MSDS 6371 Project

Jason Yoon and Andrew Peng

1.Introduction

As per the request from Century 21 Ames, we conducted an analysis on approximately 3,000 homes in Ames, Iowa, considering around 80 categorical variables that characterize residential properties. The objective was to develop and validate predictive models to estimate home prices.

This report initiates with an examination of the impact of Gross Living Area on Sales Price within three specific neighborhoods in Ames. The approach involves establishing linear relationships between variables, constructing a model, and refining it for enhanced accuracy.

Furthermore, employing a diverse set of variables, we constructed predictive models for sales prices and subsequently assessed the precision of each model to identify the most suitable one. Four distinct variable selection methods were applied to determine the significance of each variable, and these were then utilized in the construction of the four models.

2.Data Description

The dataset utilized for this analysis was generously provided by Dean De Cock via Kaggle (https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data). This dataset encapsulates the transactions of individual homes in Ames, Iowa, spanning the years 2006 to 2010. The data is partitioned into two sets: a training set employed for constructing our predictive models and a test set used to assess the models' performance. The construction of our predictive models is based on 1,460 observations and specific variables selected from the 79 available in the training dataset. Subsequently, we evaluate the models using additional data from 1,458 homes in the Ames area to predict sales prices and scrutinize their accuracy.

3.Analysis Question 1:

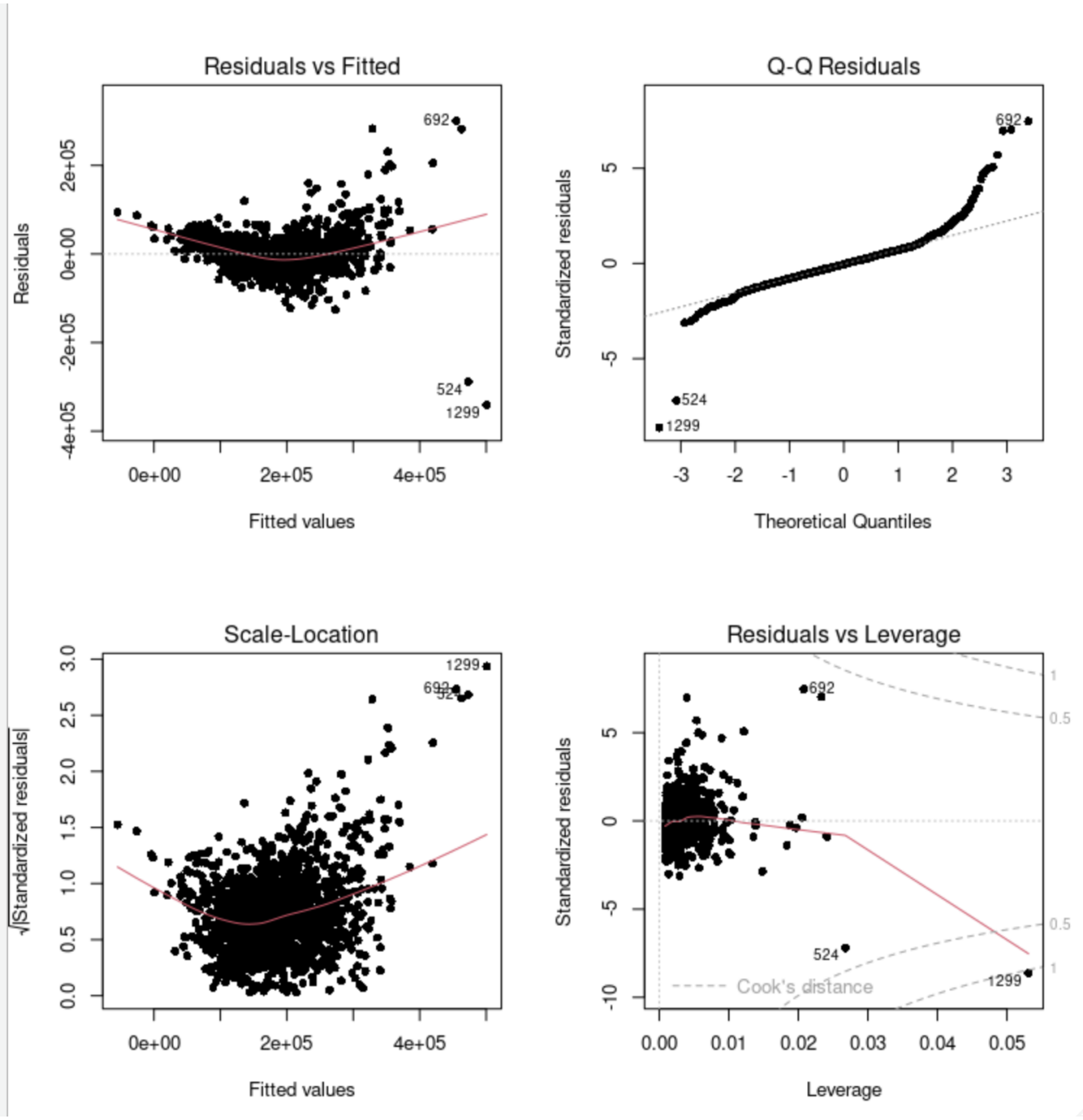
3.1 Restatement of Problem

Century 21 Ames exclusively deals with properties situated in the NAmes, Edwards, and BrkSide neighborhoods. The objective is to examine the correlation between the sales price of homes and the square footage per 100 sq ft. of living area (GrLivArea). Additionally, the study

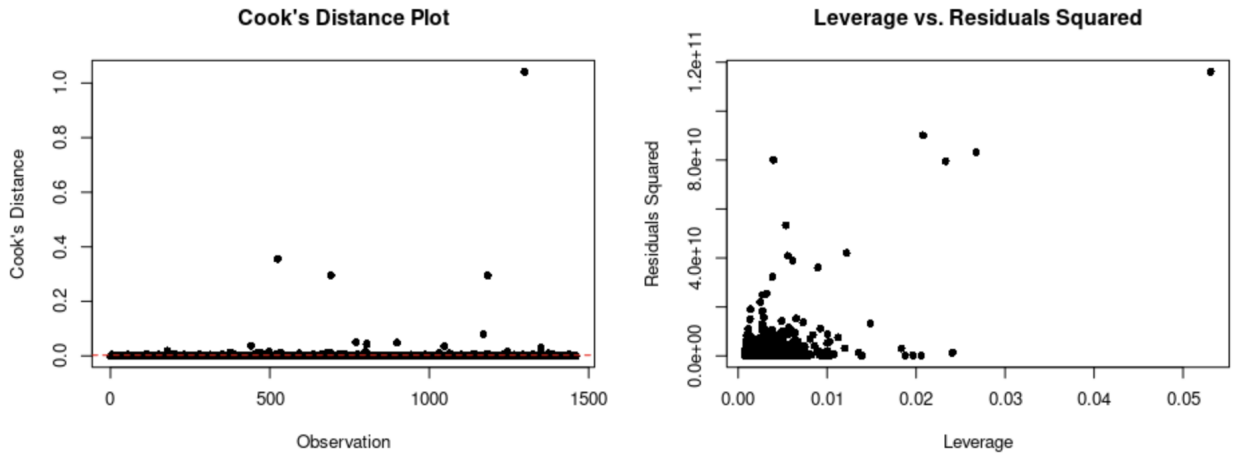
aims to ascertain whether this relationship is influenced by the specific neighborhood in which the house is situated.

3.2 Checking Assumptions

Residual Plots



Influential point analysis (Cook’s D and Leverage)



Under the assumption of independent observations, our initial exploration focuses on determining the existence of a linear relationship between the primary explanatory variable, GrLivArea, and Sales Price. This addition stems from subtle indications in the plot below that suggest a potential influence of the neighborhood on GrLivArea. This aligns with logical reasoning, where a wealthier neighborhood is likely to feature larger homes, consequently impacting SalePrice. With the foundational assumptions considered, our ensuing model can be executed, allowing for analysis and necessary adjustments to adhere to the underlying assumptions.

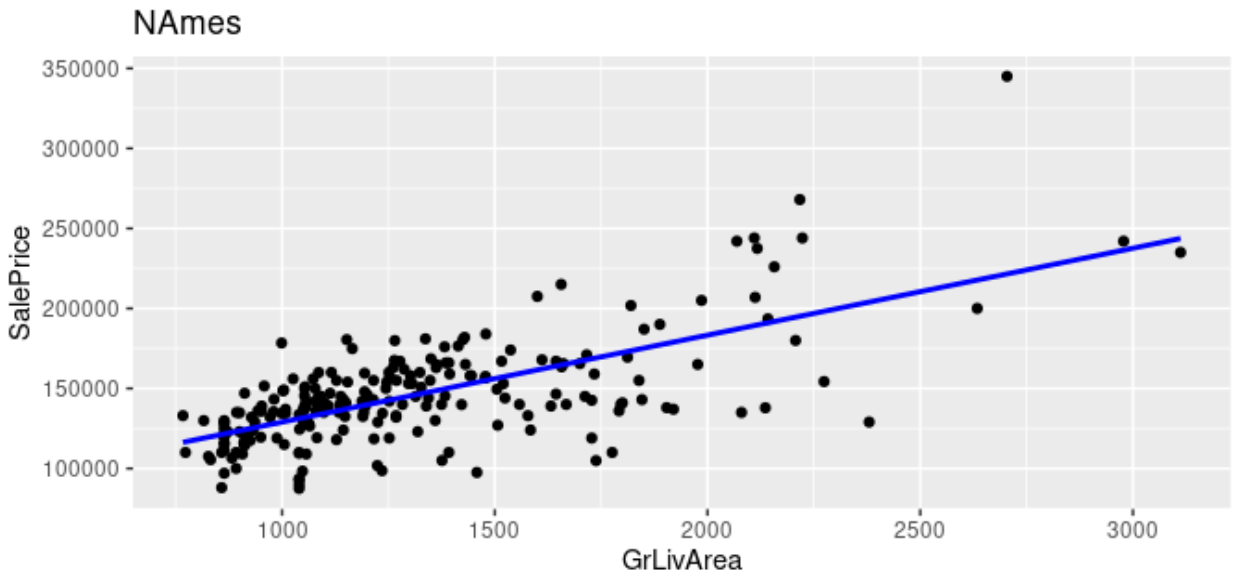
3.3 Build and Fit the Model

The initial model, illustrated plot below, affirms the presence of outliers. In its current state, the model yields an adjusted R-Square of 0.44, indicating a reasonable fit. However, the model's performance is anticipated to enhance upon the removal of outliers. Examining the plots, there is no prominent indication of significant trends or evidence contradicting a normal distribution with constant variance. It is important to note that we already assume the independence of observations in our analysis.

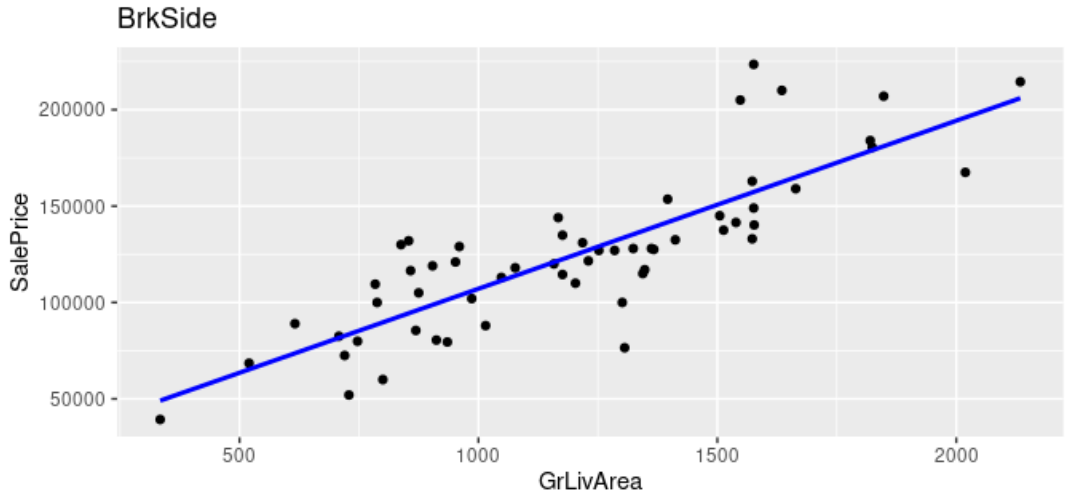
Model Equation

SalePrice = beta0 + beta1*GrLivArea + beta2*BrkSide + beta3*Edwards + beta4*GrLivArea*BrkSide + beta5*GrLivArea*Edwards

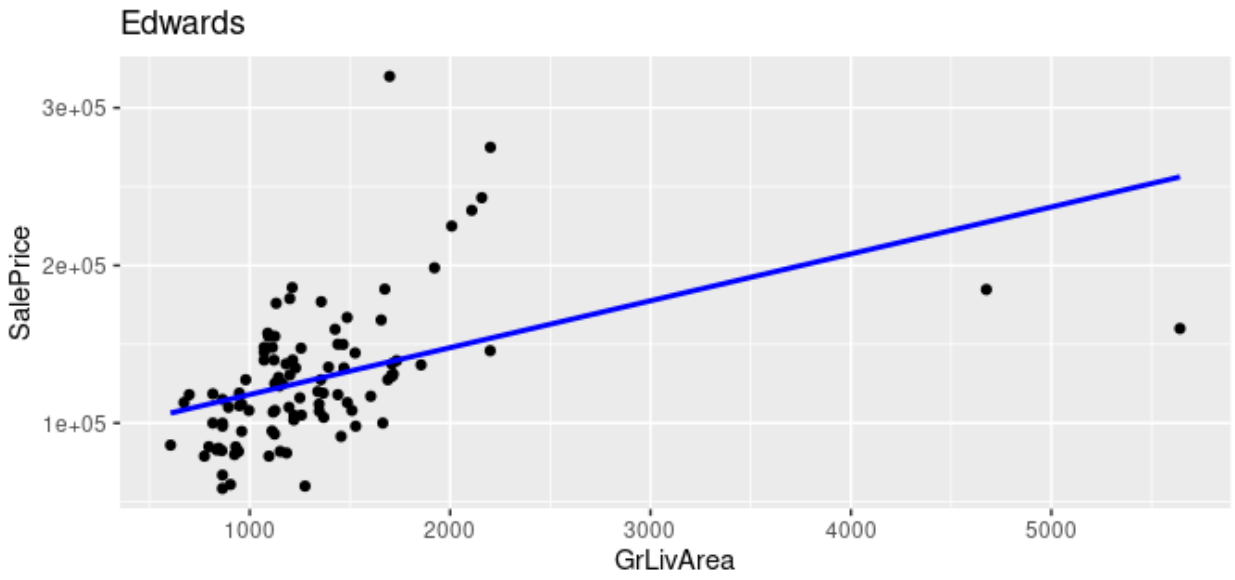
3.4 Comparing Competing Models



· For a home in NAmes, with BrkSide and Edwards held constant, a 100 sqft. increase in gross living area is linked to a mean rise of $4,966 in sale price. The slop of the regression model is 32.641 with p-value of 5.99e-15



· For a home in BrkSide, with NAmes and Edwards held constant, a 100 sqft. increase in gross living area is associated with a mean increase of $8,417 in sale price. The slop of the regression model is 62.784 with p-value of 6.86e-10 that is statistically significant.



· For a home in Edwards, with NAmes and BrkSide held constant, a 100 sqft. increase in gross living area is connected to a mean increase of $4,625 in sale price. The slop of the regression model is 7.222 with p-value of 0.303 which is bigger than 0.05. There are two outliers in the scatter plot above and this is well explained the cook's D plot before.

3.5 Conclusion

Analyzing the dollar increase per 100 sqft., one can deduce that homes in BrkSide are likely to have a higher cost/increase compared to those in NAmes or Edwards. However, it's noteworthy that the mean home sale price appears to be highest in Edwards. It is crucial to acknowledge that, being an observational study, causal inference cannot be ascribed to any parameter concerning sale price.

4.Analysis 2: Predictive model

4.1 Restatement of Problem：

Century 21 Ames would like to find the factor that has the most impact on the sales prices of homes in all of Ames, Iowa. The objective is to build predictive models for the sales prices of homes in Ames, Iowa. This involves developing a simple linear regression model and multiple linear regression models to analyze the relationship between the sale price and various features of the houses.

4.2 Data Preparation and Cleaning:

To utilize the dataset for linear regression, we handled missing values, removed unnecessary columns, checked outliers and transformed categorical values to and created data partitions.

4.3 Model Selection:

- Simple Linear Regression Model: (SalePrice~GrLivArea)
- Multiple Linear Regression Model: (SalePrice~GrLivArea + FullBath)
- Additional Multiple Linear Regression Model: (SalePrice~GrLivArea + FullBath+YearRemodAdd)
- Custom Multiple Linear Regression Model:(SalePrice ~ GrLivArea + FullBath + YearRemodAdd + TotRmsAbvGrd)

Note: TotRmsAbvGrd is customs created feature to represent Total Bathrooms = FullBath + HalfBath

4.4 Comparing Linear Regression Competing Models:

Predictive Models	**Adjusted R2**	**CV PRESS**	**Kaggle Score**
Simple Linear Regression	**0.06344**	**6.869E+12**	**0.40786**
Multiple Linear Regression	**0.50038**	**3.6367E+12**	**0.28533**

Additional MLR Model	**0.60567**	**2.8721E+12**	**0.25695**
Custom MLR Models …	**0.61328**	**2.8209E+12**	**0.25693**

4.5 Conclusion:

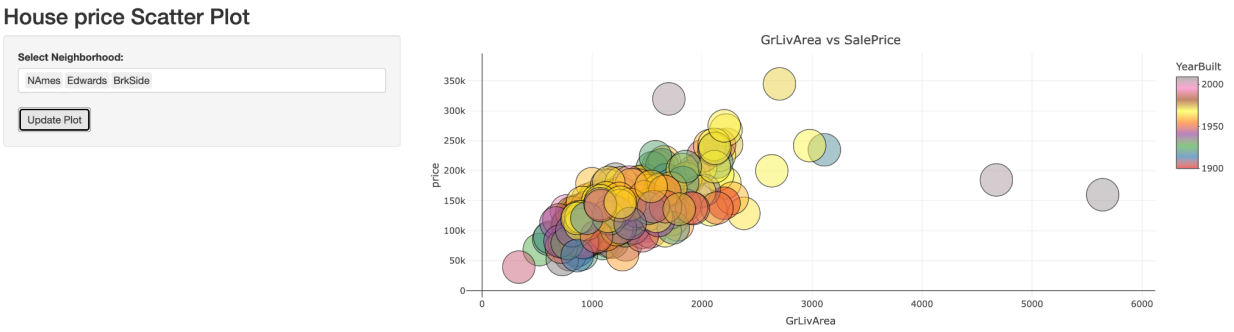
In an effort to produce a highly accurate and repeatable predictive model using linear regression, all explanatory variables were considered with four types of regression models including one single linear regression model and three multiple linear regression models. The final models suggested strictly by the automatic techniques produced lower R2 values, but performed relatively accurate on the Kaggle test set. The final custom model, however, produced best result on the Kaggle test set which suggests custom model is generalizes well to an unseen dataset.

Github page

https://github.com/andrewpeng5/DS6371-Final-Project
https://github.com/gwonchan/DS6371-Final-Project/blob/main/README.md

R Shiny: Price v. Living Area Chart

https://6jv0lq-jason-yoon.shinyapps.io/stats/



Appendix

```
#Analysis 1
library(ggplot2)

options(repr.plot.width = 12, repr.plot.height = 8)

Train = read.csv("train.csv", header = T)

Test = read.csv("test.csv", header = T)

filtered_data <- Train[Train$Neighborhood %in% c("NAmes", "Edwards", "BrkSide"), ]

head(filtered_data)


# Make sure to replace the column names with the actual names in your dataset
model <- lm(SalePrice ~ OverallQual + GrLivArea + GarageCars, data = filtered_data)

# Summary of the linear regression model
summary(model)

# Load necessary packages
library(ggplot2)
library(car)

# Assuming you have already fit the linear regression model (replace 'model' with your actual
model)
# Example: model <- lm(SalePrice ~ OverallQual + GrLivArea + GarageCars, data = Train)

# Residual plots
par(mfrow=c(2,2))
plot(model, pch = 16)
# Assuming you have already fit the linear regression model (replace 'model' with your actual
model)
# Example: model <- lm(SalePrice ~ OverallQual + GrLivArea + GarageCars, data = Train)

# Calculate Cook's Distance directly on the model
cooksd <- cooks.distance(model)

# Plot Cook's Distance
```

```
plot(cooksd, pch = 16, main = "Cook's Distance Plot", ylab = "Cook's Distance", xlab =
"Observation")
abline(h = 4/(length(cooksd) - length(coefficients(model))), col = "red", lty = 2)

# Identify influential points based on Cook's D
influential_points <- which(cooksd > 4/(length(cooksd) - length(coefficients(model))))
cat("Influential Points (based on Cook's D):", influential_points, "\n")

# Plot Leverage vs. Residuals squared
residuals_squared <- residuals(model)^2
plot(hatvalues(model), residuals_squared, pch = 16, main = "Leverage vs. Residuals Squared",
     ylab = "Residuals Squared", xlab = "Leverage")

# Identify high leverage points
high_leverage_points <- which(hatvalues(model) > 2 * mean(hatvalues(model)))
cat("High Leverage Points:", high_leverage_points, "\n")
```

Analysis 2: Sale Price

```
# Required Libraries
library(Metrics)
library(nortest)
library(dplyr)
library(tidyverse)
library(caret)
library(Metrics)
library(caTools)
library(e1071)
library(glmnet)
library(randomForest)
library(xgboost)
library(data.table)
library(lubridate)
library(carData)
library(car)
library(lattice)
library(lmtest)
library(zoo)
library(ggplot2)
library(corrplot)
library(knitr)
library(kableExtra)

# Load the data
train_data <- read.csv("C:/Users/andre/Documents/SMU Information/DS 6371/Unit 14/train.csv")
test_data <- read.csv("C:/Users/andre/Documents/SMU Information/DS 6371/Unit 14/test.csv")

# Handling Missing Values
# Imputing missing values for numerical columns with median and categorical columns with
mode
num_cols <- sapply(train_data, is.numeric)
cat_cols <- sapply(train_data, is.character)
train_data[num_cols] <- lapply(train_data[num_cols], function(x) ifelse(is.na(x), median(x, na.rm
= TRUE), x))
train_data[cat_cols] <- lapply(train_data[cat_cols], function(x) ifelse(is.na(x),
names(sort(table(x), decreasing = TRUE))[1], x))

# Transforming Variables
# Converting categorical variables to factors
train_data[cat_cols] <- lapply(train_data[cat_cols], as.factor)
```

```
# Removing Unnecessary Columns
# Dropping 'Id' column
train_data <- train_data %>% select(-Id)

# Checking and Handling Outliers
# Example with 'GrLivArea' for Multiple Linear Regression
Q1 <- quantile(train_data$GrLivArea, 0.25)
Q3 <- quantile(train_data$GrLivArea, 0.75)
IQR <- Q3 - Q1
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
train_data <- train_data %>%
  filter(GrLivArea >= lower_bound & GrLivArea <= upper_bound)

# Preparing dataset for Simple Linear Regression
# Selecting a single explanatory variable, e.g., 'LotArea'
slr_data <- train_data %>%
  select(SalePrice, LotArea)

# Preparing dataset for Multiple Linear Regression
# Including 'GrLivArea' and 'FullBath' as explanatory variables
mlr_data <- train_data %>%
  select(SalePrice, GrLivArea, FullBath)

# Feature Engineering for additional Multiple Linear Regression model
# Example: Creating a new feature 'TotalBathrooms'
train_data$TotalBathrooms <- train_data$FullBath + train_data$HalfBath

# Preparing dataset for Additional Multiple Linear Regression
# Selecting explanatory variables for the model
additional_mlr_data <- train_data %>%
  select(SalePrice, GrLivArea, FullBath, TotalBathrooms)


# Building a Simple Linear Regression Model
# Using 'LotArea' as the explanatory variable
slr_model <- lm(SalePrice ~ LotArea, data = train_data)

# Building a Multiple Linear Regression Model with 'GrLivArea' and 'FullBath'
mlr_model_1 <- lm(SalePrice ~ GrLivArea + FullBath, data = train_data)

# Building an Additional Multiple Linear Regression Model
```

```
# Building a Multiple Linear Regression Model with 'GrLivArea' and 'FullBath' and
"YearRemodAdd"
mlr_model_2 <- lm(SalePrice ~ GrLivArea + FullBath + YearRemodAdd, data = train_data)

# Building an Additional Multiple Linear Regression Model
# Selecting additional explanatory variables, e.g., 'YearRemodAdd' and 'TotalRooms'
mlr_model_3 <- lm(SalePrice ~ GrLivArea + FullBath + YearRemodAdd + TotRmsAbvGrd, data
= train_data)

# Making Predictions on Test Data
# For Simple Linear Regression
predictions_slr <- predict(slr_model, newdata = test_data)

# For Multiple Linear Regression Model 1
predictions_mlr_1 <- predict(mlr_model_1, newdata = test_data)

# For Additional Multiple Linear Regression Model
predictions_mlr_2 <- predict(mlr_model_2, newdata = test_data)

# For Additional Multiple Linear Regression Model
predictions_mlr_3 <- predict(mlr_model_3, newdata = test_data)

# The predictions can now be used for further analysis or evaluation
}

# Adjusted R² for each model
adj_r2_slr <- summary(slr_model)$adj.r.squared
adj_r2_mlr_1 <- summary(mlr_model_1)$adj.r.squared
adj_r2_mlr_2 <- summary(mlr_model_2)$adj.r.squared
adj_r2_mlr_3 <- summary(mlr_model_3)$adj.r.squared

# Function to calculate CV Press
cv_press <- function(model, data) {
  residuals <- resid(model)
  leverage <- hatvalues(model)
  press <- sum((residuals / (1 - leverage))^2)
  return(press)
}
# CV Press for each model
cv_press_slr <- cv_press(slr_model, train_data)
cv_press_mlr_1 <- cv_press(mlr_model_1, train_data)
cv_press_mlr_2 <- cv_press(mlr_model_2, train_data)
```

#change to submission formate

res1 = data.table(Id = test_data$Id, SalePrice = predictions_slr$V1)
res2 = data.table(Id = test_data$Id, SalePrice = predictions_mlr_1$V1)
res3 = data.table(Id = test_data$Id, SalePrice = predictions_mlr_2$V1)
res4 = data.table(Id = test_data$Id, SalePrice = predictions_mlr_3$V1)

#create cvs for submission

write.csv(res1, file = "res1.csv",row.names = F)
write.csv(res2, file = "res2.csv",row.names = F)
write.csv(res3, file = "res3.csv",row.names = F)
write.csv(res4, file = "res4.csv",row.names = F)

#Analysis 2
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
```

```{r include = FALSE}
# LIBRARIES ####
library(tidyverse)
library(reshape2)
library(plotly)
library(ggplot2)
library(scales)
library(pwr)
library(agricolae)
library(huxtable)
library(lawstat)
library(lsmeans)
library(nCDunnett)
library(dplyr)
library(WDI)
library(investr)
library(multcomp)
library(pairwiseCI)
library(DescTools)
library(GGally)
library(car)
library(stats)
library(plotly)
```

```
library(dplyr)
library(shinydashboard)
library(shiny)
library(shinythemes)
library(olsrr)
library(rsconnect)
```

```{r include = FALSE}
# Import Dataset ####
#setwd("C:/Users/LA026LE/OneDrive - Pitney Bowes/MASTER DEGREE/MSDS_6371_Stat_Foundations/KaggleProject/lklewis83.github.io-main/Kaggle_HousePrice")

Train = read.csv("train.csv", header = T)

Test = read.csv("test.csv", header = T)

# Split Data ####
set.seed(4)

TrainObs = sample(seq(1,dim(Train)[1]),round(.75*dim(Train)[1]),replace = FALSE)

TrainingDB = Train[TrainObs,]

TestingDB = Train[-TrainObs,]

# Transform Data ####
TrainingDB <- TrainingDB %>%
  mutate_if(is.character, as.factor)

TrainingDB <- TrainingDB %>%
  mutate_if(is.integer, as.factor)

# Convert SalePrice to numeric | Training DB ####
TrainingDB$SalePrice <- as.numeric(as.character(TrainingDB$SalePrice))

TestingDB <- TestingDB %>%
  mutate_if(is.character, as.factor)

TestingDB <- TestingDB %>%
  mutate_if(is.integer, as.factor)

# Convert SalePrice to numeric | Testing DB ####
```

```
TestingDB$SalePrice <- as.numeric(as.character(TestingDB$SalePrice))
```

```{r include = FALSE}
# FORWARD FIT BEST MODEL | FORW_fit ####
FORW_fit = lm(SalePrice ~
              +YearBuilt
              +Neighborhood
              +BsmtFinSF1
              +TotalBsmtSF
              +GrLivArea
              +GarageArea
              +EnclosedPorch
              , data = TrainingDB)

# Parameter Estimate
summary(FORW_fit)
```

```{r include = FALSE}
# CUSTOM FIT BEST MODEL | Model1_fit####
Model1_fit = lm(SalePrice ~
 + Fireplaces
 + Foundation
 + LotConfig
 + HouseStyle
 + Neighborhood
 + GarageCars
 + LotArea
 + GrLivArea
 + BsmtFullBath, data = TrainingDB)

# Parameter Estimate
summary(Model1_fit)
```