기계학습 - 230919

9조 - 위다빈 김한결

3주차: 머신러닝 프로젝트 처음부터 끝까지

- 인공 데이터 셋이 아닌 실제 데이터를 이용하자

→ 인공 데이터 셋은 가공되어있어서 이 데이터로만 실험한다면 실전에서는 잘 안될 수 있음 (실제 데이터는 노이즈가 많기 때문)

→ 캘리포니아 주택 가격을 이용

- 주택 가격 모델

데이터 :블록 그룹마다 인구/중간소득/중간 주택가격

목표 : 중간 가격 예측

- 문제 정의 (문제에 대해 정확하게 알아야함)

비즈니스의 목적이 무엇인가? → 적당한 나일론 양말

문제 구성, 알고리즘 선택, 모델 평가와 성능 지표, 모델 튜닝시에 리소스 결정

지도,비지도,강화학습,분류,회귀 어떤걸 선택할지

- 성능 측정 지표 선택

식 2-1 평균 제곱근 오차

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

h(xi) = 모델 예측

y= label(정답)

- 가정 검사

데이터 훑어보기

```
다빈: ocean_proximity는 왜 타입이 숫자로 안나올까?

한결: ocean_proximity는 범주형 데이터로 숫자형 데이터와는 다르게 이름이나 라벨로 데이터가 표현돼

다빈: 그래서 숫자형 데이터와 데이터 구조를 확인하는 함수가 다르구나
```

▽ 데이터 구조 살펴보기 - 범주형 데이터

```
[6]  housing["ocean_proximity"].value_counts()
```

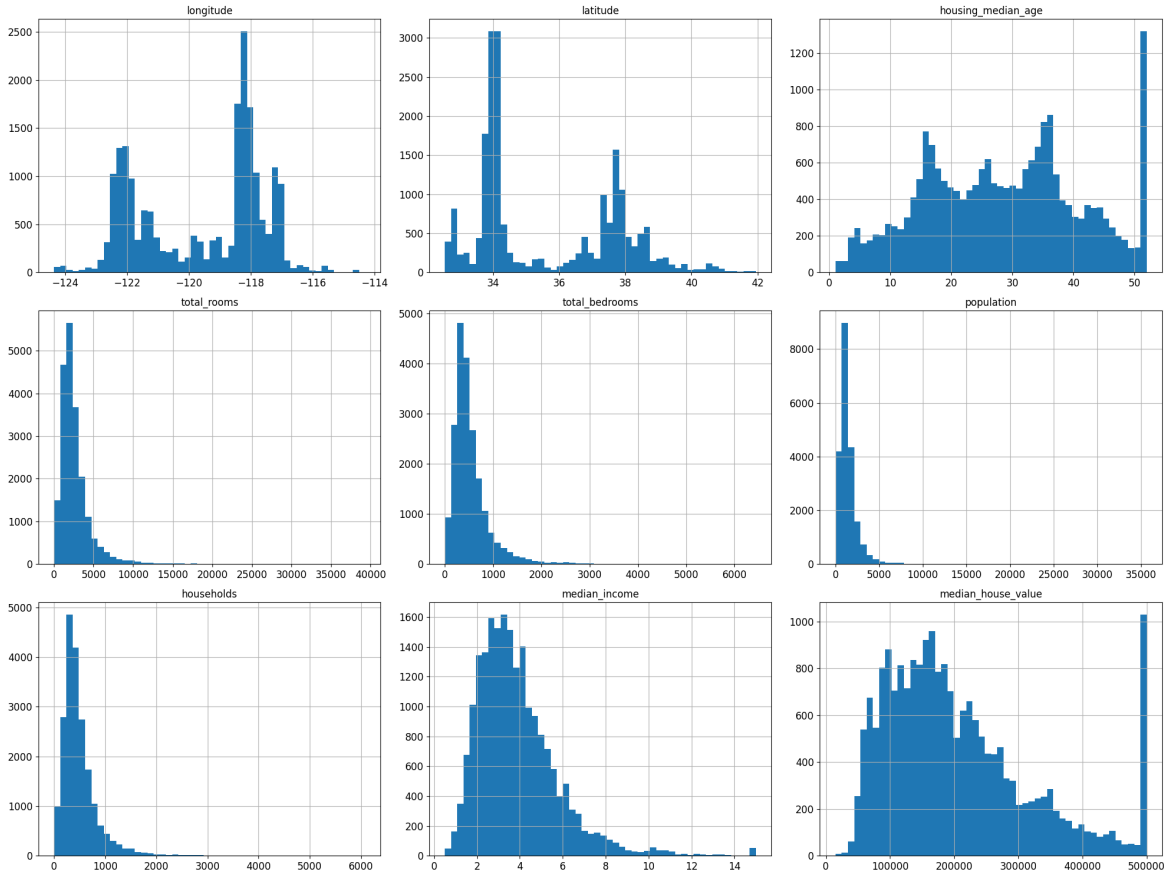
```
<1H OCEAN     9136
INLAND        6551
NEAR OCEAN    2658
NEAR BAY      2290
ISLAND           5
Name: ocean_proximity, dtype: int64
```

▽ 데이터 구조 살펴보기 - 숫자형 데이터

```
housing.describe()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	populat
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000

▼ 데이터




```
#히스토그램을 이용하면 데이터 형태를 빠르게 검토할 수 있구나
#median income을 보면 스케일링에 안맞은다 -> 모델링을 하려면 스케일링을 맞춰야 한다
```

```
#housing_median_age나 median_house_value 같은 경우 우측에 이상한 데이터가 보인다-> 머신은 하나의 패턴으로 인식할 수 있음 아예 제거를 하던지 데이터
# somethingworng발생! : median_house_value 200000을 중간으로 볼때 우측부분 꼬리가 너무 길다 -> 종모양(엎은 모자)의 정규분포에서 가장 학습 잘됨(
```

과대적합 해결방법

```
한결: 훈련세트와 테스트 세트의 비율을 적절하게 조절해야한다. 과대적합은 훈련세트에만 너무 데이터가
치중되어있어 발생하는 문제이기 때문에 훈련세트의 비율을 줄이고 테스트 세트의 비율을 늘려야한다.

다빈: 훈련세트가 100 이 있다면 어느정도 비율이 가장 학습이 잘 되는 지 모르겠지만 50만큼 학습 시키고 50만큼 테스트를 한다고 한다면
훈련세트를 10개씩 묶음으로 만들어서 각 묶음마다 A,B,C,D .. 이름을 매긴후 10개중 5개를 뽑는 경우의 수를 따져서 그만큼 테스트를 진행한뒤 평균을 내보는것이
과대적합을 필할 수 있을 것 같아.
(찾아본 내용 : 교차 검증)
```

왜 우리가 데이터를 먼저 보아야 하는가?

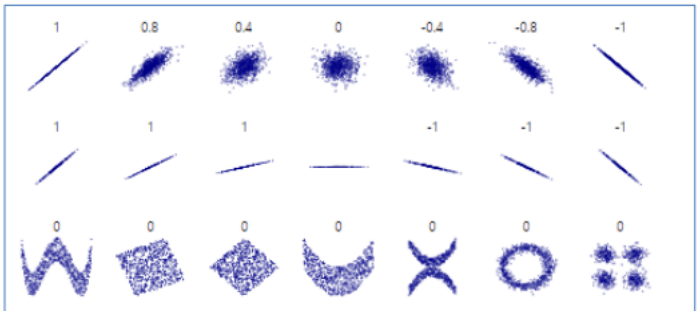
```
다빈 : 위에 정리해 두었음
한결 : 모델 성능 향상을 위한 전처리를 위해 데이터의 특성을 파악해야 하기 때문.
```

무작위 샘플링 vs 계층적 샘플링

```
한결: 무작위 샘플링에선 샘플링 편향 문제가 발생할 수 있음

다빈 : 편향된다는게 이미 존재하는 데이터가 벌써 편향적일 수 있다는 거야?

한결: 클래스나 카테고리가 불균형하게 분포되어있으면 무작위 샘플링 과정에서 샘플링 편향이
발생할 수 있는 것으로 알고있어

다빈 : 나는 이미 존재하는 데이터가 편향적 이라는건줄 알았어 뭔가 무료로? 쓸수 있는 데이터는 옛날 데이터일것같아서
옛날 데이터에는 아무래도 편향적인 사고로 쓰여진 내용이 많을거라고 생각했어
아니면 특정 집단이나 기업의 이익을 위해 의도적으로 개방된 데이터가 많다 이런건줄 알았어

다빈 : 계층적 샘플링의 정의를 찾아볼게 '모집단의 데이터 분포 비율을 유지하면서 데이터를 샘플링'
확실히 집단의 분포를 확인하면서 샘플을 뽑는다면 편향되지 않을것같아
두 샘플링중 좋은 결과를 내려면 계층적 샘플링을 앞으로 사용해야겠어
```

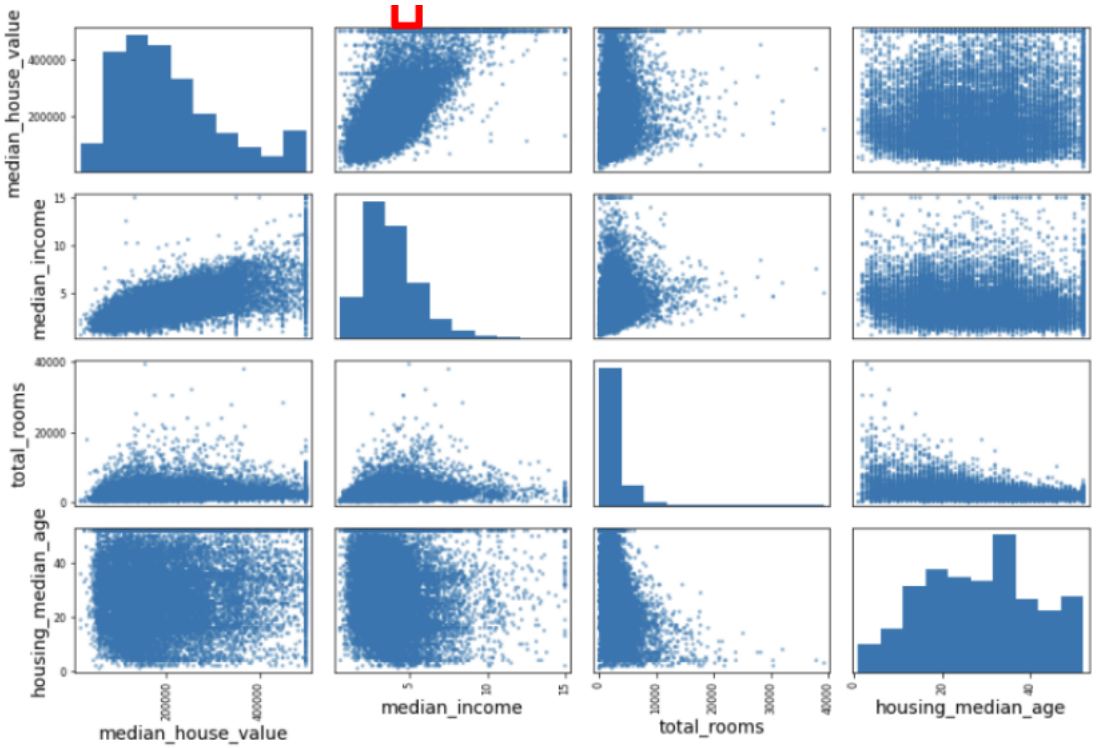
- 상관관계 조사

어떤걸 쓸지



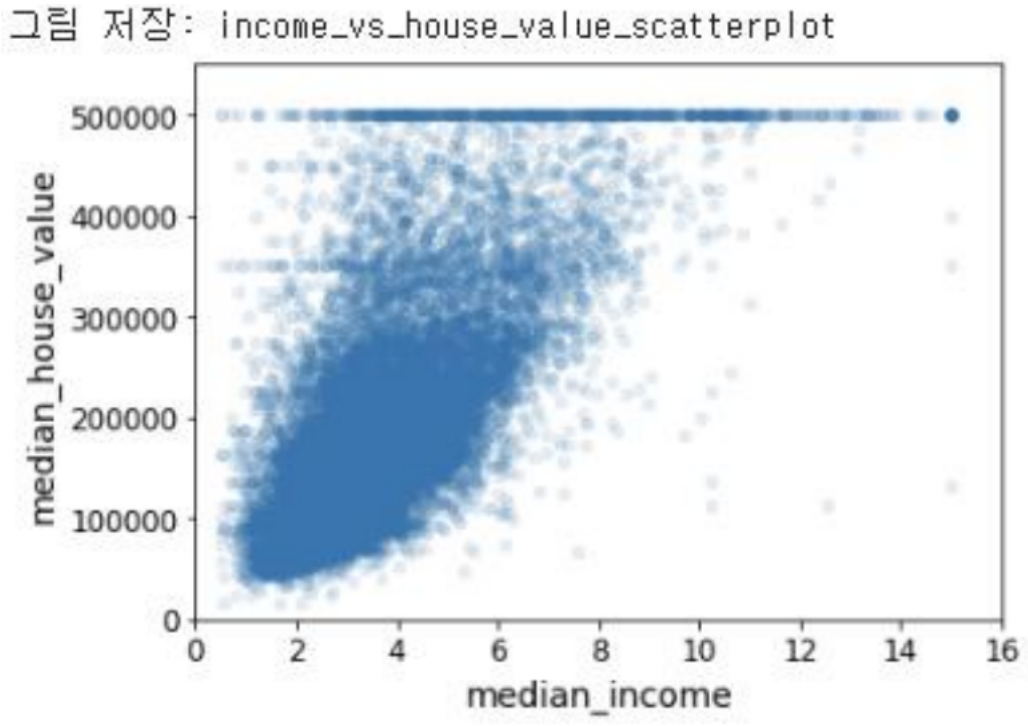
데이터 보았을 때 선형 관계가 있다면(에 상관없이 1에가깝다) 리니어로 선택

중간 주택 가격과 중간 소득의 관계
우측 상단같은 경우 패턴을 보기 힘듦



하지만 좌측 상단에서 오른쪽으로 2번째는 선형인 특성이 보임 + median_house_value,median_income 이 y축x축임

선택해서 보았더니 이상한 데이터가 보임



저 데이터를 없애거나 누그러뜨리거나 다른 조치를 취해야함

- 특성 조합

정의 : 캐릭터의 조합을 통해

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

```
median_house_value          1.000000
median_income               0.687151
rooms_per_household         0.146255
total_rooms                 0.135140
housing_median_age          0.114146
households                  0.064590
total_bedrooms              0.047781
population_per_household   -0.021991
population                 -0.026882
longitude                  -0.047466
latitude                   -0.142673
bedrooms_per_room          -0.259952
Name: median_house_value, dtype: float64
```

→음의 상관관계 25퍼 : 집이 크면클수록 선호

- 데이터 전처리가 중요
  - 수치형 데이터 전처리 과정

(1)누락된 특정값 → (판다스사용) → 해당 구역제거, 전체 특성 삭제, 특정값으로 채우기

(2)

  - 텍스트

원-핫 인코딩

ex)

	ocean	inland	island	near bay	near ocean
ocean	0				
inland					
island					
near bay					
near ocean					

- 나만의 데이터
- 특성 스케일

(1) min-max 스케일링 :

(2) 표준화 :

- 변환 파이프라인

모든 전처리 단계를 정확한 순서, 연속적으로 진행될 수 있도록

→ pipeline 클래스, Column Transformer 클래스

- 훈련 세트에서 훈련,평가하기
  - 선형 회귀 모델 훈련
  - DecisionTreeRegressor
- 교차 검증을 사용한 평가
  - k-겹 교차 검증
  - DecisionTreeRegressor
  - 선형 회귀 모델
  - 앙상블 모델
- 모델 세부 튜닝
  - 가능성 있는 모델들선정 → 튜닝
  - 튜닝방법 :

(1)그리드 탐색(좋은 하이퍼파라미터 조합 찾을때까지 수동 조정, 적은 수 일때)

(2)랜덤 탐색(조합 수 클때)
(3)앙상블