Feature Engineering

(+ Variable Selection)

21.02.24

Seunghan Lee

Feature Engineering ?

"분석가의 분석 목적에 맞게 데이터를 처리/가공하는 과정"

대표적으로...

1) 범주형 변수(Categorical Features)의 처리

2) 상호작용과 다항식

3) 비선형 변환

4) 변수 선택 (Variable(Feature) Selection)

+ Ridge & Lasso

+ Cross Valdation

Contents

1) 범주형 변수(Categorical Features)의 처리

(1) One-hot Encoding이란?

(before) K개의 종류를 가진 범주형 변수 1개

(after) 1/0값을 가진 더미 변수 K(혹은 K-1)개

```
pd.get_dummies(df['animal'])
```

	animal	age
0	cat	3
1	dog	4
2	dog	2
3	cat	4
4	snake	5

	age	cat	dog	snake
0	3	1	0	0
1	4	0	1	0
2	2	0	1	0
3	4	1	0	0
4	5	0	0	1

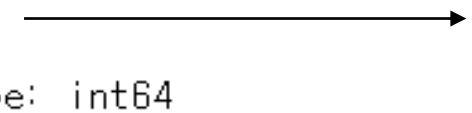
1) 범주형 변수(Categorical Features)의 처리

(2) 등장횟수 적은 값 묶기

```
df['animal'].value_counts()
```

executed in 21ms, finished 15:40:30 2021-02-18

```
cat        200
dog        200
dolphin      2
snake        2
bear         2
Name: animal, dtype: int64
```



몇 마리 안되는 소수의 동물 종들 때문에,

추가로 변수들이 더 생겨야 하는 상황! **그룹화(묶기)**

```
df['animal'] = df['animal'].replace(['dolphin','snake','bear'], value='etc')
df['animal'].value_counts()
```

executed in 10ms, finished 15:44:27 2021-02-18

```
cat    200
dog    200
etc      6
Name: animal, dtype: int64
```

2) 상호작용과 다항식

(1) 상호 작용 (Interaction term)

$$\hat{y} = b_0 + b_1X_1 + b_2X_2 + \boxed{b_3X_1X_2}$$

X1도 Y에 영향 O

X2도 Y에 영향 O

그 둘과 별개로, **X1과 X2가 서로 상호작용해서 Y에 영향을 미칠 수 있다**!

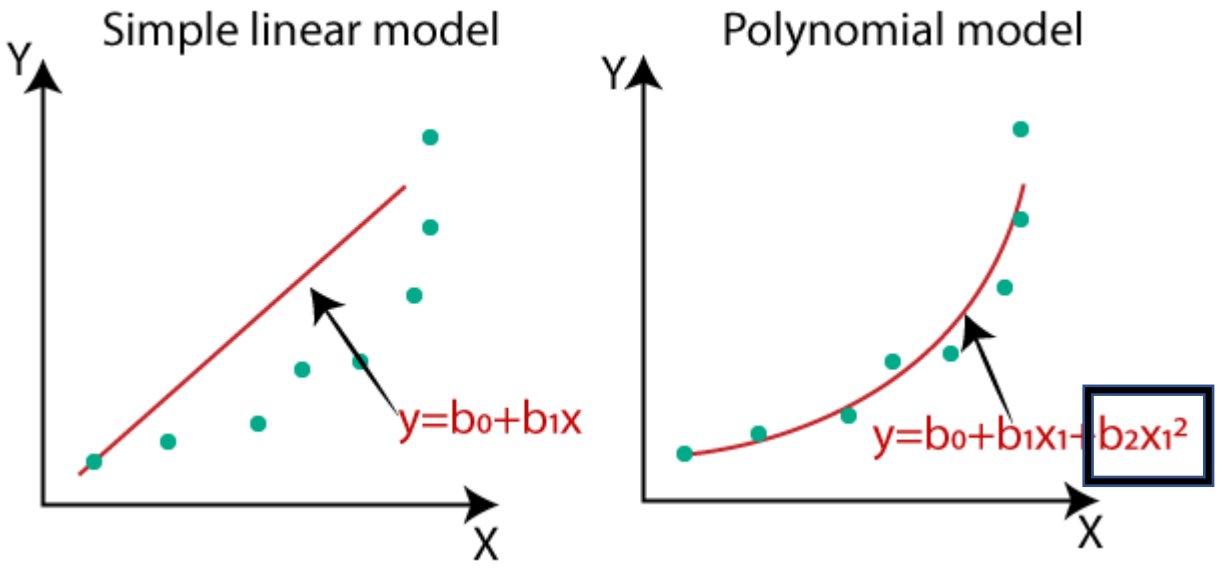
Ex) X1 : 흡연 여부 & X2 : 음주량, Y : 기대 수명

흡연만 했을 때의 해로움과, 음주만 했을 때의 해로움 외에도,

흡연과 음주를 "같이 " 했을 경우에 부가적인 해로움이 있을 수 있다!

2) 상호작용과 다항식

(2) 다항식 (Polynomial term)



데이터의

“큰 " 구조는 저차원이,

“세밀한” 구조는 고차원이 잡아낼 수 있다!

2) 상호작용과 다항식

(2) 다항식 (Polynomial term)

"1.상호작용" & "2.다항식" 추가하기

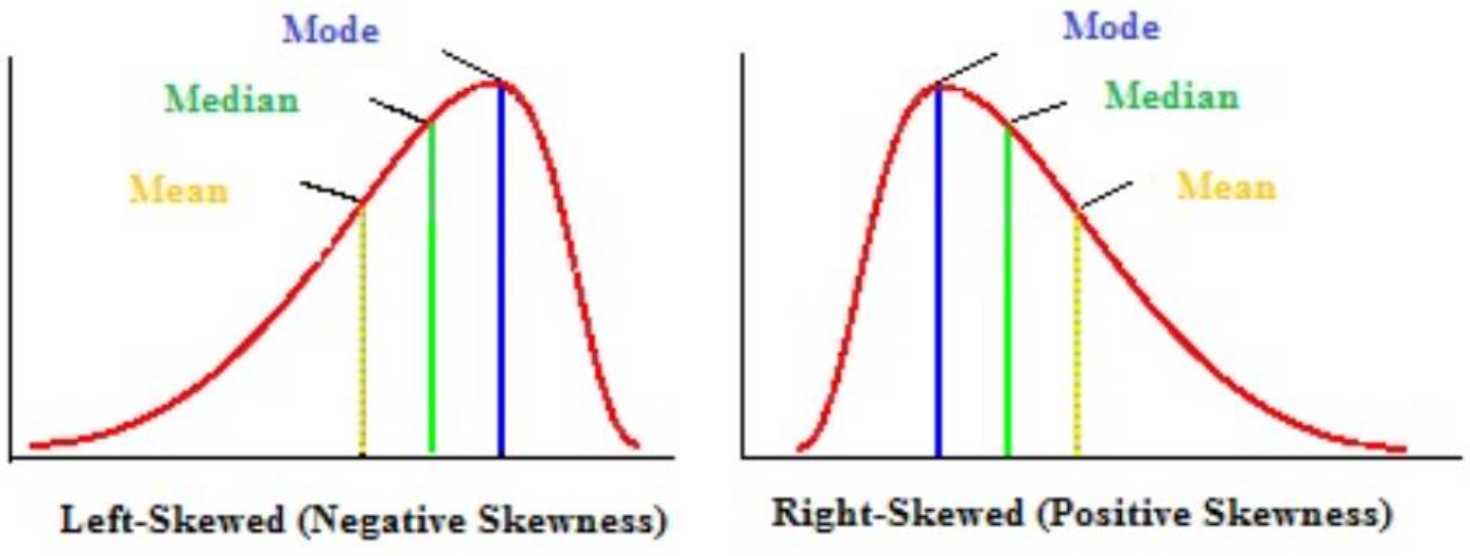
```
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(2)
X_poly = poly.fit_transform(X)
```

	X1	X2
0	0	1
1	2	3
2	4	5

	1	x0	x1	x0^2	x0 x1	x1^2
0	1.0	0.0	1.0	0.0	0.0	1.0
1	1.0	2.0	3.0	4.0	6.0	9.0
2	1.0	4.0	5.0	16.0	20.0	25.0

3) 비선형 변환

Skewed Data



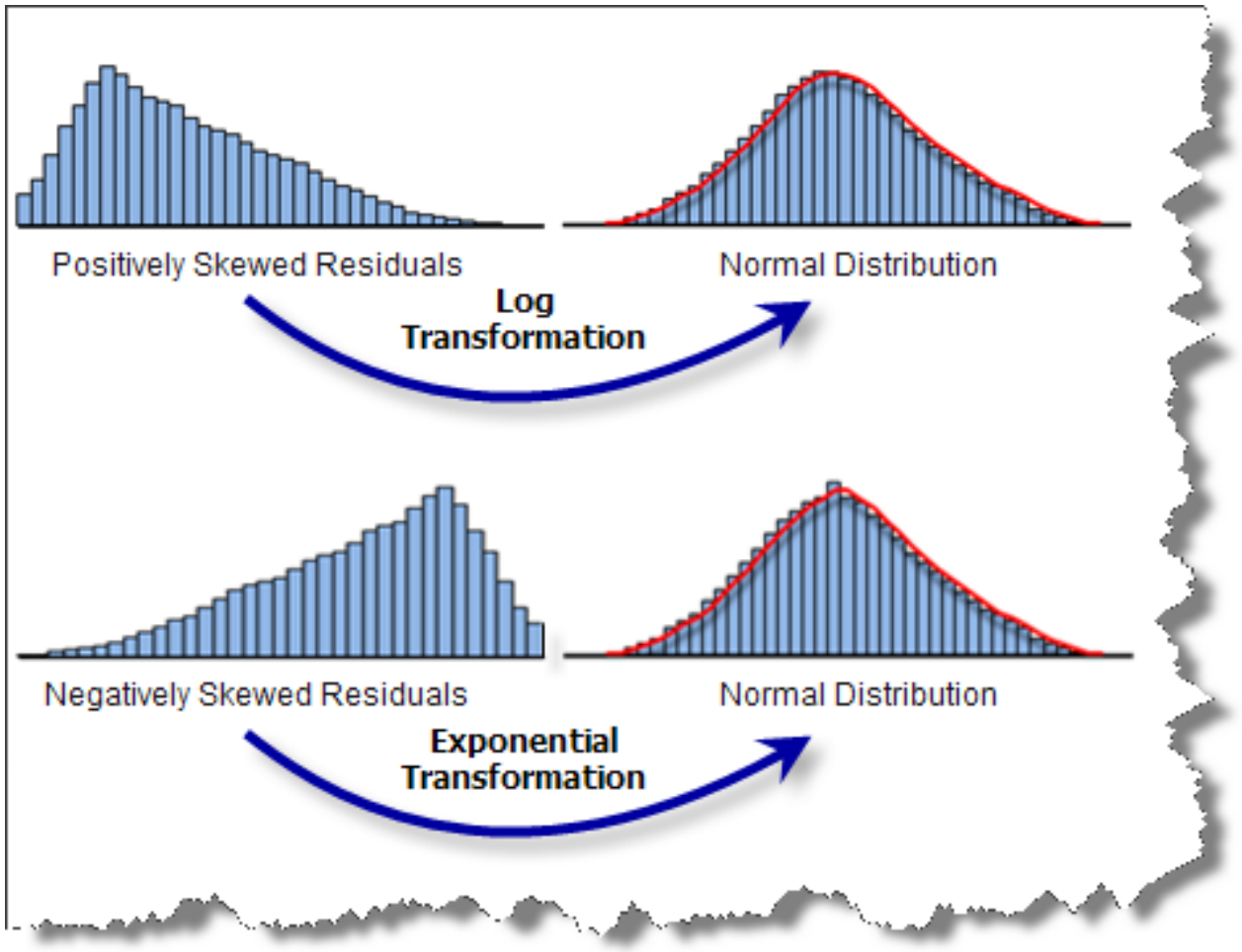
3) 비선형 변환

right skewed 되어 있을 때 :

Log Transformation

left skewed 되어 있을 때 :

Exponential Transformation



4) 변수 선택

빅데이터 시대

"데이터가 많다" = not only ROW but also **COLUMN(features)**

수 많은 변수가 존재하는 상황에서,

어떻게 해당 변수(features)를 선택/축소 해야하나?

↓

Dimension reduction (차원 축소)… ex) PCA

4) 변수 선택

크게 세 가지 방법

1) 일변량 통계 (Univariate Statistics)

2) 모델 기반 선택 (Model-based Selection)

3) 반복적 선택 (Iterative Selection)

4) 변수 선택

(1) 일변량 통계

Key Point :

X1~Xp의 독립변수와, Y 종속변수의 **"통계적 관계 " 를 파악해서, 관계가 높은 것**을 고름

대표적인 기준 : F-score

(Classification의 경우)

Source of Variation	df	Sum of Squares (SS)	Mean Sum of Squares (MSS)	F-test	p-value
Treatment	k-1	SSTr	MSTr=SSTr/(k-1)	F=MSTr/MSE	
Error	N-k	SSE	MSE=SSE/(N-k)		
Total	N-1	SSTo			

(Regression의 경우)

$$\frac{\left(\frac{\mathrm{Corr}^2}{1-\mathrm{Corr}^2}\right)}{df}$$

4) 변수 선택

(1) 일변량 통계

```
select = SelectPercentile(percentile=50,score_func=f_classif)
select.fit(X_train,y_train)
```

executed in 7ms, finished 16:47:51 2021-02-18

```
SelectPercentile(percentile=50)
```

```
select2 = SelectKBest(k=5,score_func=f_classif)
select2.fit(X_train,y_train)
```

executed in 14ms, finished 16:48:35 2021-02-18

```
SelectKBest(k=5)
```

Classification

P개의 변수 중, F-score가 높은

상위 n%의 변수만을 사용!

P개의 변수 중, F-score가 높은

상위 n개의 변수만을 사용!

4) 변수 선택

(1) 일변량 통계

```
select = SelectPercentile(percentile=50,score_func=f_regression)
select.fit(X_train,y_train)
```

executed in 7ms, finished 16:47:51 2021-02-18

```
SelectPercentile(percentile=50)
```

```
select2 = SelectKBest(k=5,score_func=f_regression)
select2.fit(X_train,y_train)
```

executed in 14ms, finished 16:48:35 2021-02-18

```
SelectKBest(k=5)
```

Regression

P개의 변수 중, F-score가 높은

상위 n%의 변수만을 사용!

P개의 변수 중, F-score가 높은

상위 n개의 변수만을 사용!

4) 변수 선택

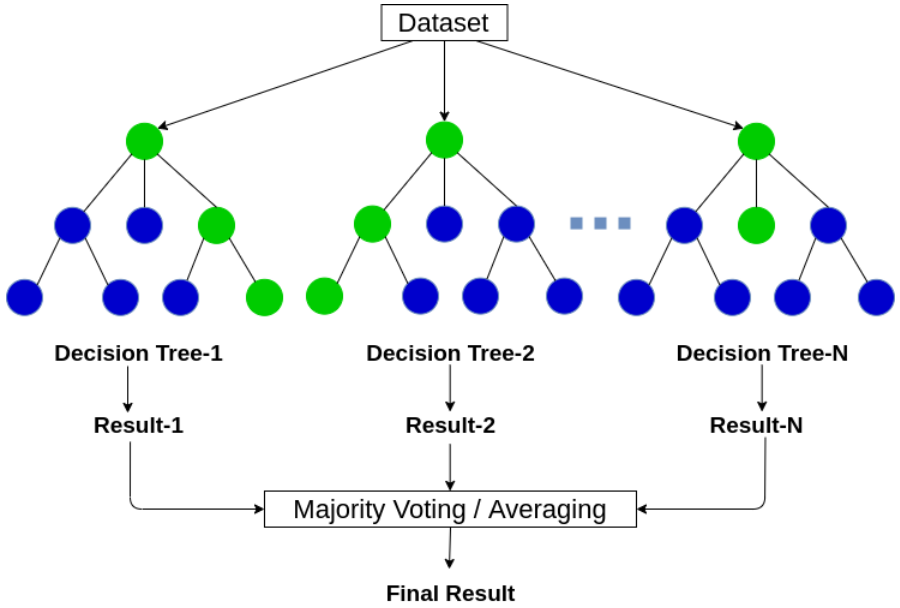
(2) 모델 기반 선택

모델을 사용해서 **특성의 중요도(feature importance)**를 파악

즉, 해당 모델이 좋은 결과값을 내기 위해,

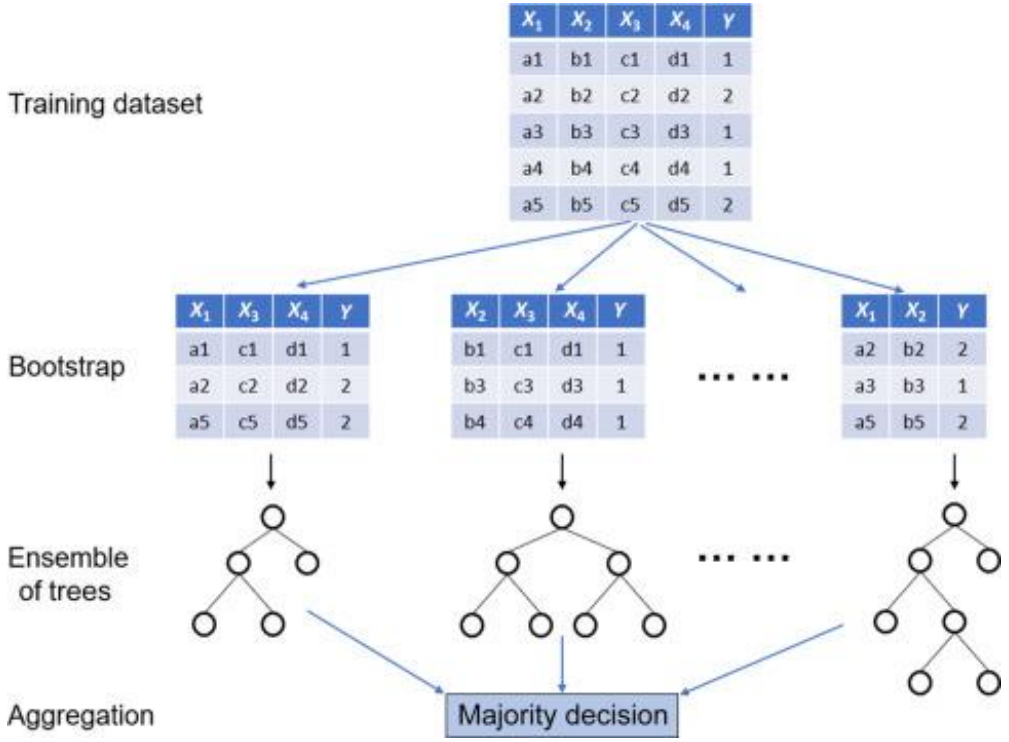
가장 큰 기여를 한 변수를 뽑는다!

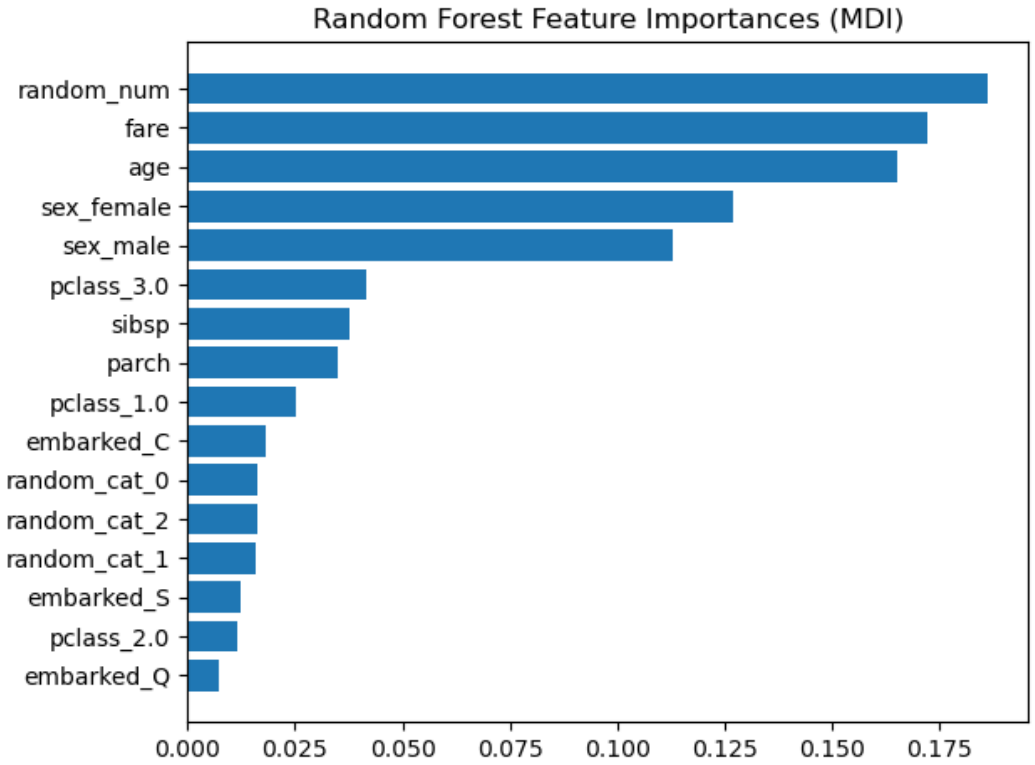
Example) Random Forest



4) 변수 선택

(2) 모델 기반 선택





4) 변수 선택

(2) 모델 기반 선택

threshold : *string or float, default=None*

The threshold value to use for feature selection. Features whose importance is greater or equal are kept while the others are discarded. If "median" (resp. "mean"), then the `threshold` value is the median (resp. the mean) of the feature importances. A scaling factor (e.g., "1.25*mean") may also be used. If None and if the estimator has a parameter penalty set to l1, either explicitly or implicitly (e.g, Lasso), the threshold used is 1e-5. Otherwise, "mean" is used by default.

```
select = SelectFromModel(RandomForestClassifier(n_estimators=100,random_state=42),
                         threshold='median')
```

executed in 17ms, finished 16:58:43 2021-02-18

```
select.fit(X_train,y_train)
```

executed in 183ms, finished 16:58:55 2021-02-18

```
SelectFromModel(estimator=RandomForestClassifier(random_state=42),
                threshold='median')
```

```
X_train_l1 = select.transform(X_train)
```

executed in 30ms, finished 16:59:08 2021-02-18

```
X_train_l1.shape
```

executed in 20ms, finished 16:59:14 2021-02-18

```
(455, 40)
```

4) 변수 선택

3) 반복적 특성 선택

1) 일변량 분석 : 모델 사용 X
2) 모델 기반 선택 : 모델 1개
3) **반복적 특성 선택 : 각기 다른 여러 개의 모델**

크게 2가지의 방법

1) feature를 하나도 선택하지 않은 상태에서 시작
   -> 특정 종료 조건 도달될 때까지 하나씩 추가
2) 모든 feature를 가지고 시작
   -> 특정 종료 조건 도달될 때까지 하나씩 제거

4) 변수 선택

3) 반복적 특성 선택

Recursive Feature Elimination

least important features are **pruned(=eliminated)** from current set of features

```
select = RFE(RandomForestClassifier(n_estimators=100,random_state=42),
             n_features_to_select=40)
select.fit(X_train,y_train)
```

executed in 11ms, finished 17:06:49 2021-02-18

```
mask = select.get_support()
```

executed in 10ms, finished 17:07:06 2021-02-18

```
mask
```

executed in 12ms, finished 17:07:08 2021-02-18

```
array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True, False,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True, False, False, False, False,  True,  True,
       False, False,  True, False,  True, False, False, False,  True,
       False,  True, False, False, False, False, False,  True, False,
        True, False,  True, False,  True, False, False, False, False,
       False, False,  True, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False])
```

4) 변수 선택

3) 반복적 특성 선택

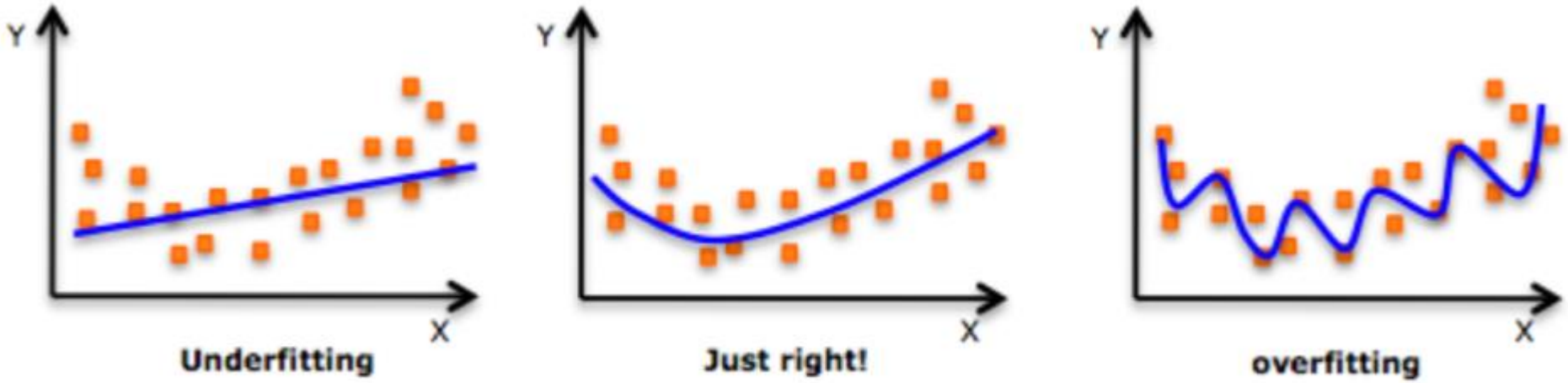
```
plt.matshow(mask.reshape(1,-1),cmap='gray_r')
plt.xlabel('Used Target')
```

executed in 77ms, finished 17:08:28 2021-02-18

```
Text(0.5, 0, 'Used Target')
```

	선택된 변수
	선택되지 않은 변수

Ridge & Lasso



너무 많은 변수 사용으로 인해 “overfitting(과적합) " 발생

변수를 줄이는(선택하는) 대표적인 방법 : **Lasso !**

Ridge & Lasso

일반적인 선형 회귀
(Linear Regression)

$$Cost\ (W) = RSS(W) = \sum_{i=1}^{N} \{y_i - \hat{y}_i\}^2 = \sum_{i=1}^{N} \left\{ y_i - \sum_{j=0}^{M} w_j\, x_{ij} \right\}^2$$

Ridge

$$Cost\ (W) = RSS(W) + \lambda * (sum\ of\ squares\ of\ weights)$$

$$= \sum_{i=1}^{N} \left\{ y_i - \sum_{j=0}^{M} w_j\, x_{ij} \right\}^2 + \lambda \sum_{j=0}^{M} w_j^2$$

Lasso

$$Cost\ (W) = RSS(W) + \lambda * (sum\ of\ absolute\ value\ of\ weights)$$

$$= \sum_{i=1}^{N} \left\{ y_i - \sum_{j=0}^{M} w_j\, x_{ij} \right\}^2 + \lambda \sum_{j=0}^{M} |w_j|$$

Ridge & Lasso

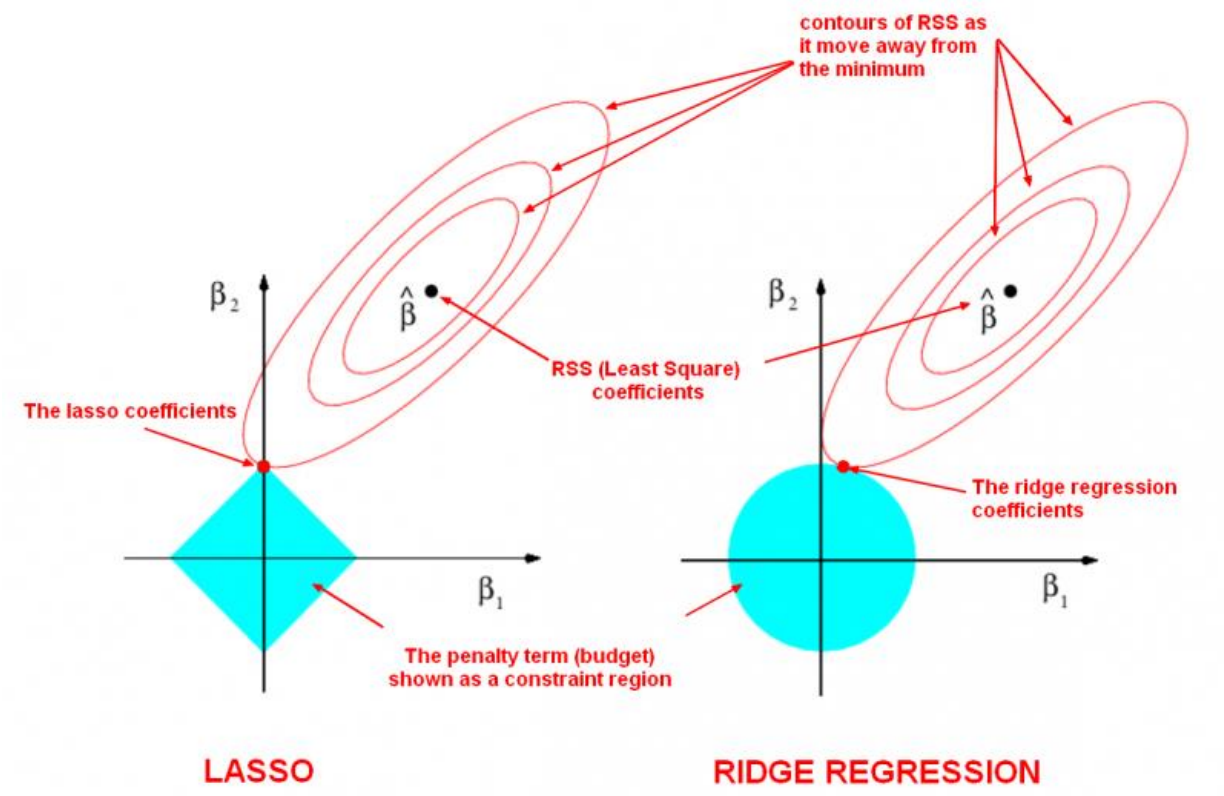
Ridge와 Lasso 모두 특정 feature의 weight(parameter)가 너무 커지는 것을 제어한다!

차이점 :

- Ridge의 경우에는 그 weight가 0이 될 수 없지만,
- **Lasso의 경우에는 그 weight가 0이 될 수 있다**

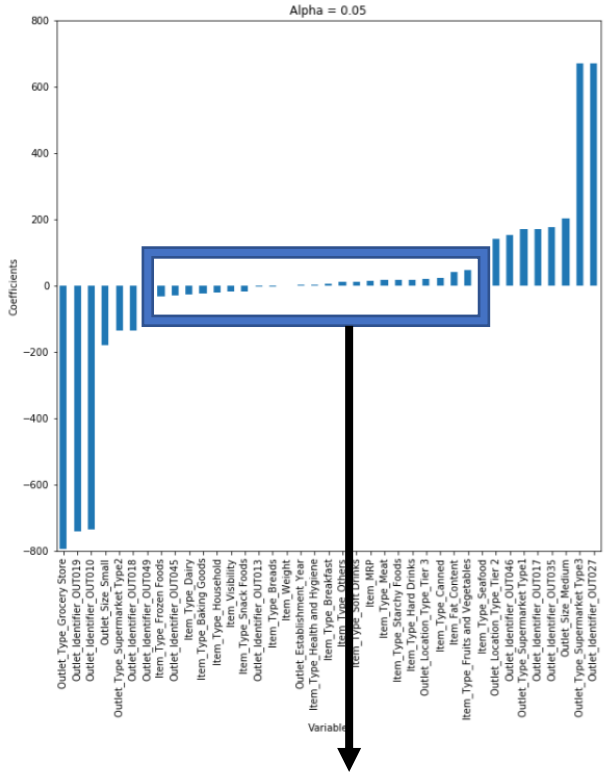
따라서, Lasso는 feature selection을 위해서 종종 사용된다!

Ridge & Lasso


contours of RSS as
it move away from
the minimum
β_2
$\hat{\beta}$
RSS (Least Square)
coefficients
The lasso coefficients
β_1
The penalty term (budget)
shown as a constraint region
LASSO
β_2
$\hat{\beta}$
The ridge regression
coefficients
β_1
RIDGE REGRESSION

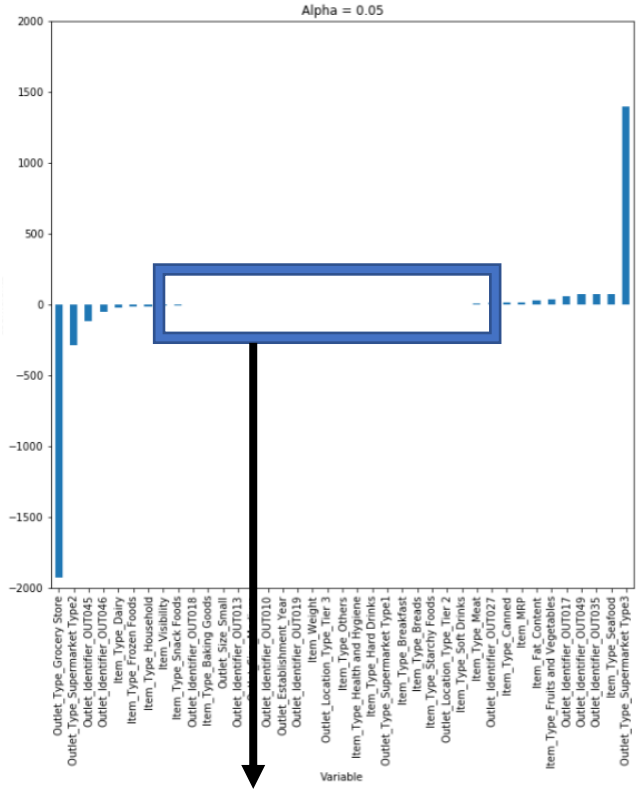
Ridge & Lasso

Ridge



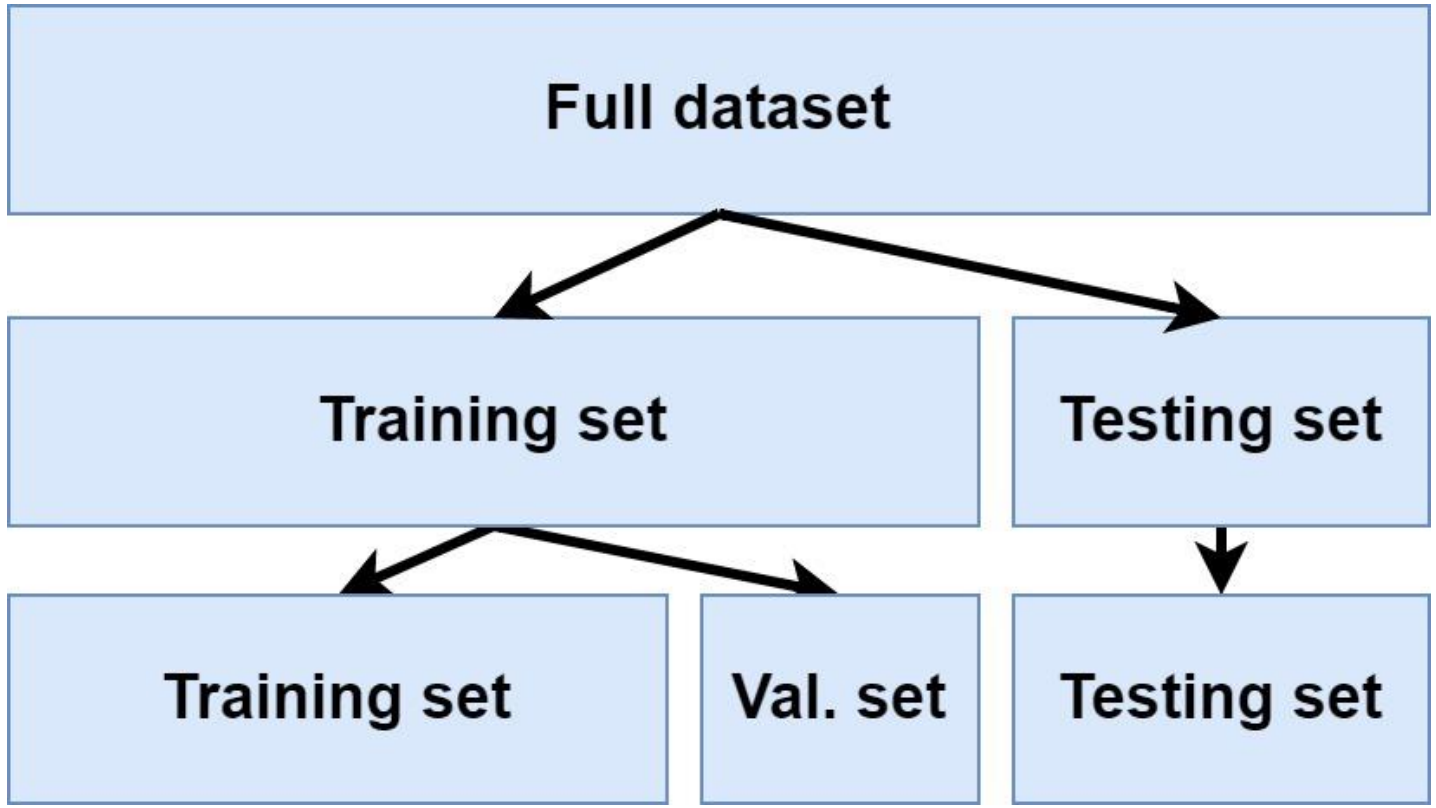
특정 변수의 weight가 매우 작아질 수는 있어도, 0이 될 수는 없다

Lasso



특정 변수의 weight가 0이 될 수 있다

Cross Validation



Cross Validation

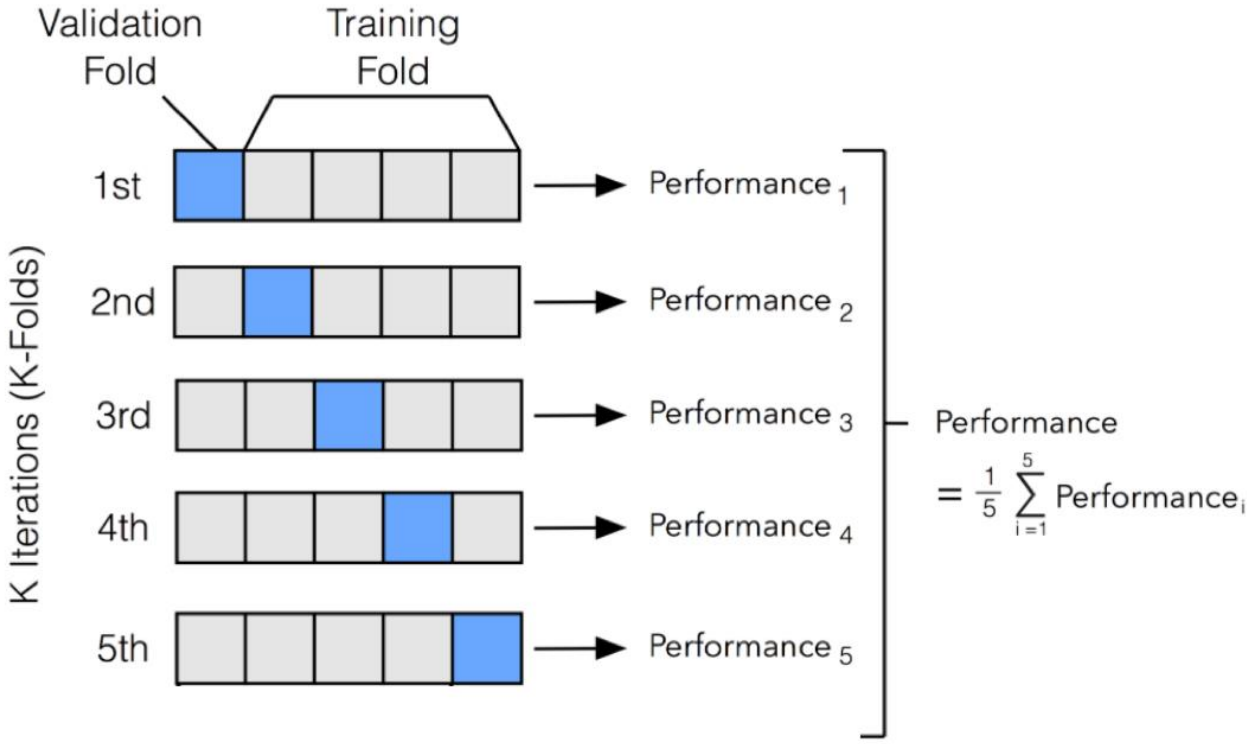
Ex) 전체의 X1~X30의 변수 중, 8개만 골라서 사용했다.

그렇다면, 이 **“8개 만을 사용 "** 한 모델이,

“30개를 전부 사용 " 한 모델보다 낫다는 것을 어떻게 검증(validate)하지?

use **Cross Validation!**

Cross Validation



K-fold CV

(1) 전체 데이터를 K 등분

(2) For k in (1:K):

k번째 부분을 validation data로

나머지 k-1개의 부분을 train data로 모델 학습

validation data로 에러(혹은 정확도) 계산

(3) 위에서 산출된 K개의 에러(혹은 정확도) 평균

Thank You