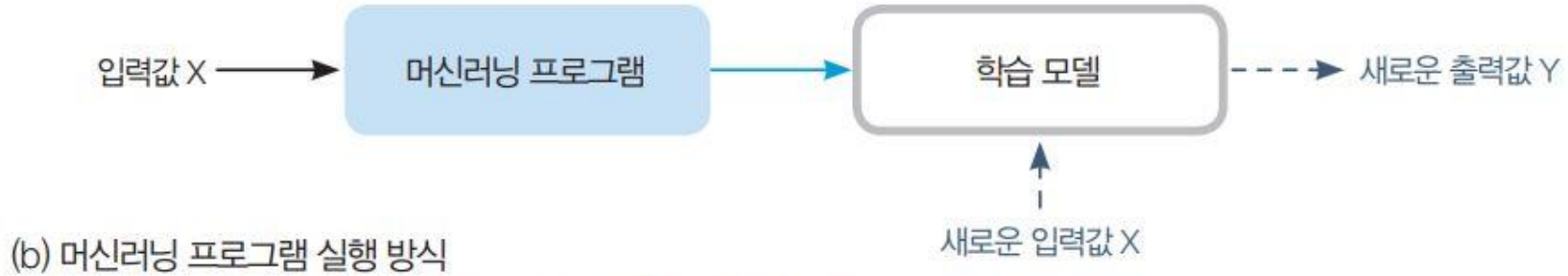
회귀 분석 I

빅데이터 분석

머신러닝

(a) 일반 프로그램 실행 방식



(b) 머신러닝 프로그램 실행 방식

그림 10-1 일반 프로그램과 머신러닝 프로그램 실행 방식 비교

< 머신러닝 프로세스 >

데이터 수집 → 데이터 전처리 및 훈련/테스트 데이터 분할 → 모델 구축 및 학습 → 모델 평가 → 예측

지도 학습 (Supervised Learning)

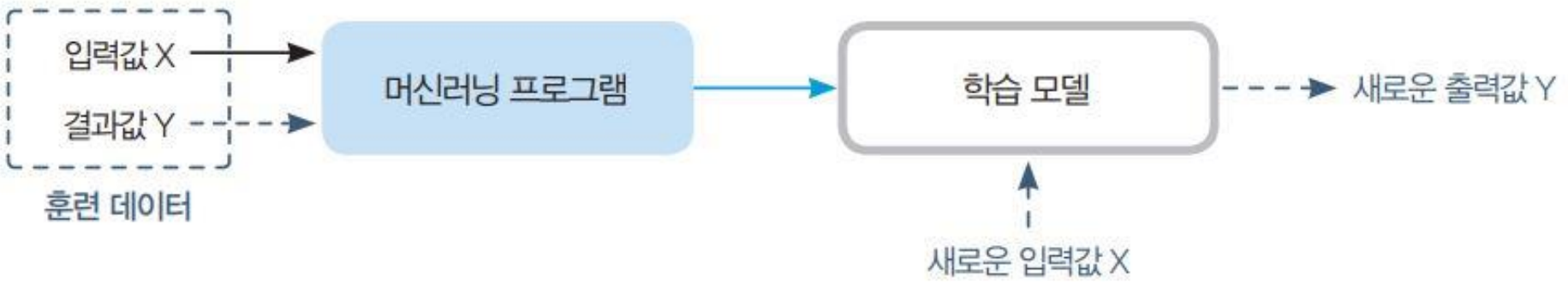


그림 10-2 머신러닝의 지도 학습 방식

분석 평가 지표

표 10-1 회귀 분석 결과에 대한 평가 지표

평가 지표	수식	사이킷런 라이브러리
MAE: Mean Absolute Error	$\frac{1}{n}\sum_{i=1}^{n}\lvert Y_i-\hat{Y}_i\rvert$	metrics.mean_absolute_error()
MSE: Mean Squared Error	$\frac{1}{n}\sum_{i=1}^{n}(Y_i-\hat{Y}_i)^2$	metrics.mean_squared_error()
RMSE: Root Mean Squared Error	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i-\hat{Y}_i)^2}$	없음
R^2: Variance score, 결정 계수coefficient of determination	$\frac{\text{예측값의 분산}}{\text{실제값의 분산}}$	metrics.r2_score()

데이터 수집, 준비 및 탐색

1. 데이터 수집

```
C:\> pip install sklearn
```

```
>>> import numpy as np
>>> import pandas as pd

>>> from sklearn.datasets import load_boston
>>> boston = load_boston()
```

2. 데이터가 이미 정리된 상태이므로 데이터셋 구성을 확인

```
>>> print(boston.DESCR)
>>> boston_df = pd.DataFrame(boston.data, columns=boston.feature_names)
>>> boston_df.head()
>>> boston_df['PRICE'] = boston.target
>>> boston_df.head()
>>> print('보스톤 주택 가격 데이터셋 크기: ', boston_df.shape)
>>> boston_df.info()
```

데이터 수집, 준비 및 탐색 (cont'd)

- CRIM : 지역별 범죄 발생률
- ZN : 25,000평방피트를 초과하는 거주 지역 비율
- INDUS : 비상업 지역의 넓이 비율
- CHAS : 찰스강의 더미변수(1은 강의 경계, 0은 경계 아님)
- NOX : 일산화질소 농도
- RM : 거주할 수 있는 방 개수
- AGE : 1940년 이전에 건축된 주택 비율
- DIS : 5개 주요 고용센터까지 가중 거리
- RAD : 고속도로 접근 용이도
- TAX : 10,000달러당 재산세 비율
- PTRATIO : 지역의 교사와 학생 수 비율
- B : 지역의 흑인 거주 비율
- LSTAT : 하위 계층의 비율
- PRICE : 본인 소유 주택 가격

분석 모델 구축, 결과 분석 및 시각화

```
>>> from sklearn.linear_model import LinearRegression
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.metrics import mean_squared_error, r2_score

#X, Y 분할하기
>>> Y = boston_df['PRICE']
>>> X = boston_df.drop(['PRICE'], axis = 1, inplace = False)

#훈련용 데이터와 평가용 데이터 분할하기
>>> X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3,
random_state = 156)

#선형 회귀 분석 : 모델 생성
>>> lr = LinearRegression()

#선형 회귀 분석 : 모델 훈련
>>> lr.fit(X_train, Y_train)

#선형 회귀 분석 : 평가 데이터에 대한 예측 수행 -> 예측 결과 Y_predict 구하기
>>> Y_predict = lr.predict(X_test)
```

분석 모델 구축, 결과 분석 및 시각화 (cont'd)

2. 선형 회귀 분석 모델을 평가 지표를 통해 평가

```
>>> mse = mean_squared_error(Y_test, Y_predict)
>>> rmse = np.sqrt(mse)
>>> print('MSE : {0:.3f}, RMSE : {1:.3f}'.format(mse, rmse))
>>> print('R^2(Variance score) : {0:.3f}'.format(r2_score(Y_test, Y_predict)))

>>> print('Y 절편 값: ', lr.intercept_)
>>> print('회귀 계수 값: ', np.round(lr.coef_, 1))

>>> coef = pd.Series(data = np.round(lr.coef_, 2), index = X.columns)
>>> coef.sort_values(ascending = False)
```

< 회귀 모델 결과를 토대로 보스톤 주택 가격에 대한 회귀식 >

$$Y_{PRICE} = -0.11X_{CRIM} + 0.07X_{ZN} + 0.03X_{INDUS} + 3.05X_{CHAS} - 19.80X_{NOX} + 3.35X_{RM} + 0.01X_{AGE} - 1.74X_{DIS} + 0.36X_{RAD} - 0.01X_{TAX} - 0.92X_{PTRATIO} + 0.01X_{B} - 0.57X_{LSTAT} + 41.00$$

분석 모델 구축, 결과 분석 및 시각화 (cont'd)

3. 회귀 분석 결과 시각화하기

```
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns

>>> fig, axs = plt.subplots(figsize = (16, 16), ncols = 3, nrows = 5)

>>> x_features = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
'TAX', 'PTRATIO', 'B', 'LSTAT']

>>> for i, feature in enumerate(x_features):
      row = int(i/3)
      col = i%3
      sns.regplot(x = feature, y = 'PRICE', data = boston_df, ax = axs[row][col])
```