23 하계 학부연구생 프로그램

ch03. 회귀 알고리즘과 모델 규제

인공지능공학과 12223547 박혜민

농어 무게 예측 프로그램

01

회귀 알고리즘의 정확도 평가 방법

- R^2
- mean-absolute-error

02

알고리즘 선정

- k-Nearest Regression
- Linear Regression

03

모델 복잡도 결정 요인 1

- k 값

04

모델 복잡도 결정 요인 2

- 데이터량

05

모델 복잡도 결정 요인 3

- 특성의 개수

06

모델 복잡도 결정 요인 4

- 릿지, 라쏘의 alpha 값

회귀 알고리즘의 정확도 평가

이산적이지 않은 타깃값은 어떻게 정확도를 평가할까?

방법 1. 결정계수

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

SSE : 제곱 오차의 합 -> (타깃 – 예측)^2 의 합
SSR: 제곱 회귀의 합
SST: 총 제곱합 -> (타깃 – 평균)^2 의 합

R^2 = 0 → R^2 = 1

SSE 감소
타깃에 가깝게 예측

회귀 알고리즘의 정확도 평가

이산적이지 않은 타깃값은 어떻게 정확도를 평가할까?

방법2. mean_absolute_error

$$MAE = \frac{\sum_{i=1}^{n} \left| Y_i - \hat{Y}_i \right|}{n}$$

단점. 데이터 값의 크기에 의존하여 다양한 데이터 셋 비교 불가

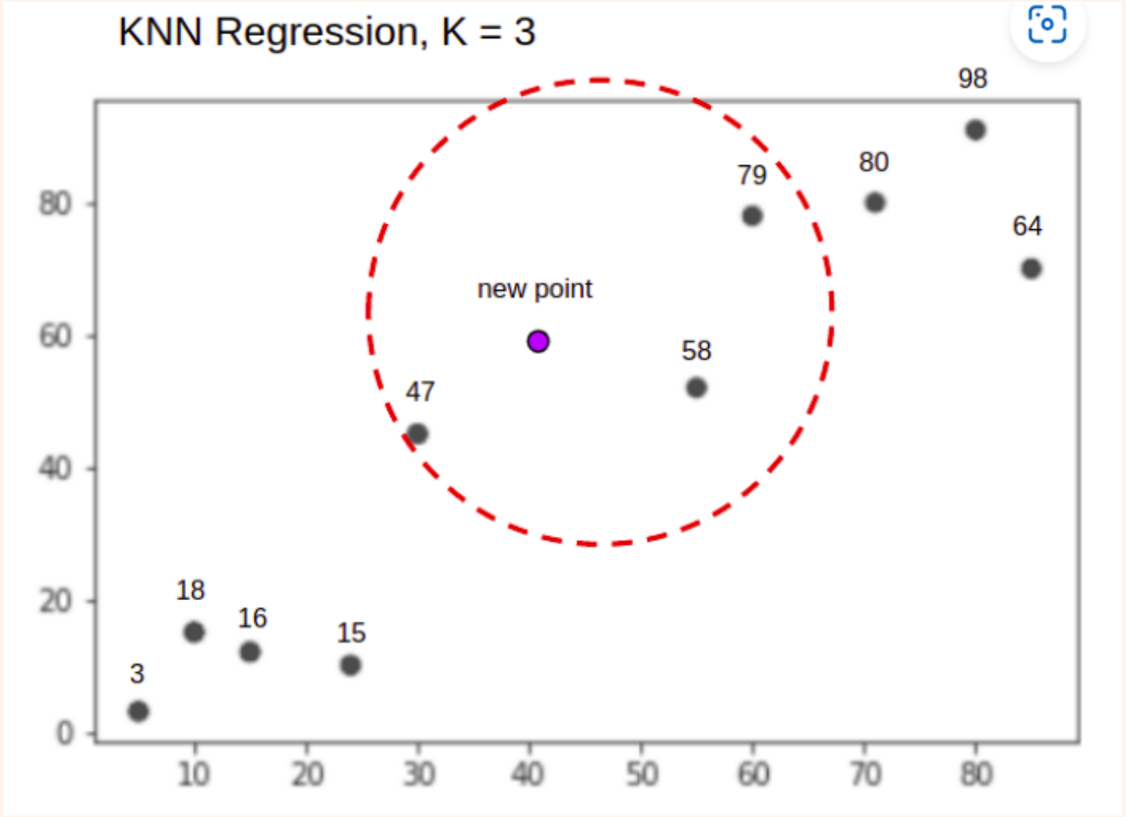
k-Nearest Regressor

주변의 가장 가까운 k개의 데이터를 참고하여 타깃값들의 평균을 정답으로 사용

대표 패키지 : sklearn

과정

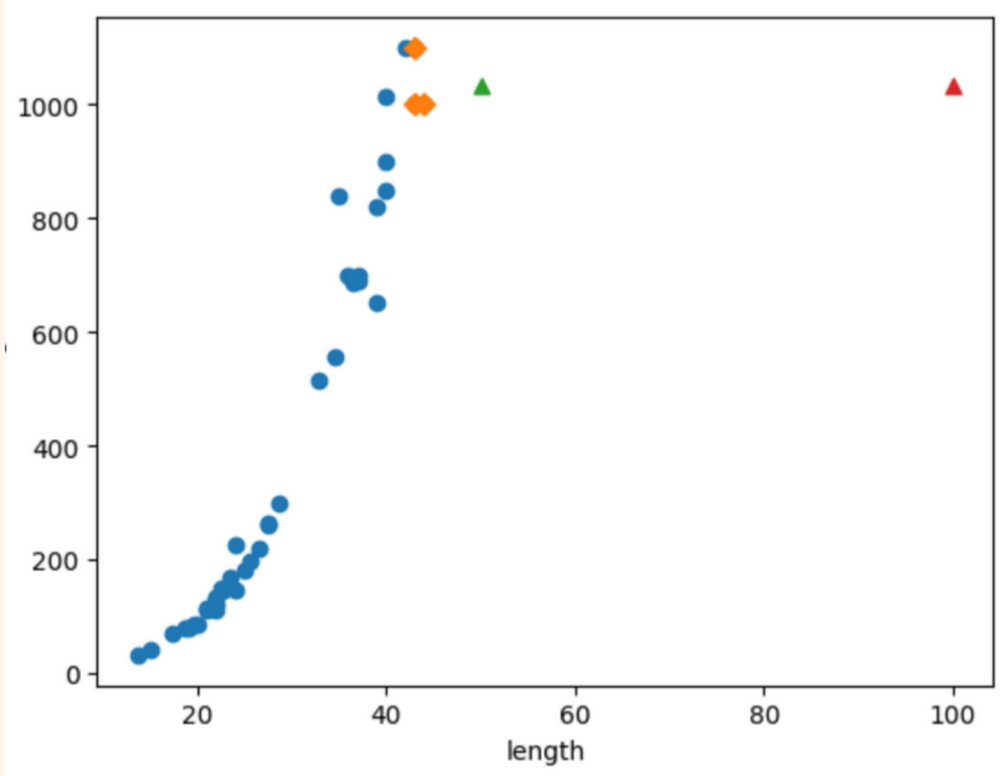
1. 새 샘플과 기존 데이터 간 거리 계산
2. 가장 가까운 k 개 선택
3. k 개 데이터들의 타깃값의 평균 계산



new point = (47 + 58 + 79) / 3

K-Nearest-Regressor의 문제점

아주 큰 길이의 농어가 들어온다면?



문제 1. 범위를 벗어나는 샘플에 대해 예측값이 실제값보다 매우 작다

- train_input : 14 ~ 43
- 테스트 샘플 : length-50, weight-1500
- 예측값 : 1033

문제2. 크기가 굉장히 큰 샘플들에 대해 예측값 동일하다

```
# k=3 인 최근접 이웃 모델
knr = KNeighborsRegressor(n_neighbors = 3)

knr.fit(train_input, train_target)

#50cm 농어 예측값
print(knr.predict([[50]]))
#100cm 농어 예측값
print(knr.predict([[100]]))
```

```
[1033.33333333]
[1033.33333333]
```

해결. 데이터의 경향을 선으로 연결하자

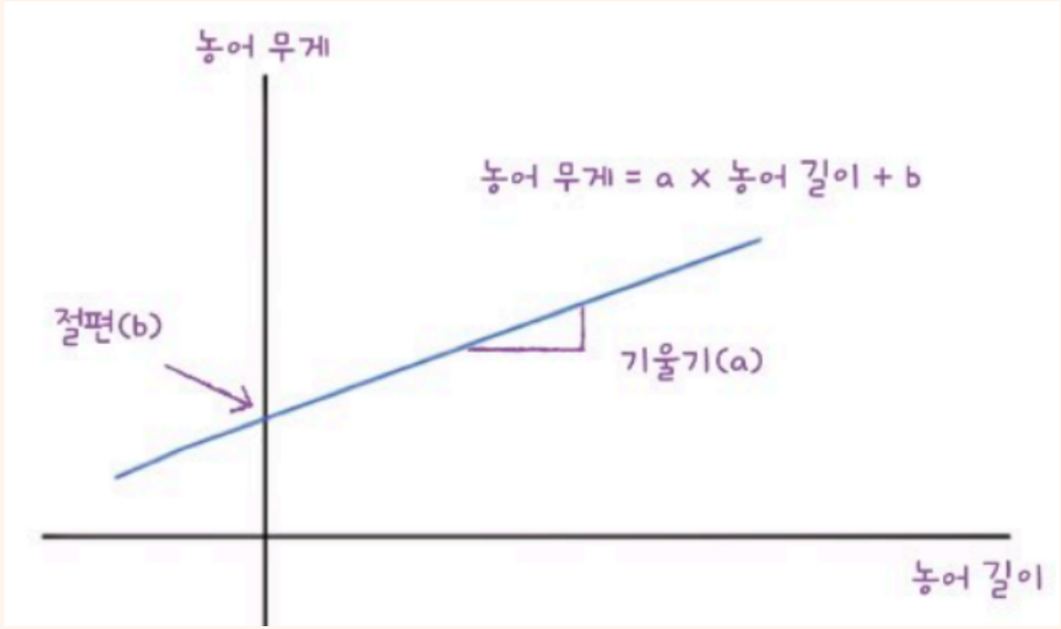
Linear Regressor

한 개 이상의 독립변수 x 와 종속변수 y의 선형 관계 모델링

분석 과정

- mean-sqared-error 의 최솟값을 내는 가중치와 편향 계산

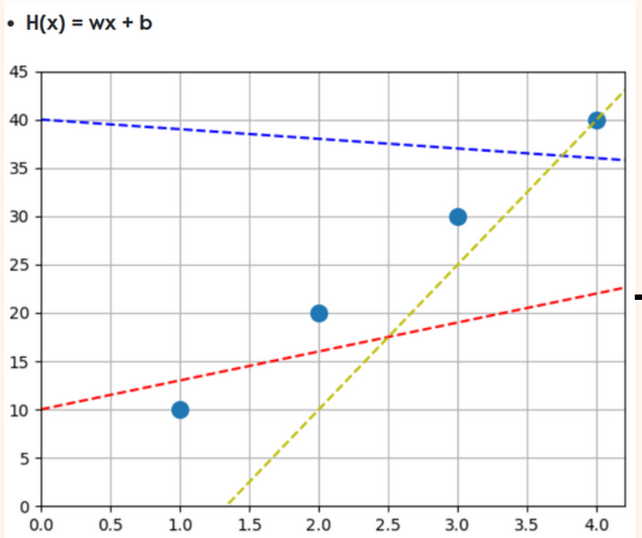
1. 단순 선형 회귀 분석 : 독립변수(특성) 1개
2. 다중 선형 회귀 분석 : 독립변수 여러개



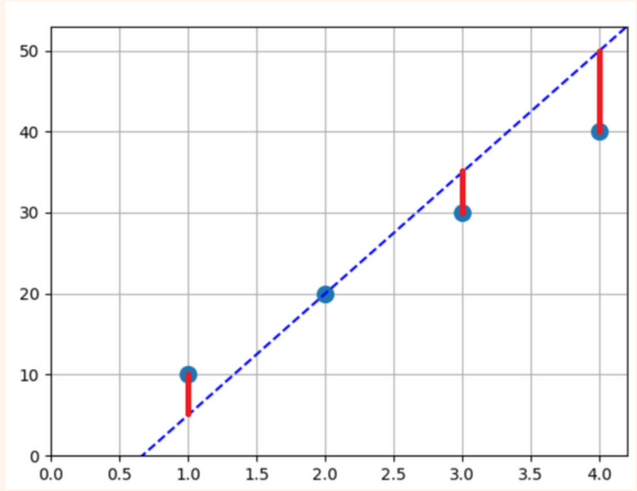
$$y = wx + b$$

Linear Regressor

분석 과정 - mean-sqared-error 의 최솟값 찾기



1. w, b에 대한 가설 설정



2. 모든 샘플에 대한 오차(실제값-예측값) 계산

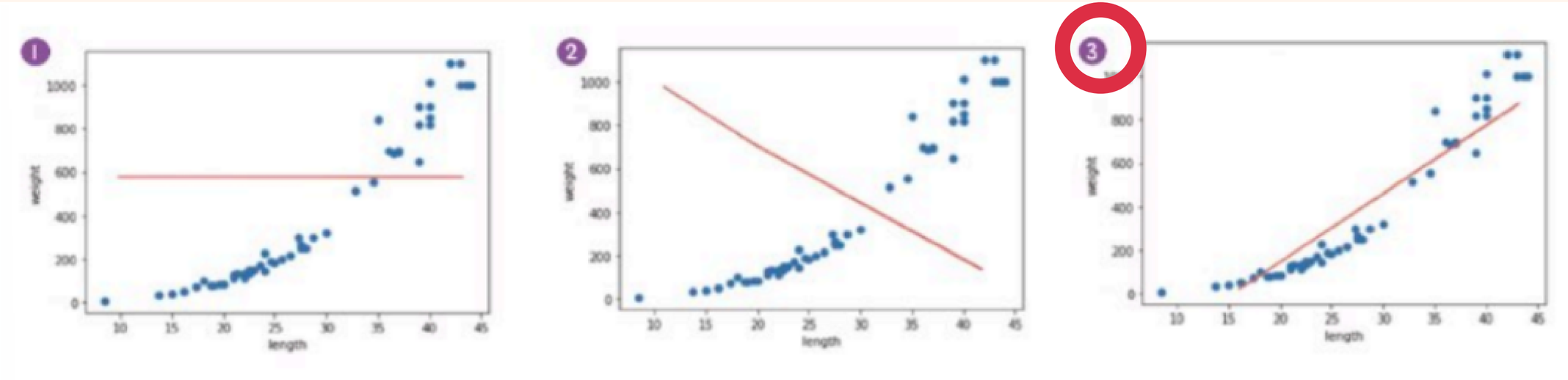
x	1	2	3	4
실제값	10	20	30	40
예측값	5	20	35	50
오차	5	0	-5	-10

$$\frac{1}{n}\sum_{i=1}^{n}[y_i - H(x_i)]^2 = \frac{5^2 + 0^2 + (-5)^2 + (-10)^2}{n} = \frac{150}{4} = 37.5$$

3. 평균 제곱 오차 계산

Linear Regressor

데이터 특성을 가장 잘 나타내는 직선은?

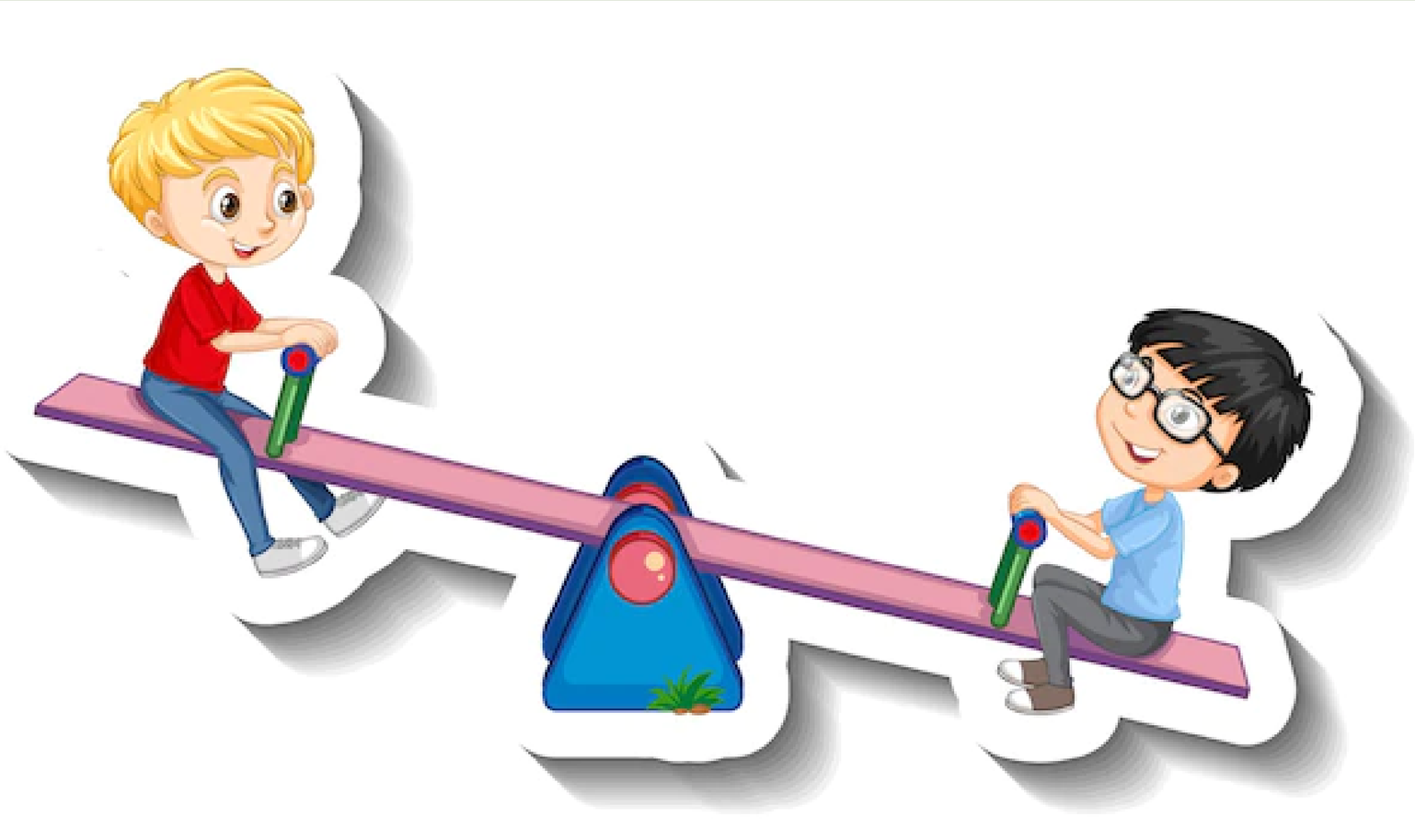


그래프 1: 농어 무게를 평균값 하나로 예측
-> R^2이 0에 가까움

그래프 2: 실제값과 반대로 예측
-> R^2 이 음수

모델이 복잡할수록 좋은가?

훈련 셋과 테스트 셋의 균형을 잘 유지하는 것이 중요



과대적합

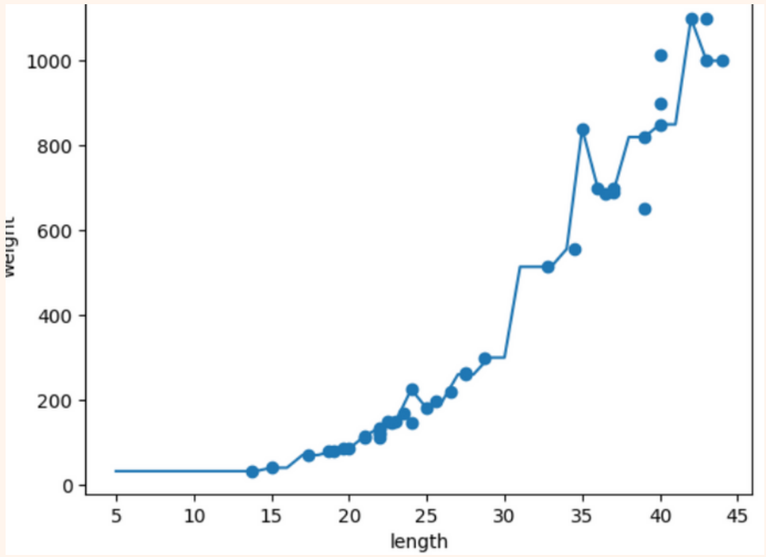
ex) 족보만 공부

과소적합

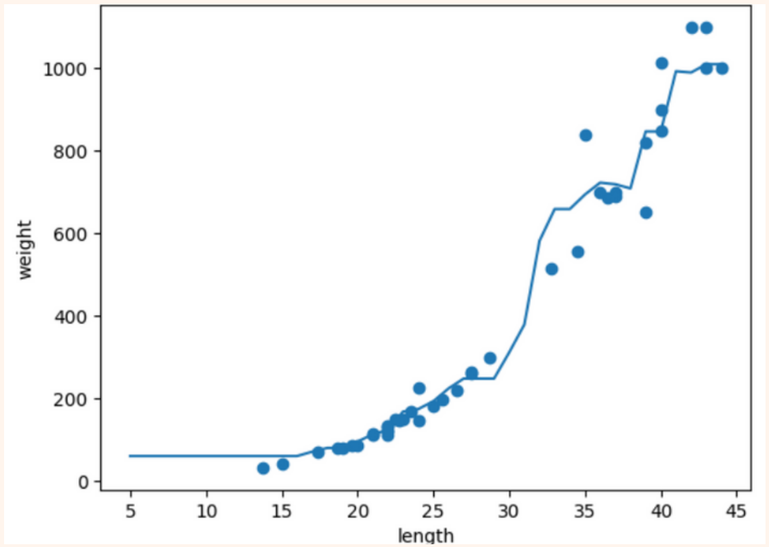
ex) 공부 안 함

최근접 이웃 회귀 – k 값

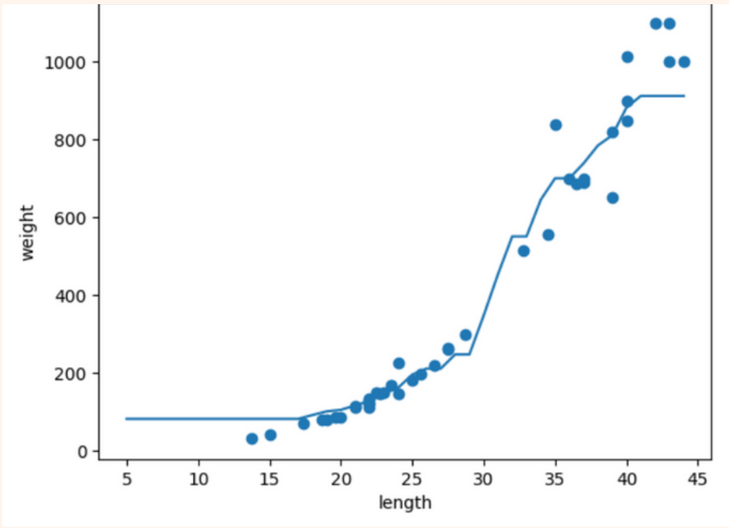
k = 1



k = 5



k = 10

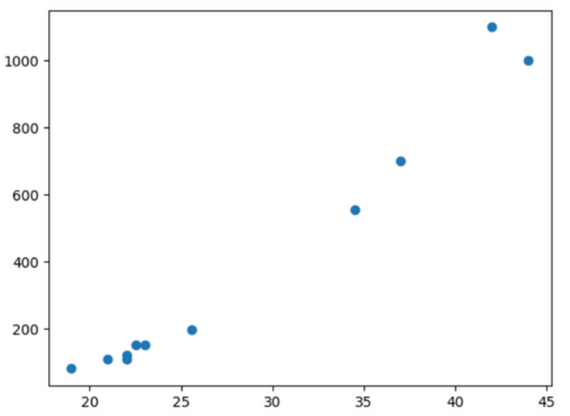


- 국지적인 패턴 학습
- 과대적합 유도

- 모델 단순해짐
- 과소적합 유도

데이터량

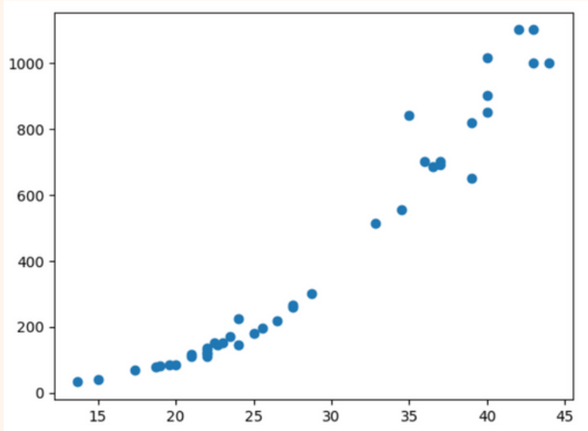
train set 비율 0.2



- train set 점수: 0.82
- test set 점수: 0.87

- train set 의 경향 거의 무시
- 과소적합 유도

train set 비율 7.5

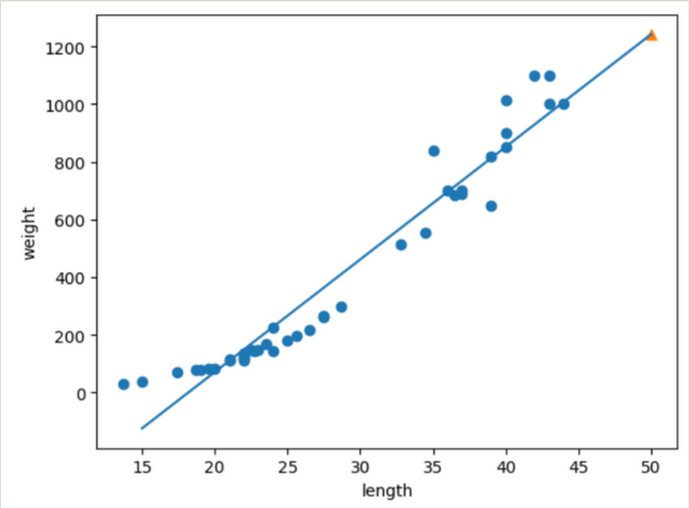


- train set 점수: 0.97
- test set 점수: 0.99

- train set 의 경향 잘 반영
- 과대적합 유도

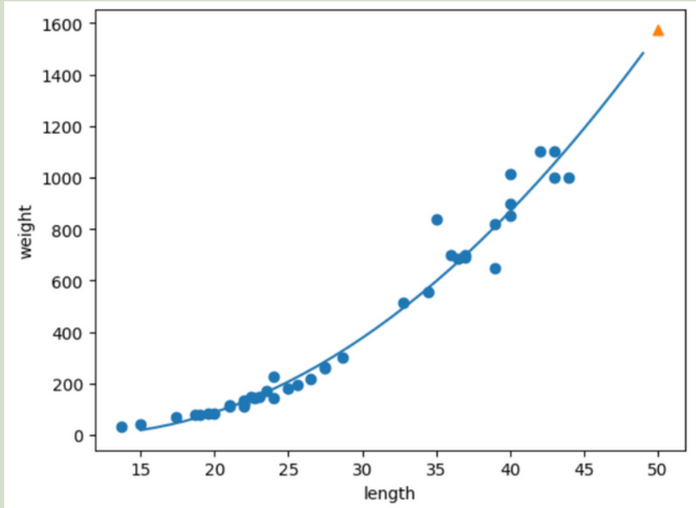
특성의 개수

단순 선형 회귀 - 특성 1개



train set 점수 : 0.94
test set 점수 : 0.82

다항 회귀 - 특성 2개



train set 점수 : 0.97
test set 점수 : 0.98

다중 회귀 - 특성 9개

train set 점수 : 0.99
test set 점수 : 0.97

다중 회귀 - 특성 55개

train set 점수 : 0.9999
test set 점수 : -144

- 특성 늘어날 수록 모델 복잡, 과대적합 유도

*특성곱을 추가하는 이유

특성 상호작용

- 하나의 특성에 대한 효과가 다른 특성값에 따라 결정되는 경우

ex) 집의 위치, 크기를 특성으로 하고 집값을 예측할 때, 위치와 크기 모두
좋은 조건일수록 예측변동량이 더욱 증가

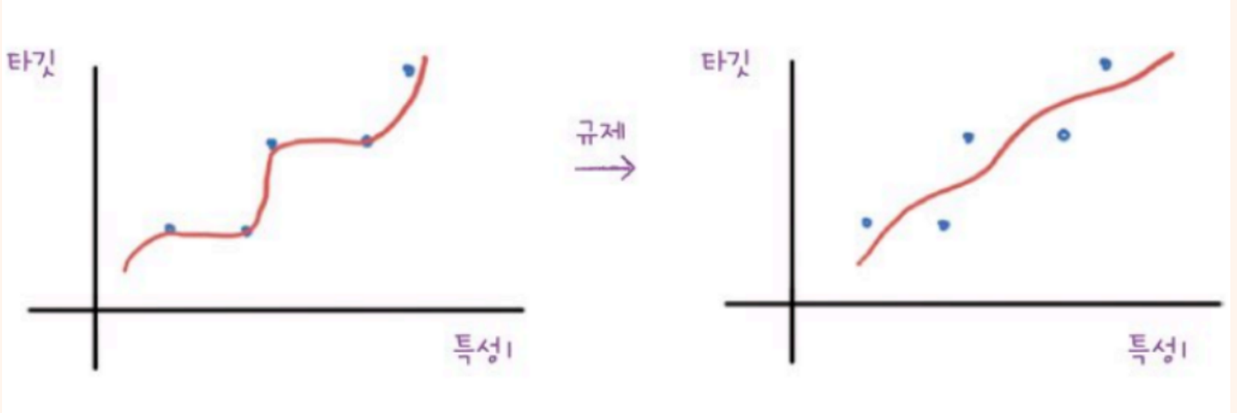
- 두 특성 간의 조합과 예측률 간의 관계를 살펴보기 위해
- 그 효과가 없다면 모델이 사용하지 않음

규제

특성에 곱해지는 계수(기울기) 감소

- 보편적인 패턴 학습
- 과대적합 억제

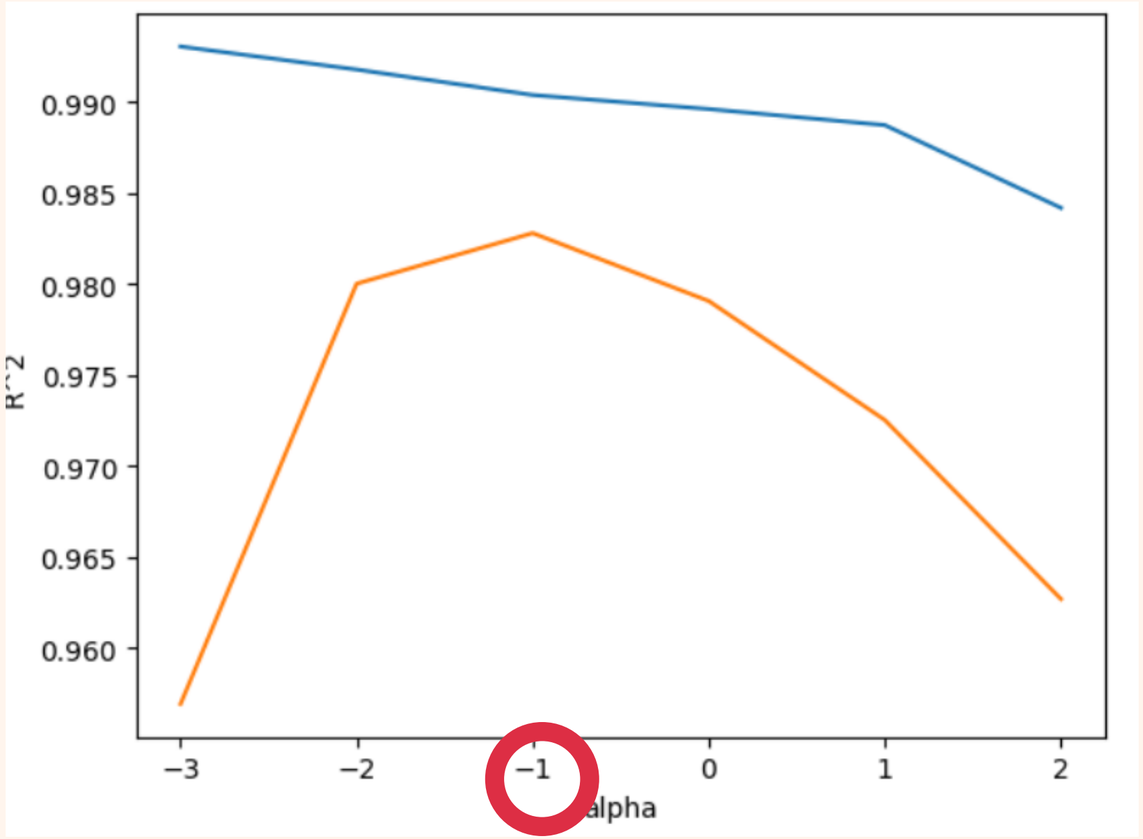
- 릿지 회귀
- 라쏘 회귀

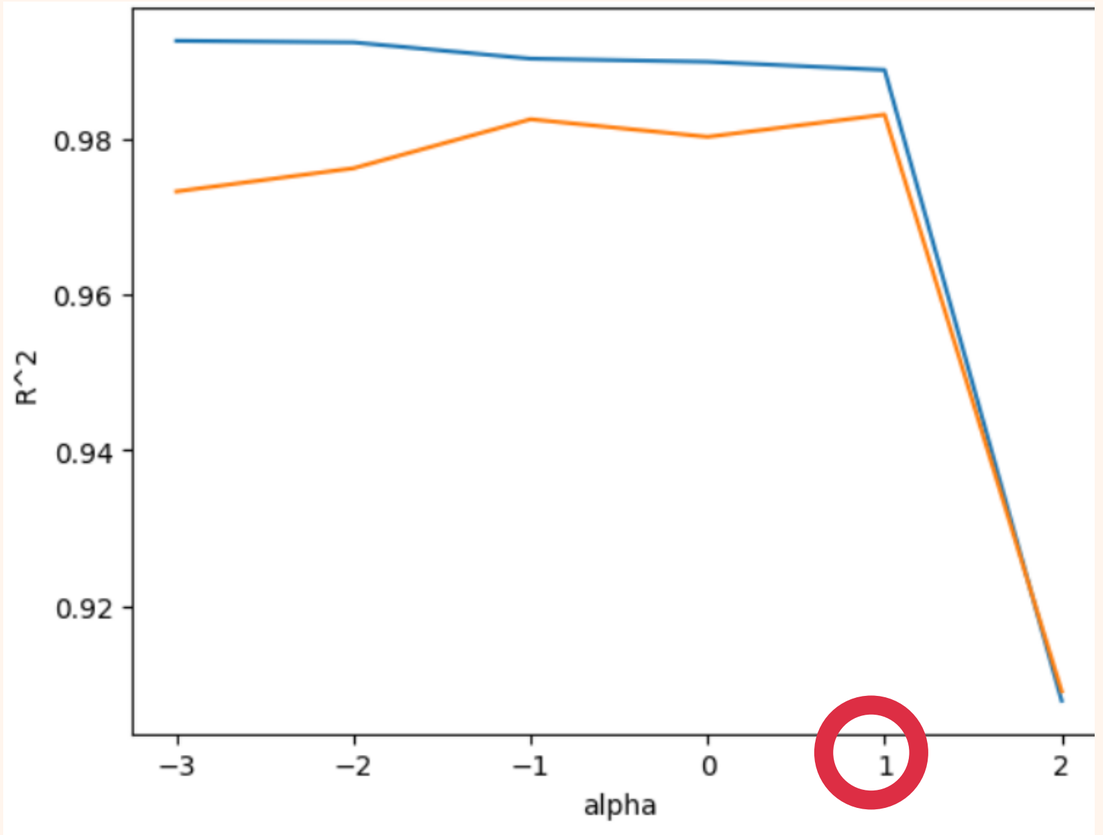


- alpha 큼 -> 규제 강함, 계수 더 줄어듬, 과소적합 유도
- alpha 작음 -> 규제 약함, 계수 덜 줄어듬, 과대적합 유도

적절한 alpha 값

훈련 셋, 테스트 셋 정확도 가장 가까운 지점





https://stackabuse.com/k-nearest-neighbors-algorithm-in-python-and-scikit-learn/

https://scikit-learn.org/stable/modules/linear_model.html

https://computer-nerd-coding.tistory.com/1