Module 3 CT Option 2

Imaobong Ubong Inyang

3/3/2022

# Predicting The Prices of Used Toyota Corolla Car (Regression Trees)

The process of predicting future outcomes from data using statistical methods is called prediction analysis. Predictive models are designed to carry out those predictions by applying certain algorithms. One example of such a model is the Regression Tree. A regression tree (RT) model, unlike a classification tree (used in classifying categorical variables), is used in predicting numerical outcome variables.

## Problem

The ToyotaCorolla.csv file contains information on used cars (Toyota Corolla) which were on sale during the late summer of 2004 in the Netherlands. The objective here is to apply the RT model to predict the price of a used Toyota Corolla based on design specifications.

## Loading the ToyotaCorolla dataset and assign to a data frame

The code below demonstrates loading the data and assigning it to a data frame, usedcars.df.

usedcars.df <- read.csv("ToyotaCorolla.csv", header = TRUE) # load ToyotaCorolla.csv data

# Data Exploration Analysis

The data exploration phase is the initial phase of exploring the dataset to gain more insight into the data. Here, it is split into two parts: descriptive analysis and data visualization.

## Find the size or dimension of the data frame

The knowledge of the dimension of the dataset is useful when it comes to splitting or partitioning the data into test, train, and validate data. Implementing the code below showed that the dataset has 1436 observations (rows) and 39 attributes, including Price, Age, Kilometers, etc.

dim(usedcars.df) # find dimension of data frame

## [1] 1436 39

## Perform descriptive analysis of data

The descriptive analysis includes summary statistics. Summary statistics are used in determining quantitative and non-quantitative attributes of the dataset. The descriptive was computed as shown below:

summary(usedcars.df) # find summary statistics for each column

## Id Model Price Age\_08\_04   
## Min. : 1.0 Length:1436 Min. : 4350 Min. : 1.00   
## 1st Qu.: 361.8 Class :character 1st Qu.: 8450 1st Qu.:44.00   
## Median : 721.5 Mode :character Median : 9900 Median :61.00   
## Mean : 721.6 Mean :10731 Mean :55.95   
## 3rd Qu.:1081.2 3rd Qu.:11950 3rd Qu.:70.00   
## Max. :1442.0 Max. :32500 Max. :80.00   
## Mfg\_Month Mfg\_Year KM Fuel\_Type   
## Min. : 1.000 Min. :1998 Min. : 1 Length:1436   
## 1st Qu.: 3.000 1st Qu.:1998 1st Qu.: 43000 Class :character   
## Median : 5.000 Median :1999 Median : 63390 Mode :character   
## Mean : 5.549 Mean :2000 Mean : 68533   
## 3rd Qu.: 8.000 3rd Qu.:2001 3rd Qu.: 87021   
## Max. :12.000 Max. :2004 Max. :243000   
## HP Met\_Color Color Automatic   
## Min. : 69.0 Min. :0.0000 Length:1436 Min. :0.00000   
## 1st Qu.: 90.0 1st Qu.:0.0000 Class :character 1st Qu.:0.00000   
## Median :110.0 Median :1.0000 Mode :character Median :0.00000   
## Mean :101.5 Mean :0.6748 Mean :0.05571   
## 3rd Qu.:110.0 3rd Qu.:1.0000 3rd Qu.:0.00000   
## Max. :192.0 Max. :1.0000 Max. :1.00000   
## CC Doors Cylinders Gears Quarterly\_Tax   
## Min. : 1300 Min. :2.000 Min. :4 Min. :3.000 Min. : 19.00   
## 1st Qu.: 1400 1st Qu.:3.000 1st Qu.:4 1st Qu.:5.000 1st Qu.: 69.00   
## Median : 1600 Median :4.000 Median :4 Median :5.000 Median : 85.00   
## Mean : 1577 Mean :4.033 Mean :4 Mean :5.026 Mean : 87.12   
## 3rd Qu.: 1600 3rd Qu.:5.000 3rd Qu.:4 3rd Qu.:5.000 3rd Qu.: 85.00   
## Max. :16000 Max. :5.000 Max. :4 Max. :6.000 Max. :283.00   
## Weight Mfr\_Guarantee BOVAG\_Guarantee Guarantee\_Period  
## Min. :1000 Min. :0.0000 Min. :0.0000 Min. : 3.000   
## 1st Qu.:1040 1st Qu.:0.0000 1st Qu.:1.0000 1st Qu.: 3.000   
## Median :1070 Median :0.0000 Median :1.0000 Median : 3.000   
## Mean :1072 Mean :0.4095 Mean :0.8955 Mean : 3.815   
## 3rd Qu.:1085 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.: 3.000   
## Max. :1615 Max. :1.0000 Max. :1.0000 Max. :36.000   
## ABS Airbag\_1 Airbag\_2 Airco   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:1.0000 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :1.0000 Median :1.0000 Median :1.0000 Median :1.0000   
## Mean :0.8134 Mean :0.9708 Mean :0.7228 Mean :0.5084   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## Automatic\_airco Boardcomputer CD\_Player Central\_Lock   
## Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.00000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.00000 Median :0.0000 Median :0.0000 Median :1.0000   
## Mean :0.05641 Mean :0.2946 Mean :0.2187 Mean :0.5801   
## 3rd Qu.:0.00000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1.0000   
## Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## Powered\_Windows Power\_Steering Radio Mistlamps   
## Min. :0.000 Min. :0.0000 Min. :0.0000 Min. :0.000   
## 1st Qu.:0.000 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.:0.000   
## Median :1.000 Median :1.0000 Median :0.0000 Median :0.000   
## Mean :0.562 Mean :0.9777 Mean :0.1462 Mean :0.257   
## 3rd Qu.:1.000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:1.000   
## Max. :1.000 Max. :1.0000 Max. :1.0000 Max. :1.000   
## Sport\_Model Backseat\_Divider Metallic\_Rim Radio\_cassette   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :0.0000 Median :1.0000 Median :0.0000 Median :0.0000   
## Mean :0.3001 Mean :0.7702 Mean :0.2047 Mean :0.1455   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000 3rd Qu.:0.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## Parking\_Assistant Tow\_Bar   
## Min. :0.000000 Min. :0.0000   
## 1st Qu.:0.000000 1st Qu.:0.0000   
## Median :0.000000 Median :0.0000   
## Mean :0.002786 Mean :0.2779   
## 3rd Qu.:0.000000 3rd Qu.:1.0000   
## Max. :1.000000 Max. :1.0000

For each of the quantitative attributes, we compute the mean, standard deviation, min, max, median, length, missing values and sum of missing values as follows:

data.frame(mean=sapply(usedcars.df[, -c(1, 2, 8, 11)], mean, na.rm=TRUE),  
 sd=sapply(usedcars.df[, -c(1, 2, 8, 11)], sd, na.rm=TRUE),  
 min=sapply(usedcars.df[, -c(1, 2, 8, 11)], min, na.rm=TRUE),  
 max=sapply(usedcars.df[, -c(1, 2, 8, 11)], max, na.rm=TRUE),  
 median=sapply(usedcars.df[, -c(1, 2, 8, 11)], median, na.rm=TRUE),  
 length=sapply(usedcars.df[, -c(1, 2, 8, 11)], length),  
 miss.val=sapply(usedcars.df[, -c(1, 2, 8, 11)], function(x)  
 sum(length(which(is.na(x))))) )

## mean sd min max median length miss.val  
## Price 1.073082e+04 3.626965e+03 4350 32500 9900.0 1436 0  
## Age\_08\_04 5.594708e+01 1.859999e+01 1 80 61.0 1436 0  
## Mfg\_Month 5.548747e+00 3.354085e+00 1 12 5.0 1436 0  
## Mfg\_Year 1.999625e+03 1.540722e+00 1998 2004 1999.0 1436 0  
## KM 6.853326e+04 3.750645e+04 1 243000 63389.5 1436 0  
## HP 1.015021e+02 1.498108e+01 69 192 110.0 1436 0  
## Met\_Color 6.747911e-01 4.686160e-01 0 1 1.0 1436 0  
## Automatic 5.571031e-02 2.294413e-01 0 1 0.0 1436 0  
## CC 1.576856e+03 4.243868e+02 1300 16000 1600.0 1436 0  
## Doors 4.033426e+00 9.526766e-01 2 5 4.0 1436 0  
## Cylinders 4.000000e+00 0.000000e+00 4 4 4.0 1436 0  
## Gears 5.026462e+00 1.885104e-01 3 6 5.0 1436 0  
## Quarterly\_Tax 8.712256e+01 4.112861e+01 19 283 85.0 1436 0  
## Weight 1.072460e+03 5.264112e+01 1000 1615 1070.0 1436 0  
## Mfr\_Guarantee 4.094708e-01 4.919075e-01 0 1 0.0 1436 0  
## BOVAG\_Guarantee 8.955432e-01 3.059588e-01 0 1 1.0 1436 0  
## Guarantee\_Period 3.815460e+00 3.011025e+00 3 36 3.0 1436 0  
## ABS 8.133705e-01 3.897496e-01 0 1 1.0 1436 0  
## Airbag\_1 9.707521e-01 1.685594e-01 0 1 1.0 1436 0  
## Airbag\_2 7.228412e-01 4.477515e-01 0 1 1.0 1436 0  
## Airco 5.083565e-01 5.001043e-01 0 1 1.0 1436 0  
## Automatic\_airco 5.640669e-02 2.307857e-01 0 1 0.0 1436 0  
## Boardcomputer 2.945682e-01 4.560072e-01 0 1 0.0 1436 0  
## CD\_Player 2.186630e-01 4.134834e-01 0 1 0.0 1436 0  
## Central\_Lock 5.800836e-01 4.937169e-01 0 1 1.0 1436 0  
## Powered\_Windows 5.619777e-01 4.963167e-01 0 1 1.0 1436 0  
## Power\_Steering 9.777159e-01 1.476575e-01 0 1 1.0 1436 0  
## Radio 1.462396e-01 3.534693e-01 0 1 0.0 1436 0  
## Mistlamps 2.569638e-01 4.371115e-01 0 1 0.0 1436 0  
## Sport\_Model 3.001393e-01 4.584780e-01 0 1 0.0 1436 0  
## Backseat\_Divider 7.701950e-01 4.208539e-01 0 1 1.0 1436 0  
## Metallic\_Rim 2.047354e-01 4.036487e-01 0 1 0.0 1436 0  
## Radio\_cassette 1.455432e-01 3.527705e-01 0 1 0.0 1436 0  
## Parking\_Assistant 2.785515e-03 5.272278e-02 0 1 0.0 1436 0  
## Tow\_Bar 2.778552e-01 4.480976e-01 0 1 0.0 1436 0

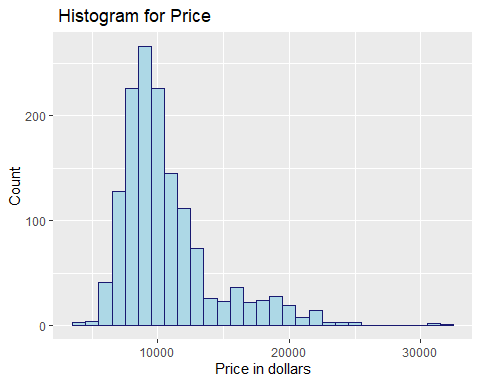
Note, the na.rm = TRUE removes the missing values and all categorical variables were negatively removed using the sapply() function.

## Data visualization

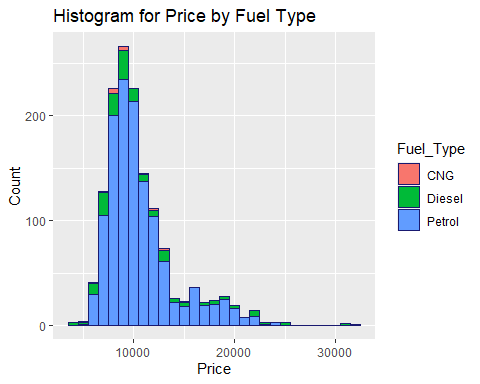
Histogram and Box plot for single variable Price by implementing the ggplot as follows:

library(ggplot2) # instantiate ggplot

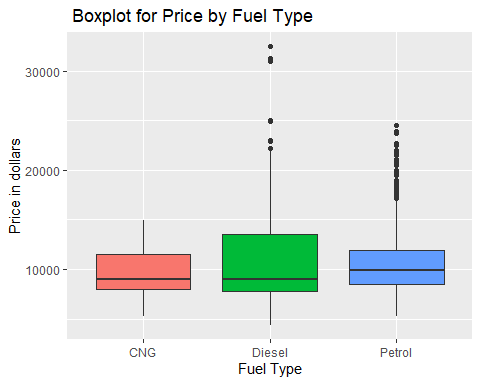
ggplot(usedcars.df) + geom\_histogram(aes(x = Price), binwidth = 1000, color = "midnightblue", fill = "lightblue") +  
 labs(title=" Histogram for Price",x="Price in dollars", y = "Count") # Histogram for variable Price



# Histogram for variable Price by Fuel Type   
  
ggplot(usedcars.df) + geom\_histogram(aes(x = Price, fill = Fuel\_Type), binwidth = 1000, color = "midnightblue") + labs(title = "Histogram for Price by Fuel Type", x = "Price", y = "Count")



ggplot(usedcars.df) + geom\_boxplot(aes(x = as.factor(Fuel\_Type), y = Price, fill = Fuel\_Type)) + xlab("Fuel Type") + theme(legend.position="none") +  
 labs(title=" Boxplot for Price by Fuel Type",y="Price in dollars") # boxplot for variable Price by Fuel\_Type



# Regression Tree Model Design

For this assignment, we are interested in predicting Price (outcome variable) based on 15 attributes of the dataset. Because Price is a continuous variable, this is a regression problem and requires the use of a regression tree. The regression tree for this dataset was built using a training set of 600 records.

## Instantiating the regression tree model

library(rpart) # for regression tree model  
library(rpart.plot) # for visualizing the model

## Data Partitioning

Split the data into training (60%), and validation (40%) datasets.

set.seed(22) # seed set to 22  
train.index <- sample(c(1:dim(usedcars.df)[1]), dim(usedcars.df)[1]\*0.6) # split 60% for train data  
train.df <- usedcars.df[train.index, ]  
valid.df <- usedcars.df[-train.index, ] # validate data partitioned

## Regression tree model specifications

Run a regression tree (RT) with the following characteristics:

* Outcome variable Price.
* Predictors Age\_08\_04, KM, Fuel\_Type, HP, Automatic, Doors, Quarterly\_Tax, Mfr\_Guarantee, Guarantee\_Period, Airco, Automatic\_Airco, CD\_Player, Powered\_Windows, Sport\_Model, and Tow\_Bar.
* The minimum number of records in a terminal node is s 1.
* The maximum number of tree levels to 30.
* cp = 0.001, to make the run least restrictive.

## Build RT model

# tree model  
  
rt <- rpart(Price ~ Age\_08\_04 + KM + Fuel\_Type + HP + Automatic + Doors + Quarterly\_Tax +  
 Mfr\_Guarantee + Guarantee\_Period + Airco + Automatic\_airco + CD\_Player +  
 Powered\_Windows + Sport\_Model + Tow\_Bar,   
 data = train.df,  
 method = "anova", minbucket = 1, maxdepth = 30, cp = 0.001)   
  
  
prp(rt, tweak = 1.15, box.col = ifelse(rt$frame$var == "<leaf>", 'gray', 'white')) # plot tree

Diagram

Description automatically generated

**Interpretation of RT Model result**

The RT model was designed to predict the price from a given Toyota Corolla used car dataset according to the specification provided in Module 3 assignment instructions. The predictor(s), that is, the decision node information was used for “dropping” the record down the tree until it reaches a terminal node (gray ovals). For example, to predict the price of a used car with Age = 70 and KM > 116000 (Accumulated Kilometers on the odometer), it was dropped down the tree until the terminal node that has the value $7132 was reached.