

# 집값 회귀 모델들을 활용한 예측보고서

Daeyeop Kim<sup>1\*</sup>

## Abstract

이 프로젝트의 목적은 주어진 데이터를 이용해 여러 회귀 모델을 사용해 집값을 예측을 목표로 하고 있습니다.

## Keywords

집 — 가격 — 예측

<sup>1</sup>Machine Learning, AI 9기, Sparta Coding Camp, Bloomington, Seoul, Korea Republic of

## Contents

1 문제 및 데이터 설명	1
2 데이터 전처리 및 가공(Data Preprocessing) & 데이터 탐색 및 분석(Exploratory Data Analysis)	1
2.1 결측치 확인 및 제거 (Missing value)	1
2.2 이상치 확인 및 제거(Outlier)	2
2.3 탐색적 데이터 분석(Exploratory Data Analysis)	3
3 알고리즘(Algorithm) 과 방법론(Methodology)	4
4 모델 실험들(Model experiment)과 결과들(Model conclusion)	5
5 프로젝트 결론(Project conclusion) 및 요약(Project summary)	7
References	7

## 1. 문제 및 데이터 설명

이 주택 데이터의 주요 문제는 주택 가격을 회귀를 이용해 예측하는 것입니다. 이 프로젝트의 목표는 주택 데이터의 다양한 특성들을 활용하여 주택 판매 가격을 예측하는 것입니다. 이 데이터는 총 506개의 행과 14개의 특성(열)로 구성되어 있습니다. 그러므로 train-test 나누는 비율을 8:2로 측정하였습니다. 또 이 주택 데이터의 14개의 특성이 있는데 위와 같습니다.

이 프로젝트는 이 중 MEDV와 다른 특성들 간의 상관관계를 분석하여 가장 영향력 있는 변수들을 식별하고, 필요에 따라 새로운 특성을 생성하거나 기존 특성을 변환하여 모델을 성능을 향상하여 MSE(평균 제곱 오차), MAE(평균 절대 오차), R<sup>2</sup> score(결정계수). 또한 회귀 분석을 할 때 가장 상관관계가 있는 특징들을 찾아 분석하는 것이 효과적일 수 있기에 상관관계를 측정 후 특징들을 뽑아 회귀분석을 실시할 예정입니다.

## 2. 데이터 전처리 및 가공(Data Preprocessing) & 데이터 탐색 및 분석(Exploratory Data Analysis)

### 2.1 결측치 확인 및 제거 (Missing value)

이 데이터는 열의 수가 그렇게 많지 않아 일단 모든 열의 데이터들의 결측치를 확인하기로 하였습니다. 데이터의 결측

특징	특징 설명
CRIM	도시별 1인당 범죄율
ZN	25,000 평방피트 이상의 주거용 토지 비율
INDUS	비소매 사업 면적 비율
CHAS	Charles River 더미 변수 (강 경계 = 1; 그 외 = 0)
NOX	산화질소 농도
RM	주택당 평균 방 수
AGE	1940년 이전에 건축된 자가소유 주택 비율
DIS	5개 보스턴 고용 센터까지의 가중 거리
RAD	방사형 고속도로 접근성 지수
TAX	10,000달러당 재산세율
PTRATIO	도시별 학생-교사 비율
B	$1000(B_k - 0.63)^2$
Bk	도시별 흑인 비율
LSTAT	인구 중 하위 계층 비율
MEDV	자가소유 주택의 중간 가치(단위: 1,000달러)

Table 1. 데이터 변수 설명

치 확인을 했을 때, CRIM, ZN, INDUS, CHAS, AGE, LSTAT에서 결측치가 각 20개씩 확인되었고 데이터 결측치를 제거하기 위해 먼저 CRIM, ZN, INDUS, CHAS, AGE, LSTAT의 분포를 히스토그램으로 확인했습니다.

\*\*\* histogram for CRIM, AGE, INDUS, LSAT

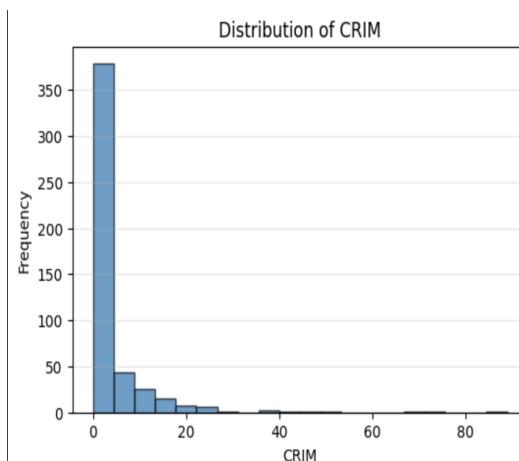
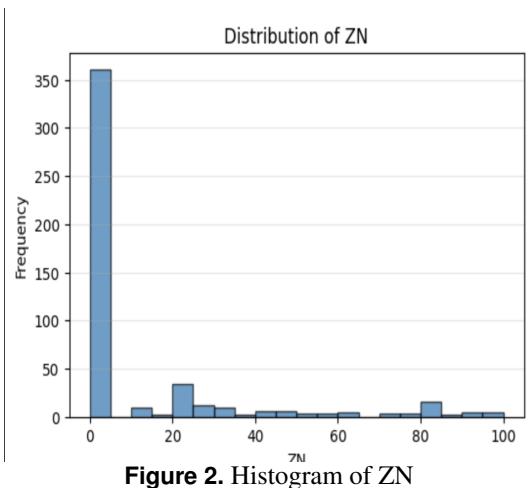
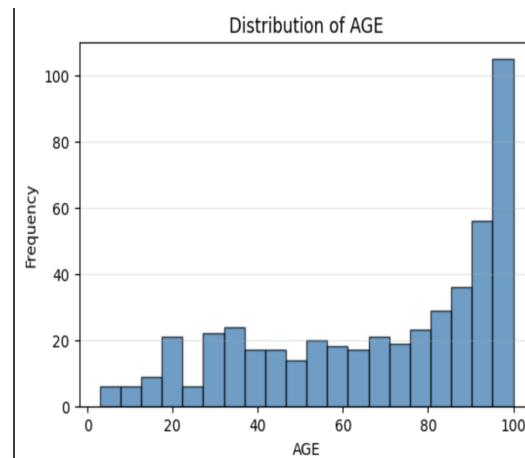
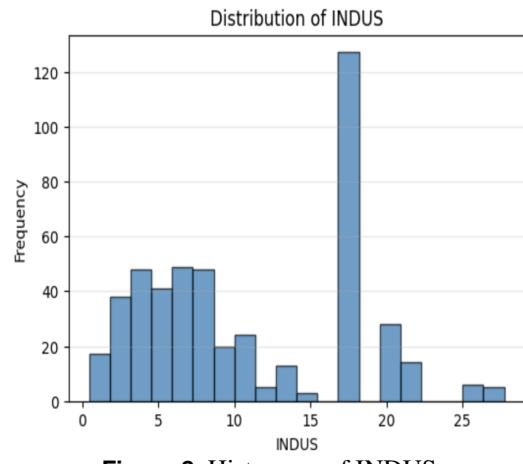
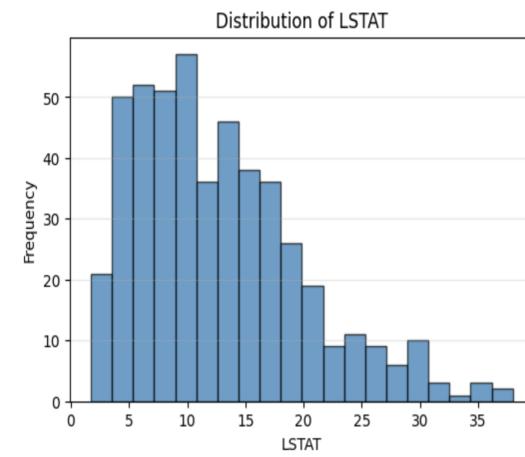
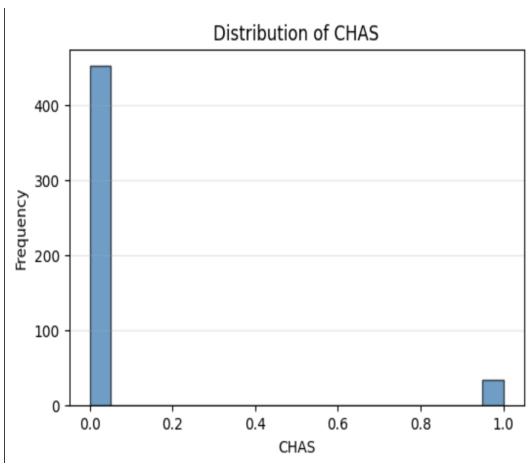


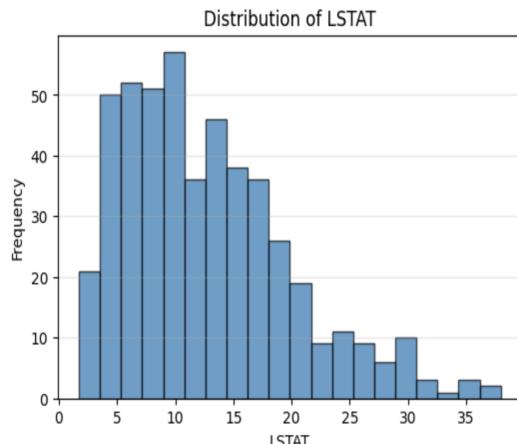
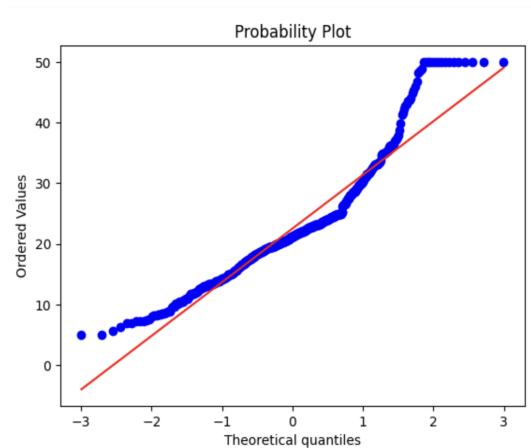
Figure 1. Histogram of CRIM

**Figure 2.** Histogram of ZN**Figure 5.** Histogram of AGE**Figure 3.** Histogram of INDUS**Figure 6.** Histogram of LSTAT**Figure 4.** Histogram of CHAS

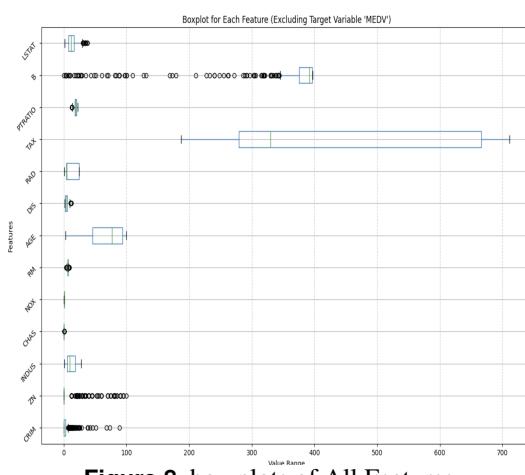
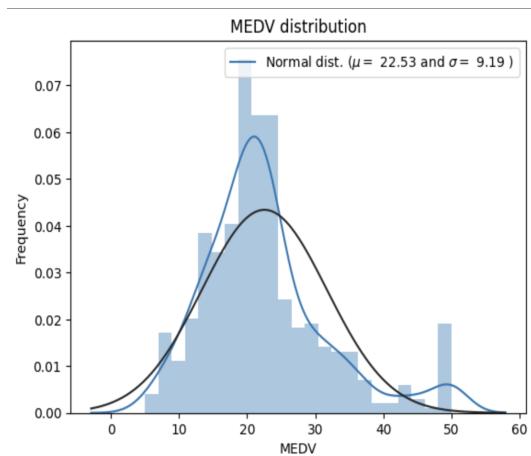
그 결과 CRIM, AGE, INDUS, LSTAT 이 넷의 특성들이 각각 (highly skewed, right-skewed, slight skewed, slight skewed) 하기에 각 특성들의 Median을 넣어 결측치는 제거하였고 나머지 ZN과 CHAS는 0이 많거나 바이너리 값들 이기 때문에 Mode를 넣어 결측치를 제거하였다.

## 2.2 이상치 확인 및 제거(Outlier)

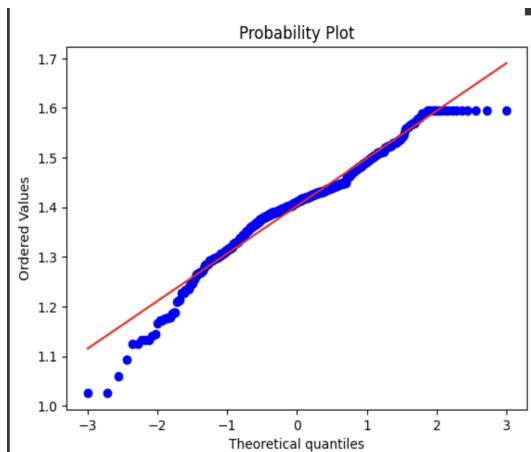
\*\*\*\* boxplot

**Figure 7.** Histogram of LSTAT**Figure 9.** QQplot with data

그후 boxplot을 사용하여 각 특성들의 이상치들을 확인하였고, 데이터의 크기가 작음으로 최대한의 데이터를 살리기위 해 z-score를 사용해 99.7% 의 데이터를 기준으로 이상치를 제거 하여 데이터는 총 401 열, 14 특성을 가진 데이터로 정제했다.

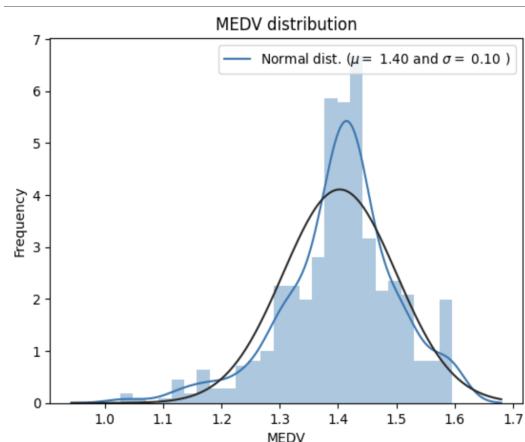
**Figure 8.** box plots of All Features**Figure 10.** Histogram and curve line with Original data

정규화를 확인했는데 데이터가 오른쪽으로 치우져져 있는것 같아 로그변환후 확인하였다.

**Figure 11.** QQ plot with Log data

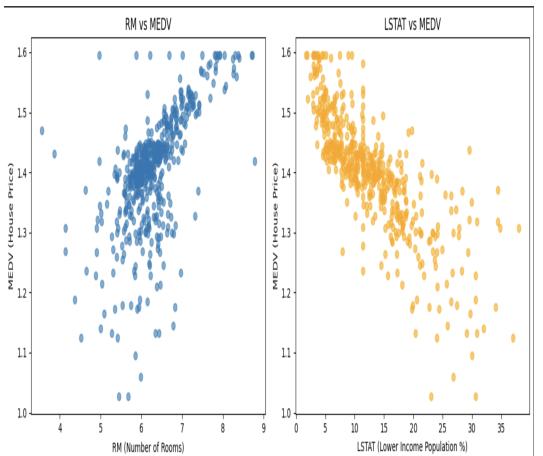
### 2.3 탐색적 데이터 분석(Exploratory Data Analysis)

우선 데이터의 정규화를 확인하기 위해, MEDV 분포 히스토그램과 정규분포 곡선과 QQ-plot을 이용하였다.



**Figure 12.** Histogram and curve line with Standardized data

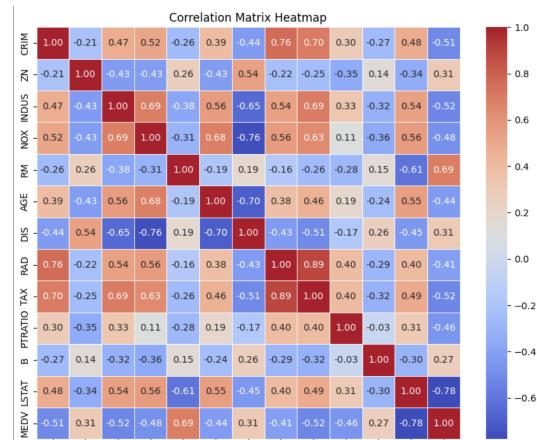
데이터를 정규화 시키고, MEDV와 RM과 LSTAT의 상관관계를 scatterplot을 이용해 데이터를 파악했다.



**Figure 13.** Scatter plot of LSTAT or RM with MEDV

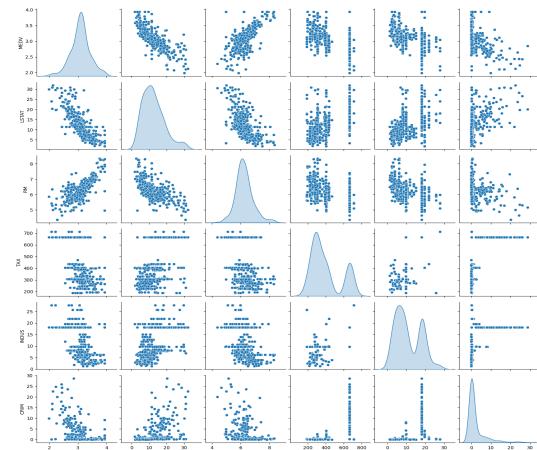
scatterplot 을 이용해 MEDV 가 RM 즉 방의 갯수와 양의 관계가 있다고 판단되며, LSTAT 즉 저소득층 비율과 MEDV가 음의 관계를 볼수 있다. 그러므로 MEDV, 즉 집값이 방의 갯수에 따라 증가하는 방향으로 보이며, 저소득층일수록 집 값이 더 낮은곳에 살고 있다고 볼수 있다.

이후 MEDV와 다른 특성들의 상관관계를 보기 위해 correlation matrix을 구현하였다.



**Figure 14.** Correlation table with MEDV

이를 통해 MEDV와 가장 연관성을 가지고 있는 특성들은 LSTAT, RM, RM, INDUS, CRIM 으로 나타날수 있다.



**Figure 15.** 연관성 있는 특징들과 MEDV의 상관 그래프

위의 pairplot을 통해 MEDV와 가장 연관성을 가지고 있는 특성들을 보면, 저소득층 비율이 높아질수록 주택가격이 감소하는 경향을 볼수 있고 앞으로 선형 모델에 적합한 패턴일 가능성이 있습니다. 방의 개수는 많을수록 주택가격이 증가하는 경향이 뚜렷한걸 볼수 있고, 이 또한 선형모델에 적합한 패턴으로 보입니다. 범죄율이 높아질수록 주택가격이 낮아지는 경향이 있지만 명확한 패턴은 나타나지 않고 세금과 비소매 상업 지역 비율 사이의 관계가 다중공선성을 유발할 가능성이 보여 변수중 하나를 제거하거나 PCA(주성분 분석)을 활용할 필요도 있을꺼 같다. 그리고 곡선들을 보았을때 모델을 시도하기전 특성별 정규화가 필요한걸 알아낼수 있었다.

### 3. 알고리즘(Algorithm) 과 방법론 (Methodology)

우선 주택 판매 가격을 예측하기 위해 총 5가지의 알고리즘 (Linear Regression, Decision Tree Regressor, Random Forest

Regressor, Gradient Boosting Regressor, XGBoost)를 이용하였습니다.

주택 판매 가격을 예측하기 위해 먼저 선형 회귀(Linear Regression)를 사용했습니다.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

이 알고리즘은 오차 제곱합(Sum of Squared Errors)을 최소화하여 훈련 데이터에 가장 적합한 직선을 생성합니다. 생성된 회귀선을 통해 테스트 데이터의 값을 예측하며, 간단하고 해석 가능한 기준 모델로 활용됩니다.

다음으로 의사결정 나무 회귀 모델(Decision Tree Regressor)를 적용했습니다.

$$\text{Split Criterion: } \operatorname{argmin}_{s,t} \sum_{i=1}^m \text{Impurity}(t_i)$$

$$\text{Gini}(t) = 1 - \sum_{i=1}^k p_i^2$$

이 모델은 데이터를 특징 값에 따라 분기(branch)시키고, 리프 노드(leaf node)에서 훈련 데이터의 평균값을 계산하여 예측값을 출력합니다. 단순한 구조지만, 적절히 가지치기(pruning)하지 않으면 과적합(overfitting)이 발생할 수 있습니다.

또한 랜덤 포레스트 회귀 모델(Random Forest Regressor)를 사용했습니다.

$$\hat{y} = \frac{1}{N} \sum_{i=1}^N T_i(x)$$

이 모델은 다수의 의사결정 나무를 기반으로 예측값을 집계하는 양상을 방법으로, 단일 결정 나무보다 과적합을 줄이고 예측 정확도를 높입니다. 각 나무의 예측 결과를 평균 내어 최종 예측값을 산출합니다.

예측을 더 개선하기 위해 그레이디언트 부스팅 회귀 모델(Gradient Boosting Regressor)을 사용했습니다.

$$F_m(x) = F_{m-1}(x) + \alpha \cdot h_m(x)$$

$$h_m(x) = \operatorname{argmin}_h \sum_{i=1}^n \ell(y_i, F_{m-1}(x_i) + h(x_i))$$

이 모델은 반복적으로 여러 결정 나무를 학습시키면서 이전 모델의 오차를 줄이는 방향으로 최적화합니다. 각 단계에서 오차를 보정하기 때문에 데이터의 복잡한 패턴을 포착하는 데 효과적입니다.

마지막으로 XGBoost를 활용했습니다.

$$\mathcal{L} = \sum_{i=1}^n \ell(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|w\|^2$$

이는 Gradient Boosting의 효율적인 구현체로, 규제(regularization)와 병렬 처리와 같은 고급 기술을 통해 성능을 최적화합니다. XGBoost는 강력한 특징 처리 기능과 계산 속도 향상을 바탕으로 높은 정확도를 자랑하며, 예측 모델링 작업에 적합합니다.

이 모델들을 조합해 간단한 선형 예측부터 복잡한 양상을 방법까지 다양한 접근 방식을 평가했으며, 주택 판매 가격 예측에 가장 적합한 모델을 도출할 수 있었습니다.

## 4. 모델 실험들(Model experiment)과 결과들 (Model conclusion)

모델들을 RMSE과 R2를 비교

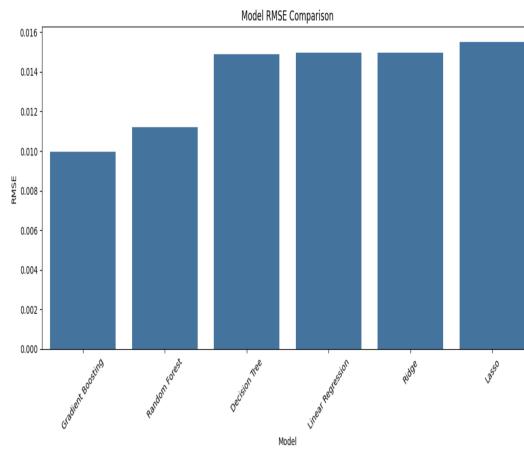


Figure 16. 모델별 RMSE 비교

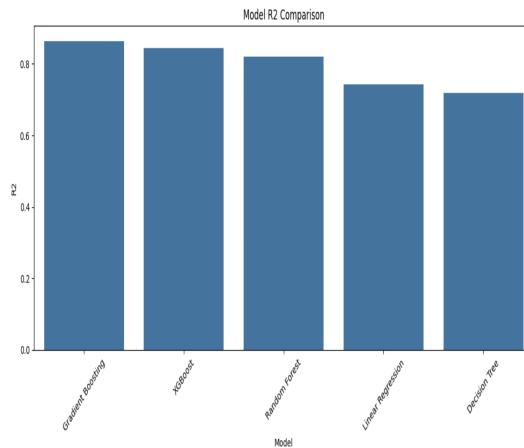


Figure 17. 모델별 R2 Score 비교

이 그래프를 볼 때 GradientBoost 모델의 RMSE, 즉 모델의 예측값과 실제값 사이의 오차를 측정한 값인데 MSE 보다 이상치에 덜 예민하여 RMSE가 낮다는 것은 예측값과 실제값의 차이가 가장 작았고, R-score, 즉 결정 계수는 모델이 데이터를 얼마나 잘 설명하는지를 나타내는 지표로써 1에 가까울수록 모델의 예측이 데이터의 변동성을 잘 설명하기에 가장 정확도가 높은 모델로 보인다. 아무래도 Gradient Boosting은 오차를 줄이기 위해 점진적으로 학습하며 학습 할 때 상호작용과 비선형 관계를 잘 잡기에 낮은 RMSE와 높은 R-score 추측된다.

5개의 모델 정확도	
모델명	정확도(%)
Gradient Boosting	97.931892%
XGBoost	97.918787%
Random Forest	97.776112%
Linear Regression	97.444976%
Decision Tree	97.309176%

평균적으로 모든 모델들이 97% 이상으로 좋은 결과가 나왔다 그중에서 Gradient Boosting 가 최선의 모델로 확인이 되었다. 모든 모델 중에서 Gradient Boosting Regressor가 가장 뛰어난 성능을 보였으며, 교차 검증에서 약 97.93%의 정확도를 기록했습니다. 이는 Gradient Boosting 알고리즘이 복잡한 패턴을 처리하는 데 강점을 가지고 있다는 점과 최근 데이터 과학 커뮤니티에서 이 알고리즘이 인기를 끌고 있는 이유를 잘 보여줍니다.

두 번째로 성능이 좋았던 모델은 XGBoost로, 교차 검증에서 약 97.92%의 정확도를 기록했습니다. Gradient Boosting보다 약간 낮은 정확도를 보였지만, XGBoost는 효율성과 성능 최적화 측면에서 강력한 기능을 제공하여 일관된 결과를 보여줍니다.

Random Forest Regressor는 97.78%의 정확도를 기록하며 우수한 결과를 냈습니다. 이 모델은 또 다른 결정 나무 기반 양상을 모델로, 과적합을 줄이는 데 효과적이지만 Gradient Boosting이나 XGBoost처럼 반복적인 개선 과정을 거치지 않기 때문에 성능이 약간 낮게 나왔을 가능성이 있습니다.

선형 회귀(Linear Regression) 모델은 97.44%의 정확도를 기록하며 비교적 높은 성능을 보여주었습니다. 단순한 모델임에도 불구하고 데이터의 상당 부분을 잘 설명할 수 있었습니다.

의사결정 나무(Decision Tree Regressor)는 97.31%의 교차 검증 정확도를 기록하며 이 중 가장 낮은 성능을 보였습니다. 단독으로 사용되는 결정 나무는 과적합이 발생하기 쉬워, 다른 양상을 모델보다 낮은 결과를 보인 것으로 보입니다.

Gradient Boosting Regressor가 가장 우수한 성능을 보였기 때문에 이 모델을 테스트 세트 예측에 활용했습니다. 그 후 혹시나 Ensemble이나 Hyperparameter Tuning을 통해 정확도를 더 높일 수 있을까 의문이 들어 가장 좋은 모델인 Gradient Boosting 을 Greed Search 와 Randomized Search 를 시도를 해보았다.

HyperParameter Tuned Models and GBM		
모델명	RMSE	R-Score
Gradient Boosting	0.043112	0.862853
Grid Search Gradient Boosting	0.043736	0.858854
Randomized Search Gradient Boosting	0.046716	0.838963

Table 2. Comparison of Gradient Boosting Model and Hyperparameter Tuned Model

위의 표를 보았을때 hyperparameter tuned 모델들과 기존 모델 중에서 Gradient Boosting이 가장 뛰어난 성능을 보였습니다. 이 모델의 RMSE(Root Mean Square Error)는

0.0431로, 가장 낮은 오차를 기록했으며, R-Score(결정 계수)는 0.8629로 가장 높은 정확도를 보였습니다. 이는 Gradient Boosting이 복잡한 데이터 패턴을 잘 학습하고 안정적인 성능을 제공한다는 것을 보여줍니다.

다음으로는 Grid Search Gradient Boosting의 RMSE 0.0437과 R-Score 0.8589를 기록하며, 약간 낮은 성능을 보였습니다. Grid Search를 통해 하이퍼파라미터를 최적화했음에도 기본 Gradient Boosting보다 성능이 약간 떨어졌는데, 이는 특정 데이터 세트에서 기본 설정이 이미 잘 작동했음을 시사합니다.

마지막으로 Randomized Search Gradient Boosting은 RMSE 0.0467과 R-Score 0.8390으로 가장 낮은 성능을 보였습니다. Randomized Search는 하이퍼파라미터 최적화에서 무작위 샘플링을 사용하기 때문에 Grid Search보다 최적의 하이퍼파라미터를 찾지 못한 것으로 보입니다. 하지만 이 방법은 계산 시간이 상대적으로 짧아, 자원이 제한된 환경에서 유용할 수 있습니다.

결과적으로 Gradient Boosting 모델이 가장 우수한 성능을 보였으며, Grid Search와 Randomized Search를 활용한 모델은 약간의 성능 차이를 보였지만 Ensemble Learning 시도해보았습니다. 그래서 hyperparameter Tuning 으로써는 큰 진척이 없어, 기존의 가장 좋은 3 가지 모델들(Gradient Boosting, XGBoost, Random Forest)을 조합하여 Ensemble Learning 시도해보았다. 위의 표를

Emsemble Learning Models and GBM		
모델명	RMSE	R-Score
Gradient Boosting	0.043112	0.862853
Voting Regressor	0.043943	0.857518
Stacking Regressor	0.044202	0.855830

Table 3. Emsemble Learning Model and Original Model Performance Comparison

보았을때 Ensemble과 기존 모델 중에서 Gradient Boosting이 가장 뛰어난 성능을 보였습니다. 이 모델의 RMSE(Root Mean Square Error)는 0.0431로, 가장 낮은 오차를 기록했으며, R-Score(결정 계수)는 0.8629로 가장 높은 정확도를 보였습니다. 이는 Gradient Boosting이 복잡한 데이터 패턴을 잘 학습하고 안정적인 성능을 제공한다는 것을 보여줍니다.

다음으로는 Voting Regressor의 RMSE 0.0439과 R-Score 0.8585를 기록하며, 약간 낮은 성능을 보였습니다. Grid Search를 통해 하이퍼파라미터를 최적화했음에도 기본 Gradient Boosting보다 성능이 약간 떨어졌는데, 이는 특정 데이터 세트에서 기본 설정이 이미 잘 작동했음을 시사합니다.

마지막으로 Stacking Regressor은 RMSE 0.0442과 R-Score 0.856으로 가장 낮은 성능을 보였습니다. Randomized Search는 하이퍼파라미터 최적화에서 무작위 샘플링을 사용하기 때문에 Grid Search보다 최적의 하이퍼파라미터를 찾지 못한 것으로 보입니다. 하지만 이 방법은 계산 시간이 상대적으로 짧아, 자원이 제한된 환경에서 유용할 수 있습니다.

결과적으로 Gradient Boosting 모델이 가장 우수한 성능을 보였으며, Grid Search와 Randomized Search를 활용한 모델은 약간의 성능 차이를 보였지만 Ensemble Learning 또한 hyperparameter tuning 과 비슷하게 RMSE는 떨어지고 R-Score이 높아지며, 오히려 모델의 성능이 떨어지는 것을

보였다.

## 5. 프로젝트 결론(Project conclusion) 및 요약(Project summary)

이번 프로젝트의 목표는 주택 판매 가격을 예측하는 것입니다. 데이터셋에는 14 개의 속성이 있었으며, 목표 변수인 판매가격 (MDVE)은 제외하였습니다. 각 속성들은 판매된 주택의 특징과 연관성을 확인을 통해 LSTAT, RM, TAX, INDUS, CRIM 특성들이 좋은 연관성이 있었습니다.

데이터 전처리 및 데이터 분석을 통해, 결측값 처리는 각 특징의 skewedness 정도를 측정하여 알맞게 mean, median, mod 를 기입해 처리를 하였고 CHAS 속성이 카테고리 특성이 있기에 제외를 시켰으며, 상관관계가 특징중 높은 5가지 특징들 중심으로 분석하였습니다.

모델링 및 결과를 보았을때, 이 프로젝트에서 사용된 모델들, Linear regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost 들을 활용해서 판매가격을 예측했습니다. 모델의 성능을 더 높이기위해 시도한 모델중 가장 좋은 결과를 가진 Gradient Boosting을 사용하여 Grid Search 와 Randomized Search 를 이용하여 시도를 해보았지만 더 좋은 결과는 없었고 그래서 가장 좋았던 3 모델들을 이용해, Ensemble learning, Voting Regressor 과 Stacking Regressor 을 사용해 시도를 했지만 유의미한 결과가 없었다. 이를 통해 정확도가 높아질수록 Hypertuning 과 Ensemble Learning 을 통한 정확도 혹은 모델의 성능 증가는 무의미 할수 있을을 알게되었다. 이 모델의 정확도를 높이기위해 한가지의 방안은 앞으로 CHAS 특성을 HotEncoding등 을 해서 연관성이 괜찮다면 모델을 사용할때 더 추가하여 정확도를 높일수 있을것 같다.

## References