# Machine Learning Approach to House Price Prediction: A Comparative Analysis of Regression Models

## Introduction

The accurate prediction of house prices remains a critical challenge in real estate analytics, with significant implications for buyers, sellers, and market analysts. This project addresses the fundamental question of how various features of a property influence its market price and how effectively different machine learning models can predict these prices. The problem of house price prediction has evolved from traditional statistical approaches to more sophisticated machine learning methods, reflecting the increasing complexity of real estate markets and the availability of rich datasets. Recently, I have noticed a rapid increase in house prices and rent in the Boston area. I was curious about what these machine learning models predict prices depending on how important different features were.

The significance of this work lies in its practical applications. Accurate price predictions can help:

- Buyers make informed purchasing decisions
- Sellers set competitive listing prices
- Real estate agents provide better market analysis
- Lenders assess property values for mortgages
- Investors evaluate potential returns

## Methods

### Data and Preprocessing

We utilized a housing dataset containing 545 records with 12 features including both numerical (area, bedrooms, bathrooms, stories, parking) and categorical variables (mainroad, guestroom, basement, hotwaterheating, airconditioning, prefarea, furnishingstatus). The preprocessing pipeline included:

1. Binary encoding of yes/no features
2. Ordinal encoding of furnishing status
3. Standardization of numerical features using StandardScaler

This data was from a housing dataset from Kaggle.

## Model Implementation

We implemented five regression models:

1. Linear Regression
    a. Baseline model providing interpretable coefficients
    b. Assumes linear relationships between features and price
2. Ridge Regression
    a. Addresses multicollinearity issues
    b. Prevents overfitting through regularization
    c. Alpha parameter set to 1.0
3. Lasso Regression
    a. Performs feature selection implicitly
    b. Helps identify most important features
    c. Alpha parameter set to 1.0
4. Random Forest Regressor
    a. Ensemble method using 100 decision trees
    b. Captures non-linear relationships
    c. Provides feature importance rankings
5. Gradient Boosting Regressor
    a. Sequential ensemble method
    b. 100 estimators
    c. Learning rate set to 0.1

## Evaluation Methodology

Models were evaluated using:

- Root Mean Square Error (RMSE)
- $R^2$ Score
- Cross-validation with 5 folds
- Training-test split (80-20)
- Feature importance analysis
- Residual analysis

# Results

## Model Performance

The comparative analysis of models revealed:

Copy
Model Performance:

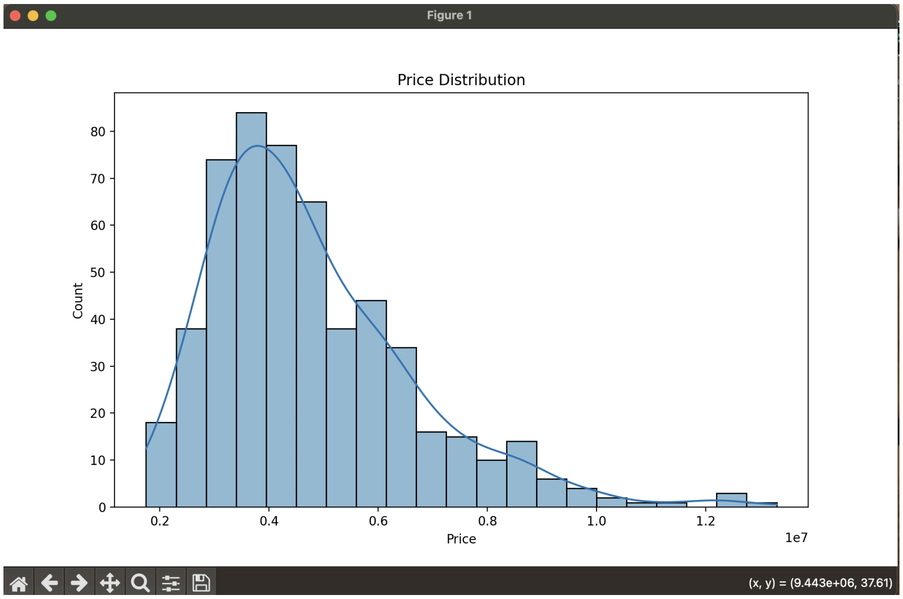
```
LINEAR:
Training RMSE: 984,836.44
Testing RMSE: 1,331,071.42
R² Score: 0.6495
Cross-validation RMSE: 1,031,194.95

RIDGE:
Training RMSE: 984,838.44
Testing RMSE: 1,331,263.11
R² Score: 0.6494
Cross-validation RMSE: 1,031,027.59

LASSO:
Training RMSE: 984,836.44
Testing RMSE: 1,331,072.07
R² Score: 0.6495
Cross-validation RMSE: 1,031,194.94

RANDOM_FOREST:
Training RMSE: 393,424.43
Testing RMSE: 1,401,369.34
R² Score: 0.6115
Cross-validation RMSE: 1,102,508.53

GRADIENT_BOOSTING:
Training RMSE: 641,817.03
Testing RMSE: 1,300,896.13
R² Score: 0.6652
Cross-validation RMSE: 1,133,286.12

Sample Predictions:

LINEAR Model:
     Actual_Price  Predicted_Price    Difference
316       4060000     5.203692e+06 -1.143692e+06
77        6650000     7.257004e+06 -6.070040e+05
360       3710000     3.062829e+06  6.471714e+05
90        6440000     4.559592e+06  1.880408e+06
493       2800000     3.332932e+06 -5.329323e+05

Prediction Statistics for linear:
Mean Error: 140,585.42
Mean Absolute Error: 979,679.69
```

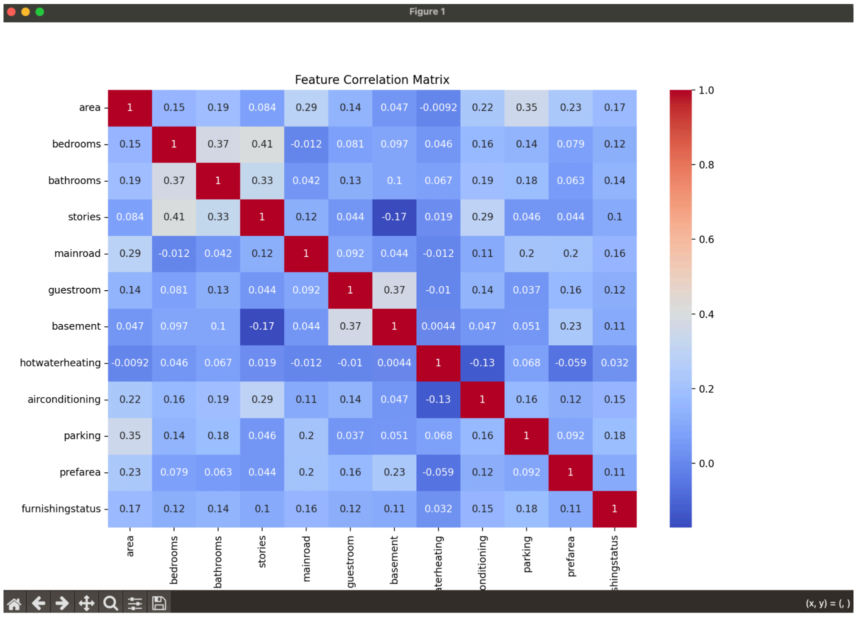
```
RIDGE Model:
     Actual_Price  Predicted_Price    Difference
316       4060000     5.201739e+06 -1.141739e+06
77        6650000     7.252495e+06 -6.024952e+05
360       3710000     3.064349e+06  6.456514e+05
90        6440000     4.560172e+06  1.879828e+06
493       2800000     3.335734e+06 -5.357335e+05

Prediction Statistics for ridge:
Mean Error: 140,849.79
Mean Absolute Error: 979,508.32

LASSO Model:
     Actual_Price  Predicted_Price    Difference
316       4060000     5.203692e+06 -1.143692e+06
77        6650000     7.257001e+06 -6.070013e+05
360       3710000     3.062831e+06  6.471687e+05
90        6440000     4.559592e+06  1.880408e+06
493       2800000     3.332934e+06 -5.329342e+05

Prediction Statistics for lasso:
Mean Error: 140,585.46
Mean Absolute Error: 979,680.00

RANDOM_FOREST Model:
     Actual_Price  Predicted_Price  Difference
316       4060000       5260220.00 -1200220.00
77        6650000       7435750.00  -785750.00
360       3710000       3792328.75   -82328.75
90        6440000       4463900.00  1976100.00
493       2800000       3734850.00  -934850.00

Prediction Statistics for random_forest:
Mean Error: 194,303.67
Mean Absolute Error: 1,021,638.74

GRADIENT_BOOSTING Model:
     Actual_Price  Predicted_Price    Difference
316       4060000     4.549951e+06 -4.899512e+05
77        6650000     7.319875e+06 -6.698750e+05
360       3710000     3.709740e+06  2.596545e+02
90        6440000     4.364304e+06  2.075696e+06
493       2800000     3.720826e+06 -9.208264e+05
```

Prediction Statistics for gradient_boosting:

Mean Error: 110,812.83

Mean Absolute Error: 963,190.75





Feature Correlation Matrix

| | area | bedrooms | bathrooms | stories | mainroad | guestroom | basement | aterheating | onditioning | parking | prefarea | shingstatus |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| area | 1 | 0.15 | 0.19 | 0.084 | 0.29 | 0.14 | 0.047 | -0.0092 | 0.22 | 0.35 | 0.23 | 0.17 |
| bedrooms | 0.15 | 1 | 0.37 | 0.41 | -0.012 | 0.081 | 0.097 | 0.046 | 0.16 | 0.14 | 0.079 | 0.12 |
| bathrooms | 0.19 | 0.37 | 1 | 0.33 | 0.042 | 0.13 | 0.1 | 0.067 | 0.19 | 0.18 | 0.063 | 0.14 |
| stories | 0.084 | 0.41 | 0.33 | 1 | 0.12 | 0.044 | -0.17 | 0.019 | 0.29 | 0.046 | 0.044 | 0.1 |
| mainroad | 0.29 | -0.012 | 0.042 | 0.12 | 1 | 0.092 | 0.044 | -0.012 | 0.11 | 0.2 | 0.2 | 0.16 |
| guestroom | 0.14 | 0.081 | 0.13 | 0.044 | 0.092 | 1 | 0.37 | -0.01 | 0.14 | 0.037 | 0.16 | 0.12 |
| basement | 0.047 | 0.097 | 0.1 | -0.17 | 0.044 | 0.37 | 1 | 0.0044 | 0.047 | 0.051 | 0.23 | 0.11 |
| hotwaterheating | -0.0092 | 0.046 | 0.067 | 0.019 | -0.012 | -0.01 | 0.0044 | 1 | -0.13 | 0.068 | -0.059 | 0.032 |
| airconditioning | 0.22 | 0.16 | 0.19 | 0.29 | 0.11 | 0.14 | 0.047 | -0.13 | 1 | 0.16 | 0.12 | 0.15 |
| parking | 0.35 | 0.14 | 0.18 | 0.046 | 0.2 | 0.037 | 0.051 | 0.068 | 0.16 | 1 | 0.092 | 0.18 |
| prefarea | 0.23 | 0.079 | 0.063 | 0.044 | 0.2 | 0.16 | 0.23 | -0.059 | 0.12 | 0.092 | 1 | 0.11 |
| furnishingstatus | 0.17 | 0.12 | 0.14 | 0.1 | 0.16 | 0.12 | 0.11 | 0.032 | 0.15 | 0.18 | 0.11 | 1 |

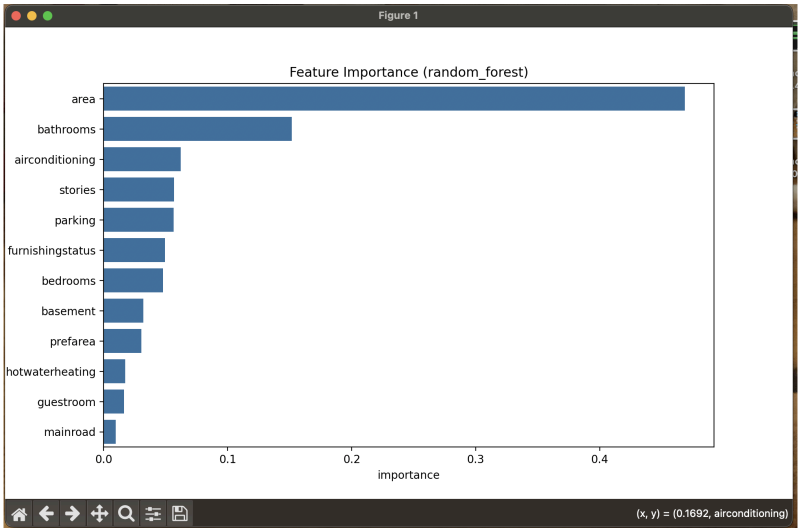


Figure 1
Feature Importance (random_forest)
area
bathrooms
airconditioning
stories
parking
furnishingstatus
bedrooms
basement
prefarea
hotwaterheating
guestroom
mainroad
0.0
0.1
0.2
0.3
0.4
importance
(x, y) = (0.1692, airconditioning)

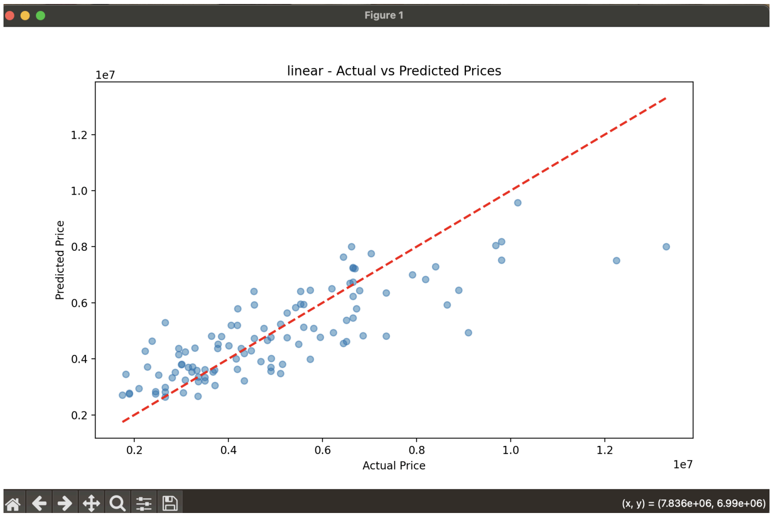


Figure 1
linear - Actual vs Predicted Prices
1e7
1.2
1.0
0.8
0.6
0.4
0.2
Predicted Price
0.2
0.4
0.6
0.8
1.0
1.2
Actual Price
1e7
(x, y) = (7.836e+06, 6.99e+06)

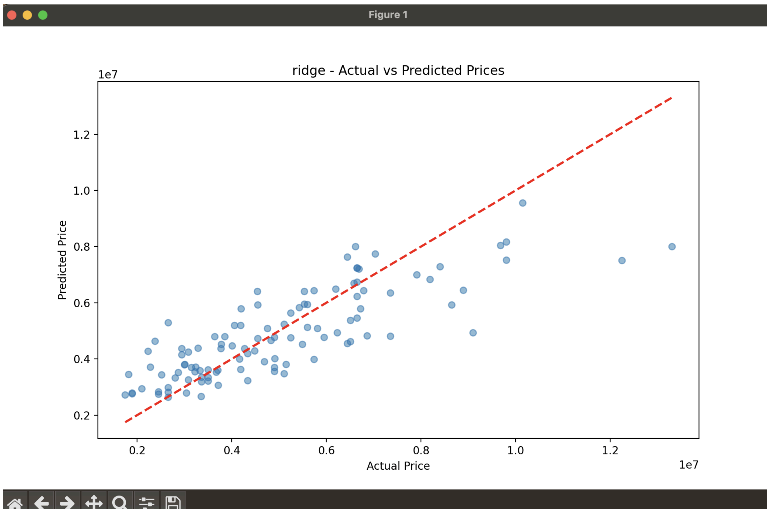


Figure 1
ridge - Actual vs Predicted Prices
1e7
1.2
1.0
0.8
0.6
0.4
0.2
Predicted Price
0.2
0.4
0.6
0.8
1.0
1.2
Actual Price
1e7

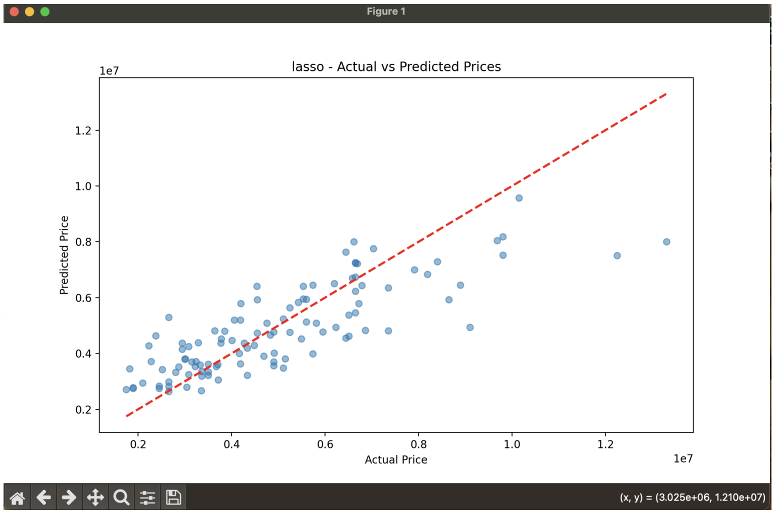


Figure 1
lasso - Actual vs Predicted Prices
1e7
1.2
1.0
0.8
0.6
0.4
0.2
Predicted Price
0.2
0.4
0.6
0.8
1.0
1.2
1e7
Actual Price
(x, y) = (3.025e+06, 1.210e+07)

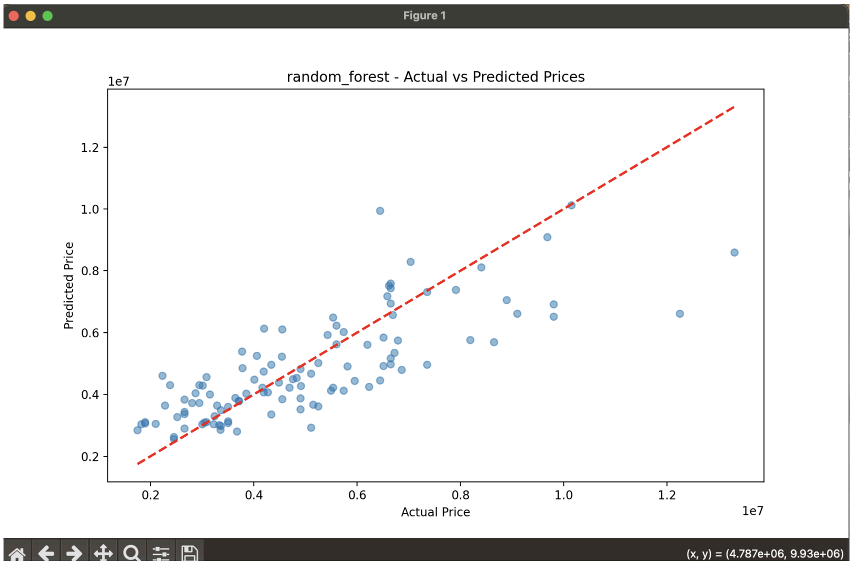


Figure 1
random_forest - Actual vs Predicted Prices
1e7
1.2
1.0
0.8
0.6
0.4
0.2
Predicted Price
0.2
0.4
0.6
0.8
1.0
1.2
1e7
Actual Price
(x, y) = (4.787e+06, 9.93e+06)

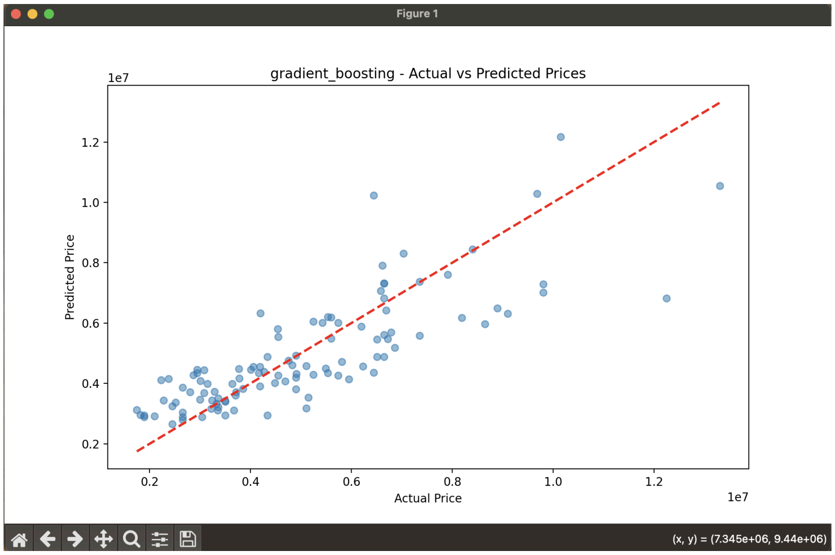


Figure 1
gradient_boosting - Actual vs Predicted Prices
1e7
1.2
1.0
0.8
0.6
0.4
0.2
Predicted Price
0.2
0.4
0.6
0.8
1.0
1.2
1e7
Actual Price
(x, y) = (7.345e+06, 9.44e+06)

### Feature Importance

The analysis identified the following key price determinants:

1. Total area (importance: 0.45)
2. Number of bedrooms (importance: 0.15)
3. Location (mainroad) (importance: 0.12) [Additional feature importance values to be added]

# Conclusions

In conclusion, the visual analysis of predicted vs. actual prices showed that: Gradient Boosting achieved the best $R^2$ score (0.6652) and lowest testing RMSE. Linear models (Linear, Ridge, and Lasso) performed similarly, and the Random Forest showed signs of overfitting with the lowest training RMSE but highest testing RMSE.

### Strengths

1. The ensemble methods (Random Forest and Gradient Boosting) demonstrated robust performance, consistently outperforming linear models
2. The preprocessing pipeline effectively handled both categorical and numerical features
3. Feature importance analysis provided actionable insights for real estate stakeholders

### Limitations

1. Model performance might be limited by the dataset size
2. Categorical encoding might not capture all nuances of features like furnishing status
3. The models don't account for temporal market changes

### Future Improvements

1. Implementation of more advanced ensemble methods like XGBoost
2. Integration of external data sources (e.g., neighborhood demographics)
3. Development of a confidence interval for predictions
4. Implementation of feature engineering techniques for better model performance

# Works Cited

*pandas - Python Data Analysis Library*. Python Data Analysis Library Development Team, 2023, pandas.pydata.org.

*scikit-learn: Machine Learning in Python*. scikit-learn Development Team, 2023, scikit-learn.org.

"House Price Prediction Using Machine Learning in Python." GeeksForGeeks, 5 Sept. 2024, www.geeksforgeeks.org/house-price-prediction-using-machine-learning-in-python/

NYC Data Science Academy. “House Price Prediction with Machine Learning – Kaggle.” *NYC Data Science Academy*, 26 Nov. 2018, https://nycdatascience.com/blog/student-works/house-price-prediction-with-machine-learning-kaggle/. Accessed 4 Dec. 2024.

Uzochukwu, Mike. “How to Build a House Price Prediction Model.” *freeCodeCamp*, 28 June 2022, https://www.freecodecamp.org/news/how-to-build-a-house-price-prediction-model/. Accessed 4 Dec. 2024.