

2장 머신러닝 프로젝트 처음부터 끝까지 (1부)

감사의 글

자료를 공개한 저자 오렐리앙 제롱과 강의자료를 지원한 한빛아카데미에게 진심어린 감사를 전합니다.

주요 내용

- 주택 가격을 예측하는 회귀 작업을 살펴보면서 선형 회귀, 결정 트리, 랜덤 포레스트 등 여러 알고리즘의 기본 사용법 소개
- 머신러닝 시스템 전체 훈련 과정 살펴보기

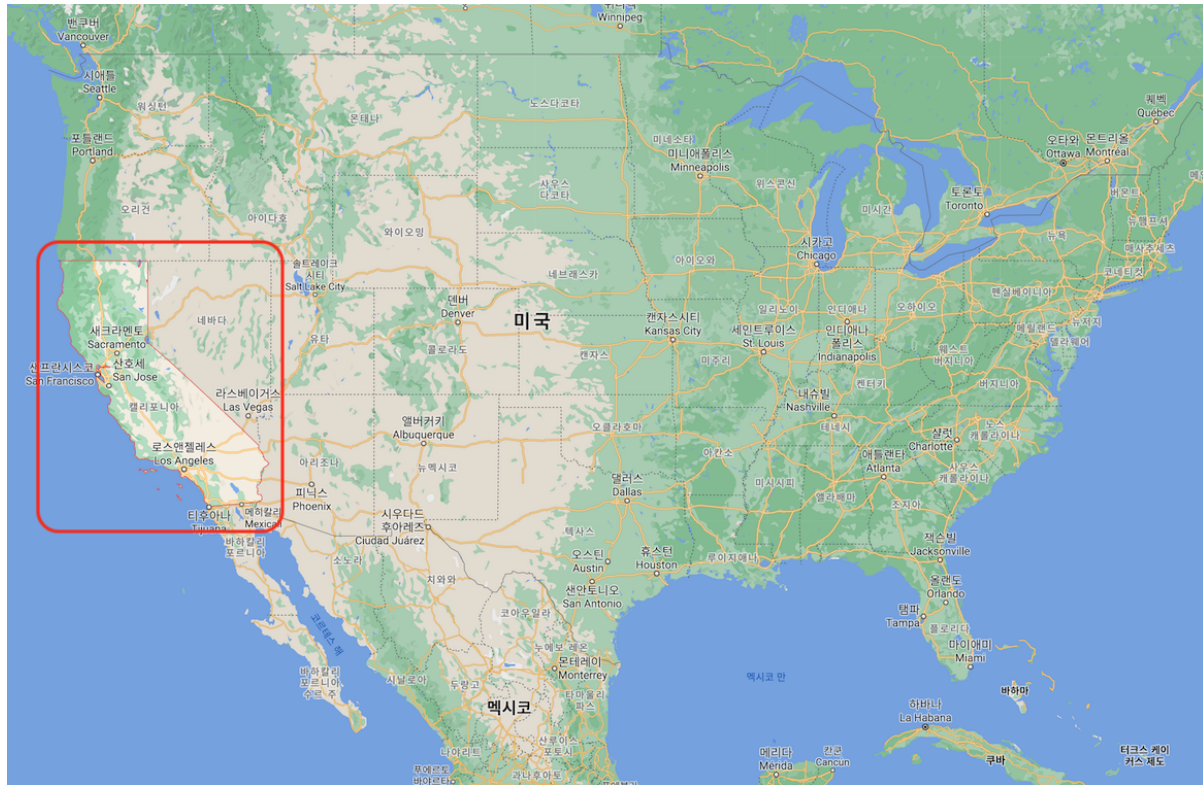


2.2 큰 그림 보기

주어진 데이터

- 미국 캘리포니아 주의 20,640개 지역별 인구조사 데이터
- 특성 10개: 경도, 위도, 중간 주택 연도, 방의 총 개수, 침실 총 개수, 인구, 가구 수, 중간 소득, 중간 주택 가격, 해안 근접도
- 목표: 구역별 중간 주택 가격 예측 시스템(모델) 구현하기

- 미국 캘리포니아 지도



2.2.1 문제 정의

- 지도 학습(supervised learning)
 - 레이블: 구역별 중간 주택 가격
- 회귀(regression): 중간 주택 가격 예측
 - 다중 회귀(multiple regression): 여러 특성을 활용한 예측
 - 단변량 회귀(univariate regression): 구역마다 하나의 가격만 예측
- 배치 학습(batch learning): 빠르게 변하는 데이터에 적응할 필요가 없음

2.2.2 성능 측정 지표 선택

사용하는 모델에 따라 모델 성능 측정 기준(norm)을 다르게 선택한다. 선형 회귀 모델의 경우 일반적으로 아래 두 기준 중 하나를 사용한다.

- 평균 제곱근 오차(RMSE)
- 평균 절대 오차(MAE)

평균 제곱근 오차(root mean square error, RMSE)

- 유클리디안 노름(Euclidean norm) 또는 ℓ_2 노름(norm)으로도 불림
- 참고: 노름(norm)은 거리 측정 기준을 나타냄.

$$\text{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(\mathbf{x}^{(i)}) - y^{(i)})^2}$$

- 기호 설명
 - \mathbf{X} : 훈련 데이터셋 전체 샘플들의 특성값들로 구성된 행렬, 레이블(타겟) 제외.
 - $\mathbf{x}^{(i)}$: i 번째 샘플의 전체 특성값 벡터. 레이블(타겟) 제외.
 - $y^{(i)}$: i 번째 샘플의 레이블
 - h : 예측 함수
 - $\hat{y}^{(i)} = h(\mathbf{x}^{(i)})$: i 번째 샘플에 대한 예측 값

평균 절대 오차(mean absolute error, MAE)

- MAE는 맨해튼 노름 또는 ℓ_1 노름으로도 불림

$$\text{MAE}(\mathbf{X}, h) = \frac{1}{m} \sum_{i=1}^m |h(\mathbf{x}^{(i)}) - y^{(i)}|$$

- 이상치가 많은 경우 활용
- ℓ_1 노름과 ℓ_2 노름을 일반해서 ℓ_n 노름을 정의할 수도 있음
- RMSE가 MAE보다 이상치에 더 민감하지만, 이상치가 많지 않을 경우 일반적으로 RMSE 사용

2.3 데이터 가져오기

2.3.3 데이터 구조 훑어보기

첫 5개 데이터 살펴보기

```
In [5]: housing.head()
```

```
Out[5]:
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population
0	-122.23	37.88	41.0	880.0	129.0	322.0
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0
2	-122.24	37.85	52.0	1467.0	190.0	496.0
3	-122.25	37.85	52.0	1274.0	235.0	558.0
4	-122.25	37.85	52.0	1627.0	280.0	565.0

전체 데이터셋 요약 정보

```
[6] 1 housing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   longitude             20640 non-null  float64
1   latitude              20640 non-null  float64
2   housing_median_age    20640 non-null  float64
3   total_rooms           20640 non-null  float64
4   total_bedrooms        20433 non-null  float64
5   population            20640 non-null  float64
6   households            20640 non-null  float64
7   median_income         20640 non-null  float64
8   median_house_value    20640 non-null  float64
9   ocean_proximity       20640 non-null  object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

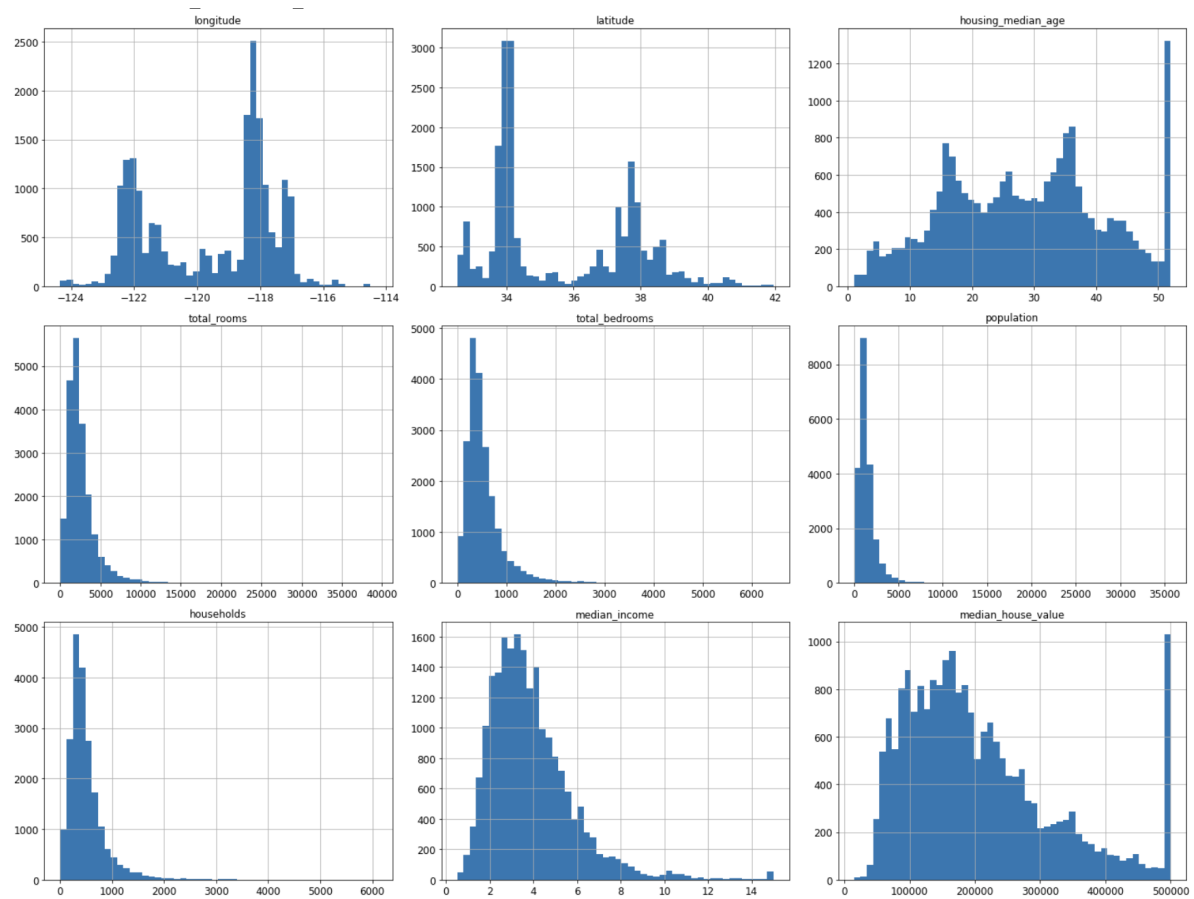
- 구역 수: 20,640개
- 구역별로 경도, 위도, 중간 주택 연도, 해안 근접도 등 총 10개의 조사 항목
 - '해안 근접도'는 범주형 특성이고 나머지는 수치형 특성.
- '방의 총 개수'의 경우 누락된 데이터인 207개의 null 값 존재

범주형 특성 탐색

- '해안 근접도'는 5개의 범주로 구분

특성값	설명
<1H OCEAN	해안에서 1시간 이내
INLAND	내륙
NEAR OCEAN	해안 근처
NEAR BAY	샌프란시스코의 Bay Area 지역
ISLAND	섬

수치형 특성별 히스토그램



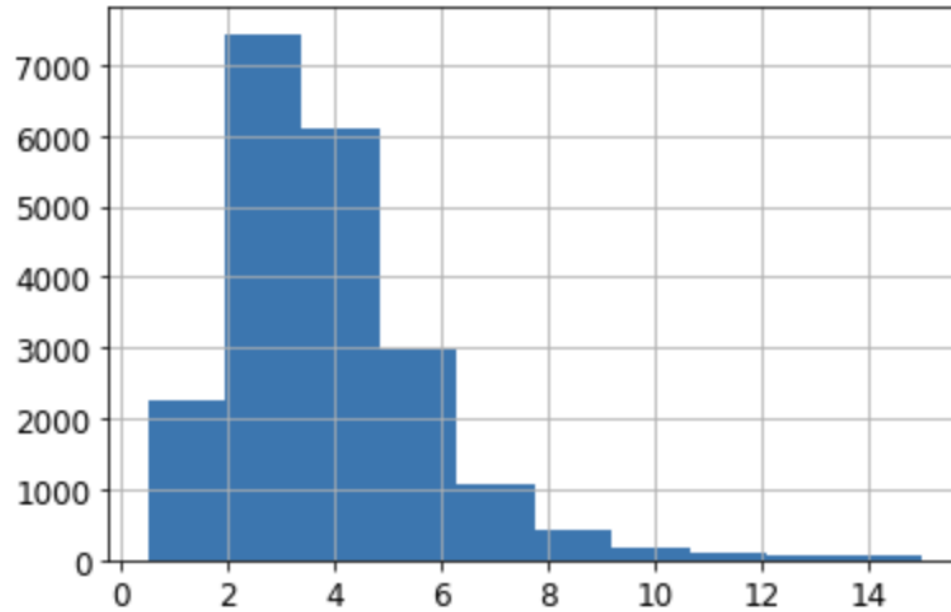
2.3.4 테스트 세트 만들기

- 모델 학습 시작 이전에 준비된 데이터셋을 훈련 세트와 테스트 세트로 구분
 - 테스트 세트 크기: 전체 데이터 셋의 20%
- 테스트 세트에 포함된 데이터는 미리 분석하지 말 것.
 - 미리 분석 시 **데이터 스누핑 편향**을 범할 가능성이 높아짐
 - 미리 보면서 알아낸 직관이 학습 모델 설정에 영향을 미칠 수 있음
- 훈련 세트와 데이터 세트를 구분하는 방식에 따라 결과가 조금씩 달라짐
 - 무작위 샘플링 vs. 계층적 샘플링
- 여기서는 계층적 샘플링 활용

계층적 샘플링

- 계층: 동질 그룹
 - 예제: 소득별 계층
- 테스트 세트: 전체 계층을 대표하도록 각 계층별로 적절한 샘플 추출
- 예제: 소득 범주
 - 계층별로 충분한 크기의 샘플이 포함되도록 지정해야 학습 과정에서 편향이 발생하지 않음
 - 특정 소득 구간에 포함된 샘플이 과하게 적거나 많으면 해당 계층의 중요도가 과대 혹은 과소 평가됨

- 전체 데이터셋의 중간 소득 히스토그램 활용



- 대부분 구역의 중간 소득이 **1.5~6.0**(15,000~60,000\$) 사이
- 소득 구간을 아래 숫자를 기준으로 5개로 구분

```
[0, 1.5, 3.0, 4.6, 6.0, np.inf]
```

계층 샘플링과 무작위 샘플링 비교

	전체	계층 샘플링	무작위 샘플링	무작위 샘플링 오류율	계층 샘플링 오류율
1	0.039826	0.039729	0.040213	0.973236	-0.243309
2	0.318847	0.318798	0.324370	1.732260	-0.015195
3	0.350581	0.350533	0.358527	2.266446	-0.013820
4	0.176308	0.176357	0.167393	-5.056334	0.027480
5	0.114438	0.114583	0.109496	-4.318374	0.127011

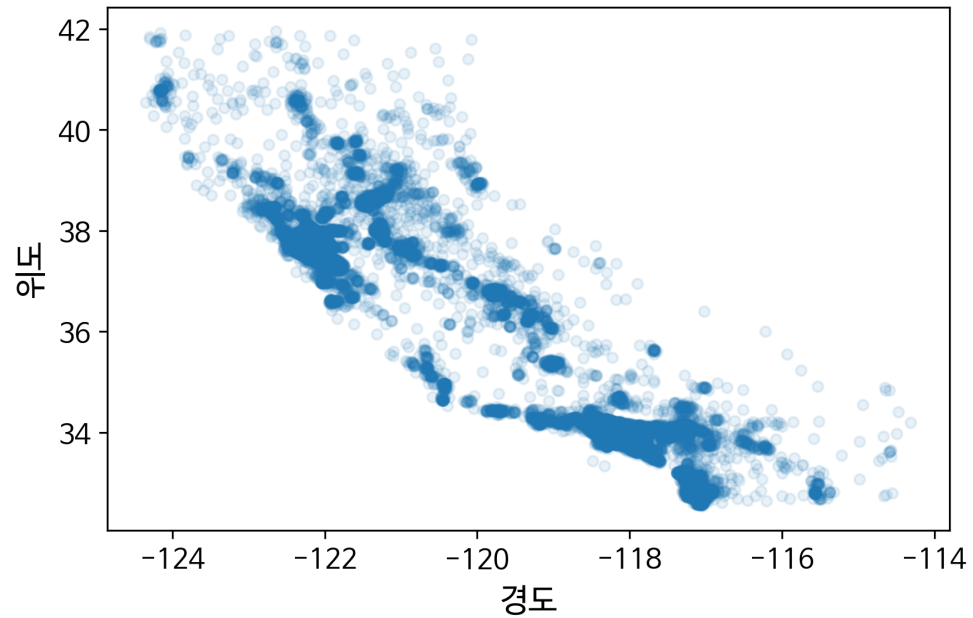
2.4 데이터 이해를 위한 탐색과 시각화

주의 사항

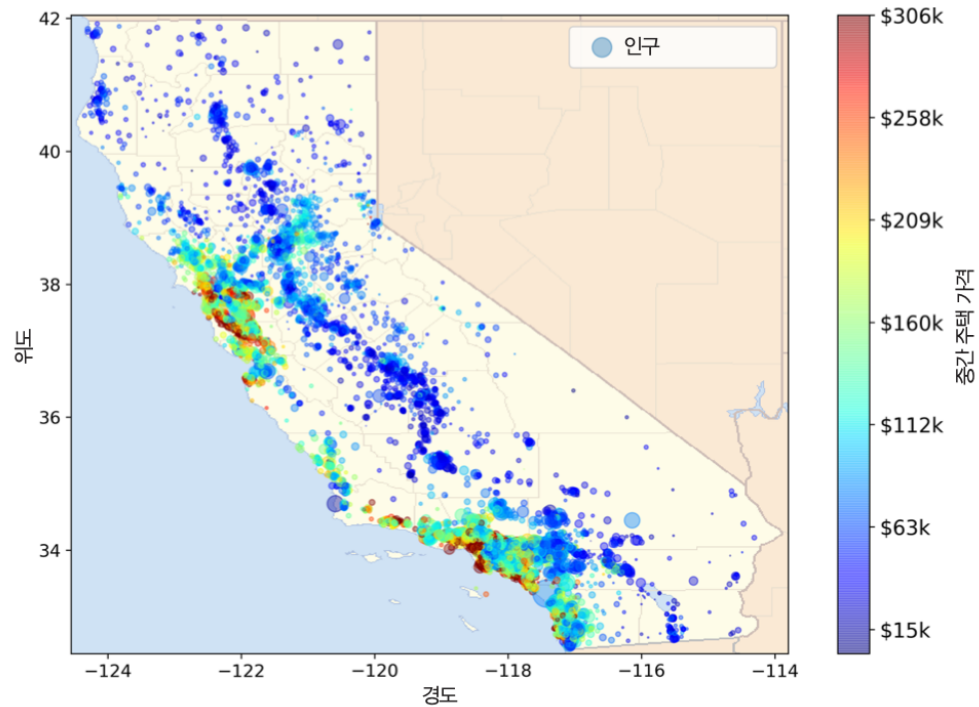
- 테스트 세트를 제외한 훈련 세트에 대해서만 시각화를 이용하여 탐색
- 데이터 스누핑 편향 방지 용도

2.4.1 지리적 데이터 시각화

- 구역이 집결된 지역과 그렇지 않은 지역 구분 가능
- 샌프란시스코의 베이 에어리어, LA, 샌디에고 등 밀집된 지역 확인 가능



- 주택 가격이 해안 근접도 또는 인구 밀도와 관련이 큼
- 해안 근접도: 위치에 따라 다르게 작용
 - 대도시 근처: 해안 근처 주택 가격이 상대적 높음
 - 북부 캘리포니아 지역: 높지 않음



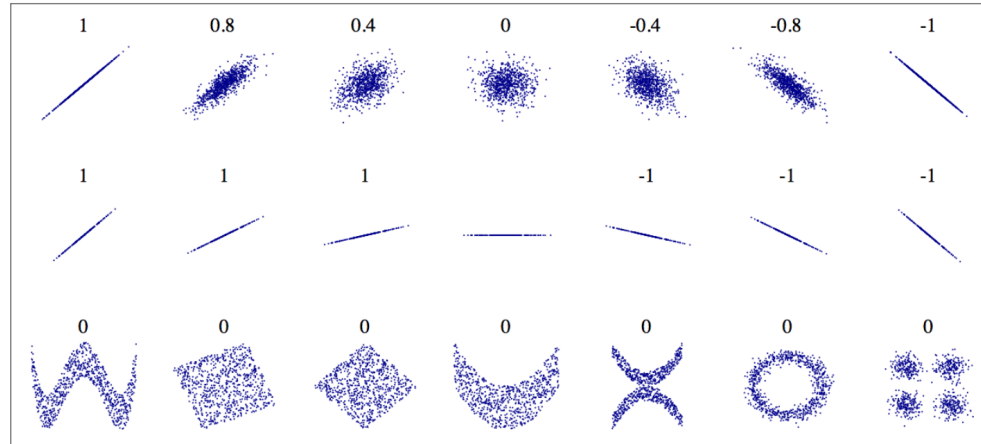
2.4.2 상관관계 조사

- 중간 주택 가격 특성과 다른 특성 사이의 상관관계: 상관계수 활용

```
In [39]: corr_matrix["median_house_value"].sort_values(ascending=False)
```

```
Out[39]: median_house_value    1.000000  
         median_income        0.687160  
         total_rooms          0.135097  
         housing_median_age    0.114110  
         households           0.064506  
         total_bedrooms        0.047689  
         population           -0.026920  
         longitude            -0.047432  
         latitude             -0.142724  
         Name: median_house_value, dtype: float64
```


상관계수의 특징



<그림 출처: 위키백과>

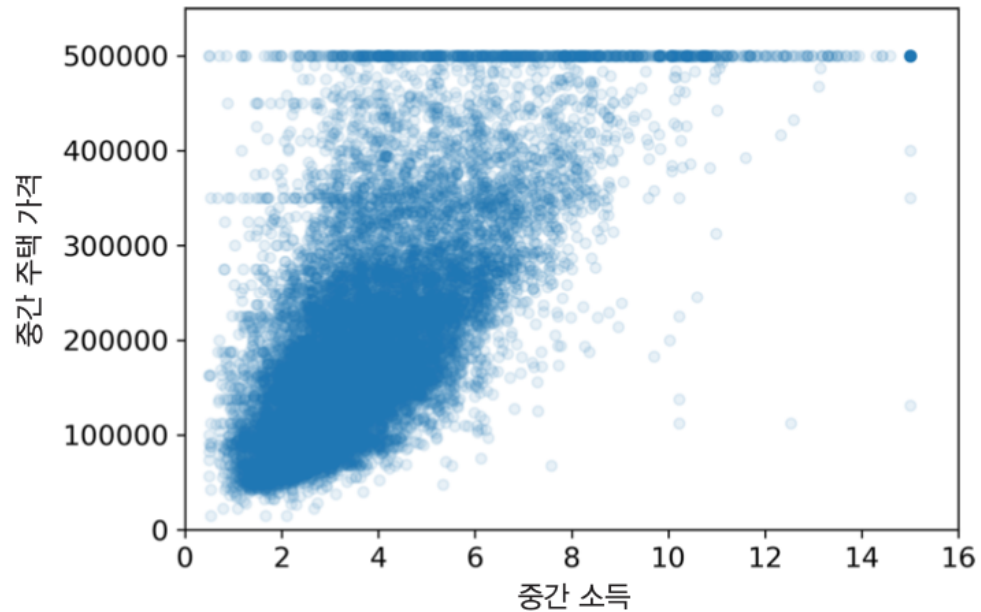
- 상관계수: $[-1, 1]$ 구간의 값
- 1에 가까울 수록: 강한 양의 선형 상관관계
- -1에 가까울 수록: 강한 음의 선형 상관관계
- 0에 가까울 수록: 매우 약한 선형 상관관계

주의사항

- 상관계수가 0에 가까울 때: 선형 관계가 거의 없다는 의미이지, 아무런 관계가 없다는 의미는 아님
- 상관계수는 기울기와 아무 연관 없음

상관계수를 통해 확인할 수 있는 정보

- 중간 주택 가격과 중간 소득의 상관계수가 0.68로 가장 높음
 - 중간 소득이 올라가면 중간 주택 가격도 상승하는 경향이 있음
 - 점들이 너무 넓게 퍼져 있음. 완벽한 선형관계와 거리 멀.



2.4.3 특성 조합으로 실험

- 구역별 방의 총 개수와 침실의 총 개수 대신 아래 특성이 보다 유용함
 - 가구당 방 개수(rooms for household)
 - 방 하나당 침실 개수(bedrooms for room)
 - 가구당 인원(population per household)

```
[47] 1 corr_matrix = housing.corr()  
     2 corr_matrix["median_house_value"].sort_values(ascending=False)
```

median_house_value	1.000000
median_income	0.687160
rooms_per_household	0.146285
total_rooms	0.135097
housing_median_age	0.114110
households	0.064506
total_bedrooms	0.047689
population_per_household	-0.021985
population	-0.026920
longitude	-0.047432
latitude	-0.142724
bedrooms_per_room	-0.259984

Name: median_house_value, dtype: float64

- 중간 주택 가격과 방 하나당 침실 개수의 연관성 다소 있음
- 가구당 방 개수의 역할은 여전히 미미함

2.5 머신러닝 알고리즘을 위한 데이터 준비

데이터 전처리

- 데이터 전처리(data preprocessing): 효율적인 모델 훈련을 위한 데이터 변환
- 수치형 특성과 범주형 특성에 대해 다른 변환과정을 사용
- 수치형 특성 전처리 과정
 - 데이터 정제
 - 조합 특성 추가
 - 특성 스케일링
- 범주형 특성 전처리 과정
 - 원-핫-인코딩(one-hot-encoding)

2.5.1 데이터 정제: 수치형 특성 전처리 과정 1

- 누락된 특성값이 존재 경우, 해당 값 또는 특성을 먼저 처리해야 함.
- `total_bedrooms` 특성에 207개 구역에 대한 값이 null로 채워져 있음, 즉, 일부 구역에 대한 정보가 누락됨.

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_p
4629	-118.30	34.07	18.0	3759.0	NaN	3296.0	1462.0	2.2708	<1+
6068	-117.86	34.01	16.0	4632.0	NaN	3038.0	727.0	5.1762	<1+
17923	-121.97	37.35	30.0	1955.0	NaN	999.0	386.0	4.6328	<1+
13656	-117.30	34.05	6.0	2155.0	NaN	1039.0	391.0	1.6675	
19252	-122.79	38.48	7.0	6837.0	NaN	3468.0	1405.0	3.1662	<1+

null 값 처리 옵션

- 옵션 1: 해당 구역 제거
- 옵션 2: 전체 특성 삭제
- 옵션 3: 평균값, 중앙값, 0, 주변에 위치한 값 등 특정 값으로 채우기. 책에서는 중앙값으로 채움.

<옵션 3 활용>

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	ocean_p
4629	-118.30	34.07	18.0	3759.0	433.0	3296.0	1462.0	2.2708	<1+
6068	-117.86	34.01	16.0	4632.0	433.0	3038.0	727.0	5.1762	<1+
17923	-121.97	37.35	30.0	1955.0	433.0	999.0	386.0	4.6328	<1+
13656	-117.30	34.05	6.0	2155.0	433.0	1039.0	391.0	1.6675	
19252	-122.79	38.48	7.0	6837.0	433.0	3468.0	1405.0	3.1662	<1+

2.5.2 텍스트와 범주형 특성 다루기: 원-핫 인코딩

- 범주형 특성인 해안 근접도(ocean_proximity)를 수치형 특성으로 변환해야 함.
- 단순 수치화 적용 가능

범주	숫자
<1H OCEAN	0
INLAND	1
ISLAND	2
NEAR BAY	3
NEAR OCEAN	4

단순 수치화의 문제점

- 해안 근접도는 단순히 구분을 위해 사용. 해안에 근접하고 있다 해서 주택 가격이 기본적으로 더 비싸지 않음.
- 반면에 수치화된 값들은 크기를 비교할 수 있는 숫자
- 따라서 모델 학습 과정에서 숫자들의 크기 때문에 잘못된 학습이 이루어질 수 있음.

원-핫 인코딩(one-hot encoding)

- 수치화된 범주들 사이의 크기 비교를 피하기 위해 더미(dummy) 특성을 추가하여 활용
 - 범주 수 만큼의 더미 특성 추가
- 예를 들어, 해안 근접도 특성 대신에 다섯 개의 범주 전부를 새로운 특성으로 추가한 후 각각의 특성 값을 아래처럼 지정
 - 해당 카테고리의 특성값: 1
 - 나머지 카테고리의 특성값: 0

```
In [68]: cat_encoder = OneHotEncoder(sparse=False)
housing_cat_1hot = cat_encoder.fit_transform(housing_cat)
housing_cat_1hot
```

```
Out[68]: array([[1., 0., 0., 0., 0.],
                [1., 0., 0., 0., 0.],
                [0., 0., 0., 0., 1.],
                ...,
                [0., 1., 0., 0., 0.],
                [1., 0., 0., 0., 0.],
                [0., 0., 0., 1., 0.]])
```

2.5.4 특성 스케일링: 수치형 특성 전처리 과정 3

- 머신러닝 알고리즘은 입력 데이터셋의 특성값들의 스케일(범위)이 다르면 제대로 작동하지 않음
- 특성에 따라 다루는 숫자의 크기가 다를 때 통일된 스케일링이 필요
- 아래 두 가지 방식이 일반적으로 사용됨.
 - min-max 스케일링
 - 표준화 (책에서 사용)
- 주의: 타깃(레이블)에 대한 스케일링은 하지 않음

min-max 스케일링

- 정규화(normalization)라고도 불림
- 특성값 x 를 $\frac{x-min}{max-min}$ 로 변환
- 변환 결과: **0에서 1** 사이
- 이상치에 매우 민감
 - 이상치가 매우 크면 분모가 매우 커져서 변환된 값이 **0** 근처에 몰림

표준화(standardization)

- 특성값 x 를 $\frac{x-\mu}{\sigma}$ 로 변환
 - μ : 특성값들의 평균값
 - σ : 특성값들의 표준편차
- 결과: 변환된 데이터들이 표준정규분포를 이룸
 - 이상치에 상대적으로 영향을 덜 받음.
- 사이킷런의 `StandardScaler` 변환기 활용 가능 (책에서 사용)