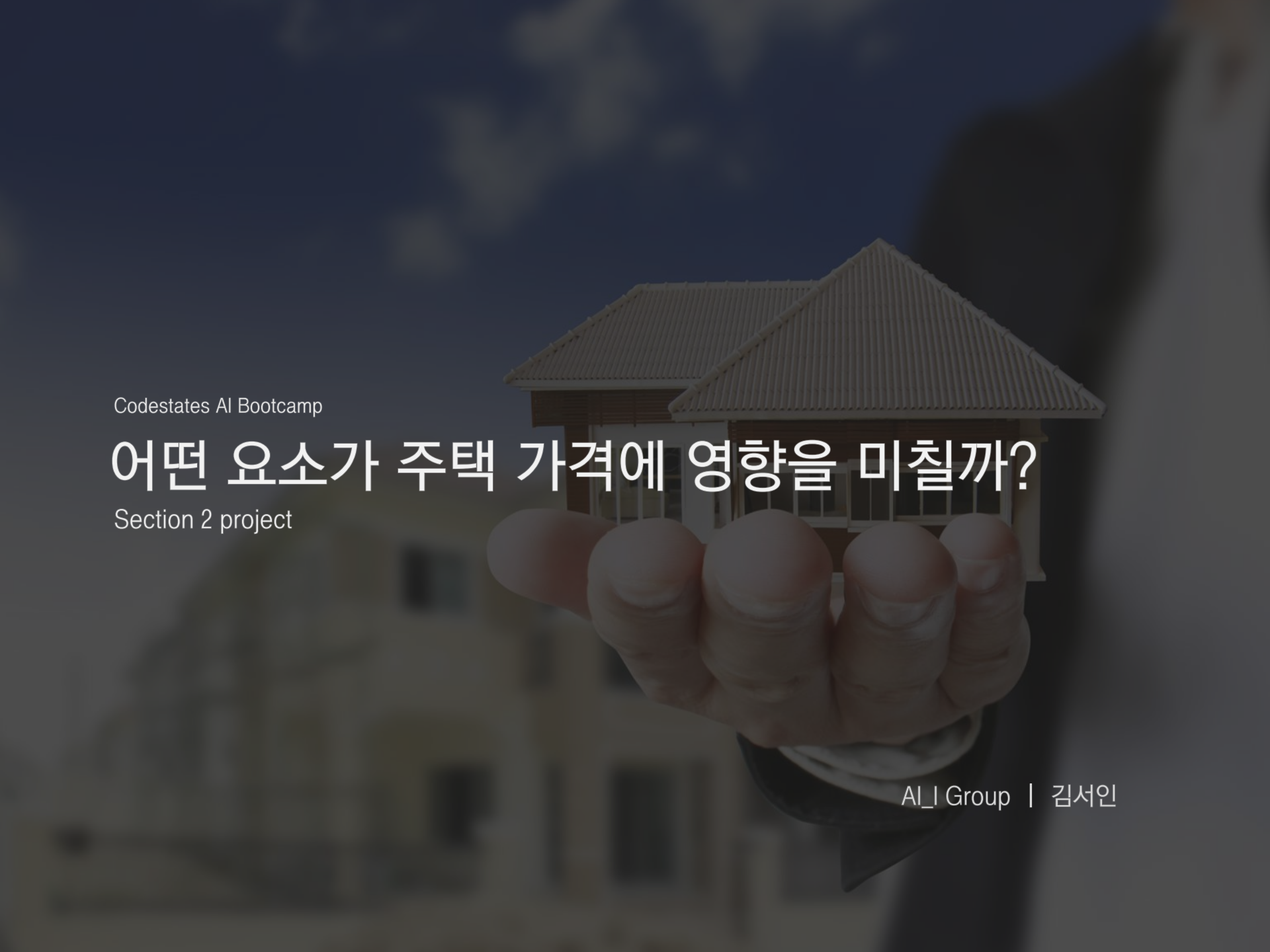


Codestates AI Bootcamp

어떤 요소가 주택 가격에 영향을 미칠까?

Section 2 project

AI_I Group | 김서인

Section 2 Project

INDEX

AI Coding Bootcamp 13th

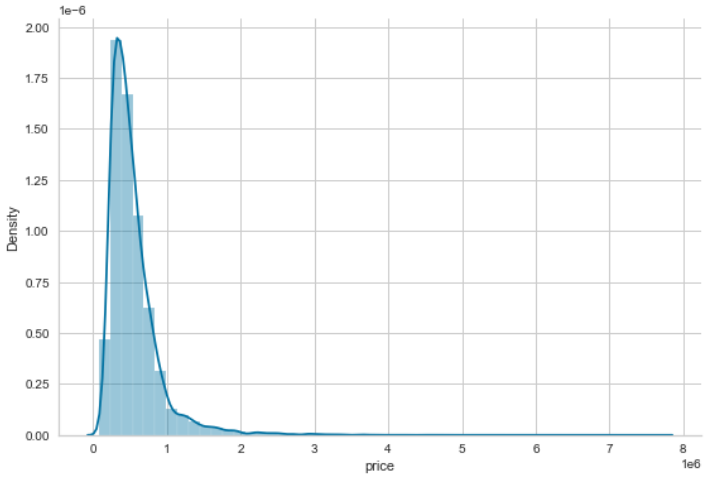

데이터 전처리
Data Preprocessing

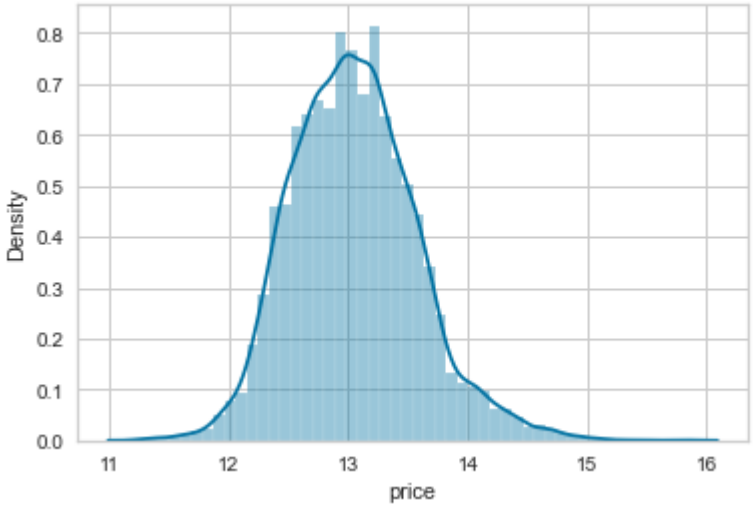
데이터 확인

```
#데이터 살펴보기

house_sales = pd.read_csv('kc_house_data.csv')
house_sales = house_sales.drop(
    ['id', 'zipcode', 'lat', 'long', 'date'], axis=1)
house_sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 16 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   price          21613 non-null  float64
 1   bedrooms       21613 non-null  int64
 2   bathrooms      21613 non-null  float64
 3   sqft_living    21613 non-null  int64
 4   sqft_lot       21613 non-null  int64
 5   floors         21613 non-null  float64
 6   waterfront     21613 non-null  int64
 7   view           21613 non-null  int64
 8   condition      21613 non-null  int64
 9   grade          21613 non-null  int64
 10  sqft_above     21613 non-null  int64
 11  sqft_basement  21613 non-null  int64
 12  yr_built       21613 non-null  int64
 13  yr_renovated   21613 non-null  int64
 14  sqft_living15  21613 non-null  int64
 15  sqft_lot15     21613 non-null  int64
dtypes: float64(3), int64(13)
memory usage: 2.6 MB
```

- 가격, 침실 숫자, 화장실 숫자, 주거 공간 크기, 창고 공간 크기, 층수, 등급, 지어진 연도, 15개의 이웃 주거 공간 평균 등 총 15개 변수






```
#x, y 축 나누기

X_train = train.drop('price', axis=1)
X_val = val.drop('price', axis=1)
X_test = test.drop('price', axis=1)

#Price를 log transformation, 정규분포화

y_train = np.log(train.price)
y_val = np.log(val.price)
y_test = np.log(test.price)
```

» 로그 변환으로 정규분포화

```
#train, test set 나누기

train, test = train_test_split(house_sales, train_size=0.80, test_size=0.20, random_state=2)
```

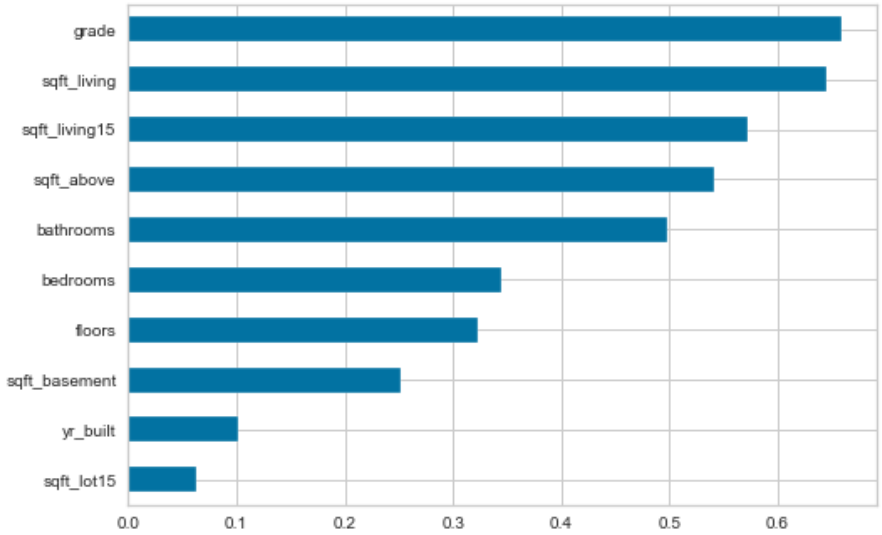
```
train, val = train_test_split(train, train_size=0.80, test_size=0.20, random_state=2)
```

```
train.shape, val.shape, test.shape
```

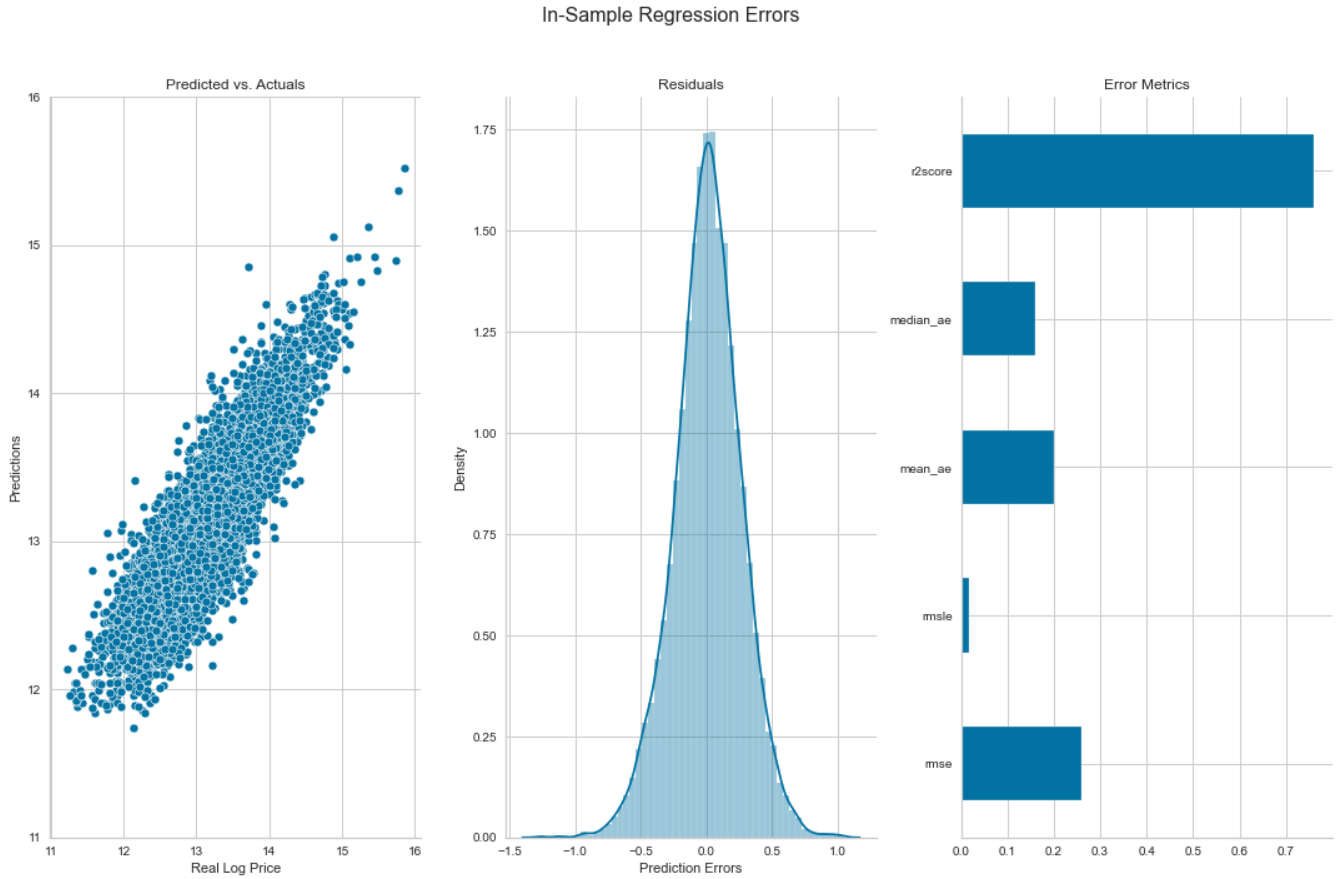
```
((13832, 16), (3458, 16), (4323, 16))
```

- 데이터 training/test 세트를 80%, 20%으로 나누기.
- 거기서 다시 training/validation 세트를 80%, 20%로 나누기

```
# Correlation과 순서 같음

correl = X.apply(lambda x: spearmanr(x, y)[0])
correl.sort_values().plot.barh();
```



- 상관계수 기준으로 설명력 높은 상위 12개만 사용.



- 선형 패턴이 존재하고, 잔차가 정규분포.
- r2 score가 0.7 이상으로 회귀분석이 유의함.

모델 기준

Model Setting

사용 모델 종류

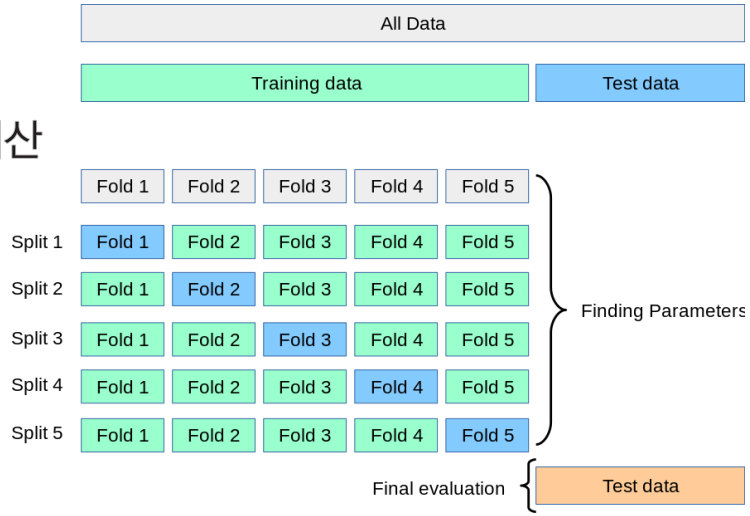
- 릿지 회귀
- K군집 회귀
- 랜덤포레스트 회귀

분석 지표

- Score (MAE)
  - 평균 절대오차

- Cross validation
  - training set에서 dataset을 나누어 score 계산
  - Score의 평균 ex〉 5개의 sample mean

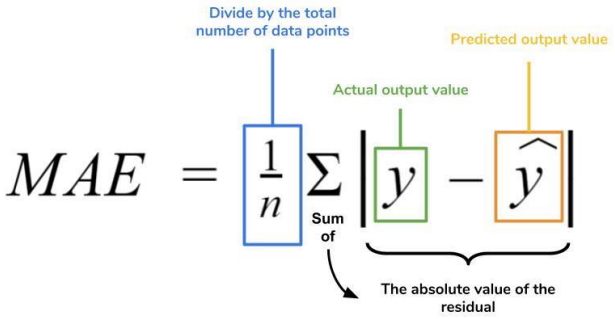
- Pdp, interaction, shap 분석



기준 모델

- 타겟 변수인 Price의 평균을 계산
- 예측 값을 실제 값에서 실제 값의 개수만큼 나눈 것

» MAE(평균절대오차) 0.406



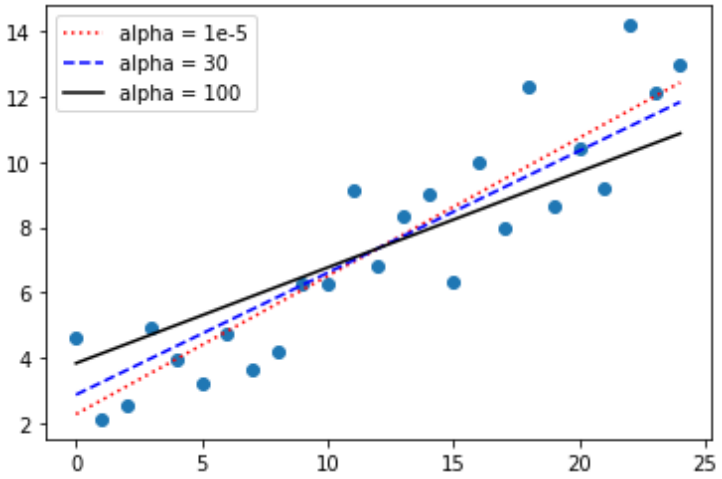
모델 실행

Model implementation

Ridge regression

$$\beta_1, \beta_2, \ldots, \beta_p$$

$$L(\beta) = \min_{\beta} \underbrace{\sum_{i=1} (y_i - \hat{y}_i)^2}_{\text{(1) Training accuracy}} + \lambda \underbrace{\sum_{j=1}^{p} \beta_j^2}_{\text{(2) Generalization accuracy}}$$



- 릿지 회귀함수는 특성변수가 많을 때 과적합이 일어날 수 있어서 특성 변수 정규화를 하는 선형회귀 모델
- Alpha가 크면 변수의 계수가 줄어들어 변수에 penalty가 생김
- 과적합을 피하려고 할 때 주로 사용

최적 alpha는 몇 일지, 그리고 평균절대오차 score는 몇 일지?

최적 Alpha 값

- 0.001
- 가장 낮은 페널티 부여하여 계수를 최대한 사용

Training set MAE(평균절대오차)

- 0.249
- 기준 모델인 0.406보다 크게 낮춤

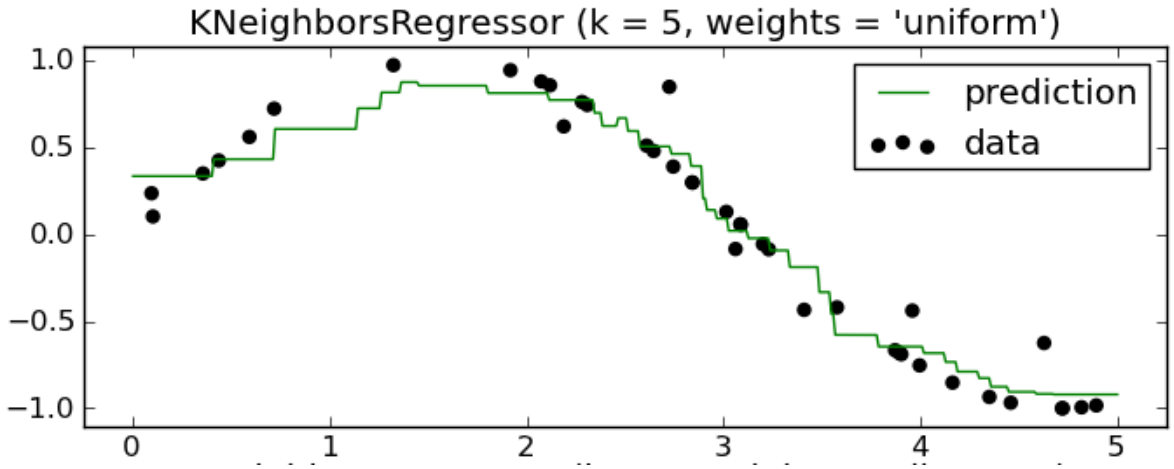
Test set MAE(평균절대오차)

- 0.246

Cross validation MAE(평균절대오차)

- 0.250
- 큰 차이가 없음을 확인

K neighborhood regression



- 거리 공식을 이용해 이웃들의 가중 평균을 계산하는 방식
- 이웃의 개수를 정할 수 있음
- 과적합을 피할 수 있음
- 표본이 많이 없는 구간에 대한 민감도 줄일 수 있음

최적 이웃 k개수, 그리고 평균절대오차 score는 몇 일지?

최적 K 값

- 5개
- 5개의 이웃일 때 error가 가장 낮음

»

Training MAE(평균절대오차)

- 0.199
- 기준 모델인 0.406보다 크게 낮춤

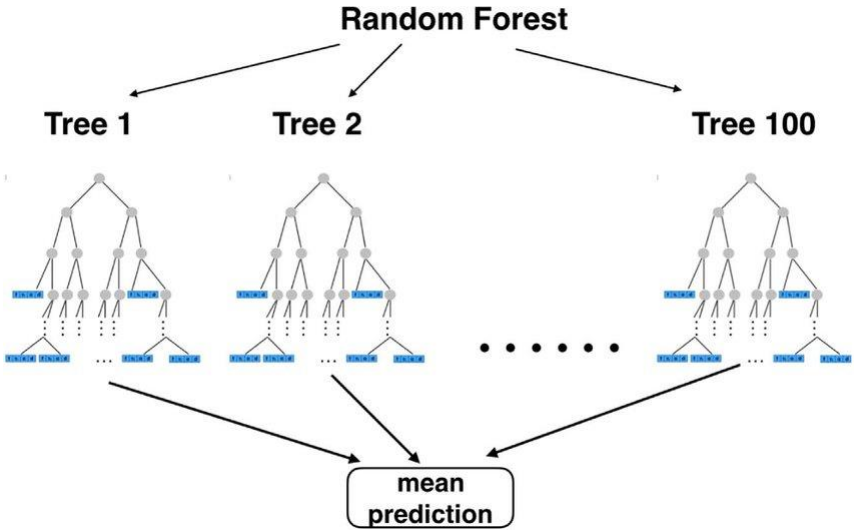
Test set MAE(평균절대오차)

- 0.252

Cross validation MAE(평균절대오차)

- 0.260

Randomforest regression



- 의사결정나무 회귀는 비용함수를 가장 줄여주는 계수를 찾아준다
- 그 의사결정나무 회귀의 평균값 구하는 과정이 랜덤포레스트 방법이다
- 정확도가 상승할 가능성이 높으나 직관적인 설명력은 떨어지는 모델

최적 estimator parameter, 그리고 평균절대오차 score는 몇 일지?

최적 parameter 값

- Max depth none
- N_estimator 454개
- Max features 0.644

Training MAE(평균절대오차)

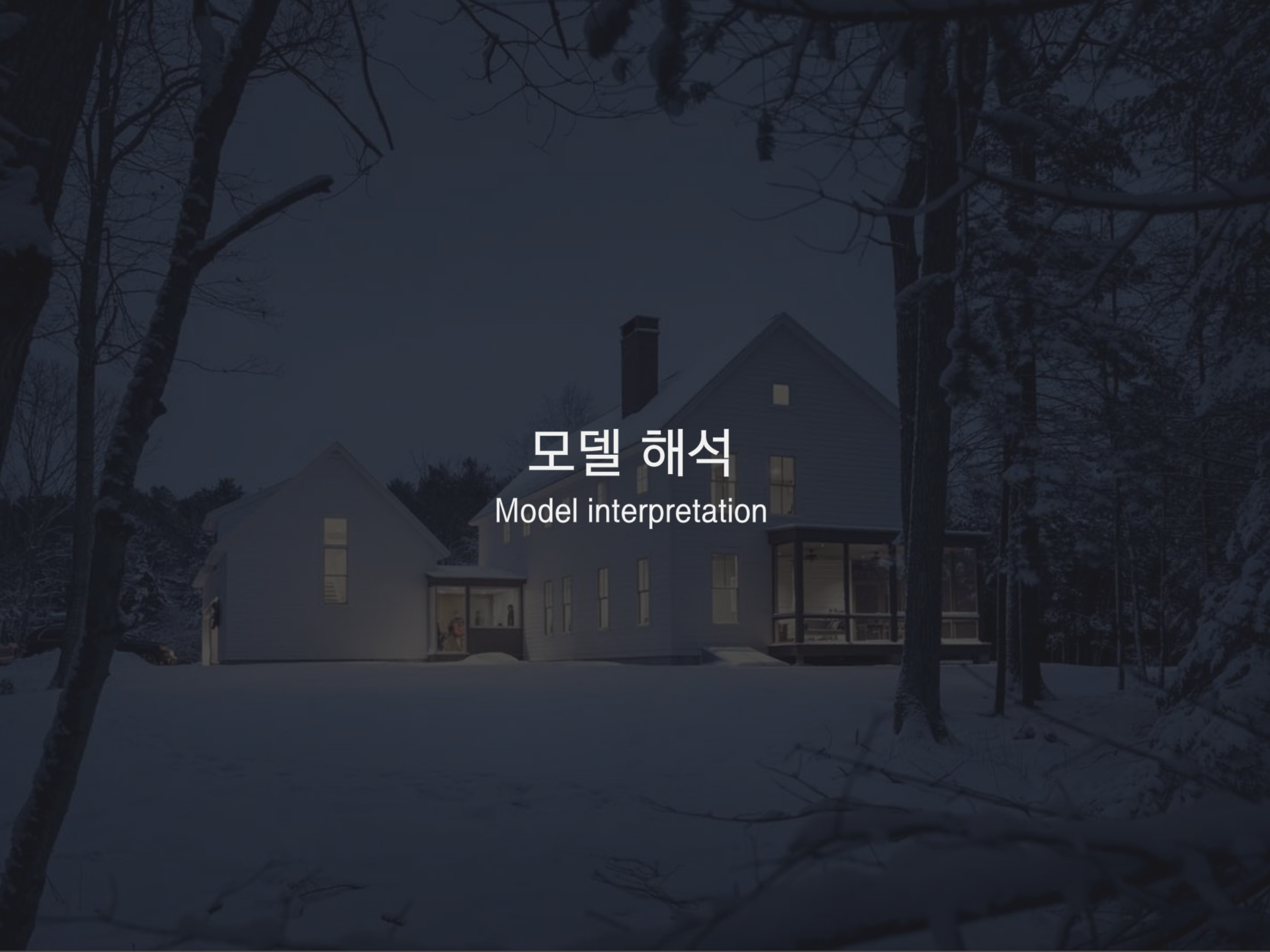
- 0.217
- 기준 모델인 0.406보다 크게 낮춤

Test set MAE(평균절대오차)

- 0.226

Cross validation MAE(평균절대오차)

- 0.252

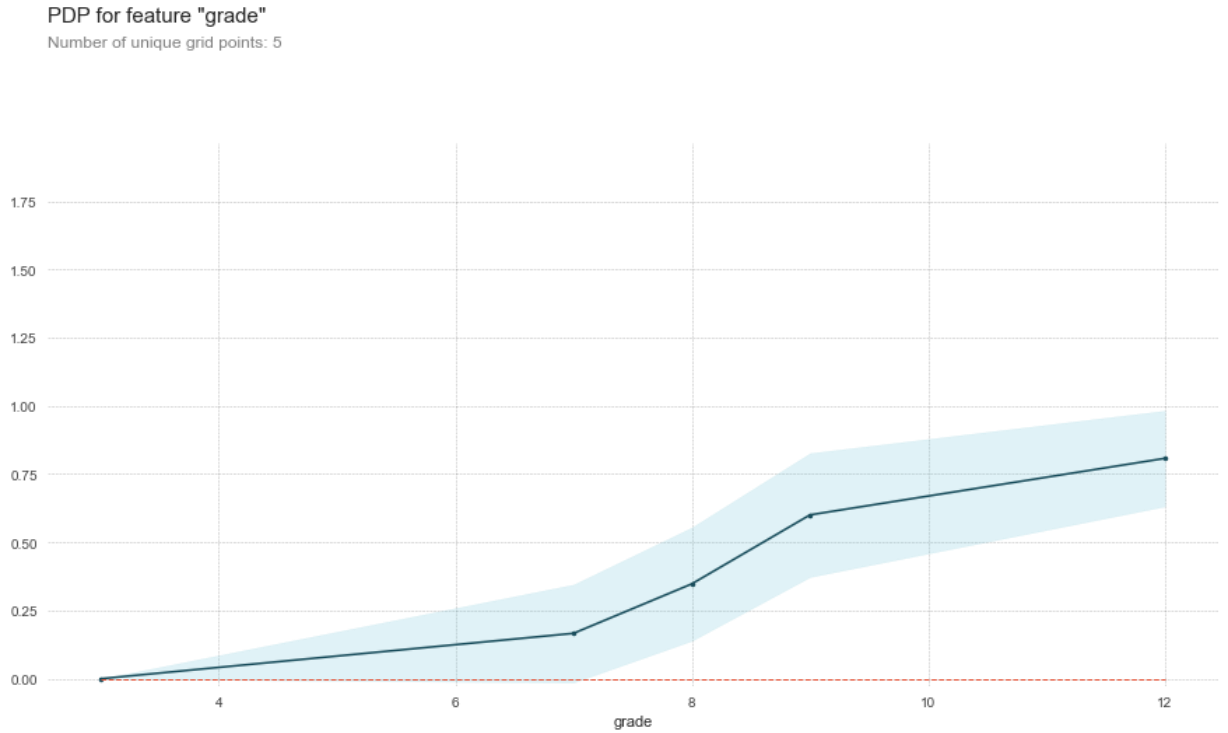


모델 해석

Model interpretation

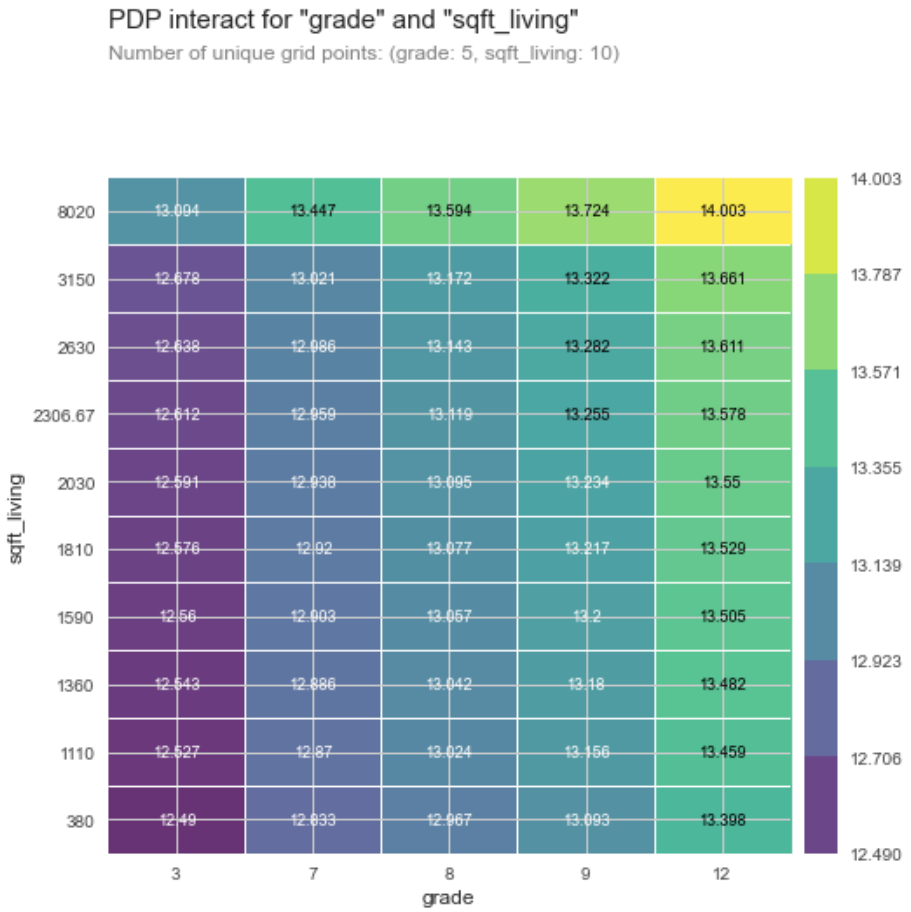
```
In [312]: #순열중요도: 군집 회귀

          feature_names = list(X_val.columns)
          pd.Series(permuter.feature_importances_, feature_names).sort_values(ascending=False)

Out[312]: grade            0.073775
          yr_built         0.067459
          sqft_living15    0.043629
          sqft_living      0.016726
          sqft_above       0.015817
          bathrooms        0.014965
          sqft_basement    0.014530
          floors           0.011943
          bedrooms         0.009857
          view             0.008890
          sqft_lot15       0.005649
          sqft_lot         0.003520
          dtype: float64
```

- 특성값의 중요도를 보여주는 순열중요도 확인 결과
- 주택 등급 점수, 연식, 이웃 주거 면적 평균, 주거 면적 순으로 중요도가 높았음.

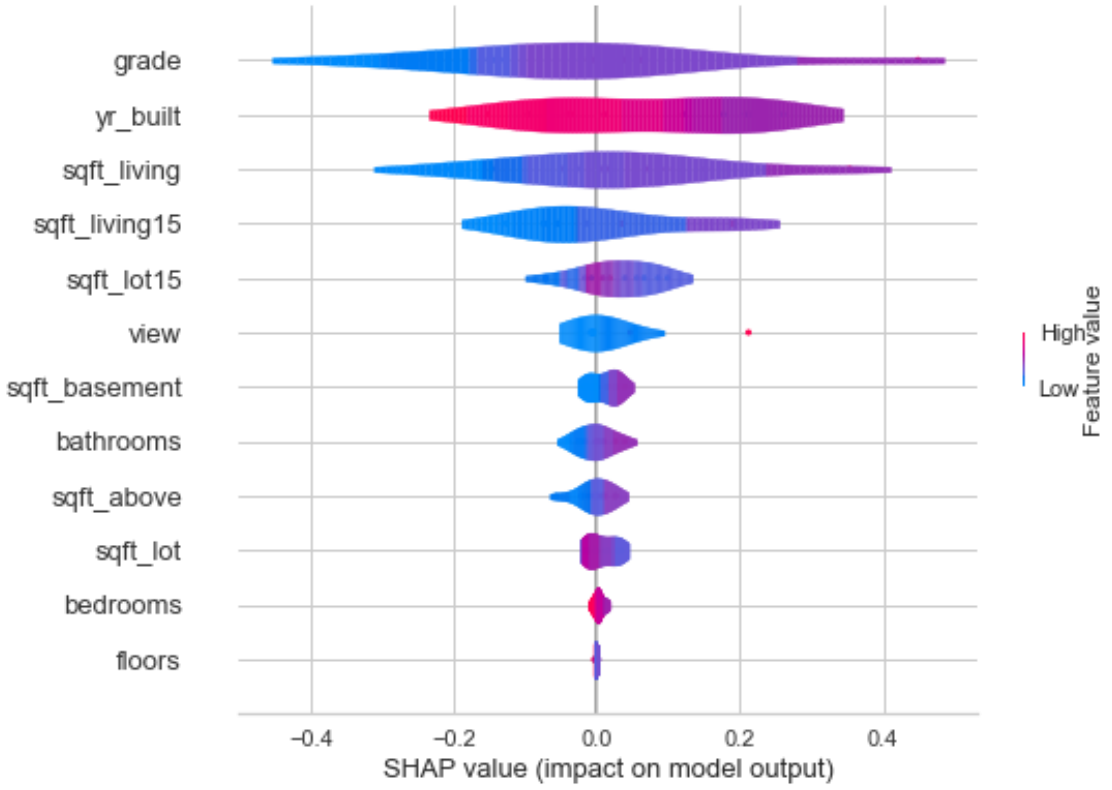


- PDP: 예측모델을 만들었을 때, 어떤 특성(feature)이 예측모델의 타겟 변수(target variable)에 어떤 영향을 미쳤는지 알기 위한 그래프

- Grade 점수와 가격은 양의 선형 관계가 있으며, 7-9점 사이에서 가격이 특히 많이 오름을 알 수 있다.



- 주택 등급 점수와 주거공간 크기 상호작용효과가 존재
- 두 가지 특성 요소가 동시에 작용했을 때 종속변수에 영향을 주는 것으로 등급이 좋고 주거 공간이 더 클 때 가격은 더 많이 상승함을 알 수 있다.



- 랜덤포레스트 테스트 데이터 셋에서 10개의 데이터를 뽑은 결과
- 빨간색이 절대적 영향이 높은 것이고, 보라색은 절대적 영향이 낮은 것
- 연식이 오래된 주택일수록 가격 하락에 절대적 영향이 높았다

결론

Result

01

어떤 주택이 비쌀까?

- 등급 점수가 높은 것
- 연식이 오래되지 않은 것
- 주거 면적이 넓은 곳
- 주거 면적이 충분하게 보장되어 있는 이웃 = 입지

어떤 모델이 좋을까?

- 릿지, K군집 회귀, 랜덤포레스트 회귀 모두 우수한 성능 발휘
- Test set 평균절대오차 점수 기준 낮은 순서대로
  - 랜덤포레스트 〉 릿지 〉 K 군집회귀
- PDP, interaction plot 통해서 예측값에 대한 각 특성 변수 설명력 증명.
- Shap 통해서 관측치에 대한 특성 변수 설명력 증명