3/28/2019

SESSION REGRESSION

류승우

Growth Hackers

CONTENTS

0. Preview
1. 단순 회귀(Simple)
2. 다중 회귀(Multiple)
3. 릿지 회귀(Ridge)
4. 라쏘 회귀(Lasso)
5. 로지스틱 회귀(Logistic)

0. Preview

- (1) carat과 price, 두 개의 변수만을 가지는 데이터프레임을 만들어주세요. 또, 이 데이터를 바탕으로 diamond의 가격을 예측하는 단순선형회귀모델을 만들어 주세요.

- (2) (1)에서 도출한 회귀식이 단순선형회귀모델의 조건들을 만족하는지 그래프를 그려 판단해 주시고 이를 바탕으로 모델을 평가해주세요.

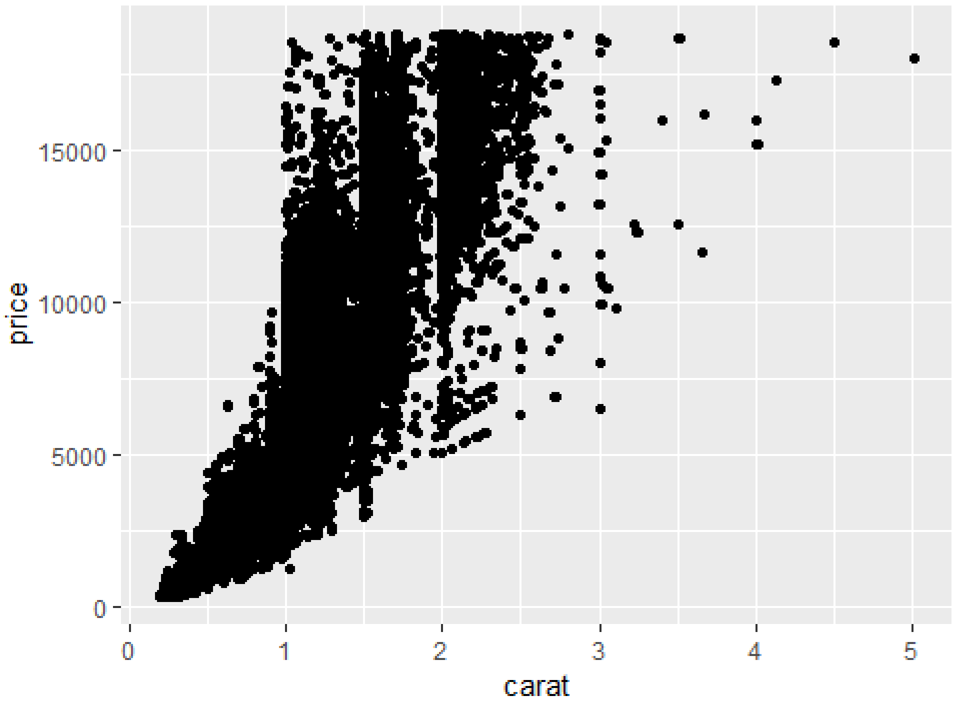
- (3) (1)에서 도출한 모델 이외에, diamond 데이터 내의 다양한 변수를 활용하여 본인이 생각하기에 가장 좋은 예측 선형회귀모델을 도출해주세요.

0. Preview

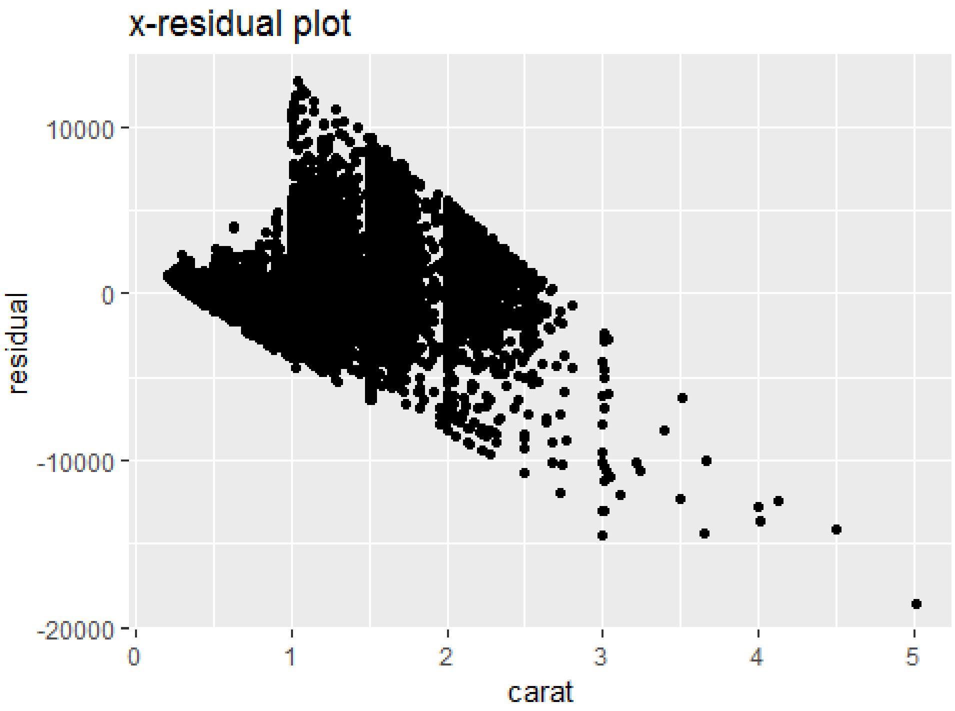
- (2) (1)에서 도출한 회귀식이 단순선형회귀모델의 조건들을 만족하는지 그래프를 그려 판단해 주시고 이를 바탕으로 모델을 평가해주세요.

- 단순선형회귀분석의 4가지 조건
  - 선형성
  - 등분산성
  - 독립성
  - 정규성

0. Preview



선형성 X



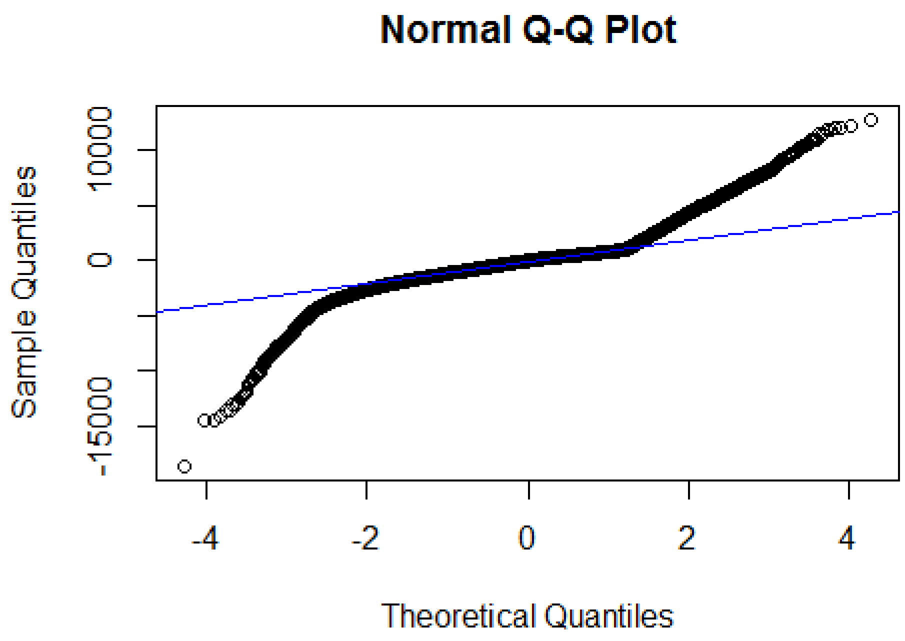
등분산성 X

0. Preview

```
        Durbin-Watson test

data:  lm_model1
DW = 0.98603, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

독립성 X



정규성 X

0. Preview

- (3) (1)에서 도출한 모델 이외에, diamond 데이터 내의 다양한 변수를 활용하여 본인이 생각하기에 가장 좋은 예측 선형회귀모델을 도출해주세요.

  - 다중공선성 평가
  - 범주형 변수의 더미변수화
  - Variable Selection (ex. ${R_p}^2, {R_a}^2, C_p,\ AIC, BIC, \ldots)$

    └ Forward Selection / Backward Elimination …

Growth Hackers

0. Preview

- 전통적인 통계학 vs Machine Learning
  - 시중의 ML 교재들: 각종 통계학적 가정에 대한 언급x
  - ex. LSE : 4가지 가정
  - 가정을 무시한다면 Variation이 너무 커져버리는 결과

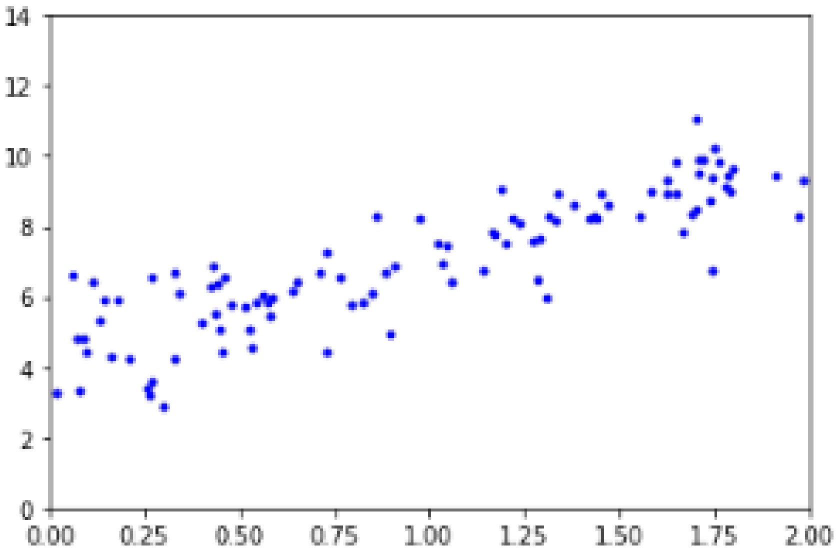
- Big Data Era?
  - 데이터의 양이 방대해지면?
  - 이러한 가정들이 만족되지 않아도 좋은 결과
  - Machine Learning

3/28/2019

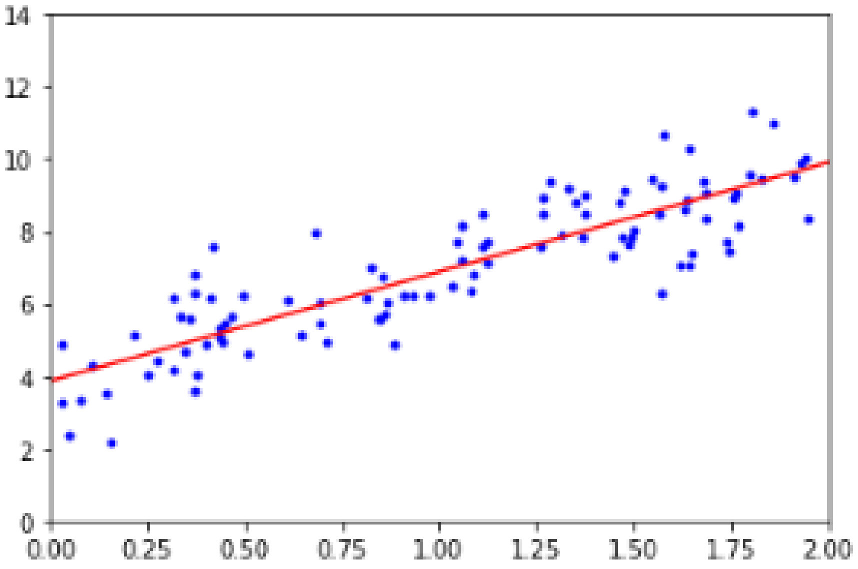
1. 단순선형회귀 (SIMPLE LINEAR REGRESSION)

Growth Hackers

1. 단순회귀



?



$$\hat{y} = \widehat{w_0} + \widehat{w_1}X$$

HOW?

1. 단순회귀

- 1-1. Normal Equation
  - $Y_i = w_0 + w_1 X_i + \varepsilon_i$, assuming $\varepsilon_i \sim N(0, \sigma^2)$
  - '$\frac{1}{n}\sum_{i=1}^{n} \varepsilon_i{}^2 (= MSE)$ 을 최소화 하는 w_0과 w_1의 값을 찾겠다.'

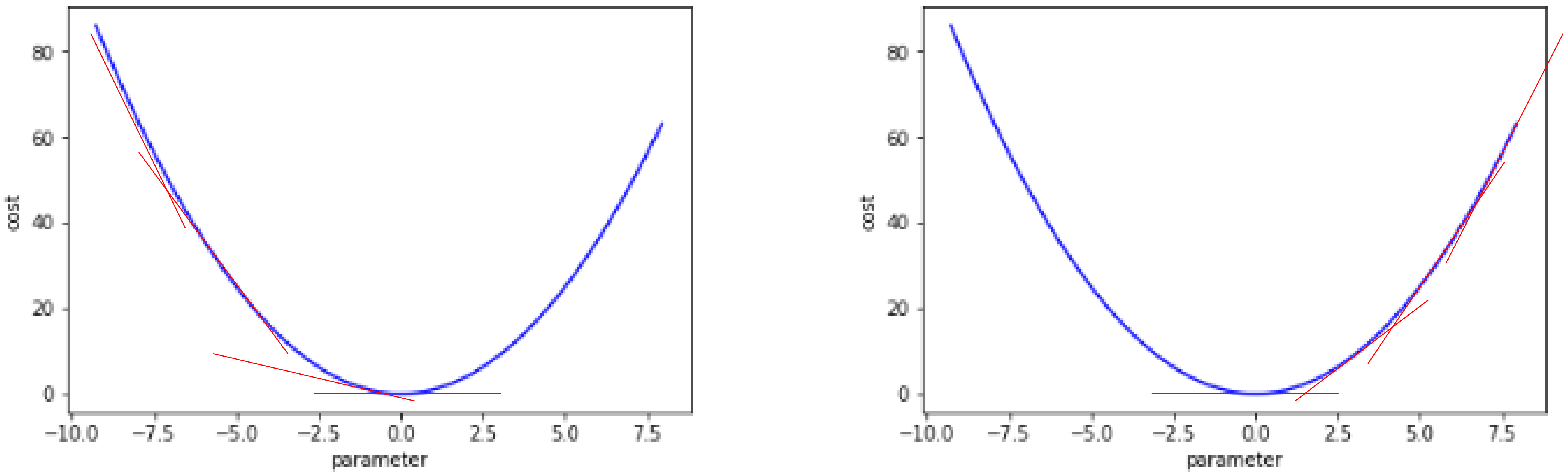
    $= \underset{w_0, w_1}{\operatorname{argmin}} \sum_{i=1}^{n} \{Y_i - (w_0 + w_1 X_i)\}^2$
  - 이 때의 $\widehat{w_0}, \widehat{w_1}$이 각각 단순회귀직선의 intercept와 기울기

  - Matrix Form으로 Y = XW이며,

    $\widehat{W} = (X^T X)^{-1} X^T Y$ -> 2x1의 벡터가 도출된다.

1. 단순회귀

- 1-2. 경사하강법(Gradient Descent)



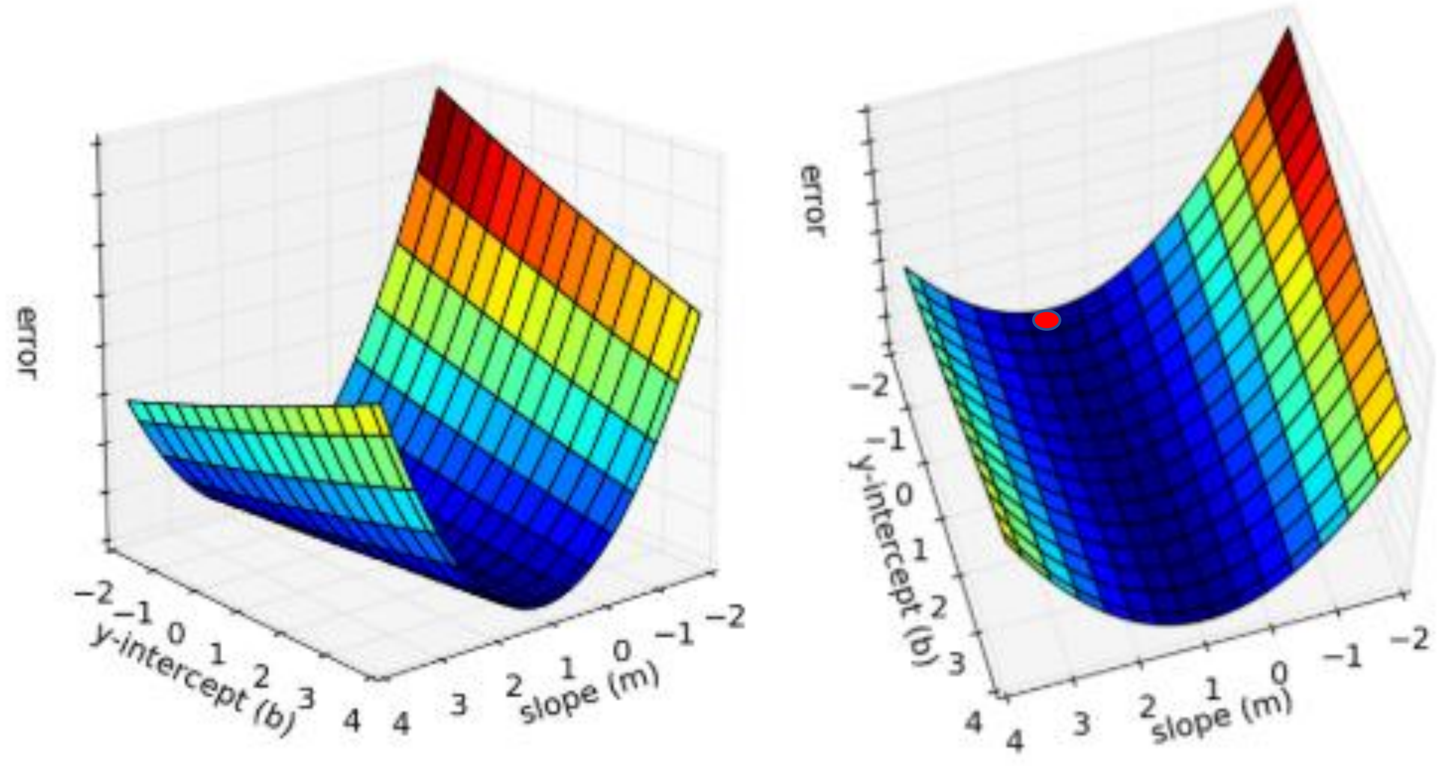
- 실제로 선형회귀의 cost function은 위와 같이 convex한 형태를 따른다.

1. 단순회귀

- 1-2. 경사하강법(Gradient Descent)
  - cost를 최소화 시킬 수 있는 방향으로 나아간다.
  - 이 cost는 Normal Equation의 MSE와 같은 개념
  - cost(W) = $\frac{1}{n}\sum_{i=1}^{n}\left(Wx^{(i)} - y^{(i)}\right)^2 (= MSE)$
  - 편의를 위해 cost(W) = $\frac{1}{2n}\sum_{i=1}^{n}(Wx^{(i)} - y^{(i)})^2$ 으로
  - W := W-$\alpha\frac{\partial}{\partial W}cost(W)$
  - 이 과정을 계속 반복하다 보면 결국 global minimum에 도달
  - 최적의 parameter를 찾아가는 것 ☞ 경사하강법을 통해 이해!

Growth Hackers

1. 단순회귀



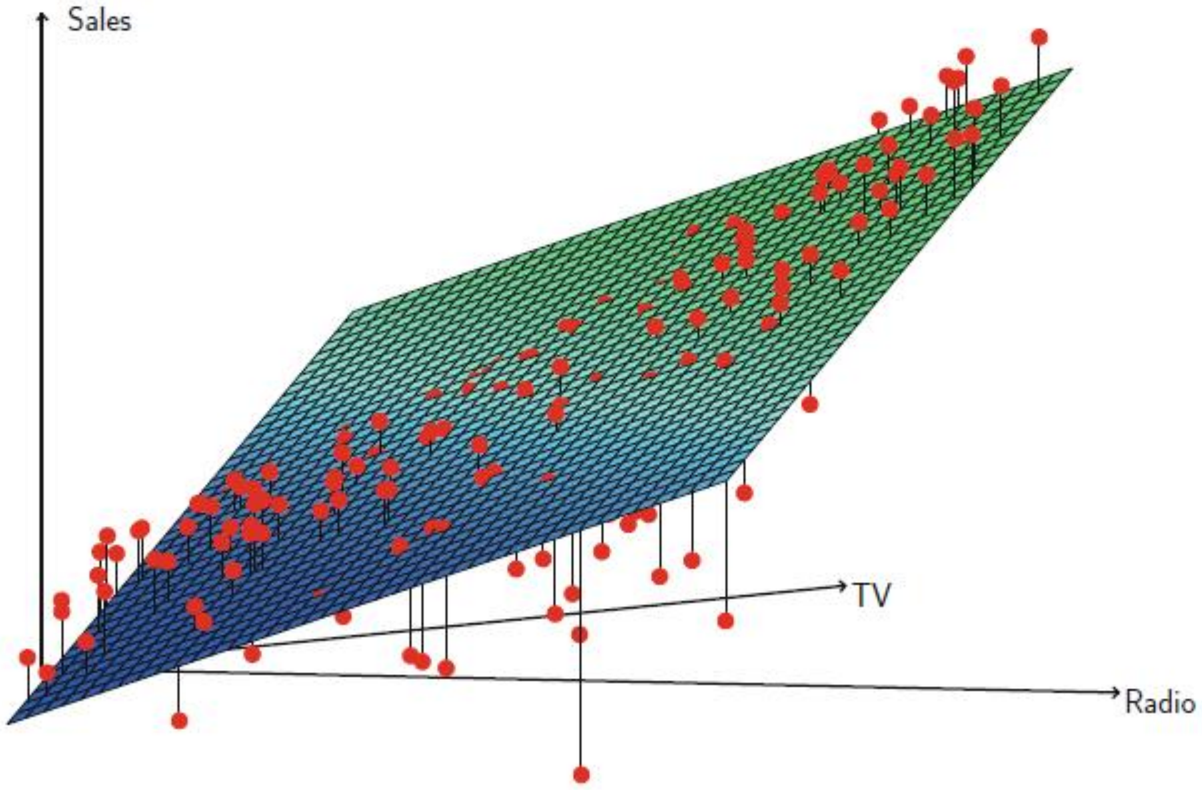
- 'Intercept와 기울기 간의 최적의 조합을 찾자!'

- 이 그림에서, $\hat{y}_i = -2 + 1.6x_i$가 단순회귀식

3/28/2019

2. 다중 회귀 (MULTIPLE REGRESSION)

2. 다중회귀



'독립변수가 하나가 아니라면?'
'변수가 추가되면 더 좋은 모델을 만들 수 있지 않을까?'

2. 다중회귀

- '단순선형회귀모델이 가장 적합한 모델은 아닐 것 같다.'
  → '더 복잡한 모델이 필요하겠다.'
  → '변수 개수를 늘려야겠다. (=parameter수가 늘어나야겠다.)
  → '다중회귀' 등장

- Y = XW + ϵ

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} W_0 \\ W_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

$$y_i = W_0 + W_1 x_i + \varepsilon_i$$

<단순선형회귀>

$\longrightarrow$

- Y = XW + ϵ

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ 1 & \cdots & x_{pn} \end{bmatrix} \begin{bmatrix} W_0 \\ \vdots \\ W_n \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

$$y_i = W_0 + W_1 x_1 + \cdots + W_n x_n + \varepsilon_i$$

<다중선형회귀>

Same Form!
(선형대수적 관점에서)

Growth Hackers

2. 다중회귀

- 따라서 다중회귀의 Cost Function의 형태도 단순선형회귀와 같다!

- cost(W) = $\frac{1}{n}\sum_{i=1}^{n}(Wx^{(i)} - y^{(i)})^2$

  (→ cost(W) = $\frac{1}{2n}\sum_{i=1}^{n}(Wx^{(i)} - y^{(i)})^2$)

2. 다중회귀

- 그런데, “모델이 복잡해진다는 것은?”

  - 단순히 하나의 독립변수만으로 종속변수를 잘 설명할 수 있는 데이터라면 단순선형회귀 모델을 사용해도 좋을 것.

  - 하지만 그렇지 않은 데이터라면 모델도 복잡해져야 하는 것이 맞다!

- 회귀모델에서는 보통 MSE를 통해 성능을 측정

  For given x_0, $E[y_0 - \hat{f}(x_0)]^2 = Var\left(\hat{f}(x_0)\right) + [Bias(\hat{f}(x_0))]^2 + \mathrm{Var}(\varepsilon)$

  모델이 복잡해진다? → 모델의 Bias 감소 but 모델의 Variance 증가

  모델이 단순해진다? → 모델의 Variance 감소 but 모델의 Bias 증가

- 따라서, 단순선형으로 표현되기 어려운 데이터에서 더 복잡한 모델을 쓴다는 것

  - (어느 지점까지) 모델의 Variance가 증가함에도 그 증가 폭보다 $Bias^2$의 감소폭이 더 크기에 MSE는 더 작아진다!

  → 좀 더 복잡한 모델이 더 적절하다!

Growth Hackers

2. 다중회귀

```
from sklearn.linear_model import LinearRegression

sim_reg = LinearRegression()
sim_reg.fit(X_sim_train, y_sim_train)
sim_reg.score(X_sim_test, y_sim_test)

print('carat을 독립변수로 하는 단순선형회귀모델의 score: {:f}')
```

carat을 독립변수로 하는 단순선형회귀모델의 score: 0.849

다중회귀에서 feature=3으로 가정하고, 그 독립변수를 carat과 tdp,table로 선택 //단지 설명력이 높아진다는 걸 보이고 싶은 것이기 때문에 임의로 선택한 것

```
X_sev_features = diamond[['carat', 'tdp', 'table']]
```

```
X_mul2_train, X_mul2_test, y_train, y_test = train_test_split(X_sev_features, y, test_size=.3, random_state=1)

mul2_reg = LinearRegression()
mul2_reg.fit(X_mul2_train, y_train)
mul2_reg.score(X_mul2_test, y_test)

print('carat, tdp, table을 독립변수로 하는 단순선형회귀모델의 score: {:.3f}'.format(mul2_reg.score(X_mul2_test, y_test)))
```

carat, tdp, table을 독립변수로 하는 단순선형회귀모델의 score: 0.854

- 설명력
  : 다중 > 단순

- (cf. MSE와 R^2 값은 반비례)

- $R^2 = 1 - \frac{SSE}{SSTO} = \frac{SSR}{SSTO}$

- MSE = $\frac{SSE}{df\ of\ SSE}$

- (미세하지만 더 나은 모델)

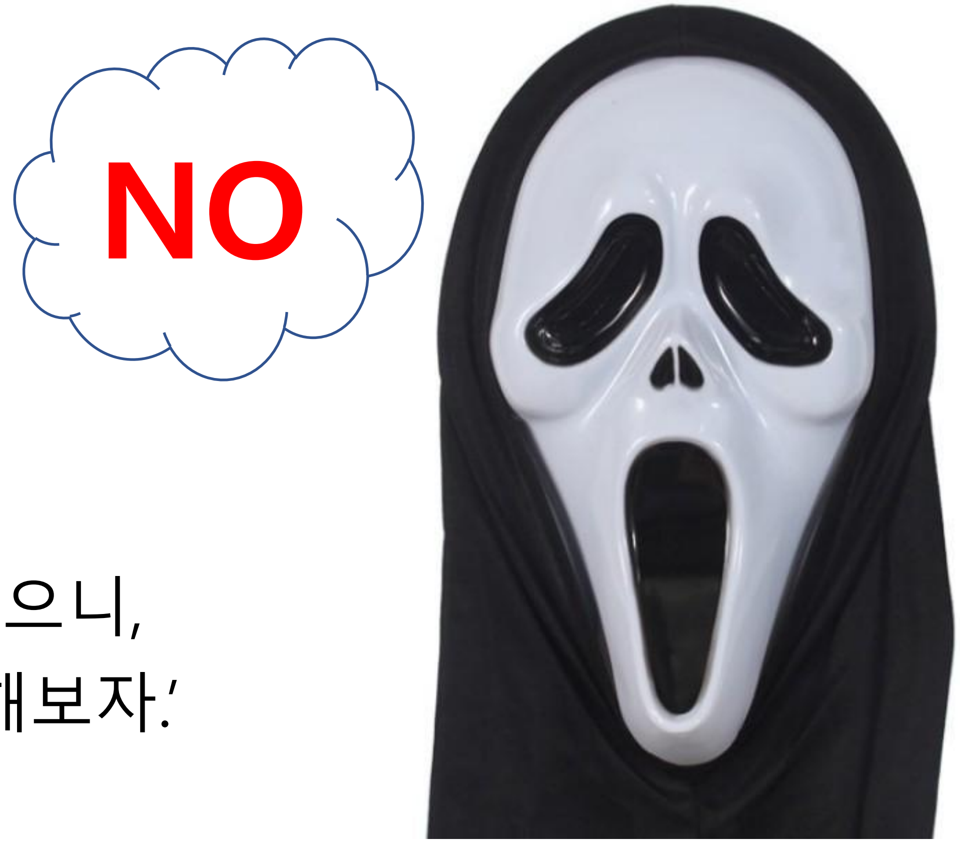
Growth Hackers

2. 다중회귀

- 의문점
  - '변수가 많을수록 좋을까?'

- 차원의 저주 (cf. 비지도 학습)

- '변수를 늘려오며 모델의 복잡도가 커졌으니,
  복잡도를 제어할 수 있는 모델을 생각해보자.'



3/28/2019

3. 릿지 회귀 (RIDGE REGRESSION)

Growth Hackers

3. 릿지회귀

- L2 규제

  - cost(W) = $\frac{1}{n}\sum_{i=1}^{n}(Wx^{(i)} - y^{(i)})^2 + \alpha\sum_{i=1}^{n} {w_i}^2$

  - 결과: |weight|을 가능한 한 0에 가깝게 만든다.

  - α↑→ 패널티↑ → weight↓
    α↓→ 패널티↓ → weight↑

3. 릿지회귀

```
In [141]: mult_model = LinearRegression()
          mult_model.fit(X_train, y_train)
          mult_model.coef_
          mult_model.score(X_test, y_test)

          print(mult_model.coef_)
          print('다중회귀의 score:{:f}'.format(mult_model.score(X_test, y_test)))

[[10397.37185625  -121.81880607   -41.66235333 -1155.72324407
     19.7933021     77.63214303  -999.42886334   -11.75181617
    498.05562555   227.6881908    285.43686317]]
다중회귀의 설명력:0.863948
```

```
In [146]: from sklearn.linear_model import Ridge

          Ridge1 = Ridge()
          Ridge20 = Ridge(alpha=10)
          Ridge1.fit(X_train, y_train)
          Ridge20.fit(X_train, y_train)

          print(Ridge1.coef_)
          print('릿지 alpha=1의 score: {:f}'.format(Ridge1.score(X_test, y_test)))
          print(Ridge20.coef_)
          print('릿지 alpha=10의 score: {:f}'.format(Ridge20.score(X_test, y_test)))

[[10371.25940966  -121.33755369   -41.64315472 -1144.56771688
     19.82526895    76.97585539  -998.69113682   -12.01876366
    497.85555975   227.51961291   285.33472782]]
릿지 alpha=1의 score: 0.863919
[[10142.70070187  -117.15643734   -41.47864189 -1047.355535
     20.27550329    71.62913565  -992.06561741   -14.37808546
    496.04438706   226.01349481   284.38582099]]
릿지 alpha=10의 score: 0.863633
```

- Ridge 모델을 씀으로써 Weight가 줄고 있다.
- 릿지모델에서 Alpha ↑ → score ↓
  : 과대적합을 줄인다.
- 다중회귀에 비해 릿지회귀를 사용 했을 때 과대적합이 줄고 있다.

Growth Hackers

3/28/2019

4. 라쏘 회귀 (LASSO REGRESSION)

4. 라쏘회귀

- L1 규제

  - cost(W) = $\frac{1}{n}\sum_{i=1}^{n}(Wx^{(i)} - y^{(i)})^2 \; + \alpha\sum_{i=1}^{n}|w_i|$

  - 결과: |weight|을 가능한 한 0에 가깝게 만든다.
    그런데 어떤 weight는 정말로 0이 된다.

  - α↑ → 패널티↑ → weight↓
    α↓ → 패널티↓ → weight↑

4. 라쏘회귀

```
[[10371.25940966  -121.33755369   -41.64315472 -1144.56771688
     19.82526895    76.97585539  -998.69113682   -12.01876366
    497.85555975   227.51961291   285.33472782]]
릿지 alpha=1의 score: 0.863919
[[10142.70070187  -117.15643734   -41.47864189 -1047.355535
     20.27550329    71.62913565  -992.06561741   -14.37808546
    496.04438706   226.01349481   284.38582099]]
릿지 alpha=10의 score: 0.863633
```

```
In [147]: from sklearn.linear_model import Lasso

Lasso1 = Lasso()
Lasso20 = Lasso(alpha=20)
Lasso1.fit(X_train, y_train)
Lasso20.fit(X_train, y_train)

print(Lasso1.coef_)
print('라쏘 alpha=1의 score: {:f}'.format(Lasso1.score(X_test, y_test)))
print(Lasso20.coef_)
print('라쏘 alpha=10의 score: {:f}'.format(Lasso20.score(X_test, y_test)))
```

```
[ 1.02681442e+04 -1.18074463e+02 -4.29373513e+01 -1.05292032e+03
  4.29441537e+00  2.25566909e+01 -1.18851200e+03 -2.26096681e+02
  2.66144943e+02 -0.00000000e+00  5.56858881e+01]
라쏘 alpha=1의 score: 0.863819
[7805.44283004 -103.10908054  -63.56969845  -13.2360921     -0.
   -0.         -452.70877782    -0.          176.94518452     0.
    0.        ]
라쏘 alpha=10의 score: 0.855919
```

- 라쏘모델을 씀으로써 Weight가 줄고 있다. (굉장히 가시적)
- 라쏘모델에서 Alpha ↑ → score ↓ : 과대적합을 줄인다.
- 릿지회귀에 비해 라쏘회귀를 사용 했을 때 과대적합이 줄고 있다.

Growth Hackers

3/28/2019

5. 로지스틱 회귀 (LOGISTIC REGRESSION)

Growth Hackers

5. 로지스틱회귀

- 회귀

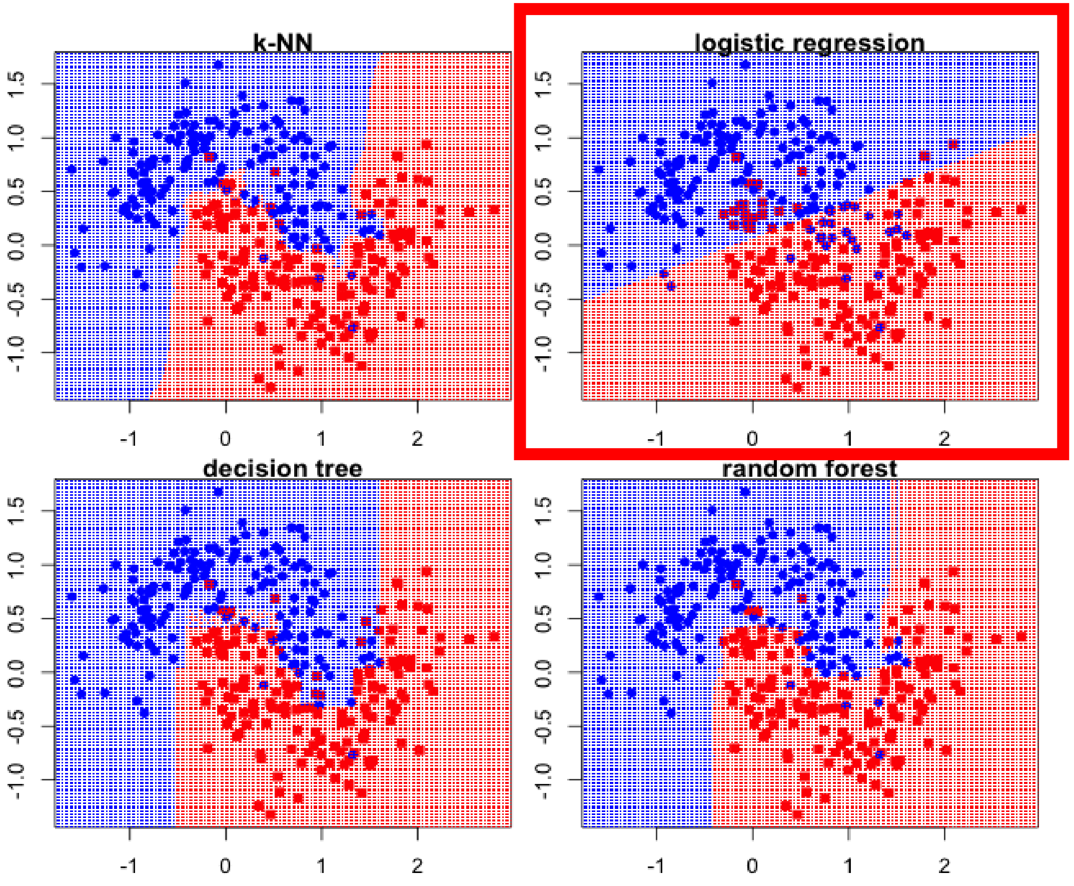
  - 1. 단순회귀
  - 2. 다중회귀
  - 3. 릿지회귀
  - 4. 라쏘회귀

  Target이 Quantitative

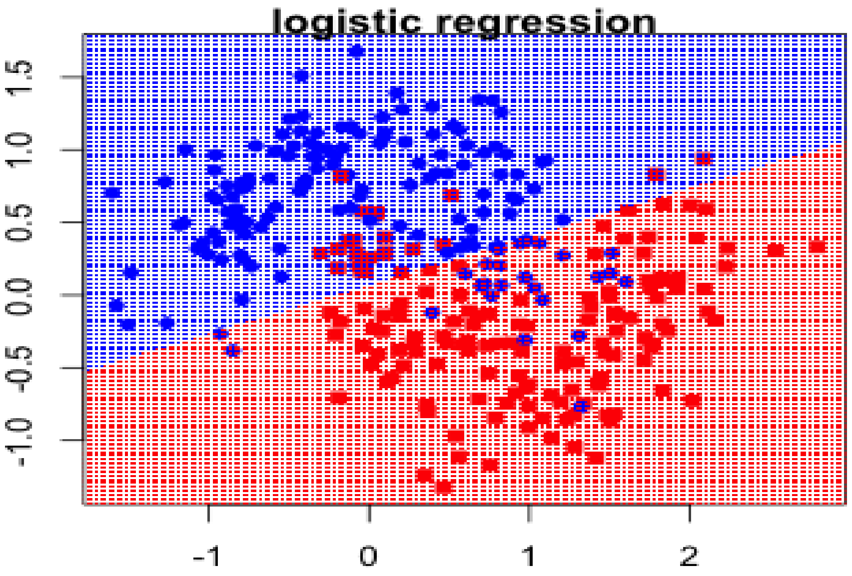
- 로지스틱회귀 — Target이 Qualitative

  - '분류'를 위한 알고리즘
  - 이진 분류에 사용 ex) Spam E-mail detection: Spam or Ham
  - 분류 알고리즘들 중 굉장히 정확도가 높은 알고리즘으로 알려져 있음

5. 로지스틱회귀

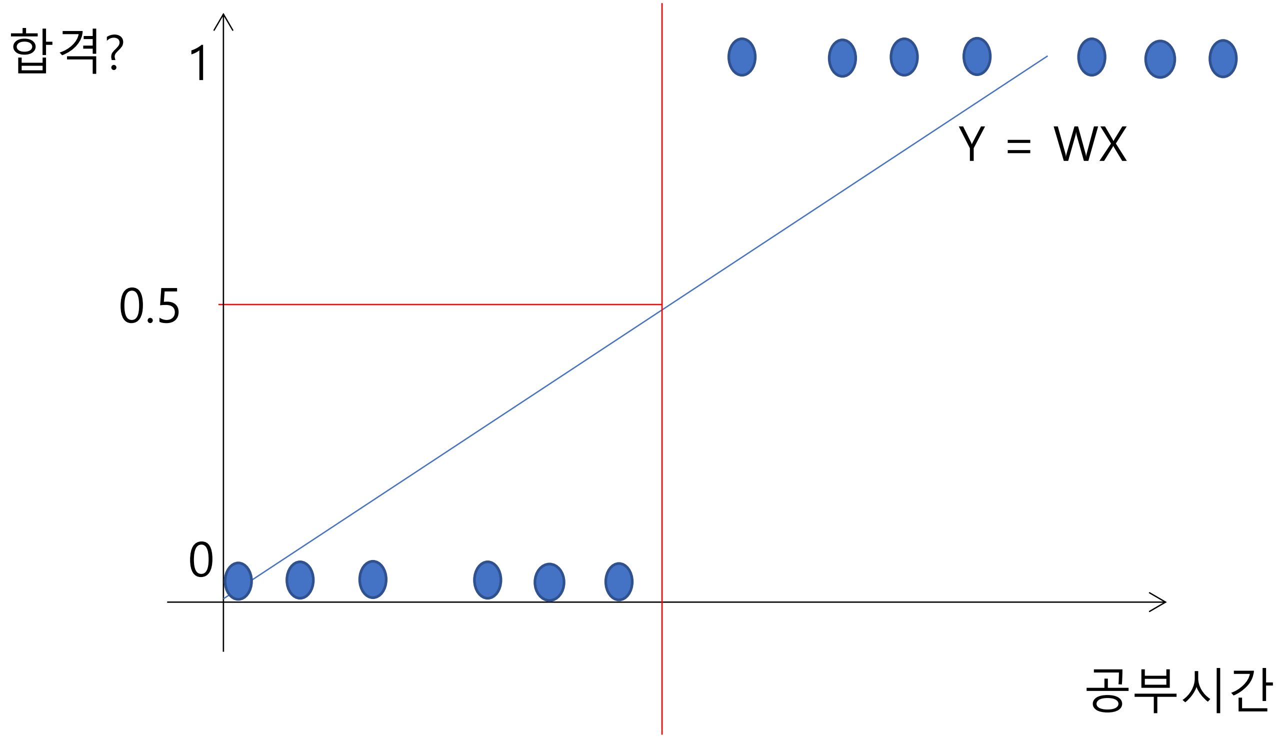


- 결정 경계(Decision Boundary)
  - 두 클래스의 영역을 나누는 경계



Growth Hackers

5. 로지스틱회귀

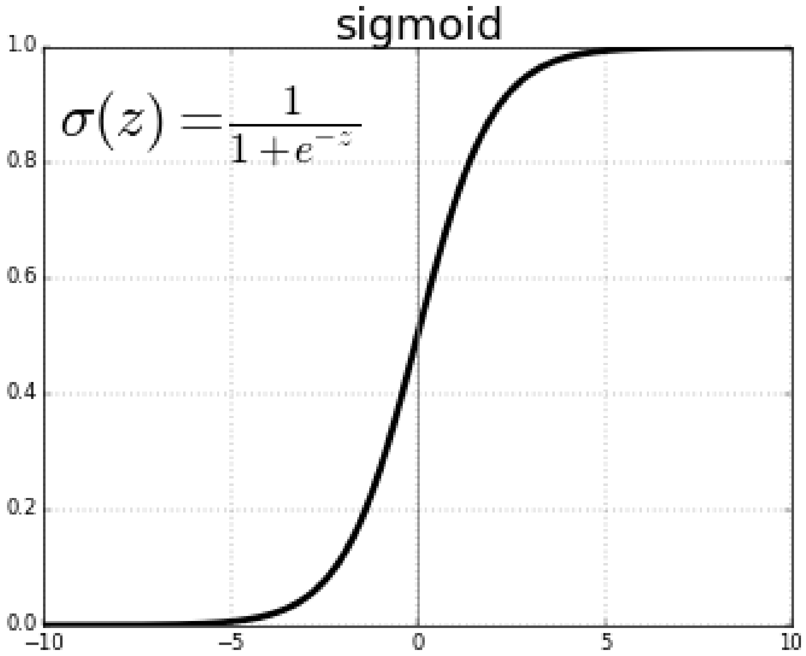


- 두 가지 문제

1. 점 하나하나에 지나치게 영향을 받는다.

2. 회귀 직선이 0~1 사이의 출력을 내어 놓을 것이란 보장이 없다.

5. 로지스틱회귀

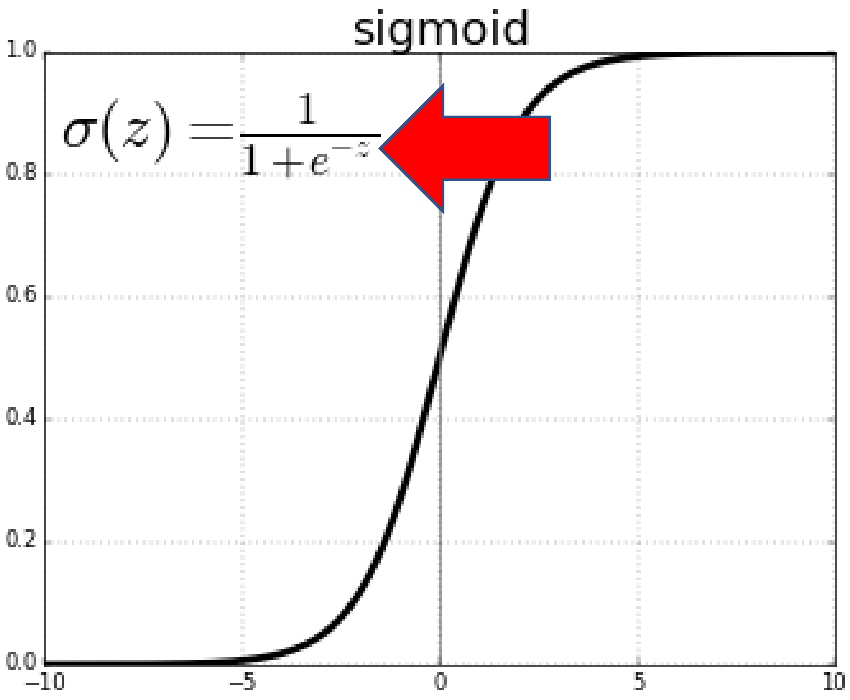


- Sigmoid Function
  (= Logistic Function)

  - 특징: 무슨 일이 있어도
    출력값이 0과 1 사이에 놓인다.

Growth Hackers

5. 로지스틱회귀



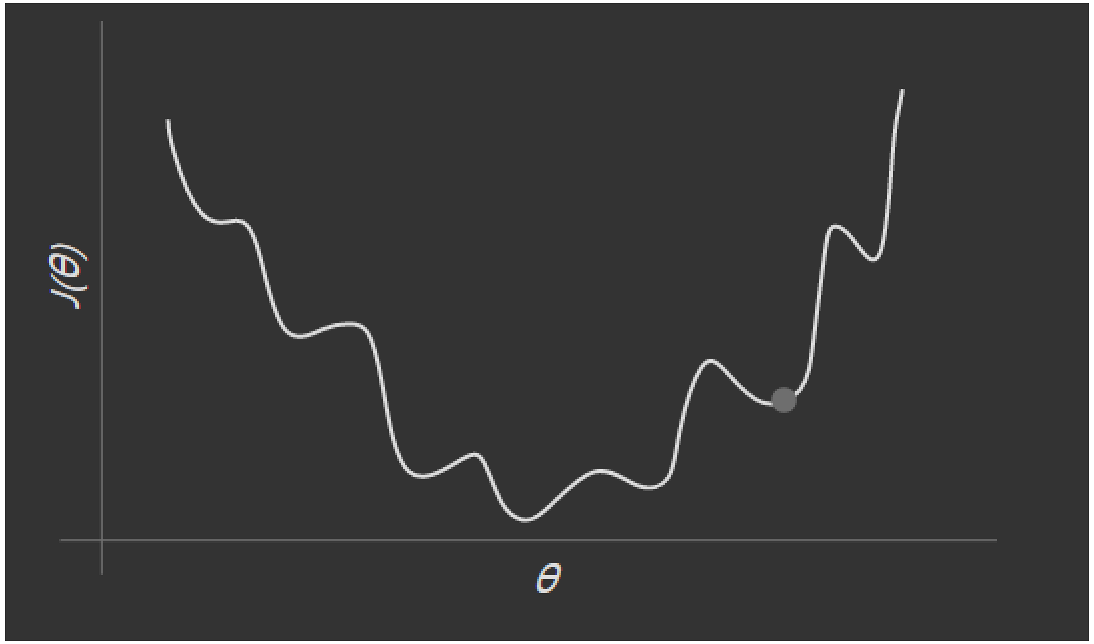
- Idea
  - ‘회귀식을 Sigmoid Function에 대입하면,
    출력 값이 반드시 0과 1 사이의 값이겠구나.’

대입 후

- H(X) = $\frac{1}{1+e^{-WX}}$
- 하지만…

5. 로지스틱회귀

- 문제 발생



- Cost function이 지나치게 울퉁불퉁

  -> 이 상태에서 '경사하강법' 사용
  -> Global Minimum을 찾기 힘들다

  - Learning Rate를 아무리 잘 조정해도 Global Minimum을 찾을 가능성은 굉장히 적다

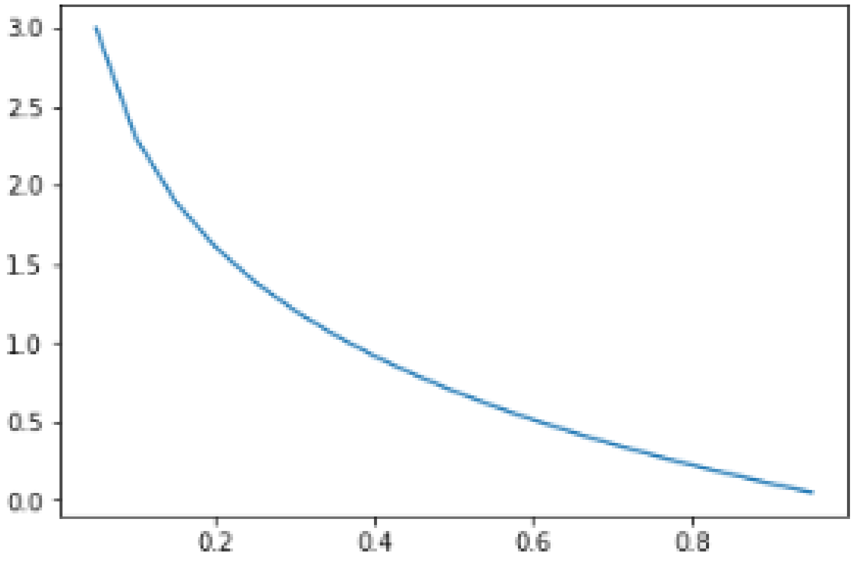
5. 로지스틱회귀

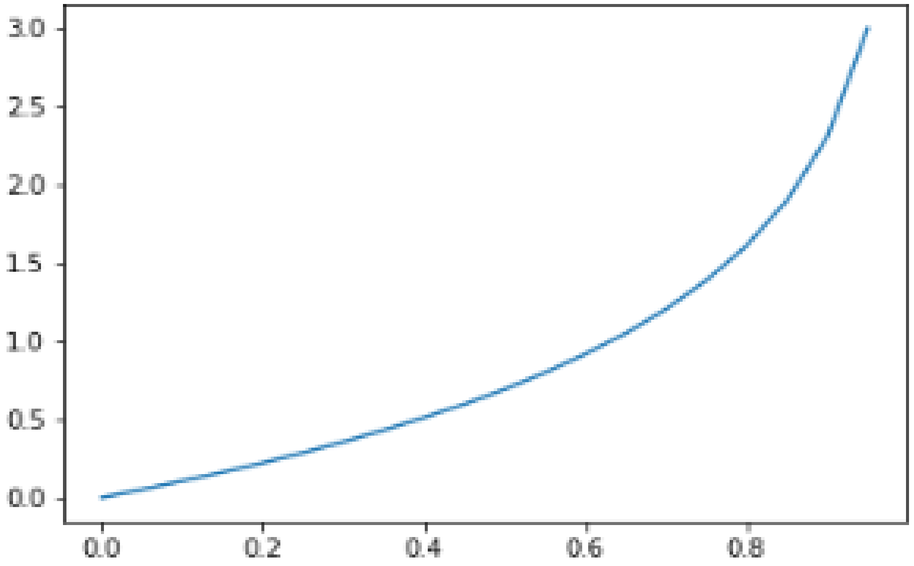
- '한 번의 변화를 더 주자.'
- cost(W) = $\frac{1}{n}\sum_{i=1}^{n} c(H(x), y)$
- c(H(x), y) = $\begin{cases} -\log\left(H(x)\right),\ if\ y = 1 \\ -\log\left(1 - H(x)\right),\ if\ y = 0 \end{cases}$

Growth Hackers

5. 로지스틱회귀

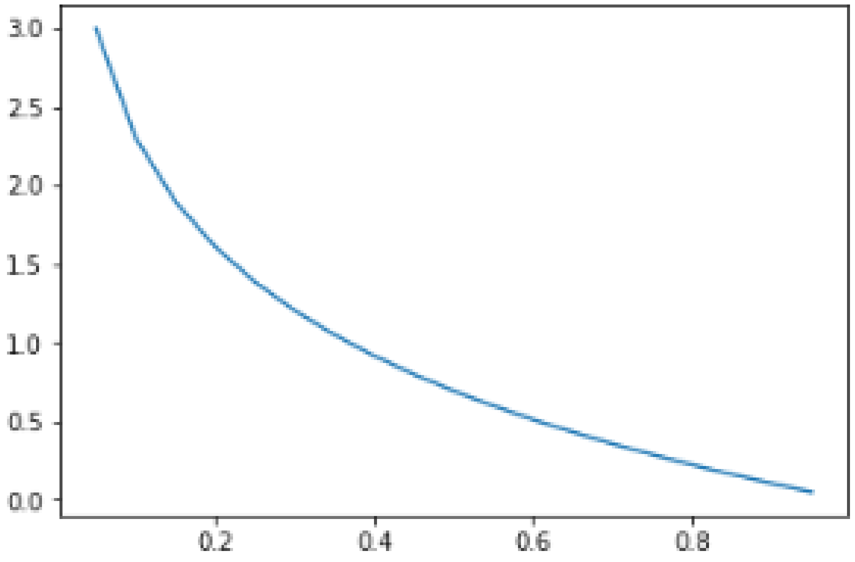
- $c(H(x), y) = \begin{cases} -\log(H(x)), & if\ y = 1 \\ -\log(1 - H(x)), & if\ y = 0 \end{cases}$

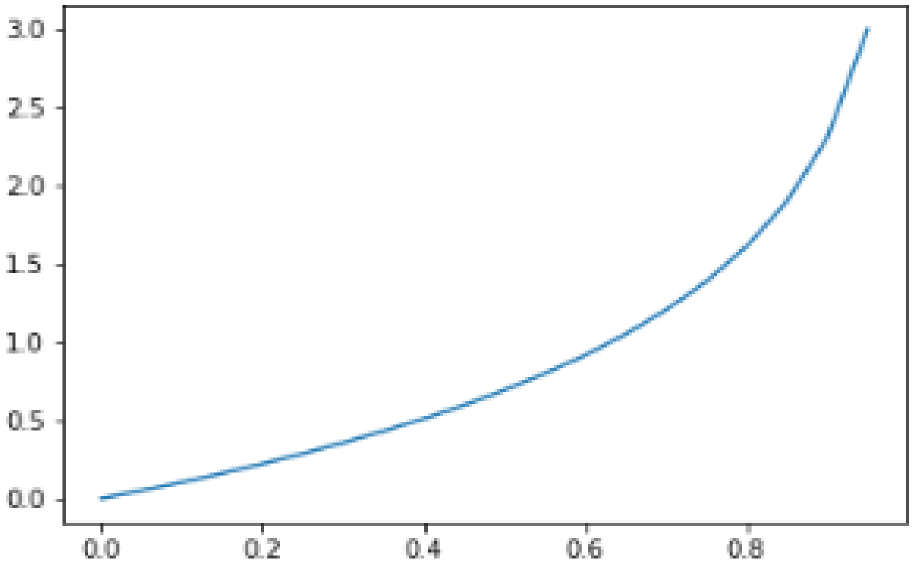




5. 로지스틱 회귀

- cost(W) $= -\frac{1}{n}\sum_{i}^{n} ylog\left(H(x)\right) + (1-y)\log(1-H(x))$





- W := W - $\alpha \frac{\partial}{\partial W} cost(W)$

Growth Hackers

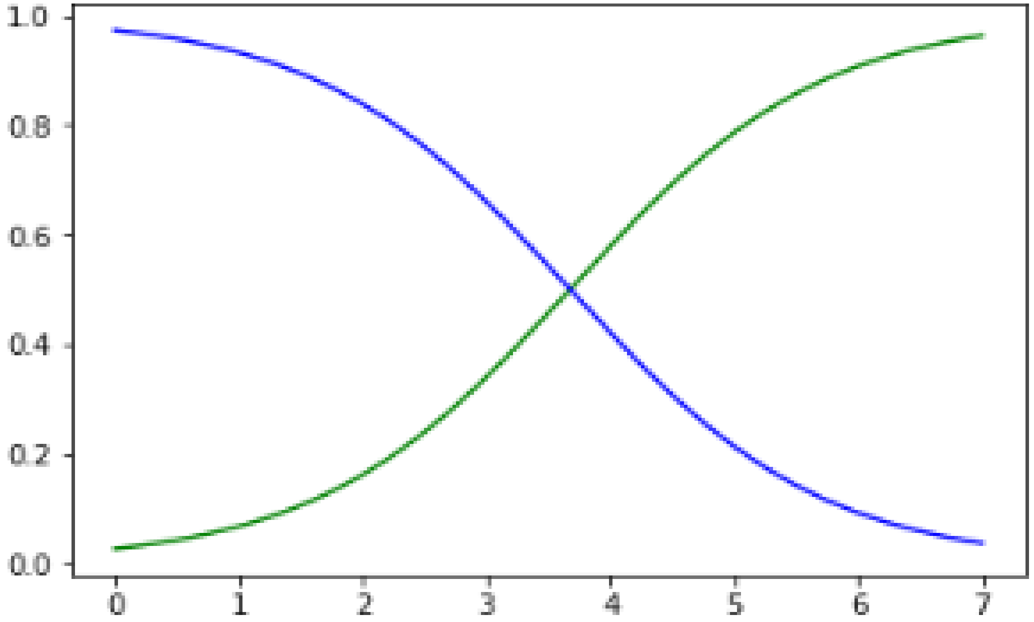
5. 로지스틱 회귀

- 고전적인 Iris data

- 꽃받침, 꽃잎의 길이와 너비를 바탕으로 꽃잎의 종을 예측 (분류)

Growth Hackers

5. 로지스틱 회귀



- 목적:
  '꽃받침 길이를 기준으로
  Sentosa/Not Sentosa를 분류하고 싶다.'
- 결과 해석:
  꽃받침 길이가 2이면 Not Sentosa
  꽃받침 길이가 4.5이면 Sentosa

Quest

- 유명한 데이터들 중 하나인 여성유방암 데이터 셋을 사용할 것입니다. Radius 변수를 기준으로 여성유방암 양성/음성을 분류하는 로지스틱 회귀분석 모델을 만들고, 이를 시각화하고, Radius 길이가 20, 0.1일 때의 결과를 해석해주세요.

- Session06_Regression.ipynb 파일 참고

- 파일 불러오기:

  from sklearn.datasets import load_breast_cancer

Growth Hackers