Module 5 Project | Housing | Part 2

Trang Tran

CPS Analytics, Northeastern University

ALY6020 | Predictive Analytics

Professor Prashant Mittal

Dec 15, 2023

Introduction

The real estate landscape in recent years, marked by challenges such as high mortgage rates and housing shortages, presents a compelling opportunity for predictive analysis. Focusing on Maine and New Hampshire, this study aims to predict property prices by exploring house features and socio-economic factors, particularly health outcomes. Our dataset comprises current listings sourced from Redfin and other tables of population and health data. With objectives set on thorough data preprocessing and predictive modeling, Part 2 builds decision tree, random forest, and boosted tree models, fitting the categorical and continuous 'PRICE' response separately. Based on multiple benchmarking metrics used to compare models with different choices of features and hyperparameters, we will discuss which model is most useful in predicting house prices.

Data Description and Preprocessing

The cleaned Redfin dataset consists of 2376 rows and 82 columns. I chose to keep 22 appropriate columns as below for predictive models, in which 'PRICE' is a target variable, and the remaining are predictors.

```
redfin_2.columns
```

```
Index(['PROPERTY TYPE', 'STATE OR PROVINCE', 'ZIP OR POSTAL CODE', 'PRICE',
       'BEDS', 'BATHS', 'SQUARE FEET', 'YEAR BUILT', 'DAYS ON MARKET',
       'LATITUDE', 'LONGITUDE', 'TotalPopulation', 'ACCESS2_AdjPrev',
       'ARTHRITIS_AdjPrev', 'CANCER_AdjPrev', 'CSMOKING_AdjPrev',
       'DEPRESSION_AdjPrev', 'DIABETES_AdjPrev', 'OBESITY_AdjPrev',
       'MHLTH_AdjPrev', 'PHLTH_AdjPrev', 'PRICE_BINARY'],
      dtype='object')
```

The first step is to recategorize the target variable into binary classes by adding a column named 'PRICE_BINARY' with values '1' for prices above the median and values '0' for prices below the median.

The next step is to check NAs per column. The ‘LOT SIZE’ variable has the highest NAs of 431 rows (~18% of the dataset), so we decided to drop off this variable. Afterward, we cleared the remaining NAs in the dataset. The final dataset has 2297 observations and 22 rows.

```
redfin_2.isnull().sum()
```

```
PROPERTY TYPE           0
STATE OR PROVINCE       0
ZIP OR POSTAL CODE      0
PRICE                   0
BEDS                   13
BATHS                  16
SQUARE FEET             6
LOT SIZE              431
YEAR BUILT             44
DAYS ON MARKET          0
LATITUDE                0
LONGITUDE               0
TotalPopulation         0
ACCESS2_AdjPrev         0
ARTHRITIS_AdjPrev       0
CANCER_AdjPrev          0
CSMOKING_AdjPrev        0
DEPRESSION_AdjPrev      0
DIABETES_AdjPrev        0
OBESITY_AdjPrev         0
MHLTH_AdjPrev           0
PHLTH_AdjPrev           0
PRICE_BINARY            0
dtype: int64
```

With the categorical variables ‘PROPERTY TYPE’ and ‘STATE OR PROVINCE’, we did one-hot encoding to create binary "dummy" variables that represent each category within the original categorical variables. Then we dropped off the original categorical ones in the model. The new dataset for training includes a total of 31 columns.

Models for BINARY PRICE response

After splitting the data set into the train and test sets with a ratio of 80%-20%, respectively, fitting the Decision Tree, Random Forest, Gradient Boosting Classifier, and XGBoost classification models on the train set and running the predicting process on the test set, we have the results of four models as follows. To be noted, we did a Randomized Grid Search for the Random Forest model and fit the best model to predict the test set.

	precision	recall	f1-score	support
0	0.81	0.86	0.83	237
1	0.84	0.78	0.81	223
accuracy			0.82	460
macro avg	0.82	0.82	0.82	460
weighted avg	0.82	0.82	0.82	460

y_pred_dt \ y_test	0	1
0	203	49
1	34	174

Decision Tree classification

	precision	recall	f1-score	support
0	0.84	0.84	0.84	237
1	0.83	0.83	0.83	223
accuracy			0.83	460
macro avg	0.83	0.83	0.83	460
weighted avg	0.83	0.83	0.83	460

y_pred_rf \ y_test	0	1
0	198	37
1	39	186

Random Forest classification

	precision	recall	f1-score	support
0	0.85	0.84	0.85	237
1	0.83	0.84	0.84	223
accuracy			0.84	460
macro avg	0.84	0.84	0.84	460
weighted avg	0.84	0.84	0.84	460

y_pred_gb \ y_test	0	1
0	199	35
1	38	188

Gradient Boosting Classification

	precision	recall	f1-score	support
0	0.83	0.84	0.84	237
1	0.83	0.82	0.82	223
accuracy			0.83	460
macro avg	0.83	0.83	0.83	460
weighted avg	0.83	0.83	0.83	460

y_pred_xgb \ y_test	0	1
0	200	41
1	37	182

XGBoost classification

Figure 1: Classification Model Results

The four models above show similarities in accuracy, in which the gradient-boosting classification model has the highest scores, and the decision tree model has the lowest result in terms of multiple metrics. The binary response is relatively balanced, so we do not need further techniques to handle this issue. Figure 2 below presents the feature importance of the random forest model, which indicates ‘Square feet” is the most important feature in the classification prediction model (> 0.175 weight), followed by the Number of Baths, Year built, Latitude, Current rate of Smoking, and other health factors.

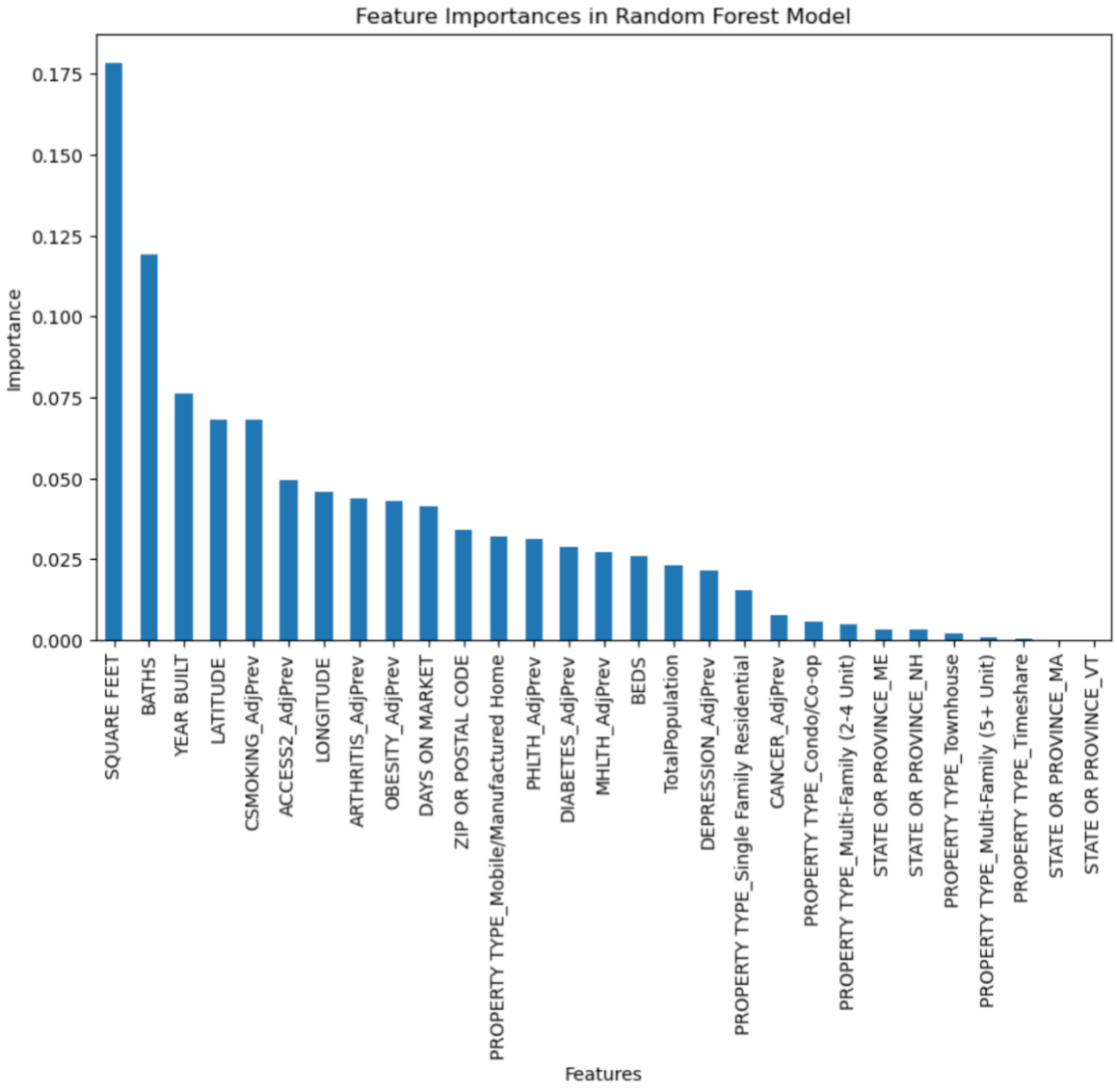


Figure 2: Feature Importance in Random Forest Classification Model

Models for CONTINUOUS PRICE response

Similarly, we follow the steps to use Decision Tree, Random Forest, Gradient Boosting Classifier, and XGBoost regression models to train for the continuous price response. The XGBoost model's result is the result of the best model of hyperparameter tuning using a Randomized Grid Search. Among four regression models, XGBoost with hyperparameter tuning shows the best performance (> 66% of R-squared) in predicting continuous prices. The feature

'Baths' becomes the most important feature in the random forest regression model (~0.3 weight of the model).

```
Decision Tree R-squared: 0.4365859033341871
Decision Tree Root Mean Squared Error: 227710.87855005777

Random Forest R-squared (R2): 0.6428518124254254
Random Forest Root Mean Squared Error (RMSE): 181298.474457738

Gradient Boosting R-squared (R2): 0.6580876420185648
Gradient Boosting Root Mean Squared Error (RMSE): 177389.26060745018

XGBoost R-squared (R2): 0.6622795914716593
XGBoost Root Mean Squared Error (RMSE): 176298.48385153743
```

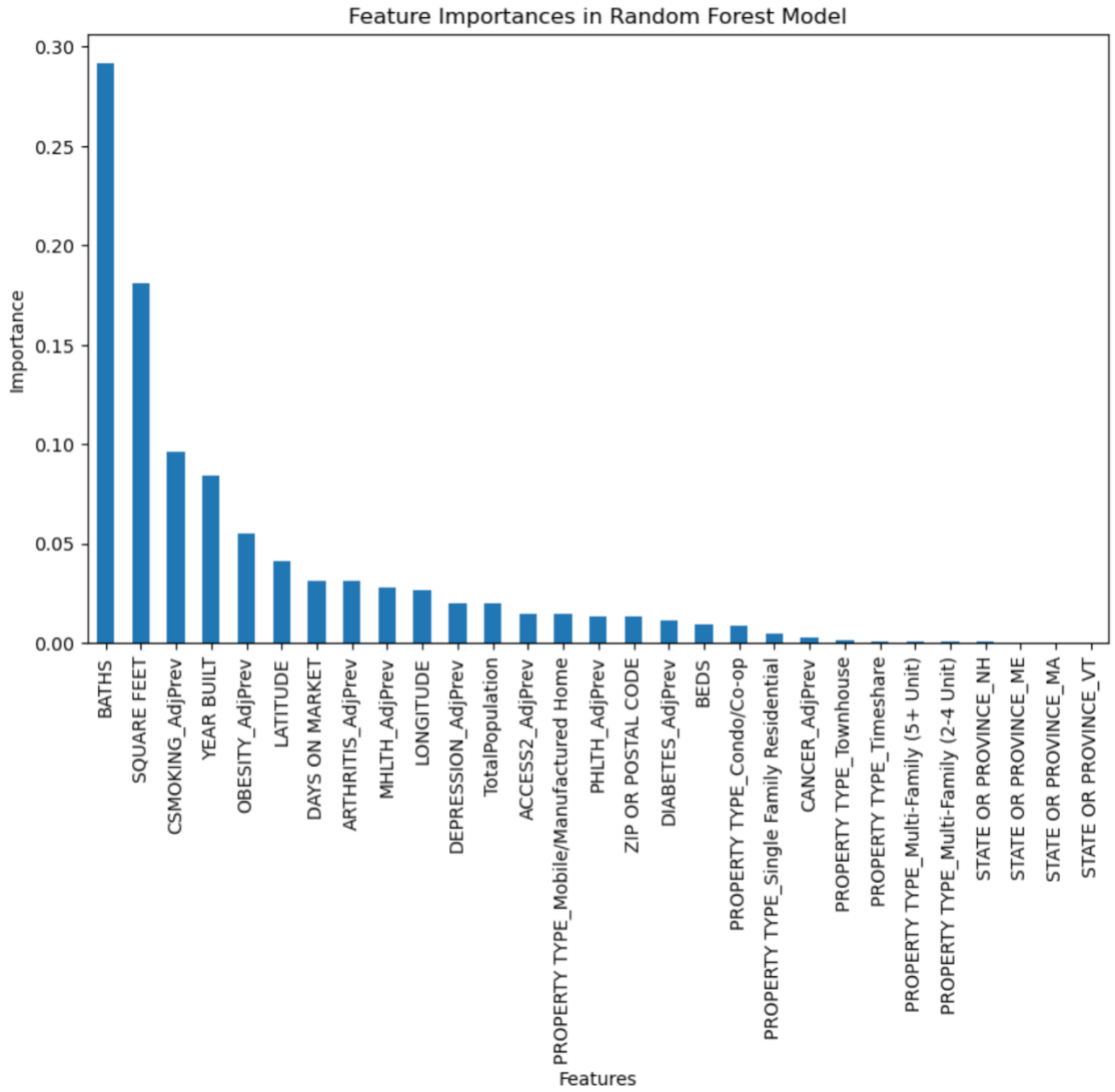


Figure 3: Feature Importance of Random Forest Regression Model

What if we include the feature '$/SQUARE FEET'?

Upon reviewing the data and conducting preliminary analyses, we contemplate incorporating the "$/square feet" feature as a predictor in our house pricing prediction model. The dataset suggests a direct match between housing prices and the product of "$/square feet" and the corresponding "square feet" values, indicating minimal variation. The "$/square feet" feature holds relevance in the housing market and has a clear interpretability within the context of house pricing. However, when considering the inclusion of both 'square feet' and '$/square feet' in our prediction models, caution is warranted to mitigate potential data leakage during the training process.

Results of classification models on BINARY PRICE response

	precision	recall	f1-score	support
0	0.95	0.97	0.96	237
1	0.97	0.95	0.96	223
accuracy			0.96	460
macro avg	0.96	0.96	0.96	460
weighted avg	0.96	0.96	0.96	460

y_test	0	1
y_pred_dt		
0	231	12
1	6	211

Decision Tree

	precision	recall	f1-score	support
0	0.91	0.91	0.91	237
1	0.91	0.90	0.90	223
accuracy			0.91	460
macro avg	0.91	0.91	0.91	460
weighted avg	0.91	0.91	0.91	460

y_test	0	1
y_pred_rf		
0	216	22
1	21	201

Random Forest

	precision	recall	f1-score	support
0	0.94	0.95	0.95	237
1	0.95	0.93	0.94	223
accuracy			0.94	460
macro avg	0.94	0.94	0.94	460
weighted avg	0.94	0.94	0.94	460

y_test / y_pred_gb	0	1
0	226	15
1	11	208

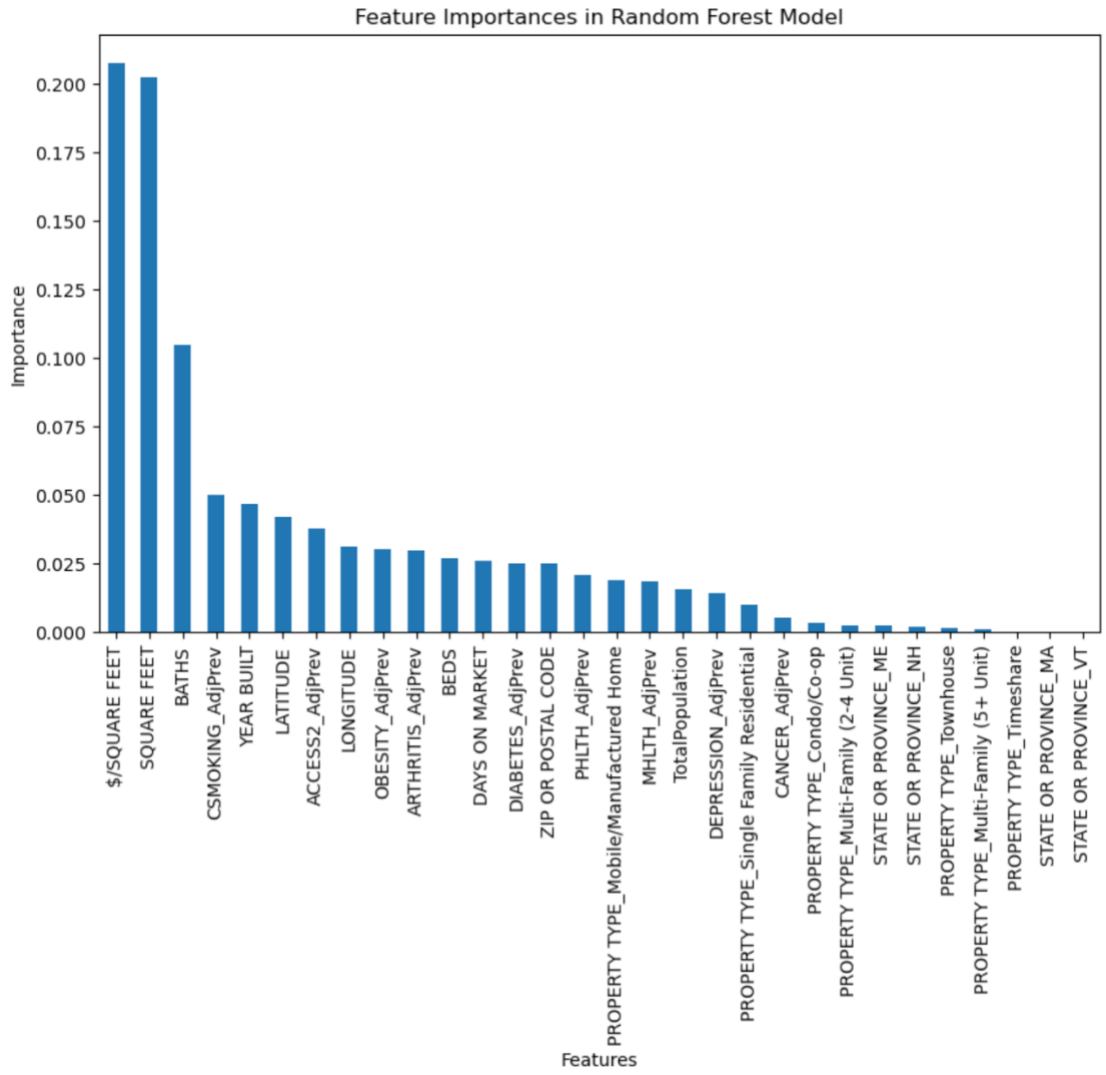
Gradient Boosting

	precision	recall	f1-score	support
0	0.95	0.94	0.94	237
1	0.94	0.95	0.94	223
accuracy			0.94	460
macro avg	0.94	0.94	0.94	460
weighted avg	0.94	0.94	0.94	460

y_test / y_pred_xgb	0	1
0	223	12
1	14	211

XGBoost

All tree-based classification models show high prediction results (> 90% accuracy). The decision tree classification model has the highest performance in classifying the binary price response, with all metrics of more than 95%. The top 1 of feature importance is replaced by the '$/ Square feet' feature, followed by 'Square Feet' and 'Baths' variables.



Results of regression models on CONTINUOUS PRICE response

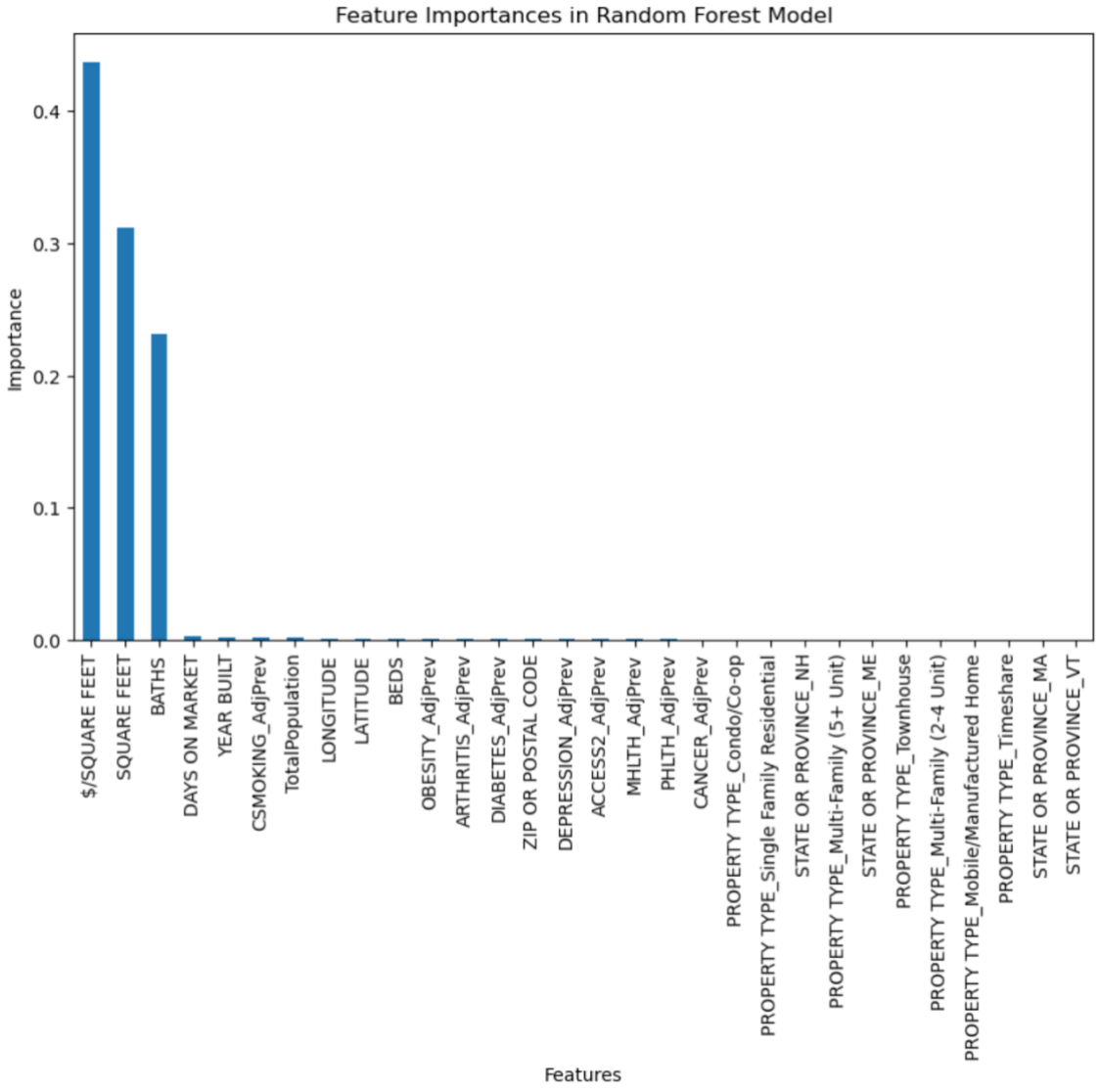
All tree-based regression models have very high results of R-squared (from 93% to 98%) when adding '$/ Square feet' as a predictor. RMSEs are also much lower than previous models. We should be careful with this predictor because it contributes more than 0.4 weight of the feature importance.

```
Decision Tree R-squared: 0.9288079160885627
Decision Tree Root Mean Squared Error: 80944.20785408333

Random Forest R-squared (R2): 0.9683236004964868
Random Forest Root Mean Squared Error (RMSE): 53993.04619137826

Gradient Boosting R-squared (R2): 0.96966096429363
Gradient Boosting Root Mean Squared Error (RMSE): 52840.97360756756

XGBoost R-squared (R2): 0.9781485697874409
XGBoost Root Mean Squared Error (RMSE): 44844.56939672149
```



The dataset's temporal aspect is crucial, as housing market trends evolve over time, and the absence of a time series component might limit our ability to capture dynamic patterns. To enhance model performance, it would be beneficial to incorporate more granular and up-to-date socio-economic indicators, consider time-related trends, and address any outliers or inconsistencies through preprocessing techniques. Additionally, expanding the dataset to include features such as neighborhood characteristics, proximity to amenities and centers, and local economic conditions could provide valuable insights for a more comprehensive predictive analysis.