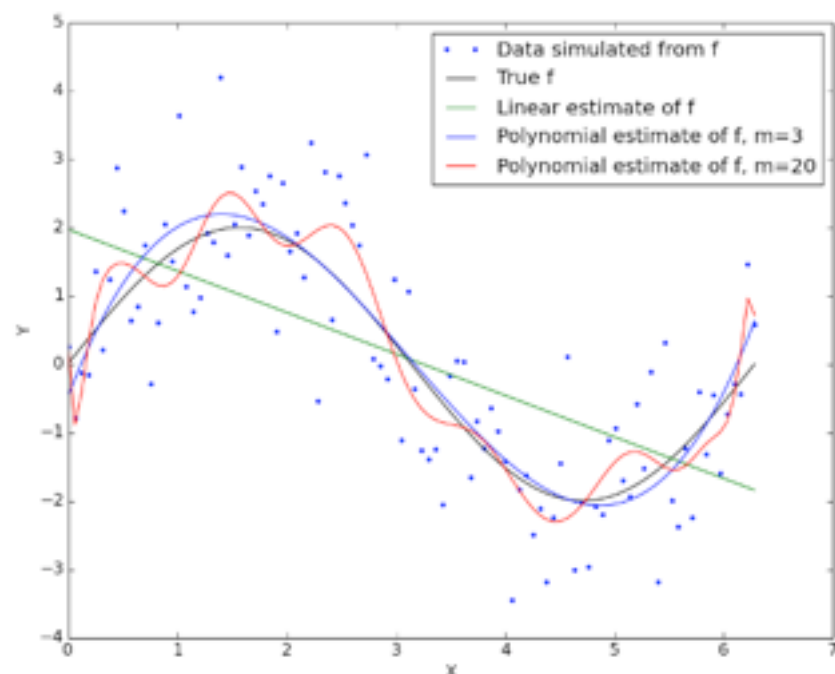
$$||\underline{X}_p|| = \left(\sum_i^n |x_i| \right)^{1/p}$$



Regularization 정규화, 일반화 개념

모형을 추정한다는 것은 관측치에 가장 적합한 모형을 찾는 것인데 이는 원래 함수가 f 임에도(검은 색) 불구하고 과적합 결과는 레드 함수로 추정되어 있다. 왜? 더 많은 정보를 사용하고도 원 함수와 다른 결과가 발생하나? 이는 (원 함수 f 에서 관측된) 관측값(검은 점)에 가장 적합한 함수를 추정하였으므로 이런 현상이 발생한다.

함수를 추정하는 경우에는 관측값과 예측값에 의해 계산되는 손실함수(loss function, cost)를 최소화하는 추정함수를 최적 추정함수라 한다.



Loss function 손실함수 $L(y_i, \hat{y}_i)$

L1 손실함수 : $L(y_i, \hat{y}_i) = \sum |y_i - \hat{y}_i|$

- Least absolute deviations(LAD), Least absolute Errors(LAE)

L2 손실함수 : $L(y_i, \hat{y}_i) = \sum (y_i - \hat{y}_i)^2$

- Least squares error(LSE)

어떤 손실함수를 사용할 것인가? 이상치가 존재하는 경우 이상치에 덜 민감한 (resistant to outliers) L1 손실함수가 적절하나 미분이 가능하고(수학 연산이 용이함) 멀리 떨어진 관측값에 보다 큰 손실을 적용하는 L2 손실함수가 가장 널리 사용된다.

L1 Regularization

Lasso Regression (Least Absolute Shrinkage and Selection Operator)

손실함수 $OLS(L(y_i, \hat{y}_i)) + \lambda \sum |\beta_j|$

- $\lambda=0$ 이면 OLS 추정과 동일하고 λ 가 커지면 회귀계수 영향이 작아져 과소적합이 된다.
- 목표변수에 영향도가 적은 예측변수의 회귀계수를 크기를 최소화 하고 영향도가 큰 예측변수의 회귀계수 크기를 상대적으로 크게 하여 주요 예측변수 'feature selection'에 사용된다.

L2 Regularization

Lasso Regression (Least Absolute Shrinkage and Selection Operator)

손실함수 $OLS(L(y_i, \hat{y}_i)) + \lambda \sum |\beta_j|^2$

- $\lambda=0$ 이면 OLS 추정과 동일하고 λ 가 커지면 불편성이 발생하고 MSE를 최소화 한다.
- 다중공선성 문제를 해결하는 정규화 방법이다.

Features Engineering

개요

① Feature 선택

Feature 순위 또는 Feature 중요도 부여하는 과정으로 중요 변수(주요 신호)의 부분집합을 선택하거나, 불필요한(중요하지 않은) 변수들을 제거하여 변수의 차원(개수)을 줄이는데 그 목적이 있다. 분류모델 중 Decision Tree 같은 경우는 트리의 상단에 있을수록 중요도가 높으므로 이를 반영하여 특징 별로 중요도를 매길 수 있다. 회귀모델의 경우에는 유의 변수선택 알고리즘을 통해 변수 선택이 가능하다.

Feature 선택기법에는 상관분석과 모델기반 방법인 Lasso Regression, Recursive Feature Elimination, Tree-based Model, (Logistics) Discriminant Analysis 등이 있다

② Feature 추출

원데이터 변수들의 구조적 관계를 활용하여 새로운 변수를 생성하거나(고차원의 공간을 저차원의 새로운 공간으로 차원축소) 변수들 간의 상관계수를 변수 유사성 척도로 하여 Feature를 탐색한다. PCA, SVD 차원축소 방법도 여기에 해당하며, 상관관계 유의한 변수군을 만드는 Canonical Correlation Analysis(CCA) 등이 Feature 추출 과정의 방법론이다.

LASSO 방법

데이터 값을 축소(shrinkage) - 중심점 평균으로 중심화 하여($L1$ 정규화 $OLS(L(y_i, \hat{y}_i)) + \lambda \sum |\beta_j|$) 목표변수에 영향을 많이 미치는(목표변수 변동을 보다 잘 설명하는) 예측변수를 선택하거나 다중공선성 문제 진단하는데도 사용된다.

손실함수 $OLS(L(y_i, \hat{y}_i)) + \lambda \sum |\beta_j|$ 을 최소화 하는 것은 $\sum |\beta_j| \leq s$ 제약 조건 하에서 OLS의 SSE를 최소화 하는 추정치를 구하는 것과 동일하다.

λ 결정

- $\lambda = 0$: OLS 추정과 동일, 모든 예측변수가 추정된다.
- $\lambda = \infty$: 모든 예측변수의 회귀계수가 0이 되어 모든 예측변수 제거된다.
- λ 가 증가하면 추정 편이는 증가하고 제거되는 예측변수의 개수는 많아진다.
- 최적 람다는 AIC, BIC를 최소화 하는 람다를 구한다.

Minimizing Information Criteria

$$AIC_{ridge} = n \ln(\underline{e}' \underline{e}) + 2df_{ridge}, BIC_{ridge} = n \ln(\underline{e}' \underline{e}) + 2df_{ridge} \ln(n)$$

$df_{ridge} = \text{tr}(X'X + \lambda I)$ 의 자유도, OLS의 경우 $df_{OLS} = \text{tr}(X'X) = p$ 이다.

모형 개선 방안[파이썬 코드 활용]

Univariate Selection

목표변수의 변동에 대한 예측변수의 설명력의 검정 통계량인 F-통계량의 크기로 설명력이 높은 변수를 선택한다.

- 회귀모형의 변수 선택은 f_regression이며 F-통계량에 의해 변수 선택된다. k 옵션은 최대 유의한 변수 개수 설정이며 디폴트는 10이다. 최대 개수와 유의한 변수 개수는 다르다.
- 결과를 보면 유의한 변수 10개까지 (pop_density_log) 이다. 유의확률 5% 이하
- 결과를 보면 상관계수 크기 순이고 상관계수 유의확률과 동일하다.
- 만약 범주형 판별분석의 유의한 예측변수를 선택하는 경우 chi2를 사용하며 y는 반드시 정수이어야 한다. `y = np.array(y).astype(int)`

```
import numpy as np
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_regression
y=smsa_clean['mortality_index']
X=smsa_clean.drop(columns=['mortality_index','southern'])
bestfeatures = SelectKBest(score_func=f_regression, k=12) #default k=10
fit=bestfeatures.fit(X,y)
dfpvalues = pd.DataFrame(fit.pvalues_)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
#concat two dataframes for better visualization
featureScores = pd.concat([dfcolumns,dfscores,dfpvalues],axis=1)
featureScores.columns = ['Specs','Score','pvalues'] #naming the dataframe columns
print(featureScores.nlargest(12,'Score')) #print best features
```

	Specs	Score	pvalues
11	non_white_ratio_sqrt	37.311710	9.553316e-08
4	education	19.835157	3.990515e-05
3	rainfall	13.161437	6.117124e-04
9	S02Pot	12.146227	9.532634e-04
6	person_houshold	8.929179	4.134816e-03
1	july_temp	6.585786	1.293209e-02
5	white_color_ratio	5.208119	2.623373e-02
7	household_income	4.973858	2.968668e-02
13	NOxPot_log	4.842531	3.183295e-02
10	pop_density_log	4.699521	3.436113e-02
8	HCPot	2.016938	1.609993e-01
2	humidity	0.588325	4.462331e-01

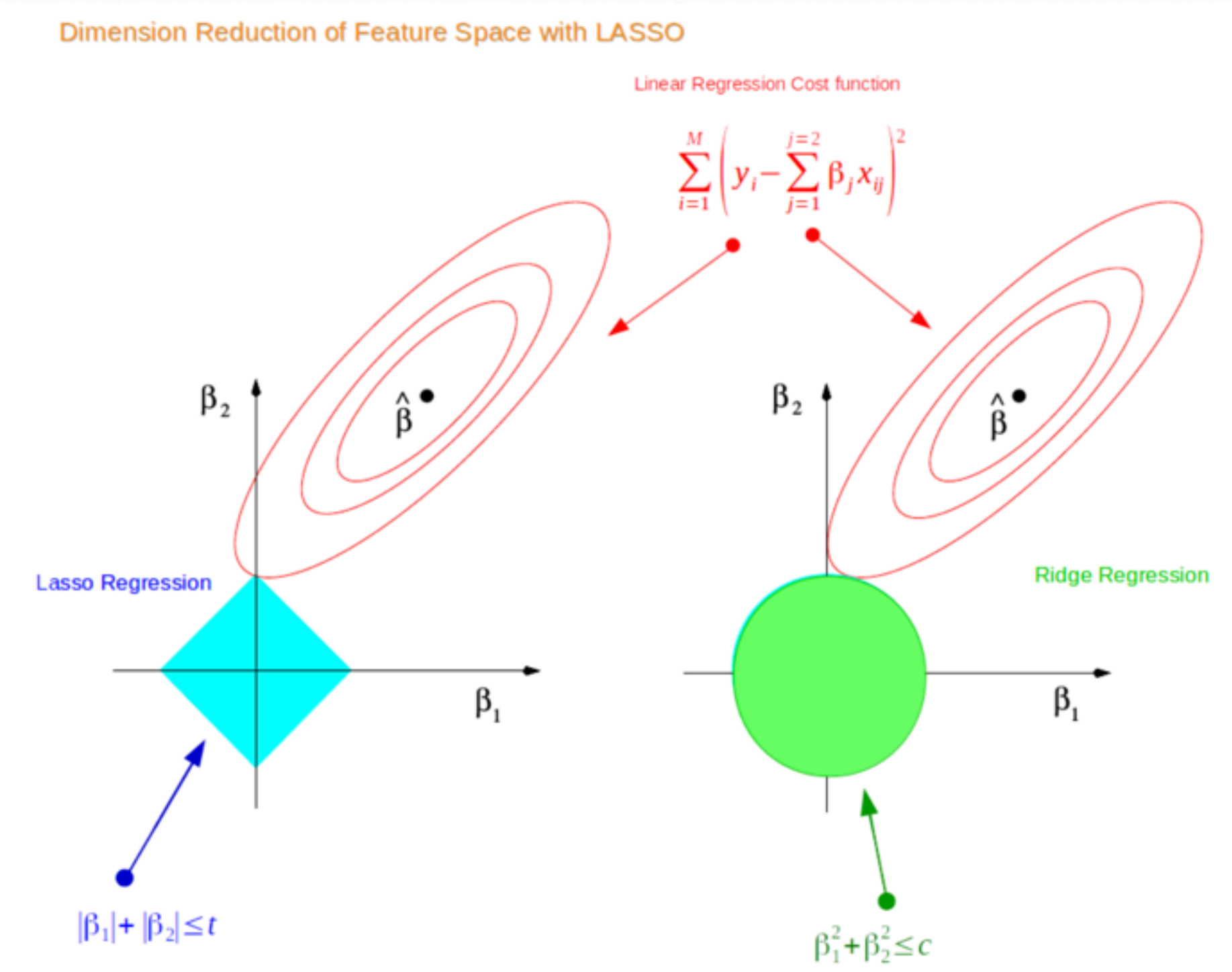
Recursive Feature Elimination

목표변수에 유의한 변수 5개(selector 옵션) 남기고 유의하지 않은 순서대로 제거하였다. 남은 5개 변수는 (습도, 강우량, 인구밀도_로그, 비백인비율_제곱근, NOxPot_로그)이다.

```
y=smsa_clean['mortality_index']
X=smsa_clean.drop(columns=['mortality_index','southern'])
from sklearn.feature_selection import RFE
from sklearn.svm import SVR
estimator = SVR(kernel="linear")
selector = RFE(estimator, 5, step=1)
selector = selector.fit(X, y)
print(X.columns,'\n', selector.support_,'\n',selector.ranking_)
```

```
Index(['jan_temp', 'july_temp', 'humidity', 'rainfall', 'education',
      'white_color_ratio', 'person_houshold', 'household_income', 'HCPot',
      'S02Pot', 'pop_density_log', 'non_white_ratio_sqrt', 'population_log',
      'NOxPot_log'],
      dtype='object')
[False False False  True  True False False False False  True  True
 False  True]
[ 9  7  5  1  1  3  4 10  8  6  1  1  2  1]
```

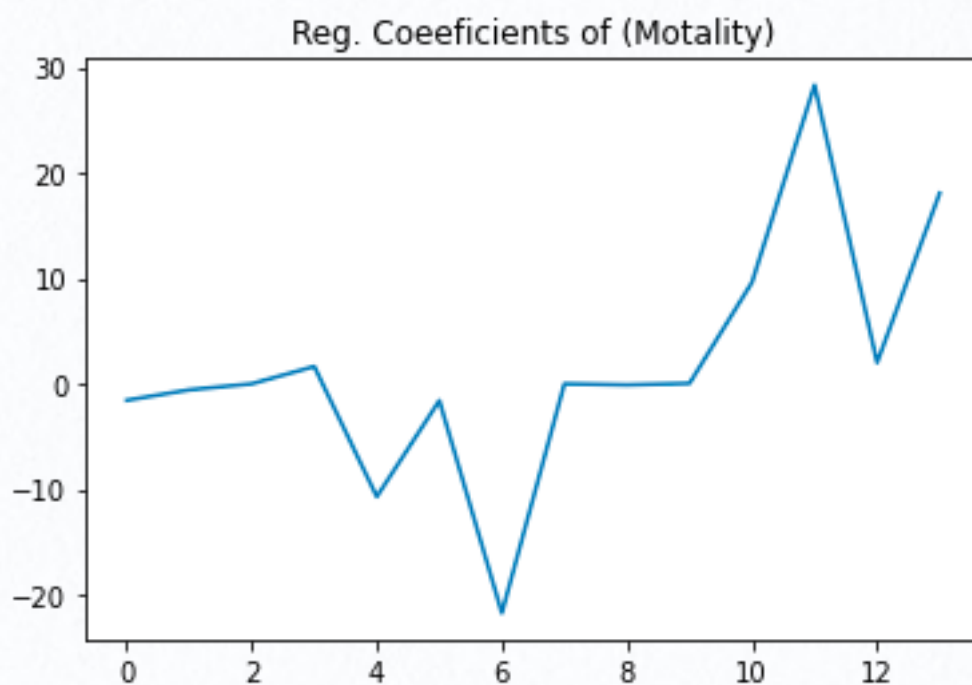

LASSO Regression



- 예측변수의 측정단위가 상이하여 변수를 표준화(normalization) 하였다.

```
from sklearn import linear_model
import matplotlib.pyplot as plt
lassoReg=linear_model.Lasso(alpha=0.1, fit_intercept=True, normalize=True)
y=smsa_clean['mortality_index']
X=smsa_clean.drop(columns=['mortality_index','southern'])
fit=lassoReg.fit(X,y)
plt.title('Reg. Coefficients of (Mortality)')
plt.plot(fit.coef_); plt.show()
```

회귀계수의 크기가 큰(단위를 표준화 하였으므로 회귀계수 크기가 크다는 것은 종속변수에 대한 영향도가 크다는 것을 의미한다.) 5개 예측변수를 선택한 결과 : (비백인비율_제곱근, 가구원수, NOxPot_로그, 교육기간, 인구밀도_로그)



	Var	Coeff
4	education	-10.696803
6	person_houshold	-21.711816
10	pop_density_log	9.619333
11	non_white_ratio_sqrt	28.299255
13	NOxPot_log	18.048101

```
var_nm=pd.Series(X.columns,name='Var')
coeff=pd.Series(fit.coef_,name='Coeff')
res=pd.concat([var_nm,coeff],axis=1)
res[abs(res['Coeff'])>10]
```

변수 선택 요약

후진제거

```
['jan_temp', 'rainfall', 'education', 'non_white_ratio_sqrt', 'NOxPot_log']
```

전진삽입

```
selected variables : ['non_white_ratio_sqrt', 'education', 'jan_temp', 'NOxPot_log']
mortality_index ~ non_white_ratio_sqrt + education + jan_temp + NOxPot_log + rainfa
0.7297682020409422
```

Univariate(상관계수 개별 유의성)

LASSO

	Specs	Score	pvalues		Var	Coeff
11	non_white_ratio_sqrt	37.311710	9.553316e-08			
4	education	19.835157	3.990515e-05	4	education	-10.696803
3	rainfall	13.161437	6.117124e-04			
9	S02Pot	12.146227	9.532634e-04			
6	person_houshold	8.929179	4.134816e-03	6	person_houshold	-21.711816
1	july_temp	6.585786	1.293209e-02			
5	white_color_ratio	5.208119	2.623373e-02	10	pop_density_log	9.619333
7	household_income	4.973858	2.968668e-02			
13	NOxPot_log	4.842531	3.183295e-02	11	non_white_ratio_sqrt	28.299255
10	pop_density_log	4.699521	3.436113e-02			
8	HCPot	2.016938	1.609993e-01			
2	humidity	0.588325	4.462331e-01	13	NOxPot_log	18.048101

Recursive 방법

```
(['jan_temp', 'july_temp', 'humidity', 'rainfall', 'education', 'white_color_ratio', 'person_houshold', 'household_income', 'HCPot', 'S02Pot', 'pop_density_log', 'non_white_ratio_sqrt', 'population_log', 'NOxPot_log'],
```

[최종 선택변수]

- 1월기온, 강우량, 교육기간, 비백인비율, 세금, NOxPot 로그, 가구원수, 인구밀도 로그
- SO2Pot, 가구원수, 7월소득, 사무직비율, 가구소득