Report

Ames Housing Regression Diagnostics with R

ALY6015 Intermediate Analytics

Module 1 Project 1

Author: Jainam Patel

Faculty: Prof. Richard He

Date: 15/01/2025

Introduction:

Ames housing dataset contains information from the Ames Office used in computing assessed values for individual residential properties sold in Ames, IA from 2006 to 2010.

The dataset contains 82 columns which include 23 nominal, 23 ordinal, 14 discrete, and 20 continuous variables (and 2 additional observation identifiers). The dataset has 2930 observations and 82 variables having both numerical and character values in the data.

Analysis:

EDA and Descriptive Analysis:

Q. Load the Ames housing dataset.

```
> ## 1. Loading Libraries and Dataset file
>
> library(dplyr)
> library(ggplot2)
> library(tidyverse)
>
> file <- read.csv("/Users/jp/Desktop/ALY6015/AmesHousing.csv")
> View(file)
> str(file)
'data.frame':   2930 obs. of  82 variables:
 $ Order           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ PID             : int  526301100 526350040 526351010 526353030
```

From above images we can interpret that there are 2930 observations and total of 82 variables in the dataset.

Q. Perform Exploratory Data Analysis and use descriptive statistics to describe the data.

```
> ## 2. Performing EDA
>
> summary(file)
     Order             PID             MS.SubClass      MS.Zoning          Lot.Frontage       Lot.Area
 Min.   :   1.0   Min.   :5.263e+08   Min.   : 20.00   Length:2930        Min.   : 21.00   Min.   :  1300
 1st Qu.: 733.2   1st Qu.:5.285e+08   1st Qu.: 20.00   Class :character   1st Qu.: 58.00   1st Qu.:  7440
 Median :1465.5   Median :5.355e+08   Median : 50.00   Mode  :character   Median : 68.00   Median :  9436
 Mean   :1465.5   Mean   :7.145e+08   Mean   : 57.39                      Mean   : 69.22   Mean   : 10148
 3rd Qu.:2197.8   3rd Qu.:9.072e+08   3rd Qu.: 70.00                      3rd Qu.: 80.00   3rd Qu.: 11555
 Max.   :2930.0   Max.   :1.007e+09   Max.   :190.00                      Max.   :313.00   Max.   :215245
                                                                          NA's   :490
```

This image contains Exploratory Data Analysis which shows the output about each column in the file about their Minimum value, 1st Quarter Tile, Median value, Mean value, 3rd Quartile and Maximum value.

In above image only first few columns are displayed and remaining columns follow the same pattern.

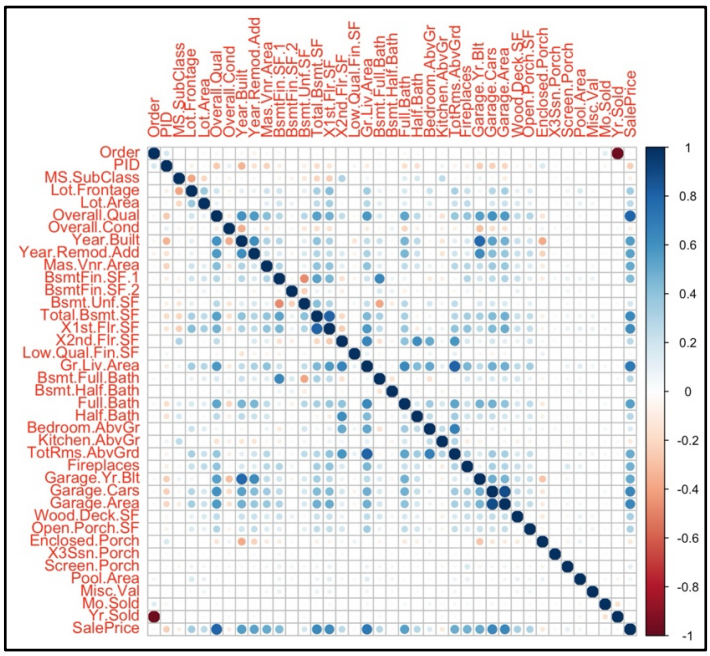
Q. Prepare the dataset for modeling by imputing missing values with the variable's **median value** or any other value that you prefer. Please find features whose missing values are more than 5% and show the list in the report.

```
# A tibble: 11 × 2
   Feature       MissingPercentage
   <chr>                     <dbl>
 1 Lot.Frontage              0.167
 2 Alley                     0.932
 3 Fireplace.Qu              0.485
 4 Garage.Type               0.0536
 5 Garage.Yr.Blt             0.0543
 6 Garage.Finish             0.0536
 7 Garage.Qual               0.0539
 8 Garage.Cond               0.0539
 9 Pool.QC                   0.996
10 Fence                     0.805
11 Misc.Feature              0.964
```

For the columns having numeric type values, I have imputed missing values in those records with the Median value of that column. For example, the column “Lot.Frontage” has median 68 and record 12 for that column has value 68 placed in it.

The 11 features listed in above image have missing values more than 5%.

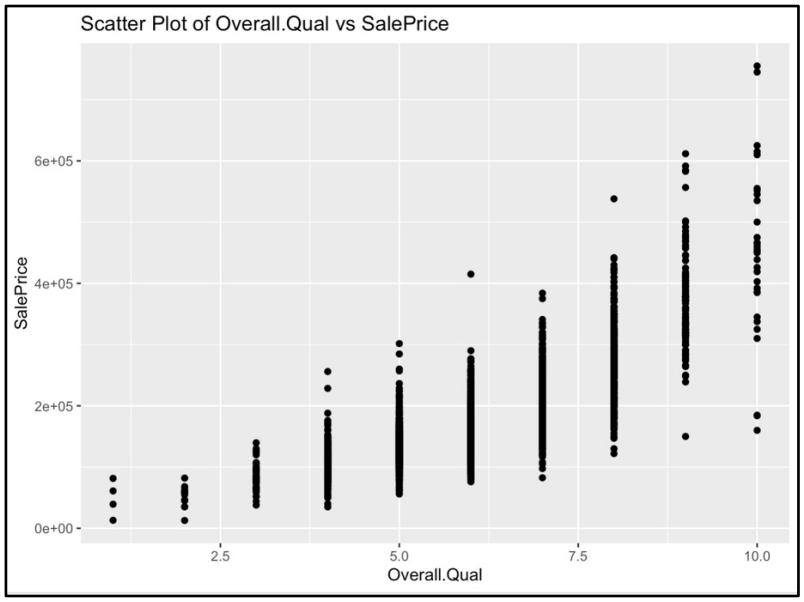
Q. Use the "cor()" function to produce a correlation matrix of the numeric values and produce a plot of the correlation matrix, and explain how to interpret it.

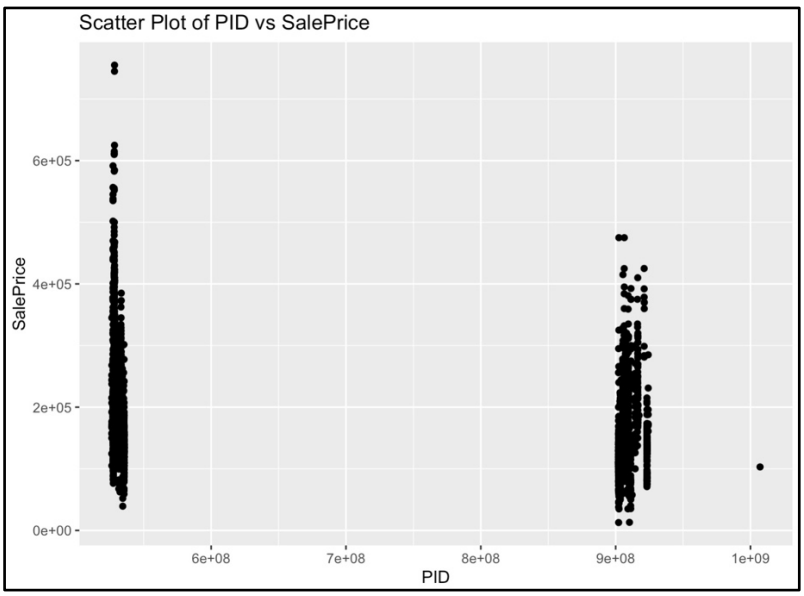


The correlation matrix above shows the relationships between various variables in a dataset. The colour and size of the circles represent the strength and direction of correlation, ranging from -1 (red, negative) to 1 (blue, positive).

From our interpretation, SalePrice has highest relation with Overall.Qual variable.

Q. Scatterplot of variables having highest, lowest and closest correlation to 0.5 with SalePrice.





The first scatter plot clearly demonstrates a positive correlation between "Overall.Qual" (overall quality) and "SalePrice," suggesting that homes of greater quality typically fetch higher prices when they are sold. As anticipated, the second scatter plot reveals no connection between "SalePrice" and "PID," which is probably a unique property identifier. The random distribution of points is explained by the fact that "PID" is an identifier and has no bearing on the sale price of the house.

Q. Regression Model using 6 Variables and interpreting in equation form.

```
Call:
lm(formula = SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Cars +
    Garage.Area + Total.Bsmt.SF + X1st.Flr.SF, data = file_clean_data)

Residuals:
    Min      1Q  Median      3Q     Max
-504291  -19381    -998   16660  280174

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.079e+05  3.249e+03 -33.208  < 2e-16 ***
Overall.Qual   2.455e+04  7.079e+02  34.687  < 2e-16 ***
Gr.Liv.Area    4.353e+01  1.862e+00  23.375  < 2e-16 ***
Garage.Cars    9.453e+03  2.071e+03   4.565 5.20e-06 ***
Garage.Area    3.020e+01  7.272e+00   4.152 3.39e-05 ***
Total.Bsmt.SF  2.493e+01  2.800e+00   8.904  < 2e-16 ***
X1st.Flr.SF    1.429e+01  3.222e+00   4.434 9.58e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 37330 on 2923 degrees of freedom
Multiple R-squared:  0.7821,    Adjusted R-squared:  0.7817
F-statistic:  1749 on 6 and 2923 DF,  p-value: < 2.2e-16
```

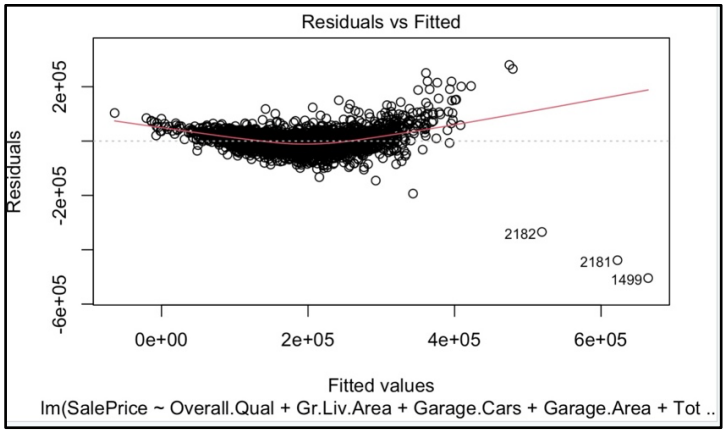
I have used variables: Overall.Qual, Gr.Liv.Area, Garage.Cars, Garage.Area, Total.Bsmt.SF and 1st.Flr.Sf and the dataframe used is 'file_clean_data' which has been prepared after EDA and imputing **Median** in missing values.

The value of Adjusted R-Squared is 0.78 as of now initially and is improved further.

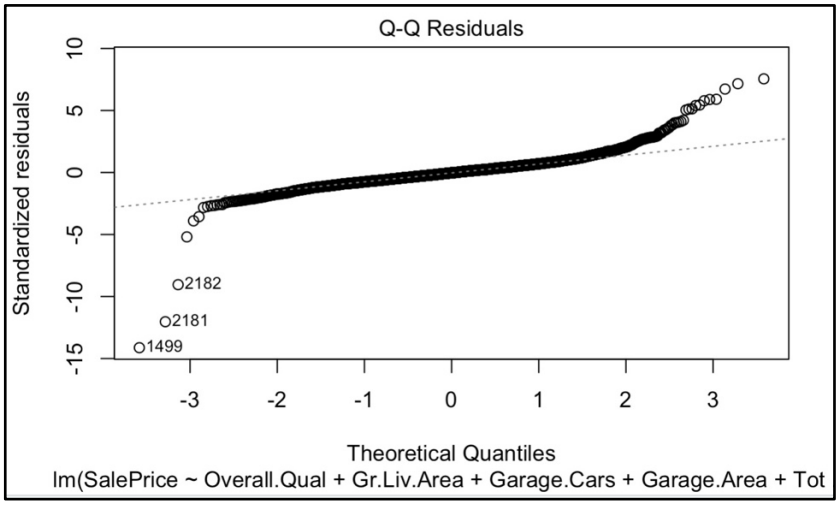
```
> cat("Regression Equation:", equation)
Regression Equation: SalePrice = -107890.33 + 24553.49 * Overall.Qual + 43.53 * Gr.Liv.Area +
9452.64 * Garage.Cars + 30.2 * Garage.Area + 24.93 * Total.Bsmt.SF + 14.29 * X1st.Flr.SF
```

This is the **regression equation** where **-107890.33** is intercept which means that if all variables are equal to zero, than **SalePrice** would be negative but this is not possible as without any variable, the property won't exist.

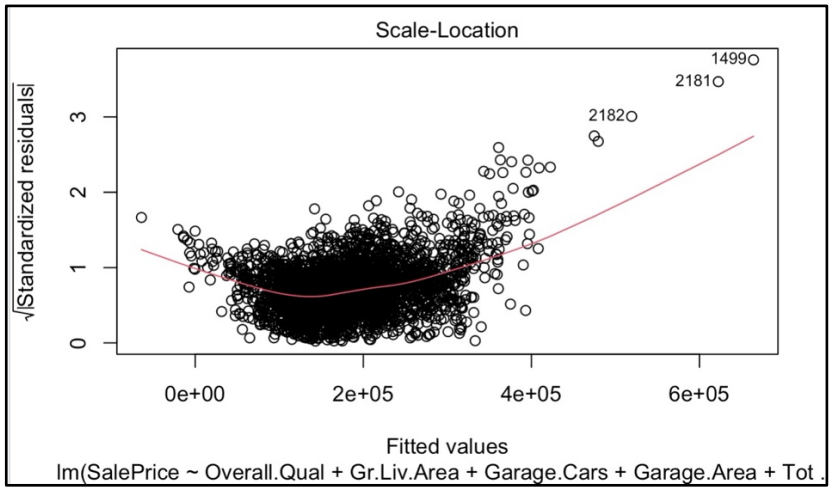
Q. "plot()" function to plot regression model and interpreting the four graphs that are produced.



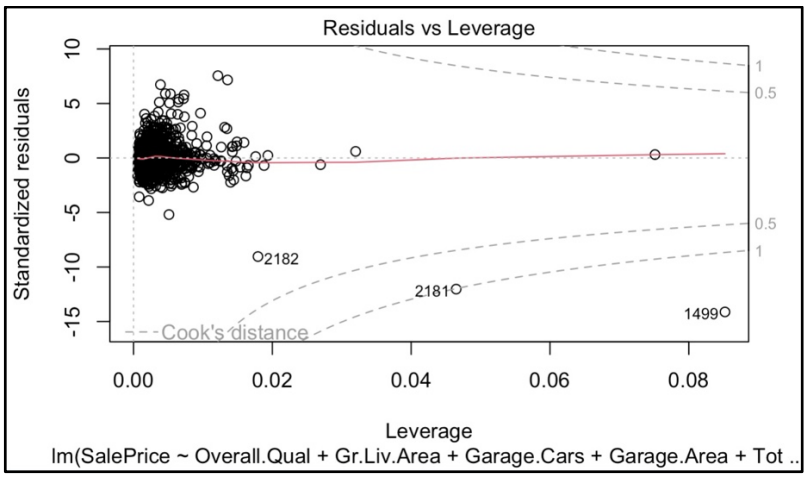
A curving pattern in the "Residuals vs. Fitted" graphic suggests non-linearity in the model and possible problems with its underlying assumptions. It seems that observations 1499, 2182, and 2181 are significant outliers.



Significant outliers like 2182, 2181, and 1499 are among the residuals that vary from normality at the tails, as the "Q-Q Residuals" graphic demonstrates.



Given that the variance of residuals rises with fitted values, the "Scale-Location" plot indicates heteroscedasticity. Here, observations 2182, 2181, and 1499 are emphasized.



Influential observations with high leverage, such as 2182, 2181, and 1499, are identified by the "Residuals vs. Leverage" plot and have a major effect on the model.

Q. Multicollinearity check and its findings.

```
> print(vif_value)
 Overall.Qual   Gr.Liv.Area   Garage.Cars   Garage.Area Total.Bsmt.SF   X1st.Flr.SF
     2.096960      1.862353      5.211460      5.138576      3.197943      3.350995
```

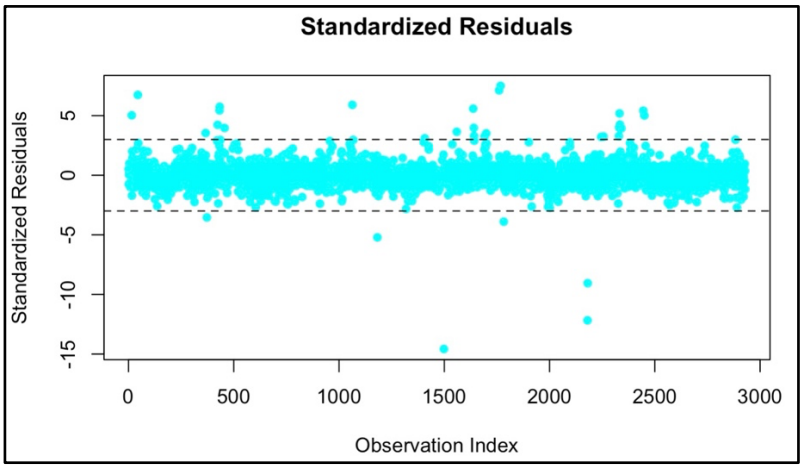
Multicollinearity is used for checking that if independent variables in our model are correlated or not and is measured using **Variance Inflation Factor (VIF**). If VIF is more than 5, then the variables are correlated and its correlation effect the output influences the dependent variable result.

Here, **"Garage.Area"** and **"Garage.Cars"** have **VIF** more than 5. **I would take steps to remove any one variable and check the model again.**

```
Call:
lm(formula = SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Area +
    Total.Bsmt.SF + X1st.Flr.SF, data = file_clean_data)

Residuals:
    Min      1Q  Median      3Q     Max
-528345  -19383    -911   16788  278972

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.084e+05  3.258e+03 -33.271  < 2e-16 ***
Overall.Qual   2.532e+04  6.901e+02  36.689  < 2e-16 ***
Gr.Liv.Area    4.406e+01  1.865e+00  23.628  < 2e-16 ***
Garage.Area    5.753e+01  4.141e+00  13.891  < 2e-16 ***
Total.Bsmt.SF  2.429e+01  2.806e+00   8.656  < 2e-16 ***
X1st.Flr.SF    1.386e+01  3.231e+00   4.288 1.86e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 37460 on 2924 degrees of freedom
Multiple R-squared:  0.7805,    Adjusted R-squared:  0.7802
F-statistic:  2080 on 5 and 2924 DF,  p-value: < 2.2e-16

>
> coefficients <- coef(regress_model)
>
> equation <- paste0("SalePrice = ", round(coefficients[1], 2), " + ",
+                    paste(
+                      paste0(round(coefficients[-1], 2), " * ", names(coefficients[-1])), co
llapse = " + "
+                    ))
>
> cat("Regression Equation:", equation)
Regression Equation: SalePrice = -108397.84 + 25318.3 * Overall.Qual + 44.06 * Gr.Liv.Area +
57.53 * Garage.Area + 24.29 * Total.Bsmt.SF + 13.86 * X1st.Flr.SF>
> print(vif(regress_model))
 Overall.Qual   Gr.Liv.Area   Garage.Area Total.Bsmt.SF   X1st.Flr.SF
     1.979500      1.855019      1.655320      3.189857      3.348119
```

As seen from above image, I have removed variable **"Garage.Cars**" and checked the model again for multicollinearity. As we can interpret that all the 5 variables now have **VIF** value less 5. This is a good model as of now.

Q. Outliers and Residuals Findings



The above image shows standardized residuals against observations indices to evaluate model's residuals' behaviour.

From above image we can interpret that most residuals lie within range -3 to +3, indicating they are normal but few are extreme outliers.

Points above black-dashed lines are potential influential points and should be removed.

Q. Attempting to remove any issues in model.

```
Call:
lm(formula = SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Area +
    Total.Bsmt.SF + X1st.Flr.SF, data = file_clean_data)

Residuals:
    Min      1Q  Median      3Q     Max
-105506  -17759     -36   16935  113925

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.024e+05  2.679e+03 -38.215  < 2e-16 ***
Overall.Qual   2.292e+04  5.593e+02  40.978  < 2e-16 ***
Gr.Liv.Area    4.562e+01  1.545e+00  29.529  < 2e-16 ***
Garage.Area    5.772e+01  3.337e+00  17.296  < 2e-16 ***
Total.Bsmt.SF  2.944e+01  2.299e+00  12.806  < 2e-16 ***
X1st.Flr.SF    1.373e+01  2.600e+00   5.279 1.39e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 29950 on 2892 degrees of freedom
Multiple R-squared:  0.8304,    Adjusted R-squared:  0.8302
F-statistic:  2833 on 5 and 2892 DF,  p-value: < 2.2e-16
```

As seen in the above image, I have removed outliers from the model and again ran the regression model with 5 variables. As we can observe, the value of Adjusted R-Squared is increased to 0.8302 from previously 0.78. This shows that our model is improved after removing the outliers.

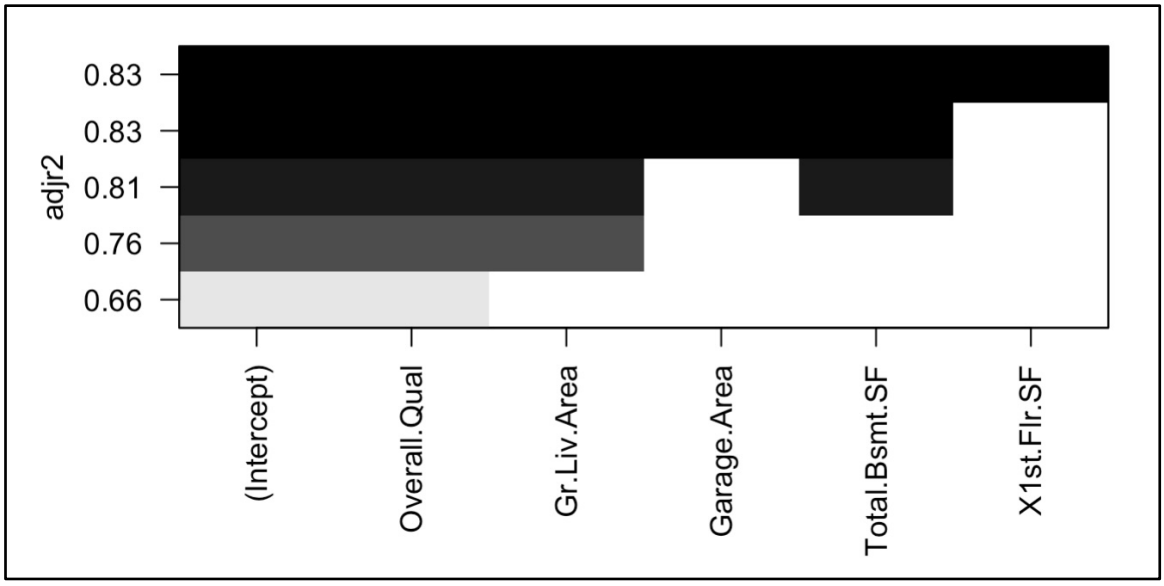
Along with Adjusted R-Squared, Root-Mean-Squared-Error has decreased from approximately 37000 to 29000.

These indicators mean that model is overall improved.

```
Random Forest

2898 samples
   5 predictor

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 2898, 2898, 2898, 2898, 2898, 2898, ...
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared   MAE
  2     27185.03  0.8605112  19280.42
  3     27382.33  0.8582650  19364.67
  5     27868.17  0.8532906  19666.75

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 2.
> predictions <- predict(model_cv, newdata = testing_data)
>
> rmse_value <- sqrt(mean((predictions - testing_data$SalePrice)^2))
> r_squared <- cor(predictions, testing_data$SalePrice)^2
> print(paste("RMSE:", rmse_value))
[1] "RMSE: 12403.8938494054"
> print(paste("R-squared:", r_squared))
[1] "R-squared: 0.970280816994056"
> n <- nrow(testing_data)
> p <- length(model_cv$finalModel$coefficients) - 1
> adjusted_r_squared <- 1 - ((1 - r_squared) * (n - 1) / (n - p - 1))
> print(paste("Adjusted R-squared:", adjusted_r_squared))
[1] "Adjusted R-squared: 0.970332145462115"
```

Further, I applied 10 Fold Cross-Validation to improve my model. After applying Cross-Validation, the Adjusted R-Squared is increased to 0.97 from previous 0.8302 and Root-Mean-Squared-Error has been decreased to approximately 12403 from 29000 previously, which is very significant jump.

Q. All Subsets Regression to identify best feature model.

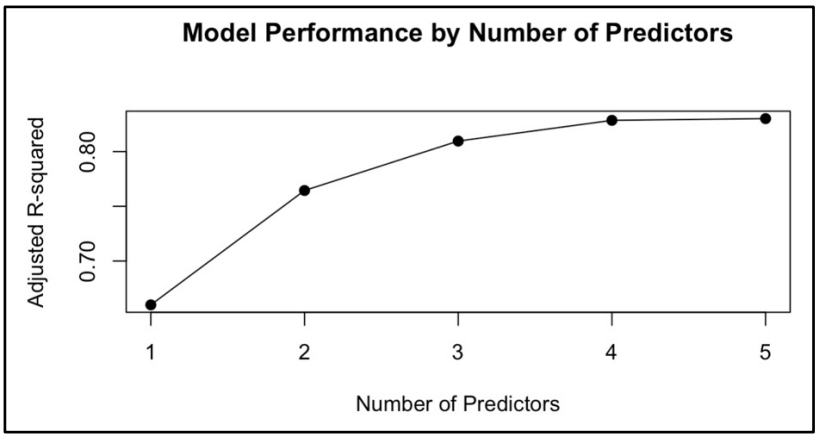


From the above image, we can interpret that variables “Overall.Qual” and “Gr.Liv.Area” significantly improves the Adjusted R-Squared and indicating their strong contribution in model’s predictive power.

Also, using variables “Garage.Area” and “Total.Bsmt.Sf” results in smaller gains only.

```
5 Variables  (and intercept)
             Forced in Forced out
Overall.Qual      FALSE      FALSE
Gr.Liv.Area       FALSE      FALSE
Garage.Area       FALSE      FALSE
Total.Bsmt.SF     FALSE      FALSE
X1st.Flr.SF       FALSE      FALSE
1 subsets of each size up to 5
Selection Algorithm: exhaustive
         Overall.Qual Gr.Liv.Area Garage.Area Total.Bsmt.SF X1st.Flr.SF
1  ( 1 ) "*"          " "         " "         " "           " "
2  ( 1 ) "*"          "*"         " "         " "           " "
3  ( 1 ) "*"          "*"         " "         "*"           " "
4  ( 1 ) "*"          "*"         "*"         "*"           " "
5  ( 1 ) "*"          "*"         "*"         "*"           "*"
>
```

Upon using “regsubsets()” function, the algorithm makes subsets as indicated by ‘*’ in above image. I have used upto 5-features because my model contains 5 features instead of atleast 4 numeric variables.



From the above image we can see, that having 4 features and 5 features subsets have almost same Adjusted R-Squared values.

```
Preferred Model Equation:
 SalePrice = -97832.41 + 22045.82 * Overall.Qual + 48.73 * Gr.Liv.Area + 15727.48 * Garage.Cars + 40.35 * Total.Bsmt.SF
```

I have selected 4 feature model because it strikes best balance between complexity and predictive accuracy. Its Adjusted R-Squared value is 0.8256 while for 5 feature model is 0.8277 but it avoids marginal gain from adding X1st.Flr.SF.

Adding 5th feature X1st.Flr.SF increases Adjusted R-Squared and its coefficient suggests relatively low contribution to SalePrice.

Q. Comparing Preferred Model from **Step 13(Best Subsets Feature)** and **Step 12(My Model).**

```
> # Print metrics for comparison
> cat("Step 12 Model:\n")
Step 12 Model:
> cat("  Adjusted R^2:", round(adj_r2_step12, 4), "\n")
  Adjusted R^2: 0.8302
> cat("  RMSE:", round(rmse_step12, 2), "\n")
  RMSE: 12524.05
>
> cat("Step 13 Model:\n")
Step 13 Model:
> cat("  Adjusted R^2:", round(adj_r2_step13, 4), "\n")
  Adjusted R^2: 0.8256
> cat("  RMSE:", round(rmse_step13, 2), "\n")
  RMSE: 30321.78
```

I would prefer Model in Step 12 has its **RMSE is 12524.05** compared to **RMSE of 30321.78** for 4-Feature Model which is significantly less.

Also, Adjusted R-Squared is almost same while Model in Step 12 has marginally higher value than 4-Feature model.

Model in Step 12 is improved significantly because Outliers were removed and Cross-Validation was applied.

Conclusion:

1. The Ames Housing Dataset investigation gave useful insights into the factors that influenced residential property selling prices in Ames, Iowa, between 2006 and 2010. A full exploratory data analysis was performed, followed by data preparation to handle missing values and outliers, resulting in a clean and robust dataset for modeling.
2. A predictive model was created utilizing regression modeling and key parameters such as "Overall.Qual," "Gr.Liv.Area," and "Garage.Area," with an initial Adjusted R-Squared of 0.78. This was subsequently modified by tackling multicollinearity and outlier effects, resulting in an improved Adjusted R-Squared of 0.83.
3. Advanced techniques, such as 10-fold cross-validation and all subsets regression, were used to optimize the model. Cross-validation considerably improved the model's performance, bringing Adjusted R-Squared to 0.97 and lowering RMSE to around 12,403.
4. The comparison of the preferred model and the best subsets feature model proved the efficacy of reducing outliers and using cross-validation. The favored model demonstrated greater accuracy and stability, making it ideal for predictive applications.
5. Finally, the revised model provides a viable tool for estimating residential property selling values, emphasizing the necessity of data pretreatment and validation in achieving robust prediction results. Future research could look into additional features and nonlinear interactions to improve forecast accuracy.

References:

Books:

- Bluman, A. (2018). Elementary statistics: A step by step approach (10th ed.). McGraw Hill.ISBN 13: 978-1-259-755330.
- R. Kabacoff, *R in Action*, 2nd Edition, Manning Publisher ISBN 978-161-7291-388.

Canvas:

Northeastern University. (n.d.).Pre-Assignment Lab Video. Northeastern University Canvas. https://northeastern.instructure.com/courses/200351/assignments/2542446

Appendix:

```
## 1. Loading Libraries and Dataset file


library(dplyr)
library(ggplot2)
library(tidyverse)


file <- read.csv("/Users/jp/Desktop/ALY6015/AmesHousing.csv")
View(file)
str(file)


## 2. Performing EDA


summary(file)


#install.packages("skimr")
library(skimr)
skim(file)


## 3. Missing values and replacing them


# Replace empty values for numeric columns with the median of the column
file_clean_data <- file %>%
  mutate(across(where(is.numeric),
                ~ ifelse(is.na(.) | . == "",
                         median(as.numeric(na.omit(as.numeric(.))), na.rm = TRUE),
                         as.numeric(.))))


# Verify the count of empty or missing values after cleaning
empty_or_na_count_1 <- sum(is.na(file))
print(paste("Total Empty or Missing Values before cleaning:", empty_or_na_count_1))
```

```
empty_or_na_count_2 <- sum(is.na(file_clean_data))
print(paste("Total empty or missing values after cleaning:", empty_or_na_count_2))

library(tidyverse)
missing_features <- file %>%
  summarise_all(~mean(is.na(.))) %>%
  pivot_longer(cols = everything(), names_to = "Feature", values_to = "MissingPercentage") %>%
  filter(MissingPercentage > 0.05)

print(missing_features)

## 4. Correlation Matrix

##install.packages("corrplot")
library(corrplot)
library(dplyr)
correlation_matrix <- cor(select(file_clean_data, where(is.numeric)), use = "complete.obs")
print(correlation_matrix)

## 5. Correlation Plot

corrplot(correlation_matrix, method = "circle")

## 6. Scatter Plot

corr_saleprice <- cor(select(file_clean_data, where(is.numeric)), use = "complete.obs")["SalePrice", ]
highest_corr_saleprice <- names(sort(corr_saleprice, decreasing = TRUE)[2])
lowest_corr_saleprice <- names(which.min(corr_saleprice))
closest_05_corr_saleprice <- names(which.min(abs(corr_saleprice - 0.5)))

# Sort correlations in decreasing order, excluding SalePrice itself
```

```
top_6_features <- names(sort(corr_saleprice[-which(names(corr_saleprice) == "SalePrice")], decreasing = TRUE)[1:6])

# Print the top 6 features
top_6_features

print(paste("Variable with Highest Correlation Coefficient with SalePrice is:",highest_corr_saleprice))
print(paste("Variable with Lowest Correlation Coefficient with SalePrice is:",lowest_corr_saleprice))
print(paste("Variable with Closest Correlation Coefficient with SalePrice is:",closest_05_corr_saleprice))

library(ggplot2)

ggplot(file_clean_data, aes_string(x = highest_corr_saleprice, y = "SalePrice")) +
  geom_point() +
  labs(title = paste("Scatter Plot of", highest_corr_saleprice, "vs SalePrice"),
       x = highest_corr_saleprice, y = "SalePrice")

ggplot(file_clean_data, aes_string(x = lowest_corr_saleprice, y = "SalePrice")) +
  geom_point() +
  labs(title = paste("Scatter Plot of", lowest_corr_saleprice, "vs SalePrice"),
       x = lowest_corr_saleprice, y = "SalePrice")

ggplot(file_clean_data, aes_string(x = closest_05_corr_saleprice, y = "SalePrice")) +
  geom_point() +
  labs(title = paste("Scatter Plot of", closest_05_corr_saleprice, y = "SalePrice"),
       x = closest_05_corr_saleprice, y = "SalePrice")

## 7. Regression Model

regress_model <- lm(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Cars + Garage.Area + Total.Bsmt.SF + X1st.Flr.SF,
```

```
                 data = file_clean_data)
summary(regress_model)

## 8. Interpreting equation model

coefficients <- coef(regress_model)

equation <- paste0("SalePrice = ", round(coefficients[1], 2), " + ",
                   paste(
                     paste0(round(coefficients[-1], 2), " * ", names(coefficients[-1])), collapse = " + "
                   ))

cat("Regression Equation:", equation)

## 9. Plotting Regression Model

plot(regress_model)

## 10. Multicollinearity Check

##install.packages("car")
library(car)

vif_value <- vif(regress_model)
print(vif_value)

regress_model <- lm(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Area + Total.Bsmt.SF +
X1st.Flr.SF,
        data = file_clean_data)
summary(regress_model)
```

```
coefficients <- coef(regress_model)

equation <- paste0("SalePrice = ", round(coefficients[1], 2), " + ",
                   paste(
                     paste0(round(coefficients[-1], 2), " * ", names(coefficients[-1])), collapse = " + "
                   ))

cat("Regression Equation:", equation)

print(vif(regress_model))

## 11. Outliers and Residuals

## Standardized residuals
standardized_residuals <- rstandard(regress_model)

# Potential outliers
outliers <- which(abs(standardized_residuals) > 3)
print(outliers)

plot(standardized_residuals,
    main = "Standardized Residuals",
    xlab = "Observation Index",
    ylab = "Standardized Residuals",
    pch = 16, col = "cyan")
abline(h = c(-3, 3), col = "black", lty = 2)

## 12. Model Correction Attempt

file_clean_data <- file_clean_data[-c(outliers), ]
```

```
regress_model <- lm(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Area + Total.Bsmt.SF + X1st.Flr.SF,
                    data = file_clean_data)
summary(regress_model)


coefficients <- coef(regress_model)


equation <- paste0("SalePrice = ", round(coefficients[1], 2), " + ",
                   paste(
                     paste0(round(coefficients[-1], 2), " * ", names(coefficients[-1])), collapse = " + "
                   ))


cat("Regression Equation:", equation)


### Cross-Validation


##install.packages("caret")
library(caret)


set.seed(123)
training_index <- createDataPartition(file_clean_data$SalePrice, p = 0.8, list = FALSE)
training_data <- file_clean_data[training_index, ]
testing_data <- file_clean_data[-training_index, ]


training_control <- trainControl(method = "cv", number = 10)
model_cv <- train(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Area + Total.Bsmt.SF + X1st.Flr.SF,
                  data = file_clean_data)


print(model_cv)
predictions <- predict(model_cv, newdata = testing_data)
```

```
rmse_value <- sqrt(mean((predictions - testing_data$SalePrice)^2))
r_squared <- cor(predictions, testing_data$SalePrice)^2

n <- nrow(testing_data)
p <- length(model_cv$finalModel$coefficients) - 1
adjusted_r_squared <- 1 - ((1 - r_squared) * (n - 1) / (n - p - 1))

print(paste("RMSE:", rmse_value))
print(paste("R-squared:", r_squared))
print(paste("Adjusted R-squared:", adjusted_r_squared))

## 13. Subsets Regression Method

##install.packages("leaps")
library(leaps)

every_subsets <- regsubsets(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Area + Total.Bsmt.SF
+ X1st.Flr.SF,
                            data = file_clean_data)
summary(every_subsets)

sub_summary <- summary(every_subsets)

adj_r2 <- sub_summary$adjr2
best_models <- which(adj_r2 == max(adj_r2))
best_models

plot(every_subsets, scale = "adjr2")
```

```
# Perform all subsets regression
all_subsets <- regsubsets(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Area + Total.Bsmt.SF + X1st.Flr.SF,
                          data = file_clean_data,
                          nvmax = 5)

# View a summary of the results
all_subsets_summary <- summary(all_subsets)

# View metrics for each model size
all_subsets_summary

plot(all_subsets_summary$adjr2, type = "o", pch = 19,
     xlab = "Number of Predictors", ylab = "Adjusted R-squared",
     main = "Model Performance by Number of Predictors")

# Best 1-feature model
model_1 <- lm(SalePrice ~ Overall.Qual, data = file_clean_data)
summary(model_1)

# Best 2-feature model
model_2 <- lm(SalePrice ~ Overall.Qual + Gr.Liv.Area, data = file_clean_data)
summary(model_2)

# Best 3-feature model
model_3 <- lm(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Cars, data = file_clean_data)
summary(model_3)

# Best 4-feature model
model_4 <- lm(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Cars + Total.Bsmt.SF, data = file_clean_data)
```

```
summary(model_4)

# Best 5-feature model
model_5 <- lm(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Cars + Total.Bsmt.SF + X1st.Flr.SF, data = file_clean_data)
summary(model_5)

# Coefficients of the preferred model
coefficients <- coef(model_4)

# Construct the regression equation as a string
equation <- paste0(
  "SalePrice = ", round(coefficients[1], 2),
  " + ", round(coefficients[2], 2), " * Overall.Qual",
  " + ", round(coefficients[3], 2), " * Gr.Liv.Area",
  " + ", round(coefficients[4], 2), " * Garage.Cars",
  " + ", round(coefficients[5], 2), " * Total.Bsmt.SF"
)

# Print the equation to the console
cat("Preferred Model Equation:\n", equation, "\n")

## 14. Model Comparison

# Calculate RMSE for Step 12 Model
predictions_step12 <- predict(model_cv, file_clean_data)
rmse_step12 <- sqrt(mean((predictions_step12 - file_clean_data$SalePrice)^2))
adj_r2_step12 <- summary(regress_model)$adj.r.squared

# Calculate RMSE for Step 13 Model
predictions_step13 <- predict(model_4, file_clean_data)
```

```
rmse_step13 <- sqrt(mean((predictions_step13 - file_clean_data$SalePrice)^2))
adj_r2_step13 <- summary(model_4)$adj.r.squared

# Print metrics for comparison
cat("Step 12 Model:\n")
cat("  Adjusted R^2:", round(adj_r2_step12, 4), "\n")
cat("  RMSE:", round(rmse_step12, 2), "\n")

cat("Step 13 Model:\n")
cat("  Adjusted R^2:", round(adj_r2_step13, 4), "\n")
cat("  RMSE:", round(rmse_step13, 2), "\n")
```