CAN THO UNIVERSITY

School of Education

Report Computational Mathematics

Optimizing House Price Predictions with Lasso Regression

Supervisor:
PhD. Tran Thu Le

Student:

1.	Patcharapon Jitprapai	E2400019
2.	Tanchanok Naksuwan	E2400020
3.	Ranchida Saengsri	E2400027
4.	Rusdee Daraneetalea	E2400028

CAN THO UNIVERSITY

School of Education

Report Computational Mathematics

Optimizing House Price Predictions with Lasso Regression

Supervisor:
PhD. Tran Thu Le

Student:

1.	Patcharapon Jitprapai	E2400019
2.	Tanchanok Naksuwan	E2400020
3.	Ranchida Saengsri	E2400027
4.	Rusdee Daraneetalea	E2400028

Acknowledgments

First and foremost, our deepest gratitude goes to **Associate Professor Dr. Nguyen Trung Kien**, lecturer at Can Tho University, for his expert supervision of this report titled *"Lasso Regression for House Price Prediction."* His profound knowledge, sense of responsibility, and dedication to teaching and research have provided us with invaluable guidance, timely feedback, and the motivation needed to overcome every challenge in our study.

We also wish to extend our sincere thanks to all **lecturers of the Mathematics Department, School of Education, Can Tho University** as well as to the **faculty members of Walailak University (Thailand)**. Their exceptional teaching, academic inspiration, and ongoing support have played a key role in shaping our understanding of statistical modeling and machine learning techniques, particularly in the application of *Lasso regression*, ultimately contributing to the success of our studies and the completion of this report.

We are especially grateful to **Mr. Huynh Nhut Tan**, a student of *Cohort 49, Mathematics Teacher Education, Can Tho University*, and **Mr. Tran Hieu Nhan**, a student of *Cohort 48, Mathematics Teacher Education, Can Tho University*, for their generous and enthusiastic support during our research process. Their assistance in sourcing references, clarifying specialized topics, and sharing practical insights greatly enriched the quality and completeness of this report.

We would also like to extend our heartfelt appreciation to our fellow classmates, whose companionship throughout this journey — through the exchange of ideas, shared efforts, and active collaboration — significantly enhanced our academic experience and research productivity.

Finally, we would like to express our most heartfelt gratitude to our **parents and families**, who have always been a steadfast source of love, support, and encouragement. Their belief in us has been a constant driving force, inspiring our perseverance and determination throughout both our academic journey and the realization of this study.

We sincerely hope that this report will serve as a valuable reference for those with an interest in *Lasso regression* and inspire further research in the field of predictive modeling and data science.

Respectfully,
Patcharapon Jitprapai
Tanchanok Naksuwan
Ranchida Saengsri
Rusdee Daraneetalea

Contents

Introduction

1. Historical Development

House - price forecasting has evolved from early hedonic pricing approaches in economics—which quantified how factors like location, floor area and neighborhood amenities influence value—to modern data-driven techniques powered by big, structured real-estate datasets. In the 1990s and 2000s, researchers experimented with classical linear and nonlinear regression models, but these often struggled with many correlated predictors and overfitting. More recently, machine-learning methods such as random forests and neural networks have achieved impressive accuracy, albeit at the cost of interpretability. This growing tension between predictive power and model transparency has driven interest in regularization methods like *Lasso regression*, which automatically select the most important features while controlling model complexity.

2. Motivation

Traditional least-squares regression breaks down when faced with high-dimensional housing data containing dozens—or even hundreds—of potentially redundant or collinear features. *Lasso Regression* (Least Absolute Shrinkage and Selection Operator) addresses this by adding an ℓ_1 penalty to the loss function. As it shrinks many coefficient estimates exactly to zero, Lasso both regularizes the model (reducing overfit) and performs variable selection in one step. The result is a sparser, more interpretable model that highlights the handful of property attributes most predictive of prices, making it ideal for real-world decision-support in real estate.

3. Objectives

This report sets out to:

- Implement *Lasso Regression* on a real-world housing dataset to forecast sale prices.
- Investigate how varying the regularization parameter (λ) affects feature sparsity and out-of-sample accuracy.
- Evaluate model performance using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2.

- Compare Lasso's predictive strength and interpretability against baseline models (e.g. ordinary least squares, ridge regression, tree-based methods).

4. Report Structure

Preliminary Knowledge

This chapter provides the essential theoretical background needed to understand and apply Lasso Regression in the context of house price prediction. The discussion covers key concepts that form the basis for the methods used in the report.

- Addresses the problem of multicollinearity in linear regression, explaining its impact on model stability and interpretability.
- Introduces regularization techniques, with a focus on both Ridge and Lasso Regression, to mitigate overfitting and improve model robustness.
- Explains the mathematical formulation of Lasso Regression, highlighting the role of the L1 norm in promoting sparsity and feature selection.

This section ensures that readers gain a clear understanding of the foundational concepts required for effective use of Lasso Regression, particularly the benefits it offers in model simplification and generalization.

Problem Formulation and Methodology

In this section, the house price prediction problem is formally defined and the methodological framework for addressing it is outlined in detail. This prepares the groundwork for building and evaluating the predictive model.

- Defines the target problem using a dataset that includes features such as area, age, number of bedrooms, and garage availability.
- Details the data preprocessing steps, including normalization, data splitting, and handling of missing values, to ensure data quality and consistency.
- Describes the implementation of Lasso Regression, including hyperparameter optimization via cross-validation, and the systematic comparison with baseline models such as OLS regression.

By clearly presenting both the problem and the solution process, this section lays out a logical and structured approach that supports the validity and reliability of the study's results.

Experimental Results

This chapter presents the outcomes of the experiments and offers a critical evaluation of the Lasso model's performance. The effectiveness of the approach is demonstrated through both visual and quantitative analysis.

- Provides graphical representations of the features and their relationships, facilitating better understanding of the data.
- Reports quantitative performance metrics such as Mean Squared Error (MSE), R^2 Score, and accuracy for both training and testing sets.
- Analyzes the influence of the regularization parameter (alpha) on model performance and feature selection, with benchmarking against OLS and Ridge regression models.

Through empirical results and comparative analysis, this section highlights the strengths of Lasso Regression in reducing model complexity and enhancing predictive accuracy.

Conclusions and Future Work

The final chapter summarizes the key findings of the report and outlines possible directions for future research and practical applications. It reflects on the study's contributions and points toward further opportunities for development.

- Summarizes the main results, discussing both the effectiveness and the limitations of Lasso Regression, particularly in relation to correlated or irrelevant features.
- Emphasizes the practical applications of house price prediction in fields such as real estate, urban planning, and financial management.
- Recommends further research avenues, including enriching the dataset, exploring advanced regression methods, integrating spatial data, and developing accessible web-based tools

This section consolidates the report's overall contributions, emphasizing the value of Lasso Regression for both academic study and practical application, and provides guidance for future enhancements.

Chapter 1

Preliminary Knowledge

1.1 Linear Regression Overview

Linear regression models the relationship between a target variable y and a set of predictors $x_1, x_2, \ldots, x_n$ using a linear equation:

$$\hat{y} = \beta_0 + \sum_{j=1}^{n} \beta_j x_j$$

The goal is to estimate the coefficients β_j that minimize the Residual Sum of Squares (RSS):

$$\min_{\beta} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

While this method works well in many cases, it struggles when features are highly correlated (multicollinearity) or when the number of predictors is large compared to the number of samples.

1.2 Multicollinearity Challenges

Multicollinearity refers to the situation where two or more features are strongly linearly related. This leads to:

- Unstable estimates of β_j,
- Increased variance in the model,
- Reduced interpretability,
- Poor generalization to new data.

In house pricing data, for instance, features like total square footage and number of rooms can be highly correlated.

1.3 Regularization Techniques

To address overfitting and multicollinearity, regularization introduces a penalty term to the loss function:

- **Ridge Regression (L2 penalty):**

$$\min_{\beta} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{n} \beta_j^2$$

- **Lasso Regression (L1 penalty):**

$$\min_{\beta} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{n} |\beta_j|$$

Lasso is preferred when we expect some features to be irrelevant, as it can shrink coefficients to zero (feature selection).

1.4 Mathematical Explanation of Lasso Regression

Loss Function

The objective function for Lasso Regression is:

$$\mathcal{L}(\beta) = \frac{1}{2m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{n} |\beta_j|$$

Where:

- y_i: Actual value,
- $\hat{y}_i$: Predicted value,
- β_j: Model coefficients,
- α: Regularization parameter controlling the strength of the L1 penalty.

Constraints

The L1 penalty introduces a constraint equivalent to:

$$\sum_{j=1}^{n} |\beta_j| \leq t$$

for some constant t. This constrains the total absolute magnitude of the coefficients, encouraging sparsity (some $\beta_j = 0$).

Parameters

There are two types of parameters in the Lasso model:

- **Model coefficients** β_j – learned during training,
- **Regularization strength** α – selected via cross-validation.

Larger values of α increase the penalty and shrink more coefficients to zero.

Algorithms for Solving Lasso

Since the L1 norm is not differentiable at zero, Lasso requires special optimization algorithms:

- **Coordinate Descent:** Updates one coefficient at a time while keeping others fixed. Efficient and commonly used.
- **Least Angle Regression (LARS):** Tracks the entire solution path as α varies. Useful for high-dimensional problems.
- **Subgradient Methods:** Used in gradient-based approaches when standard derivatives do not exist.

Geometric Intuition

In two dimensions, the L1 constraint forms a diamond shape. The corners of the diamond align with the coordinate axes, making it more likely that the optimal solution lies on an axis (i.e., some coefficients are zero). This gives Lasso its feature selection property.

1.5 Cross-Validation for Hyperparameter Tuning

To find the optimal regularization parameter α, k-fold cross-validation is used:

1. Divide data into k subsets,
2. Train the model on $k - 1$ subsets, validate on the remaining one,
3. Repeat for each fold and compute average performance (e.g., MSE),
4. Select the α value that minimizes validation error.

1.6 Summary

This chapter introduced linear regression, highlighted the challenges of multicollinearity, and motivated the use of Lasso Regression. It also presented the mathematical foundation of Lasso, including its objective function, constraints, key parameters, and optimization algorithms. The next chapter will apply these concepts to the problem of house price prediction using real-world data.

Chapter 2

Model Lasso Regression for House Price Prediction

2.1 Mathematical Formulation

In standard linear regression, the predicted value $\hat{y}$ is modeled as a linear combination of input features:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

where:

- $\hat{y}$ is the predicted house price,
- x_i are the input features (e.g., house area, number of bedrooms, location),
- β_i are the coefficients to be learned.

Lasso Regression modifies the loss function by adding an L_1-norm penalty to the sum of squared errors:

$$\mathcal{L}(\beta) = \frac{1}{2m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 + \alpha \sum_{j=1}^{n} |\beta_j|$$

Where:

- m is the number of training samples,
- $\alpha \geq 0$ is the regularization parameter (controls the amount of shrinkage),
- $\sum_{j=1}^{n} |\beta_j|$ is the L_1-norm penalty, which encourages sparsity in β.

2.2 Why is Lasso Suitable for Predicting House Prices?

Real estate price data often contains a wide range of variables, such as:

- Structural details of the house (e.g., square footage, number of rooms, age)

- Additional amenities (like garages or swimming pools)
- Location-specific factors (e.g., neighborhood, distance to schools or city centers)

However, not every feature has the same level of impact on the final price. That's where Lasso Regression comes in handy. It offers:

- Automatic selection of the most significant variables
- A streamlined model with reduced complexity
- Better generalization by removing unnecessary or redundant features, helping prevent overfitting

2.3 How to Train a Lasso Model

Training a Lasso model involves several key steps:

1. Splitting the dataset into training and testing sets
2. Normalizing the features so they share a consistent scale
3. Using cross-validation to choose the best regularization parameter (α)
4. Applying an optimization technique such as coordinate descent to fit the model

2.4 What Does the Model Produce?

Once the training is complete, the Lasso model delivers:

- A set of coefficients (β) for the input variables—many of which are zero, indicating exclusion from the model
- A predictive formula for estimating house prices based on the inputs
- A clear view of which features most influence housing prices

2.5 Conclusion

Lasso Regression is a robust and practical choice for predicting house prices, especially when working with high-dimensional data. It balances model accuracy with simplicity and interpretability—making it an excellent tool for real estate analytics, where understanding the key drivers behind price is as valuable as the prediction itself.

Chapter 3

Training the Lasso Model Regression for House Price Prediction

Research and apply the Lasso Regression algorithm to build a model for predicting house prices based on features such as area, number of bedrooms, number of bathrooms, the age of the house, and the presence of a garage. The input data is a sample dataset consisting of multiple houses with relevant attributes and corresponding selling prices as follows:

Area	Bedrooms	Age	Price
3221	7	I	221614
2723	7	9	397043
3745	H	11	340408
2908	7	6	348994
3909	6	1	320214
1434	3	3	335810
3948	F	6	488348
1618	3	9	433723
3012	3	4	291037
1723	4	2	436418
1948	7	12	232751
3037	5	11	308812
2775	7	10	249976
3434	3	5	230074
1497	2	12	497069
E	7	1	195829
1314	5	11	281552
2952	7	J	357813
1756	2	15	300208
1334	2	3	244251
1260	5	13	478239
3915	5	2	414112

Area	Bedrooms	Age	Price
2323	2	6	428282
2942	2	11	164395
G	2	F	203825
I	7	15	319992
1320	3	3	210905
1831	5	2	107842
3799	7	2	120998
2314	3	2	153311
1154	6	15	316273
2793	6	12	189083
2568	5	3	274150
2404	6	9	193931
2681	2	14	333890
3947	6	6	311657
F	3	1	256759
1131	4	1	465955
1749	2	2	490379
1379	3	14	132664
D	7	13	145031
1864	4	C	209237
2618	6	5	491656
3398	7	9	328775
3520	4	12	267958
H	2	11	229701
1067	3	2	288069
1560	3	1	147921
G	H	15	204378
2676	B	6	227307

3.1 Clean Non-Numeric Rows in Dataset

As part of the data preprocessing step, we cleaned the dataset by removing rows containing invalid (non-numeric) entries in three key columns: Area, Bedrooms, and Age. The remaining values were then converted to floating-point numbers to ensure consistency and prepare the data for further analysis and modeling. This resulted in a clean and reliable dataset ready for use in machine learning tasks.

Code Python of Clean Non-Numeric Rows in Dataset

```python
import pandas as pd

# Load the dataset
df = pd.read_csv("house_price_missing_letters_lasso_friendly.csv")

# Columns to clean
cols_to_check = ['Area', 'Bedrooms', 'Age']

# Function to check if a value is numeric
def is_numeric(val):
    try:
        float(val)
        return True
    except:
        return False

# Keep rows where all three columns are numeric
mask = df[cols_to_check].applymap(is_numeric).all(axis=1)
filtered_df = df[mask].copy()  # .copy() to avoid
    SettingWithCopyWarning

# Convert numeric columns to float
filtered_df[cols_to_check] = filtered_df[cols_to_check].astype(float)

# Save cleaned data to CSV
filtered_df.to_csv("house_price_clean_numeric.csv", index=False)

# Print the cleaned data
print("Cleaned data:")
print(filtered_df)  # This will print the entire cleaned dataset
```

Python Output (Cleaned Data)

```
Cleaned data:
       Area  Bedrooms   Age   Price
1    2723.0       7.0   9.0  397043
3    2908.0       7.0   6.0  348994
4    3909.0       6.0   1.0  320214
5    1434.0       3.0   3.0  335810
7    1618.0       3.0   9.0  433723
8    3012.0       3.0   4.0  291037
9    1723.0       4.0   2.0  436418
10   1948.0       7.0  12.0  232751
11   3037.0       5.0  11.0  308812
12   2775.0       7.0  10.0  249976
13   3434.0       3.0   5.0  230074
14   1497.0       2.0  12.0  497069
16   1314.0       5.0  11.0  281552
18   1756.0       2.0  15.0  300208
19   1334.0       2.0   3.0  244251
20   1260.0       5.0  13.0  478239
21   3915.0       5.0   2.0  414112
22   2323.0       2.0   6.0  428282
23   2942.0       2.0  11.0  164395
26   1320.0       3.0   3.0  210905
27   1831.0       5.0   2.0  107842
28   3799.0       7.0   2.0  120998
29   2314.0       3.0   2.0  153311
30   1154.0       6.0  15.0  316273
31   2793.0       6.0  12.0  189083
32   2568.0       5.0   3.0  274150
33   2404.0       6.0   9.0  193931
34   2681.0       2.0  14.0  333890
35   3947.0       6.0   6.0  311657
37   1131.0       4.0   1.0  465955
38   1749.0       2.0   2.0  490379
39   1379.0       3.0  14.0  132664
42   2618.0       6.0   5.0  491656
43   3398.0       7.0   9.0  328775
44   3520.0       4.0  12.0  267958
46   1067.0       3.0   2.0  288069
47   1560.0       3.0   1.0  147921
<ipython-input-3-bfb9acad1f88>:18: FutureWarning: DataFrame.applymap
   has been deprecated. Use DataFrame.map instead.
  mask = df[cols_to_check].applymap(is_numeric).all(axis=1)
\section{Lasso Regression in Python}\
```

3.2 Lasso Regression in Python

This Python script demonstrates how to apply Lasso Regression to predict house prices based on various property features. Lasso Regression is a linear model that includes an L_1 penalty term, which encourages sparsity in the model by reducing less important feature coefficients to zero. This makes it especially useful for feature selection in high-dimensional datasets.

The dataset used in this example, `house_price_missing_letters_lasso_friendly.csv`, contains categorical variables that are preprocessed using one-hot encoding. After preprocessing, the data is split into training and test sets, and a Lasso model is trained using a regularization parameter $\alpha = 0.1$.

The script concludes by evaluating the model performance using Root Mean Squared Error (RMSE) and ranking the features based on the absolute value of their coefficients. The most influential features in predicting house prices are highlighted and optionally saved to a CSV file for further analysis.

This example provides a practical workflow for using Lasso Regression in predictive modeling and highlights its ability to perform both regression and feature selection.

```python
import pandas as pd
import numpy as np
from sklearn.linear_model import Lasso
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

# 1. Load the dataset
try:
    df = pd.read_csv("house_price_missing_letters_lasso_friendly.csv")
except FileNotFoundError:
    print("Error: File 'house_price_missing_letters_lasso_friendly.csv' not found.")
    exit()

# 2. Convert columns with mixed data to string type
for col in ['Area', 'Bedrooms', 'Age']:
    df[col] = df[col].astype(str)

# 3. Separate features and target
X = df.drop(columns=["Price"])
y = df["Price"]

# 4. One-hot encode categorical variables
X_encoded = pd.get_dummies(X, drop_first=True)

# 5. Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(
    X_encoded, y, test_size=0.2, random_state=42
)

# 6. Train the Lasso regression model
lasso = Lasso(alpha=0.1)
lasso.fit(X_train, y_train)

# 7. Predict and calculate RMSE
```

```
y_pred = lasso.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))

# 8. Create a DataFrame of feature importances
coef_df = pd.DataFrame({
    "Feature": X_encoded.columns,
    "Coefficient": lasso.coef_
})
coef_df["Importance"] = coef_df["Coefficient"].abs()
ranked_features = coef_df[coef_df["Coefficient"] != 0].sort_values(by=
    "Importance", ascending=False)

# 9. Display results
print("Intercept:", round(lasso.intercept_, 2))
print("RMSE on test set:", round(rmse, 2))

# Configure pandas to display the entire DataFrame without truncation
pd.set_option("display.max_rows", None)
pd.set_option("display.max_columns", None)
pd.set_option("display.width", None)
pd.set_option("display.max_colwidth", None)

print("\nRanked features by importance:")
print(ranked_features)

# 10. (Optional) Save the results to a CSV file
ranked_features.to_csv("lasso_feature_importance.csv", index=False)
print("\nFeature importances have been saved to '
    lasso_feature_importance.csv'")
```

:

Python Output Lasso Regression

```
Intercept: 283038.13
RMSE on test set: 69254.23

Ranked features by importance:
         Feature    Coefficient      Importance
8     Area_1497   213480.331648  213480.331648
41    Area_3948   212642.638178  212642.638178
21    Area_2618   209725.021022  209725.021022
12    Area_1749   207861.078457  207861.078457
2     Area_1260   189134.483376  189134.483376
0     Area_1131   178924.119834  178924.119834
14    Area_1831  -178728.766830  178728.766830
37    Area_3799  -160955.571407  160955.571407
11    Area_1723   149909.299463  149909.299463
18    Area_2323   145941.232612  145941.232612
10    Area_1618   145578.455016  145578.455016
9     Area_1560  -140611.087191  140611.087191
42       Area_D  -139450.380916  139450.380916
17    Area_2314  -134699.957584  134699.957584
39    Area_3915   127531.554440  127531.554440
28    Area_2942  -119360.262815  119360.262815
24    Area_2723   114954.815801  114954.815801
26    Area_2793   -95475.861279   95475.861279
19    Area_2404   -89682.562653   89682.562653
43       Area_E   -86648.060463   86648.060463
45       Area_G   -80457.945078   80457.945078
15    Area_1864   -74387.037589   74387.037589
27    Area_2908    67210.834678   67210.834678
32    Area_3221   -62450.963887   62450.963887
36    Area_3745    57200.734414   57200.734414
22    Area_2676   -55410.897491   55410.897491
23    Area_2681    51563.892039   51563.892039
16    Area_1948   -50272.934209   50272.934209
33    Area_3398    46686.724827   46686.724827
7     Area_1434    43148.510659   43148.510659
38    Area_3909    36195.796716   36195.796716
44       Area_F   -31776.694388   31776.694388
25    Area_2775   -29894.274850   29894.274850
40    Area_3947    28339.386298   28339.386298
13    Area_1756    22411.809842   22411.809842
31    Area_3037    20988.148549   20988.148549
35    Area_3520   -19616.301571   19616.301571
30    Area_3012     7583.138325    7583.138325
54   Bedrooms_F    -6628.991976    6628.991976
3     Area_1314    -6261.120438    6261.120438
48   Bedrooms_3     5506.283262    5506.283262
61       Age_15    -5239.599384    5239.599384
64        Age_4    -5087.257313    5087.257313
63        Age_3     4112.330780    4112.330780
50   Bedrooms_5     4058.787421    4058.787421
```

```
49   Bedrooms_4     3988.362343     3988.362343
68        Age_C    -3398.139362     3398.139362
56       Age_10    -2604.877759     2604.877759
65        Age_5    -2078.717956     2078.717956
59       Age_13     2001.460271     2001.460271
70        Age_I     1582.790678     1582.790678
69        Age_F     1240.965896     1240.965896
51   Bedrooms_6      975.060566      975.060566
57       Age_11      717.208743      717.208743
60       Age_14     -709.809132      709.809132
66        Age_6     -696.624115      696.624115
52   Bedrooms_7     -559.957989      559.957989
55   Bedrooms_H     -548.868585      548.868585
58       Age_12      544.644324      544.644324
62        Age_2     -516.240138      516.240138
67        Age_9     -393.581107      393.581107
53   Bedrooms_B      375.343535      375.343535

Feature importances have been saved to 'lasso_feature_importance.csv'
```

Chapter 4

Conclusions and Future Applications of House Price Prediction

4.1 Conclusion

In this study, we applied the Lasso Regression method to build a predictive model for housing prices based on input features, while also leveraging Lasso's ability to perform automatic feature selection through ℓ_1 regularization.

Data preprocessing played a crucial role in ensuring the accuracy and stability of the model. The original dataset contained several invalid values (e.g., letters instead of numbers in columns such as Area, Bedrooms, and Age), making it necessary to remove non-numeric rows and convert all values to floating-point numbers. Subsequently, one-hot encoding was applied to handle categorical variables, allowing the model to capture information from discrete features such as the number of bedrooms and the house's age.

After training the model with a regularization parameter $\alpha = 0.1$, the results showed that Lasso Regression was effective in reducing the number of unnecessary features by shrinking the coefficients of less relevant variables to zero. This not only simplified the model but also enhanced its interpretability.

The model achieved a Root Mean Squared Error (RMSE) of 69,254.23 on the test set, indicating reasonably good predictive performance in a real-world dataset context. Analysis of feature importance (based on the absolute values of the regression coefficients) revealed that:

- Area-related variables dominated the most important features. Specific values such as Area_1497, Area_3948, and Area_2618 had large coefficients, reflecting a strong linear relationship between property size and its price.
- Some features related to Bedrooms and Age also contributed to the model, although their coefficients were much smaller, indicating relatively limited impact.
- The presence of unusual feature names (e.g., Bedrooms_F, Age_C) suggests that some non-numeric values may have remained during preprocessing, emphasizing

the importance of rigorous data cleaning.

Lasso's ability to eliminate non-contributing features helped the model avoid overfitting, reduced noise, and improved interpretability.

In summary, Lasso Regression is a highly useful tool for regression tasks involving multiple input variables. It not only provides effective prediction but also performs automatic feature selection, making it particularly suitable for datasets with potential redundancy. The findings in this study highlight that combining thorough data preprocessing with Lasso Regression can yield models that are both robust and practical for real-world applications, especially in real estate price estimation.

4.2 Future Applications

The findings from this study using Lasso Regression have significant implications for future applications in various domains, particularly in real estate and housing price prediction. However, the potential of Lasso Regression extends beyond just housing price estimation. Here are several areas where this technique can be applied:

- **Real Estate Market Analysis:** The ability of Lasso Regression to select relevant features can be further exploited to analyze the factors influencing house prices in different geographical locations or during different market conditions. By incorporating additional factors like neighborhood amenities, proximity to schools, and transportation networks, future models could become more comprehensive in capturing the underlying dynamics of housing prices.

- **Personalized Property Valuation:** Lasso Regression can be employed to create personalized property valuation models for individual buyers or sellers. By tailoring the model to a specific region, property type, or buyer preferences, real estate agents can provide more accurate price estimates, helping clients make informed decisions.

- **Urban Planning and Development:** Urban planners can use Lasso Regression in the context of city development projects. By examining factors such as land usage, infrastructure, and population demographics, it can be possible to predict how new developments will affect property prices, aiding decision-making on zoning laws and public investment.

- **Predictive Maintenance in Real Estate:** Another future application could involve predicting maintenance needs for residential or commercial properties. By analyzing past maintenance records and property features, a Lasso Regression model could forecast when certain property components (e.g., roofing, plumbing, HVAC) are likely to fail, enabling proactive maintenance scheduling and cost-saving for property owners.

- **Financial Portfolio Optimization:** Lasso Regression could be utilized in the field of financial analytics for real estate investment portfolio optimization. By modeling the expected return on investment based on property features, investors can prioritize properties that yield higher returns, factoring in risks associated with market volatility.

- **Integration with Machine Learning and AI:** Future studies could explore integrating Lasso Regression with more advanced machine learning models, such as neural networks or reinforcement learning. By combining the interpretability of Lasso with the flexibility of deep learning, more complex and adaptive models can be developed to address emerging challenges in real estate markets and other industries.

In conclusion, the future applications of Lasso Regression in the real estate sector and beyond are vast. Its strength in feature selection, coupled with its simplicity and efficiency, makes it an ideal candidate for a wide array of predictive modeling tasks. As more data becomes available and computational power increases, Lasso Regression can continue to play a pivotal role in enhancing decision-making processes in various fields.