

k-최근접 이웃 회귀



데이터 준비

```
In [1]: import numpy as np
```

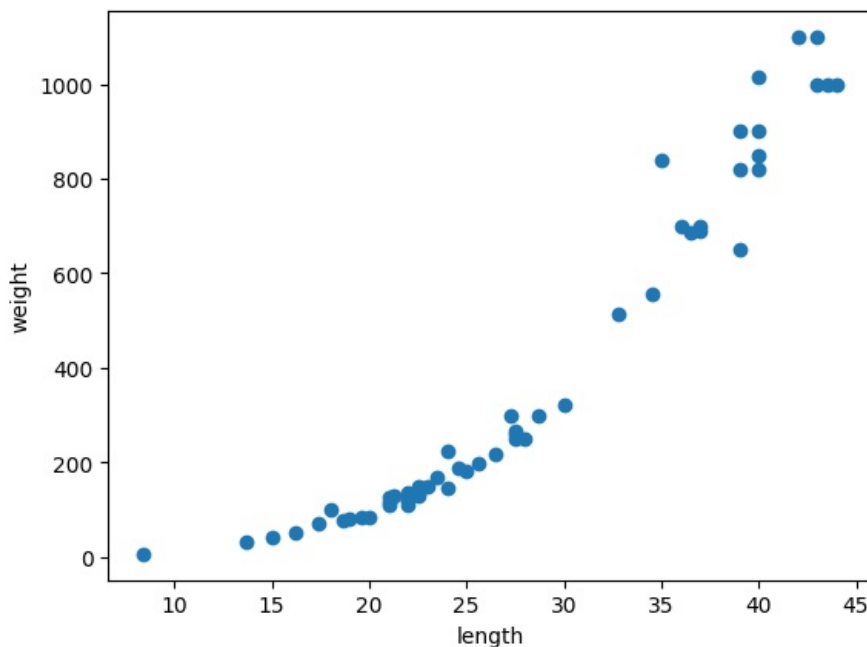
```
In [2]: perch_length = np.array([8.4, 13.7, 15.0, 16.2, 17.4, 18.0, 18.7, 19.0, 19.6, 20.0,
    21.0, 21.0, 21.0, 21.3, 22.0, 22.0, 22.0, 22.0, 22.0, 22.5,
    22.5, 22.7, 23.0, 23.5, 24.0, 24.0, 24.6, 25.0, 25.6, 26.5,
    27.3, 27.5, 27.5, 27.5, 28.0, 28.7, 30.0, 32.8, 34.5, 35.0,
    36.5, 36.0, 37.0, 37.0, 39.0, 39.0, 39.0, 40.0, 40.0, 40.0,
    40.0, 42.0, 43.0, 43.0, 43.5, 44.0]) # 놓여 길이
perch_weight = np.array([5.9, 32.0, 40.0, 51.5, 70.0, 100.0, 78.0, 80.0, 85.0, 85.0,
    110.0, 115.0, 125.0, 130.0, 120.0, 120.0, 130.0, 135.0, 110.0,
    130.0, 150.0, 145.0, 150.0, 170.0, 225.0, 145.0, 188.0, 180.0,
    197.0, 218.0, 300.0, 260.0, 265.0, 250.0, 250.0, 300.0, 320.0,
    514.0, 556.0, 840.0, 685.0, 700.0, 700.0, 690.0, 900.0, 650.0,
    820.0, 850.0, 900.0, 1015.0, 820.0, 1100.0, 1000.0, 1100.0,
    1000.0, 1000.0]) # 놓여 무게

print(perch_length.shape)
print(perch_weight.shape)

(56,)
(56,)
```

```
In [3]: import matplotlib.pyplot as plt
```

```
In [4]: plt.scatter(perch_length, perch_weight)
plt.xlabel('length')
plt.ylabel('weight')
plt.show()
```



```
In [5]: from sklearn.model_selection import train_test_split
```

```
In [6]: train_input, test_input, train_target, test_target = train_test_split(
    perch_length, perch_weight, random_state=42)
```

```
In [7]: print(train_input.shape, test_input.shape)

(42,) (14,)
```

```
In [8]: test_array = np.array([1,2,3,4])
print(test_array.shape)

(4,)
```

```
In [9]: test_array = test_array.reshape(2, 2)
print(test_array.shape)

(2, 2)
```

```
In [ ]: # 아래 코드의 주석을 제거하고 실행하면 에러가 발생합니다
# test_array = test_array.reshape(2, 3) # total 원소 개수가 다르므로 에러 발생
```

```
In [10]: train_input = train_input.reshape(-1, 1) # (42,1)
test_input = test_input.reshape(-1, 1) # (14,1)
```

```
In [11]: print(train_input.shape, test_input.shape)

(42, 1) (14, 1)
```

결정 계수 (R^2)

```
In [12]: from sklearn.neighbors import KNeighborsRegressor
# KNN regression model import
```

```
In [13]: knr = KNeighborsRegressor()
knr.fit(train_input, train_target)
```

```
Out[13]: ▼ KNeighborsRegressor
KNeighborsRegressor()
```

```
In [14]: knr.score(test_input, test_target)
# 결정 계수로 회귀의 결과를 평가 : 만약 평균 정도를 예측하는 수준이라면 0에 가까운 값, 예측이 타겟에 가까워지면 1에 가까운 값
```

```
Out[14]: 0.992809406101064
```

```
In [15]: from sklearn.metrics import mean_absolute_error # MAE
```

```
In [16]: test_prediction = knr.predict(test_input)
mae = mean_absolute_error(test_target, test_prediction)
print(mae)

19.157142857142862
```

과대적합 vs 과소적합

```
In [18]: print(knr.score(train_input, train_target))
# train data 결과 보다 test_data 결과가 더 높음 -> underfitting

0.9698823289099254
```

```
In [19]: knr.n_neighbors = 3 # 이전 모델은 default값으로 사용하였고 default 값은 n_neighbors = 5.
# 이웃 개수를 줄이면 좀 더 모델이 complex 해지므로 줄여보자
knr.fit(train_input, train_target)
print(knr.score(train_input, train_target))

0.9804899950518966
```

```
In [20]: print(knr.score(test_input, test_target))

0.9746459963987609
```