머신러닝
(Machine Learning)

LECTURE VII: 지도 학습 1
(Supervised Learning)

Dai-Gyoung Kim

Department of Applied Mathematics

Hanyang University ERICA

지도 학습 (Supervised Learning)

Contents

- 분류와 회귀
- 일반화, 과대적합, 과소적합
- 지도 학습 알고리즘
- 분류 예측의 불확실성 추정

지도학습

지도학습은 이미 알려진 사례를 바탕으로 일반화된 모델을 만들어 의사 결정 프로세스를 자동화하는 학습 모델로서, 입력과 출력의 훈련샘플 데이터를 기반으로 학습하여 새로운 입력 데이터의 출력을 예측하는 학습 모델이다. 지도 학습의 핵심 문제는 분류(classification)와 회귀(regression)이다.

세부 분류

- 지도학습에서 알고리즘에 주입되는 훈련 데이터는 항상 레이블(label)을 포함함.
- 세부적으로 어떤 것을 예측하느냐에 따라 데이터 레이블의 성질이 규명됨.
- 데이터를 통해 학습하는 레이블이 어떤 성질을 지니는지에 따라 크게 분류, 회귀, 랭킹으로 구분.

수학적 모델

- 지도 학습의 목적은 입력 데이터 x로부터 출력 데이터 y를 매팽(mapping)하는 것을 훈련하는 것임.
- 주어진 입력 데이터셋 X를 정의역으로 하고, 출력 데이터셋(또는 타깃 데이터셋) Y를 치역으로 하는 미지의 함수 f를 가정함.

$$f : X \to Y$$

- 타깃 데이터셋 Y의 연속성에 따라 분류와 회귀 문제로 나눌 수 있음.
  - ☑ Y가 유한 이산형 집합일 경우: **분류 문제**, **랭킹 문제**
  - ☑ Y가 연속형 집합일 경우: **회귀 문제**

- 학습 목표는 훈련 데이터 세트 (X, Y)로부터 함수 f를 추정하고, 추정된 함수 $\hat{f}$를 통하여 새로운 입력 데이터 x의 출력 예측을 $\hat{y} = \hat{f}(x) \in Y$ 구하는 것임.
- 새로운 입력에 대한 출력을 예측하는 것은 **일반화**(generalization)라고 함.

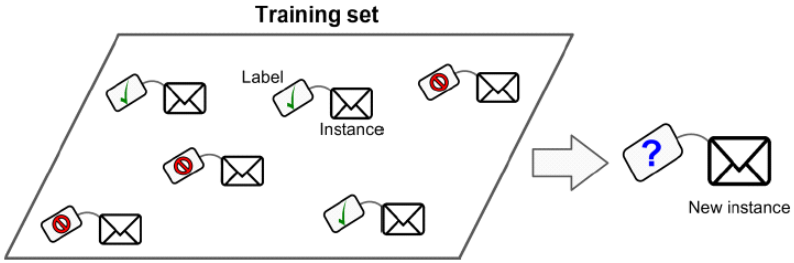
■ 분류와 회귀

분류 (Classification)

- 분류는 클래스 레이블 (class label) 중 하나를 예측하는 모델임.
  (클래스는 데이터의 항목의 집합)
- 분류는 입력 데이터의 항목을 나누는 것임.
  - ✓ 이진 분류 (binary classification) (두개의 클래스 모델)
  - ✓ 다중 클래스 분류 (multiclass classification) (셋 이상의 클래스 모델)
  - ✓ 다중 레이블 분류 (multi-label classification) (다중 출력 모델)

❍ 분류 문제의 예시

[문서 분류와 이메일 스팸 필터링]
메일 샘플과 소속 정보 (스팸인지 아닌지)로 훈련하여 어떻게 새 메일을 분류하는 문제.

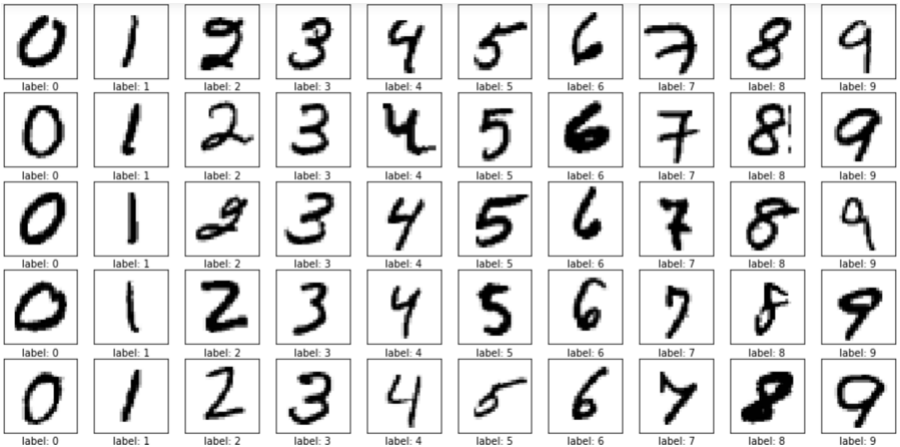


[꽃의 분류]

아이리스 꽃 (붓꽃)의 세 가지 종류인 **setosa, versicolor, virginica** 를 분류하는 문제. 꽃받침의 길이와 너비, 꽃잎의 길이와 너비의 네 가지 특성을 기준으로 분류.

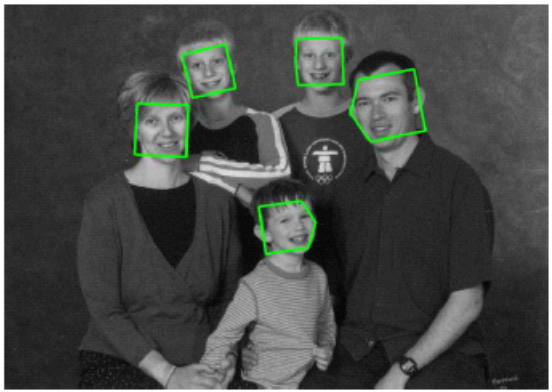
[이미지 분류, 글씨 인식]

손으로 쓴 문자와 숫자로 구성된 이미지에서 글씨 인식을 수행하기 위한 분류 문제. 이 분야에서 사용되는 표준적인 데이터셋은 MNIST(Modified National Institute of Standards)이며, 미국 고등학생과 인구 조사국 직원들이 손으로 쓴 0부터 9까지 숫자 이미지의 60,000개 훈련 데이터셋과 10,000개의 테스트 데이터셋으로 구성되어 있음.



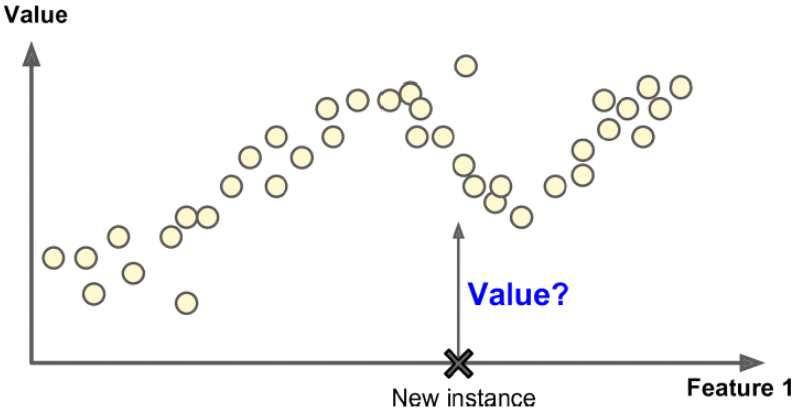
[얼굴 검출과 인식]

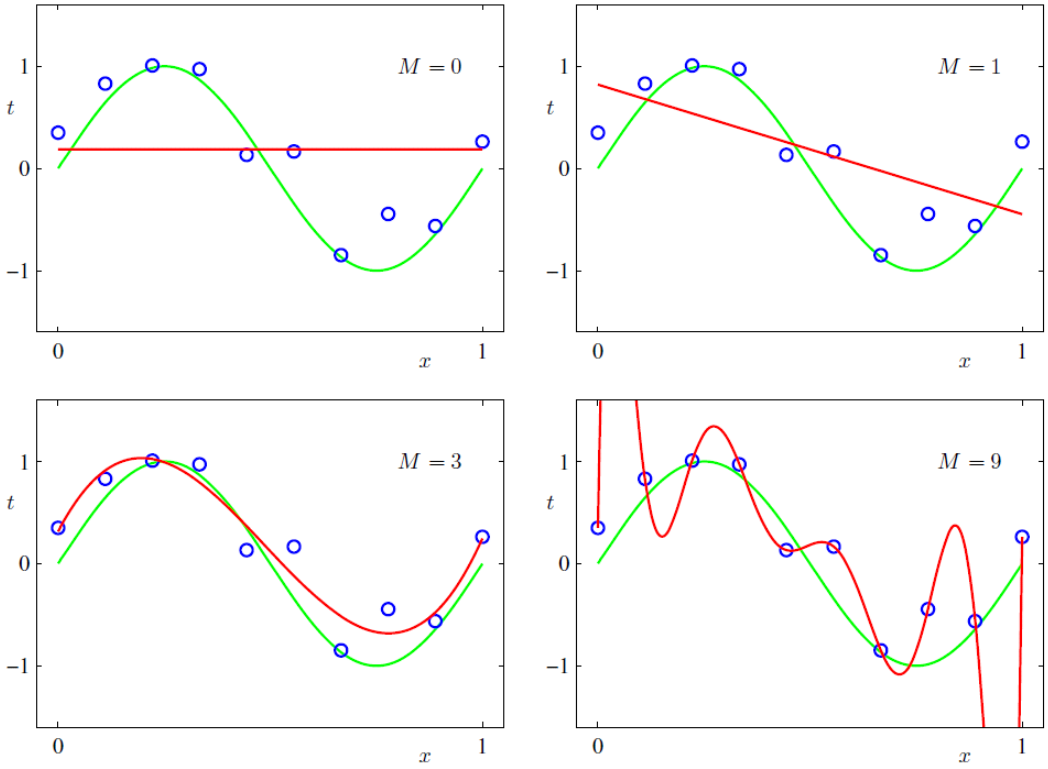
이미지 안의 물체를 인식하는 문제이며, 물체 검출(object detection)이라고 함. 이 문제의 가장 중요한 사례는 얼굴 검출(face detection)임. 얼굴을 찾은 다음, 사람의 신원을 추정하는 얼굴 인식(face recognition) 수행 할 수 있음. 이 경우는 다중 레이블 분류 문제임.



회귀 (Regression)

- 회귀는 연속적인 수치 값 중에서 출력을 예측하는 모델임.
- 어떤 함수가 내재되어 있다고 가정하고 예측 변수라고 하는 특성(feature)을 사용하여 타깃(target) 수치를 예측함.





❍ 회귀 문제의 예시

[주식 예측 문제]
현재의 시장 상태와 다른 가능한 부수적 정보로 미래의 주식을 예측함.

[유튜브 시장 문제]
유튜브에서 특정 비디오를 보는 사용자의 나이를 예측함.

[로봇 팔 문제]
여러 모터로 보낸 제어 신호(토크)가 주어졌을 때 로봇 팔 엔드 이펙트의 3D 공간에서 위치를 예측함.

[의료/의학 문제]
몸속의 전립선 특이 항원(PSA Prostate Specific Antigen)의 양을 다양한 의학적 측정치에 대한 함수로 예측함.

[건물 안의 온도 측정 문제]
날씨 데이터와 시간, 문의 센서 등을 사용해서 건물 안의 온도를 예측함.

[중고차 가격 결정 문제]
주행거리, 연식, 브랜드 등의 특성을 사용해 중고차 가격을 예측함.

[광고비 결정 문제]
광고비와 매출이라는 두 특성을 연관 짓는 함수를 추정함으로써 매출액에 맞춰 광고비를 얼마나 지출해야 할지 예측함.

랭킹 (Ranking)

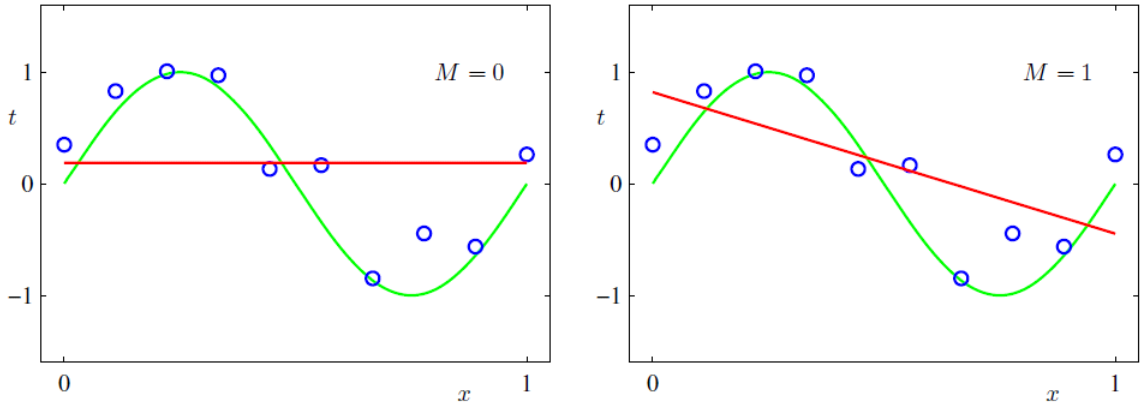
- **데이터의 순위 또는 순서를 예측하는 모델임.**

[추천 시스템]
상품에 대한 사용자 선호도(별점, 구매여부 등)를 예측하는 시스템

■ 과소적합, 과대적합, 일반화

과소적합

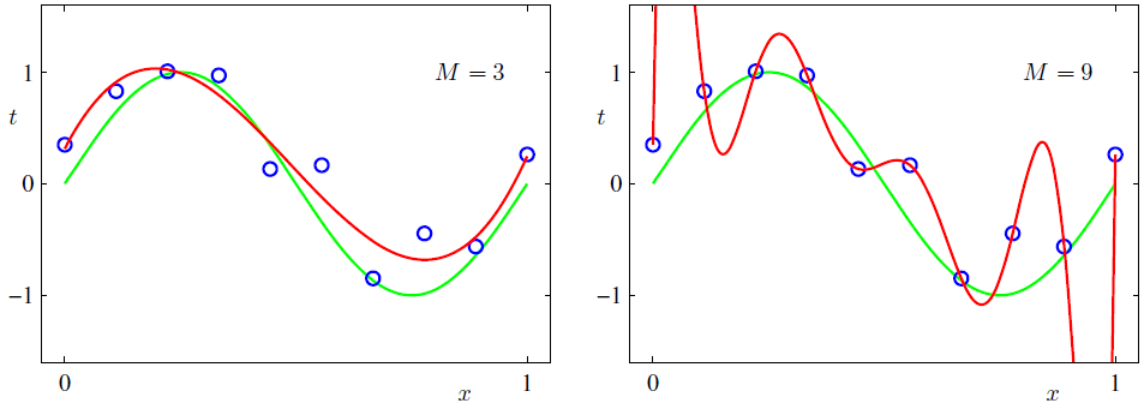
- **과소적합** (underfitting)은 모델이 너무 단순하여 훈련 데이터의 내재된 구조를 학습하지 못함에 따라 훈련 데이터 세트에 잘 맞지 않는 경우.



- 과소적합을 줄이는 방법
  - ✓ 파라미터의 수가 더 많은 모델을 선택함.
  - ✓ 훈련 데이터에 더 좋은 특성을 제공 (특성공학).
  - ✓ 모델의 제약을 줄임.

과대적합

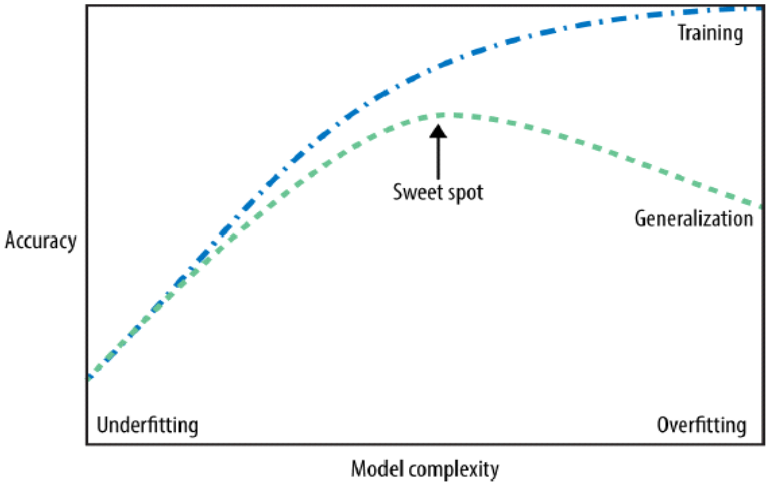
- 가진 정보를 모두 사용하여 너무 복잡한 모델을 만드는 것을 **과대적합** (overfitting)이라고 함.
  - ✓ 모델의 복잡도는 모델 파라미터의 개수와 관련됨
  - ✓ 과대적합 모델은 훈련 데이터셋에는 잘 맞춰져 있지만 새로운 데이터에 일반화되기 어려움.
  - ✓ 과대적합 모델은 훈련 데이터셋에 잡음이 있을 때, 잡음이 섞인 패턴을 도출 할 수 있음.



- 과대적합을 줄이는 방법
  - ✓ 파라미터의 수가 적은 모델을 선택
  - ✓ 모델에 제약을 가하여 단순화시킴 (**모델 규제, regularization**)
  - ✓ 훈련 데이터를 더 많이 수집함
  - ✓ 훈련 데이터의 잡음 (Data errors, Outliers)을 줄임
  - ✓ 데이터를 맞추는 것과 모델을 단순화하는 것 사이의 균형을 최적화함
    - ▷ 규제 하이퍼파라미터 (**hyperparameter**)로 튜닝함.

일반화

- 새로운 입력에 대한 출력을 예측하는 것을 **일반화**(generalization)라고 함.
- 훈련된 모델이 새로운 테스트 데이터셋에 대해 잘 예측할 수 있으면, 이를 훈련 데이터 세트에서 테스트 데이터 세트로 일반화되었다고 함.
- 일반화 성능이 좋은 모델을 얻기 위해서는 과소적합과 과대적합 사이의 균형을 최적화해야 함.



▶ 모델 복잡도와 데이터셋 크기의 관계

- 모델의 복잡도는 훈련 데이터셋에 담긴 입력 데이터의 다양성과 관련이 깊음.
- 일반적으로 큰 데이터셋은 복잡한 모델을 만들 수 있게 해줌.
- 데이터 포인트를 많이 모으는 것은 데이터의 다양성을 키워줌. 그러나 같은 데이터 포인트를 중복하거나 매우 비슷한 데이터를 모으는 것은 좋은 복잡한 모델을 만드는데 도움이 되지 않음.
- 머신러닝에서는 좋은 다양한 데이터셋을 모으는 것이 중요함.

■ 지도 학습 알고리즘

1. k-최근접 이웃 (k-Nearest Neighbors, kNN)
2. 선형 모델 (Linear Model)
3. 나이브 베이즈 분류기 (Naive Bayes Classifier, NBC)
4. 결정 트리 (Decision Tree)
5. 앙상블 결정 트리 (Ensemble of decision Tree)
6. 커널 서포트 벡터 머신 (Kernel Support Vector Machine, KSVM)
7. 신경망, 딥러닝 (Neural Networks, Deep Learning)

▶ 예제에 사용할 데이터셋

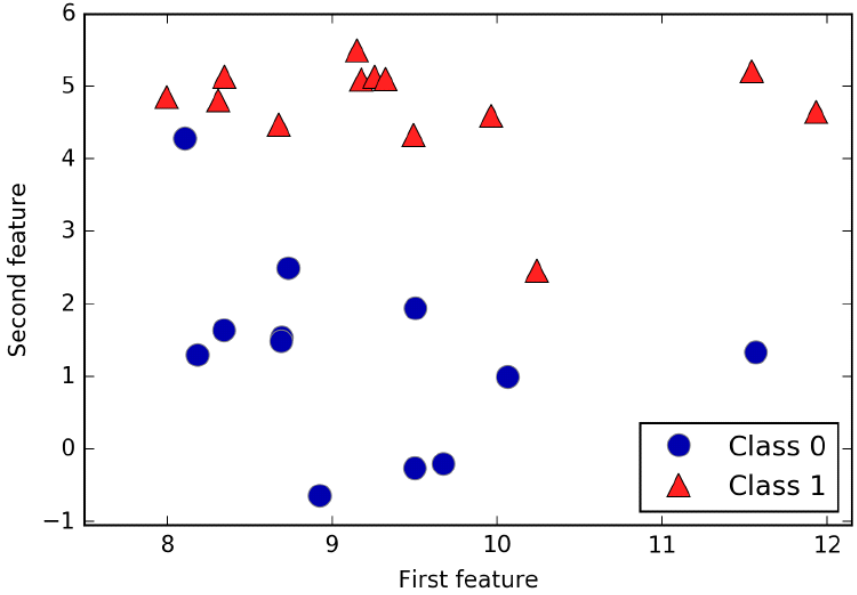
- 여러 알고리즘을 설명하기 위해 사용할 데이터셋을 소개함.

forge 데이터셋

- 두 개의 특성을 가진 forge 데이터셋은 인위적으로 만든 이진 분류 데이터셋임.
- 다음은 “mglearn”에서 forge 데이터셋을 호출하고 이 데이터의 산점도를 그리는 코드임.

```
# generate dataset
X, y = mglearn.datasets.make_forge()

# plot dataset (scatter plot)
mglearn.discrete_scatter(X[:, 0], X[:, 1], y)
plt.legend(["Class 0", "Class 1"], loc=4)
plt.xlabel("First feature")
plt.ylabel("Second feature")
```




```
print("X.shape:", X.shape)
```

```
X.shape: (26, 2)
```

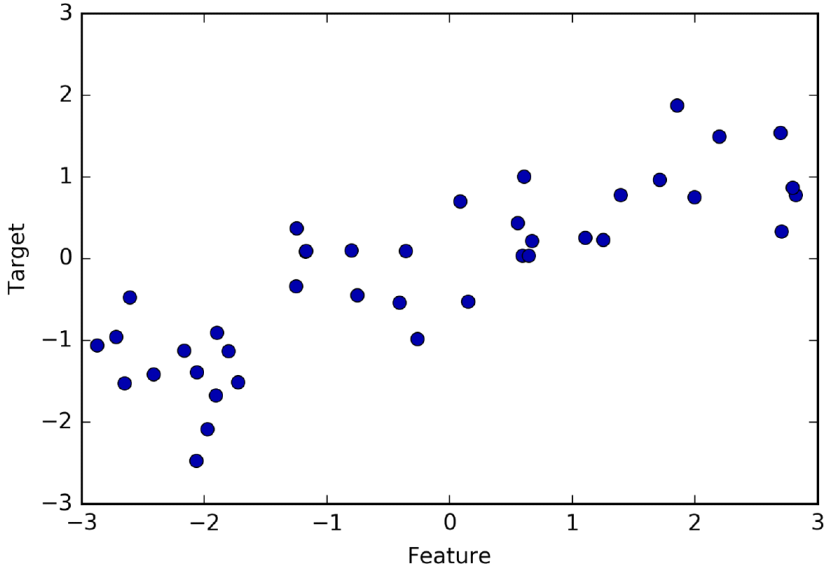
✓ 위 데이터셋은 데이터 포인트 26개와 특성 2개, 타깃 레이블 2개를 갖고 있음.

wave 데이터셋

- 회귀 알고리즘 설명에 이용할 wave 데이터셋은 입력 특성 하나와 모델링할 타깃 변수를 가짐.
- 다음은 “mglearn”에서 wave 데이터셋을 호출하고 이 데이터의 그래프를 그리는 코드임.

```
X, y = mglearn.datasets.make_wave(n_samples=40)


plt.plot(X, y, 'o')
plt.ylim(-3, 3)
plt.xlabel("Feature")
plt.ylabel("Target")
```



유방암 데이터셋

- 다음은 “scikit-learn”에 들어 있는 실제 데이터셋으로서, 유방암 종양의 임상 데이터를 기록해놓은 위스콘신 유방암(Breast Cancer) 데이터셋임.

- 각 종양은 양성(benign)과 악성(malignant)으로 레이블되어 있고, 조직 데이터 기반으로 종양이 악성인지를 예측할 수 있도록 학습하는 것이 과제임.
- 다음은 “scikit-learn”에 있는 “load_breast_cancer” 함수를 사용하여 데이터셋을 불러오는 코드임.

```
from sklearn.datasets import load_breast_cancer

cancer = load_breast_cancer()
print("cancer.keys():\n", cancer.keys())
```

```
cancer.keys():
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names', 'filename'])
```

```
print("Shape of cancer data:", cancer.data.shape)
```

```
Shape of cancer data: (569, 30)
```

✓ 이 데이터셋에는 데이터 포인트 569개와 특성 30개가 있음.

```
print("Sample counts per class:\n",
      {n: v for n, v in zip(cancer.target_names, np.bincount(cancer.target))})
```

```
Sample counts per class:
      {'malignant': 212, 'benign': 357}
```

✓ 569개 데이터 포인트 중 212개는 악성이고 357개는 양성임.

```
print("Feature names:\n", cancer.feature_names)
```

```
Feature names:
['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
```

주택가격 데이터셋

- 회귀 분석용 실제 데이터셋으로는 보스턴 주택가격 데이터셋을 사용함.
- 이 데이터셋의 범죄율, 찰스강 인접도, 고속도로 접근성 등의 정보를 이용하여 1970년대 보스턴 주변의 주택 평균 가격을 예측할 수 있도록 학습하는 것이 과제임.

```
from sklearn.datasets import load_boston

boston = load_boston()
print("Data shape:", boston.data.shape)
```

```
Data shape: (506, 13)
```

✓ 이 데이터셋에는 데이터 포인트 506개와 특성 13개가 있음.

✓ boston 객체에의 DESCR 속성에서 더 자세한 정보를 확인할 수 있음.

- 이 데이터셋에서 13개의 특성들을 곱하여 특성간의 상호작용을 새로운 특성으로 추가하여 데이터의 특성 수를 확장할 수 있음.
  - ✓ 예를 들어 범죄율과 고속도로 접근성을 곱하여 범죄율과 고속도로 접근성의 상호작용을 새로운 특성으로 추가할 수 있음 (특성공학).

- 다음은 “load_extended_boston” 함수를 사용하여 특성 수를 확장하는 코드임.

```
X, y = mglearn.datasets.load_extended_boston()
print("X.shape:", X.shape)
```

```
X.shape: (506, 104)
```

  - ✓ 13개의 원래 특성에 13개에서 2개씩(중복을 허락한 조합) 짝지은 91개의 특성을 더해 총 104개의 특성으로 확장함.