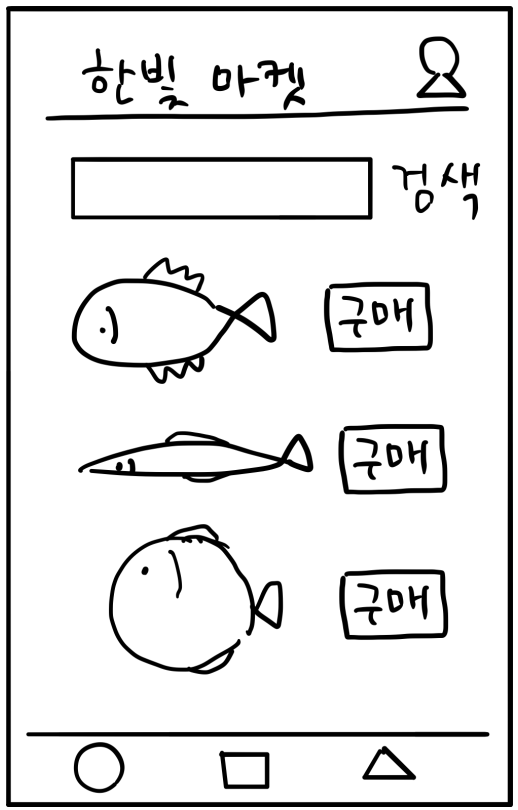
머신러닝 무작정 따라해보기

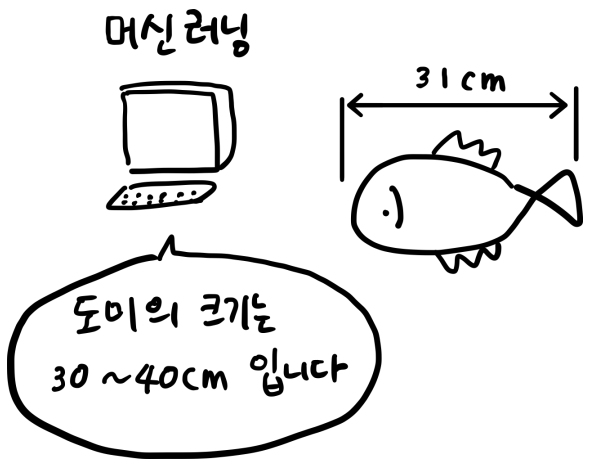
임 경 태

강의 진행방향

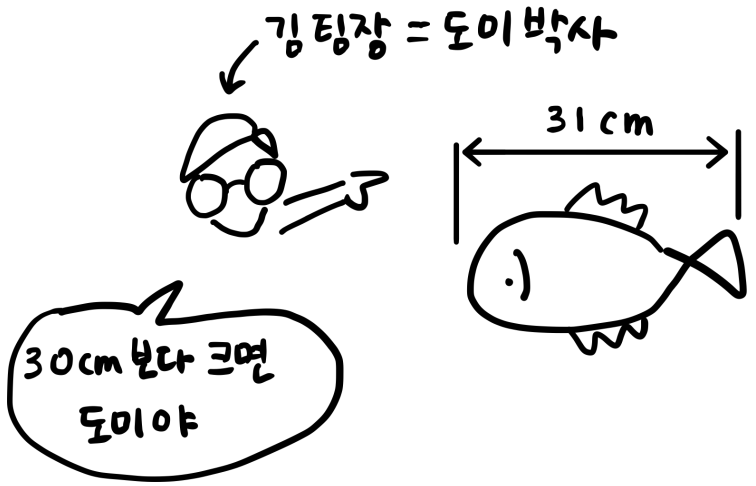
- 강의는 코딩 위주의 강의로 구성할 예정입니다.
- 머신러닝의 경우 난이도 편차가 커서 다음과 같이 투 트랙의 강의자료를 활용할 예정입니다.
    - 1. 최대한 주교재 위주로 쉽게 한번 설명해 드리고 해당 내용을 실습 (70%)
    - 2. 교재 내용보다 약간 어려운 이론을 추가로 설명 드리고 약간 더 어려운 코드를 실습 (30%)

첫 번째 머신러닝 프로그램





전통적인 프로그램




```
if fish_length >= 30:
    print("도미")
```

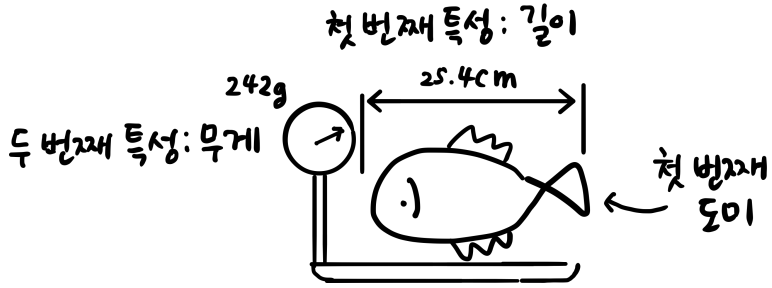
도미 vs 빙어

2개의 클래스(class)

분류(classification)

이진 분류(binary classification)

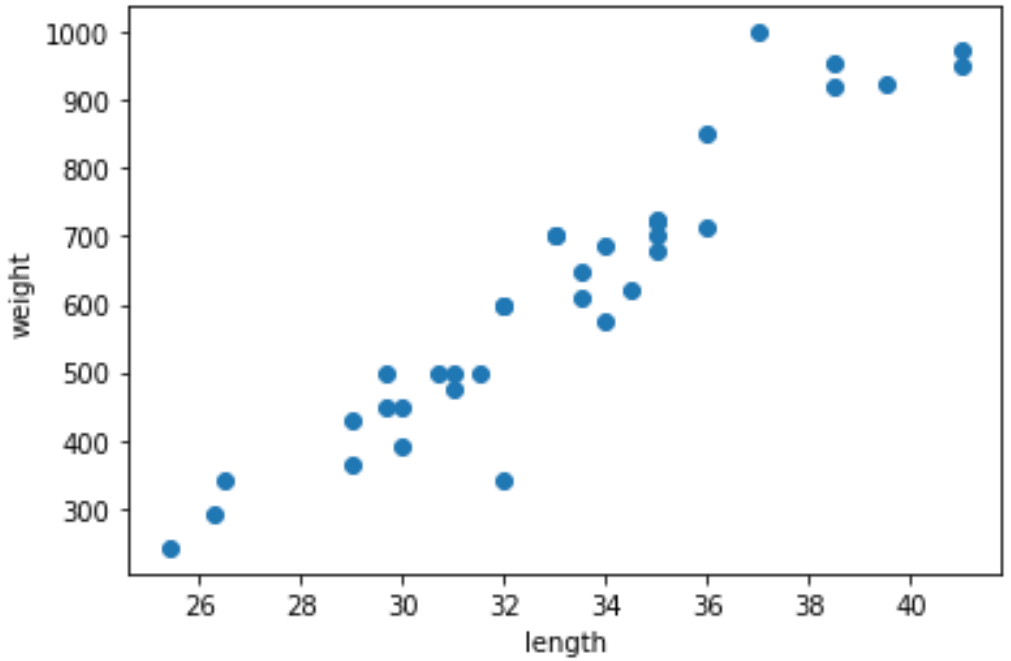
도미 데이터




```
bream_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]
bream_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]
```

산점도(scatter plot)

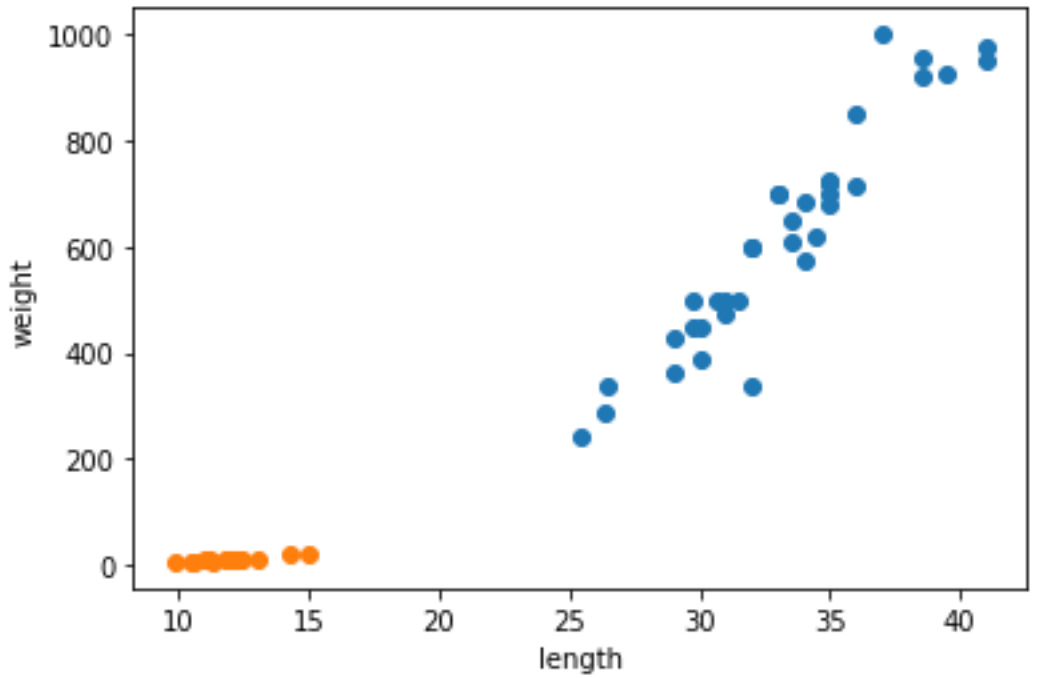
```
import matplotlib.pyplot as plt

plt.scatter(bream_length, bream_weight)
plt.xlabel('length')
plt.ylabel('weight')
plt.show()
```



빙어 데이터

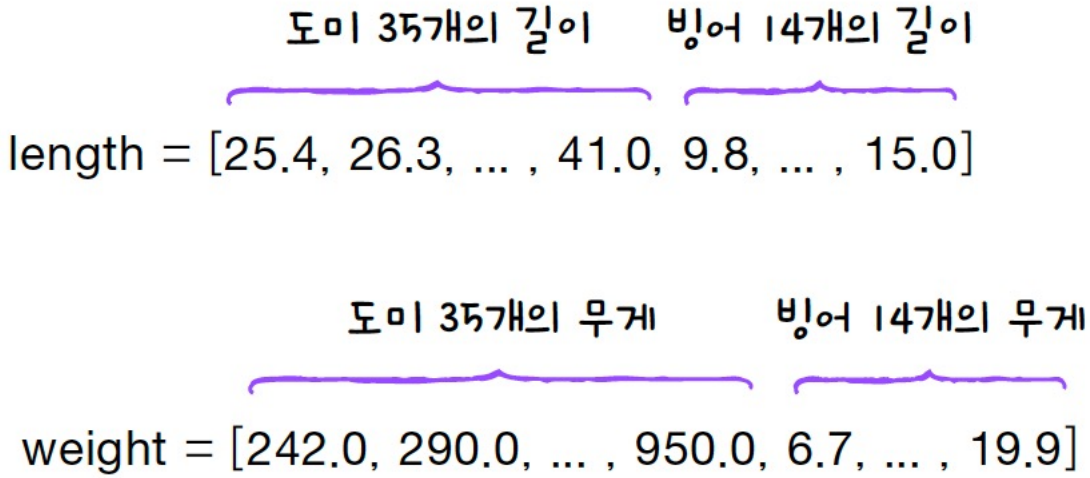
```
smelt_length = [9.8, 10.5, 10.6, 11.0, 11.2, 11.3, 11.8, 11.8, 12.0, 12.2,
                12.4, 13.0, 14.3, 15.0]
smelt_weight = [6.7, 7.5, 7.0, 9.7, 9.8, 8.7, 10.0, 9.9, 9.8, 12.2, 13.4,
                12.2, 19.7, 19.9]
```

```
plt.scatter(bream_length, bream_weight)
plt.scatter(smelt_length, smelt_weight)
plt.xlabel('length')
plt.ylabel('weight')
plt.show()
```

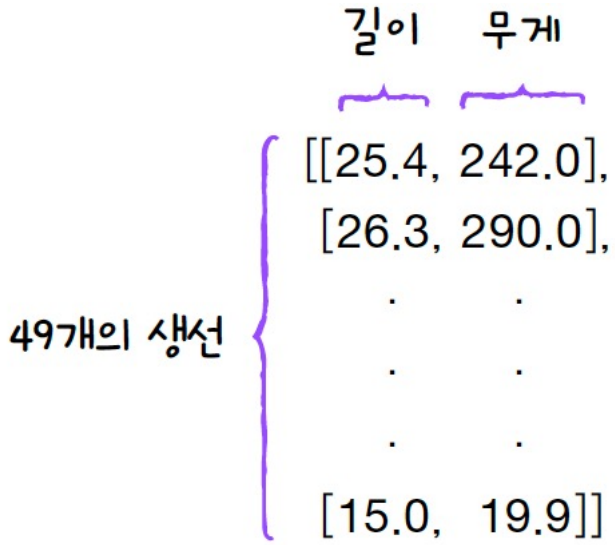


도미와 빙어 합치기

```
length = bream_length+smelt_length
weight = bream_weight+smelt_weight
```



사이킷런이 기대하는 데이터 형태



리스트 내포

```
fish_data = [[l, w] for l, w in zip(length, weight)]
```

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

정답 준비

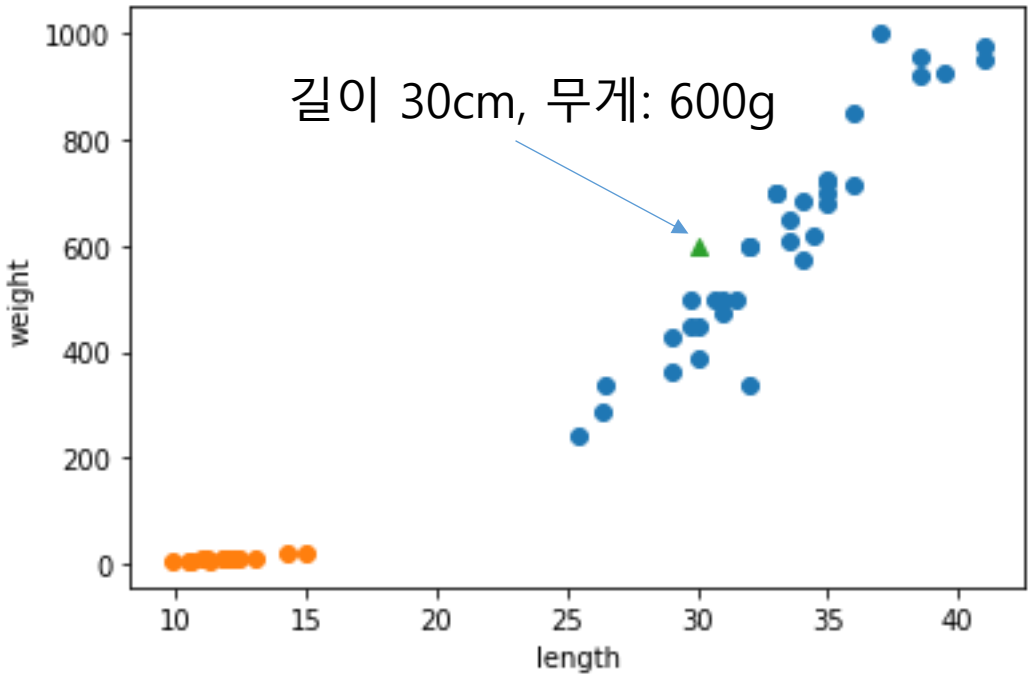
```
fish_target = [1]*35 + [0]*14
```

```
[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]
```

k-최근접 이웃

```
from sklearn.neighbors import KNeighborsClassifier

kn = KNeighborsClassifier()

kn.fit(fish_data, fish_target)

kn.score(fish_data, fish_target)
1.0
```

새로운 생선 예측




```
kn.predict([[30, 600]])

array([1])
```

무조건 도미

```
kn49 = KNeighborsClassifier(n_neighbors=49)

kn49.fit(fish_data, fish_target)
kn49.score(fish_data, fish_target)
```

```
0.7142857142857143
```

```
print(35/49)
```

```
0.7142857142857143
```