Predictive Modeling of House Prices in Ames

Group Info(Members)

1. Lakshmi Sai Akash Kandru(LK23E)
2. Sai Nikhil Aratipamula(SA23L)
3. Sameera Rompicherla(SR23BA)
4. Nandhakumar Ranganathan Ganesh(NR23G)

Introduction: The Ames Housing Dataset is a comprehensive collection of housing-related attributes designed to facilitate the analysis of residential properties in Ames, Iowa. This dataset offers valuable insights into various features of homes, ranging from physical characteristics to qualitative assessments of quality and condition. The observations in the dataset provide a rich source of information for understanding the real estate landscape in the Ames area.

Description:

The dataset comprises a diverse set of attributes, each providing distinct insights into the properties under consideration. These observations encompass details such as lot size, construction quality, overall condition, construction and remodel dates, basement characteristics, living area dimensions, bathroom counts, bedroom counts, kitchen counts, room totals, fireplace presence, garage specifications, and various porch and deck areas. Additionally, the dataset includes the crucial variable of the property's sale price, measured in dollars, serving as a key indicator for the economic value of the homes.

Data Dictionary

The full data set contains 1,359 homes and the following 27 variables:

- `LotArea`: Lot size in square feet.
- `OverallQual`: Overall quality of the house's material and finish. The scale ranges from 1 (Very Poor) to 9 (Very Excellent).
- `OverallCond`: Overall condition rating. The scale ranges from 1 (Very Poor) to 9 (Very Excellent).
- `YearBuilt`: Original construction date.
- `YearRemodAdd`: Remodel date.
- `BsmtFinSf1`: Type 1 finished square feet.
- `BsmtFinSf2`: Type 2 finished square feet.
- `TotalBsmtSf`: Total square feet of basement area.
- `FirstFlrSf`: First floor square feet.
- `SecondFlrSf`: Second floor square feet.
- `GrLivArea`: Above grade (ground) living area square feet.
- `BsmtFullBath`: Number of full bathrooms in the basement.
- `BsmtHalfBath`: Number of half baths in the basement.
- `FullBath`: Number of full bathrooms above ground.

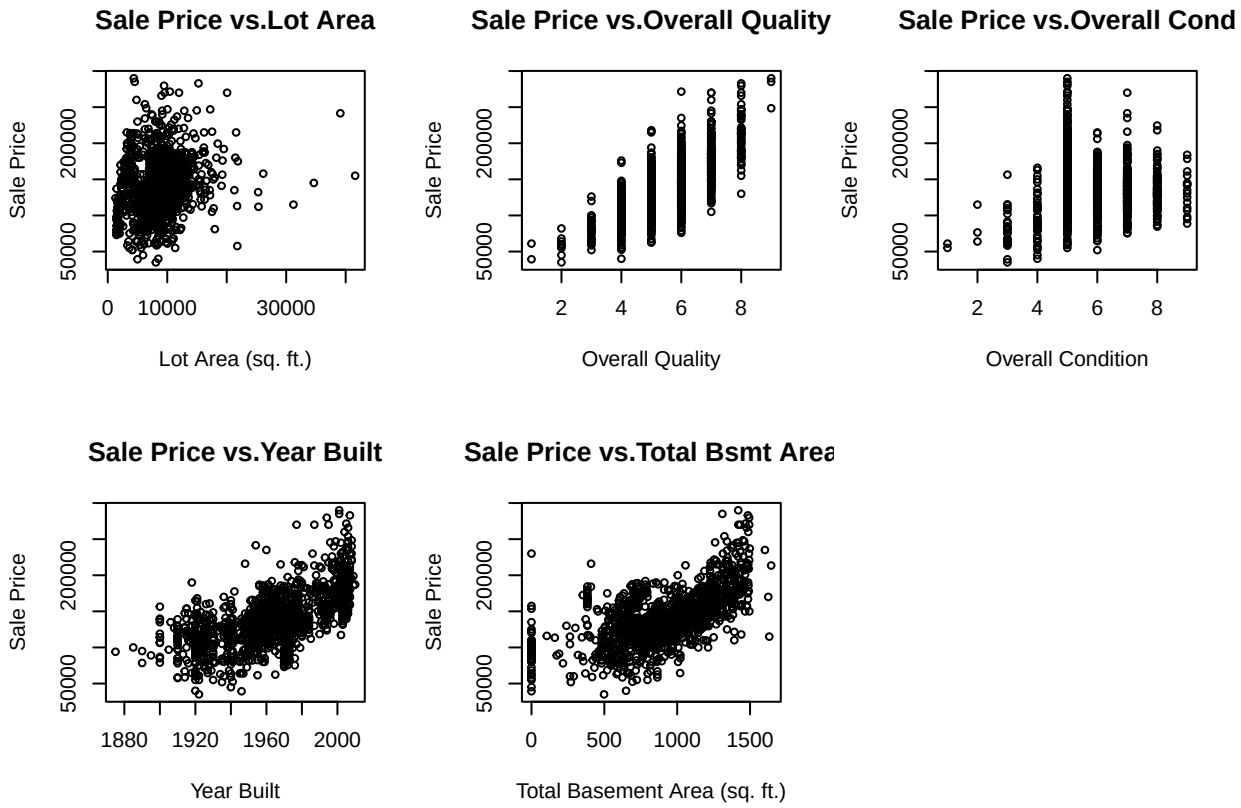
- `HalfBath`: Number of half baths above ground.
- `BedroomAbvGr`: Number of Bedrooms above ground.
- `KitchenAbvGr`: Number of Kitchens above ground.T
- `TotRmsAbvGrd`: Total rooms above ground (does not include bathrooms).
- `Fireplaces`: Number of fireplaces.
- `GarageCars`: Size of garage in car capacity.
- `GarageArea`: Size of garage in square feet.
- `WoodDeckSf`: Wood deck area in square feet.
- `OpenPorchSf`: Open porch area in square feet.
- `EnclosedPorch`: Enclosed porch area in square feet.
- `ThreeSsnProch`: Three season porch area in square feet.
- `ScreenPorch`: Screen porch area in square feet.
- `SalePrice`: The property's sale price in dollars.

• The dataset is loaded and summarized to provide an initial understanding of its structure and contents.

Exploratory Data Analysis (EDA)

• To gain insights into the data, exploratory data analysis (EDA) is performed.Histogram is drawn and Initial scatter plots are generated to visualize the relationships between the target variable, Sale Price, and some key features

• The histogram has a long tail to the right, meaning that there are a few homes with very high sale prices.



• These plots focus on key variables like Lot Area, Overall Quality, Overall Condition, Year Built, Total Basement Area.These tend to show positive correlation with SalePrice.

• Before proceeding with the analysis,it's crucial to understand the presence of missing values in the dataset. No missing values were identified in the dataset

Regression Analysis

Model Building

• To understand the relationship between various features and saleprices, a linear regression model was fitted.

• The model, denoted as full_OLS_model, regresses Sale Price against all available predictors in the dataset.

• The full_OLS_model demonstrates strong predictive power, explaining approximately 86.68% of the variability in Sale Price. The model is statistically significant (F-statistic: 333.3, p-value: $<$ 2.2e-16), reinforcing its ability to reliably predict housing prices based on the available features.

Model Evaluation

• To assess the predictive performance of the full_OLS_model, Leave-One-Out Cross-Validation (LOOCV) Root Mean Squared Error (RMSE) was calculated,with all predictors which given Loocv-Rmse score 13791.69 and the Multiple R square 0.8668.

Variable Selection

• To identify the optimal model for predicting housing prices, we employed several model selection techniques, each aiming to strike a balance between model complexity and predictive performance.

Model Selection

• Information Criteria-based Selection Utilizing the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), both forward and backward model selection approaches were explored. These

criteria consider the trade-off between model fit and complexity. The table below summarizes the outcomes, presenting root mean squared error for leave-one-out cross-validation (LOOCV RMSE) as the key performance metric.

• After thorough exploration of various model selection techniques, the forward AIC method led us to the following optimal model:

Optimal Model Details

• SalePrice-LotArea+OverallQual+OverallCond+YearBuilt+YearRemodAdd+BsmtFinSf1+BsmtFinSf2+TotalBsmtSf+FirstFlrSf+GrLivArea+FullBath+BedroomAbvGr+KitchenAbvGr+TotRmsAbvGrd+Fireplaces+GarageCars+GarageArea+WoodDeckSf+OpenPorchSf+ScreenPorch

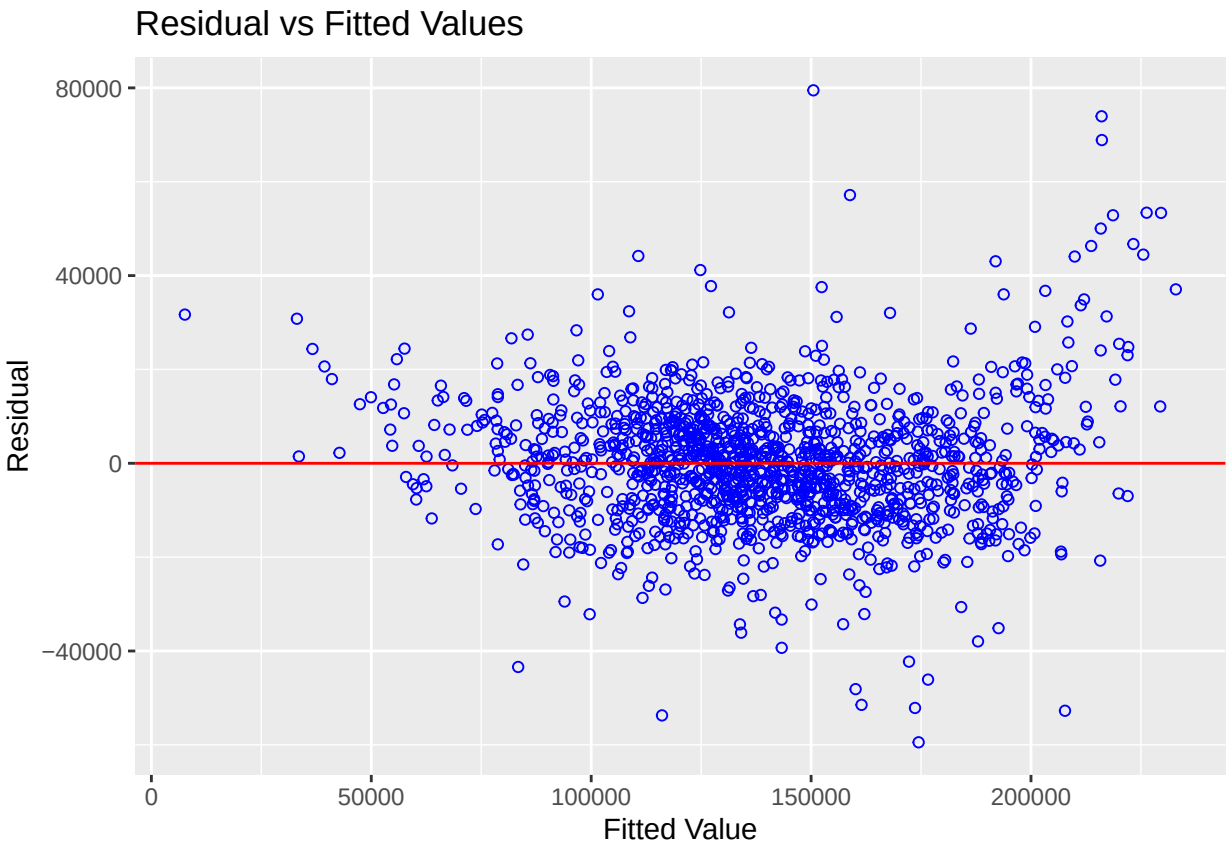
This model, selected based on its minimized AIC value

Performance Metrics

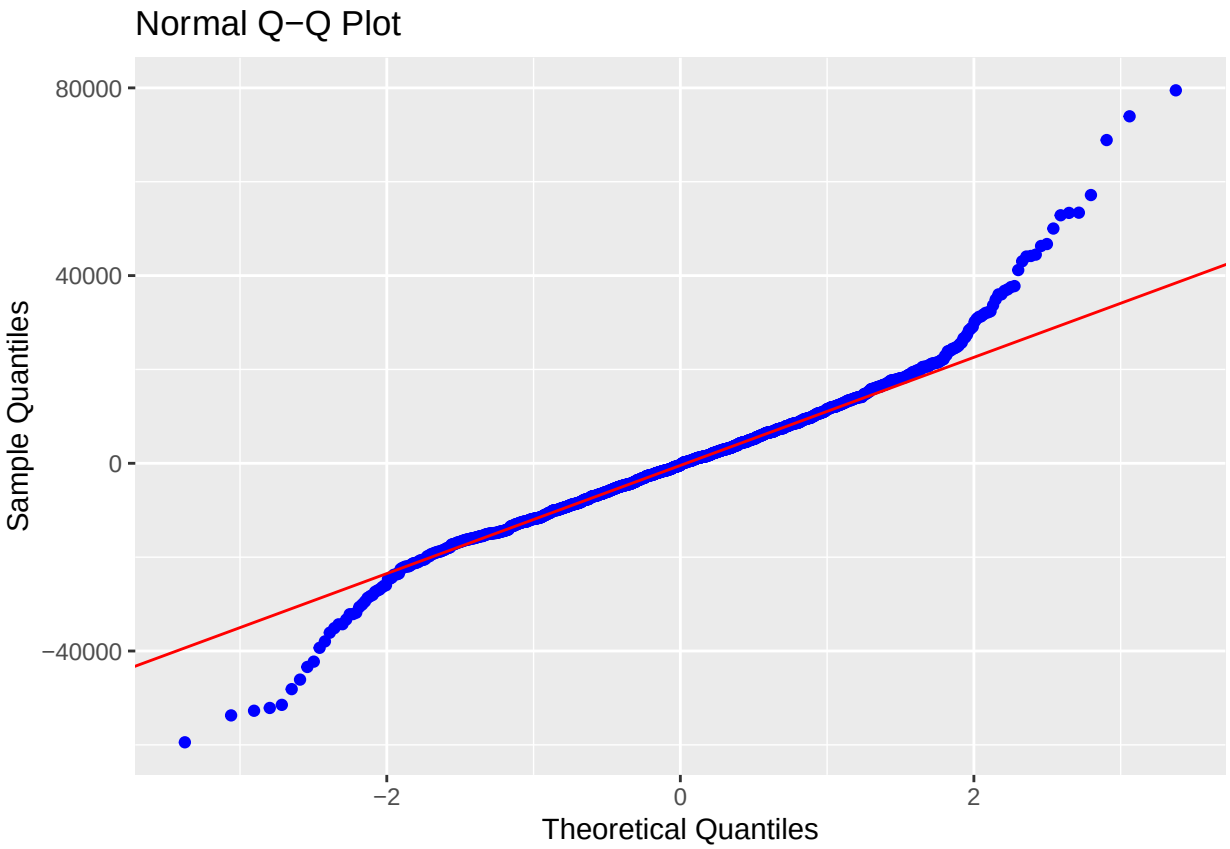
Criterion and Search method	Scores (RMSE_LOOCV)
Forward AIC model	13758.33
Forward BIC model	13793.35
Backward AIC model	13758.33
Backward BIC model	13793.35
Stepwise AIC model	13758.33
Stepwise BIC model	13793.35
Exhaustive AIC model	13758.33
Exhaustive BIC model	13793.35
Adjusted R2 model	13766.84

Consistency Across Methods:

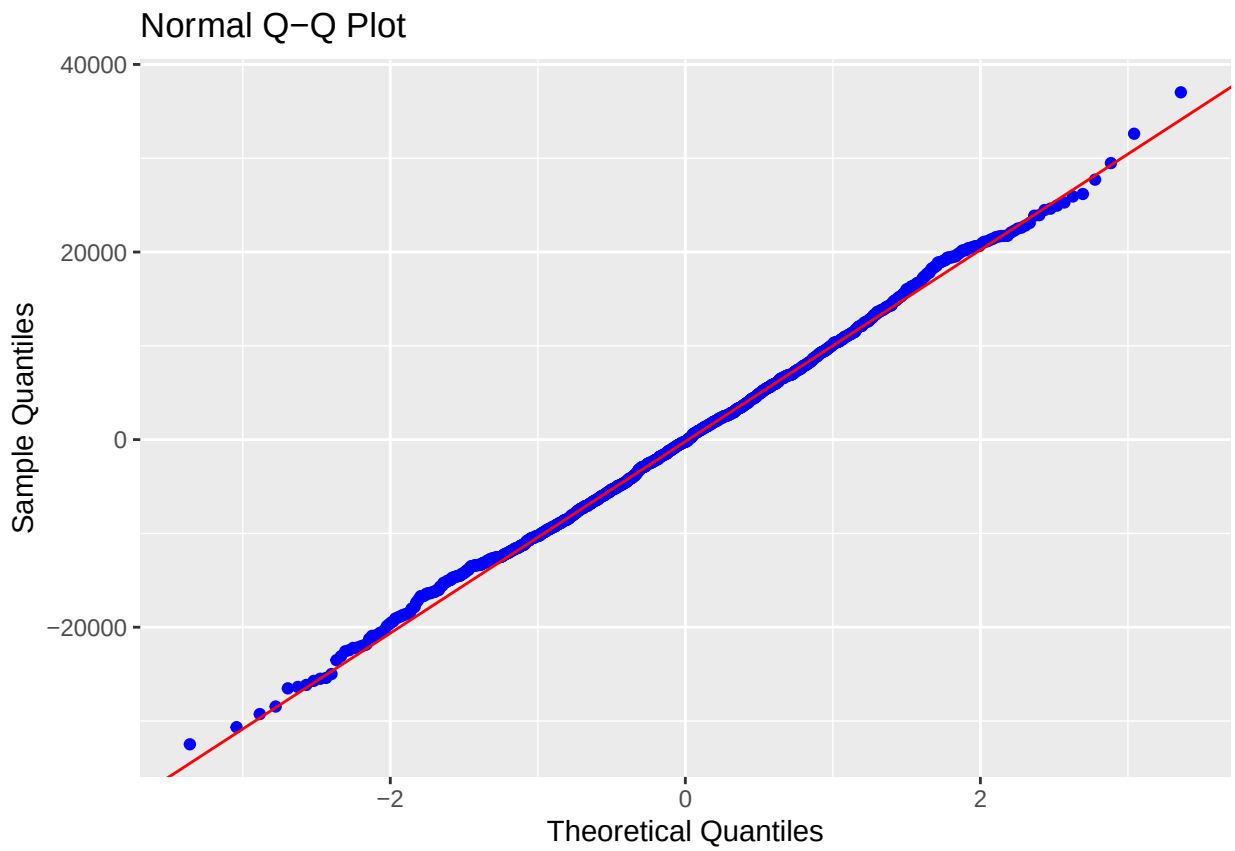
• Remarkably, the forward, backward, and stepwise AIC methods resulted in the same optimal model with a minimized AIC value of 13758.33 for RMSE_LOOCV. This consistency across different selection methods reinforces the robustness of the chosen model.



• The studentized Breusch-Pagan test was conducted to assess the presence of heteroscedasticity in the model residuals.The Breusch-Pagan test yields a p-value below 0.05, signaling a violation of the constant variance assumption. This is reinforced by the Fitted vs. Residual plot, where we observe an increasing variance as we move along the fitted values.



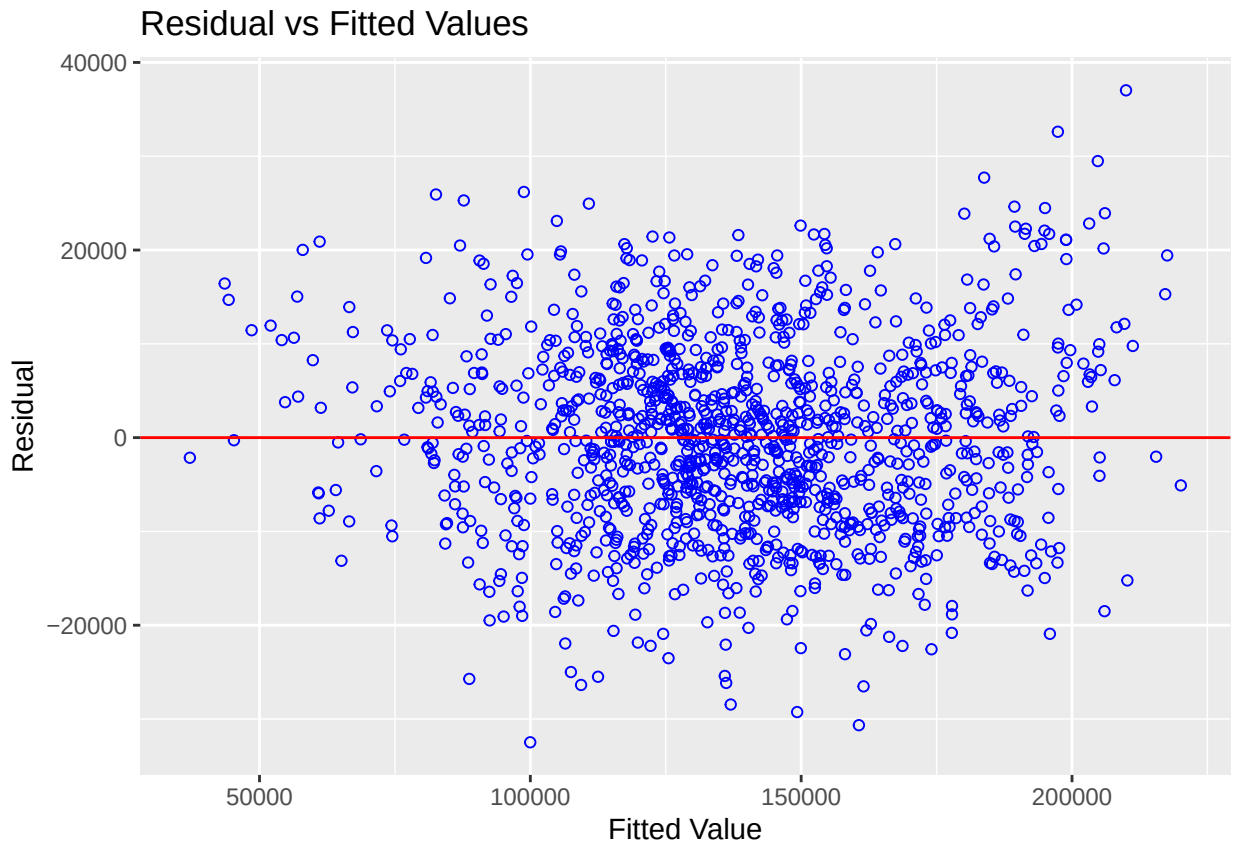
• The Shapiro-Wilk test applied to the residuals of model_selected reveals a departure from normality ($p < 0.05$). This departure is visually evident in the Quantile-Quantile (QQ) plot.



Handling Violations

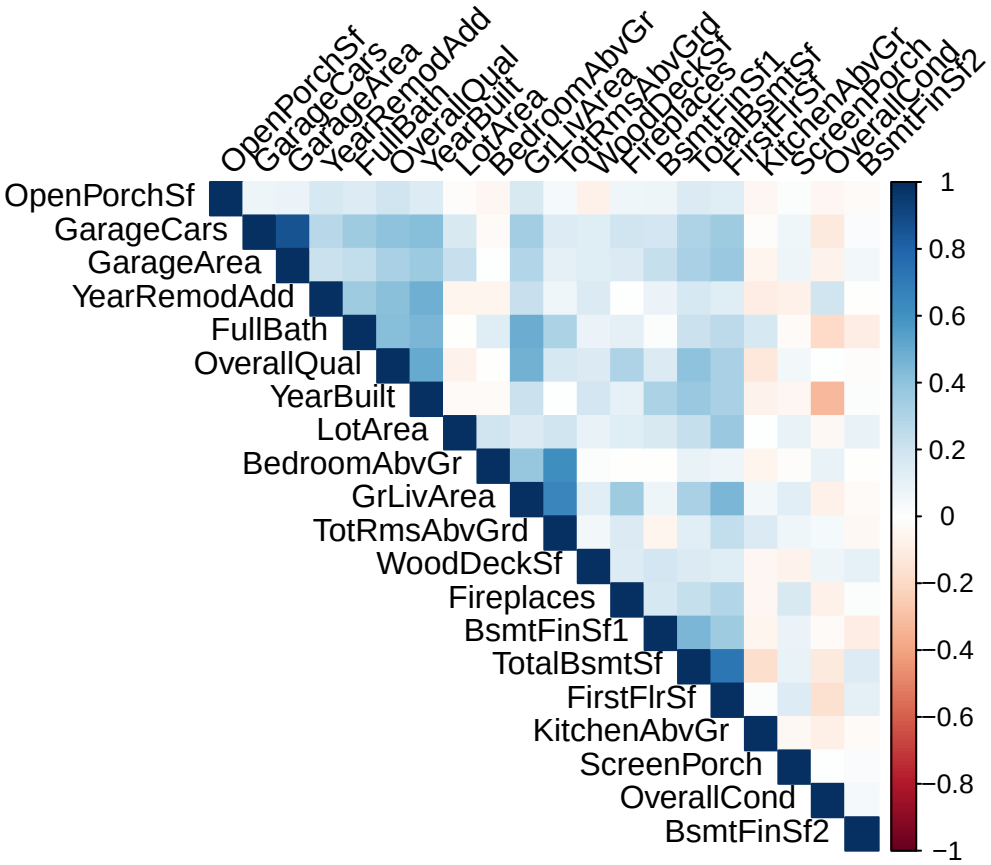
QQ Plot

- The QQ plot visually assesses the normality of residuals in the model.
- In the plot,points closely follow a straight line, it indicates that the residuals are approximately normally distributed.



- The plot shows a random scattering of points around the horizontal axis.

- Before removing influential points, we observed a normality issue in our data. We addressed this by eliminating highly influential points. Subsequently, we confirmed the improvement through the Shapiro-Wilk test, where the p-value exceeded 0.05, indicating that normality is no longer a concern.However, the problem of constant variance still persists.

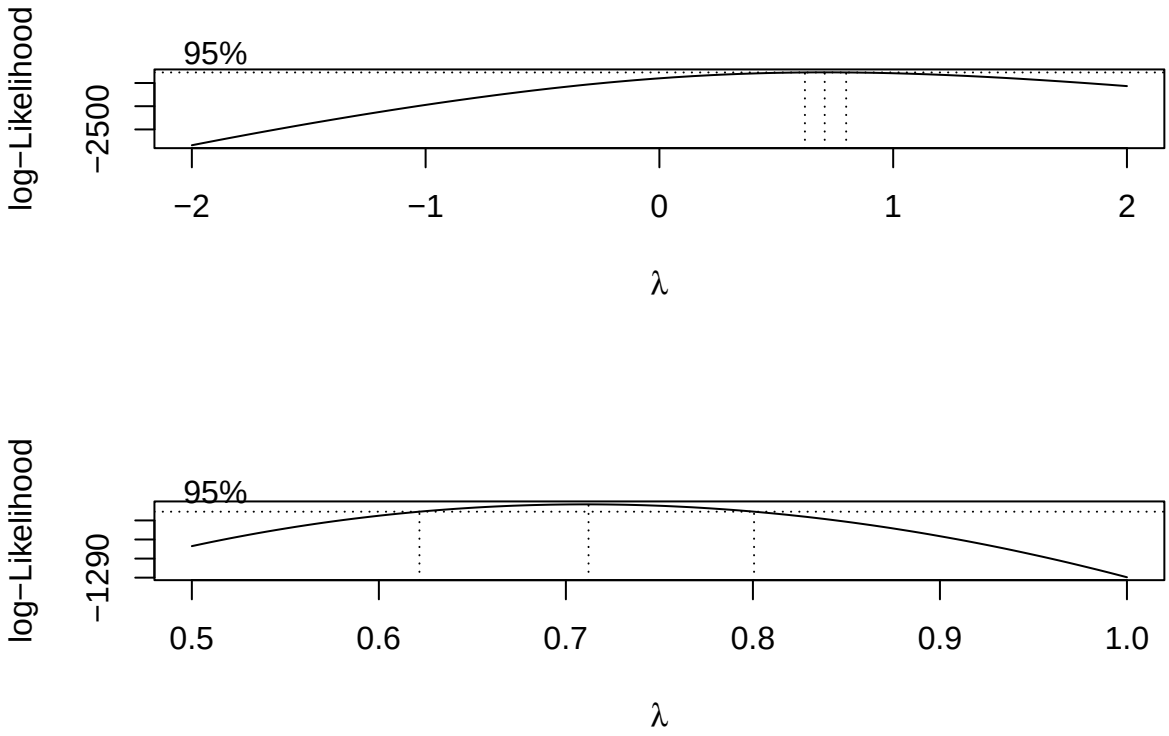


• Correlation between these two pairs: TotalBsmtSf and FirstFlrSf: 0.72172935 GarageCars and GarageArea: 0.86352489

• so we choose to drop one from each pair and then we refitted the model using the remaining set of non-correlated predictor but still assumption of constant variance remains unmet

• Specifically, "GarageCars" and "FirstFlrSf" were chosen for removal.

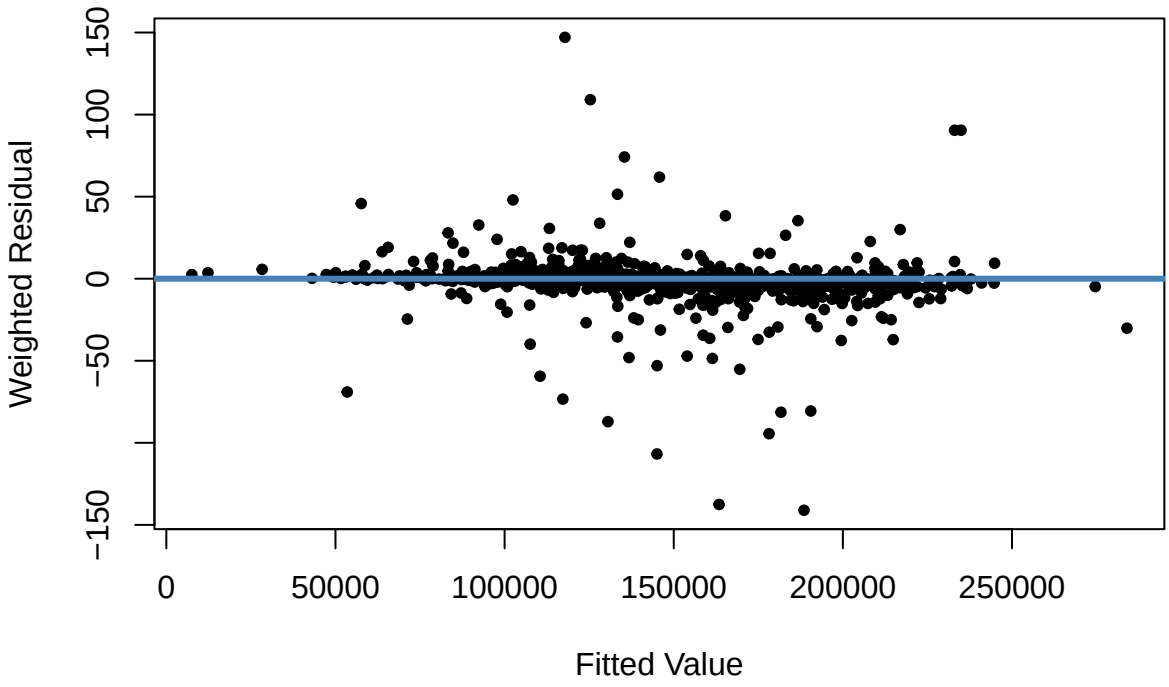
• Additionally, we calculated the Variance Inflation Factor (VIF).The results indicated that all VIF values were below 5, suggesting that there is no significant issue of multicollinearity among these predictors.



• As the Box-Cox transformation suggested a lambda value of 0.75, we proceeded with a square root transformation to address the non-constant variance in the model.

Following the square root transformation a Breusch-Pagan test was conducted. The obtained p-value was less than 0.05,Subsequently, in response to the detected heteroscedasticity, we opted for the Weighted Least Squares (WLS) model.

• Upon implementing the WLS model and conducting the Breusch-Pagan test again, the p-value is now observed to be 1, suggesting that the assumption of homoscedasticity is met.



The plot of residuals against fitted values is a diagnostic plot commonly used to assess the performance of regression models.WLS model has addressed the issue of heteroscedasticity.

Discussion

• In our exploration of regression models, we carefully crafted and assessed several variations with distinct predictor subsets. These models were tailored to address issues such as multicollinearity, interpretability, and predictive accuracy. The ensuring table encapsulates crucial metrics for each model, shedding light on their explanatory strength and predictive capabilities.

Model	R2	RMSE
OLS with all Predictors	0.8668	13791.69
OLS with variable selected model	0.8663	13758.33
OLS with non correlated predictors	0.904	10450.81
Weighted Least Squares	0.991	19765.93

Conclusion

• "In summary, opting for the Weighted Least Squares (WLS) model reflects a thoughtful decision, addressing the heteroscedasticity present in the residuals. This choice enhances the reliability of our regression analysis, leading to more accurate parameter estimates and a deeper understanding of the factors impacting house prices in Ames."

Code Appendix

```
head(data)
library(dplyr)
library(corrplot)
data <- read.csv("C:/Users/sam/Desktop/ames_housing (1).csv")
# Display summary statistics of the dataset
summary(data)
data <- read.csv("C:/Users/sam/Desktop/ames_housing (1).csv")
hist(data$SalePrice,
     main = "Histogram of Sale Prices",
     xlab = "Sale Price",
     ylab = "Frequency",
     col = "skyblue",       # Set color
     border = "black"       # Set border color
)
# Set the plot size
par(mfrow = c(2, 3))  # 2 rows, 3 columns

# Scatter plot for SalePrice vs.LotArea
plot(data$LotArea, data$SalePrice,
     xlab = "Lot Area (sq. ft.)",
     ylab = "Sale Price",
     main = "Sale Price vs.Lot Area",
     cex = 0.7)  # Adjust the size of points

# Scatter plot for SalePrice vs.OverallQual
plot(data$OverallQual, data$SalePrice,
     xlab = "Overall Quality",
     ylab = "Sale Price",
     main = "Sale Price vs.Overall Quality",
     cex = 0.7)  # Adjust the size of points

# Scatter plot for SalePrice vs.OverallCond
plot(data$OverallCond, data$SalePrice,
     xlab = "Overall Condition",
     ylab = "Sale Price",
     main = "Sale Price vs.Overall Cond",
     cex = 0.7)  # Adjust the size of points

# Scatter plot for SalePrice vs. YearBuilt
plot(data$YearBuilt, data$SalePrice,
     xlab = "Year Built",
     ylab = "Sale Price",
     main = "Sale Price vs.Year Built",
     cex = 0.7)  # Adjust the size of points

# Scatter plot for SalePrice vs. TotalBsmtSf
plot(data$TotalBsmtSf, data$SalePrice,
     xlab = "Total Basement Area (sq. ft.)",
     ylab = "Sale Price",
     main = "Sale Price vs.Total Bsmt Area",
     cex = 0.7)  # Adjust the size of points
```

```r
# Reset the plot size to the default
par(mfrow = c(1, 1))
colSums(is.na(data))
full_OLS_model=lm(SalePrice~ .,data=data)
summary(full_OLS_model)
calc_loocv_rmse(full_OLS_model)
#Backward BIC
n=nrow(data)
print(n)
mod_back_bic=step(full_OLS_model,direction='backward',k=log(n))

calc_loocv_rmse(mod_back_bic)
summary(mod_back_bic)$adj.r.squared

#AIC (Backward selection)
mod_back_aic=step(full_OLS_model,direction='backward')

calc_loocv_rmse(mod_back_aic)
summary(mod_back_aic)$adj.r.squared
# Subset
library(leaps)

mod_exhaustive = summary(regsubsets(SalePrice~. , data = data, nvmax = 27))
#mod_exhaustive$which
#mod_exhaustive$rss
best_r2_ind = which.max(mod_exhaustive$adjr2)
Adj_R2<-mod_exhaustive$which[best_r2_ind,]
Adj_R2
model_exh_r2=lm(SalePrice ~.-BsmtFullBath-BsmtHalfBath-ThreeSsnPorch,data=data)
calc_loocv_rmse(model_exh_r2)
p = ncol(mod_exhaustive$which)

mod_aic = n * log(mod_exhaustive$rss / n) + 2 * (2:p)

best_aic_ind = which.min(mod_aic)

mod_exhaustive$which[best_aic_ind, ]
model_exh_aic=lm(SalePrice ~.-BsmtFullBath-BsmtHalfBath-ThreeSsnPorch-HalfBath-SecondFlrSf
                 -EnclosedPorch,data=data)
calc_loocv_rmse(model_exh_aic)
n = nrow(data)
p = ncol(data)

mod_bic = n * log(mod_exhaustive$rss / n) + log(n) * (2:p)

#mod_bic

best_bic_ind = which.min(mod_bic)
#best_bic_ind

mod_exhaustive$which[best_bic_ind, ]
model_exh_bic=lm(SalePrice ~.-BsmtFullBath-BsmtHalfBath-ThreeSsnPorch-HalfBath-SecondFlrSf
```

```
                -EnclosedPorch-TotRmsAbvGrd-KitchenAbvGr-GarageArea,data=data)
calc_loocv_rmse(model_exh_bic)
mod_start = lm(SalePrice ~ 1, data = data)
mod_forwd_aic = step(
    mod_start,
    scope = SalePrice ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf + FirstFlrSf + SecondFlrSf+
    GrLivArea +BsmtFullBath+BsmtHalfBath+ FullBath +HalfBath+ BedroomAbvGr +KitchenAbvGr
    +TotRmsAbvGrd+ Fireplaces + GarageCars + GarageArea+
    WoodDeckSf + OpenPorchSf + ScreenPorch+ EnclosedPorch+ThreeSsnPorch,
    direction = 'forward')
calc_loocv_rmse(mod_forwd_aic)
mod_start = lm(SalePrice ~ 1, data = data)
mod_step_aic = step(
    mod_start,
    scope = SalePrice ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf + FirstFlrSf + SecondFlrSf+
    GrLivArea +BsmtFullBath+BsmtHalfBath+ FullBath +HalfBath+ BedroomAbvGr
    +KitchenAbvGr+TotRmsAbvGrd+ Fireplaces + GarageCars + GarageArea+
    WoodDeckSf + OpenPorchSf + ScreenPorch+ EnclosedPorch+ThreeSsnPorch,
    direction = 'both')
calc_loocv_rmse(mod_step_aic)
model_selected=lm(SalePrice ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf + FirstFlrSf +
    GrLivArea + FullBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars + GarageArea + WoodDeckSf + OpenPorchSf +
    ScreenPorch,data=data)
summary(model_selected)
library(lmtest)
bptest(model_selected)
model_selected=lm(SalePrice ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf + FirstFlrSf +
    GrLivArea + FullBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars + GarageArea + WoodDeckSf + OpenPorchSf +
    ScreenPorch,data=data)
library(olsrr)
ols_plot_resid_fit(model_selected)
shapiro.test(resid(model_selected))
ols_plot_resid_qq(model_selected)
which(cooks.distance(model_selected) > 4 / length(cooks.distance(model_selected)))
influence_rmvd=which(cooks.distance(model_selected)<= 4/length(cooks.distance(model_selected)))
non_inflnc_model = lm(SalePrice ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf + FirstFlrSf +
    GrLivArea + FullBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars + GarageArea + WoodDeckSf + OpenPorchSf +
    ScreenPorch,data = data,subset = influence_rmvd)
shapiro.test(resid(non_inflnc_model))
influence_rmvd=which(cooks.distance(model_selected)<= 4/length(cooks.distance(model_selected)))
non_inflnc_model = lm(SalePrice ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf + FirstFlrSf +
    GrLivArea + FullBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars + GarageArea + WoodDeckSf + OpenPorchSf +
    ScreenPorch,data = data,subset = influence_rmvd)
```

```r
ols_plot_resid_qq(non_inflnc_model)
ols_plot_resid_fit(non_inflnc_model)
bptest(non_inflnc_model)
library(dplyr)
library(corrplot)
data_preds=select(data, LotArea ,OverallQual , OverallCond , YearBuilt ,
    YearRemodAdd , BsmtFinSf1 , BsmtFinSf2 , TotalBsmtSf , FirstFlrSf ,
    GrLivArea , FullBath , BedroomAbvGr , KitchenAbvGr , TotRmsAbvGrd ,
    Fireplaces , GarageCars , GarageArea , WoodDeckSf , OpenPorchSf ,
    ScreenPorch)


corrplot(cor(data_preds), method = "color", type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45)
non_corr_preds<-lm(SalePrice ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf  +
    GrLivArea + FullBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars  + WoodDeckSf + OpenPorchSf +
    ScreenPorch, data=data,subset=influence_rmvd
   )
summary(non_corr_preds)
calc_loocv_rmse(non_corr_preds)
bptest(non_corr_preds)
library(faraway)
vif(non_corr_preds)
library(MASS)
non_corr_preds<-lm(SalePrice ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf  +
    GrLivArea + FullBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars  + WoodDeckSf + OpenPorchSf +
    ScreenPorch, data=data,subset=influence_rmvd
   )
# Set the plot size
par(mfrow = c(2,1))
bc = boxcox(non_corr_preds, plotit = TRUE)

boxcox(non_corr_preds, lambda = seq(0.5, 1, by = 0.1), plotit = TRUE)
par(mfrow=c(1,1))
sqrt_trans_mod<-lm(sqrt(SalePrice) ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf  +
    GrLivArea + FullBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars  + WoodDeckSf + OpenPorchSf +
    ScreenPorch, data=data,subset=influence_rmvd )
summary(sqrt_trans_mod)
bptest(sqrt_trans_mod)
weights <- 1 / residuals(non_corr_preds)^2
nrow(data)  # Length of the dataset
length(weights)  # Length of the weights vector

# Subset the data
subset_data <- data[1:length(weights), ]

# Fit WLS model
```

```
wls_model <- lm(SalePrice ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf  +
    GrLivArea + FullBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars  + WoodDeckSf + OpenPorchSf +
    ScreenPorch, data = subset_data, weights = weights)
summary(wls_model)
bptest(wls_model)
calc_loocv_rmse(wls_model)
weights <- 1 / residuals(non_corr_preds)^2
subset_data <- data[1:length(weights), ]
wls_model <- lm(SalePrice ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf  +
    GrLivArea + FullBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars  + WoodDeckSf + OpenPorchSf +
    ScreenPorch, data = subset_data, weights = weights)

plot(fitted(wls_model), weighted.residuals(wls_model),
     pch = 20, xlab = 'Fitted Value', ylab = 'Weighted Residual')
abline(h=0, lwd=3, col='steelblue')
sqrt_trans_mod<-lm(sqrt(SalePrice) ~ LotArea + OverallQual + OverallCond + YearBuilt +
    YearRemodAdd + BsmtFinSf1 + BsmtFinSf2 + TotalBsmtSf  +
    GrLivArea + FullBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars  + WoodDeckSf + OpenPorchSf +
    ScreenPorch, data=data,subset=influence_rmvd )
outlier_test_cutoff = function(model, alpha = 0.05) {
  n = length(resid(model))
  qt(alpha/(2 * n), df = df.residual(model) - 1, lower.tail = FALSE)
}

# vector of indices for observations deemed outliers.
cutoff = outlier_test_cutoff(sqrt_trans_mod, alpha = 0.05)

players_data_no_out=data[which(abs(rstudent(sqrt_trans_mod))<=cutoff),]
players_data_no_out
calc_loocv_rmse(sqrt_trans_mod)
```