DATA MINING AND MACHINE LEARNING

ASSESSMENT 3

ANTHONY OGUNNA

2413689

An Assignment Submitted in the partial fulfilment for
the award of Master of Science degree in Data Analytics and Technologies

Tutor : Dr. Pradeep Hewage

April 2025

The University of Bolton

Deane Road, Bolton, BL3 5AB

http://www.bolton.ac.uk

Table of Contents

List of figures ……………………………………………………………………………………………………i

Fig. 1: Loading the housing dataset on Rstudio……………………………………………………. 2

Fig. 2: A view of the housing data ………………………………………………………………………. 3

Fig. 3: Reposition the SALE_PRC column to the last column……………………………………4

Fig. 4: Check for correlation with the SALE_PRC …………………………………………………4

Fig. 5: Correlation of SALE_PRC using Heatmap……………………………………………………5

Fig. 6: Statistical test for correlation significance……………………………………………………5

Fig.7: Check for duplicates and null value………………………………………………………………6

Fig. 8: Drop columns with insignificant correlation…………………………………………………7

Fig. 9: Structure of Housing data……………………………………………………………………………7

Fig. 10: Check for outliers……………………………………………………………………………………8

Fig. 11: Split data into training and testing……………………………………………………………10

Fig. 12: Fitting multiple linear regression model on training data and prediction ……10

Fig. 13: Fitting support vector regression linear model on training data and prediction……………………………………………………………………………………………11

Fig. 14: Fitting support vector regression radial model on training data and prediction……………………………………………………………………………………………..11

Fig. 15: Fitting support vector regression polynomial model on training data and prediction…………………………………………………………………………………………………12

Fig.16: Fitting decision tree model on training data and prediction………………………… 12

Fig. 17: Fitting random forest model with n=100 on training data and prediction….13

Fig. 18: Fitting random forest model with n=200 and n=500 on training data…………13

Fig. 19: Evaluation of models…………………………………………………………………………………14

Fig. 20: Graphical Evaluation of models…………………………………………………………………15

Fig. 21: Hyperparameter tuning………………………………………………………………………………15

Fig. 22: Prediction with optimal svr linear model……………………………………………………16

Abstract ……..iii

Keywords: Data mining, Machine Learning models, Modeling techniques, Data science project management, Models Evaluation.

Chapter 1: Introduction ……………………………………………………………………………………1

1.1 Business understanding……………………………………………………………………………………1

1.2 Data understanding……………………………………………………………………………………….1

Chapter 2: Data Preparation …………………………………………………………………………………6

Chapter 3: Modeling …………………………………………………………………………………………...9

3.1 Modeling technique ………………………………………………………………………………………..9

3.2 Machine Learning models ……………………………………………………………………………….9

Chapter 4: Evaluation of Models……………………………………………………………………….14

Chapter 5: Discussion…………………………………………………………………………………………17

Chapter 6: Conclusion ……………………………………………………………………………………...18

References ……………………………………………………………………………………………………….19

Word count : 2133

ABSTRACT

This study focuses on using the regression modeling technique to predict the prices of house in a given location with machine learning predictive algorithms. For effective assessment of the project, the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology is employed to build a comprehensive framework to manage the process from the business understanding phase down to the deployment stage.

CHAPTER 1

INTRODUCTION

1.1 Business understanding

Housing, which is one of the most basic needs for human existence, is at the center of economic firmness, along with food, water, and many other necessities (Zulkifley et al., 2020). House pricing, which deals with the valuation of residential properties, is influenced by factors such as supply and demand, population growth, employment rates, availability of mortgage financing and affordable interest rates, property size and features, economic conditions, government policies, and others. It plays an important role in shaping economic stability and overall societal wellbeing. In addition, an insight into house pricing is essential for buyers, sellers, investors, and policymakers, as it helps to make informed decisions regarding property transactions and market trends (Kang et al., 2021). Furthermore, the value of statistical models and machine learning in the profession of property valuation is a topic of continuous debate among appraisers and real estate market experts (Forys, 2022). The key objective of this project is to analyse and predict house pricing as this is one of the key business problems in real estate business using data-driven approaches with machine learning by developing predictive models and evaluate them to get the best model that meet the business need to help stakeholders make informed decisions with respect to house prices.

1.2 Data understanding

The housing price dataset contains the following columns:

PARCELNO: unique identifier for each property. About 1% appear multiple times.

SALE_PRC: sale price in U.S. dollars ($)

LND_SQFOOT: This is the total land area of the property measured in square feet

TOT_LVG_AREA: This is the total living (floor) area of the property measured in square feet

SPEC_FEAT_VAL: This is the estimated value of special features (e.g., swimming pools) in U.S. dollars ($)

RAIL_DIST: This is the distance from the property to the nearest rail line (an indicator of noise) measured in feet

OCEAN_DIST: This is the distance from the property to the ocean (feet)

WATER_DIST: This is the distance from the property to the nearest body of water eg lake, river etc measured in feet

CNTR_DIST: This is the distance from the property to the central business district measured in feet

SUBCNTR_DI: This is the distance from the property to the nearest subcenter (a secondary commercial hub) measured in feet

HWY_DIST: This is the distance from the property to the nearest highway (an indicator of noise) measured in feet

age: This is the age of the structure in years

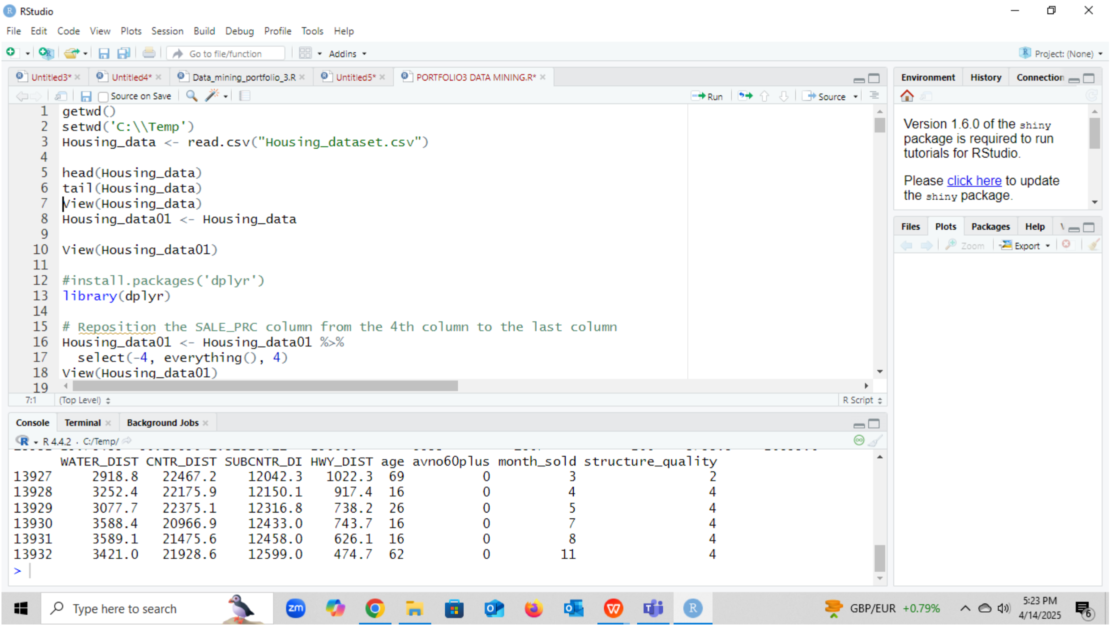
avno60plus: dummy variable for airplane noise exceeding an acceptable level with the threshold yes =1 and no = 0

structure_quality: This is a qualitative measure of the quality of the property on numerical scale

month_sold: This is the month the property was sold in 2023 (1 = jan, 2 = feb, 3 = mar, etc)

LATITUDE: Geographical coordinate of the property

LONGITUDE: Geographical coordinate of the property



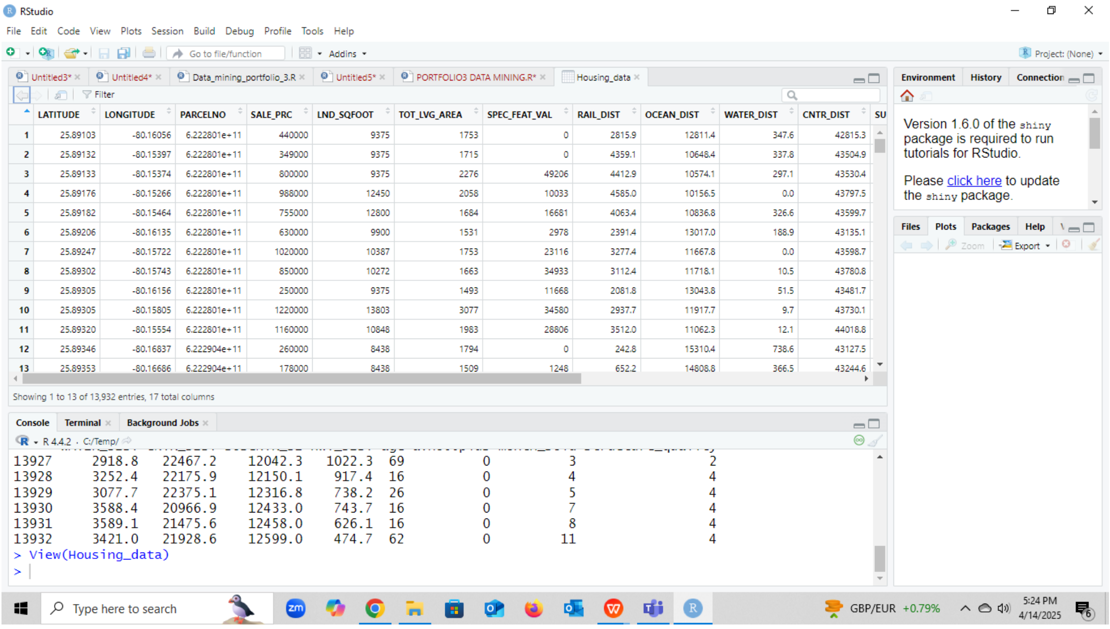
```
getwd()
setwd('C:\\Temp')
Housing_data <- read.csv("Housing_dataset.csv")

head(Housing_data)
tail(Housing_data)
View(Housing_data)
Housing_data01 <- Housing_data

View(Housing_data01)

#install.packages('dplyr')
library(dplyr)

# Reposition the SALE_PRC column from the 4th column to the last column
Housing_data01 <- Housing_data01 %>%
  select(-4, everything(), 4)
View(Housing_data01)
```

```
      WATER_DIST CNTR_DIST SUBCNTR_DI HWY_DIST age avno60plus month_sold structure_quality
13927     2918.8   22467.2    12042.3   1022.3  69          0          3                 2
13928     3252.4   22175.9    12150.1    917.4  16          0          4                 4
13929     3077.7   22375.1    12316.8    738.2  26          0          5                 4
13930     3588.4   20966.9    12433.0    743.7  16          0          7                 4
13931     3589.1   21475.6    12458.0    626.1  16          0          8                 4
13932     3421.0   21928.6    12599.0    474.7  62          0         11                 4
> 
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

Fig. 1: Loading the housing dataset on Rstudio


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled3*
Untitled4*
Data_mining_portfolio_3.R
Untitled5*
PORTFOLIO3 DATA MINING.R*
Housing_data
Filter
Environment History Connection
Files Plots Packages Help
Zoom Export
Console Terminal Background Jobs
R 4.4.2 · C:/Temp/
Type here to search
GBP/EUR +0.79%
5:24 PM
4/14/2025

	LATITUDE	LONGITUDE	PARCELNO	SALE_PRC	LND_SQFOOT	TOT_LVG_AREA	SPEC_FEAT_VAL	RAIL_DIST	OCEAN_DIST	WATER_DIST	CNTR_DIST	SU
1	25.89103	-80.16056	6.222801e+11	440000	9375	1753	0	2815.9	12811.4	347.6	42815.3	
2	25.89132	-80.15397	6.222801e+11	349000	9375	1715	0	4359.1	10648.4	337.8	43504.9	
3	25.89133	-80.15374	6.222801e+11	800000	9375	2276	49206	4412.9	10574.1	297.1	43530.4	
4	25.89176	-80.15266	6.222801e+11	988000	12450	2058	10033	4585.0	10156.5	0.0	43797.5	
5	25.89182	-80.15464	6.222801e+11	755000	12800	1684	16681	4063.4	10836.8	326.6	43599.7	
6	25.89206	-80.16135	6.222801e+11	630000	9900	1531	2978	2391.4	13017.0	188.9	43135.1	
7	25.89247	-80.15722	6.222801e+11	1020000	10387	1753	23116	3277.4	11667.8	0.0	43598.7	
8	25.89302	-80.15743	6.222801e+11	850000	10272	1663	34933	3112.4	11718.1	10.5	43780.8	
9	25.89305	-80.16156	6.222801e+11	250000	9375	1493	11668	2081.8	13043.8	51.5	43481.7	
10	25.89305	-80.15805	6.222801e+11	1220000	13803	3077	34580	2937.7	11917.7	9.7	43730.1	
11	25.89320	-80.15554	6.222801e+11	1160000	10848	1983	28806	3512.0	11062.3	12.1	44018.8	
12	25.89346	-80.16837	6.222904e+11	260000	8438	1794	0	242.8	15310.4	738.6	43127.5	
13	25.89353	-80.16686	6.222904e+11	178000	8438	1509	1248	652.2	14808.8	366.5	43244.6	

Showing 1 to 13 of 13,932 entries, 17 total columns

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

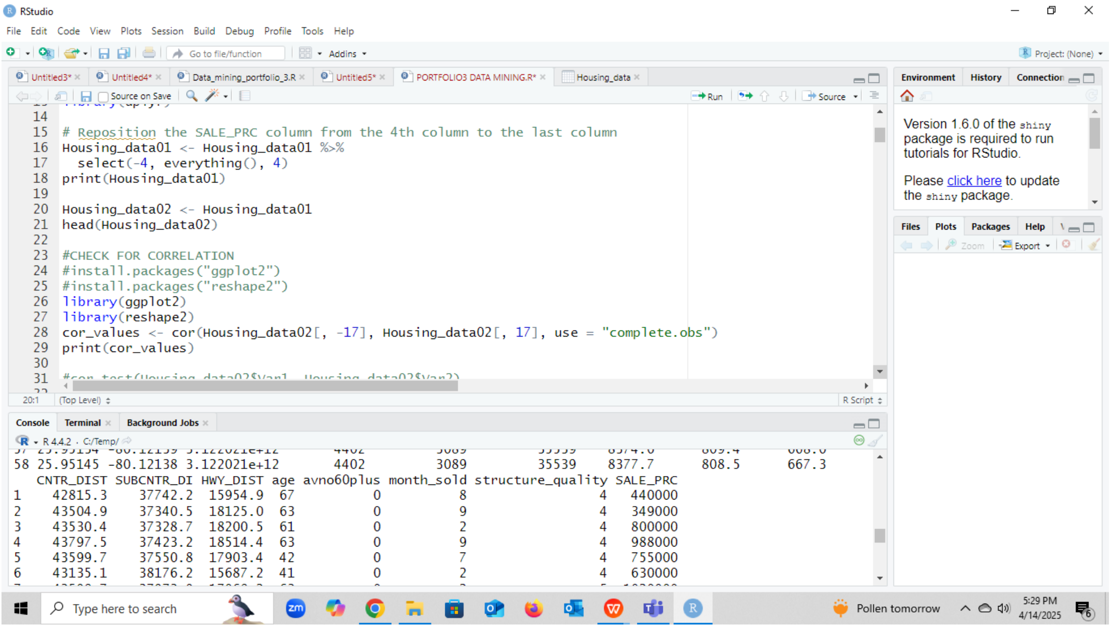
Please click here to update the shiny package.

```
13927     2918.8   22467.2    12042.3   1022.3  69          0         3                 2
13928     3252.4   22175.9    12150.1    917.4  16          0         4                 4
13929     3077.7   22375.1    12316.8    738.2  26          0         5                 4
13930     3588.4   20966.9    12433.0    743.7  16          0         7                 4
13931     3589.1   21475.6    12458.0    626.1  16          0         8                 4
13932     3421.0   21928.6    12599.0    474.7  62          0        11                 4
> View(Housing_data)
> 
```

Fig. 2: A view of the housing dataset

Check for variables with significant correlation with sale price

In the course of understanding the data, the researcher checked for the relationship of other variables with the house sales price in order to know which ones have significant impact on house sale price for model building using correlation analysis alongside heatmap visualisation in R as shown in fig.3, fig.4, fig.5 and fig 6 below. The outcome of the analysis shows that structure quality, HWY_DIST, SPEC_FEAT_VAL, TOT_LVG_AREA and LND_SQFOOT correlate positively with SALE_PRC while SUBCNTR_DI, CNTR_DIST, OCEAN_DIST, age, WATER_DIST and RAIL_DIST correlate negatively with SALE_PRC with LONGITUDE, LATITUDE, avno60plus, month sold and PARCELNO having insignificant impact on SALE_PRC since the houses are in the same location geographically and the other variables listed do not impact on house sale prices.



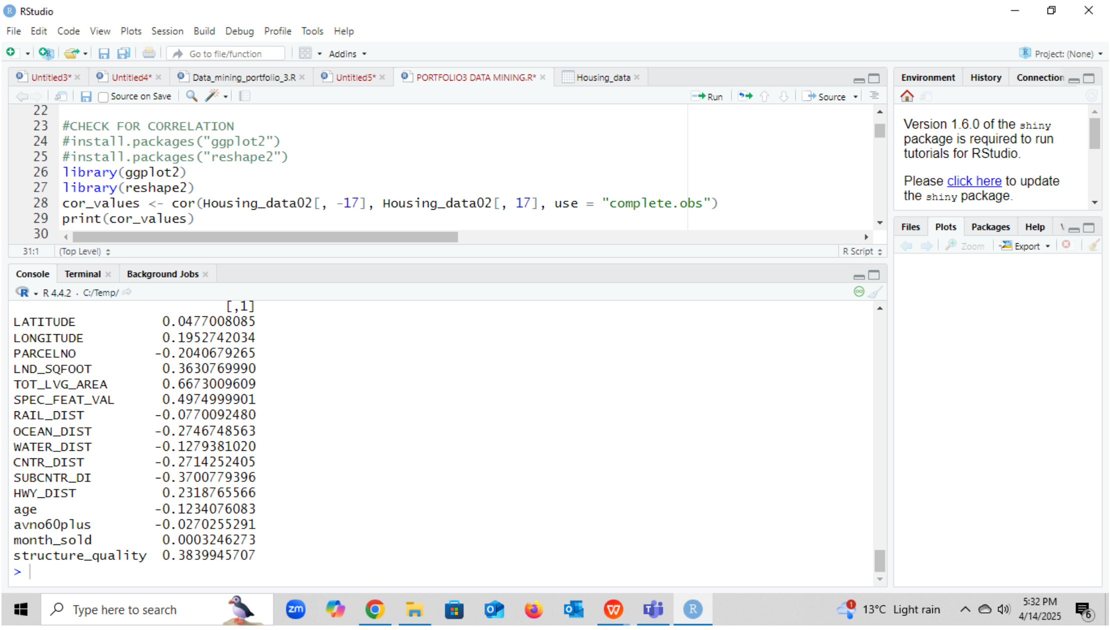
```
# Reposition the SALE_PRC column from the 4th column to the last column
Housing_data01 <- Housing_data01 %>%
  select(-4, everything(), 4)
print(Housing_data01)

Housing_data02 <- Housing_data01
head(Housing_data02)

#CHECK FOR CORRELATION
#install.packages("ggplot2")
#install.packages("reshape2")
library(ggplot2)
library(reshape2)
cor_values <- cor(Housing_data02[, -17], Housing_data02[, 17], use = "complete.obs")
print(cor_values)
```

```
58 25.95145 -80.12138 3.122021e+12       4402        3089        35539      8377.7       808.5       667.3
  CNTR_DIST SUBCNTR_DI HWY_DIST age avno60plus month_sold structure_quality SALE_PRC
1   42815.3    37742.2  15954.9  67          0          8                 4   440000
2   43504.9    37340.5  18125.0  63          0          9                 4   349000
3   43530.4    37328.7  18200.5  61          0          2                 4   800000
4   43797.5    37423.2  18514.4  63          0          9                 4   988000
5   43599.7    37550.8  17903.4  42          0          7                 4   755000
6   43135.1    38176.2  15687.2  41          0          2                 4   630000
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

Fig. 3: Reposition the SALE_PRC column from the 4th column to the last column



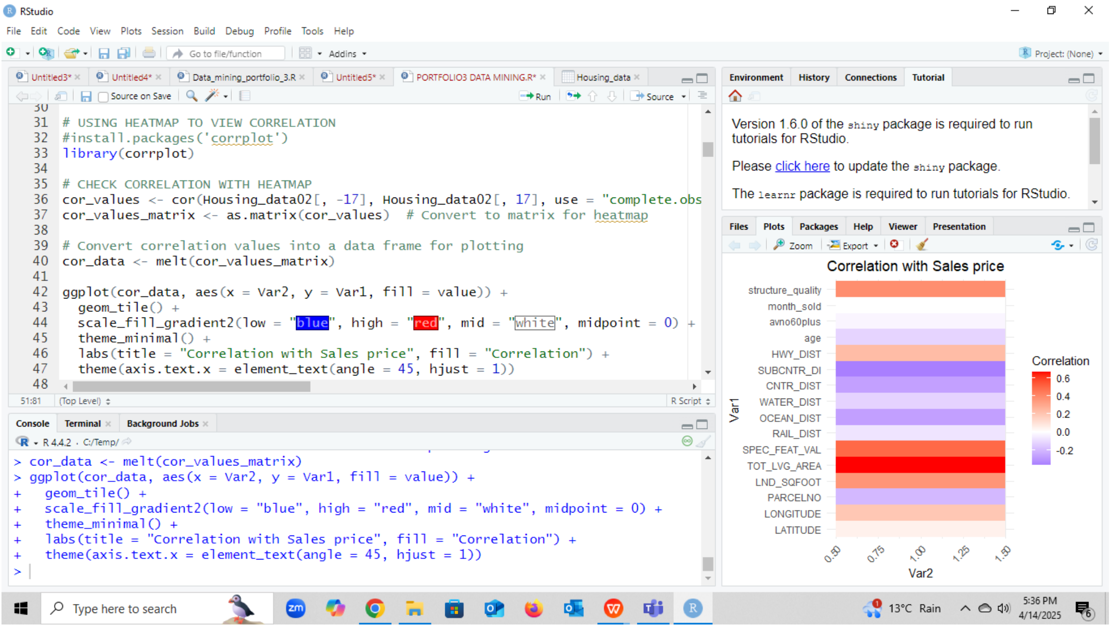
```
#CHECK FOR CORRELATION
#install.packages("ggplot2")
#install.packages("reshape2")
library(ggplot2)
library(reshape2)
cor_values <- cor(Housing_data02[, -17], Housing_data02[, 17], use = "complete.obs")
print(cor_values)
```

```
                           [,1]
LATITUDE           0.0477008085
LONGITUDE          0.1952742034
PARCELNO          -0.2040679265
LND_SQFOOT         0.3630769990
TOT_LVG_AREA       0.6673009609
SPEC_FEAT_VAL      0.4974999901
RAIL_DIST         -0.0770092480
OCEAN_DIST        -0.2746748563
WATER_DIST        -0.1279381020
CNTR_DIST         -0.2714252405
SUBCNTR_DI        -0.3700779396
HWY_DIST           0.2318765566
age               -0.1234076083
avno60plus        -0.0270255291
month_sold         0.0003246273
structure_quality  0.3839945707
>
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

Fig. 4: Check for the correlation of other variables with the SALE_PRC



```
# USING HEATMAP TO VIEW CORRELATION
#install.packages('corrplot')
library(corrplot)

# CHECK CORRELATION WITH HEATMAP
cor_values <- cor(Housing_data02[, -17], Housing_data02[, 17], use = "complete.obs
cor_values_matrix <- as.matrix(cor_values)  # Convert to matrix for heatmap

# Convert correlation values into a data frame for plotting
cor_data <- melt(cor_values_matrix)

ggplot(cor_data, aes(x = Var2, y = Var1, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0) +
  theme_minimal() +
  labs(title = "Correlation with Sales price", fill = "Correlation") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

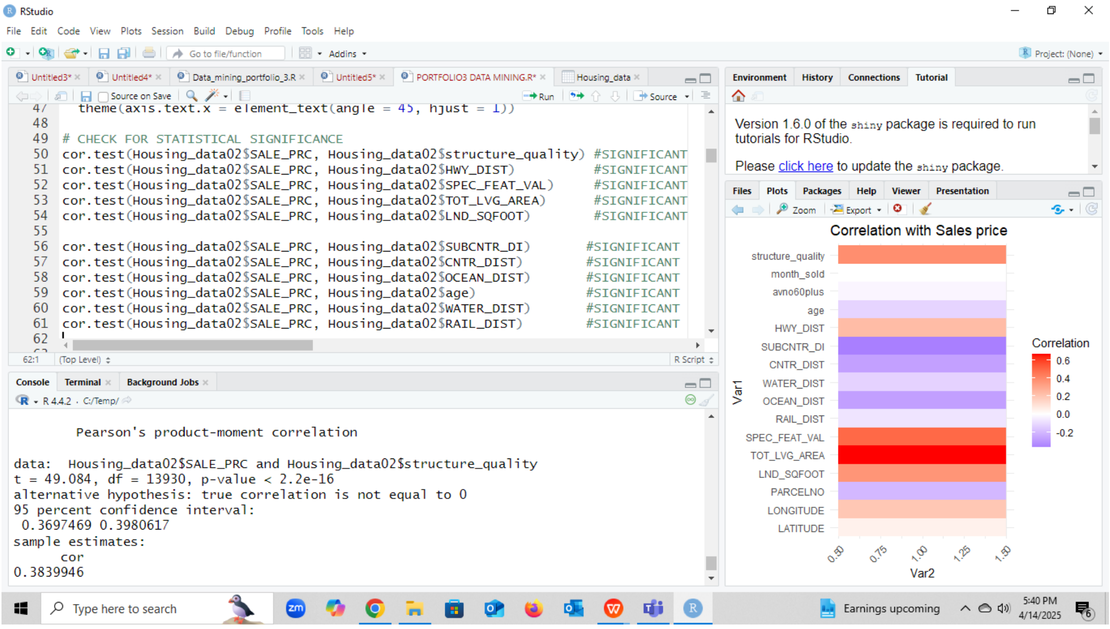
```
> cor_data <- melt(cor_values_matrix)
> ggplot(cor_data, aes(x = Var2, y = Var1, fill = value)) +
+   geom_tile() +
+   scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0) +
+   theme_minimal() +
+   labs(title = "Correlation with Sales price", fill = "Correlation") +
+   theme(axis.text.x = element_text(angle = 45, hjust = 1))
> 
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

The learnr package is required to run tutorials for RStudio.

Fig. 5: Correlation test on the relationship of SALE_PRC with other variables using Heatmap



```
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# CHECK FOR STATISTICAL SIGNIFICANCE
cor.test(Housing_data02$SALE_PRC, Housing_data02$structure_quality) #SIGNIFICANT
cor.test(Housing_data02$SALE_PRC, Housing_data02$HWY_DIST)          #SIGNIFICANT
cor.test(Housing_data02$SALE_PRC, Housing_data02$SPEC_FEAT_VAL)     #SIGNIFICANT
cor.test(Housing_data02$SALE_PRC, Housing_data02$TOT_LVG_AREA)      #SIGNIFICANT
cor.test(Housing_data02$SALE_PRC, Housing_data02$LND_SQFOOT)        #SIGNIFICANT

cor.test(Housing_data02$SALE_PRC, Housing_data02$SUBCNTR_DI)       #SIGNIFICANT
cor.test(Housing_data02$SALE_PRC, Housing_data02$CNTR_DIST)        #SIGNIFICANT
cor.test(Housing_data02$SALE_PRC, Housing_data02$OCEAN_DIST)       #SIGNIFICANT
cor.test(Housing_data02$SALE_PRC, Housing_data02$age)              #SIGNIFICANT
cor.test(Housing_data02$SALE_PRC, Housing_data02$WATER_DIST)       #SIGNIFICANT
cor.test(Housing_data02$SALE_PRC, Housing_data02$RAIL_DIST)        #SIGNIFICANT
```

```

        Pearson's product-moment correlation

data:  Housing_data02$SALE_PRC and Housing_data02$structure_quality
t = 49.084, df = 13930, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3697469 0.3980617
sample estimates:
      cor 
0.3839946 
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

Fig. 6: Statistical test for correlation significance

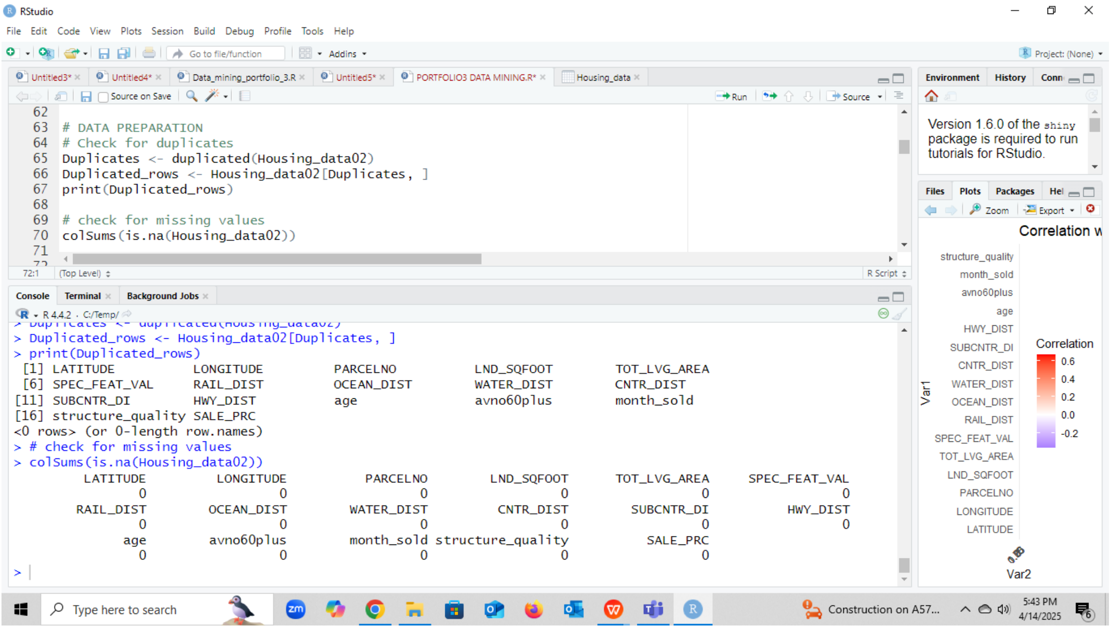
CHAPTER 2

DATA PREPARATION

This phase of the task shows the in-depth data preparation process after identifying the variables that have significant impact on house sale price for model building. It ranges from check for duplicates , drop unwanted columns, check and handling of null values and outliers and carry out other necessary data cleaning procedures.

Below are screenshots of the data preparation process;

Check for duplicate



Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

```
# DATA PREPARATION
# Check for duplicates
Duplicates <- duplicated(Housing_data02)
Duplicated_rows <- Housing_data02[Duplicates, ]
print(Duplicated_rows)

# check for missing values
colSums(is.na(Housing_data02))
```

```
> Duplicates <- duplicated(Housing_data02)
> Duplicated_rows <- Housing_data02[Duplicates, ]
> print(Duplicated_rows)
 [1] LATITUDE          LONGITUDE         PARCELNO          LND_SQFOOT        TOT_LVG_AREA
 [6] SPEC_FEAT_VAL     RAIL_DIST         OCEAN_DIST        WATER_DIST        CNTR_DIST
[11] SUBCNTR_DI        HWY_DIST          age               avno60plus        month_sold
[16] structure_quality SALE_PRC
<0 rows> (or 0-length row.names)
> # check for missing values
> colSums(is.na(Housing_data02))
         LATITUDE         LONGITUDE          PARCELNO        LND_SQFOOT      TOT_LVG_AREA     SPEC_FEAT_VAL
                0                 0                 0                 0                 0                 0
        RAIL_DIST        OCEAN_DIST        WATER_DIST         CNTR_DIST        SUBCNTR_DI          HWY_DIST
                0                 0                 0                 0                 0                 0
              age        avno60plus        month_sold structure_quality          SALE_PRC
                0                 0                 0                 0                 0
> |
```

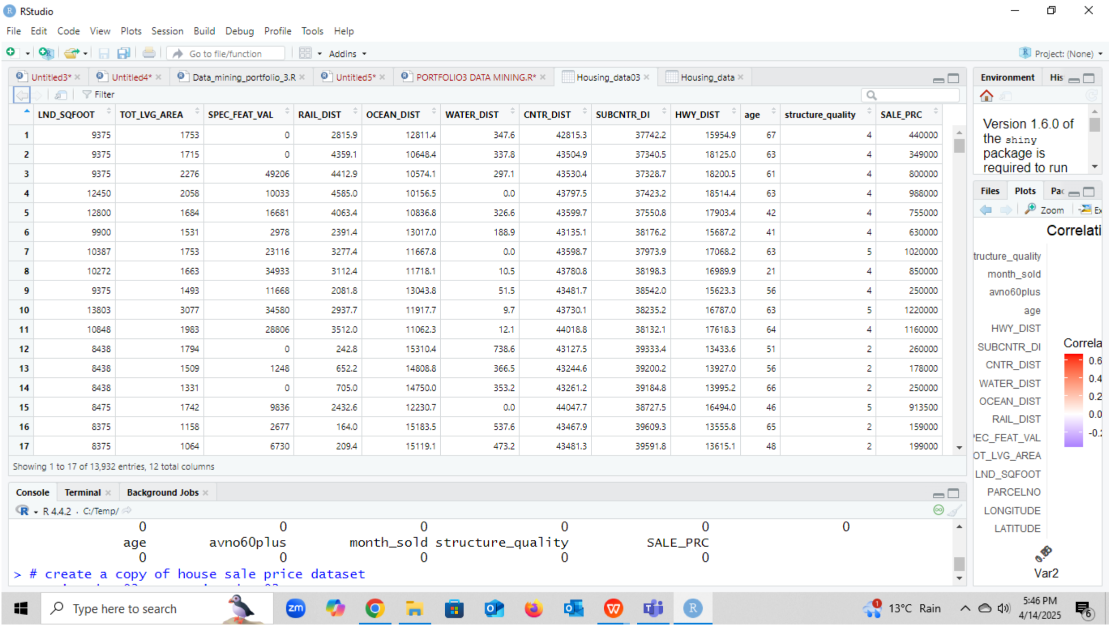
Fig. 7: Check for duplicates and null value

From the above screenshot in Fig. 7, it can be seen that the housing data does not contain any duplicated value after checking for duplicates using the duplicate function.

Check for null value

The housing data does not contain null value as seen in Fig. 7 above.

Drop columns with insignificant correlation on sale price



	LND_SQFOOT	TOT_LVG_AREA	SPEC_FEAT_VAL	RAIL_DIST	OCEAN_DIST	WATER_DIST	CNTR_DIST	SUBCNTR_DI	HWY_DIST	age	structure_quality	SALE_PRC
1	9375	1753	0	2815.9	12811.4	347.6	42815.3	37742.2	15954.9	67	4	440000
2	9375	1715	0	4359.1	10648.4	337.8	43504.9	37340.5	18125.0	63	4	349000
3	9375	2276	49206	4412.9	10574.1	297.1	43530.4	37328.7	18200.5	61	4	800000
4	12450	2058	10033	4585.0	10156.5	0.0	43797.5	37423.2	18514.4	63	4	988000
5	12800	1684	16681	4063.4	10836.8	326.6	43599.7	37550.8	17903.4	42	4	755000
6	9900	1531	2978	2391.4	13017.0	188.9	43135.1	38176.2	15687.2	41	4	630000
7	10387	1753	23116	3277.4	11667.8	0.0	43598.7	37973.9	17068.2	63	5	1020000
8	10272	1663	34933	3112.4	11718.1	10.5	43780.8	38198.3	16989.9	21	4	850000
9	9375	1493	11668	2081.8	13043.8	51.5	43481.7	38542.0	15623.3	56	4	250000
10	13803	3077	34580	2937.7	11917.7	9.7	43730.1	38235.2	16787.0	63	5	1220000
11	10848	1983	28806	3512.0	11062.3	12.1	44018.8	38132.1	17618.3	64	4	1160000
12	8438	1794	0	242.8	15310.4	738.6	43127.5	39333.4	13433.6	51	2	260000
13	8438	1509	1248	652.2	14808.8	366.5	43244.6	39200.2	13927.0	56	2	178000
14	8438	1331	0	705.0	14750.0	353.2	43261.2	39184.8	13995.2	66	2	250000
15	8475	1742	9836	2432.6	12230.7	0.0	44047.7	38727.5	16494.0	46	5	913500
16	8375	1158	2677	164.0	15183.5	537.6	43467.9	39609.3	13555.8	65	2	159000
17	8375	1064	6730	209.4	15119.1	473.2	43481.3	39591.8	13615.1	48	2	199000

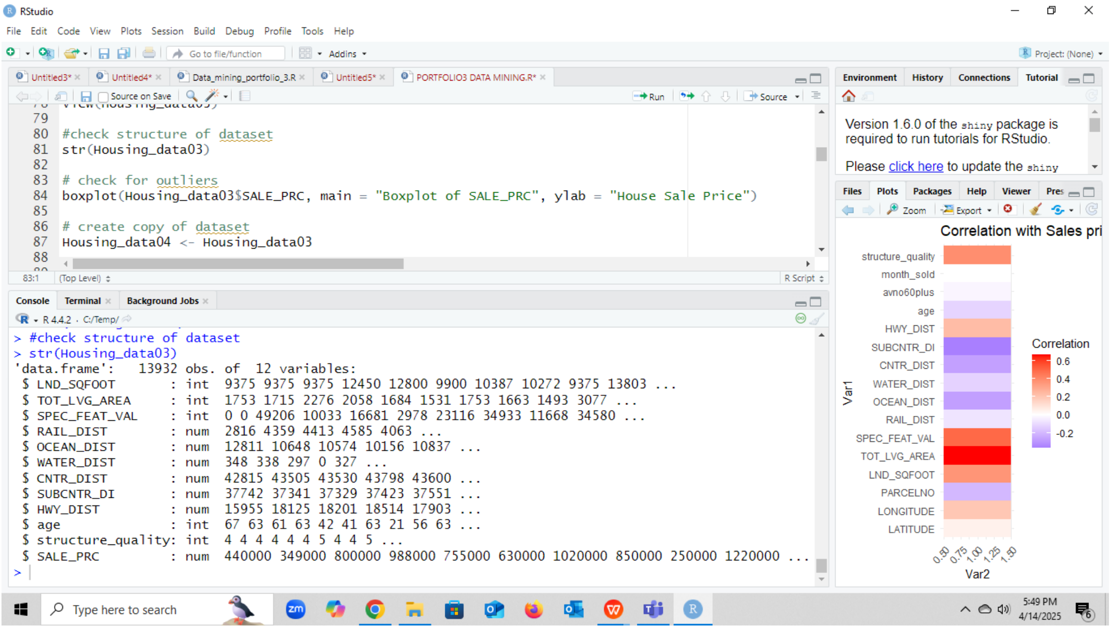
Showing 1 to 17 of 13,932 entries, 12 total columns

```
               0              0              0              0              0              0
             age     avno60plus     month_sold structure_quality       SALE_PRC
               0              0              0              0              0
> # create a copy of house sale price dataset
```

Fig. 8: Drop columns with insignificant correlation

After carrying out correlation analysis, the following columns; LATITUDE, LONGITUDE, PARCELNO, avno60plus and month_sold are considered to be less relevant in the model building for house price prediction due to their insignificant correlation with house sale price thereby removing them from the dataset as shown above in Fig.8

Structure of Housing data



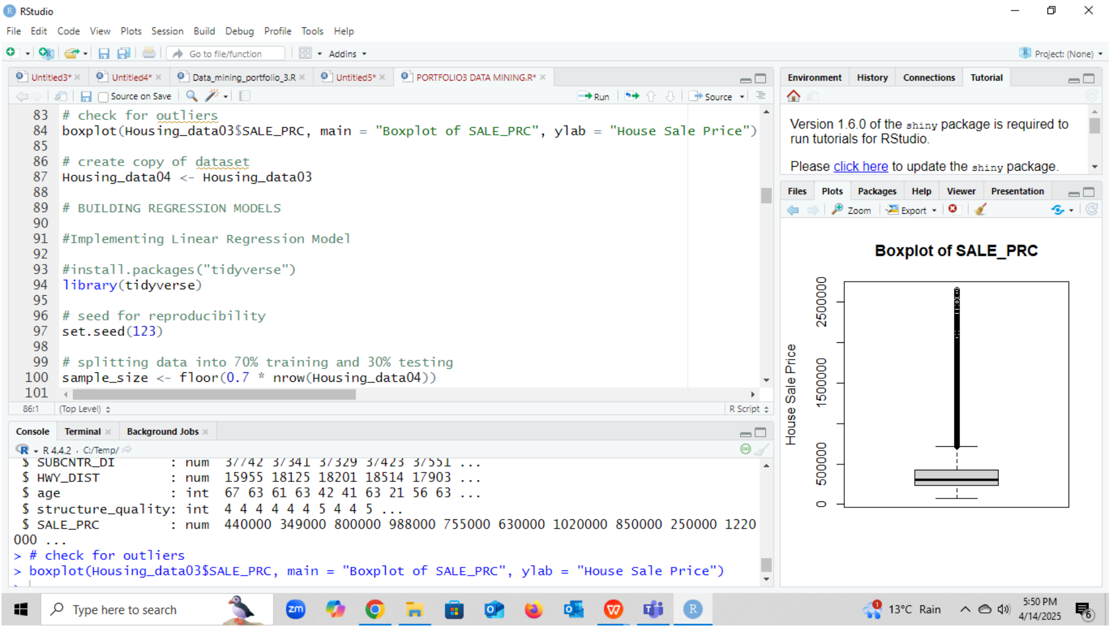
```
#check structure of dataset
str(Housing_data03)

# check for outliers
boxplot(Housing_data03$SALE_PRC, main = "Boxplot of SALE_PRC", ylab = "House Sale Price")

# create copy of dataset
Housing_data04 <- Housing_data03

```

```
> #check structure of dataset
> str(Housing_data03)
'data.frame':	13932 obs. of  12 variables:
 $ LND_SQFOOT       : int  9375 9375 9375 12450 12800 9900 10387 10272 9375 13803 ...
 $ TOT_LVG_AREA     : int  1753 1715 2276 2058 1684 1531 1753 1663 1493 3077 ...
 $ SPEC_FEAT_VAL    : int  0 0 49206 10033 16681 2978 23116 34933 11668 34580 ...
 $ RAIL_DIST        : num  2816 4359 4413 4585 4063 ...
 $ OCEAN_DIST       : num  12811 10648 10574 10156 10837 ...
 $ WATER_DIST       : num  348 338 297 0 327 ...
 $ CNTR_DIST        : num  42815 43505 43530 43798 43600 ...
 $ SUBCNTR_DI       : num  37742 37341 37329 37423 37551 ...
 $ HWY_DIST         : num  15955 18125 18201 18514 17903 ...
 $ age              : int  67 63 61 63 42 41 63 21 56 63 ...
 $ structure_quality: int  4 4 4 4 4 4 5 4 4 5 ...
 $ SALE_PRC         : num  440000 349000 800000 988000 755000 630000 1020000 850000 250000 1220000 ...
> 
```

Fig. 9: Structure of Housing data

The structure of the housing data for model building as shown in fig.9 has twelve (12) variables consisting of eleven explanatory variables and one dependent variable being the sale price. The total number of observations is thirteen thousand nine hundred and thirty two with all columns data type being numerical.

Handling outliers



```
# check for outliers
boxplot(Housing_data03$SALE_PRC, main = "Boxplot of SALE_PRC", ylab = "House Sale Price")

# create copy of dataset
Housing_data04 <- Housing_data03

# BUILDING REGRESSION MODELS

#Implementing Linear Regression Model

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

# seed for reproducibility
set.seed(123)

# splitting data into 70% training and 30% testing
sample_size <- floor(0.7 * nrow(Housing_data04))
```

```
 $ SUBCNTR_DI        : num  37742 37341 37329 37423 37551 ...
 $ HWY_DIST          : num  15955 18125 18201 18514 17903 ...
 $ age               : int  67 63 61 63 42 41 63 21 56 63 ...
 $ structure_quality: int  4 4 4 4 4 4 5 4 4 5 ...
 $ SALE_PRC          : num  440000 349000 800000 988000 755000 630000 1020000 850000 250000 1220
000 ...
> # check for outliers
> boxplot(Housing_data03$SALE_PRC, main = "Boxplot of SALE_PRC", ylab = "House Sale Price")
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

Fig. 10: Check for outliers

The above boxplot in Fig.10 shows the outliers chart. Because of the high margins in the house sale prices as a result of different components ranging from land and house sizes, special features among others, the house sale prices that appear as outliers but not truly outliers in the actual sense going by the context of the dataset will be included in the model building process for effective and accurate predictions.

CHAPTER 3

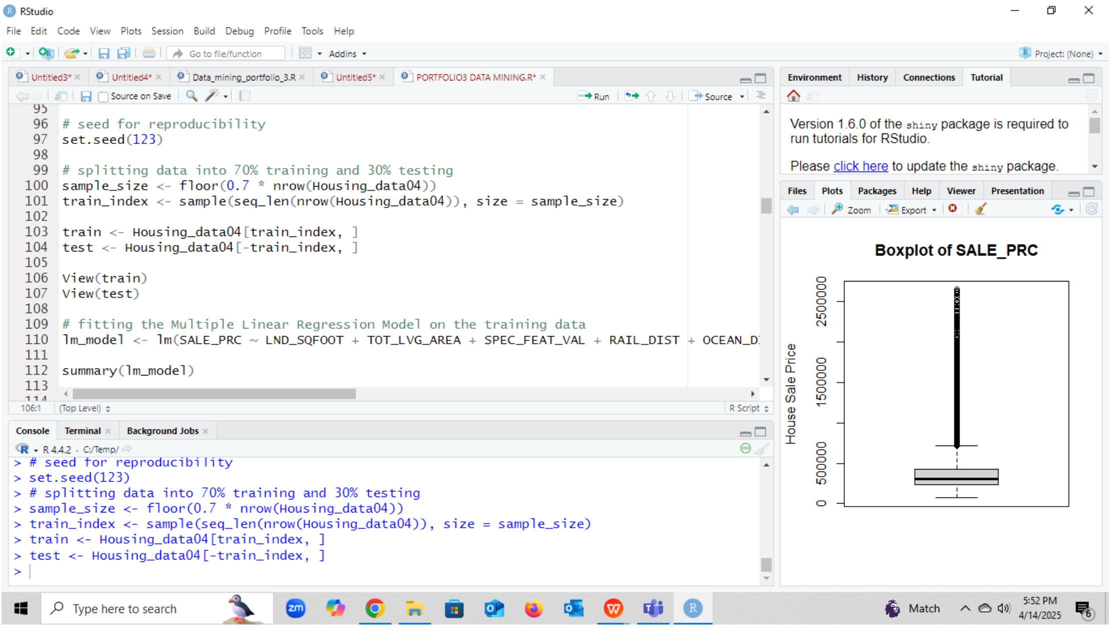
MODELING

3.1 Modeling technique

The algorithm to build and assess the various models for this project is the regression modeling technique. This technique model relationship between a dependent variable in this case the house sale price and other independent variables as contained in the house data as shown in Fig.8 as it gives solid predictions for continuous outcomes like the house sale price.

3.2 Machine Learning Models

In this project, eight (8) machine learning models are built, namely; multiple linear regression model, support vector regression linear model, support vector regression radial model, support vector regression polynomial model, decision tree model, random forest model_n100, random forest model_n200, and random forest model_n500 to predict the house sale price (SALE_PRC) using the following explanatory variables; LND_SQFOOT, TOT_LVG_AREA, SPEC_FEAT_VAL , RAIL_DIST, OCEAN_DIST, WATER_DIST, CNTR_DIST, SUBCNTR_DI, HWY_DIST, age and structure_quality. The model building process starts with setting seed for reproducibility at 123 to ensure that the same random results are obtained every time the code is executed as shown in Fig…... The data is then divided into training data and testing data with 70% of the total data accounting for the training data and the remaining 30% data assigned to the testing data. The next step in the model building phase is fitting the models on the training data where SALE_PRC is the dependent variable and the rest are the independent variables. When the model fitting phase is completed, prediction is then made on the testing data using the trained model as shown in fig. 12, fig. 13, fig. 14, fig. 15, fig. 16, fig. 17 and fig. 18 below . This process is carried out on the eight (8) models to train them with the training data before generating predictions using the testing data. At the end of this phase, the models are then evaluated based on their performance metrics to identify the model that best meet the business purpose for the project.


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Untitled3* Untitled4* Data_mining_portfolio_3.R Untitled5* PORTFOLIO3 DATA MINING.R*
Source on Save
Run
Source
Environment History Connections Tutorial
Files Plots Packages Help Viewer Presentation
Zoom Export
Boxplot of SALE_PRC
House Sale Price
0
500000
1500000
2500000
106:1 (Top Level) R Script
Console Terminal Background Jobs
R 4.4.2 · C:/Temp/
Type here to search
Match
5:52 PM
4/14/2025

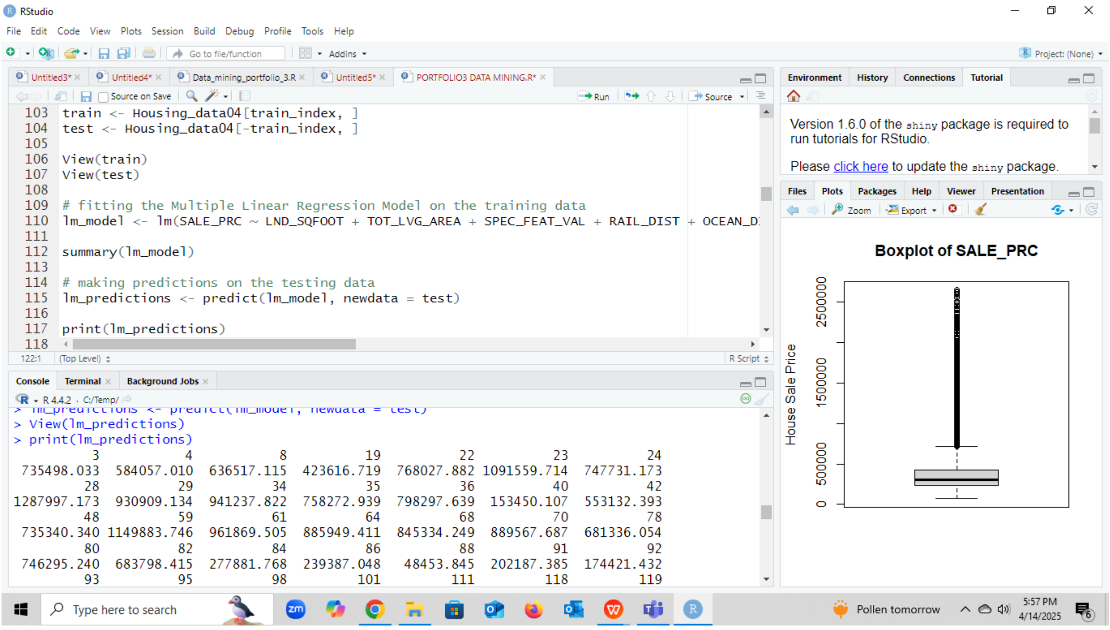
```
# seed for reproducibility
set.seed(123)

# splitting data into 70% training and 30% testing
sample_size <- floor(0.7 * nrow(Housing_data04))
train_index <- sample(seq_len(nrow(Housing_data04)), size = sample_size)

train <- Housing_data04[train_index, ]
test <- Housing_data04[-train_index, ]

View(train)
View(test)

# fitting the Multiple Linear Regression Model on the training data
lm_model <- lm(SALE_PRC ~ LND_SQFOOT + TOT_LVG_AREA + SPEC_FEAT_VAL + RAIL_DIST + OCEAN_D

summary(lm_model)
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

```
> # seed for reproducibility
> set.seed(123)
> # splitting data into 70% training and 30% testing
> sample_size <- floor(0.7 * nrow(Housing_data04))
> train_index <- sample(seq_len(nrow(Housing_data04)), size = sample_size)
> train <- Housing_data04[train_index, ]
> test <- Housing_data04[-train_index, ]
> 
```

Fig. 11: Set seed and split data into training and testing data for model building


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Untitled3* Untitled4* Data_mining_portfolio_3.R Untitled5* PORTFOLIO3 DATA MINING.R*
Source on Save
Run
Source
Environment History Connections Tutorial
Files Plots Packages Help Viewer Presentation
Zoom Export
Boxplot of SALE_PRC
House Sale Price
0
500000
1500000
2500000
122:1 (Top Level) R Script
Console Terminal Background Jobs
R 4.4.2 · C:/Temp/
Type here to search
Pollen tomorrow
5:57 PM
4/14/2025

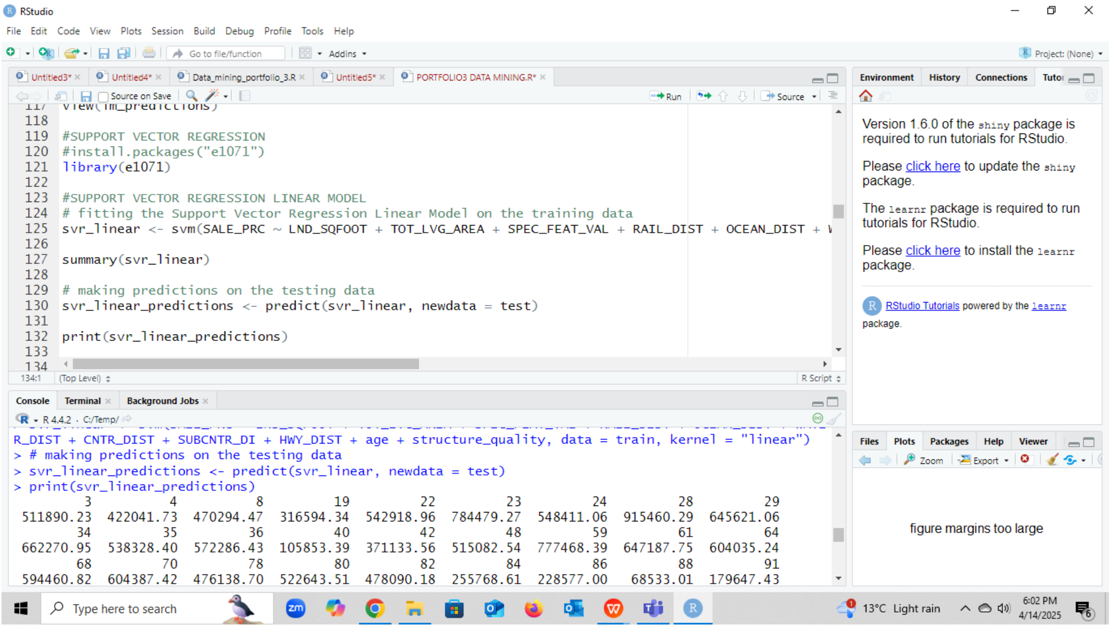
```
train <- Housing_data04[train_index, ]
test <- Housing_data04[-train_index, ]

View(train)
View(test)

# fitting the Multiple Linear Regression Model on the training data
lm_model <- lm(SALE_PRC ~ LND_SQFOOT + TOT_LVG_AREA + SPEC_FEAT_VAL + RAIL_DIST + OCEAN_D

summary(lm_model)

# making predictions on the testing data
lm_predictions <- predict(lm_model, newdata = test)

print(lm_predictions)
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

```
> lm_predictions <- predict(lm_model, newdata = test)
> View(lm_predictions)
> print(lm_predictions)
          3           4           8          19          22          23          24 
 735498.033  584057.010  636517.115  423616.719  768027.882 1091559.714  747731.173 
         28          29          34          35          36          40          42 
1287997.173  930909.134  941237.822  758272.939  798297.639  153450.107  553132.393 
         48          59          61          64          68          70          78 
 735340.340 1149883.746  961869.505  885949.411  845334.249  889567.687  681336.054 
         80          82          84          86          88          91          92 
 746295.240  683798.415  277881.768  239387.048   48453.845  202187.385  174421.432 
         93          95          98         101         111         118         119 
```

Fig. 12: Fitting multiple linear regression model on training data and prediction



```
View(lm_predictions)

#SUPPORT VECTOR REGRESSION
#install.packages("e1071")
library(e1071)

#SUPPORT VECTOR REGRESSION LINEAR MODEL
# fitting the Support Vector Regression Linear Model on the training data
svr_linear <- svm(SALE_PRC ~ LND_SQFOOT + TOT_LVG_AREA + SPEC_FEAT_VAL + RAIL_DIST + OCEAN_DIST + W

summary(svr_linear)

# making predictions on the testing data
svr_linear_predictions <- predict(svr_linear, newdata = test)

print(svr_linear_predictions)
```

```
R_DIST + CNTR_DIST + SUBCNTR_DI + HWY_DIST + age + structure_quality, data = train, kernel = "linear")
> # making predictions on the testing data
> svr_linear_predictions <- predict(svr_linear, newdata = test)
> print(svr_linear_predictions)
        3         4         8        19        22        23        24        28        29 
511890.23 422041.73 470294.47 316594.34 542918.96 784479.27 548411.06 915460.29 645621.06 
       34        35        36        40        42        48        59        61        64 
662270.95 538328.40 572286.43 105853.39 371133.56 515082.54 777468.39 647187.75 604035.24 
       68        70        78        80        82        84        86        88        91 
594460.82 604387.42 476138.70 522643.51 478090.18 255768.61 228577.00  68533.01 179647.43 
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

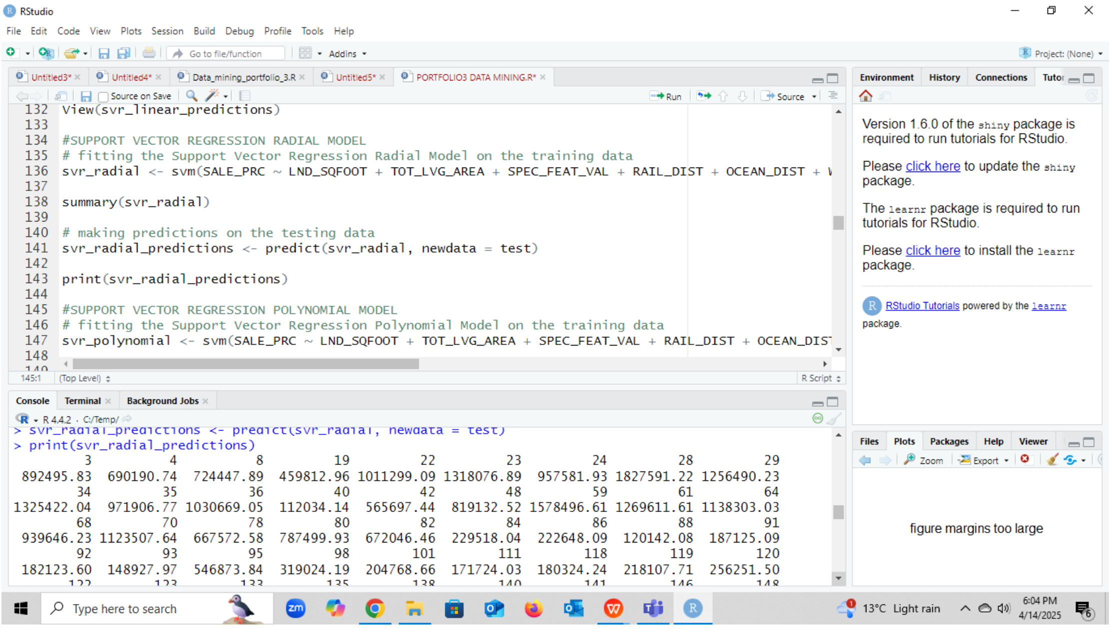
The learnr package is required to run tutorials for RStudio.

Please click here to install the learnr package.

RStudio Tutorials powered by the learnr package.

figure margins too large

Fig. 13: Fitting support vector regression linear model on training data and prediction



```
View(svr_linear_predictions)

#SUPPORT VECTOR REGRESSION RADIAL MODEL
# fitting the Support Vector Regression Radial Model on the training data
svr_radial <- svm(SALE_PRC ~ LND_SQFOOT + TOT_LVG_AREA + SPEC_FEAT_VAL + RAIL_DIST + OCEAN_DIST + W

summary(svr_radial)

# making predictions on the testing data
svr_radial_predictions <- predict(svr_radial, newdata = test)

print(svr_radial_predictions)

#SUPPORT VECTOR REGRESSION POLYNOMIAL MODEL
# fitting the Support Vector Regression Polynomial Model on the training data
svr_polynomial <- svm(SALE_PRC ~ LND_SQFOOT + TOT_LVG_AREA + SPEC_FEAT_VAL + RAIL_DIST + OCEAN_DIST
```

```
> svr_radial_predictions <- predict(svr_radial, newdata = test)
> print(svr_radial_predictions)
        3         4         8        19        22        23        24        28        29 
 892495.83  690190.74  724447.89  459812.96 1011299.09 1318076.89  957581.93 1827591.22 1256490.23 
        34         35         36         40         42         48         59         61         64 
1325422.04  971906.77 1030669.05  112034.14  565697.44  819132.52 1578496.61 1269611.61 1138303.03 
        68         70         78         80         82         84         86         88         91 
 939646.23 1123507.64  667572.58  787499.93  672046.46  229518.04  222648.09  120142.08  187125.09 
        92         93         95         98        101        111        118        119        120 
 182123.60  148927.97  546873.84  319024.19  204768.66  171724.03  180324.24  218107.71  256251.50 
       122        123        133        135        138        140        141        146        148 
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

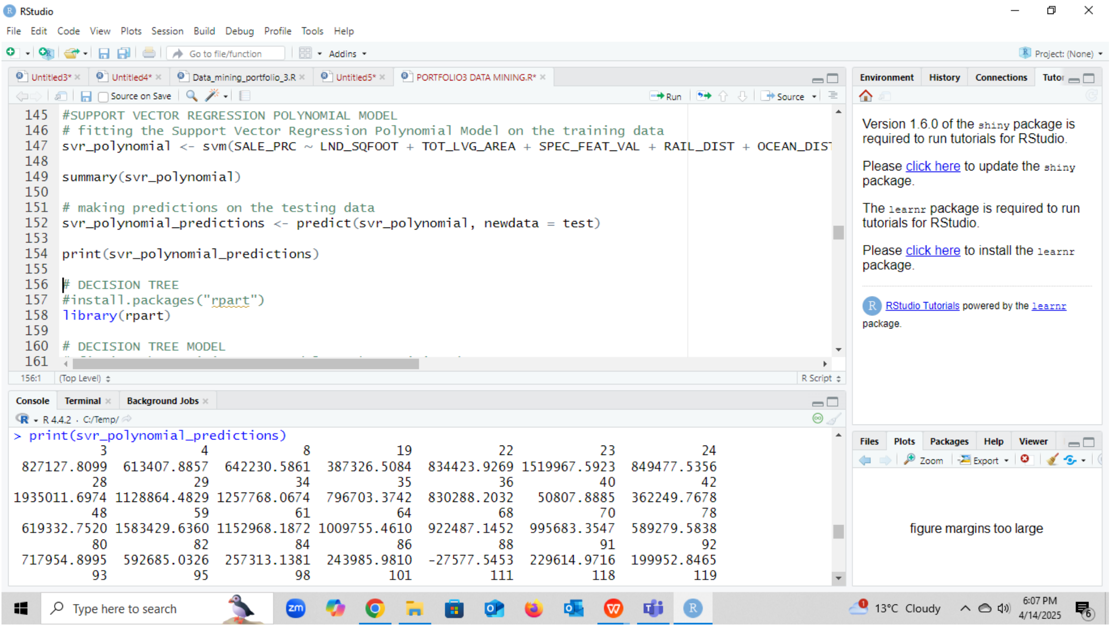
The learnr package is required to run tutorials for RStudio.

Please click here to install the learnr package.

RStudio Tutorials powered by the learnr package.

figure margins too large

Fig. 14: Fitting support vector regression radial model on training data and prediction



```
#SUPPORT VECTOR REGRESSION POLYNOMIAL MODEL
# fitting the Support Vector Regression Polynomial Model on the training data
svr_polynomial <- svm(SALE_PRC ~ LND_SQFOOT + TOT_LVG_AREA + SPEC_FEAT_VAL + RAIL_DIST + OCEAN_DIST

summary(svr_polynomial)

# making predictions on the testing data
svr_polynomial_predictions <- predict(svr_polynomial, newdata = test)

print(svr_polynomial_predictions)

# DECISION TREE
#install.packages("rpart")
library(rpart)

# DECISION TREE MODEL
```

```
> print(svr_polynomial_predictions)
          3           4           8          19          22          23          24 
 827127.8099  613407.8857  642230.5861  387326.5084  834423.9269 1519967.5923  849477.5356 
         28          29          34          35          36          40          42 
1935011.6974 1128864.4829 1257768.0674  796703.3742  830288.2032   50807.8885  362249.7678 
         48          59          61          64          68          70          78 
 619332.7520 1583429.6360 1152968.1872 1009755.4610  922487.1452  995683.3547  589279.5838 
         80          82          84          86          88          91          92 
 717954.8995  592685.0326  257313.1381  243985.9810  -27577.5453  229614.9716  199952.8465 
         93          95          98         101         111         118         119 
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

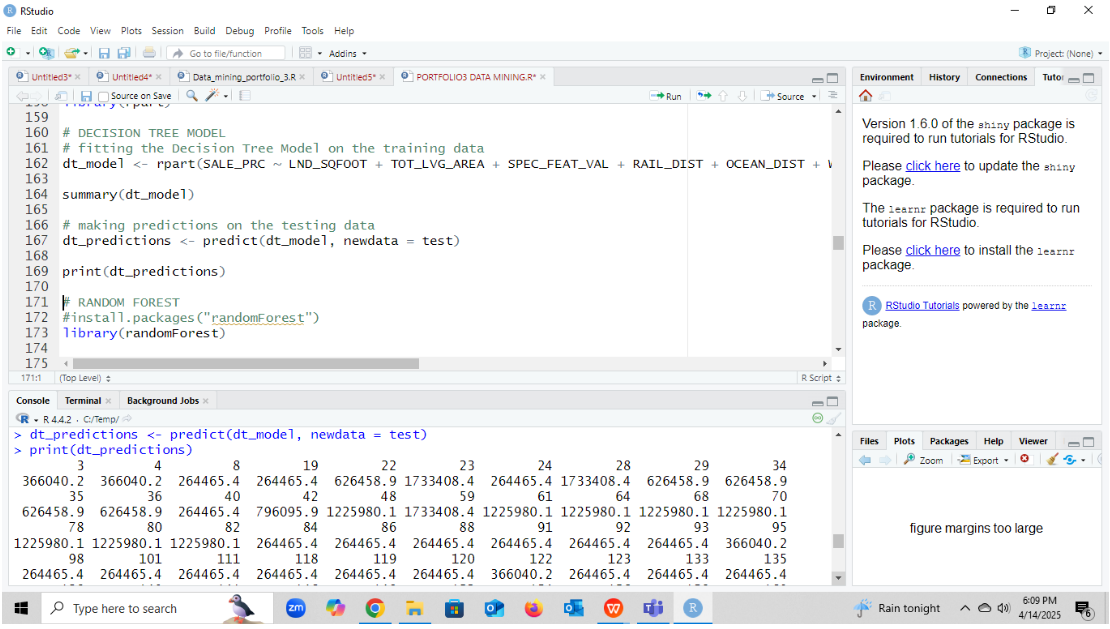
The learnr package is required to run tutorials for RStudio.

Please click here to install the learnr package.

RStudio Tutorials powered by the learnr package.

figure margins too large

Fig. 15: Fitting support vector regression polynomial model on training data and prediction



```
# DECISION TREE MODEL
# fitting the Decision Tree Model on the training data
dt_model <- rpart(SALE_PRC ~ LND_SQFOOT + TOT_LVG_AREA + SPEC_FEAT_VAL + RAIL_DIST + OCEAN_DIST + V

summary(dt_model)

# making predictions on the testing data
dt_predictions <- predict(dt_model, newdata = test)

print(dt_predictions)

# RANDOM FOREST
#install.packages("randomForest")
library(randomForest)
```

```
> dt_predictions <- predict(dt_model, newdata = test)
> print(dt_predictions)
        3         4         8        19        22        23        24        28        29        34 
 366040.2  366040.2  264465.4  264465.4  626458.9 1733408.4  264465.4 1733408.4  626458.9  626458.9 
       35        36        40        42        48        59        61        64        68        70 
 626458.9  626458.9  264465.4  796095.9 1225980.1 1733408.4 1225980.1 1225980.1 1225980.1 1225980.1 
       78        80        82        84        86        88        91        92        93        95 
1225980.1 1225980.1 1225980.1  264465.4  264465.4  264465.4  264465.4  264465.4  264465.4  366040.2 
       98       101       111       118       119       120       122       123       133       135 
 264465.4  264465.4  264465.4  264465.4  264465.4  264465.4  366040.2  264465.4  264465.4  264465.4 
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

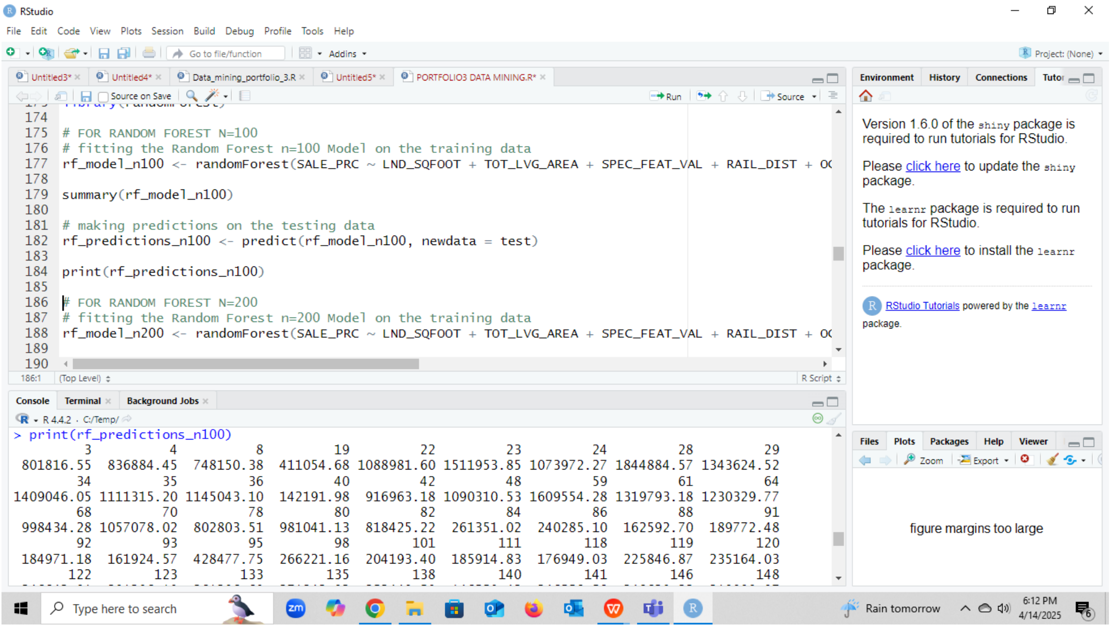
The learnr package is required to run tutorials for RStudio.

Please click here to install the learnr package.

RStudio Tutorials powered by the learnr package.

figure margins too large

Fig. 16: Fitting decision tree model on training data and prediction



```
# FOR RANDOM FOREST N=100
# fitting the Random Forest n=100 Model on the training data
rf_model_n100 <- randomForest(SALE_PRC ~ LND_SQFOOT + TOT_LVG_AREA + SPEC_FEAT_VAL + RAIL_DIST + OC

summary(rf_model_n100)

# making predictions on the testing data
rf_predictions_n100 <- predict(rf_model_n100, newdata = test)

print(rf_predictions_n100)

# FOR RANDOM FOREST N=200
# fitting the Random Forest n=200 Model on the training data
rf_model_n200 <- randomForest(SALE_PRC ~ LND_SQFOOT + TOT_LVG_AREA + SPEC_FEAT_VAL + RAIL_DIST + OC
```

```
> print(rf_predictions_n100)
         3          4          8         19         22         23         24         28         29 
 801816.55  836884.45  748150.38  411054.68 1088981.60 1511953.85 1073972.27 1844884.57 1343624.52 
        34         35         36         40         42         48         59         61         64 
1409046.05 1111315.20 1145043.10  142191.98  916963.18 1090310.53 1609554.28 1319793.18 1230329.77 
        68         70         78         80         82         84         86         88         91 
 998434.28 1057078.02  802803.51  981041.13  818425.22  261351.02  240285.10  162592.70  189772.48 
        92         93         95         98        101        111        118        119        120 
 184971.18  161924.57  428477.75  266221.16  204193.40  185914.83  176949.03  225846.87  235164.03 
       122        123        133        135        138        140        141        146        148 
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

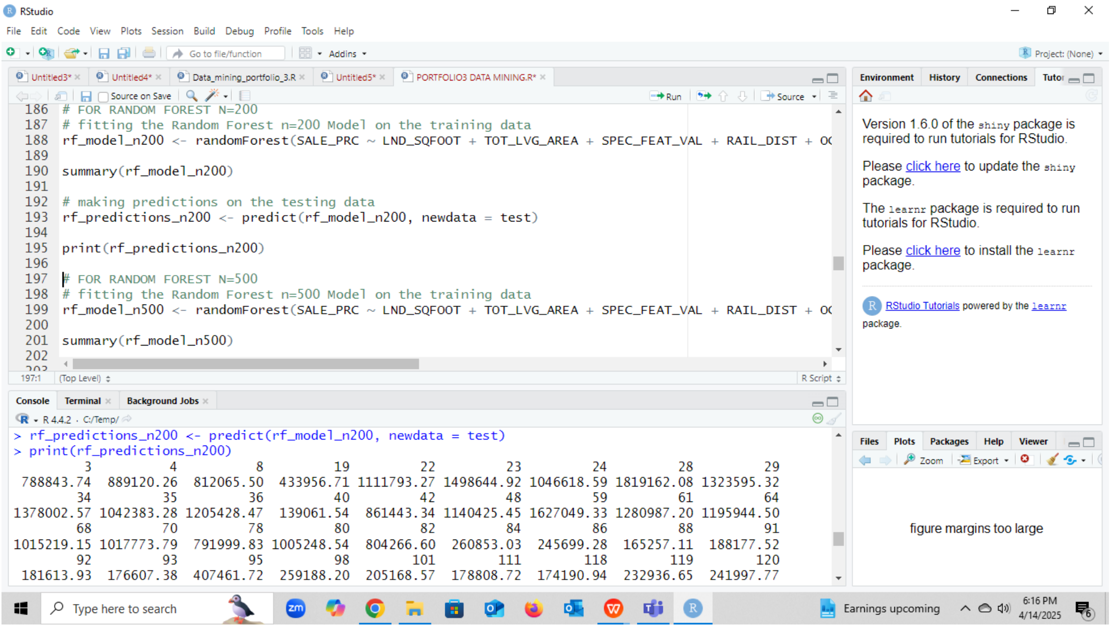
The learnr package is required to run tutorials for RStudio.

Please click here to install the learnr package.

RStudio Tutorials powered by the learnr package.

figure margins too large

Fig. 17: Fitting random forest model with n=100 on training data and prediction



```
# FOR RANDOM FOREST N=200
# fitting the Random Forest n=200 Model on the training data
rf_model_n200 <- randomForest(SALE_PRC ~ LND_SQFOOT + TOT_LVG_AREA + SPEC_FEAT_VAL + RAIL_DIST + OC

summary(rf_model_n200)

# making predictions on the testing data
rf_predictions_n200 <- predict(rf_model_n200, newdata = test)

print(rf_predictions_n200)

# FOR RANDOM FOREST N=500
# fitting the Random Forest n=500 Model on the training data
rf_model_n500 <- randomForest(SALE_PRC ~ LND_SQFOOT + TOT_LVG_AREA + SPEC_FEAT_VAL + RAIL_DIST + OC

summary(rf_model_n500)
```

```
> rf_predictions_n200 <- predict(rf_model_n200, newdata = test)
> print(rf_predictions_n200)
         3          4          8         19         22         23         24         28         29 
 788843.74  889120.26  812065.50  433956.71 1111793.27 1498644.92 1046618.59 1819162.08 1323595.32 
        34         35         36         40         42         48         59         61         64 
1378002.57 1042383.28 1205428.47  139061.54  861443.34 1140425.45 1627049.33 1280987.20 1195944.50 
        68         70         78         80         82         84         86         88         91 
1015219.15 1017773.79  791999.83 1005248.54  804266.60  260853.03  245699.28  165257.11  188177.52 
        92         93         95         98        101        111        118        119        120 
 181613.93  176607.38  407461.72  259188.20  205168.57  178808.72  174190.94  232936.65  241997.77 
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

The learnr package is required to run tutorials for RStudio.

Please click here to install the learnr package.

RStudio Tutorials powered by the learnr package.

figure margins too large

Fig. 18: Fitting random forest model with n=200 and n=500 on training data and prediction

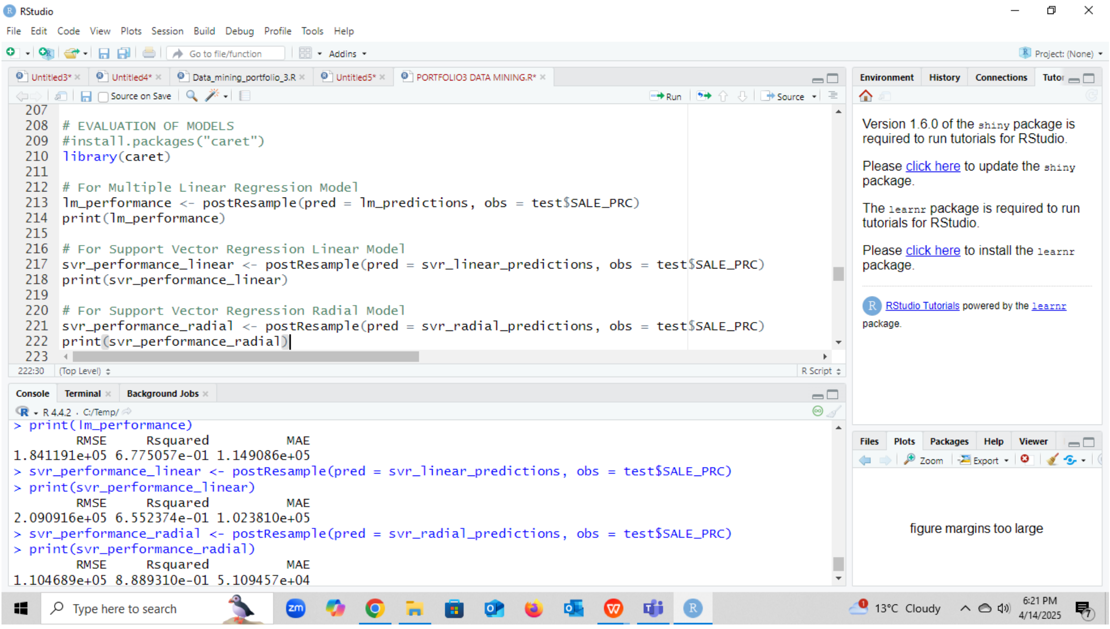
CHAPTER 4

EVALUATION OF MODELS

The models are evaluated by calculating their individual performance metrics and compare using visualisation procedure to identify the best model with the least error using the mean squared error. This is done by using the postResample function from caret package to compute standard performance metrics for the regression models as shown in fig.19 and fig.20 below. The outcome from the evaluation phase shows the various models with their corresponding mean squared error values represented graphically as seen in the bar chart. From the mean squared error values obtained, support vector regression linear model has the least error making it the best fit for the business project. In order to optimise the support vector regression with linear kernel, hyper parameter tuning is carried out on the model to enhance its effectiveness and prediction capacity.

Optimal model test

Given the following information LND_SQFOOT: 11247, TOTLVGAREA: 4552, SPECFEATVAL: 2105, RAIL_DIST: 4871.9, OCEAN_DIST: 18507.2, WATER_DIST:375.8, CNTR_DIST: 43897.9, SUBCNTR_DI: 40115.7, HWY_DIST: 41917.1, age: 42 and structure_quality: 5 to make prediction on the house sale price using the optimised support vector regression model with linear kernel which is the optimal model in the project, the predicted house sale price is 837089.31 as shown in fig. 22.



```
# EVALUATION OF MODELS
#install.packages("caret")
library(caret)

# For Multiple Linear Regression Model
lm_performance <- postResample(pred = lm_predictions, obs = test$SALE_PRC)
print(lm_performance)

# For Support Vector Regression Linear Model
svr_performance_linear <- postResample(pred = svr_linear_predictions, obs = test$SALE_PRC)
print(svr_performance_linear)

# For Support Vector Regression Radial Model
svr_performance_radial <- postResample(pred = svr_radial_predictions, obs = test$SALE_PRC)
print(svr_performance_radial)
```

```
> print(lm_performance)
        RMSE     Rsquared          MAE
1.841191e+05 6.775057e-01 1.149086e+05
> svr_performance_linear <- postResample(pred = svr_linear_predictions, obs = test$SALE_PRC)
> print(svr_performance_linear)
        RMSE     Rsquared          MAE
2.090916e+05 6.552374e-01 1.023810e+05
> svr_performance_radial <- postResample(pred = svr_radial_predictions, obs = test$SALE_PRC)
> print(svr_performance_radial)
        RMSE     Rsquared          MAE
1.104689e+05 8.889310e-01 5.109457e+04
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

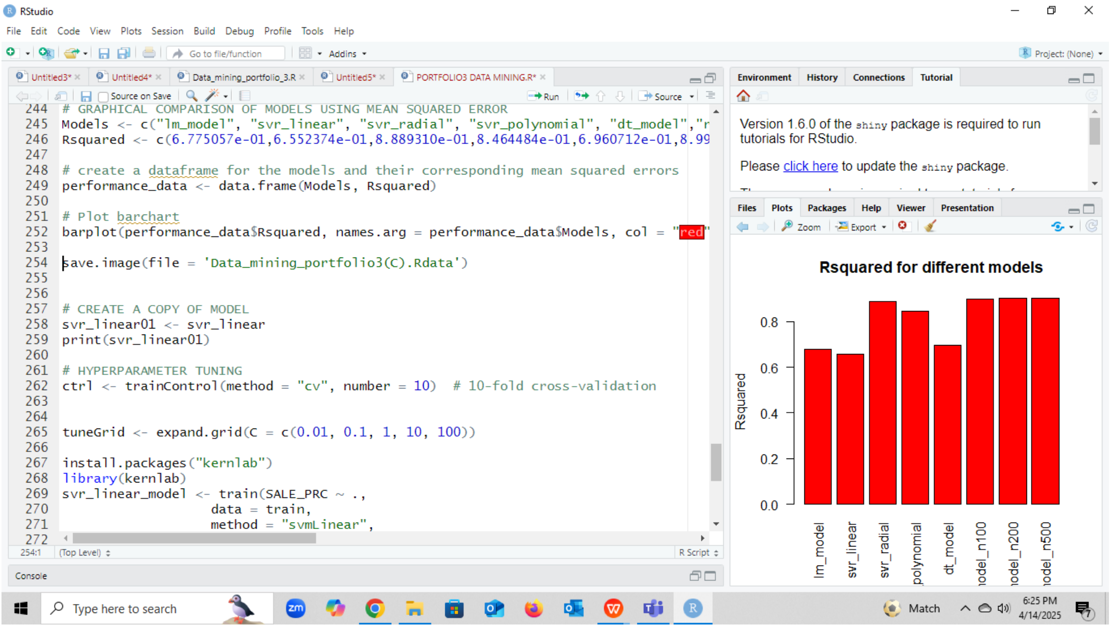
The learnr package is required to run tutorials for RStudio.

Please click here to install the learnr package.

RStudio Tutorials powered by the learnr package.

figure margins too large

Fig. 19: Evaluation of models

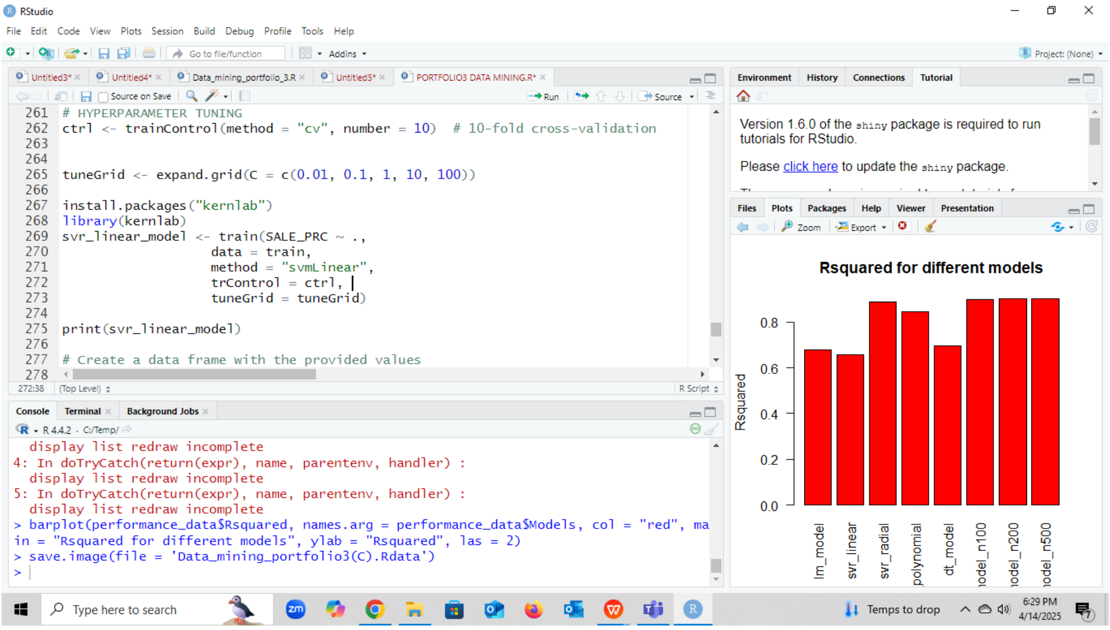


```
# GRAPHICAL COMPARISON OF MODELS USING MEAN SQUARED ERROR
Models <- c("lm_model", "svr_linear", "svr_radial", "svr_polynomial", "dt_model","r
Rsquared <- c(6.775057e-01,6.552374e-01,8.889310e-01,8.464484e-01,6.960712e-01,8.99

# create a dataframe for the models and their corresponding mean squared errors
performance_data <- data.frame(Models, Rsquared)

# Plot barchart
barplot(performance_data$Rsquared, names.arg = performance_data$Models, col = "red"

save.image(file = 'Data_mining_portfolio3(C).Rdata')


# CREATE A COPY OF MODEL
svr_linear01 <- svr_linear
print(svr_linear01)

# HYPERPARAMETER TUNING
ctrl <- trainControl(method = "cv", number = 10)  # 10-fold cross-validation


tuneGrid <- expand.grid(C = c(0.01, 0.1, 1, 10, 100))

install.packages("kernlab")
library(kernlab)
svr_linear_model <- train(SALE_PRC ~ .,
                   data = train,
                   method = "svmLinear",
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

Fig. 20: Graphical Evaluation of models



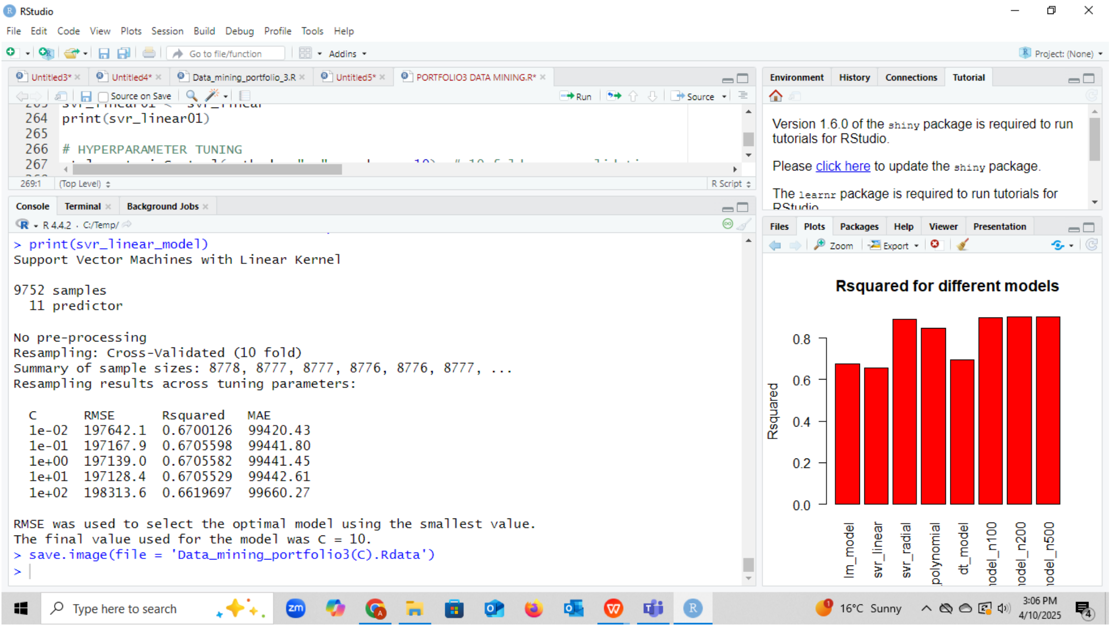
```
# HYPERPARAMETER TUNING
ctrl <- trainControl(method = "cv", number = 10)  # 10-fold cross-validation


tuneGrid <- expand.grid(C = c(0.01, 0.1, 1, 10, 100))

install.packages("kernlab")
library(kernlab)
svr_linear_model <- train(SALE_PRC ~ .,
                   data = train,
                   method = "svmLinear",
                   trControl = ctrl, 
                   tuneGrid = tuneGrid)

print(svr_linear_model)

# Create a data frame with the provided values
```

```
  display list redraw incomplete
4: In doTryCatch(return(expr), name, parentenv, handler) :
  display list redraw incomplete
5: In doTryCatch(return(expr), name, parentenv, handler) :
  display list redraw incomplete
> barplot(performance_data$Rsquared, names.arg = performance_data$Models, col = "red", ma
in = "Rsquared for different models", ylab = "Rsquared", las = 2)
> save.image(file = 'Data_mining_portfolio3(C).Rdata')
> 
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

Fig. 21: Hyperparameter tuning



```
print(svr_linear01)

# HYPERPARAMETER TUNING
```

```
> print(svr_linear_model)
Support Vector Machines with Linear Kernel

9752 samples
  11 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 8778, 8777, 8777, 8776, 8776, 8777, ...
Resampling results across tuning parameters:

  C      RMSE      Rsquared   MAE
  1e-02  197642.1  0.6700126  99420.43
  1e-01  197167.9  0.6705598  99441.80
  1e+00  197139.0  0.6705582  99441.45
  1e+01  197128.4  0.6705529  99442.61
  1e+02  198313.6  0.6619697  99660.27

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was C = 10.
> save.image(file = 'Data_mining_portfolio3(C).Rdata')
> 
```

Version 1.6.0 of the shiny package is required to run tutorials for RStudio.

Please click here to update the shiny package.

The learnr package is required to run tutorials for RStudio.

Fig. 22: Prediction with optimised support vector regression linear model

CHAPTER 5

DISCUSSION

The primary goal of this project is to predict house sale prices based on attributes such as size, distance from key areas, special features, age of property, and other relevant property characteristics. Predictive accuracy is key to supporting pricing decisions for buyers, sellers, and real estate professionals. The real estate industry thrives on accurate property valuation. In this project, eight (8) regression models were explored to predict house sale prices based on historical housing data. The models were evaluated using Mean Squared Error (MSE) as the primary metric, due to its sensitivity to large errors . Among the eight models developed, Support Vector Regression with a linear kernel emerged as the best performing model with the lowest MSE, making it the most suitable for business deployment. To ensure fair comparison, all models were train on the same training set and validated on the same test set for uniform and unbias evaluation. A vital phase in this project is the evaluation of all predictive models. The goal was not just to identify the most accurate model, but to select the one that would provide the most consistent and reliable predictions in a real-world house sale price setting. Hyperparameter was performed using grid search on support vector regression linear model to ensure the model performs at its best as shown in Fig.....

In the real estate industry, pricing is everything. A predictive model that accurately estimates house sale prices is not just a data science tool , it is a growth driver.

Deploying the support vector regression linear model would bring well defined business benefits such as risk reduction, revenue optimisation, efficiency in decision making, market intelligence and strategic planning, client confidence, promote stability across market changes among others.

CHAPTER 6

CONCLUSION

In conclusion, this project successfully demonstrates the application of regression modeling techniques in predicting house prices using machine learning algorithms. By following the structured Cross Industry Standard Process for Data Mining (CRISP-DM) methodology, each phase of the project from understanding the business problem to evaluating the predictive models was well addressed. A total of eight (8) machine learning models were developed and evaluated based on their performance in predicting house sale prices using a set of relevant explanatory variables. Among these, the support vector regression model proved to be the most effective, delivering superior accuracy compared to the other models with respect to having the least mean squared error. This outcome brings to light the potential of advanced regression techniques in real estate price prediction and validate the importance of careful model selection and evaluation in building reliable predictive systems.

REFERENCES

Awad, M., Khanna, R., Awad, M. and Khanna, R., 2015. Support vector regression. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, pp.67-80.

Basysyar, F.M. and Dwilestari, G., 2022. House price prediction using exploratory data analysis and machine learning with feature selection. *Acadlore Transactions on AI and Machine Learning*, *1*(1), pp.11-21.

Fahrmeir, L., Kneib, T., Lang, S. and Marx, B.D., 2022. Regression models. In *Regression: Models, methods and applications* (pp. 23-84). Berlin, Heidelberg: Springer Berlin Heidelberg

Foryś, I., 2022. Machine learning in house price analysis: regression models versus neural networks. *Procedia Computer Science*, *207*, pp.435-445.

Ho, W.K., Tang, B.S. and Wong, S.W., 2021. Predicting property prices with machine learning algorithms. *Journal of Property Research*, *38*(1), pp.48-70.

Kang, Y., Zhang, F., Peng, W., Gao, S., Rao, J., Duarte, F. and Ratti, C., 2021. Understanding house price appreciation using multi-source big geo-data and machine learning. *Land use policy*, *111*, p.104919.

Kuvalekar, A., Manchewar, S., Mahadik, S. and Jawale, S., 2020, April. House price forecasting using machine learning. In *Proceedings of the 3rd international conference on advances in science & technology (ICAST)*.

Manasa, J., Gupta, R. and Narahari, N.S., 2020, March. Machine learning based predicting house prices using regression techniques. In *2020 2nd International conference on innovative mechanisms for industry applications (ICIMIA)* (pp. 624-630). IEEE.

Monika, R., Nithyasree, J., Valarmathi, V., Hemalakshmi, G.R. and Prakash, N.B., 2021. House price forecasting using machine learning methods. *Turkish Journal of Computer and Mathematics Education*, *12*(11), pp.3624-3632.

Pardoe, I., 2020. *Applied regression modeling*. John Wiley & Sons.

Rigatti, S.J., 2017. Random forest. *Journal of Insurance Medicine*, *47*(1), pp.31-39.

Satish, G.N., Raghavendran, C.V., Rao, M.S. and Srinivasulu, C., 2019. House price prediction using machine learning. *Journal of Innovative Technology and Exploring Engineering*, *8*(9), pp.717-722.

Sharma, S., Arora, D., Shankar, G., Sharma, P. and Motwani, V., 2023, February. House price prediction using machine learning algorithm. In *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 982-986). IEEE.

Truong, Q., Nguyen, M., Dang, H. and Mei, B., 2020. Housing price prediction via improved machine learning techniques. *Procedia Computer Science*, *174*, pp.433-442.

Yu, Y., Lu, J., Shen, D. and Chen, B., 2021. Research on real estate pricing methods based on data mining and machine learning. *Neural Computing and Applications*, *33*, pp.3925-3937.

Zhang, Z., 2016. Decision tree modeling using R. *Annals of translational medicine*, *4*(15), p.275.

Zulkifley, N.H., Rahman, S.A., Ubaidullah, N.H. and Ibrahim, I., 2020. House price prediction using a machine learning model: a survey of literature. *International Journal of Modern Education and Computer Science*, *12*(6), pp.46-54.