**MODULE: 01**

## **BY: PAYAL SHARMA**

**“Regression Diagnostic With R**

**For Ames Housing Dataset “**

**Course: Intermediate Analytics**

**Winter 2025 – January 11**

**College of Professional Studies, CPS**

**A black background with a black square

Description automatically generated with medium confidence**

**Module: 01**

**Table of Contents:**

1. **Introduction...**
2. **Data Analysis for Ames Housing Dataset ...**
3. **Conclusion ...**
4. **References…**
5. **Appendix...**

**Introduction:**

The Ames Housing dataset is a rich source of information for evaluating the factors influencing property values in Ames, Iowa, between 2006 and 2010. The primary objective of this report is to apply linear regression techniques to predict SalePrice and assess how various features contribute to housing prices. This analysis uses R programming to:

* Conduct exploratory data analysis (EDA).
* Fit a robust linear regression model.
* Validate key assumptions through regression diagnostics.
* Present findings using visualizations and statistical summaries.

The dataset comprises 82 variables describing property characteristics such as lot size, building quality, basement area, and garage capacity. This report focuses on identifying key predictors of housing prices and building an interpretable and accurate regression model.

1. **Reading the CSV File:**

data <- read.csv("C:/Users/ravin/Downloads/AmesHousing.csv")

This line reads the CSV file located at the specified path and stores it in a variable named data. The read.csv function is used to import data from a CSV file into a data frame in R.

1. **Inspecting the Structure of the Data:**

str(data)

The str function provides a compact display of the internal structure of an R object. In this case, it shows the structure of the data data frame. The output indicates that data is a data frame with 2930 observations (rows) and 82 variables (columns).

1. **Variables in the Data Frame:** The output of the str function lists all the variables (columns) in the data frame along with their data types and a preview of their values. Here are a few examples:
   * $ Order: int 1 2 3 4 5 6 7 8 9 10 ...  
     This indicates that the Order variable is of type integer (int) and shows the first few values.
   * $ MS.Zoning: chr "RL" "RH" "RL" "RL" ...  
     This indicates that the MS.Zoning variable is of type character (chr) and shows the first few values.
   * $ Lot.Frontage: int 141 80 81 93 74 78 41 43 39 60 ...  
     This indicates that the Lot.Frontage variable is of type integer (int) and shows the first few values.

Each variable in the data frame represents a different attribute of the houses in the Ames Housing dataset. The data types include integers (int), characters (chr), and possibly other types not shown in the snippet.

1. **Calculating Missing Values:**

missing\_values <- colSums(is.na(data))

This line calculates the number of missing values in each column of the data data frame. The is.na function checks for NA values, and colSums sums these NA values for each column.

1. **Filtering Columns with Missing Values:**

missing\_values[missing\_values > 0]

This line filters the missing\_values vector to show only the columns that have missing values. The result is a named vector where the names are the column names and the values are the counts of missing values.

Here's the interpretation of the output:

* **Lot.Frontage**: 490 missing values
* **Alley**: 2732 missing values
* **Mas.Vnr.Area**: 23 missing values
* **Bsmt.Qual**: 79 missing values
* **Bsmt.Cond**: 79 missing values
* **Bsmt.Exposure**: 79 missing values
* **BsmtFin.Type.1**: 79 missing values
* **BsmtFin.SF.1**: 1 missing value
* **BsmtFin.Type.2**: 79 missing values
* **BsmtFin.SF.2**: 1 missing value
* **Bsmt.Unf.SF**: 1 missing value
* **Total.Bsmt.SF**: 1 missing value
* **Bsmt.Full.Bath**: 2 missing values
* **Bsmt.Half.Bath**: 2 missing values
* **Fireplace.Qu**: 1422 missing values
* **Garage.Type**: 157 missing values
* **Garage.Yr.Blt**: 159 missing values
* **Garage.Finish**: 157 missing values
* **Garage.Cars**: 1 missing value
* **Garage.Area**: 1 missing value
* **Garage.Qual**: 158 missing values
* **Garage.Cond**: 158 missing values
* **Pool.QC**: 2917 missing values
* **Fence**: 2358 missing values
* **Misc.Feature**: 2824 missing values

This output helps identify which columns have missing data and how many values are missing in each column. This is crucial for data cleaning and preprocessing steps.

1. Summary statistics for continuous variables

summary(data)

* + 1. **Order**:

Range: 1 to 2930

Mean: 1465.5

Median: 1465.5

* + 1. **PID**:

Range: 526301100 to 1007000000

Mean: 714500000

Median: 535500000

* + 1. **MS.SubClass**:

Range: 20 to 190

Mean: 57.39

Median: 50

* + 1. **Lot.Frontage**:

Range: 21 to 313

Mean: 69.22

Median: 68

* + 1. Missing values: 490

**Lot.Area**:

Range: 1300 to 215245

Mean: 10148

Median: 9436

* + 1. **Overall.Qual**:

Range: 1 to 10

Mean: 6.095

Median: 6

* + 1. **Overall.Cond**:

Range: 1 to 9

Mean: 5.563

Median: 5

* + 1. **Year.Built**:

Range: 1872 to 2010

Mean: 1971

Median: 1973

* + 1. **Year.Remod.Add**:

Range: 1950 to 2010

Mean: 1984

Median: 1993

* + 1. **Mas.Vnr.Area**:

Range: 0 to 1600

Mean: 101.9

Median: 0

Missing values: 23

* + 1. **BsmtFin.SF.1**:

Range: 0 to 5644

Mean: 442.6

Median: 370

Missing values: 1

* + 1. **Total.Bsmt.SF**:

Range: 0 to 6110

Mean: 1052

Median: 990

Missing values: 1

* + 1. **X1st.Flr.SF**:

Range: 334 to 5095

Mean: 1159.6

Median: 1084

* + 1. **Gr.Liv.Area**:

Range: 334 to 5642

Mean: 1500

Median: 1442

* + 1. **Garage.Area**:

Range: 0 to 1488

Mean: 472.8

Median: 480

Missing values: 1

* + 1. **SalePrice**:

Range: 12789 to 755000

Mean: 180796

Median: 160000

These summary statistics provide a quick overview of the distribution and central tendency of the continuous variables in our dataset. They help in understanding the range, mean, median, and presence of missing values for each variable.

1. **Data Cleaning**
2. **Imputing Missing Values:**

if("Lot.Frontage" %in% colnames(data)) {

data$Lot.Frontage[is.na(data$Lot.Frontage)] <- median(data$Lot.Frontage, na.rm = TRUE)

}

This code checks if the Lot.Frontage column exists in the data data frame. If it does, it replaces any missing values (NA) in the Lot.Frontage column with the median value of that column. The na.rm = TRUE argument ensures that the median is calculated excluding the missing values.

* The reason of choosing median over mean : *"The median was chosen for imputation because it is less sensitive to outliers, ensuring that extreme values in the dataset do not skew the imputed values. This approach is particularly useful for variables like Lot.Frontage, where outliers could distort the distribution."*

1. **Removing Outliers:**

data <- data[data$Gr.Liv.Area <= 4000, ]

This line removes rows from the data data frame where the Gr.Liv.Area (above-ground living area) is greater than 4000 square feet. This step is based on the documentation recommendation to exclude extremely large houses that might skew the analysis.

* Justification for removing houses with Gr. Liv.Area : *"Houses with extremely large living areas (above 4,000 sq. ft.) were removed to prevent skewing the analysis. These properties are often outliers and may represent luxury homes with unique pricing dynamics that do not align with the majority of the dataset. Removing them ensures that the model is more representative of typical housing prices."*

1. **Exploratory Data Analysis**

Distribution of the Target Variable (SalePrice):

ggplot(data, aes(x = SalePrice)) +

geom\_histogram(binwidth = 10000, fill = "**pink",** color = "**black**") +

labs(title = "Distribution of Sale Prices", x = "Sale Price", y = "Count")

A graph of a graph of a price

Description automatically generated with medium confidence

* The histogram shows the distribution of SalePrice for houses. It is right-skewed, indicating that most houses are priced between $100,000 and $300,000, with a peak around $200,000. *"The right-skewed distribution of SalePrice indicates that most houses are priced in the lower range (between $100,000 and $300,000), with fewer properties in the higher price brackets. This skewness suggests that a log transformation of SalePrice could be beneficial for linear regression models, as it would help normalize the distribution and improve model performance."*

1. **Correlation Heatmap of Numerical Variables**

**Selecting Numerical Variables:**

num\_vars <- data %>% select\_if(is.numeric)

This line selects all numerical variables from the data data frame using the select\_if function with the is.numeric predicate.

**Calculating the Correlation Matrix:**

corr\_matrix <- cor(num\_vars, use = "complete.obs")

This line calculates the correlation matrix for the selected numerical variables. The use = "complete.obs" argument ensures that only complete observations (rows without missing values) are used in the calculation.

**Plotting the Correlation Heatmap:**

corrplot(corr\_matrix, method = "color", tl.cex = 0.6)

A screen shot of a graph

Description automatically generated

This line creates a correlation heatmap using the corrplot function. The method = "color" argument specifies that the heatmap should use colors to represent correlation values, and tl.cex = 0.6 adjusts the text label size.

* "The correlation heatmap revealed strong positive relationships between Gr.Liv.Area, Total.Bsmt.SF, and Garage.Area. While these variables are important predictors, their high correlation (multicollinearity) can inflate the variance of the coefficient estimates and reduce the interpretability of the model. To address this, we could consider dimensionality reduction techniques like Principal Component Analysis (PCA) or feature selection methods to retain only the most relevant predictors."

**Scatter Plot for Ground Living Area vs Sale Price**

**Creating the Scatter Plot:**

ggplot(data, aes(x = Gr.Liv.Area, y = SalePrice)) +

geom\_point(alpha = 0.5) +

geom\_smooth(method = "lm", fill = "black", color = "orange") +

labs(title = "Gr Liv Area vs Sale Price", x = "Ground Living Area", y = "Sale Price")

A graph showing a line of black dots

Description automatically generated

This code creates a scatter plot to visualize the relationship between Gr.Liv.Area (ground living area) and SalePrice.

1. Linear Growth in Price: As the above ground living area increases, the SalePrice rises consistently. This trend reflects the inherent value that buyers place on larger living spaces, which offer greater functionality and comfort.
2. Outliers at Larger Areas: A few data points with exceptionally high Gr.Liv.Area correspond to disproportionately high or low prices.These could represent luxury homes or mispriced properties, which may need further investigation.
3. Impact of Size: The slope of the trendline suggests that each additional unit of living area contributes significantly to housing price increments, emphasizing the importance of space in determining property value.
4. **Feature Engineering**

**Convert Categorical Variables to Factors:**

data <- data %>% mutate(across(where(is.character), as.factor))

This line converts all character variables in the data data frame to factors. This is useful for modeling purposes, as many machine learning algorithms require categorical variables to be in factor format.

The reason for converting categorical factors:

* *"Categorical variables like MS.Zoning and Neighborhood were converted to factors to ensure that they are treated as discrete levels rather than numeric values. This encoding is crucial for regression models, as it allows the model to correctly interpret and utilize these variables in the analysis."*

**Encode Ordinal Variables:**

data$Overall.Qual <- as.numeric(data$Overall.Qual)

This line converts the Overall.Qual variable to numeric. Since Overall.Qual is an ordinal variable (with a meaningful order), converting it to numeric allows to use it in regression models and other analyses.

* Justification for the conversion of Overall.Qual to numeric:
  + *"The Overall.Qual variable, which represents the overall quality of the house, was converted to numeric to preserve its ordinal nature. This encoding allows the model to capture the inherent ranking in the data, where higher values indicate better quality, which is directly related to higher sale prices."*

1. **Interpretation of the Linear Regression Model**

**Model Summary:**

The linear regression model aims to predict SalePrice based on three predictors: Gr.Liv.Area (Ground Living Area), Overall.Qual (Overall Quality), and Garage.Cars (Number of Garage Cars).

lm(formula = SalePrice ~ Gr.Liv.Area + Overall.Qual + Garage.Cars, data = trainData)

This specifies the formula for the linear regression model and the data used for training.

**Residuals:**

* **Min**: -214523
* **1Q (1st Quartile)**: -23408
* **Median**: -1927
* **3Q (3rd Quartile)**: 19733
* **Max**: 249089

The residuals represent the differences between the observed and predicted sale prices. The median residual is close to zero, indicating that the model's predictions are generally accurate, but there are some large residuals (both positive and negative), suggesting the presence of outliers.

**Coefficients:**

| Predictor | Estimate | Std. Error | t value | Pr(>|t|) | |---------------|------------|------------|---------|----------| | (Intercept) | -105400 | 3524 | -29.91 | < 2e-16 | | Gr.Liv.Area | 55.58 | 1.955 | 28.43 | < 2e-16 | | Overall.Qual | 27730 | 754 | 36.78 | < 2e-16 | | Garage.Cars | 18920 | 1316 | 14.37 | < 2e-16 |

* **Intercept**: The estimated intercept is -105,400, which is the expected sale price when all predictors are zero. This value is not practically meaningful but is necessary for the regression equation.
* **Gr.Liv.Area**: For each additional square foot of ground living area, the sale price increases by approximately $55.58, holding other variables constant.
* **Overall.Qual**: For each one-unit increase in overall quality, the sale price increases by approximately $27,730, holding other variables constant.
* **Garage.Cars**: For each additional garage car space, the sale price increases by approximately $18,920, holding other variables constant.

All predictors have very low p-values (< 2e-16), indicating that they are statistically significant.

**Model Fit:**

* **Residual standard error**: 37320 on 2337 degrees of freedom
* **Multiple R-squared**: 0.7722
* **Adjusted R-squared**: 0.7719
* **F-statistic**: 2641 on 3 and 2337 DF, p-value: < 2.2e-16
* **Residual Standard Error**: The average distance that the observed values fall from the regression line is approximately $37,320.
* **Multiple R-squared**: 0.7722 indicates that approximately 77.22% of the variability in SalePrice is explained by the model.
* **Adjusted R-squared**: 0.7719 adjusts the R-squared value for the number of predictors in the model, providing a more accurate measure of model fit.
* **F-statistic**: 2641 with a p-value < 2.2e-16 indicates that the model is statistically significant.

**Conclusion:**

The model shows that Gr.Liv.Area, Overall.Qual, and Garage.Cars are significant predictors of SalePrice. The high R-squared value suggests that the model explains a substantial portion of the variability in sale prices. However, the presence of large residuals indicates that there may be other factors influencing sale prices that are not included in the model.

1. **Diagnostic Plots for Linear Model:**

par(mfrow = c(2, 2))

plot(lm\_model)

This code sets up a 2x2 plotting layout and generates the four diagnostic plots for your linear regression model.

A group of graphs showing different values

Description automatically generated

* 1. **Residuals vs Fitted Purpose**: This plot checks for non-linearity, unequal error variance (heteroscedasticity), and outliers.
* "The Residuals vs Fitted plot revealed a slight U-shaped pattern, indicating potential non-linearity in the relationship between the predictors and the target variable. This suggests that the linear model may not fully capture the underlying relationships. To address this, we could explore polynomial regression or include interaction terms to better model non-linear effects."

**Observation**:

* + If the residuals are randomly scattered around the horizontal line (y=0), it suggests that the linear model is appropriate.
  + In this case, there seems to be a slight curve, indicating potential non-linearity in the data. Additionally, some extreme points are visible, which might be outliers.

**Action**: Consider transforming variables or including higher-order terms (e.g., quadratic terms) to address non-linearity.

* 1. **Normal Q-Q Plot Purpose**: This plot assesses whether the residuals follow a normal distribution.

**Observation**:

*"The Q-Q plot showed deviations from normality at the tails, indicating that the residuals are not perfectly normally distributed. This non-normality could affect the validity of hypothesis tests and confidence intervals. Applying a log transformation to SalePrice or using robust regression techniques could help mitigate this issue."*

Most of the residuals follow the diagonal line, but deviations occur at the tails, indicating that the residuals are not perfectly normal.

**Action**: Consider data transformations (e.g., log, square root) or robust regression techniques if the deviations are severe.

* 1. **Scale-Location (Spread-Location) Purpose**: This plot checks for homoscedasticity (equal variance of residuals). **Observation**:

"The Scale-Location plot revealed an upward trend in the spread of residuals, indicating heteroscedasticity. This non-constant variance can lead to inefficient estimates and biased standard errors. To address this, we could apply a weighted least squares approach or transform the dependent variable using a log or square root transformation

A flat line with randomly scattered points indicates equal variance. Here, the upward trend suggests heteroscedasticity (variance of residuals increases with fitted values).

**Action**: Use weighted least squares or transform the dependent variable (e.g., log transformation).

* 1. **Residuals vs Leverage Purpose**: This plot identifies influential observations that have a significant impact on the model. **Observation**: Points outside the red Cook’s distance lines are influential. These points should be investigated further to ensure they are not overly distorting the model.

**Action**: Consider removing or analyzing these influential points separately.

* 1. **Residuals vs Fitted plot to check for homoscedasticity**

ggplot(lm\_model, aes(x = .fitted, y = .resid)) +geom\_point(alpha = 0.5) + geom\_smooth(method = "loess", color = "**orange**") + labs(title = "Residuals vs Fitted", x = "Fitted Values", y = "Residuals")

A graph showing a dotted line

Description automatically generated

**What This Plot Shows:**

* There is a **non-linear trend** in the residuals, as evidenced by the curvature in the orange LOESS smooth line.
* The residuals are not evenly scattered; instead, they show a "U-shaped" pattern. This suggests:
  + The model is **not properly capturing the non-linear relationships** between the predictors and the target variable.
  + There may be omitted variables or interactions that are important for explaining the response variable.
* Additionally, the spread of residuals appears to increase slightly for higher fitted values, which could indicate **heteroscedasticity** (non-constant variance).
  1. **Q-Q plot for normality of residuals** ggplot(lm\_model, aes(sample = .resid)) +geom\_qq() + geom\_qq\_line(color = "brown") + labs(title = "Q-Q Plot of Residuals")

A graph of a graph

Description automatically generated

**What This Plot Shows:**

* The points deviate significantly from the line at both ends (tails), which suggests non-normality in the residuals.
* Specifically:
  + Left Tail (Negative Residuals): The points deviate downward, suggesting an excess of extreme negative residuals compared to a normal distribution.
  + Right Tail (Positive Residuals): The points deviate upward, indicating an excess of extreme positive residuals.
  1. **Pairplot equivalent (scatterplots with histograms)** top\_features <- c("SalePrice", "Overall.Qual", "Gr.Liv.Area", "Garage.Cars", "Garage.Area", "Total.Bsmt.SF")pairs(df\_cleaned[, top\_features], main = "Scatterplot Matrix of Top Features")

A screenshot of a graph

Description automatically generated

1. SalePrice vs Features:
   * Overall.Qual, Gr.Liv.Area, Garage.Cars, Garage.Area, and Total.Bsmt.SF show strong positive relationships with SalePrice. Higher values generally correspond to higher prices.
2. Feature Interrelationships:
   * Gr.Liv.Area, Garage.Area, and Total.Bsmt.SF are positively correlated, indicating larger homes tend to have bigger garages and basements.
   * Garage.Cars and Garage.Area are highly correlated.
3. Distributions:
   * Histograms on the diagonal show skewed distributions, with most values concentrated in specific ranges for features like Garage.Cars and Overall.Qual.

This analysis highlights key predictors and multicollinearity risks.

1. Train Random Forest Model

# Load additional libraries library(randomForest) # Train the random forest model set.seed(123) rf\_model <- randomForest(SalePrice ~ ., data = trainData, importance = TRUE, ntree = 500) print(rf\_model) importance(rf\_model) varImpPlot(rf\_model)

A close-up of a computer code

Description automatically generated

1. **Type of Random Forest**: Regression - This indicates that the model is used for predicting continuous values (in this case, SalePrice).
2. **Number of Trees**: 500 - The model consists of 500 decision trees, which helps improve the robustness and accuracy of the predictions.
3. **Number of Variables Tried at Each Split**: 27 - At each split in the trees, 27 variables are considered, which is a common approach to ensure a good balance between bias and variance.
4. **Mean of Squared Residuals**: 558,740,332 - This is the average of the squared differences between the observed and predicted values. Lower values indicate better model performance.
5. **% Variance Explained**: 90.84% - This indicates that the model explains 90.84% of the variance in the SalePrice variable, which suggests a very strong predictive power.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **%IncMSE**: This stands for the percentage increase in Mean Squared Error (MSE) when the feature is randomly permuted. Higher values indicate that the feature is more important for the model's predictions.
2. **IncNodePurity**: This measures the total decrease in node impurities (measured by residual sum of squares) from splitting on the feature, averaged over all trees. Higher values indicate greater importance.

Here's a brief interpretation of some of the features:

* **Gr.Liv.Area**: With a %IncMSE of 49.23 and an IncNodePurity of 1.38e+12, this feature is highly important for predicting SalePrice.
* **Neighborhood**: This feature has a %IncMSE of 29.58 and an IncNodePurity of 1.76e+12, indicating significant importance.
* **Overall.Qual**: With a %IncMSE of 25.39 and an IncNodePurity of 3.29e+12, this feature is also very important.
* **Total.Bsmt.SF**: This feature has a %IncMSE of 23.34 and an IncNodePurity of 6.26e+11, showing its importance in the model.
* **X1st.Flr.SF**: With a %IncMSE of 27.13 and an IncNodePurity of 5.27e+11, this feature is crucial for the model's predictions.
* "The random forest model identified Gr.Liv.Area, Neighborhood, and Overall.Qual as the most important predictors of SalePrice. This aligns with real-world intuition, as larger living areas, better neighborhoods, and higher-quality homes are typically associated with higher prices. The model's ability to capture non-linear relationships and interactions between variables makes it a powerful tool for predicting housing prices."

Visual representation:

A graph of a function

Description automatically generated with medium confidence

1. **Construct the best-fit line equation**

best\_fit\_equation <- paste("SalePrice =", round(intercept, 2), "+", round(slope, 2), "\* Gr.Liv.Area")

cat("Best-fit line equation: ", best\_fit\_equation, "\n")

The best-fit line equation provided is:

**SalePrice = 8303.74 + 114.5 × Gr.Liv.Area**

This equation represents the relationship between the **SalePrice** (dependent variable) and **Gr.Liv.Area** (independent variable), which was likely determined through linear regression.

**Equation in Practice:**

* If a home has, for example, **2000 square feet** of ground living area: SalePrice=8303.74+114.5×2000=8303.74+229000=237,303.74
* The predicted sale price of the house would be approximately **$237,303.74**.

1. **Visualize the comparison**

ggplot(model\_comparison, aes(x = Model, y = RMSE, fill = Model)) +

geom\_bar(stat = "identity") +

theme\_minimal() +

labs(title = "Model Comparison: RMSE",

x = "Model",

y = "Root Mean Squared Error (RMSE)")

Model RMSE

1 Linear Regression = 33661.83

2 Decision Tree = 37004.97

3 Random Forest = 29704.99

A graph of different colored squares

Description automatically generated

1. **Overall Conclusion:**

The analysis revealed that **Gr.Liv.Area**, **Overall.Qual**, and **Neighborhood** are the strongest predictors of housing prices. These insights align with real-world expectations, where the size, quality, and location are critical determinants of property value.

* The **random forest model** provided a more comprehensive understanding of the factors influencing sale price, capturing non-linear interactions and complex relationships.
* The **linear regression model** offered a clear and interpretable relationship between **Gr.Liv.Area** and **SalePrice**.
* **"The random forest model outperformed the linear regression model, with a lower RMSE (29,704.99 vs. 33,661.83) and a higher percentage of variance explained (90.84%). This improvement is due to the random forest's ability to capture complex, non-linear relationships and interactions between variables, which the linear model cannot fully account for."**

Key Findings of Analysis:

* *"The analysis revealed that Gr.Liv.Area, Overall.Qual, and Neighborhood are the strongest predictors of housing prices. These findings align with real-world expectations, as larger living areas, higher-quality homes, and desirable neighborhoods are typically associated with higher property values. The random forest model provided a more comprehensive understanding of these relationships, capturing non-linear interactions and complex patterns that the linear regression model could not fully explain."*

Model Recommendations:

* *"To further improve the model, we could explore additional techniques such as gradient boosting (e.g., XGBoost or LightGBM) or neural networks, which may offer even better predictive performance. Additionally, incorporating more granular data, such as school district ratings or proximity to amenities, could provide further insights into the factors influencing housing prices."*

Practical Implementations:

* *"These findings have important implications for real estate professionals, homebuyers, and policymakers. For example, real estate agents can use these insights to price homes more accurately, while homebuyers can make more informed decisions about which features to prioritize when purchasing a property. Policymakers could also use this analysis to identify areas where housing affordability is a concern and develop targeted interventions to address these issues."*

1. References

Akinwande, M. O., Dikko, H. G., & Samson, A. (2015). Variance inflation factor: As a condition for the inclusion of suppressor Variable(s) in regression analysis. Open Journal of Statistics, 05(07), 754-767. https://doi.org/10.4236/ojs.2015.57075 Hawkins, D. M. (1980). Outliers from the linear model. Identification of Outliers, 85- 103. https://doi.org/10.1007/978-94-015-3994-4\_7 Politis, D. N. (2015). Model-based prediction in regression. Model-Free Prediction and Regression, 33-56. <https://doi.org/10.1007/978-3-319-21347-7_3>

**Appendix**

**# Load necessary libraries**

install.packages("caret")

install.packages("glmnet")

install.packages("gridExtra")

library(tidyverse)

library(ggplot2)

library(caret)

library(dplyr)

library(corrplot)

library(glmnet)

library(gridExtra)

library(randomForest)

**# Step 1: Load Data**

**# Set file path and load the Ames Housing dataset**

data <- read.csv("C:/Users/ravin/Downloads/AmesHousing.csv")

**# Step 2: Initial Exploration**

**# View the structure of the dataset**

str(data)

**# Check for missing values**

missing\_values <- colSums(is.na(data))

missing\_values[missing\_values > 0]

**# Summary statistics for continuous variables**

summary(data)

**# Step 3: Data Cleaning**

**# Handle missing values by imputing or removing**

**# Example: Impute Lot Frontage with median value**

if("Lot.Frontage" %in% colnames(data)) {

data$Lot.Frontage[is.na(data$Lot.Frontage)] <- median(data$Lot.Frontage, na.rm = TRUE)

}

**# Remove houses with more than 4000 sq ft (as per documentation recommendation)**

data <- data[data$Gr.Liv.Area <= 4000, ]

**# Step 4: Exploratory Data Analysis**

**# Distribution of the target variable (SalePrice)**

ggplot(data, aes(x = SalePrice)) +

geom\_histogram(binwidth = 10000, fill = "pink", color = "black") +

labs(title = "Distribution of Sale Prices", x = "Sale Price", y = "Count")

**# Correlation heatmap of numerical variables**

**# Select numerical variables**

***num\_vars <- data %>% select\_if(is.numeric)***

**# Calculate the correlation matrix**

corr\_matrix <- cor(num\_vars, use = "complete.obs")

**# Plot the correlation heatmap**

corrplot(corr\_matrix, method = "color", tl.cex = 0.6)

**# Scatter plot for Gr Liv Area vs SalePrice**

ggplot(data, aes(x = Gr.Liv.Area, y = SalePrice)) +

geom\_point(alpha = 0.5) +

geom\_smooth(method = "lm", fill = "black", color = "orange") +

labs(title = "Gr Liv Area vs Sale Price", x = "Ground Living Area", y = "Sale Price")

**# Step 5: Feature Engineering**

**# Convert categorical variables to factors**

data <- data %>% mutate(across(where(is.character), as.factor))

**# Encode ordinal variables if needed (example below for OverallQual)**

data$Overall.Qual <- as.numeric(data$Overall.Qual)

**# Step 6: Linear Regression Analysis**

**# Split data into training and test sets**

set.seed(123)

trainIndex <- createDataPartition(data$SalePrice, p = 0.8, list = FALSE)

trainData <- data[trainIndex, ]

testData <- data[-trainIndex, ]

**# Build a simple linear regression model**

lm\_model <- lm(SalePrice ~ Gr.Liv.Area + Overall.Qual + Garage.Cars, data = trainData)

summary(lm\_model)

**# Diagnostic Plots for Linear Model**

par(mfrow = c(2, 2))

plot(lm\_model)

**# Explaination on the two graphs.**

**# Residuals vs Fitted plot to check for homoscedasticity**

ggplot(lm\_model, aes(x = .fitted, y = .resid)) +

geom\_point(alpha = 0.5) +

geom\_smooth(method = "loess", color = "orange") +

labs(title = "Residuals vs Fitted", x = "Fitted Values", y = "Residuals")

**# Q-Q plot for normality of residuals**

ggplot(lm\_model, aes(sample = .resid)) +

geom\_qq() +

geom\_qq\_line(color = "brown") +

labs(title = "Q-Q Plot of Residuals")

**# Pairplot equivalent (scatterplots with histograms)**

top\_features <- c("SalePrice", "Overall.Qual", "Gr.Liv.Area", "Garage.Cars", "Garage.Area", "Total.Bsmt.SF")

pairs(df\_cleaned[, top\_features], main = "Scatterplot Matrix of Top Features")

**# Load additional libraries**

library(randomForest)

**# Train the random forest model**

set.seed(123)

rf\_model <- randomForest(SalePrice ~ ., data = trainData, importance = TRUE, ntree = 500)

print(rf\_model)

importance(rf\_model)

varImpPlot(rf\_model)

**# Linear Regression Model**

lm\_model <- lm(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Cars + Garage.Area + Total.Bsmt.SF, data = trainData)

lm\_pred <- predict(lm\_model, testData)

lm\_rmse <- sqrt(mean((lm\_pred - testData$SalePrice)^2))

**# Install the rpart package**

install.packages("rpart")

**# Load the rpart package**

library(rpart)

**# Decision Tree Model**

tree\_model <- rpart(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Cars + Garage.Area + Total.Bsmt.SF, data = trainData, method = "anova")

tree\_pred <- predict(tree\_model, testData)

tree\_rmse <- sqrt(mean((tree\_pred - testData$SalePrice)^2))

**# Install the randomForest package if you haven't already**

install.packages("randomForest")

**# Load the randomForest package**

**#library(randomForest)**

**# Random Forest Model**

**#rf\_model <- randomForest(SalePrice ~ Overall.Qual + Gr.Liv.Area + Garage.Cars + Garage.Area + Total.Bsmt.SF, data = trainData, ntree = 500)**

**#rf\_pred <- predict(rf\_model, testData)**

**#rf\_rmse <- sqrt(mean((rf\_pred - testData$SalePrice)^2))**

**# Build a simple linear regression model (only for demonstration of best-fit equation)**

simple\_lm\_model <- lm(SalePrice ~ Gr.Liv.Area, data = trainData)

**# Extract coefficients**

intercept <- coef(simple\_lm\_model)[1]

slope <- coef(simple\_lm\_model)[2]

**# Construct the best-fit line equation**

best\_fit\_equation <- paste("SalePrice =", round(intercept, 2), "+", round(slope, 2), "\* Gr.Liv.Area")

cat("Best-fit line equation: ", best\_fit\_equation, "\n")

**# Plot the data and the best-fit line**

ggplot(trainData, aes(x = Gr.Liv.Area, y = SalePrice)) +

geom\_point(alpha = 0.5, color = "lightsteelblue") +

geom\_smooth(method = "lm", formula = y ~ x, color = "pink") +

labs(title = "Best Fit Line for SalePrice vs Gr.Liv.Area",

subtitle = best\_fit\_equation,

x = "Ground Living Area",

y = "Sale Price") +

theme\_minimal()

print(model\_comparison)

**# Visualize the comparison**

ggplot(model\_comparison, aes(x = Model, y = RMSE, fill = Model)) +

geom\_bar(stat = "identity") +

theme\_minimal() +

labs(title = "Model Comparison: RMSE",

x = "Model",

y = "Root Mean Squared Error (RMSE)")