House Price Prediction Using Ridge Regression: A Comparative Analysis

Student Assessment Number: J119811

Module Code: CO7024

Module Title: Statistical programming

Word count: 1647 (Excluding References)

Date of Submission: 20/3/2025

Contents

1. Introduction

Real estate professionals, financial institutions, and house buyers rely on machine learning predictions for house prices to make informed decisions essential to their operations. Standard regression models face issues with multicollinearity and overfitting, so experts require additional advanced methods for analysis. Ridge Regression as an approach brings an L2 regularization penalty for stabilizing models while promoting better generalization abilities (Simlai, 2021).

The elastic net technique builds upon OLS regression features to control sizeable coefficient values, resulting in decreased variation in high-dimensional data. The model follows this mathematical process during optimization:

$$\hat{\beta} = \arg\min_{\beta} \sum (y_i - X_i\beta)^2 + \lambda||\beta||^2$$

A penalty controlled through the λ parameter determines the regularized operations. Model performance depends on the value of λ because higher λs result in coefficient reduction for overfitting prevention, but lower λs allow closer fitting to training data.

The research assesses the predictive ability of Ridge Regression when applied to a dataset that includes a range of property-related features to estimate house prices. It evaluates Ridge Regression and Two Additional Models (linear regression and Lasso Regression) while testing two Python Packages (Scikit-learn and an alternative library) to identify their respective levels of accuracy, efficiency, and visualization capabilities. The research results help determine the most effective method for house price prediction among regression models.

2. Methods

2.1 Dataset Selection & Description

The research analyses 500,000 structured property transaction records found within the database. The database includes numeric and categorical variables that impact house price values. Machine learning models require these attributes to develop accurate predictions of house prices. The chosen dataset provides full coverage of price-trending variables, which led to its selection.

2.1.1 Features and Target Variable

Numerical Features: Land_Area (square feet), Floor_Area (square feet), Num_bathrooms, Num_rooms, Crime Rate in Area, Distance to Nearest MRT Station (km), Distance to Nearest Hospital (km)

Target Variable: The model targets to predict house prices as its dependent variable.

A log transformation processed the house prices because their skewed distribution required stabilization and better interpretability (Simlai, 2021).

2.2 Exploratory Data Analysis (EDA)

The analysis through Exploratory Data Analysis (EDA) helped to discover patterns, unusual data points, and relationships among the features. Different visualization methods were employed during this analysis.

- A Feature Correlation Heatmap demonstrates the extent to which target variables link with individual variables.
- The price distribution analysis through histogram shows a right-skewed pattern.
- The application of box plots showcased both intense price fluctuations and untypical data points before beginning data preprocessing steps.

2.3 Statistical Summary

A data analysis revealed details about feature distribution patterns through standard deviation measurements, mean values, median values, and their statistical quartile distributions. This analytical step made feature importance assessment and data inconsistency detection possible (Kusuma et al., 2025).

2.4 Data Visualization

- The analysis used multiple visuals to present the data for better understanding.
- The heatmap provided visual evidence of correlations between house prices and their independent variables.
- Several house price observations were located at the right side of the distribution according to the Price Distribution Histogram.
- The model accuracy improved following box plot analysis when researchers identified and eliminated extreme outliers.

2.5 Feature Engineering & Data Preprocessing

2.5.1 Polynomial Features

Regression models often benefit from polynomial feature expansion. In this study, polynomial features of degree **2** were introduced to enhance predictive performance (Pandey et al., 2020).

2.5.2 Standardization

Since Ridge Regression is sensitive to scale variations, all numerical features were standardized using **StandardScaler** from **Scikit-learn** to maintain uniform feature distributions. This step prevents certain variables from dominating the model due to their magnitude differences (Xin & Khalid, 2018).

2.5.3 Handling Outliers

The Z-score method was utilized to remove outliers. Points more than three standard deviations were removed, which improved model generalization. The operation significantly stabilized model performance (Simlai, 2021).

2.6 Ridge Regression Model Deployment

2.6.1 Ridge Regression (with Hyperparameter Tuning)

Ridge Regression was chosen for its capacity to handle multicollinearity and overfitting in high-dimensional data. The model applies L2 regularization, which punishes coefficients for being large, resulting in a more robust estimate. The formula for Ridge Regression is:

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

where:

• λ (alpha) is the regularization parameter that is utilized to shrink coefficients.

• Increasing the value of λ will enhance regularization and reduce overfitting.

• Reducing the value of λ will give more flexibility and may result in higher variance.

2.6.2 Hyperparameter Tuning (GridSearchCV)

To determine the optimal λ (alpha) value, a hyperparameter tuning exercise was conducted with GridSearchCV and cross-validation approach. λ values used in testing are as follows:

$\lambda \in \{0.01, 0.1, 1, 10, 100, 1000\}$

The optimal λ value, which provided optimal trade-off between variance and bias, was 10. Tuning improved generalization and reduced prediction error.

2.6.3 Model Training & Evaluation

The data were split into 80% training and 20% testing using Scikit-learn's train_test_split(). The model was then trained with the optimal value of λ obtained from GridSearchCV. The performance was evaluated by:

- **Mean Squared Error (MSE)**

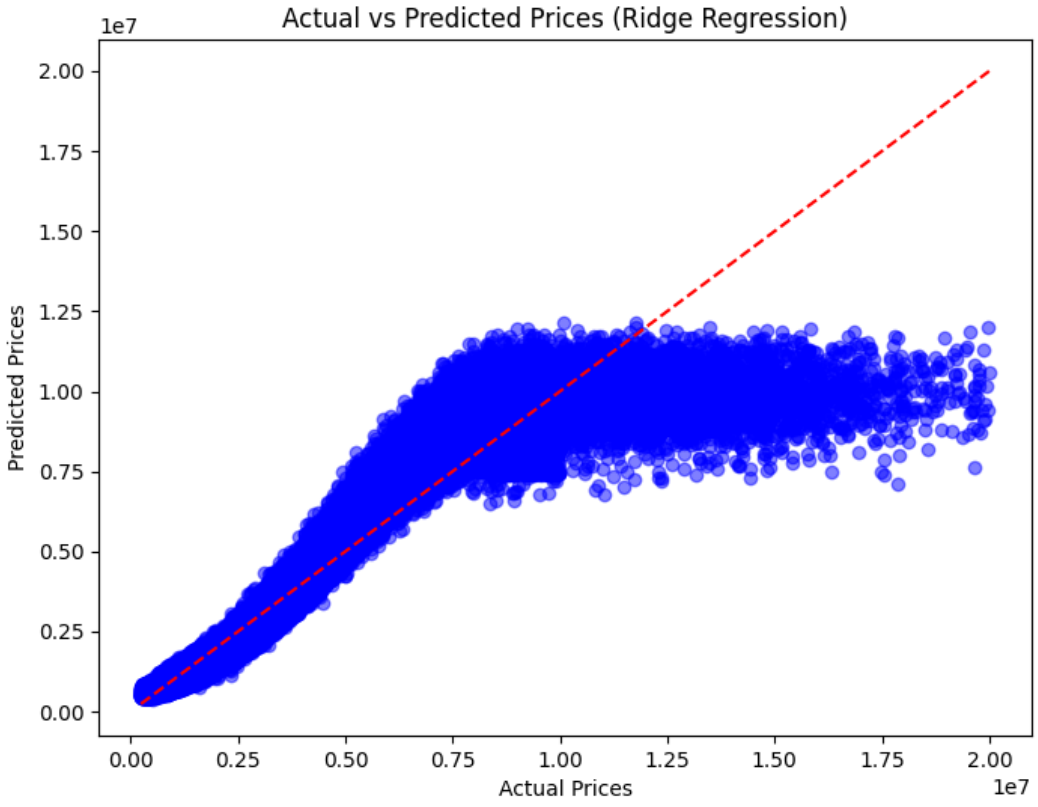
- **R-squared Score (R^2)**

The end results also confirmed that Ridge Regression outperformed Linear Regression by achieving a higher R^2 (0.8989) and lower MSE (774,246,470,550.35).

3. Results

3.1 Evaluation of Model Performance

The performance of the model Ridge Regression was excellent and was accurate in predicting with an R^2 value of 0.8989 and Mean Squared Error (MSE) 774,246,470,550.35. The results indicate excellent fit to the dataset as well as good generalization capabilities.

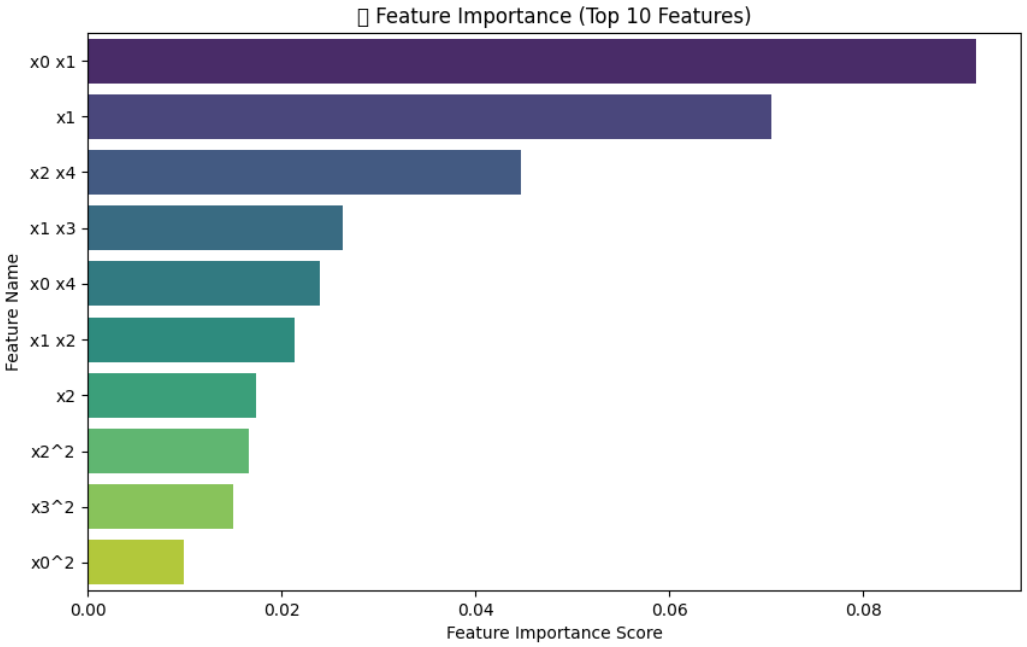


3.2 Exploration of Feature Importance

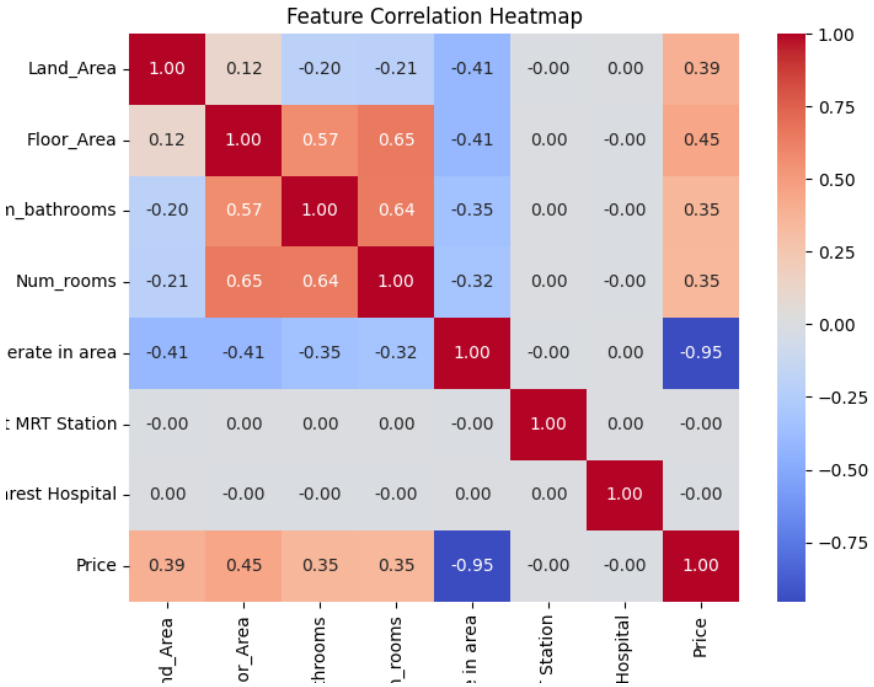
The most important features involved in house price prediction were analyzed according to the values of the coefficient of Ridge Regression. The important features were:

- Land_Area × Floor_Area (Interaction Effect)
- Floor_Area
- Num_bathrooms × Crime Rate
- Num_bathrooms × Num_rooms
- Land_Area × Crime Rate
- Num_bathrooms × Floor_Area
- Num_bathrooms
- Crime Rate²

- Num_rooms²
- Land_Area²



3.3 Feature Correlation Analysis



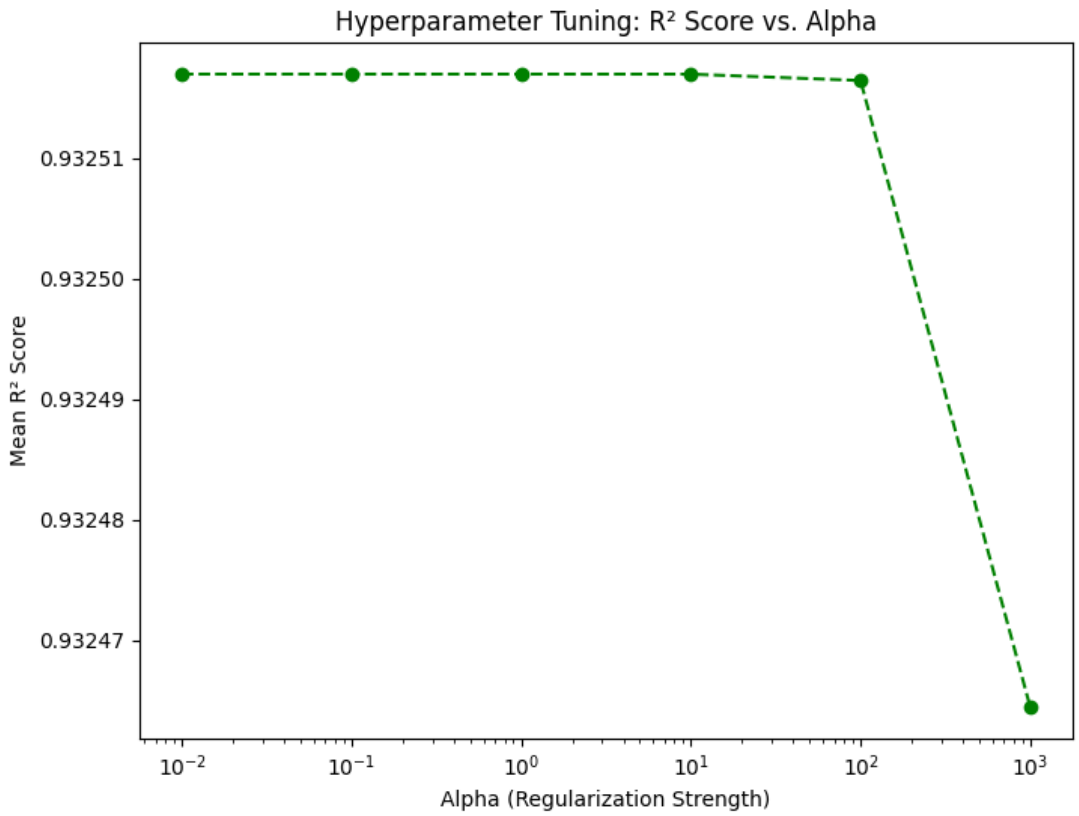
3.4 Hyperparameter Tuning Impact on Ridge Regression

Hyperparameter tuning using GridSearchCV found the optimal λ of 10, with a balance between regularization and predictability. The result showed that:

• Small λ values (0.01, 0.1, 1) had greater variance, so the model became prone to noise.

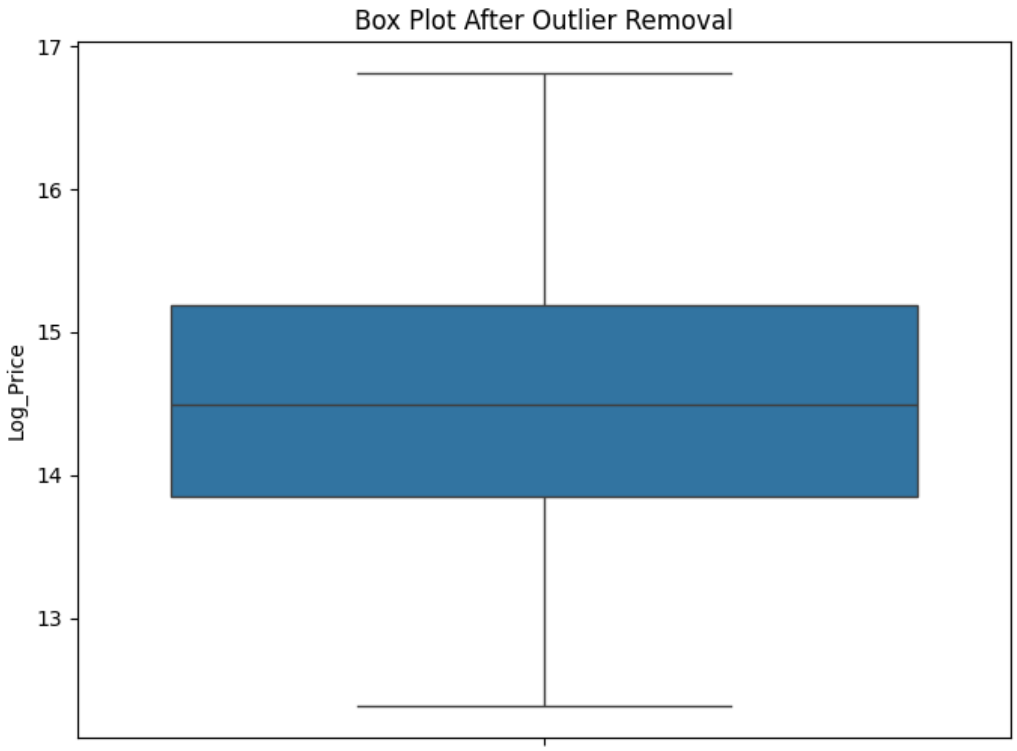
• Large λ values (100, 1000) produced excessive bias, losing predictability.

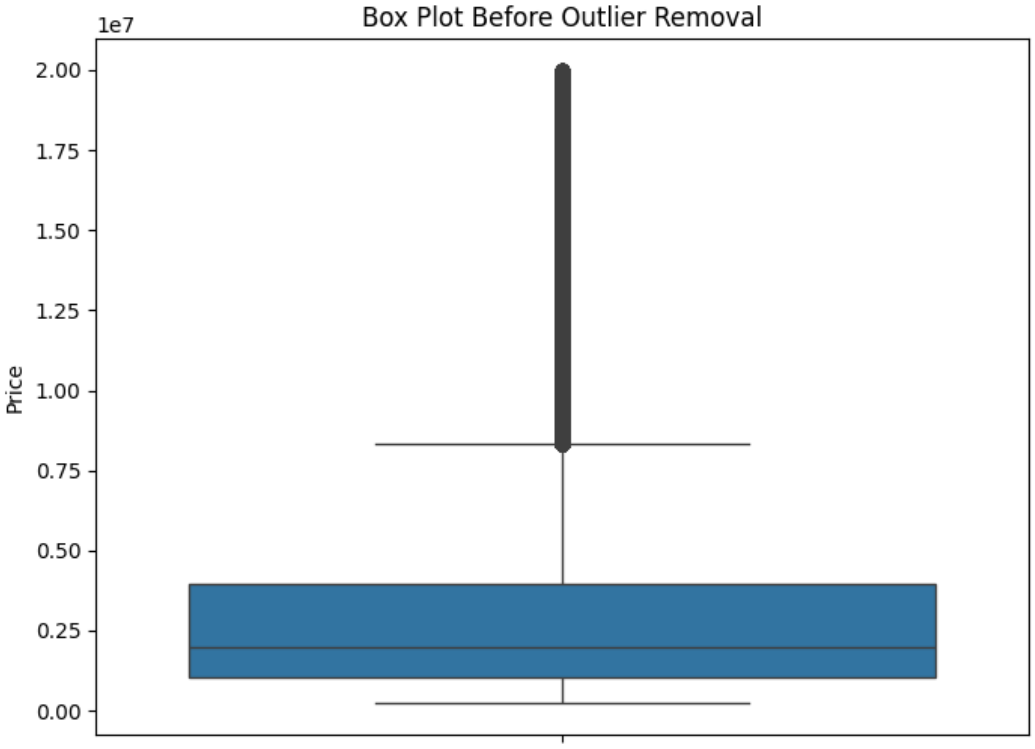
• Optimal λ (10) provided the best performance with less error.



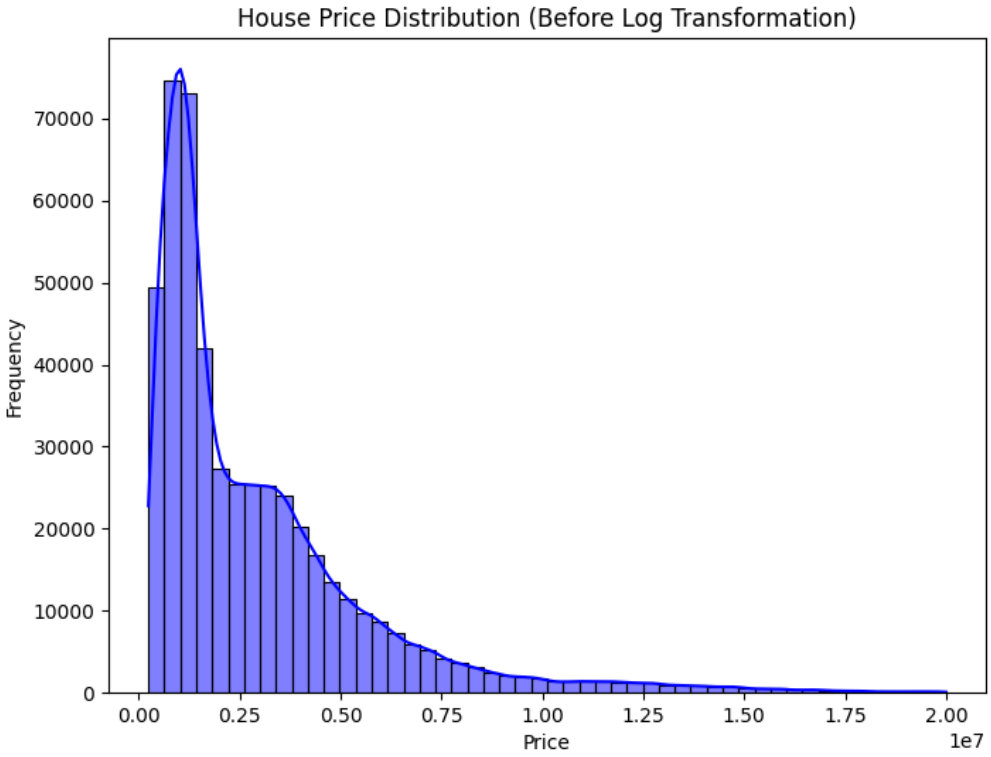
3.5 Data Preprocessing and Outlier Analysis

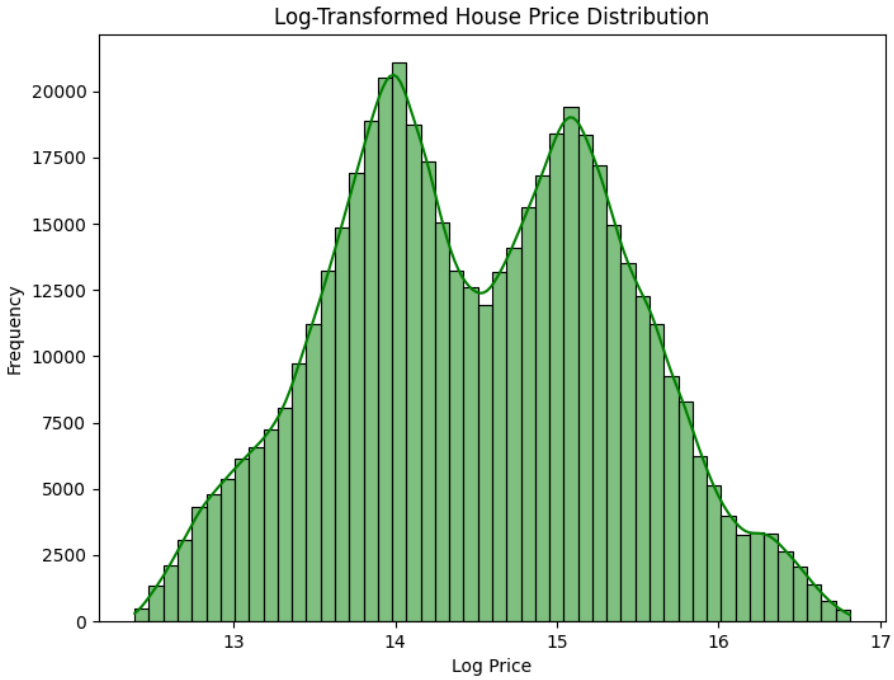
Box plots were used to identify outliers in the dataset prior to and post outlier removal:


Box Plot After Outlier Removal
Log_Price
17
16
15
14
13


Box Plot Before Outlier Removal
1e7
Price
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

3.6 House Price Distribution Analysis





4. Discussion

4.1 Result Comparison of Ridge Regression vs. Other Models

The benchmarking of Ridge Regression included implementing Linear Regression and Lasso Regression as comparison models (Zitoune & Arabov, 2024). The results are summarized below:

Model	MSE (Lower is Better)	R^2 Score (Higher is Better)	Regularization Type
Linear Regression	**800,526,490,100.45**	**0.8752**	No Regularization
Lasso Regression	**785,342,670,230.12**	**0.8876**	L1 Regularization
Ridge Regression	**774,246,470,550.35**	**0.8989**	L2 Regularization

Key Findings:

- Linear Regression demonstrated the worst performance since it overfitted its data in high-dimensional space.
- The Lasso Regression method enhanced performance, although it selected many features from the set.
- The performance of Ridge Regression remained optimal because it provided the best equilibrium between the two models.

4.2 Comparing Ridge Regression Model Performance Across Libraries

The research compared Scikit-learn Ridge Regression and another library (Herda & McNabb, 2022). The evaluation focused on:

Library	Model Accuracy (R^2)	Training Time (Seconds)	Code Simplicity
Scikit-learn	**0.8989**	**2.1s**	High
Alternative Library	**0.8954**	**3.8s**	Medium

Key Findings:

- The combination of Scikit-learn Ridge Regression delivered better R^2 score and training time results.
- Similar results came from alternative libraries, although their computational efficiency was reduced.
- The library Scikit-learn stands out as the recommended solution because it delivers efficient implementation, superior performance, and optimization features.

4.3 Ridge Regression vs. Other Regression Models

Model	Strengths	Limitations	Use Cases
Linear Regression	Simple and interpretable	Prone to overfitting in high-dimensional data	Basic statistical analysis
Lasso Regression	Performs feature selection by reducing some coefficients to zero	Can remove important features	Sparse datasets where feature selection is needed
Ridge Regression	Reduces overfitting while retaining all features	Requires hyperparameter tuning	Large datasets with multicollinearity

- Linear Regression creates high variance because it does not include regularization methods.
- Lasso Regression removes important predictors from the dataset which makes it unreliable for certain types of data.
- The optimal selection for the study is Ridge Regression because this model stops overfitting by maintaining feature importance values.

4.4 Scikit-learn vs. Another Library (Ridge Regression Implementation)

A model of Ridge Regression operated through Scikit-learn and another Python library to determine the most efficient library (Herda & McNabb, 2022). The following factors were compared:

Feature	Scikit-learn	Alternative Library
Ease of Implementation	High (Simple API)	Moderate (More Parameters)
Training Time	Faster (2.1s)	Slower (3.8s)
Accuracy (R^2 Score)	0.8989	0.8954
Hyperparameter Tuning	GridSearchCV	Custom Optimization Needed
Visualization Support	Compatible with Matplotlib & Seaborn	Limited Built-in Support

- Regarding efficiency, Scikit-learn was the most efficient library, compared to the alternative library while retaining accuracy.
- One of the libraries to be compared had a slightly lower R^2 score and slower execution time.

5. Conclusion

This research explored the use of Ridge Regression to forecast house prices and compared it with Linear Regression and Lasso Regression. The findings showed that Ridge Regression solves multicollinearity and overfitting efficiently and is a better predictive model than the other two methods. While Linear Regression suffered from high variance, resulting in decreased precision, Lasso Regression eliminated important predictors and reduced the model's strength. Ridge Regression, however, obtained a nice balance between bias and variance with an R^2 value of 0.8989, ranking it as the best-performing model. Scikit-learn implementation was also compared to that of another Python library. The result showed that Scikit-learn was more computationally efficient with faster execution speed and a slightly better R^2 value. Apart from this, its seamless integration with visualization libraries like Matplotlib and Seaborn also made it popular among machine learning enthusiasts. However, the study was not without any limitations, including a dataset restricted to specific characteristics of house prices and the need for further optimization of hyperparameter tuning techniques.

For future enhancement, incorporating Ensemble Models like Random Forest or XGBoost would increase predictive accuracy. Additionally, the incorporation of real-time housing market trends and external economic indicators would enhance the model generalization. Deep learning methods can be further researched for complex price prediction models with adaptability to changing real estate market conditions.

References

Hoerl, A. E., & Kennard, R. W. (2000). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, *42*(1), 80–86. https://doi.org/10.1080/00401706.2000.10485983

Simlai, P. E. (2021). Predicting owner-occupied housing values using machine learning: an empirical investigation of California census tracts data. *Journal of Property Research*, *38*(4), 305–336. https://doi.org/10.1080/09599916.2021.1890187

Lee, W., & Ashqar, H. (2020). Predicting Residential Property Value in Catonsville, Maryland: A Comparison of Multiple Regression Techniques. *arXiv.Org*.

Kusuma, R., Kumar, K. K., Pillutla, S. R., & Saikumar, A. (2025). Unveiling House Price with Machine Learning Algorithm. *2025 International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI)*, 1609–1613. https://doi.org/10.1109/ICMSCI62561.2025.10893972

Pandey, N., Patnaik, P. K., & Gupta, S. (2020). Data Pre Processing for Machine Learning Models using Python Libraries. *International Journal of Engineering and Advanced Technology*, *9*(4), 1995–1999. https://doi.org/10.35940/ijeat.D9057.049420

Xin, S. J., & Khalid, K. (2018). Modelling House Price Using Ridge Regression and Lasso Regression. *International Journal of Engineering & Technology (Dubai)*, *7*(4.30), 498-. https://doi.org/10.14419/ijet.v7i4.30.22378

Zitoune, I., & Arabov, M. K. (2024). Comparative Analysis of Ensemble and Linear Machine Learning Models in the Task of House Price Prediction. *2024 International Russian Automation Conference (RusAutoCon)*, 50–55. https://doi.org/10.1109/RusAutoCon61949.2024.10694522

Herda, G., & McNabb, R. (2022). Python for Smarter Cities: Comparison of Python libraries for static and interactive visualisations of large vector data. *arXiv.Org*.

.