훈련 세트와 테스트 세트


구글 코랩에서 실행하기

훈련 세트와 테스트 세트

data

In [2]:
```
fish_length = [25.4, 26.3, 26.5, 29.0, 29.0, 29.7, 29.7, 30.0, 30.0, 30.7, 31.0, 31.0,
               31.5, 32.0, 32.0, 32.0, 33.0, 33.0, 33.5, 33.5, 34.0, 34.0, 34.5, 35.0,
               35.0, 35.0, 35.0, 36.0, 36.0, 37.0, 38.5, 38.5, 39.5, 41.0, 41.0, 9.8,
               10.5, 10.6, 11.0, 11.2, 11.3, 11.8, 11.8, 12.0, 12.2, 12.4, 13.0, 14.3, 15.0]
fish_weight = [242.0, 290.0, 340.0, 363.0, 430.0, 450.0, 500.0, 390.0, 450.0, 500.0, 475.0, 500.0,
               500.0, 340.0, 600.0, 600.0, 700.0, 700.0, 610.0, 650.0, 575.0, 685.0, 620.0, 680.0,
               700.0, 725.0, 720.0, 714.0, 850.0, 1000.0, 920.0, 955.0, 925.0, 975.0, 950.0, 6.7,
               7.5, 7.0, 9.7, 9.8, 8.7, 10.0, 9.9, 9.8, 12.2, 13.4, 12.2, 19.7, 19.9]
```

data preprocessing

In [9]:
```
fish_data = [[l, w] for l, w in zip(fish_length, fish_weight)]
fish_target = [1]*35 + [0]*14 # 도미 : 35마리 (1), 빙어 : 14마리 (0)
print(fish_target)
```

```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

In [4]:
```
from sklearn.neighbors import KNeighborsClassifier

kn = KNeighborsClassifier() # KNN
```

In [5]:
```
print(fish_data[4]) # Print fifth fish data : 도미
```

```
[29.0, 430.0]
```

In [6]:
```
print(fish_data[0:5]) # 도미 데이터 확인
```

```
[[25.4, 242.0], [26.3, 290.0], [26.5, 340.0], [29.0, 363.0], [29.0, 430.0]]
```

In [7]:
```
print(fish_data[:5]) # 위코드와 똑같은 방법으로 확인
```

```
[[25.4, 242.0], [26.3, 290.0], [26.5, 340.0], [29.0, 363.0], [29.0, 430.0]]
```

In [8]:
```
print(fish_data[44:]) # 빙어 데이터
```

```
[[12.2, 12.2], [12.4, 13.4], [13.0, 12.2], [14.3, 19.7], [15.0, 19.9]]
```

In [10]:
```
# 도미 학습 데이터 / 레이블
train_input = fish_data[:35]
train_target = fish_target[:35]

# 빙어 학습 데이터 / 레이블
test_input = fish_data[35:]
test_target = fish_target[35:]
```

In [12]:
```
kn = kn.fit(train_input, train_target)
print('빙어 데이터로 테스트',kn.score(test_input, test_target)) # Return the mean accuracy on the given test data
print('도미 데이터로 테스트',kn.score(train_input,train_target)) # It always returns 1.0
```

```
빙어 데이터로 테스트 0.0
도미 데이터로 테스트 1.0
```

넘파이

In [13]:
```
import numpy as np
```

In [14]:
```
input_arr = np.array(fish_data)
target_arr = np.array(fish_target)
print('input_arr',input_arr.shape)
print('target_arr',target_arr.shape)
```

```
input_arr (49, 2)
target_arr (49,)
```

In [15]:
```
print(input_arr)
```

```
[[  25.4  242. ]
 [  26.3  290. ]
 [  26.5  340. ]
 [  29.   363. ]
 [  29.   430. ]
 [  29.7  450. ]
 [  29.7  500. ]
 [  30.   390. ]
 [  30.   450. ]
 [  30.7  500. ]
 [  31.   475. ]
 [  31.   500. ]
 [  31.5  500. ]
 [  32.   340. ]
 [  32.   600. ]
 [  32.   600. ]
 [  33.   700. ]
 [  33.   700. ]
 [  33.5  610. ]
 [  33.5  650. ]
 [  34.   575. ]
 [  34.   685. ]
 [  34.5  620. ]
 [  35.   680. ]
 [  35.   700. ]
 [  35.   725. ]
 [  35.   720. ]
 [  36.   714. ]
 [  36.   850. ]
 [  37.  1000. ]
 [  38.5  920. ]
 [  38.5  955. ]
 [  39.5  925. ]
 [  41.   975. ]
 [  41.   950. ]
 [   9.8    6.7]
 [  10.5    7.5]
 [  10.6    7. ]
 [  11.     9.7]
 [  11.2    9.8]
 [  11.3    8.7]
 [  11.8   10. ]
 [  11.8    9.9]
 [  12.     9.8]
 [  12.2   12.2]
 [  12.4   13.4]
 [  13.    12.2]
 [  14.3   19.7]
 [  15.    19.9]]
```

In [16]:
```
print(input_arr.shape)
```

```
(49, 2)
```

In [18]:
```
np.random.seed(42)
index = np.arange(49) # 0~48 index mapping
print('index',index)
np.random.shuffle(index) # index shuffling : Shuffling sea bream and smelt data
```

```
index [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
 48]
```

In [19]:
```
print(index)
```

```
[13 45 47 44 17 27 26 25 31 19 12  4 34  8  3  6 40 41 46 15  9 16 24 33
 30  0 43 32  5 29 11 36  1 21  2 37 35 23 39 10 22 18 48 20  7 42 14 28
 38]
```

In [20]:
```
print(input_arr[[1,3]]) # print second and fourth data
```

```
[[ 26.3 290. ]
 [ 29.  363. ]]
```

In [21]:
```
train_input = input_arr[index[:35]]
print('train_input',train_input.shape)
train_target = target_arr[index[:35]]
print('train_target',train_target.shape)
```

```
train_input (35, 2)
train_target (35,)
```

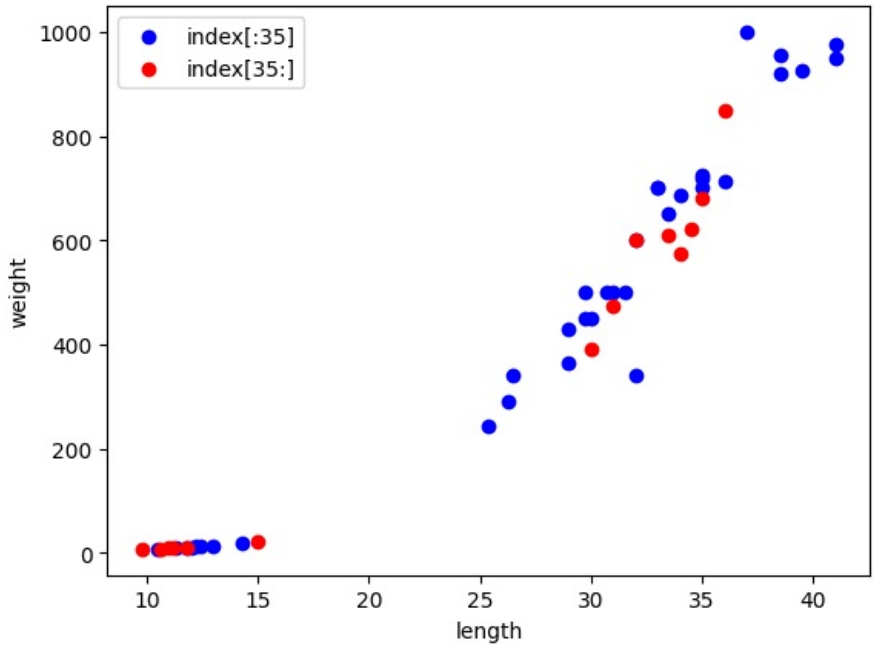
In [23]:
```
print(input_arr[13], train_input[0])
```

```
[ 32. 340.] [ 32. 340.]
```

In [24]:
```
test_input = input_arr[index[35:]]
test_target = target_arr[index[35:]]
```

In [35]:
```
import matplotlib.pyplot as plt
```

```
plt.scatter(train_input[:, 0], train_input[:, 1],c='blue',label='index[:35]') # 35
plt.scatter(test_input[:, 0], test_input[:, 1],c='red',label='index[35:]')   # 35
plt.xlabel('length')
plt.ylabel('weight')
plt.legend()
plt.show()
```



두 번째 머신러닝 프로그램

In [ ]:
```
print(train_input)
```

In [29]:
```
kn = kn.fit(train_input, train_target)
```

In [30]:
```
kn.score(test_input, test_target)
```

Out[30]:
```
1.0
```

In [36]:
```
y_hat=kn.predict(test_input)
```

In [37]:
```
test_target
```

Out[37]:
```
array([0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0])
```

In [38]:
```
y_hat == test_target
```

Out[38]:
```
array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True])
```