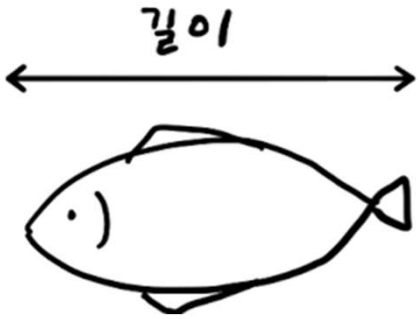
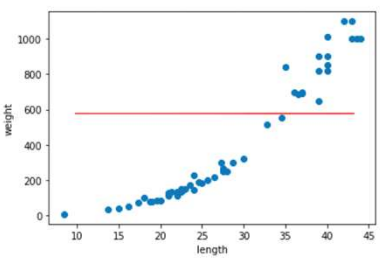
농어의 무게를 예측하라 2

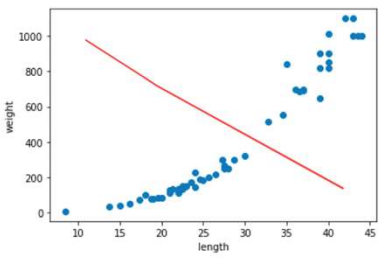


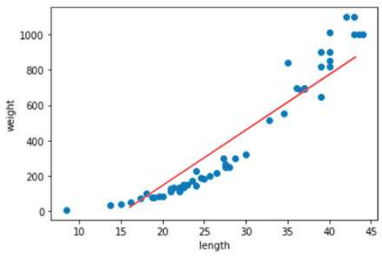
1

선형 회귀 Linear regression

- 대표적인 회귀 알고리즘
- 비교적 간단하고 성능이 뛰어남
- 특성이 하나인 경우 어떤 직선을 학습하는 알고리즘

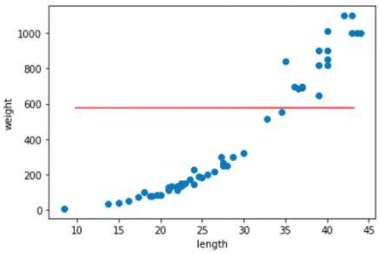




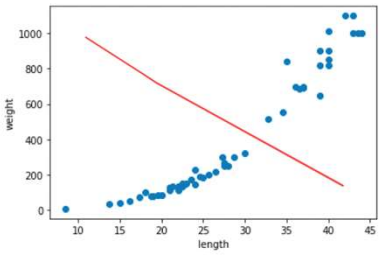


2

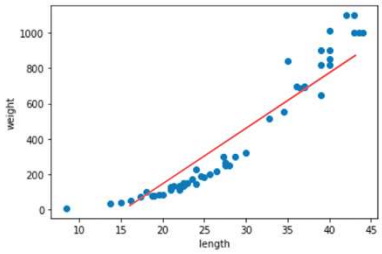
선형 회귀



- 모든 농어의 무게를 하나로 예측
- 직선의 위치가 만약 훈련 세트의 평균에 가깝다면 R^2는 0에 가까운 값이 됨



- 완전 반대로 예측함
- 길이가 작은 농어의 무게가 높음
- 길이가 큰 농어는 무게가 낮음
- 예측을 반대로 하면 R^2 는 음수가 됨



- 제일 그럴싸 함

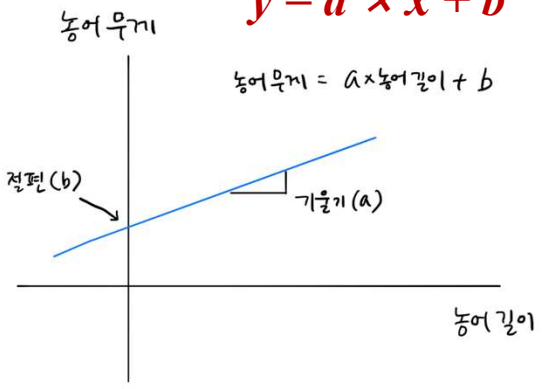
3

LinearRegression

```
from sklearn.linear_model import LinearRegression

lr = LinearRegression()
# 선형 회귀 모델 훈련
lr.fit(train_input, train_target)

# 50cm 농어에 대한 예측
print(lr.predict([[50]]))
[1241.83860323]

print(lr.coef_, lr.intercept_)
[39.01714496] -709.0186449535477
```

Model Parameter
- coef_ : Coefficient 계수(a 계수)
- Intercept_ : Weight 가중치(b 절편)

$$y = a \times x + b$$

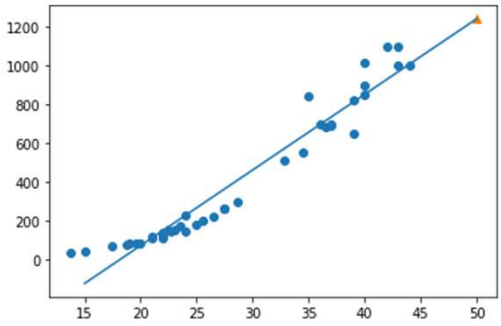


4

학습한 직선 그리기

```
# 훈련 세트의 산점도를 그립니다
plt.scatter(train_input, train_target)

# 15에서 50까지 1차 방정식 그래프를 그립니다
plt.plot([15, 50], [15*lr.coef_+lr.intercept_, 50*lr.coef_+lr.intercept_])

# 50cm 농어 데이터
plt.scatter(50, 1241.8, marker='^')
plt.show()
```

농어의 길이 15에서 50까지 직선으로 그려봄
길이 x 기울기 x 절편

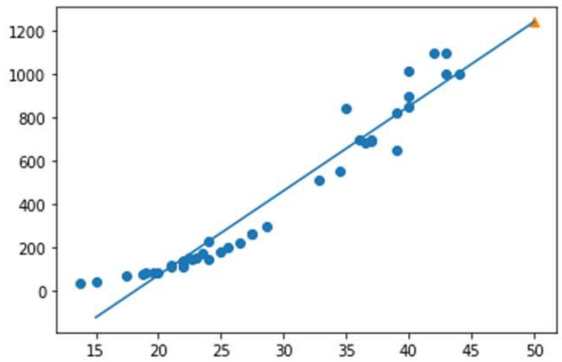


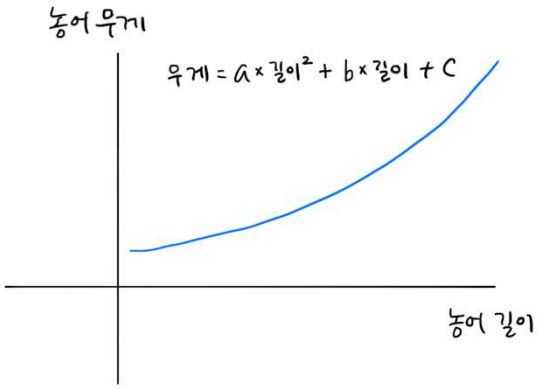

```
print(lr.score(train_input, train_target))
0.9398463339976039

print(lr.score(test_input, test_target))
0.8247503123313558
```

데이터 결과 : 과소적합

5

다항 회귀

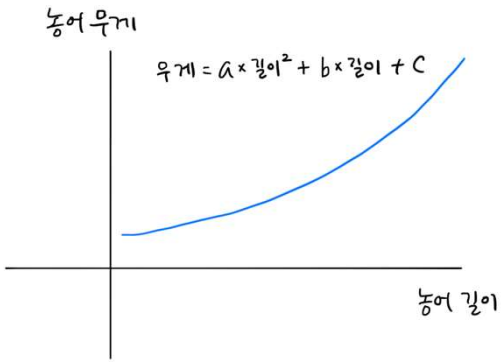


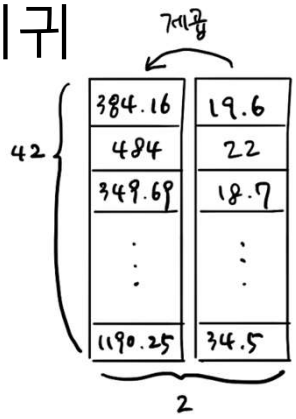


2차 방정식의 그래프를 그리기 위해 길이를 제곱한 항이 훈련세트에 추가 되어야 함

6

다항 회귀





농어의 길이를 제곱해
원래 데이터 옆에 붙임

```
train_poly = np.column_stack((train_input ** 2, train_input))
test_poly = np.column_stack((test_input ** 2, test_input))
```

7

모델 다시 훈련

```
lr = LinearRegression()
lr.fit(train_poly, train_target)

print(lr.predict([[50**2, 50]]))
[1573.98423528]

print(lr.coef_, lr.intercept_)
[  1.01433211 -21.55792498] 116.0502107827827
```

Model Parameter
- coef_ : Coefficient 계수(a 계수)
- Intercept_ : Weight 가중치(b 절편)

무게 = 1.01×길이² − 21.6×길이 + 116.05

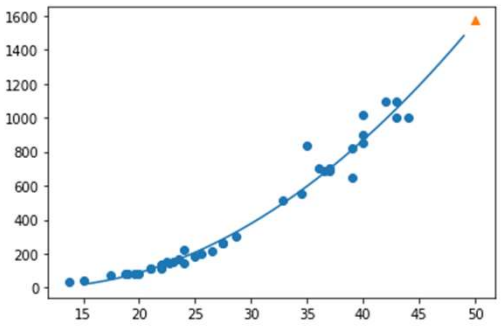
← 다항식
다항식을 이용한 선형회귀
다항회귀라고 함

8

학습한 직선 그리기

```
# 구간별 직선을 그리기 위해 15에서 49까지 정수 배열을 만듭니다
point = np.arange(15, 50)

# 훈련 세트의 산점도를 그립니다
plt.scatter(train_input, train_target)

# 15에서 49까지 2차 방정식 그래프를 그립니다
plt.plot(point, 1.01*point**2 - 21.6*point + 116.05)

# 50cm 농어 데이터
plt.scatter([50], [1574], marker='^')
plt.show()

print(lr.score(train_poly, train_target))
0.9706807451768623
print(lr.score(test_poly, test_target))
0.9775935108325122
```

그래도 과소적합이 보임 ㅠㅠ



9

감사합니다

내용 출처 정보 : https://www.hanbit.co.kr/store/books/look.php?p_code=B2002963743

10