

✓ 샘플링 편향

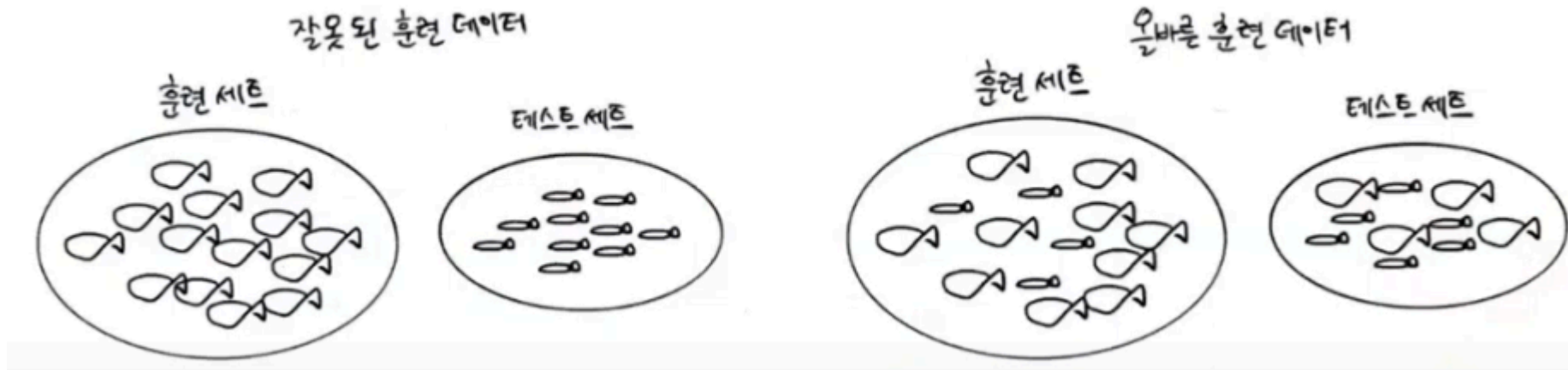
✓ 샘플링 편향이란 무엇인가요?

✓ 표본 추출 편향

- 매우 큰 샘플도 표본 추출 방법이 잘못되면 대표성을 띠지 못할 수 있음을 나타내는 현상

- 관심 모집단의 일부 구성원이 다른 구성원보다 샘플링 확률이 낮거나 높은 방식으로 샘플을 수집하는 경우 발생
- 이는 비무작위 샘플링 방법, 자기 선택 편향, 연구자 편향 등 다양한 이유로 발생할 수 있음

- 표본 편향은 전체 모집단을 대표하지 않을 수 있는 특정 특성이나 관점에 유리하도록 표본을 왜곡하여 연구 결과의 타당성과 일반화 가능성을 저해할 수 있음



```

1 #생선의 무게, 길이 데이터를 이용해 k-최근접 이웃 알고리즘으로 빙어와 도미를 분류
2 import matplotlib.pyplot as plt
3 import numpy as np
4 # 데이터 입력
5 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,
6                31.5, 32.0, 32.0, 32.0, 33.0, 33.0, 33.5, 33.5, 34.0, 34.0, 34.5, 35.0,
7                35.0, 35.0, 35.0, 36.0, 36.0, 37.0, 38.5, 38.5, 39.5, 41.0, 41.0, 9.8,
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]
9 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,
10               500.0, 340.0, 600.0, 600.0, 700.0, 700.0, 610.0, 650.0, 575.0, 685.0, 620.0, 680.0,
11               700.0, 725.0, 720.0, 714.0, 850.0, 1000.0, 920.0, 955.0, 925.0, 975.0, 950.0, 6.7,
12               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]
13
14 # zip 함수를 이용하여 두 리스트의 데이터 값에 대한 반복문을 진행
15 # 두 리스트의 데이터 값을 리스트로 만들어 하나의 리스트로 통합
16
17 # 분류 결과 데이터 생성
18 fish_data = [[l, w] for l, w in zip(fish_length, fish_weight)]
19 fish_target = [1]*35 + [0]*14
20

```

```

21 # 데이터 구조 확인
22 ## fish_data의 요소로 길이 값과 무게 값의 리스트를 가진다는 것을 확인
23 print(fish_data[4]) # [29.0, 430.0]
24
25 ## fish_data 0~4까지 데이터 호출
26 print(fish_data[0:5])
27
28 ### 시작 인덱스가 0일 경우 인덱스를 생략 가능
29 print(fish_data[:5])
30 ### 종료 인덱스가 데이터의 마지막일 경우 생략 가능
31 print(fish_data[44:])
32
33 # 훈련 세트로 입력값 중 0부터 34번째 인덱스까지 사용
34 train_input = fish_data[:35]
35 # 훈련 세트로 타깃값 중 0부터 34번째 인덱스까지 사용
36 train_target = fish_target[:35]
37 # 테스트 세트로 입력값 중 35번째부터 마지막 인덱스까지 사용
38 test_input = fish_data[35:]
39 # 테스트 세트로 타깃값 중 35번째부터 마지막 인덱스까지 사용
40 test_target = fish_target[35:]
41

```

```

↔ [29.0, 430.0]
[[25.4, 242.0], [26.3, 290.0], [26.5, 340.0], [29.0, 363.0], [29.0, 430.0]]
[[25.4, 242.0], [26.3, 290.0], [26.5, 340.0], [29.0, 363.0], [29.0, 430.0]]
[[12.2, 12.2], [12.4, 13.4], [13.0, 12.2], [14.3, 19.7], [15.0, 19.9]]

```


```

1 # 사이킷런의 KNeighborsClassifier(K - 최근접 이웃)의 객체 생성
2
3 from sklearn.neighbors import KNeighborsClassifier
4 kn= KNeighborsClassifier()
5
6 # 모델링 및 모델 평가
7 kn= kn.fit(train_input, train_target)
8 kn.score(test_input, test_target) # 0.0
9 #샘플링 편향 : train을 진행할 때 훈련 데이터로 도미의 데이터만 사용되었고, test를 진행할 때는 빙어의 데이터만 사용되어서 평가 점수가 0점이 니

```

 0.0

```
1 # 데이터형 numpy array로 변경
2
3 input_arr = np.array(fish_data)
4 target_arr = np.array(fish_target)
5
6 print(input_arr.shape) # 이 명령을 사용하면 (샘플 수, 특성 수)를 출력함, (49, 2)
7
8 np.random.seed(42) # random으로 출력되는 수를 동일하게 함
9 index = np.arange(49) # 0부터 48까지 49개의 수를 출력
10 np.random.shuffle(index) # index의 원소들을 섞음
11
12 # 섞은 index list에서 0~34번째 수를 뽑고 input_arr에서 뽑힌 수에 해당하는 원소를 train_input에 할당
13 train_input = input_arr[index[:35]]
14
15 # 섞은 index list에서 0~34번째 수를 뽑고 target_arr에서 뽑힌 수에 해당하는 원소를 train_target에 할당
16 train_target = target_arr[index[:35]]
17
18 # 섞은 index list에서 35~마지막째 수를 뽑고 input_arr에서 뽑힌 수에 해당하는 원소를 test_input에 할당
19 test_input = input_arr[index[35:]]
20
21 # 섞은 index list에서 35~마지막째 수를 뽑고 target_arr에서 뽑힌 수에 해당하는 원소를 test_target에 할당
22 test_target = target_arr[index[35:]]
23
24 kn= kn.fit(train_input, train_target)
25 kn.score(test_input, test_target)
```

 (49, 2)
1.0

```
1 #https://aquaraga.github.io/python/data-analysis/pandas/2017/05/25/sampling-bias.html
2 import pandas as pd
```

```

3 import numpy as np
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 import warnings
7 warnings.filterwarnings('ignore')
8
9 %matplotlib notebook
10
11 female_heights = np.random.normal(loc = 151.9, scale = 6, size = 16000)
12 male_heights = np.random.normal(loc = 164.9, scale = 7, size = 4000)
13
14 all_heights = np.append(female_heights, male_heights)
15 all_genders = ['F'] * 16000 + ['M'] * 4000
16
17 df = pd.DataFrame({'Gender': all_genders, 'Height': all_heights})
18 df.head()

```



	Gender	Height
0	F	151.058689
1	F	151.700840
2	F	147.405541
3	F	147.229708
4	F	157.593057

```

1 plt.figure()
2 sns.distplot(pd.Series(female_heights, name = "Height(cm)"), hist = False, label = "Females",
3               color='pink', kde_kws={'linestyle': 'dotted'})
4 sns.distplot(pd.Series(male_heights, name = "Height(cm)"), hist = False, label = "Males",
5               color = 'blue', kde_kws={'linestyle': 'dotted'})
6 sns.distplot(all_heights, hist = False, label = "All",
7               color = 'green')
8
9 plt.gca().set_ylabel('Density(scaled)')

```

```

10 plt.gca().set_title('Distribution of heights - with sampling bias')
11 plt.legend()
12 plt.show()

```



```

1 def set_weight(row):
2     if row['Gender'] == 'F':
3         row['Weight'] = 0.25
4     else:
5         row['Weight'] = 1
6     return row
7
8 df_with_weights = df.apply(set_weight, axis = 1)
9 df_with_weights.head()

```



	Gender	Height	Weight
0	F	151.058689	0.25
1	F	151.700840	0.25
2	F	147.405541	0.25
3	F	147.229708	0.25
4	F	157.593057	0.25

```

1 sample = df_with_weights.sample(n=2000, weights='Weight')
2 print('Number of males: ', sample[sample.Gender == 'M'].shape[0])
3 print('Number of females: ', sample[sample.Gender == 'F'].shape[0])

```



```

Number of males: 975
Number of females: 1025

```

```

1 plt.figure()
2 sns.distplot(pd.Series(sample[sample.Gender == 'F'].Height, name = "Height(cm)"), hist = False, label = "Females",

```

```
3         color='pink', kde_kws={'linestyle': 'dotted'})
4 sns.distplot(pd.Series(sample[sample.Gender == 'M'].Height, name = "Height(cm)"), hist = False, label = "Males",
5             color = 'blue', kde_kws={'linestyle': 'dotted'})
6 sns.distplot(sample.Height, hist = False, label = "All",
7             color = 'green')
8
9 plt.gca().set_xlabel('Height(cm)')
10 plt.gca().set_ylabel('Density(scaled)')
11 plt.gca().set_title('Distribution of heights - with sample correction')
12 plt.legend()
13 plt.show()
```



✓ <<<참조자료 사이트>>>

- 1.[AI 훈련용 빅데이터 2026년 고갈...문제점과 대책은](#)
- 2.["학습 데이터 부족" AI개발에 닥친 난관... 차세대 모델 개발 지연](#)
- 3.[AI 데이터 고갈 위기-"2년 후 AI 성장 멈출 수도"...데이터 절벽 '경고'](#)
- 4.[Undersampling과 Oversampling이란?](#)
- 5.[불균형 데이터\(imbalanced data\) 처리를 위한 샘플링 기법](#)
- 6.[샘플링 편향이라는 문제](#)
- 7.[데이터 품질\(Data Quality\)이란?](#)
- 8.[데이터에 제값 매기는 데이터 품질 관리](#)
- 9.[과대적합\(overfitting\) 및 과소적합\(underfitting\) 개념\(+Early Stopping\)](#)
- 10.[Sampling bias](#)

