

## 지도학습 (1) - 회귀

## #01. 작업준비

패키지 가져오기

## #02. 단순 선형 회귀

데이터 가져오기

분석 알고리즘을 포함하고 있는 객체 생성

분석모델을 학습시킴

학습 결과 확인

직선의 기울기

직선의 절편

회귀식 추정

설명력

회귀분석 결과 활용

모델을 활용한 결과값 얻기

## #03. 다중 선형 회귀

데이터 가져오기

데이터 전처리

독립변수 추출

종속변수 추출

# 지도학습 (1) - 회귀

지도 학습은 크게 회귀 와 분류 로 나눌 수 있다.

필요한 용어 정리...

## #01. 작업준비

### 패키지 가져오기

```
from pandas import read_excel # 엑셀 데이터를 가져오기
from sklearn.linear_model import LinearRegression # 선형회귀분석 모듈
```

## #02. 단순 선형 회귀

### 데이터 가져오기

자동차의 속도(speed)에 따른 제동거리(dist) 조사 데이터

```
origin = read_excel("https://data.hossam.kr/E04/cars.xlsx")
origin.head()
```

## 지도학습 (1) - 회귀

## #01. 작업준비

패키지 가져오기

## #02. 단순 선형 회귀

데이터 가져오기

분석 알고리즘을 포함하고 있는 객체 생성

분석모델을 학습시킴

학습 결과 확인

직선의 기울기

직선의 절편

회귀식 추정

설명력

회귀분석 결과 활용

모델을 활용한 결과값 얻기

## #03. 다중 선형 회귀

데이터 가져오기

데이터 전처리

독립변수 추출

종속변수 추출

	speed	dist
0	4	2
1	4	10
2	7	4
3	7	22
4	8	16

## 분석 알고리즘을 포함하고 있는 객체 생성

```
model = LinearRegression()
```

## 분석모델을 학습시킴

```
fit(독립변수, 종속변수) 형태로 사용
```

직접 데이터를 파라미터로 전달할 경우

1. 2차원 배열(numpy)설정
2. 완전한 데이터 프레임 형태로 전달

```
fit = model.fit(origin[['speed']], origin[['dist']])
fit
```

## 지도학습 (1) - 회귀

## #01. 작업준비

패키지 가져오기

## #02. 단순 선형 회귀

데이터 가져오기

분석 알고리즘을 포함하고 있는 객체 생성

분석모델을 학습시킴

학습 결과 확인

직선의 기울기

직선의 절편

회귀식 추정

설명력

회귀분석 결과 활용

모델을 활용한 결과값 얻기

## #03. 다중 선형 회귀

데이터 가져오기

데이터 전처리

독립변수 추출

종속변수 추출

▼ LinearRegression

LinearRegression()

## 학습 결과 확인

## 직선의 기울기

```
coef = fit.coef_
coef
```

```
array([[3.93240876]])
```

```
coef[0][0]
```

```
3.932408759124089
```

## 직선의 절편

```
intercept = fit.intercept_
intercept
```

```
array([-17.57909489])
```

## 지도학습 (1) - 회귀

## #01. 작업준비

패키지 가져오기

## #02. 단순 선형 회귀

데이터 가져오기

분석 알고리즘을 포함하고 있는 객체 생성

분석모델을 학습시킴

학습 결과 확인

직선의 기울기

직선의 절편

회귀식 추정

설명력

회귀분석 결과 활용

모델을 활용한 결과값 얻기

## #03. 다중 선형 회귀

데이터 가져오기

데이터 전처리

독립변수 추출

종속변수 추출

intercept[0]

-17.579094890510973

## 회귀식 추정

```
print("y = {0:.2f} * X + {1:.2f}".format(coef[0][0], intercept[0]))
```

$$y = 3.93 * X + -17.58$$

## 설명력

```
rscore = fit.score(origin[['speed']], origin[['dist']])
rscore
```

0.6510793807582509

```
print("설명력: {:.1%}".format(rscore))
```

설명력: 65.1%

## 지도학습 (1) - 회귀

## #01. 작업준비

패키지 가져오기

## #02. 단순 선형 회귀

데이터 가져오기

분석 알고리즘을 포함하고 있는 객체 생성

분석모델을 학습시킴

학습 결과 확인

직선의 기울기

직선의 절편

회귀식 추정

설명력

회귀분석 결과 활용

모델을 활용한 결과값 얻기

## #03. 다중 선형 회귀

데이터 가져오기

데이터 전처리

독립변수 추출

종속변수 추출

## 회귀분석 결과 활용

## 모델을 활용한 결과값 얻기

```
speed = [[10], [15], [20], [25], [30], [35], [40]]
pred = fit.predict(speed)
pred
```

```
c:\Users\leekh\AppData\Local\Programs\Python\Python311\Lib\site-packages
warnings.warn(
```

```
array([[ 21.7449927 ],
       [ 41.4070365 ],
       [ 61.06908029],
       [ 80.73112409],
       [100.39316788],
       [120.05521168],
       [139.71725547]])
```

## #03. 다중 선형 회귀

## 데이터 가져오기

## 지도학습 (1) - 회귀

## #01. 작업준비

패키지 가져오기

## #02. 단순 선형 회귀

데이터 가져오기

분석 알고리즘을 포함하고 있는 객체 생성

분석모델을 학습시킴

학습 결과 확인

직선의 기울기

직선의 절편

회귀식 추정

설명력

회귀분석 결과 활용

모델을 활용한 결과값 얻기

## #03. 다중 선형 회귀

데이터 가져오기

데이터 전처리

독립변수 추출

종속변수 추출

```
origin = read_excel("https://data.hossam.kr/E04/boston.xlsx")
origin
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222
...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273

506 rows × 15 columns

## 데이터 전처리

## 지도학습 (1) - 회귀

## #01. 작업준비

패키지 가져오기

## #02. 단순 선형 회귀

데이터 가져오기

분석 알고리즘을 포함하고 있는 객체 생성

분석모델을 학습시킴

학습 결과 확인

직선의 기울기

직선의 절편

회귀식 추정

설명력

회귀분석 결과 활용

모델을 활용한 결과값 얻기

## #03. 다중 선형 회귀

데이터 가져오기

데이터 전처리

독립변수 추출

종속변수 추출

머신러닝에서 데이터 전처리라 함은 독립변수와 종속변수를 분리하는 과정을 의미

## 독립변수 추출

```
x_train = origin[["CRIM", "INDUS", "CHAS", "NOX", "RM", "AGE", "DIS", "TAX", "PTRATIO", "MEDV"]]
x_train.head()
```

	CRIM	INDUS	CHAS	NOX	RM	AGE	DIS	TAX	PTRATIO	MEDV
0	0.00632	2.31	0	0.538	6.575	65.2	4.0900	296	15.3	396.91
1	0.02731	7.07	0	0.469	6.421	78.9	4.9671	242	17.8	396.91
2	0.02729	7.07	0	0.469	7.185	61.1	4.9671	242	17.8	392.92
3	0.03237	2.18	0	0.458	6.998	45.8	6.0622	222	18.7	394.92
4	0.06905	2.18	0	0.458	7.147	54.2	6.0622	222	18.7	396.92

## 종속변수 추출

```
y_train = origin[["MEDV"]]
y_train.head()
```

	MEDV
0	24.0
1	21.6

## 지도학습 (1) - 회귀

## #01. 작업준비

패키지 가져오기

## #02. 단순 선형 회귀

데이터 가져오기

분석 알고리즘을 포함하고 있는 객체 생성

분석모델을 학습시킴

학습 결과 확인

직선의 기울기

직선의 절편

회귀식 추정

설명력

회귀분석 결과 활용

모델을 활용한 결과값 얻기

## #03. 다중 선형 회귀

데이터 가져오기

데이터 전처리

독립변수 추출

종속변수 추출

	MEDV
2	34.7
3	33.4
4	36.2

## 분석 수행

```

model = LinearRegression()
fit = model.fit(x_train, y_train)

print("계수: ", fit.coef_)
print("절편: ", fit.intercept_)
print("설명력: ", fit.score(x_train, y_train))

print("y = {0:.2f} * X + {1:.2f}".format(coef[0][0], intercept[0]))

```

```

계수:  [[ 9.01317705e-16  8.81239526e-16  2.54951835e-16  3.65393412e-15
        7.53940638e-16  7.49704118e-15  3.83970199e-16 -6.62664368e-16
        4.71898996e-17  1.50920942e-16  1.24076097e-15  1.00000000e+00]]
절편:  [-3.30402372e-13]
설명력:  1.0
y = 3.93 * X + -17.58

```