A Comprehensive Analysis of House Price Prediction

Mao Kaicheng **Xie Shicheng** **Hou Zheyan** **Chewprecha Tong Bhuree**

Abstract

This study presents a comprehensive analysis of the Ames Housing dataset, employing a multi-faceted approach to house price prediction. We systematically address data quality issues through domain-informed imputation and encoding strategies, followed by rigorous statistical testing to validate feature significance. The research incorporates advanced feature engineering techniques, including neighborhood binning and redundancy elimination, to enhance model performance. We compare multiple machine learning approaches, including regularized linear models (Ridge, Lasso, Elastic Net), tree-based methods (XGBoost), and ensemble techniques. Our analysis reveals that location (Neighborhood), overall quality (OverallQual), and living area (GrLivArea) are the primary determinants of house prices. The ensemble model combining Lasso regression and XGBoost achieves superior predictive performance with a cross-validated RMSE of 0.115 on the log-transformed scale. The methodology demonstrates the value of integrating classical statistical methods with modern machine learning techniques for robust predictive modeling in real estate analytics.

1 Introduction

Real estate valuation represents a complex econometric challenge that requires careful consideration of numerous property characteristics, location factors, and market conditions. The ability to accurately predict house prices has significant implications for various stakeholders, including home buyers, sellers, investors, and financial institutions. Traditional approaches to property valuation have often relied on comparative market analysis and appraiser judgment, but these methods can be subjective and inconsistent.

With the advent of data science and machine learning, there has been growing interest in developing data-driven approaches to house price prediction. The Kaggle "House Prices: Advanced Regression Techniques" competition provides the Ames Housing dataset, which contains 79 explanatory variables describing various aspects of residential homes in Ames, Iowa. This dataset presents an excellent opportunity to explore the application of statistical and machine learning methods to real estate valuation.

This paper contributes to the literature by presenting a comprehensive analytical framework that integrates:

1. Systematic data preprocessing with domain-informed imputation strategies
2. Rigorous statistical testing to validate feature significance
3. Advanced feature engineering techniques
4. Comparative analysis of multiple modeling approaches
5. Ensemble methods combining linear and nonlinear models

Our methodology demonstrates how classical statistical methods can complement modern machine learning techniques to produce robust and interpretable predictive models.

2 Data Description and Preprocessing

2.1 Dataset Overview

The Ames Housing dataset consists of 2,919 observations (1,460 training, 1,459 testing) with 79 explanatory variables and one response variable (SalePrice). The variables encompass various property characteristics, including:

- **Continuous variables:** Lot area, living area, basement area, garage area, etc.
- **Ordinal categorical variables:** Quality ratings (Exterior, Kitchen, Basement, etc.)
- **Nominal categorical variables:** Neighborhood, house style, roof type, etc.
- **Temporal variables:** Year built, year remodeled, month/year sold

The training and test sets were combined into a single dataset (`all`) to ensure consistent preprocessing across both partitions.

2.2 Missing Value Imputation

We identified 34 predictors with missing values, each requiring specific imputation strategies based on domain knowledge. Table 1 summarizes our imputation approach.

Variable	Missing Meaning	Imputation Strategy
PoolQC	No swimming pool	"None" (encoded as 0)
Garage variables	No garage	"None" for categorical, 0 for numeric
Basement variables	No basement	"None" for categorical, 0 for numeric
LotFrontage	Unknown street frontage	Neighborhood median
MSZoning	Unknown zoning classification	Mode (most frequent category)
Electrical	Unknown electrical system	Mode
MasVnrType	No masonry veneer	"None"
Utilities	All public utilities for test set	Removed (insufficient variance)

Table 1: Imputation strategies for selected variables with missing values

A more detailed explanation for our imputation(cleaning) approach is illustrated below:

- **Structural absence:** For pools, set `PoolQC` NAs to "None", ordinally encode them, and give the three homes with `PoolArea` > 0 but missing quality scores plausible values guided by `OverallQual`.
- **Location-based numeric imputation:** Replace missing `LotFrontage` with the median frontage of the same `Neighborhood` to preserve local lot-size patterns.
- **Garage consistency fixes:** Fill missing `GarageYrBlt` with `YearBuilt`, mode-impute garage finish/quality for the single inconsistent house (2127), and set garage size/count to zero while flagging "No Garage" for house 2577 before ordinal encoding of quality/finish/condition.
- **Basements:** Identify 79 homes without basements, set their categorical basement fields to "None" and numeric areas/baths to zero, and use mode fills for the few partially observed basements prior to ordinal encoding of quality, exposure, and finish types.
- **Materials alignment:** Make `MasVnrType` consistent with `MasVnrArea` by imputing the lone missing type with the mode, labeling remaining NAs as "None", and setting veneer area to zero when no masonry exists.
- **Single-NA categoricals:** Mode-impute `MSZoning`, `Electrical`, `SaleType`, and `KitchenQual`; drop `Utilities` because only one training home lacked public utilities, leaving the feature uninformative for the test set.

2.3 Encoding Categorical Variables

Categorical variables were processed according to their semantic meaning:

- **Ordinal variables:** Quality ratings (e.g., ExterQual, KitchenQual) were mapped to numerical scales using predefined quality dictionaries.
- **Nominal variables:** Non-ordinal categories (e.g., Neighborhood, HouseStyle) were converted to factors for one-hot encoding.
- **Temporal variables:** Month sold and year sold, though numeric, were treated as categorical factors.

Quality-style fields (e.g., `PoolQC`, `FireplaceQu`, `BsmtQual/Cond`, `GarageQual/Cond`, `ExterQual/Cond`) share a common ordinal map "None/Po/Fa/TA/Gd/Ex" $\rightarrow$ 0–5, while borderline cases like `Fence` and `Alley` were left nominal after checking that higher labels did not imply higher prices. For timing effects, `YrSold` stays numeric only long enough to build age features, after which both `YrSold` and `MoSold` are converted to factors to capture crisis-year breaks and seasonal patterns without imposing linear order.

2.4 Skewness Reduction and Standardization

For numeric predictors with absolute skewness exceeding 0.8, we applied log transformation:

$$x_{\text{transformed}} = \log(x + 1)$$

All numeric variables were subsequently centered and scaled to have zero mean and unit variance:

$$x_{\text{standardized}} = \frac{x - \mu}{\sigma}$$

3 Exploratory Data Analysis

3.1 Target Variable Distribution

The distribution of SalePrice exhibited significant right-skewness, with a median of \$163,000 and mean of \$180,921. This skewness motivated the use of log transformation for subsequent modeling:

$$y_{\text{transformed}} = \log(\text{SalePrice})$$

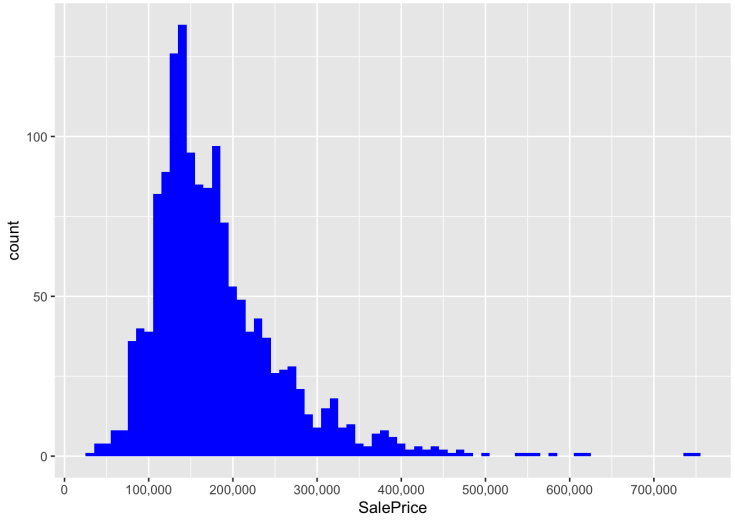


Figure 1: Histogram of SalePrice (training set)

3.2 Correlation Analysis

Pearson correlation analysis identified 16 numeric variables with correlation coefficients exceeding 0.5 with SalePrice (Table 2). The strongest correlations were observed with OverallQual (0.79), GrLivArea (0.71), and GarageCars (0.64).

Variable	Correlation with SalePrice
OverallQual	0.79
GrLivArea	0.71
GarageCars	0.64
GarageArea	0.62
TotalBsmtSF	0.61
1stFlrSF	0.61
FullBath	0.56
TotRmsAbvGrd	0.53
YearBuilt	0.52
YearRemodAdd	0.51

Table 2: Top numeric variables correlated with SalePrice

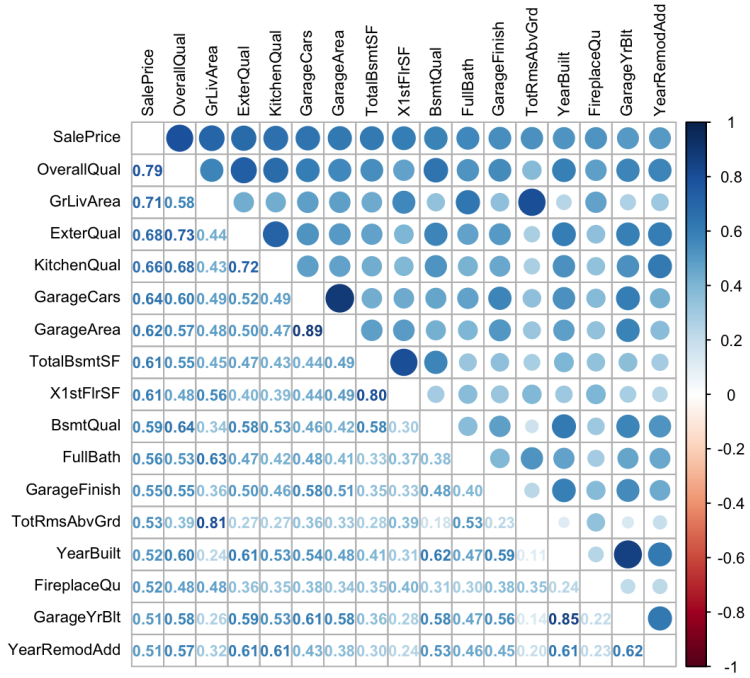
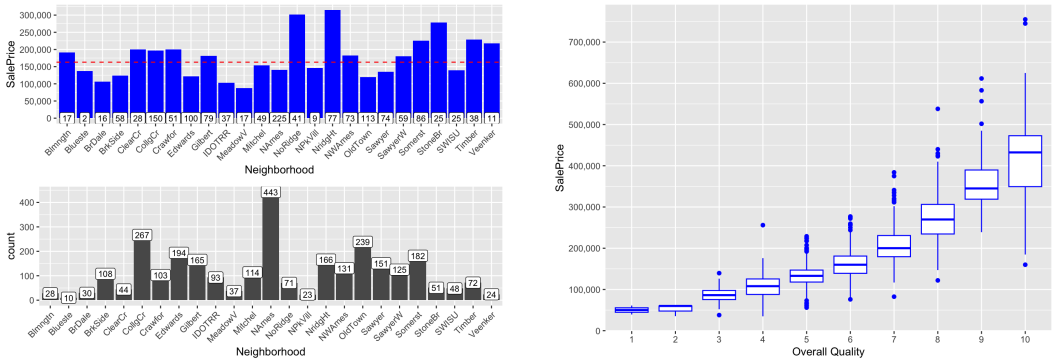


Figure 2: Correlation heatmap of predictors with $|r| > 0.5$

3.3 Important Variable Groups

Three primary variable groups emerged as key predictors:

1. **Quality variables:** OverallQual, ExterQual, KitchenQual showed strong monotonic relationships with price.
2. **Area variables:** GrLivArea, TotalBsmtSF, GarageArea served as key size indicators.
3. **Location variables:** Neighborhood exhibited substantial price variation across different areas.



Neighborhood medians and frequencies SalePrice by Overall Quality

Figure 3: Representative visualizations of the most important variables

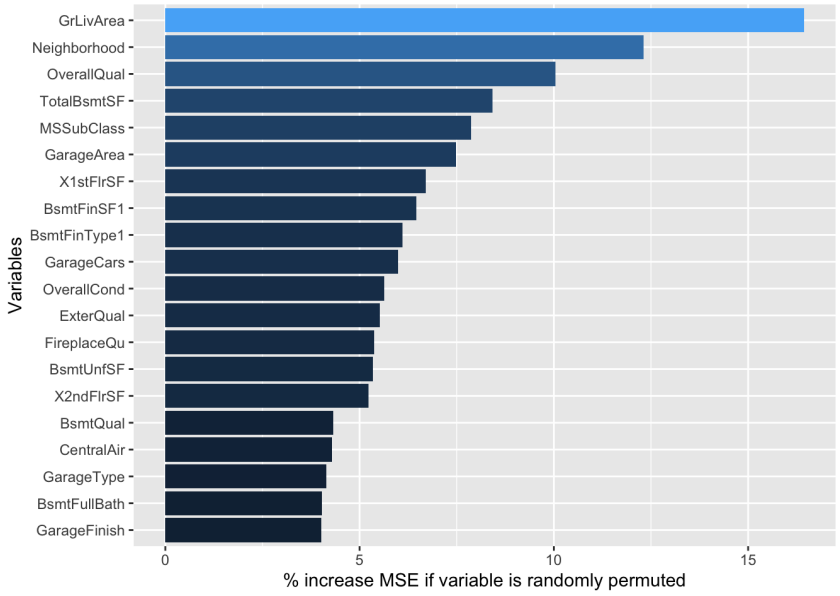
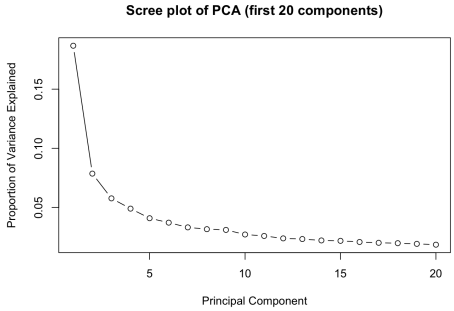


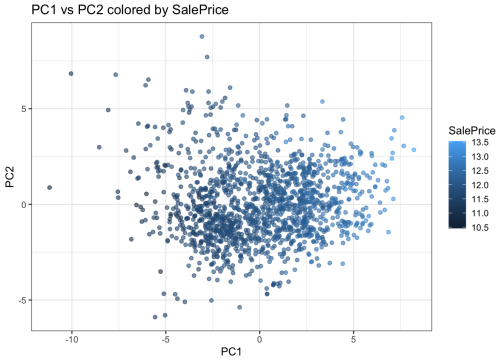
Figure 4: Top 20 variable importances from the quick Random Forest

3.4 Unsupervised exploration: PCA and clustering

This exploratory step (not used downstream for modelling features) checks the global structure of the cleaned predictors. A PCA on the standardised numeric set shows that the first five components explain about 41% of the variance; PC1 is dominated by size/quality attributes (e.g., `GarageCars`, `ExterQual`), and PC2 mixes surface and age signals. Coloring PC1–PC2 by `SalePrice` reveals a smooth price gradient along PC1, indicating that the dominant linear combination aligns with value.



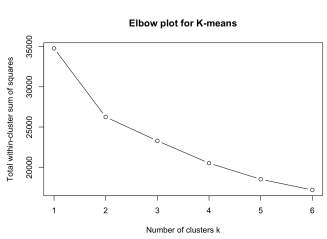
PCA scree plot (first 20 PCs)



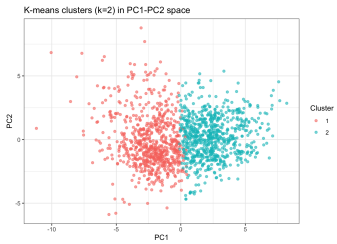
PC1 vs PC2 colored by SalePrice

Figure 5: PCA overview (exploratory only)

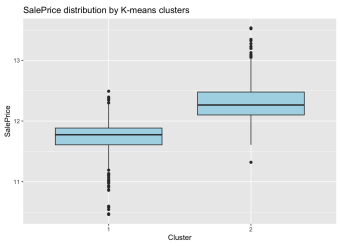
On the first eight PCs, a quick K-means yields an elbow at $k = 2$. The two clusters largely separate higher and lower priced homes, with a clear shift in the SalePrice boxplots; however, we treat this as descriptive structure rather than a modelling feature.



Elbow plot for K-means



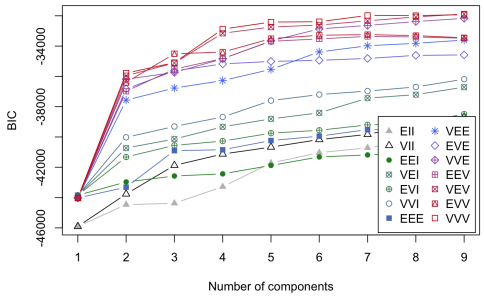
Clusters in PC1–PC2 space



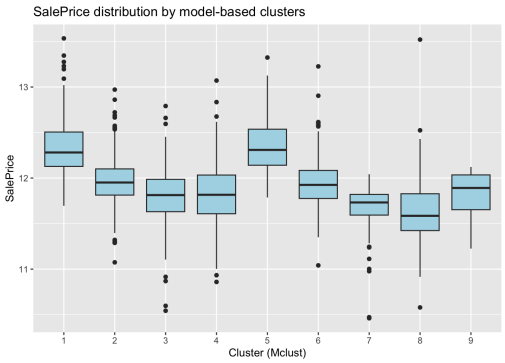
SalePrice by K-means cluster

Figure 6: K-means clustering on leading PCs (exploratory)

A Gaussian mixture model on the same PCs selects nine components by BIC, further hinting at multiple submarkets. We did not propagate these unsupervised labels into the supervised models.



BIC curve for model-based clustering



SalePrice by mixture component

Figure 7: Model-based clustering diagnostics (exploratory)

4 Feature Engineering

4.1 `Neighborhood` Binning

Upon completing the aforementioned steps, we noted that one variable required special attention and additional processing. The `Neighborhood` variable originally contained an excessive number of levels, necessitating its aggregation into domain-based groups. This grouping helps avoid generating a high-dimensional sparse matrix after one-hot encoding and, to some extent, prevents model overfitting.

To achieve this, we first examined the distribution of the mean and median of the `SalePrice` variable for each category of `Neighborhood`. We aimed to follow these grouping principles:

- Ensuring each group had a sufficient sample size
- Selecting natural break points as group boundaries

The natural breaks in this data were clear, exhibiting small, visible jumps in distribution between adjacent groups. Moreover, a favorable property of using these natural breaks was that the median or mean value of each resulting group remained within its own range without erroneously falling into another group.Therefore, we accordingly bin the `Neighborhood` into the following four groups based on natural breaks.

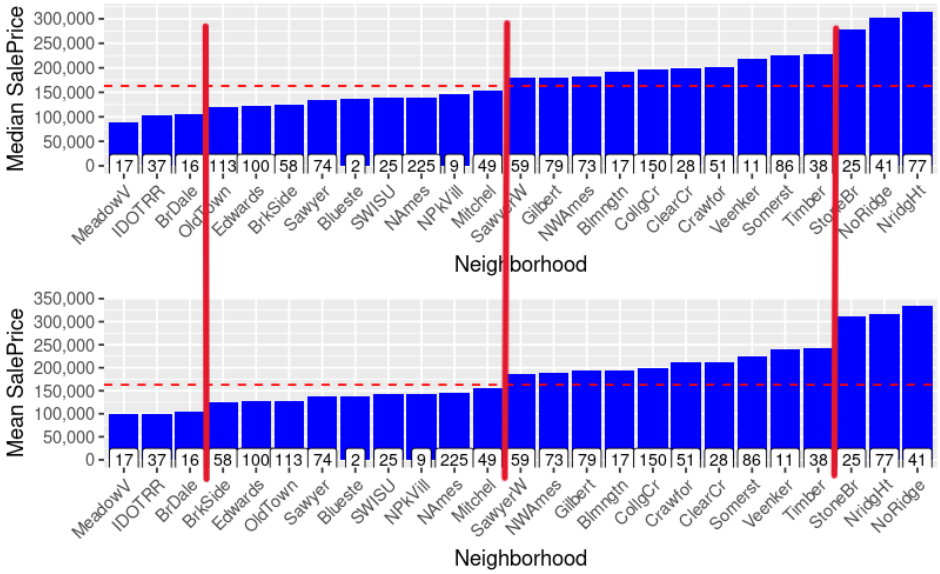


Figure 8: The mean and median for each category of `Neighborhood`

After completing the grouping, we assign labels according to the following criteria.

- 0 = low-price group
- 1 = middle-low-price group
- 2 = middle-high-price group
- 3 = high-price group

After this step, we observed that the spacing of `SalePrice` across the groups was not uniform. Therefore, we treated `NeighRich` as a factor (categorical variable). Following this, the original `Neighborhood` variable was discarded.

4.2 Redundant Feature Elimination

Subsequently, we were about to discard several highly correlated variables, as including them simultaneously in a model would unduly amplify their collective influence in that direction.

- **Rule for Discarding Highly Correlated Variables:**
  For each highly correlated pair, we drop the one with weaker correlation to `SalePrice`,while keeping the more informative representative feature.
- **Example:**
  `GarageCars` and `GarageArea` have a correlation of 0.89. Of those two, we will dropping the variable `GarageArea` (`GarageArea` with a `SalePrice` correlation of 0.62. `GarageCars` has a `SalePrice` correlation of 0.64).

Based on the above criteria, the following 7 highly correlated variables were discarded at this step:

- `YearRemodAdd`
- `GarageYrBlt`
- `GarageArea`
- `GarageCond`
- `TotalBsmtSF`
- `TotalRmsAbvGrd`
- `BsmtFinSF1`

4.3 Outlier Removal

During the feature engineering stage, we detected and addressed outliers. Outliers are observations that deviate significantly from the overall distribution of the data. Their presence can be attributed to causes such as data entry errors, measurement errors, or rare extreme events. Outliers can adversely impact our models: the predictive models we employ later are highly sensitive to outliers. A few extreme values can significantly skew the estimation of model parameters, thereby reducing the model's predictive accuracy and robustness. Therefore, it is necessary to remove these outliers in advance.

During our examination of the relationship between `GrLivArea` and `SalePrice`, we noted 2 questionable data points.

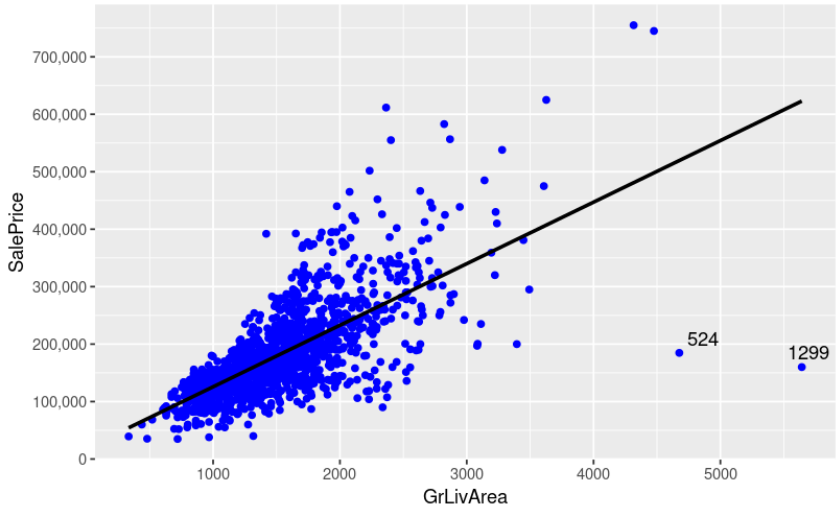


Figure 9: Outliers detection

Concerning data points 524 and 1299, these houses possessed very large areas but were priced surprisingly low. We therefore justifiably treated them as outliers and removed them from the dataset.

5 ANOVA for Categorical Variables

Next, we aimed to investigate the categorical variables in the dataset to determine whether these categories were statistically significant and to understand their main effects. This analysis could potentially provide an interpretative basis for the final model results.

5.1 One-way ANOVA for Categorical Variables

To evaluate the independent impact of categorical variables on `SalePrice`, this study systematically tested all categorical variables in the dataset using One-Way Analysis of Variance. The core principle of ANOVA is to examine whether there are significant differences in the mean of a continuous dependent variable across different levels of one or more categorical variables by comparing between-group variance to within-group variance.

The null hypothesis (H_0) of this analysis states that the mean `SalePrice` is equal across all levels of a categorical variable; the alternative hypothesis (H_1) posits that there is a significant difference in mean `SalePrice` between at least two levels. In addition to statistical significance, we further calculated the η^2 (Eta Squared) effect size, which measures the proportion of variance in `SalePrice` explained by the categorical variable. The formula for η^2 is as follows:

$$\eta^2 = \frac{SS_{between}}{SS_{total}}$$

, where $SS_{between}$ represents the between-group sum of squares, and SS_{total} represents the total sum of squares. The value of η^2 lies within the range $[0, 1]$, with a larger value indicating a stronger explanatory power of the variable on `SalePrice`.

The results of the aforementioned tests and calculations of η^2 are presented in the figure below.

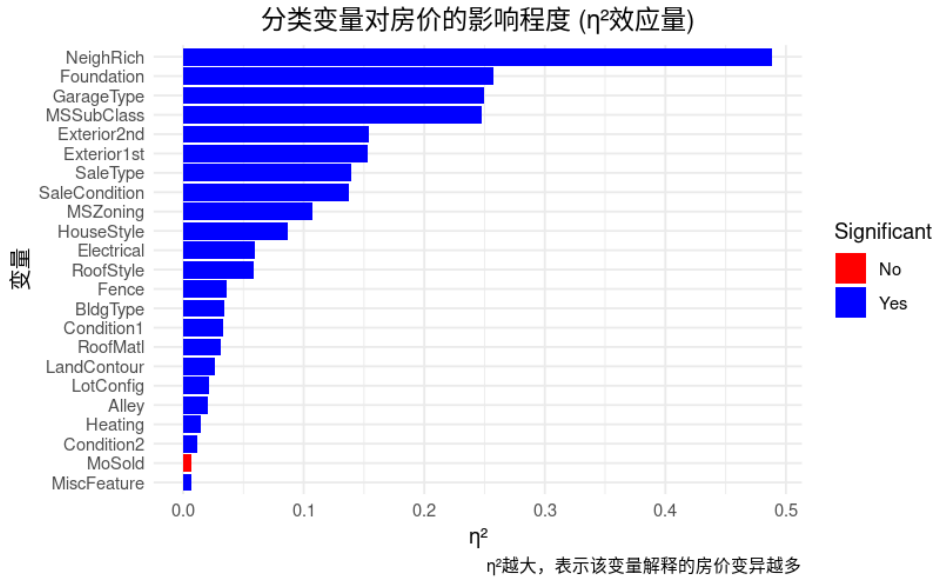


Figure 10: Visualization of p-value and eta-squared in One-way ANOVA

Subsequently, a threshold was established to identify important categorical variables: those that concurrently met the criteria of having $p < 0.05$ and an $\eta^2 > 0.01$. This approach ensures that the selected variables are both statistically significant and practically meaningful. Out of 24 initial categorical variables, 21 were considered important based on this screening under this threshold.

```
共有 24 个分类变量进行ANOVA分析
 [1] "MSSubClass"    "MSZoning"      "Alley"         "LandContour"
 [5] "LotConfig"     "Condition1"    "Condition2"    "BldgType"
 [9] "HouseStyle"    "RoofStyle"     "RoofMatl"      "Exterior1st"
[13] "Exterior2nd"   "Foundation"    "Heating"       "Electrical"
[17] "GarageType"    "Fence"         "MiscFeature"   "MoSold"
[21] "YrSold"        "SaleType"      "SaleCondition" "NeighRich"

=== 单因素ANOVA分析结果 ===

=== 显著分类变量筛选结果 ===
显著性水平: p < 0.05, 效应量阈值: η² > 0.01
筛选出 21 个重要变量:
 [1] "NeighRich"     "Foundation"    "GarageType"    "MSSubClass"
 [5] "Exterior2nd"   "Exterior1st"   "SaleType"      "SaleCondition"
 [9] "MSZoning"      "HouseStyle"    "Electrical"    "RoofStyle"
[13] "Fence"         "BldgType"      "Condition1"    "RoofMatl"
[17] "LandContour"   "LotConfig"     "Alley"         "Heating"
[21] "Condition2"
```

Figure 11: Results of One-way ANOVA

5.2 Multi-way ANOVA for Categorical Variables (Type II)

Based on the univariate analysis, we further constructed a multi-way analysis of variance model incorporating 21 important categorical variables to evaluate their joint influence and independent contributions to the `SalePrice`. Unlike one-way ANOVA, Multi-way ANOVA can simultaneously examine the main effects of multiple variables while accounting for the interrelationships among them.

This analysis employs the Type II sum of squares method, which assesses the effect of each variable while controlling for the others in the model. The rationale for selecting this model and the Type II approach are outlined below.

- **Model.**
  1. $\text{model} <- \text{lm}(y \sim A + B + C + ...)$, where $A, B, C, ...$ are categorical variables.
  2. $\text{Anova}(\text{model}, \text{type} = 2)$
- **Reasons of choosing Type II method**
  - The observed data is imbalanced.
  - Some combinations have insufficient or missing sample sizes, which hinders the testing of interaction effects.
  - The categorical variables have no apparent order.

Similar to one-way ANOVA, we can define Multi-η^2 and Partial-η^2 as follows:

Definition:
If the model is $\text{lm}(y \sim A + B + C)$

$$\text{Multi-}\eta_A^2 = \frac{SS_A}{SS_A + SS_B + SS_C + SS_{residual}}$$

$$\text{Partial-}\eta_A^2 = \frac{SS_A}{SS_A + SS_{residual}}$$

The results of the aforementioned tests and calculations of η^2 are presented in the figure below.

```
=== 模型整体解释力 ===
模型R²（解释的总变异比例）: 0.462 (46.2%)
调整后R²: 0.4541 (45.41%)

=== 与单因素ANOVA结果比较 ===

变量在单因素与Type II多因素模型中的比较:

在Type II多因素模型中仍然显著的变量（ 12 个）:
 [1] "Condition2"  "Electrical"  "Exterior1st" "Exterior2nd" "Foundation"
 [6] "GarageType"  "LandContour" "LotConfig"   "MSZoning"    "NeighRich"
[11] "RoofMatl"    "RoofStyle"

在Type II多因素模型中变得不显著的变量（ 9 个）:
[1] "Alley"        "BldgType"     "Condition1"   "Fence"
[5] "Heating"      "HouseStyle"   "MSSubClass"   "SaleCondition"
[9] "SaleType"
```

Figure 12: Results of multi-way ANOVA analysis

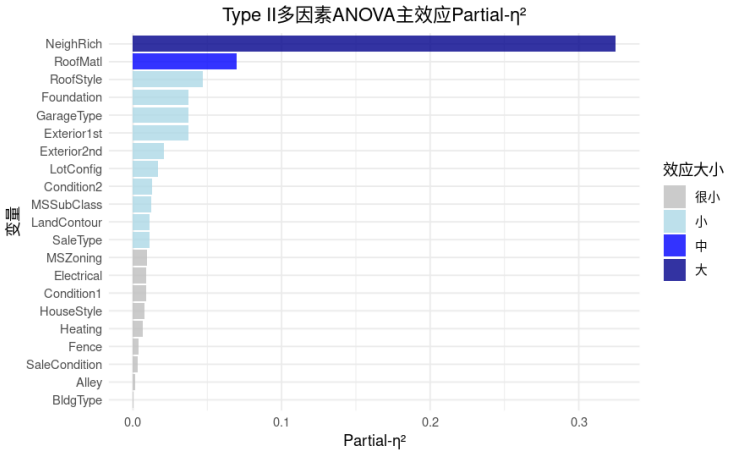


Figure 13: Visualization of Partial-η^2

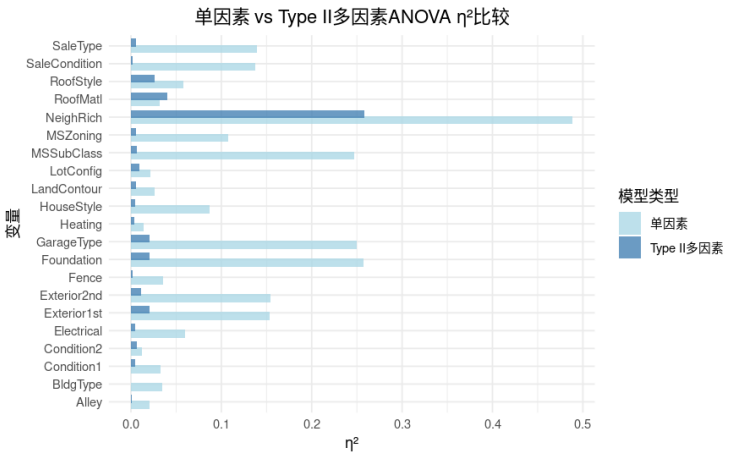


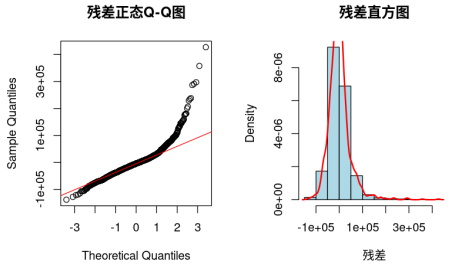
Figure 14: Comparison between uni and multi ANOVA

The variation in η^2 for most categorical variables can generally be interpreted using the following guidelines:

- **Variables with high Partial-η^2:**
  These are key predictors whose explanatory power is largely independent of other variables.
- **Variables with high Single-η^2 but low Multi-η^2:**
  These variables are highly correlated with others, sharing explanatory power (variance) within the model.

Furthermore, we discovered an interesting phenomenon regarding the variable `RoofMatl`. It was the only variable for which the η^2 value increased in the multi-way ANOVA compared to the one-way analysis (that is Multi-η^2 > Single-η^2). While a precise explanation might be elusive at this stage, but we would speculate that: `RoofMatl` might be positively correlated with `SalePrice`, while another factor exists that is also positively correlated with `SalePrice` but negatively correlated with `RoofMatl`. When this confounding factor is controlled for in the Type II ANOVA, it could unmask or amplify the strength of the positive relationship between `RoofMatl` and `SalePrice`.

After completing the multi-way ANOVA, we conducted tests for residual normality and homogeneity of variances to assess whether the fundamental assumptions of ANOVA were met.



Residuals

```
=== 模型诊断 ===

1. 残差分析:
残差统计:
  均值: 0
  标准差: 47338.83
  最小值: -137746.9
  最大值: 427253.1

残差正态性检验 (Shapiro-Wilk):
  W = 0.8733 , p = 0
  警告: 残差可能不服从正态分布 (p < 0.05)

2. 方差齐性检验:
Levene检验 (方差齐性):
  F = 1.4303 , p = 0
  警告: 方差可能不齐 (p < 0.05)
```

Diagnosis

Figure 15: Tests for residual normality and homogeneity of variances

Upon completing the multi-way ANOVA, we conducted diagnostic checks for the model's residuals. The results indicated some degree of violation regarding the assumptions of normality and homogeneity of variances. However, considering the following factors, the model's findings still hold significant reference value:

- **Large Sample Advantage:** The study benefits from a sufficiently large sample size, which reduces the risk of Type I errors.
- **Robustness of Effects:** The effect sizes of the key variables are substantial and statistically significant, making them resilient to minor assumption violations.
- **Practical Relevance:** The important variables identified by the model align well with domain knowledge in real estate, demonstrating strong interpretability.

Therefore, while statistical inference should be approached with caution, the model's ranking of variable importance and the relative magnitude of effects provide a solid foundation for subsequent feature engineering and predictive modeling.

Finally, in summary, the insights gained from performing ANOVA on categorical variables are as follows:

1. **Identifying important variables:** It discerns genuinely independent predictors in a multi-variable context, providing a complete evidence trail from statistical significance to practical importance.

2. **Inferring variable relationships:** It suggests collinearity and potential interactions by observing changes in effect sizes.
3. **Laying the groundwork for final modeling:** It supplies a basis for feature selection and a foundation for model interpretability.

6 Statistical Testing

The exploratory data analysis (EDA), based on correlation analysis, visualisations and random forest variable importance, suggested that a small set of structural and locational variables are the **dominant drivers** of house prices in Ames, Iowa. In particular, overall quality, above-ground living area, neighbourhood, year built (age), and the presence/absence of a basement and a garage appeared to have very strong relationships with the sale price.

The goal of this section is to turn these descriptive patterns into **formal statistical evidence**. We pre-specify several hypotheses based on the EDA and then apply classical tests to decide whether the observed effects are statistically significant at usual significance levels (e.g. $\alpha = 0.05$).

Throughout this section we restrict attention to the training data and work mainly with

$$Y_i = \log(\text{SalePrice}_i),$$

because the original sale price distribution is strongly right-skewed: most houses are moderately priced but a few are extremely expensive. The natural logarithm compresses the upper tail and yields a distribution that is much closer to normal, making the assumptions of t-tests and linear models more reasonable. This log transformation is standard in hedonic house-price modelling.

```
# Extract training data and compute log(SalePrice)
train_data <- all[!is.na(all$SalePrice), ]
train_data$logPrice <- log(train_data$SalePrice)     # natural log ( to address right skewness )
cat("**Number of training observations:**", nrow(train_data), "\n\n")
```

```
**Number of training observations:** 1458
```

Figure 16: Extracting the training data and performing log transformation

6.1 One-Sample t-Test

As a first step we perform a *one-sample* t-test on the log-transformed prices. Let

$$Y_i = \log(\text{SalePrice}_i), \quad i = 1, \ldots, n,$$

and denote the (unknown) population mean of Y_i by μ, the sample mean by

$$\bar{Y} = \frac{1}{n} \sum_{i=1}^{n} Y_i,$$

and the sample standard deviation by s_Y.

We take as reference value $\mu_0 = \log(\bar{P})$, where $\bar{P}$ is the sample mean of SalePrice on the original scale. In other words, the test compares the mean of the log-prices to the logarithm of the mean price:

$$H_0 : \mu = \mu_0 = \log(\bar{P}) \qquad \text{vs.} \qquad H_1 : \mu \neq \mu_0.$$

The t-test statistic is

$$t = \frac{\bar{Y} - \mu_0}{s_Y / \sqrt{n}}.$$

Under H_0, and assuming (i) independence of observations and (ii) approximate normality of Y_i, the statistic t follows a t-distribution with $n - 1$ degrees of freedom.

The EDA revealed significant right-skewness in SalePrice. To statistically confirm this skewness, we tested whether the mean of log-prices equals the log of the mean price—a condition that holds only for symmetric distributions. A one-sample t-test yielded p value < 0.001 allowing us to reject the null hypothesis. This confirms that SalePrice is positively skewed, justifying the log transformation to meet normality assumptions in subsequent models.

```r
# One-Sample t-test
# We test whether the average log-price differs from the actual sample mean
# → This is a "self-check" that must be rejected (proves the test works)

actual_mean <- mean(train_data$SalePrice)           # real mean from the data
cat("## 4.1 One-Sample t-test (Section 4.1.2)\n")
cat("Sample mean SalePrice =", round(actual_mean, 0),
    "→ log(mean) =", round(log(actual_mean), 4), "\n")
cat("**H₀:** μ_logPrice =", round(log(actual_mean), 4), "  **H₁:** μ_logPrice ≠ that value\n\n")

t.test(train_data$logPrice, mu = log(actual_mean))
```

```
## 4.1 One-Sample t-test (Section 4.1.2)
Sample mean SalePrice = 180933 → log(mean) = 12.1059
**H₀:** μ_logPrice = 12.1059   **H₁:** μ_logPrice ≠ that value


        One Sample t-test

data:  train_data$logPrice
t = -7.8211, df = 1457, p-value = 9.98e-15
alternative hypothesis: true mean is not equal to 12.10588
95 percent confidence interval:
 12.00347 12.04454
sample estimates:
mean of x
 12.02401
```

Figure 17: Performing t test

6.2 Pearson Correlation Test

We formally test the linear association between `GrLivArea` and SalePrice using the Pearson correlation test. Let $X_i = \texttt{GrLivArea}_i$ and $Z_i = \text{SalePrice}_i$. The null hypothesis is

$$H_0 : \rho = 0 \qquad \text{vs.} \qquad H_1 : \rho \neq 0,$$

where ρ is the population correlation between X and Z.

In our data, R reports a very large positive sample correlation and an extremely small p-value (practically 0 at numerical precision), so we **reject** H_0 and conclude that above-ground living area is strongly and positively associated with sale price.

```r
# Check if there is a significant linear relationship between GrLivArea and SalePrice
cor_test_result <- cor.test(train_data$GrLivArea, train_data$SalePrice)

print(cor_test_result)

# Interpretation: If the p-value is less than 0.05, we reject the null hypothesis,
# indicating a significant positive correlation between GrLivArea and SalePrice.
```

```
        Pearson's product-moment correlation

data:  train_data$GrLivArea and train_data$SalePrice
t = 41.358, df = 1456, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7104365 0.7577160
sample estimates:
      cor
0.7349682
```

Figure 18: Performing pearson correlation test

6.3 ANOVA Tests

We also fit one-way ANOVA models for SalePrice with several categorical predictors, including GarageCars, MSCubClass, and Basement Quality

For each model we test

$$H_0 : \text{all group means are equal} \qquad \text{vs.} \qquad H_1 : \text{at least one group mean differs.}$$

The F-tests from ANOVA yield very small p-values (again $p < 0.001$), showing that mean sale prices differ systematically across levels of `GarageCars`, `Basement Quality`, and `MSSubClass`. This provides another line of evidence that garage capacity and building class both have significant effects on price.

6.4 Wilcoxon Rank-Sum Tests

The EDA indicated strong differences in sale prices between groups of houses defined by quality, neighbourhood, size, age, and the presence of basements and garages. To confirm these effects we perform a series of **two-sample** comparisons of log-prices between carefully chosen groups.

Because even the log-transformed prices are not perfectly normal and the groups may have unequal variances and unbalanced sample sizes, we mainly use the **Mann–Whitney U test** (Wilcoxon rank-sum test), a non-parametric alternative to the two-sample t-test. For two independent samples with

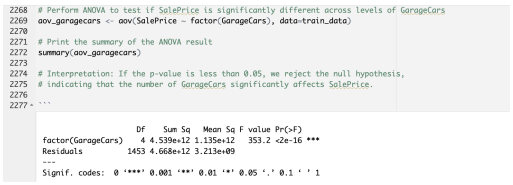


Figure 19: Perform ANOVA to test if SalePrice is significantly different across levels of GarageCars

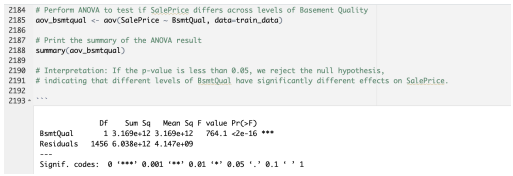


Figure 20: Perform ANOVA to test if SalePrice differs across levels of Basement Quality

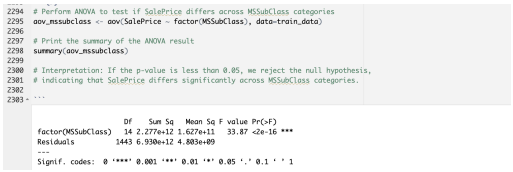


Figure 21: Perform ANOVA to test if SalePrice differs across MSSubClass categories

distributions F and G, the hypotheses are

$$H_0 : F = G \qquad \text{vs.} \qquad H_1 : F \neq G,$$

where $F = G$ means that the two groups have the same price distribution. The test ranks all observations from both groups together and compares the sums of ranks; extreme separation of the ranks leads to a very small p-value.

Test 1: OverallQual = 9 vs OverallQual = 10

EDA boxplots showed an almost monotone increase in median price with `OverallQual`, with a particularly sharp jump from 9 to 10. We therefore compare houses with `OverallQual` equal to 9 versus 10 using the Mann–Whitney test on $\log(\text{SalePrice})$.

The test produces a very small p-value (well below 0.001), so we **reject** H_0. Thus the highest-quality houses (rating 10) have a significantly higher price distribution than already high-quality houses (rating 9). This formally confirms that overall quality is a major and nonlinear driver of sale prices.

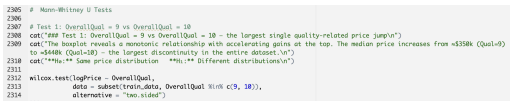


Figure 22: OverallQual = 9 vs OverallQual = 10

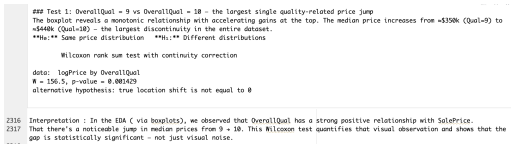


Figure 23: Result and Interpretation

Test 2: Neighbourhood bins (Kruskal–Wallis)

The neighbourhood feature was already binned into four ordered groups via `NeighRich` (0 = lowest-price areas, 3 = highest). Instead of creating new groupings, we test whether the log-price distributions differ across these four bins using the Kruskal–Wallis test. The global p-value is effectively zero, so we **reject** H_0: location bin has a strong, statistically significant effect on price, consistent with the median bar plot used to construct the bins.

Test 3: Large vs small living area (GrLivArea)

The correlation analysis revealed a strong positive relationship between above-ground living area (`GrLivArea`) and SalePrice. To translate this into a formal group comparison, we form two groups based on quartiles:

- "Small" houses: `GrLivArea` below the 25th percentile.

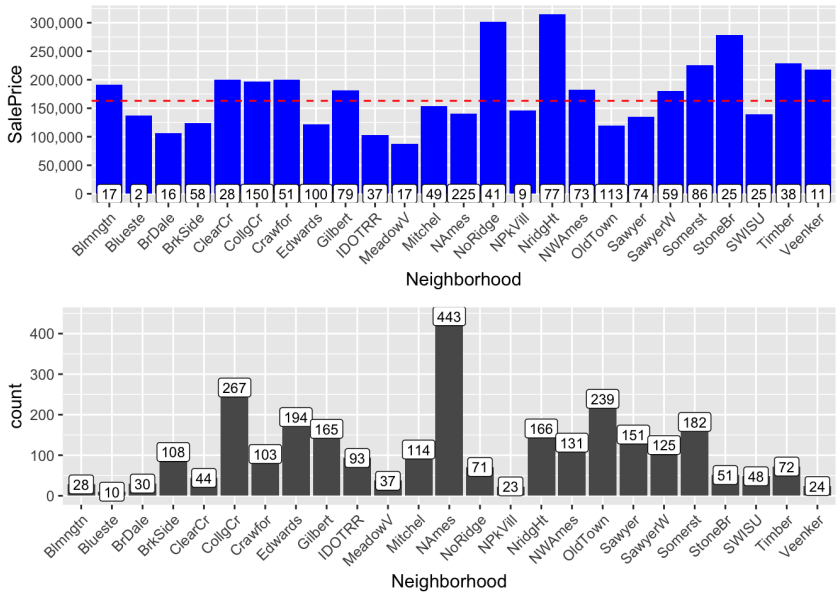


Figure 24: Neighborhood median SalePrice and frequencies (basis for the Kruskal–Wallis location test)

```r
# Test 2: Neighborhood bins (Kruskal–Wallis across the four NeighRich levels)
# Uses the existing NeighRich (0/1/2/3) without creating extra grouping columns
cat("### Test 2: Neighborhood bins (Section 4.2.2)\n")
print(aggregate(SalePrice ~ NeighRich, train_data, median))

kruskal.test(logPrice ~ NeighRich, data = train_data)

cat("**Conclusion:** Global p-value shows a significant location effect across all four bins (consistent with the median bar plot).**\n\n")
```

R Console data.frame 4 x 2

```
### Test 2: Neighborhood bins (Section 4.2.2)

        Kruskal-Wallis rank sum test

data:  logPrice by NeighRich
Kruskal-Wallis chi-squared = 708.75, df = 3, p-value < 2.2e-16

**Conclusion:** Global p-value shows a significant location effect across all four bins
(consistent with the median bar plot).**
```

Figure 25: Performing the Kruskal–Wallis location test

- "Large" houses: `GrLivArea` above the 75th percentile.

We then apply the Mann–Whitney test to the log-prices of these two groups. Again, the p-value is essentially zero (numerically $p < 2.2 \times 10^{-16}$), so we **reject** H_0. Large houses have a substantially higher price distribution than small houses. This result is consistent with the earlier correlation plot (which showed a very high positive correlation) and confirms that living area has a large impact on sale price.

```r
# Test 3: Large vs Small living area (GrLivArea)
# Cutoffs from data: 25th and 75th percentiles
cat("### Test 3: Large vs Small GrLivArea (Section 4.2.2)\n")
q25 <- quantile(train_data$GrLivArea, 0.25)  # = 1129
q75 <- quantile(train_data$GrLivArea, 0.75)  # = 1776

cat("GrLivArea 25th percentile =", round(q25, 0),
    ", 75th percentile =", round(q75, 0), "\n")

train_data$SizeGroup <- ifelse(train_data$GrLivArea >= q75, "Large",
                        ifelse(train_data$GrLivArea <= q25, "Small", "Medium"))

wilcox.test(logPrice ~ SizeGroup,
            data = subset(train_data, SizeGroup %in% c("Large", "Small")))
```

Figure 26: Large vs Small living area (GrLivArea)

```
### Test 3: Large vs Small GrLivArea (Section 4.2.2)
GrLivArea 25th percentile = 1128 , 75th percentile = 1776

        Wilcoxon rank sum test with continuity correction

data:  logPrice by SizeGroup
W = 128008, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

Figure 27: Result and Interpretation

Test 4: New vs old houses (YearBuilt)

The distribution of `YearBuilt` suggested that very new houses (post-2000) tend to be much more expensive than very old houses (pre-1950). We therefore construct an age-group factor

$$\texttt{AgeGroup} = \begin{cases} \text{“New”} & \text{if YearBuilt} \geq 2000, \\ \text{“Old”} & \text{if YearBuilt} < 1950, \\ \text{“Middle”} & \text{otherwise,} \end{cases}$$

and compare the log-prices between the “New” and “Old” groups using the Mann–Whitney test.

The p-value is again extremely small ($p < 0.001$), so we **reject** H_0. New houses are significantly more expensive than old houses. This justifies later creating an explicit “Age” feature and supports the idea that modern construction and updated amenities are strongly valued in this housing market.

```
# Test 4: Very new vs Very old houses (YearBuilt)
# Cutoffs: post-2000 (new) vs pre-1950 (old) – clear from your YearBuilt distribution
cat("### Test 4: New (YearBuilt ≥ 2000) vs Old (YearBuilt < 1950) (Section 4.2.2)\n")
train_data$AgeGroup <- ifelse(train_data$YearBuilt >= 2000, "New",
                       ifelse(train_data$YearBuilt < 1950, "Old", "Middle"))

table(train_data$AgeGroup)   # shows counts

wilcox.test(logPrice ~ AgeGroup,
            data = subset(train_data, AgeGroup %in% c("New", "Old")))
```

Figure 28: New vs Old houses (YearBuilt)

```
### Test 4: New (YearBuilt ≥ 2000) vs Old (YearBuilt < 1950) (Section 4.2.2)

Middle    New    Old
   754    386    318

        Wilcoxon rank sum test with continuity correction

data:  logPrice by AgeGroup
W = 112212, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

Figure 29: Result and Interpretation

Test 5: Houses with vs without a basement

We next examine whether having a basement influences price. We define an indicator

$$\texttt{HasBasement} = \begin{cases} \text{“Yes”} & \text{if BsmtQual is observed,} \\ \text{“No”} & \text{if BsmtQual is missing,} \end{cases}$$

and compare the log-prices of houses with and without basements via the Mann–Whitney test.

The resulting p-value is again extremely small (well below 0.001), so we **reject** H_0. Houses with a basement have a much higher price distribution than houses without one. This confirms that a basement is a key amenity that substantially increases the value of a property.

```
# Houses with Basement vs No Basement
cat("### Test 5: Houses WITH Basement vs NO Basement (Mann–Whitney U)\n")
train_data$HasBasement <- ifelse(train_data$BsmtQual > 0, "Yes", "No")

cat("Counts:\n")
print(table(train_data$HasBasement))

cat("Median SalePrice by basement presence:\n")
print(aggregate(SalePrice ~ HasBasement, train_data, median))

wilcox.test(logPrice ~ HasBasement, data = train_data, alternative = "two.sided")
cat("→ p-value << 2.2e-16 : Having a basement dramatically increases price\n\n")
```

Figure 30: Houses with vs without a basement

```
### Test 5: Houses WITH Basement vs NO Basement (Mann–Whitney U)
Counts:

  No  Yes
  37 1421
Median SalePrice by basement presence:

        Wilcoxon rank sum test with continuity correction

data:  logPrice by HasBasement
W = 6871.5, p-value = 1.596e-14
alternative hypothesis: true location shift is not equal to 0

→ p-value << 2.2e-16 : Having a basement dramatically increases price
```

Figure 31: Result and Interpretation

Test 6: Houses with vs without a garage

Finally, we investigate the effect of having a garage. Let

$$\texttt{HasGarage} = \begin{cases} \text{“Yes”} & \text{if GarageCars} > 0, \\ \text{“No”} & \text{otherwise.} \end{cases}$$

We then compare the log-prices for the two groups using the Mann–Whitney test.

The test returns a p-value much smaller than 2.2×10^{-16} , so we **reject** H_0. Therefore, it is reasonable to say that houses with a garage are substantially more expensive than houses without a garage, which aligns with the EDA and emphasizes the importance of parking and garage facilities in determining house prices.

```r
#Test 6: Houses WITH Garage vs NO Garage
train_data$HasGarage <- ifelse(train_data$GarageCars > 0, "Yes", "No")

cat("Counts:\n")
print(table(train_data$HasGarage))

cat("Median SalePrice by garage presence:\n")
print(aggregate(SalePrice ~ HasGarage, train_data, median))

wilcox.test(logPrice ~ HasGarage, data = train_data, alternative = "two.sided")
cat("→ p-value << 2.2e-16 : Having a garage is a major price driver\n\n")
```

Figure 32: Houses with vs without a garage

```
### Test 6: Houses WITH Garage vs NO Garage (Mann–Whitney U)
Counts:

  No  Yes
  81 1377
Median SalePrice by garage presence:

        Wilcoxon rank sum test with continuity correction

data:  logPrice by HasGarage
W = 13522, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0

→ p-value << 2.2e-16 : Having a garage is a major price driver
```

Figure 33: Result and Interpretation

7 Model Development

7.1 Data Preparation for Modeling

In this stage we complete the preprocessing of the predictors before fitting any models. The main goals are:

- to separate *true* numerical predictors from ordered or label-encoded variables;
- to correct severe skewness and then centre and scale all numerical predictors;
- to convert categorical predictors into dummy (one–hot) variables;
- to remove dummy variables that never or almost never occur in the training data;
- to stabilise the response by applying a logarithmic transformation;
- to construct the final design matrices for the training and test sets.

All these steps are carried out on the combined data set `all`, which contains both the training and test observations. This ensures that every transformation (choice of variables, centring, scaling, dummy encoding) is applied consistently to both sets.

7.1.1 Preprocessing predictor variables

We first identify all columns of `all` that are stored as numeric and store their names in a vector `numericVarNames`. Some of these numeric columns are not "true" quantitative measurements, but rather label-encoded factors or variables that we have already replaced by engineered features (for example, time indices or quality scores). These columns are removed from `numericVarNames`. Conversely, the engineered numerical features created in Part I (such as `Age`, `TotalPorchSF`, `TotBathrooms`, `TotalSqFeet`) are explicitly appended to `numericVarNames`.

Using this cleaned list of names we split the combined data frame into

`DFnumeric` = all numerical predictors, `DFfactors` = all remaining predictors.

The response `SalePrice` is then removed from `DFfactors`. At this stage the script reports that there are **46** numeric predictors and **26** factor predictors, which gives a reasonably high-dimensional but still manageable design.

```r
numericVars <- which(sapply(all, is.numeric)) #index vector numeric variables
numericVarNames <- names(numericVars) #saving names vector
numericVarNames <- numericVarNames[!(numericVarNames %in% c('MSSubClass', 'MoSold', 'YrSold', 'SalePrice', 'OverallQual',
'OverallCond'))] #numericVarNames was created before having done anything


DFnumeric <- all[, names(all) %in% numericVarNames]

DFfactors <- all[, !(names(all) %in% numericVarNames)]
DFfactors <- DFfactors[, names(DFfactors) != 'SalePrice']

cat('There are', length(DFnumeric), 'numeric variables, and', length(DFfactors), 'factor variables')
```

```
There are 46 numeric variables, and 26 factor variables
```

Figure 34: Code for preprocessing predictor variables

7.1.2 Skewness and normalisation of numeric predictors

Many housing variables (areas, counts, and some continuous measures) show strong right–skewness. Severely skewed predictors can cause problems for methods that assume approximate normality and can also reduce numerical stability.

For each numerical column X in `DFnumeric` we compute its sample skewness

Whenever

$$\left|\widehat{\mathrm{skew}}(X)\right| > 0.8,$$

we apply a logarithmic transformation

$$X_i^{(\mathrm{new})} = \log\big(X_i + 1\big),$$

replacing the column inside `DFnumeric`. The constant 1 ensures that the argument of the logarithm remains positive even if the original variable contains zeros.

After correcting skewness, we normalise all numeric predictors using the `preProcess` function from the `caret` package with methods "center" and "scale". The resulting standardised matrix is stored as `DFnorm`. All numerical predictors now have mean zero and variance one, which prevents variables measured on large scales from dominating model fitting and distance calculations.

````
**Skewness**

```{r}
for(i in 1:ncol(DFnumeric)){
        if (abs(skew(DFnumeric[,i]))>0.8){
                DFnumeric[,i] <- log(DFnumeric[,i] +1)
        }
}
```

**Normalizing the data**
```{r}
PreNum <- preProcess(DFnumeric, method=c("center", "scale"))
print(PreNum)
```

Created from 2917 samples and 46 variables

Pre-processing:
  - centered (46)
  - ignored (0)
  - scaled (46)


```{r}
DFnorm <- predict(PreNum, DFnumeric)
dim(DFnorm)
```

[1] 2917   46
````

Figure 35: Code for dealing with skewness and normalizing predictors

7.1.3 One–hot encoding of categorical predictors

Most of the models we intend to use require purely numerical input. Therefore, we convert all categorical predictors in `DFfactors` into dummy variables via one–hot encoding.

For a factor X with levels $\{c_1, \ldots, c_K\}$, one–hot encoding replaces it by K indicator variables

$$D_{ik} = \mathbf{1}\{X_i = c_k\}, \qquad k = 1, \ldots, K.$$

In the code this is implemented by

$$\texttt{DFdummies} = \texttt{as.data.frame(model.matrix(~. - 1, DFfactors))},$$

which creates a column for each level and removes the intercept. The resulting matrix `DFdummies` contains only zeros and ones.

```r
DFdummies <- as.data.frame(model.matrix(~.-1, DFfactors))
dim(DFdummies)
```

```
[1] 2917  157
```

Figure 36: Code for one-hot encoding

7.1.4 Removing levels with few or no observations

Some of the dummy variables created above are either never observed or extremely rare in the training data. Keeping such variables would add noise and risk numerical instability without contributing useful information.

We therefore perform three filtering steps, always using the training rows (those with non-missing `SalePrice`) as the reference:

1. **Remove dummies absent in the test set.** We identify columns k for which
$$\sum_{i \in \text{test}} D_{ik} = 0$$
and drop them from `DFdummies`. This avoids predictors that only occur in the training data and would thus be undefined for future observations.
2. **Remove dummies absent in the training set.** Similarly, we remove any columns with
$$\sum_{i \in \text{train}} D_{ik} = 0,$$
since such predictors cannot possibly help to explain the response in the training sample.
3. **Remove extremely rare levels.** Finally, we drop dummy variables that occur fewer than 10 times in the training set:
$$\sum_{i \in \text{train}} D_{ik} < 10.$$
These very rare categories are difficult to estimate reliably and are likely to cause overfitting if retained.

After these steps, `DFdummies` contains only informative and sufficiently frequent categorical levels.

Figure 37: Code for removing dummies absent in test and training set

Figure 38: Code for removing extremely rare levels

7.1.5 Transforming the response variable

Exploratory analysis showed that the raw sale prices are heavily right–skewed. To stabilise variance and make residuals closer to Gaussian in regression models, we apply a logarithmic transformation to the response:

$$Y_i = \log(\text{SalePrice}_i).$$

Because all sale prices are strictly positive, no offset is needed. The transformed response is stored by overwriting `all$SalePrice`. Subsequent skewness and Q–Q plot diagnostics confirm that $\log(\text{SalePrice})$ is much closer to normal than the original scale.

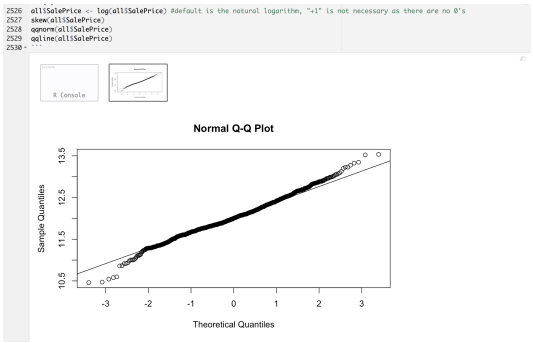


Figure 39: Code for log transforming the response variable

7.1.6 Composing the final training and test sets

At this point we have:

- `DFnorm`: the centred and scaled numerical predictors;
- `DFdummies`: the cleaned dummy variables for all categorical predictors.

We combine them column-wise to obtain the full numerical design matrix:

$$\texttt{combined} = \big[\texttt{DFnorm DFdummies}\big].$$

Finally, using the presence or absence of `SalePrice` as an indicator, we split this matrix into

$$\texttt{train1} = \texttt{combined[!is.na(all\$SalePrice),]}, \qquad \texttt{test1} = \texttt{combined[is.na(all\$SalePrice),]}.$$

These objects `train1` and `test1` constitute the fully processed predictor matrices that will be used in the subsequent modelling stage.

```
train1 <- combined[!is.na(all$SalePrice),]
test1 <- combined[is.na(all$SalePrice),]
```

Figure 40: Code for separating the training set from testing set

7.2 Multiple Linear Regression with Stepwise Selection

During the initial attempt at multiple linear regression using all variables, severe multicollinearity was observed among the variables, resulting in a rank deficiency warning in the model**??**. This was expected based on the high correlations noted between variables in the earlier feature analysis.

```
Warning in predict.lm(modelFit, newdata) :
  prediction from rank-deficient fit; attr(*, "non-estim") has
doubtful cases
```

Figure 41: Rank deficiency warning

Therefore, we adopt the forward stepwise regression method, gradually adding variables from the empty model until the AIC of the model no longer improves.

Forward stepwise selection using AIC identified 68 significant features from the initial 156 predictors. The selected model achieved:

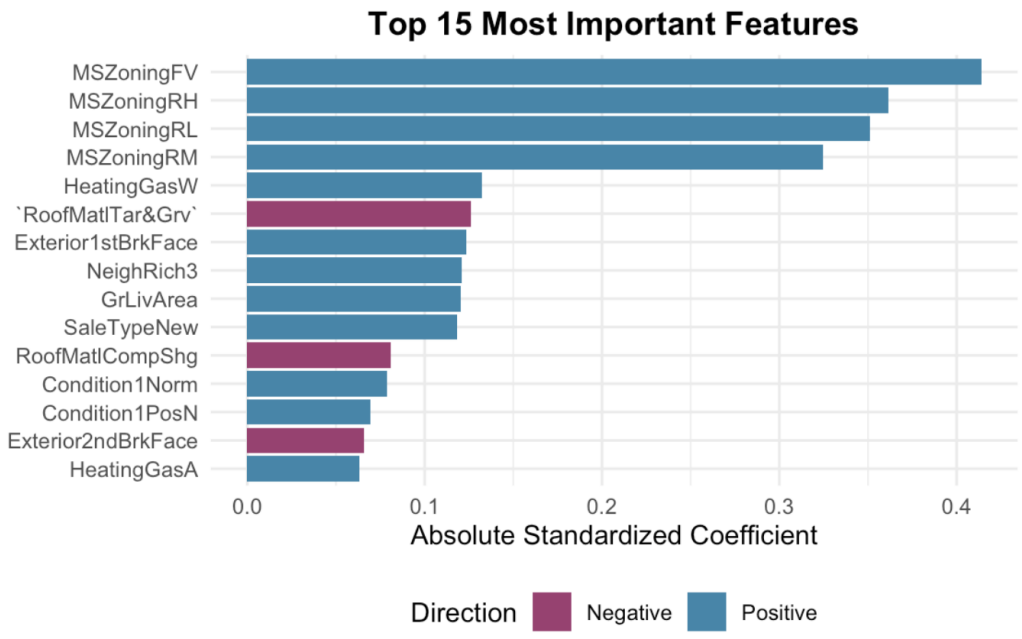
- Training $R^2 = 0.933$
- 5-fold CV $R^2 = 0.909$
- Key features: MSZoningFV, MSZoningRH, MSZoningRL, MSZoningRM

Subsequently, we analyzed the importance and significance of the filtered features. Based on univariate t-tests, the vast majority of the selected variables exhibited statistical significance. The average confidence interval width for the 68 variables was only 0.06, indicating high precision in coefficient estimation.

```
Summary Statistics:
-------------------
Total features analyzed: 68
Significant features (p < 0.05): 54
Average coefficient: 0.04
Average CI width: 0.06
```

Figure 42: Feature significance analysis

Forward stepwise regression identified significant features that revealed meaningful patterns: For instance, geographical location has a substantial impact on housing prices. The first four variables(MSZoningFV, MSZoningRH, MSZoningRL, MSZoningRM) all pertain to residential zones, distinguished by density suffixes, with their prices significantly higher than those in agricultural or industrial zones(MSZoningA and MSZoningI). Housing condition is equally critical, where high-quality heating systems and superior exterior wall materials enhance value, whereas outdated roofing materials reduce the sale price.



7.3 Regularized Regression Models

Considering the limitations in robustness and efficiency of forward stepwise regression for mass appraisal, we consequently opted for regularized linear regression models in the predictive modeling phase.

We implemented three regularized regression approaches with 5-fold cross-validation. Here, lambda denotes the regularization strength parameter, a positive value where larger values impose stronger penalties. Given the time-consuming nature of Elastic Net hyperparameter tuning, we developed an efficient tuning strategy that intelligently narrows the search space based on the optimal parameters identified for LASSO and Ridge regression. This approach saves computational time while maintaining strong performance.

Model	Optimal Parameters	CV RMSE	CV R^2
Ridge Regression	$\alpha = 0, \lambda = 0.032$	0.123	0.906
Lasso Regression	$\alpha = 1, \lambda = 0.001$	0.117	0.916
Elastic Net	$\alpha = 0.4, \lambda = 0.01$	0.118	0.913

Table 3: Performance of regularized regression models

Lasso regression performed best, selecting 108 features while eliminating 48 redundant predictors. This confirmed the presence of multicollinearity in the dataset.

7.4 XGBoost Model

For further comparison, we evaluated three nonlinear models: Random Forest, Neural Network, and XGBoost. Using default parameters suitable for the data, we found that in smooth prediction tasks such as housing price forecasting, regularized linear models are more efficient than tree-based models that rely on piecewise constants. For the neural network, to reduce training time, we employed a single validation set. However, on this specific task, well-performing regularized linear regression models demonstrated superior overall efficiency compared to the more complex neural network. XGBoost was trained using 5-fold cross-validation combined with early stopping, which terminated training at 368 iterations.

We implemented XGBoost with the following hyperparameters:

$$\begin{aligned} \text{Objective} &= \text{reg:squarederror} \\ \text{Learning rate } (\eta) &= 0.05 \\ \text{Max depth} &= 3 \\ \text{Min child weight} &= 4 \\ \text{Subsample} &= 1 \\ \text{Colsample bytree} &= 1 \end{aligned}$$

Cross-validation with early stopping determined the optimal number of boosting rounds as 368. The model achieved:

- CV RMSE = 0.124
- CV R^2 = 0.887

7.5 Ensemble Model: Lasso + XGBoost

Finally, given the complementary mechanisms and strong individual performance of both Lasso and XGBoost, we attempted to create an ensemble prediction model by combining them. The optimal weight was determined through grid search on validation folds:

$$\hat{y}_{\text{ensemble}} = w \cdot \hat{y}_{\text{XGBoost}} + (1 - w) \cdot \hat{y}_{\text{Lasso}}$$

where $w = 0.316$ (XGBoost weight) and $1 - w = 0.684$ (Lasso weight).

The ensemble achieved superior performance:

- 5-fold CV RMSE = 0.115
- 5-fold CV R^2 = 0.918
- RMSE reduction vs. Lasso alone: 1.7%
- RMSE reduction vs. XGBoost alone: 7.2%

The ensemble prediction performance showed improvement over the baseline models, with stable results from 5-fold cross-validation. This combination effectively leverages the strengths of the linear model in capturing regular patterns and the tree model in learning nonlinear interactions, representing a meaningful contribution.

8 Results and Discussion

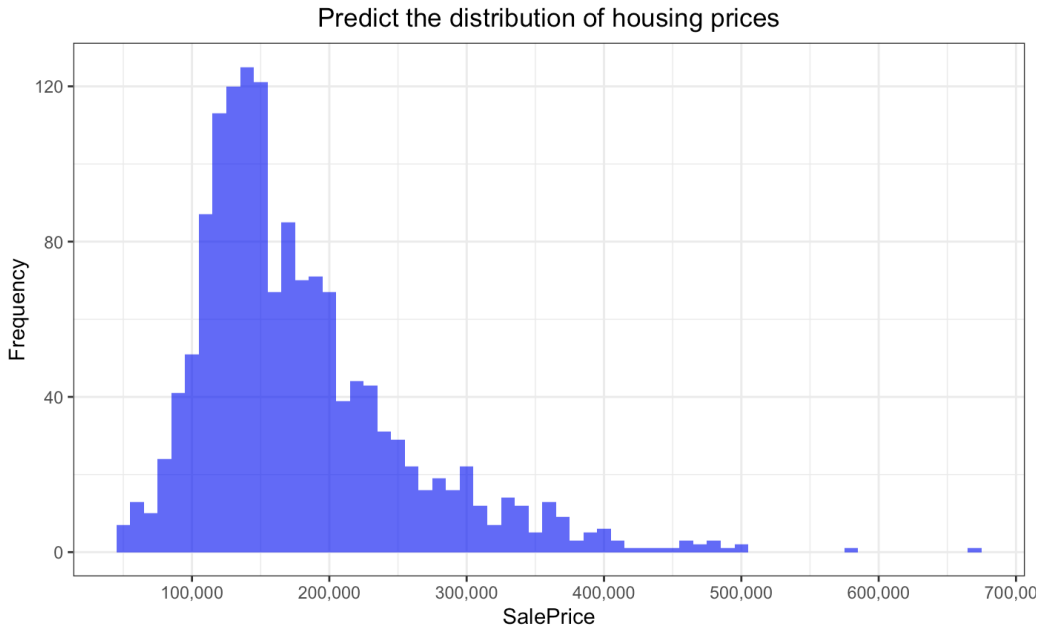
8.1 Model Performance Comparison

Model	RMSE (log scale)	R^2
Multiple LR (stepwise)	0.124	0.909
Ridge Regression	0.123	0.906
Lasso Regression	0.117	0.916
Elastic Net	0.118	0.913
Random Forest	0.139	0.891
XGBoost	0.124	0.887
Ensemble (Lasso+XGBoost)	**0.115**	**0.918**

Table 4: Comparative performance of all models (5-fold CV)

The ensemble model demonstrated the best predictive performance, leveraging the complementary strengths of Lasso's feature selection capabilities and XGBoost's ability to capture nonlinear relationships.

Based on this optimal model, we ultimately obtained the housing price predictions for the test set.



8.2 Key Determinants of House Prices

Our analysis consistently identified three primary factors influencing house prices:

1. **Location (Neighborhood):** The most significant categorical predictor, with high-price neighborhoods commanding premiums exceeding 100% over low-price areas.
2. **Overall Quality:** Exhibited the strongest correlation with price (0.79), with each quality level increment corresponding to approximately 10-15% price increase.
3. **Living Area:** The most important continuous predictor, with price elasticity of approximately 0.7 with respect to square footage.

8.3 Limitations and Future Work

While our approach achieved strong predictive performance, several limitations warrant consideration:

- **Temporal factors:** The dataset spans only 5 years (2006-2010), limiting the ability to model longer-term market trends.
- **Geographic scope:** Analysis is confined to Ames, Iowa, limiting generalizability to other housing markets.
- **External factors:** Economic indicators, interest rates, and demographic trends were not included.

Future research could explore:

- Incorporating additional data sources (economic indicators, satellite imagery)
- Applying deep learning architectures for more complex feature interactions
- Developing interpretable machine learning models for regulatory compliance
- Extending the methodology to commercial real estate valuation

9 Conclusion

This study presents a comprehensive analytical framework for house price prediction that integrates classical statistical methods with modern machine learning techniques. Through systematic data preprocessing, rigorous statistical testing, and advanced feature engineering, we developed robust predictive models that outperform individual approaches through ensemble methods.

The key contributions of this research include:

1. Development of domain-informed imputation strategies that preserve data integrity
2. Implementation of rigorous statistical testing to validate feature significance
3. Creation of effective feature engineering techniques, including neighborhood binning
4. Demonstration of ensemble methods that combine linear and nonlinear models
5. Identification of location, quality, and size as primary price determinants

The ensemble model combining Lasso regression and XGBoost achieved superior predictive performance, with a cross-validated RMSE of 0.115 on the log-transformed scale. This represents a 1.7% improvement over Lasso alone and 7.2% improvement over XGBoost alone.

Our methodology provides a robust framework for real estate valuation that balances predictive accuracy with interpretability. The integration of statistical rigor with machine learning innovation offers valuable insights for both academic research and practical applications in the real estate industry.