Apdian, Jose Angelo M.
Labrado, Joshua Reynald Q.

DATA SCIENCE 4
GROUP 24

# S.A CHAPTER PROJECT

## *(CART & Decision Tree)*

### 1.) Loading Necessary Libraries and Dataset

RealEstate.R

Source on Save

```r
# Load required packages
library(caret)
library(rpart)
library(rpart.plot)
library(dplyr)
library(Metrics)
library(mlr)
library(ggplot2)
library (plotly)
library(magrittr)
library(caTools)
library(ggcorrplot)
library(corrplot)

# Load the dataset
setwd("D:\\ACADEMICS\\3RD YEAR COLLEGE\\3RD TERM\\Data Science 4\\Module 1\\SA")
data <- read.csv("D:\\ACADEMICS\\3RD YEAR COLLEGE\\3RD TERM\\Data Science 4\\Module 1\\SA\\Housing.csv")
#display
head(data)
dim(data)
```

```
> # Load the dataset
> setwd("D:\\ACADEMICS\\3RD YEAR COLLEGE\\3RD TERM\\Data Science 4\\Module 1\\SA")
> data <- read.csv("D:\\ACADEMICS\\3RD YEAR COLLEGE\\3RD TERM\\Data Science 4\\Module 1\\SA\\Housing.csv")
> #display
> head(data)
     price area bedrooms bathrooms stories mainroad guestroom basement hotwaterheating airconditioning parking
1 13300000 7420        4         2       3      yes        no       no              no             yes       2
2 12250000 8960        4         4       4      yes        no       no              no             yes       3
3 12250000 9960        3         2       2      yes        no      yes              no              no       2
4 12215000 7500        4         2       2      yes        no      yes              no             yes       3
5 11410000 7420        4         1       2      yes       yes      yes              no             yes       2
6 10850000 7500        3         3       1      yes        no      yes              no             yes       2
  prefarea furnishingstatus
1      yes        furnished
2       no        furnished
3      yes   semi-furnished
4      yes        furnished
5       no        furnished
6      yes   semi-furnished
> dim(data)
[1] 545  13
> 
```

First step is we loaded the necessary libraries to be used in this model. Importantly, we utilized rpart to build classification and regression trees, as well as other plot packages for visualization.

We also loaded the source dataset which is specifically from Kaggle with the title " Housing Prices Dataset" by M YASSER H. The following dataset includes various relevant housing variables from price to furnishing status. We figured that price will act as our dependent variable while the other remaining variables are independent.

## 2.) Data Preparation

```r
#converting other variables to numeric form
data$mainroad <- as.numeric(factor(data$mainroad))
data$guestroom <- as.numeric(factor(data$guestroom))
data$basement <- as.numeric(factor(data$basement))
data$hotwaterheating <- as.numeric(factor(data$hotwaterheating))
data$airconditioning <- as.numeric(factor(data$airconditioning))
data$furnishingstatus <- as.numeric(factor(data$furnishingstatus))

#Removing irrelevant variables
data_clean <-data%>%
    select(-c(prefarea))
head(data_clean)
summary(data_clean)
```

```
Console   Terminal ×   Background Jobs ×
R 4.2.2 · D:/ACADEMICS/3RD YEAR COLLEGE/3RD TERM/Data Science 4/Module 1/SA/
> #converting other variables to numeric form
> data$mainroad <- as.numeric(factor(data$mainroad))
> data$guestroom <- as.numeric(factor(data$guestroom))
> data$basement <- as.numeric(factor(data$basement))
> data$hotwaterheating <- as.numeric(factor(data$hotwaterheating))
> data$airconditioning <- as.numeric(factor(data$airconditioning))
> data$furnishingstatus <- as.numeric(factor(data$furnishingstatus))
>
> #Removing irrelevant variables
> data_clean <-data%>%
+     select(-c(prefarea))
> head(data_clean)
     price area bedrooms bathrooms stories mainroad guestroom basement hotwaterheating airconditioning parking furnishingstatus
1 13300000 7420        4         2       3        2         1        1               1               2       2                1
2 12250000 8960        4         4       4        2         1        1               1               2       3                1
3 12250000 9960        3         2       2        2         1        2               1               1       2                2
4 12215000 7500        4         2       2        2         1        2               1               2       3                1
5 11410000 7420        4         1       2        2         2        2               1               2       2                1
6 10850000 7500        3         3       1        2         1        2               1               2       2                2
>
```

In this step, we cleaned the data by converting variable values that are in character to a numeric format. We also removed 1 irrelevant variable which is the prefarea.

Apdian, Jose Angelo M.
Labrado, Joshua Reynald Q.


## 3.) Data Splitting

```r


#data splitting
create_split <- function(data_clean, size = 0.8, train = TRUE) {

  n_row = nrow(data_clean)
  total_row = size * n_row
  train_sample <- 1: total_row

  if (train ==TRUE){
    return(data_clean[train_sample, ])
  } else {
    return(data_clean[-train_sample, ])
  }

}

#Assigning of train and test data
train_set <- create_split(data_clean, 0.8, train=TRUE)
test_set <- create_split(data_clean, 0.8, train=FALSE)
dim(train_set)
dim(test_set)

```

```
> #Assigning of train and test data
> train_set <- create_split(data_clean, 0.8, train=TRUE)
> test_set <- create_split(data_clean, 0.8, train=FALSE)
> dim(train_set)
[1] 436  12
> dim(test_set)
[1] 109  12
>
```

In this step, we split the data into 80-20 ratio. 80% percent is allocated for training, while the remaining 20% is for testing set.

As you can see, the dimension for training set is larger in size as we allocated majority (80%) of the data into it with 436 rows and 12 columns. For test set, it has 109 rows and 12 columns.

Apdian, Jose Angelo M.
Labrado, Joshua Reynald Q.



## 4.) Building a Decision Tree, Prediction Testing, Confusion Matrix, and Accuracy

```r

#Decision tree creation
tree = rpart(train_set$price~., data=train_set)
rpart.plot(tree)


#Test prediction
predict_price <- predict(tree, test_set)
table_price <-table(test_set$price, predict_price)
print(table_price)

```

```
Console   Terminal ×   Background Jobs ×
R 4.2.2 · D:/ACADEMICS/3RD YEAR COLLEGE/3RD TERM/Data Science 4/Module 1/SA/
> predict_price <- predict(tree, test_set)
> table_price <-table(test_set$price, predict_price)
> print(table_price)
         predict_price
          4060269.23076923 4639141.97530864 4673433.33333333 4828975.6097561
  1750000                3                0                0               0
  1767150                1                0                0               0
  1820000                1                0                0               0
  1855000                1                0                0               0
  1890000                2                0                0               0
  1960000                1                0                0               0
  2100000                3                0                0               0
  2135000                1                0                0               0
  2233000                0                1                0               0
  2240000                1                0                0               0
  2275000                3                0                0               0
  2310000                1                0                0               0
  2345000                1                0                0               0
  2380000                2                0                0               1
  2408000                1                0                0               0
  2450000                4                1                1               0
  2485000                1                1                0               0
  2520000                3                0                0               0
  2590000                1                1                0               1
  2604000                1                0                0               0
  2653000                2                0                0               0
  2660000                6                0                0               1
  2695000                1                0                0               0
  2730000                1                0                1               0
  2800000                2                0                0               0
  2835000                2                1                0               0
  2852500                0                1                0               0
  2870000                2                1                1               0
  2940000                5                2                1               0
  2961000                1                0                0               0
  2975000                0                1                0               0
  3003000                0                0                1               0
  3010000                5                2                0               0
  3045000                1                0                0               0
  3080000                2                2                0               0
  3087000                0                0                1               0
  3115000                3                0                0               0
```

```
  3290000                1                1                0
> accuracy_Test <- sum(diag(table_price)) / sum(table_price)
> print(paste('Accuracy for test', accuracy_Test))
[1] "Accuracy for test 0.0275229357798165"
> 
```

In this section, we ran a prediction test using price as the dependent variable. The following table shows the statistical probability of price prediction. Accuracy test result was also shown.

## 5. Hyper Parameter Fine Tuning and Result

```r


#Hyperparameter Tuning
accuracy_tune <- function(tree){
  predict_unseen <- predict(tree, test_set)
  table_mat <- table(test_set$parking, predict_unseen)
  accuracy_Test <-sum(diag(table_mat)) / sum(table_mat)
  accuracy_Test

}
control <- rpart.control (minsplit = 5, minbucket = round(5 /3), maxdepth = 3, cp = 0)


#Fine-Tuned results
tune_fit <-rpart(price~., data = train_set, control = control)
accuracy_tune(tune_fit)
rpart.plot(tune_fit)

```

```
Console   Terminal ×   Background Jobs ×
R 4.2.2 · D:/ACADEMICS/3RD YEAR COLLEGE/3RD TERM/Data Science 4/Module 1/SA/
+
+ }
> control <- rpart.control (minsplit = 5, minbucket = round(5 /3), maxdepth = 3, cp = 0)
> #Fine-Tuned results
> tune_fit <-rpart(price~., data = train_set, control = control)
> accuracy_tune(tune_fit)
[1] 0.6055046
> rpart.plot(tune_fit)
>
```

***RESULT OF ACCURACY TUNE:***

0.6055045 or 60%

**DECISION TREE:**