데이터 분석 기초
Fundamental of data analysis

Exercise 13

1 1차 함수 분포를 갖는 데이터 학습

- y = 3x+7에 대한 데이터가 주어졌습니다.

학습 데이터

x	1	0	-5	10	-20	42	-16	-9	8
y	10	7	-8	37	-53	133	-41	-20	31

- 위 데이터에 대해 선형회귀를 학습해보고 모델이 잘 학습되었는지 확인하기 위해 평가데이터를 사용한 예측값, 모델의 기울기, 절편을 출력해봅니다.

평가 데이터 예시

x	2	3	4	20	-1

1 1차 함수 분포를 갖는 데이터 학습

- 왜 train과 test set의 데이터가 다른가요?

  -> test에 train과 완전히 일치하는 데이터가 있을 시이미 학습에 사용되었기 때문에 당연히 그 문제의 답을 모델은 알고 있습니다. 연습문제가 그대로 시험에 나온 것과 같은 의미 입니다.

1 1차 함수 분포를 갖는 데이터 학습

- 평가 데이터에 대한 모델의 예측 값을 출력하고 학습된 모델의 기울기와 절편을 출력해봅시다.

입출력 예시

입력값	출력값
2 3 4 20 -10	[13. 16. 19. 67. -23.] [3.] 6.999999999999998

2 2차 함수 분포를 갖는 데이터 학습

- y = x^2+2에 대한 데이터가 주어졌습니다.

학습 데이터

x	1	0	-2	3	-4	5	-6	7	-8
y	3	2	6	11	18	27	38	51	66

- 위 데이터에 대해 선형회귀를 학습해보고 모델이 잘 학습되었는지 모델의 기울기와 절편, 평가데이터를 사용하여 예측값을 출력해봅니다.

평가 데이터 예시

x	2	3	4	20	-1
y	6	11	18	402	3

2 2차 함수 분포를 갖는 데이터 학습

- 1. 평가 데이터에 대한 모델의 예측 값을 출력하고 2. 학습된 모델의 기울기와 절편, 그리고 3. 입력받은 test set에 대한 score를 출력해봅시다.

입출력 예시

입력값	출력값
2 3 4 20 -10 6 11 18 402 102	[22.08791209 21.03296703 19.97802198 3.0989011 34.74725275] [-1.05494505] 24.1978021978022 -0.4338304794983663

Test set

Test label(answer)

2 2차 함수 분포를 갖는 데이터 학습

- 왜 1번에서 학습한 결과와 너무도 다를까?

  저희가 이번 시간에 배우는 모델은 선형 회귀 모델입니다. 즉 모델은 선형적으로 데이터의 분포를 학습하고 예측합니다. 따라서 이와 같은 2차 함수의 분포를 가진 데이터는 선형 회귀 모델의 학습을 하기에는 적절치 않습니다.

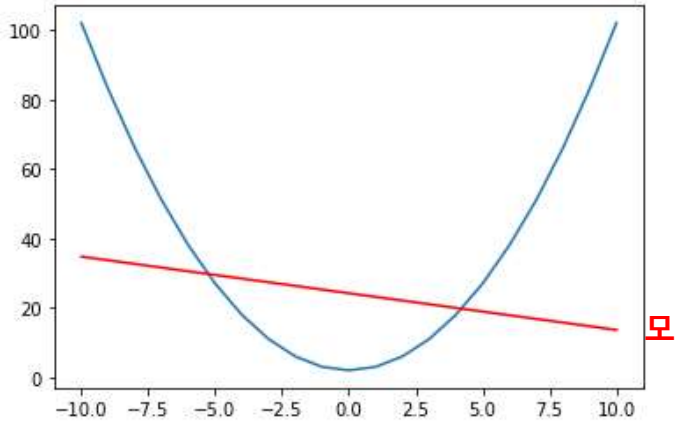
2 Colab – 데이터 분포/ 모델 시각화

```
1 from sklearn.linear_model import LinearRegression
2 import matplotlib.pyplot as plt
3 import numpy as np
4 x = np.array(range(-10, 11)) #x영역 -10~11까지 정의
5
```

모델 학습 코드

```
14
15 plt.plot(x, x**2+2)
16 plt.plot(x, lr.coef_*x+lr.intercept_ , 'r') #기울기와 절편을 이용한 모델 plot
17 plt.show()
```

데이터 분포



모델 예측

3 선형 회귀의 변수 exercise

- 이론 강의 자료의 성별, 키, 몸무게 데이터로 선형 회귀 모델을 학습해봅니다. 학습할 회귀 모델은 2가지 입니다.
- 1. 키, 성별 모두 학습 데이터로 쓴 모델
- 2. 키만 학습 데이터로 쓴 모델

학습 데이터

```
X = [ [171,0], [169,0], [176,0], [168,0], [181,0], [166,0], [180,0], [175,0],
      [163,1], [162,1], [171,1], [162,1], [164,1], [162,1], [158,1], [173,1] ]

Y = [69, 65, 72, 67, 71, 65, 80, 71, 55, 51, 59, 53, 61, 56, 47, 57]
```

3 선형 회귀의 변수 exercise

- 다중 선형 회귀 모델, 단일 선형 회귀 모델의 train data에 대한 점수를 score()함수로 순서대로 출력하고, input으로 입력된 입력 데이터에 대한 다중 선형 회귀 모델과 단일 선형 회귀 모델의 예측값을 순서대로 각각 출력해주세요.

3 선형 회귀의 변수 exercise

입출력 예시

입력값	출력값
178 0	0.8906355513939724 0.7339009423725997 [73.136285] [72.74532854]

입력값	출력값
157 1	0.8906355513939724 0.7339009423725997 [50.00550488] [49.18457759]

4 Boston housing 데이터 선형 회귀

- Boston housing 데이터를 가지고 선형 회귀 학습을 진행해 볼 것입니다.
- 단 exercise 3번과 마찬가지로 학습에 사용되는 변수의 개수를 조절해가며 학습된 모델을 비교해 볼 것입니다.

4 Boston housing 데이터 선형 회귀

- 모델은 3개를 학습해볼 것입니다.
- 1. 모든 변수를 사용해 학습한 모델
- 2. 선형 회귀 모델의 계수의 절대값이 낮은 변수 5개를 제외하고 학습한 모델
- 3. 선형 회귀 모델을 잘 학습 시킬 수 있는 데이터 변수 몇가지만 뽑아내서 학습한 모델

- 위 모델들을 비교하여 어떤 변수를 써야 모델을 학습을 잘 시킬 수 있는지 알아봅시다.

4 Boston housing 데이터 선형 회귀

- 구체적인 실습 4의 풀이 순서는 다음과 같습니다.
- 1. 데이터셋 준비(train, test set 분리) test : 앞의 100개, train : 나머지
- 2. 모든 변수를 사용한 모델 학습
- 3. 선형 회귀 모델의 변수 별 모델 계수 확인.
- 4. 데이터셋2, 3 준비(2 – 낮은 절대값 모델 계수 drop, 3 – 선형 회귀 모델에서 데이터를 잘 설명하는 변수 set)
- 5. 데이터셋 2, 3으로 각각의 모델 학습
- 6. 모든 모델의 score 확인

4 Boston housing 데이터 선형 회귀

- 3. 선형 회귀 모델을 잘 학습 시킬 수 있는 데이터 변수 몇가지만 뽑아내서 학습 시 변수는 어떻게 아나요?

- 변수를 골라내는 데에는 여러가지 방법이 있습니다. 전진 선택법, 후진 선택법, 단계별 선택법 등이 있지만 이번 시간에 여러분이 도전하기에는 무리가 있기 때문에 그러한 방법을 써서 골라낸 변수를 여러분께 제공(해설 때)하고 여러분은 학습하는 변수가 달라졌을 시 어떻게 변화하는 지를 이해해주시면 됩니다.

4 Boston housing 데이터 선형 회귀

- 2번 모델은 가장 모델 계수의 절대값이 낮은 변수인 'AGE', 'B', 'TAX', 'INDUS', 'ZN' 을 제외하고 학습해봅시다.

모델 계수

RM	3282.285404
CHAS	2531.364351
RAD	318.774694
ZN	58.899630
INDUS	53.214311
B	9.287794
AGE	-4.043961
TAX	-12.518089
CRIM	-109.124839
LSTAT	-581.756268
PTRATIO	-1011.862186
DIS	-1783.297687
NOX	-21159.335829
dtype: float64	

4 Boston housing 데이터 선형 회귀

- (도전!) 3번 모델은 두 개의 변수를 골라 제외하고 학습을 시키면서 test set score에 대해 성능이 좋은 모델을 찾아봅니다! -> 변수 제거 정답은 해설강의에서 공개

4 Boston housing 데이터 선형 회귀

입출력 예시

```
> RM          3282.285404
CHAS          2531.364351
RAD            318.774694
ZN              58.899630
INDUS           53.214311
B                9.287794
AGE             -4.043961
TAX            -12.518089
CRIM           -109.124839
LSTAT          -581.756268
PTRATIO       -1011.862186
DIS           -1783.297687
NOX          -21159.335829
dtype: float64
0.7465346879380911
0.6377771855934954

0.7276877334642398
0.6429009228353517

0.7217510503099396
0.7058589152434713
```

모델 1의 train / test score

모델 2의 train / test score

모델 3의 train / test score

Train set과 test set 에 대한 score가 다른 값을 가진 것을 볼 수 있습니다. 이것을 train set에 대해 과적합 되었다 라고 합니다.

모델 2는 train set에 대한 score는 모델 1에 비해 낮지만 test score는 모델 1 보다 높은 것을 볼 수 있습니다.

모델 3은 train set에 대한 score가, test set에 대한 Score가 비슷하므로 test set도 train set과 비슷하게 예측할 수 있는 모델입니다.

4 Boston housing 데이터 선형 회귀

입출력 예시

```
> RM          3282.285404
CHAS          2531.364351
RAD            318.774694
ZN              58.899630
INDUS           53.214311
B                9.287794
AGE             -4.043961
TAX            -12.518089
CRIM          -109.124839
LSTAT         -581.756268
PTRATIO      -1011.862186
DIS          -1783.297687
NOX         -21159.335829
dtype: float64
0.7465346879380911
0.6377771855934954

0.7276877334642398
0.6429009228353517

0.7217510503099396
0.7058589152434713
```

모델 1의 train / test score

모델 2의 train / test score

모델 3의 train / test score

이렇게 저희는 변수를 조절하여 학습함으로
모든 변수를 사용한 모델1 보다 과적합이 덜 된
모델2. 모델2 보다 과적합이 덜 된
모델 3을 학습해 보았습니다.

Train으로 학습 후 test로 검증하는 것은 매우 중요한
과정이며 test와 train의 데이터의 분포가 다를 수 있으므로
꼭 확인해봐야 하는 과정입니다.

Thank you