Project 1

Brief Problem Description:

Lightning McQueen is lost. He is now in Radiator Springs where he found a new home. He met Mator, a new friend of his, and Mator wants to start a new used car resale business. But Mator does not know how to set a price to a car. But Mator has received an incomplete data set about used car sales. Mator reached out to us; Marcus, Tai, and Simon because he heard we are some of the best business analytics students at Seattle University.

Objective:

Mator is asking us to use our expertise to predict whether a used car price should be high or low.

Data Set:

As scholars at the Albers school of business we see some issues with this incomplete data set. We say the data set is incomplete because it does not have exact prices. Instead we will use high vs low to determine how to set the price (presumably based on prevailing market rates) and market used cars. High priced ones will be marketed differently from low priced ones. While the price for some used cars are obvious, others are not. From the data set we will build and select an appropriate model for the company and predict the price of the car if it would be higher or lower for a new customer.

The reason we choose the positive to be low is because the description of our assignment it states that normally this data is set to the minority class. Which in this case is the low price norm.

```
library(caret)
```

```
Loading required package: ggplot2
```

```
Loading required package: lattice
```

```
library(ggplot2)
library(lattice)
library(dplyr)
```

```

Attaching package: 'dplyr'
```

```
The following objects are masked from 'package:stats':
```

```

    filter, lag
```

```
The following objects are masked from 'package:base':
```

```

    intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ forcats   1.0.0     ✓ stringr   1.5.0
✓ lubridate 1.9.2     ✓ tibble    3.2.1
✓ purrr     1.0.1     ✓ tidyr     1.3.0
✓ readr     2.1.4
```

```
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
✖ purrr::lift()   masks caret::lift()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(rpart)
library(rpart.plot)
library(forecast)
```

```
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
```

I. Data

1. Clean the Data

We need to clean the data to get it ready for analysis.

```
cars <- read.csv("car_train_class_12.csv", header = TRUE)
head(cars, 10)
```

```
   X.1       X               vin back_legroom       body_type    cabin
1    1  693300 KM8J3CALXMU301004         38.2 SUV / Crossover
2    2  186812 1G1JD6SB2L4136775         34.6       Hatchback
3    3  951451 5J6RW2H96LL022148         40.4 SUV / Crossover
4    4 1568313 KM8K2CAA5LU562934         34.6 SUV / Crossover
5    5 1408062 5FNYF8H99LB008980         39.6 SUV / Crossover
6    6  924769 1FTEW1EP3LKE58267         43.6    Pickup Truck
7    7 1538348 1FT7W2BT9KEG46773         43.6    Pickup Truck Crew Cab
8    8   81900 4S4BSANC3K3294390         38.1           Wagon
```

```
9    9 1551017 1G1105S31LU105922         39.8          Sedan
10  10 1506441 1N4AZ1CPXLC301005           NA       Hatchback
                 city daysonmarket dealer_zip     engine_cylinders
1           Chantilly           25      20151                   I4
2             Pawling           10      12564                   I4
3           Knoxville           50      37912                   I4
4  Inver Grove Heights          134      55077                   I4
5             Lincoln           197      68516                   V6
6            Bay City           56      48706                   V6
7               Miami           29      33126         V8 Biodiesel
8               Troy            75      12180                   H4
9     Fort Lauderdale           33      33304 V6 Flex Fuel Vehicle
10     Port Charlotte          219      33980
   engine_displacement          engine_type        exterior_color fleet
1                 2400                   I4             Dusk Blue
2                 1400                   I4    Cajun Red Tintcoat
3                 1500                   I4   Crystal Black Pearl
4                 2000                   I4            Pulse Red
5                 3500                   V6   Obsidian Blue Pearl
6                 3500                   V6         Silver Spruce False
7                 6700         V8 Biodiesel Ingot Silver Metallic  True
8                 2500                   H4     Crimson Red Pearl  True
9                 3600 V6 Flex Fuel Vehicle   Silver Ice Metallic  True
10                  NA                      2-Tone White/Black
   frame_damaged franchise_dealer franchise_make front_legroom fuel_tank_volume
1                           True        Hyundai          41.5             16.4
2                           True      Chevrolet          41.8             12.2
3                           True          Honda          41.3             14.0
4                           True        Hyundai          41.5             13.2
5                           True          Honda          40.9             19.5
6          False            True           Ford          43.9             26.0
7          False           False                         43.9             34.0
8          False            True         Subaru          42.9             18.5
9          False            True      Chevrolet          45.8             18.5
10                          True         Nissan                            NA
          fuel_type has_accidents height horsepower            interior_color
1          Gasoline                65.0        181             Brown (Beige)
2          Gasoline                59.7        138                 Jet Black
3          Gasoline                66.5        190                     Black
4          Gasoline                61.6        147                      None
5          Gasoline                72.2        280                      Gray
6          Gasoline        False   77.2        375 Gray (Medium Light Camel)
7         Biodiesel        False   81.5        450                     Other
8          Gasoline        False   66.1        175               Slate Black
9  Flex Fuel Vehicle        False   58.9        305                 Jet Black
10         Electric                  NA         NA                     Black
   is_cpo is_new is_oemcpo length listed_date listing_color listing_id
1   False   True     False  176.4  2020-08-16          BLUE  279447562
2   False   True     False  159.8  2020-08-30           RED  280796643
3   False   True     False  182.1  2020-07-22         BLACK  277235352
4   False   True     False  164.0  2020-04-29           RED  271147110
5   False   True     False  190.5  2020-02-26          BLUE  266916841
6   False   True     False  231.9  2020-07-16        SILVER  276750193
7   False  False     False  250.0  2020-08-12        SILVER  279042831
8    True  False     False  189.9  2020-06-26           RED  275081413
9   False  False     False  201.3  2020-08-09        SILVER  278777141
10  False   True     False     NA  2020-02-04         WHITE  264965475

major_options
1                                                                  ['Leather Seats', 'Alloy Wheels', 'Bluetooth', 'Backup Camera', 'Remote Start', 'Blind
Spot Monitoring', 'Cargo Package', 'Heated Seats']
2                                                                                                                  ['Alloy Wheels', 'Backup Camera',
'Android Auto', 'CarPlay', 'Convenience Package']
3                                                                                                  ['Leather Seats', 'Sunroof/Moonroof', 'Navigation
System', 'Bluetooth', 'Backup Camera', 'Remote Start']
4
5                                                                                  ['Sunroof/Moonroof', 'Navigation System', 'Adaptive Cruise Control',
'Alloy Wheels', 'Backup Camera', 'Heated Seats']
6                                                                                                                  ['Navigation System', 'Alloy
Wheels', 'Bluetooth', 'Backup Camera', 'Remote Start']
7
['Backup Camera']
8                ['Leather Seats', 'Sunroof/Moonroof', 'Navigation System', 'Adaptive Cruise Control', 'Alloy Wheels', 'Bluetooth', 'Backup Camera',
'Blind Spot Monitoring', 'Heated Seats', 'CarPlay']
9             ['Leather Seats', 'Navigation System', 'Alloy Wheels', 'Bluetooth', 'Backup Camera', 'Remote Start', 'Blind Spot Monitoring', 'Parking
Sensors', 'Heated Seats', 'Android Auto', 'CarPlay']
10 ['Navigation System', 'Adaptive Cruise Control', 'Alloy Wheels', 'Technology Package', 'Bluetooth', 'Backup Camera', 'Remote Start', 'Blind Spot
Monitoring', 'Heated Seats', 'Cold Weather Package']
   make_name maximum_seating mileage       model_name owner_count
1    Hyundai               5       5           Tucson           0
2  Chevrolet               5      NA            Sonic           0
3      Honda               5       0             CR-V           0
4    Hyundai               5       4             Kona           0
5      Honda               5       4         Passport           0
6       Ford               6      10            F-150           0
7       Ford               6   28096 F-250 Super Duty           1
8     Subaru               5   23388          Outback           1
9  Chevrolet               5   18683           Impala           1
10    Nissan              NA     120             LEAF           0
                power salvage savings_amount seller_rating  sp_id
1  181 hp @ 6,000 RPM                     0      3.789474 370177
2  138 hp @ 4,900 RPM                     0      4.333333 344283
3  190 hp @ 5,600 RPM                     0      4.200000  59357
4  147 hp @ 6,200 RPM                     0      4.000000 300069
5  280 hp @ 6,000 RPM                     0      4.548387  59048
6  395 hp @ 5,750 RPM   False             0      4.142857  56402
7  450 hp @ 2,800 RPM   False         12787      2.398601 382122
8  175 hp @ 5,800 RPM   False           229      3.739130 274312
```

```
 9  305 hp @ 6,800 RPM   False            473      3.739130 397830
 10                                        0      3.241379 276028
                            sp_name theft_title                torque
 1                 Hyundai of Chantilly             175 lb-ft @ 4,000 RPM
 2            Ingersoll Auto of Pawling             148 lb-ft @ 2,500 RPM
 3                 Rusty Wallace Honda              179 lb-ft @ 2,000 RPM
 4                 Inver Grove Hyundai              132 lb-ft @ 4,500 RPM
 5                    Honda of Lincoln              262 lb-ft @ 4,700 RPM
 6                      Hagen Ford Inc       False 400 lb-ft @ 4,500 RPM
 7             Car Factory Outlet Miami       False 935 lb-ft @ 1,800 RPM
 8                      Carbone Subaru        False 174 lb-ft @ 4,000 RPM
 9  Grieco Chevrolet of Fort Lauderdale       False 264 lb-ft @ 5,300 RPM
 10                       Harbor Nissan
    transmission        transmission_display trimId           trim_name
 1             A                   Automatic t93993         Limited AWD
 2             A           6-Speed Automatic t85498    LT Hatchback FWD
 3             A                   Automatic t89603         Touring AWD
 4                                 t87010             SEL AWD
 5             A           9-Speed Automatic t90159         Touring AWD
 6             A                   Automatic t87739   XLT SuperCrew 4WD
 7             A           6-Speed Automatic t78394 Lariat Crew Cab 4WD
 8             A                   Automatic t82803    2.5i Limited AWD
 9             A 6-Speed Automatic Overdrive t86811         Premier FWD
 10            A                   Automatic t90999              SV FWD
    wheel_system wheel_system_display wheelbase width year power_hp power_rpm
 1           AWD      All-Wheel Drive     105.1  72.8 2021      181      6000
 2           FWD    Front-Wheel Drive      99.4  68.3 2020      138      4900
 3           AWD      All-Wheel Drive     104.7  73.0 2020      190      5600
 4           AWD      All-Wheel Drive     102.4  70.9 2020      147      6200
 5           AWD      All-Wheel Drive     111.0  78.6 2020      280      6000
 6           4WD     Four-Wheel Drive     145.0  96.8 2020      395      5750
 7           4WD     Four-Wheel Drive     159.8 105.9 2019      450      2800
 8           AWD      All-Wheel Drive     108.1  81.3 2019      175      5800
 9           FWD    Front-Wheel Drive     111.7  84.3 2020      305      6800
 10          FWD    Front-Wheel Drive        NA    NA 2020       NA        NA
    torque_lbft torque_rpm price_nom
 1          175       4000         0
 2          148       2500         0
 3          179       2000         0
 4          132       4500         0
 5          262       4700         0
 6          400       4500         0
 7          935       1800         0
 8          174       4000         0
 9          264       5300         0
 10          NA         NA         0
```

```
str(cars)
```

```
'data.frame':   29233 obs. of  59 variables:
 $ X.1                : int  1 2 3 4 5 6 7 8 9 10 ...
 $ X                  : int  693300 186812 951451 1568313 1408062 924769 1538348 81900 1551017 1506441 ...
 $ vin                : chr  "KM8J3CALXMU301004" "1G1JD6SB2L4136775" "5J6RW2H96LL022148" "KM8K2CAA5LU562934" ...
 $ back_legroom       : num  38.2 34.6 40.4 34.6 39.6 43.6 43.6 38.1 39.8 NA ...
 $ body_type          : chr  "SUV / Crossover" "Hatchback" "SUV / Crossover" "SUV / Crossover" ...
 $ cabin              : chr  "" "" "" "" ...
 $ city               : chr  "Chantilly" "Pawling" "Knoxville" "Inver Grove Heights" ...
 $ daysonmarket       : int  25 10 50 134 197 56 29 75 33 219 ...
 $ dealer_zip         : chr  "20151" "12564" "37912" "55077" ...
 $ engine_cylinders   : chr  "I4" "I4" "I4" "I4" ...
 $ engine_displacement : int  2400 1400 1500 2000 3500 3500 6700 2500 3600 NA ...
 $ engine_type        : chr  "I4" "I4" "I4" "I4" ...
 $ exterior_color     : chr  "Dusk Blue" "Cajun Red Tintcoat" "Crystal Black Pearl" "Pulse Red" ...
 $ fleet              : chr  "" "" "" "" ...
 $ frame_damaged      : chr  "" "" "" "" ...
 $ franchise_dealer   : chr  "True" "True" "True" "True" ...
 $ franchise_make     : chr  "Hyundai" "Chevrolet" "Honda" "Hyundai" ...
 $ front_legroom      : chr  "41.5" "41.8" "41.3" "41.5" ...
 $ fuel_tank_volume   : num  16.4 12.2 14 13.2 19.5 26 34 18.5 18.5 NA ...
 $ fuel_type          : chr  "Gasoline" "Gasoline" "Gasoline" "Gasoline" ...
 $ has_accidents      : chr  "" "" "" "" ...
 $ height             : num  65 59.7 66.5 61.6 72.2 77.2 81.5 66.1 58.9 NA ...
 $ horsepower         : int  181 138 190 147 280 375 450 175 305 NA ...
 $ interior_color     : chr  "Brown (Beige)" "Jet Black" "Black" "None" ...
 $ is_cpo             : chr  "False" "False" "False" "False" ...
 $ is_new             : chr  "True" "True" "True" "True" ...
 $ is_oemcpo          : chr  "False" "False" "False" "False" ...
 $ length             : num  176 160 182 164 190 ...
 $ listed_date        : chr  "2020-08-16" "2020-08-30" "2020-07-22" "2020-04-29" ...
 $ listing_color      : chr  "BLUE" "RED" "BLACK" "RED" ...
 $ listing_id         : int  279447562 280796643 277235352 271147110 266916841 276750193 279042831 275081413 278777141 264965475 ...
 $ major_options      : chr  "['Leather Seats', 'Alloy Wheels', 'Bluetooth', 'Backup Camera', 'Remote Start', 'Blind Spot Monitoring', 'Cargo"|
__truncated__ "['Alloy Wheels', 'Backup Camera', 'Android Auto', 'CarPlay', 'Convenience Package']" "['Leather Seats', 'Sunroof/Moonroof', 'Navigation
System', 'Bluetooth', 'Backup Camera', 'Remote Start']" "" ...
 $ make_name          : chr  "Hyundai" "Chevrolet" "Honda" "Hyundai" ...
 $ maximum_seating    : int  5 5 5 5 5 6 6 5 5 NA ...
 $ mileage            : num  5 NA 0 4 4 ...
 $ model_name         : chr  "Tucson" "Sonic" "CR-V" "Kona" ...
 $ owner_count        : int  0 0 0 0 0 0 1 1 1 0 ...
 $ power              : chr  "181 hp @ 6,000 RPM" "138 hp @ 4,900 RPM" "190 hp @ 5,600 RPM" "147 hp @ 6,200 RPM" ...
 $ salvage            : chr  "" "" "" "" ...
 $ savings_amount     : int  0 0 0 0 0 0 12787 229 473 0 ...
 $ seller_rating      : num  3.79 4.33 4.2 4 4.55 ...
 $ sp_id              : int  370177 344283 59357 300069 59048 56402 382122 274312 397830 276028 ...
 $ sp_name            : chr  "Hyundai of Chantilly" "Ingersoll Auto of Pawling" "Rusty Wallace Honda" "Inver Grove Hyundai" ...
 $ theft_title        : chr  "" "" "" "" ...
 $ torque             : chr  "175 lb-ft @ 4,000 RPM" "148 lb-ft @ 2,500 RPM" "179 lb-ft @ 2,000 RPM" "132 lb-ft @ 4,500 RPM" ...
```

```
 $ transmission       : chr  "A" "A" "A" "" ...
 $ transmission_display: chr  "Automatic" "6-Speed Automatic" "Automatic" "" ...
 $ trimId             : chr  "t93993" "t85498" "t89603" "t87010" ...
 $ trim_name          : chr  "Limited AWD" "LT Hatchback FWD" "Touring AWD" "SEL AWD" ...
 $ wheel_system       : chr  "AWD" "FWD" "AWD" "AWD" ...
 $ wheel_system_display: chr  "All-Wheel Drive" "Front-Wheel Drive" "All-Wheel Drive" "All-Wheel Drive" ...
 $ wheelbase          : num  105.1 99.4 104.7 102.4 111 ...
 $ width              : num  72.8 68.3 73 70.9 78.6 ...
 $ year               : int  2021 2020 2020 2020 2020 2020 2019 2019 2020 2020 ...
 $ power_hp           : int  181 138 190 147 280 395 450 175 305 NA ...
 $ power_rpm          : int  6000 4900 5600 6200 6000 5750 2800 5800 6800 NA ...
 $ torque_lbft        : int  175 148 179 132 262 400 935 174 264 NA ...
 $ torque_rpm         : int  4000 2500 2000 4500 4700 4500 1800 4000 5300 NA ...
 $ price_nom          : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
t(t(names(cars)))
```

```
      [,1]
 [1,] "X.1"
 [2,] "X"
 [3,] "vin"
 [4,] "back_legroom"
 [5,] "body_type"
 [6,] "cabin"
 [7,] "city"
 [8,] "daysonmarket"
 [9,] "dealer_zip"
[10,] "engine_cylinders"
[11,] "engine_displacement"
[12,] "engine_type"
[13,] "exterior_color"
[14,] "fleet"
[15,] "frame_damaged"
[16,] "franchise_dealer"
[17,] "franchise_make"
[18,] "front_legroom"
[19,] "fuel_tank_volume"
[20,] "fuel_type"
[21,] "has_accidents"
[22,] "height"
[23,] "horsepower"
[24,] "interior_color"
[25,] "is_cpo"
[26,] "is_new"
[27,] "is_oemcpo"
[28,] "length"
[29,] "listed_date"
[30,] "listing_color"
[31,] "listing_id"
[32,] "major_options"
[33,] "make_name"
[34,] "maximum_seating"
[35,] "mileage"
[36,] "model_name"
[37,] "owner_count"
[38,] "power"
[39,] "salvage"
[40,] "savings_amount"
[41,] "seller_rating"
[42,] "sp_id"
[43,] "sp_name"
[44,] "theft_title"
[45,] "torque"
[46,] "transmission"
[47,] "transmission_display"
[48,] "trimId"
[49,] "trim_name"
[50,] "wheel_system"
[51,] "wheel_system_display"
[52,] "wheelbase"
[53,] "width"
[54,] "year"
[55,] "power_hp"
[56,] "power_rpm"
[57,] "torque_lbft"
[58,] "torque_rpm"
[59,] "price_nom"
```

We removed unnecessary variables here. We did this because we wanted to choose the quality variables to give us a model that could predict the correct accuracy. So we can have the right business evaluation.

```
#Remove unnecessary variables
cars <- cars[, c(4,14:16,19,21:23,25:28,34,37,39,44,59)]
t(t(names(cars)))
```

```
      [,1]
 [1,] "back_legroom"
 [2,] "fleet"
 [3,] "frame_damaged"
 [4,] "franchise_dealer"
 [5,] "fuel_tank_volume"
 [6,] "has_accidents"
 [7,] "height"
 [8,] "horsepower"
 [9,] "is_cpo"
[10,] "is_new"
[11,] "is_oemcpo"
```

```
[12,] "length"
[13,] "maximum_seating"
[14,] "owner_count"
[15,] "salvage"
[16,] "theft_title"
[17,] "price_nom"
```

```
str(cars)
```

```
'data.frame':   29233 obs. of  17 variables:
 $ back_legroom    : num  38.2 34.6 40.4 34.6 39.6 43.6 43.6 38.1 39.8 NA ...
 $ fleet           : chr  "" "" "" "" ...
 $ frame_damaged   : chr  "" "" "" "" ...
 $ franchise_dealer: chr  "True" "True" "True" "True" ...
 $ fuel_tank_volume: num  16.4 12.2 14 13.2 19.5 26 34 18.5 18.5 NA ...
 $ has_accidents   : chr  "" "" "" "" ...
 $ height          : num  65 59.7 66.5 61.6 72.2 77.2 81.5 66.1 58.9 NA ...
 $ horsepower      : int  181 138 190 147 280 375 450 175 305 NA ...
 $ is_cpo          : chr  "False" "False" "False" "False" ...
 $ is_new          : chr  "True" "True" "True" "True" ...
 $ is_oemcpo       : chr  "False" "False" "False" "False" ...
 $ length          : num  176 160 182 164 190 ...
 $ maximum_seating : int  5 5 5 5 5 6 6 5 5 NA ...
 $ owner_count     : int  0 0 0 0 0 0 1 1 1 0 ...
 $ salvage         : chr  "" "" "" "" ...
 $ theft_title     : chr  "" "" "" "" ...
 $ price_nom       : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
#reorder the data frame
cars <- cars[, c(2:4,6,9:11,15:16,1,5,7:8,12:14,17)]
str(cars)
```

```
'data.frame':   29233 obs. of  17 variables:
 $ fleet           : chr  "" "" "" "" ...
 $ frame_damaged   : chr  "" "" "" "" ...
 $ franchise_dealer: chr  "True" "True" "True" "True" ...
 $ has_accidents   : chr  "" "" "" "" ...
 $ is_cpo          : chr  "False" "False" "False" "False" ...
 $ is_new          : chr  "True" "True" "True" "True" ...
 $ is_oemcpo       : chr  "False" "False" "False" "False" ...
 $ salvage         : chr  "" "" "" "" ...
 $ theft_title     : chr  "" "" "" "" ...
 $ back_legroom    : num  38.2 34.6 40.4 34.6 39.6 43.6 43.6 38.1 39.8 NA ...
 $ fuel_tank_volume: num  16.4 12.2 14 13.2 19.5 26 34 18.5 18.5 NA ...
 $ height          : num  65 59.7 66.5 61.6 72.2 77.2 81.5 66.1 58.9 NA ...
 $ horsepower      : int  181 138 190 147 280 375 450 175 305 NA ...
 $ length          : num  176 160 182 164 190 ...
 $ maximum_seating : int  5 5 5 5 5 6 6 5 5 NA ...
 $ owner_count     : int  0 0 0 0 0 0 1 1 1 0 ...
 $ price_nom       : int  0 0 0 0 0 0 0 0 0 0 ...
```

2. Data for kNN model

Next we get the data ready for kNN model. We will need to drop all the NAs variables inside the data frame.

```
cars_kNN <- drop_na(cars)
# this code drop all the NA variable inside the data frame
```

```
head(cars_kNN, 10)
```

```
   fleet frame_damaged franchise_dealer has_accidents is_cpo is_new is_oemcpo
1                                 True               False   True     False
2                                 True               False   True     False
3                                 True               False   True     False
4                                 True               False   True     False
5                                 True               False   True     False
6  False         False             True         False  False   True     False
7   True         False            False         False  False  False     False
8   True         False             True         False   True  False     False
9   True         False             True         False  False  False     False
10                                True               False   True     False
   salvage theft_title back_legroom fuel_tank_volume height horsepower length
1                            38.2             16.4   65.0        181  176.4
2                            34.6             12.2   59.7        138  159.8
3                            40.4             14.0   66.5        190  182.1
4                            34.6             13.2   61.6        147  164.0
5                            39.6             19.5   72.2        280  190.5
6    False       False         43.6             26.0   77.2        375  231.9
7    False       False         43.6             34.0   81.5        450  250.0
8    False       False         38.1             18.5   66.1        175  189.9
9    False       False         39.8             18.5   58.9        305  201.3
10                           39.9             15.6   65.4        252  183.1
   maximum_seating owner_count price_nom
1                5           0         0
2                5           0         0
3                5           0         0
4                5           0         0
5                5           0         0
6                6           0         0
7                6           1         0
8                5           1         0
9                5           1         0
10               5           0         0
```

```
str(cars_kNN)
```

```
'data.frame':   26732 obs. of  17 variables:
 $ fleet           : chr  "" "" "" "" ...
 $ frame_damaged   : chr  "" "" "" "" ...
 $ franchise_dealer: chr  "True" "True" "True" "True" ...
 $ has_accidents   : chr  "" "" "" "" ...
 $ is_cpo          : chr  "False" "False" "False" "False" ...
 $ is_new          : chr  "True" "True" "True" "True" ...
 $ is_oemcpo       : chr  "False" "False" "False" "False" ...
 $ salvage         : chr  "" "" "" "" ...
 $ theft_title     : chr  "" "" "" "" ...
 $ back_legroom    : num  38.2 34.6 40.4 34.6 39.6 43.6 43.6 38.1 39.8 39.9 ...
 $ fuel_tank_volume: num  16.4 12.2 14 13.2 19.5 26 34 18.5 18.5 15.6 ...
 $ height          : num  65 59.7 66.5 61.6 72.2 77.2 81.5 66.1 58.9 65.4 ...
 $ horsepower      : int  181 138 190 147 280 375 450 175 305 252 ...
 $ length          : num  176 160 182 164 190 ...
 $ maximum_seating : int  5 5 5 5 5 6 6 5 5 5 ...
 $ owner_count     : int  0 0 0 0 0 0 1 1 1 0 ...
 $ price_nom       : int  0 0 0 0 0 0 0 0 0 0 ...
```

Here we set our categorical variables as factors because these columns have 2 or more classes.

Cars_kNN will be our data frame for the test of the cars.

```
# Set categorical variables as factor because columns have more than 2 classes

cars_kNN$fleet <- as.factor(cars_kNN$fleet)
cars_kNN$frame_damaged <- as.factor(cars_kNN$frame_damaged)
cars_kNN$franchise_dealer <- as.factor(cars_kNN$franchise_dealer)
cars_kNN$has_accidents <- as.factor(cars_kNN$has_accidents)
cars_kNN$is_cpo <- as.factor(cars_kNN$is_cpo)
cars_kNN$is_new <- as.factor(cars_kNN$is_new)
cars_kNN$is_oemcpo <- as.factor(cars_kNN$is_oemcpo)
cars_kNN$salvage <- as.factor(cars_kNN$salvage)
cars_kNN$theft_title <- as.factor(cars_kNN$theft_title)

str(cars_kNN)
```

```
'data.frame':   26732 obs. of  17 variables:
 $ fleet           : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 3 3 3 1 ...
 $ frame_damaged   : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ franchise_dealer: Factor w/ 2 levels "False","True": 2 2 2 2 2 2 1 2 2 2 ...
 $ has_accidents   : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ is_cpo          : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 2 1 1 ...
 $ is_new          : Factor w/ 2 levels "False","True": 2 2 2 2 2 2 1 1 1 2 ...
 $ is_oemcpo       : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ salvage         : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ theft_title     : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ back_legroom    : num  38.2 34.6 40.4 34.6 39.6 43.6 43.6 38.1 39.8 39.9 ...
 $ fuel_tank_volume: num  16.4 12.2 14 13.2 19.5 26 34 18.5 18.5 15.6 ...
 $ height          : num  65 59.7 66.5 61.6 72.2 77.2 81.5 66.1 58.9 65.4 ...
 $ horsepower      : int  181 138 190 147 280 375 450 175 305 252 ...
 $ length          : num  176 160 182 164 190 ...
 $ maximum_seating : int  5 5 5 5 5 6 6 5 5 5 ...
 $ owner_count     : int  0 0 0 0 0 0 1 1 1 0 ...
 $ price_nom       : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
cars_kNN$price_nom <- factor(cars_kNN$price_nom,
                             levels = c('0', '1'),
                             labels = c('low', 'high'))

head(cars_kNN)
```

```
  fleet frame_damaged franchise_dealer has_accidents is_cpo is_new is_oemcpo
1                                 True               False   True     False
2                                 True               False   True     False
3                                 True               False   True     False
4                                 True               False   True     False
5                                 True               False   True     False
6 False         False             True         False  False   True     False
  salvage theft_title back_legroom fuel_tank_volume height horsepower length
1                            38.2             16.4   65.0        181  176.4
2                            34.6             12.2   59.7        138  159.8
3                            40.4             14.0   66.5        190  182.1
4                            34.6             13.2   61.6        147  164.0
5                            39.6             19.5   72.2        280  190.5
6   False       False         43.6             26.0   77.2        375  231.9
  maximum_seating owner_count price_nom
1               5           0       low
2               5           0       low
3               5           0       low
4               5           0       low
5               5           0       low
6               6           0       low
```

```
table(cars_kNN$price_nom)
```

```
  low  high
23612  3120
```

```
str(cars_kNN)
```

```
'data.frame':   26732 obs. of  17 variables:
 $ fleet           : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 3 3 3 1 ...
 $ frame_damaged   : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
```

```
 $ franchise_dealer: Factor w/ 2 levels "False","True": 2 2 2 2 2 2 1 2 2 2 ...
 $ has_accidents   : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ is_cpo          : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 2 1 1 ...
 $ is_new          : Factor w/ 2 levels "False","True": 2 2 2 2 2 2 1 1 1 2 ...
 $ is_oemcpo       : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ salvage         : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ theft_title     : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ back_legroom    : num  38.2 34.6 40.4 34.6 39.6 43.6 43.6 38.1 39.8 39.9 ...
 $ fuel_tank_volume: num  16.4 12.2 14 13.2 19.5 26 34 18.5 18.5 15.6 ...
 $ height          : num  65 59.7 66.5 61.6 72.2 77.2 81.5 66.1 58.9 65.4 ...
 $ horsepower      : int  181 138 190 147 280 375 450 175 305 252 ...
 $ length          : num  176 160 182 164 190 ...
 $ maximum_seating : int  5 5 5 5 5 6 6 5 5 5 ...
 $ owner_count     : int  0 0 0 0 0 0 1 1 1 0 ...
 $ price_nom       : Factor w/ 2 levels "low","high": 1 1 1 1 1 1 1 1 1 1 ...
```

3. Car Test data

Now we need to get our car test data uploaded and ready to compare.

```
car_test <- read.csv("car_test_12.csv", header = TRUE)
names(car_test)
```

```
 [1] "X"                    "vin"                  "back_legroom"
 [4] "body_type"            "cabin"                "city"
 [7] "daysonmarket"         "dealer_zip"           "engine_cylinders"
[10] "engine_displacement"  "engine_type"          "exterior_color"
[13] "fleet"                "frame_damaged"        "franchise_dealer"
[16] "franchise_make"       "front_legroom"        "fuel_tank_volume"
[19] "fuel_type"            "has_accidents"        "height"
[22] "horsepower"           "interior_color"       "is_cpo"
[25] "is_new"               "is_oemcpo"            "length"
[28] "listed_date"          "listing_color"        "listing_id"
[31] "major_options"        "make_name"            "maximum_seating"
[34] "mileage"              "model_name"           "owner_count"
[37] "power"                "salvage"              "savings_amount"
[40] "seller_rating"        "sp_id"                "sp_name"
[43] "theft_title"          "torque"               "transmission"
[46] "transmission_display" "trimId"               "trim_name"
[49] "wheel_system"         "wheel_system_display" "wheelbase"
[52] "width"                "year"                 "power_hp"
[55] "power_rpm"            "torque_lbft"          "torque_rpm"
```

```
car_test <- car_test[, c(13:15,20,24:26,38,43,3,18,21,22,27,33,36)]
names(car_test)
```

```
 [1] "fleet"           "frame_damaged"    "franchise_dealer" "has_accidents"
 [5] "is_cpo"          "is_new"           "is_oemcpo"        "salvage"
 [9] "theft_title"     "back_legroom"     "fuel_tank_volume" "height"
[13] "horsepower"      "length"           "maximum_seating"  "owner_count"
```

```
str(car_test)
```

```
'data.frame':   6 obs. of  16 variables:
 $ fleet           : chr  "" "False" "" "False" ...
 $ frame_damaged   : chr  "" "False" "" "False" ...
 $ franchise_dealer: chr  "True" "False" "True" "True" ...
 $ has_accidents   : chr  "" "True" "" "False" ...
 $ is_cpo          : chr  "False" "False" "False" "False" ...
 $ is_new          : chr  "True" "False" "True" "False" ...
 $ is_oemcpo       : chr  "False" "False" "False" "False" ...
 $ salvage         : chr  "" "False" "" "False" ...
 $ theft_title     : chr  "" "False" "" "False" ...
 $ back_legroom    : num  43.4 36.8 43.6 38 38.3 38.4
 $ fuel_tank_volume: num  24 18.5 26 15.8 18.5 24.6
 $ height          : num  75.5 57.6 77.2 57.5 57.9 70.7
 $ horsepower      : int  355 197 375 310 158 302
 $ length          : num  232 192 232 198 189 ...
 $ maximum_seating : int  5 5 6 5 5 5
 $ owner_count     : int  0 2 0 1 1 1
```

II. kNN model

1. Car Test data frame

1.1 Prepare the Car Test data for kNN

```
# we will normalize the data frame for the test of the cars for the kNN model
car_test_kNN <- car_test
```

```
car_test_kNN$fleet <- as.factor(car_test_kNN$fleet)
car_test_kNN$frame_damaged <- as.factor(car_test_kNN$frame_damaged)
car_test_kNN$franchise_dealer <- as.factor(car_test_kNN$franchise_dealer)
car_test_kNN$has_accidents <- as.factor(car_test_kNN$has_accidents)
car_test_kNN$is_cpo <- as.factor(car_test_kNN$is_cpo)
car_test_kNN$is_new <- as.factor(car_test_kNN$is_new)
car_test_kNN$is_oemcpo <- as.factor(car_test_kNN$is_oemcpo)
car_test_kNN$salvage <- as.factor(car_test_kNN$salvage)
car_test_kNN$theft_title <- as.factor(car_test_kNN$theft_title)
```

```
str(car_test_kNN)
```

```
'data.frame':   6 obs. of  16 variables:
 $ fleet           : Factor w/ 2 levels "","False": 1 2 1 2 2 2
 $ frame_damaged   : Factor w/ 2 levels "","False": 1 2 1 2 2 2
 $ franchise_dealer: Factor w/ 2 levels "False","True": 2 1 2 2 2 2
 $ has_accidents   : Factor w/ 3 levels "","False","True": 1 3 1 2 2 2
 $ is_cpo          : Factor w/ 1 level "False": 1 1 1 1 1 1
 $ is_new          : Factor w/ 2 levels "False","True": 2 1 2 1 1 1
 $ is_oemcpo       : Factor w/ 1 level "False": 1 1 1 1 1 1
 $ salvage         : Factor w/ 2 levels "","False": 1 2 1 2 2 2
 $ theft_title     : Factor w/ 2 levels "","False": 1 2 1 2 2 2
 $ back_legroom    : num  43.4 36.8 43.6 38 38.3 38.4
 $ fuel_tank_volume: num  24 18.5 26 15.8 18.5 24.6
 $ height          : num  75.5 57.6 77.2 57.5 57.9 70.7
 $ horsepower      : int  355 197 375 310 158 302
 $ length          : num  232 192 232 198 189 ...
 $ maximum_seating : int  5 5 6 5 5 5
 $ owner_count     : int  0 2 0 1 1 1
```

1.2 Levels of training data and new data

A. Before the adding the levels to the new data

```
str(car_test_kNN)
```

```
'data.frame':   6 obs. of  16 variables:
 $ fleet           : Factor w/ 2 levels "","False": 1 2 1 2 2 2
 $ frame_damaged   : Factor w/ 2 levels "","False": 1 2 1 2 2 2
 $ franchise_dealer: Factor w/ 2 levels "False","True": 2 1 2 2 2 2
 $ has_accidents   : Factor w/ 3 levels "","False","True": 1 3 1 2 2 2
 $ is_cpo          : Factor w/ 1 level "False": 1 1 1 1 1 1
 $ is_new          : Factor w/ 2 levels "False","True": 2 1 2 1 1 1
 $ is_oemcpo       : Factor w/ 1 level "False": 1 1 1 1 1 1
 $ salvage         : Factor w/ 2 levels "","False": 1 2 1 2 2 2
 $ theft_title     : Factor w/ 2 levels "","False": 1 2 1 2 2 2
 $ back_legroom    : num  43.4 36.8 43.6 38 38.3 38.4
 $ fuel_tank_volume: num  24 18.5 26 15.8 18.5 24.6
 $ height          : num  75.5 57.6 77.2 57.5 57.9 70.7
 $ horsepower      : int  355 197 375 310 158 302
 $ length          : num  232 192 232 198 189 ...
 $ maximum_seating : int  5 5 6 5 5 5
 $ owner_count     : int  0 2 0 1 1 1
```

```
# levels of each variables inside the new data
```

```
str(cars_kNN)
```

```
'data.frame':   26732 obs. of  17 variables:
 $ fleet           : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 3 3 3 1 ...
 $ frame_damaged   : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ franchise_dealer: Factor w/ 2 levels "False","True": 2 2 2 2 2 2 1 2 2 2 ...
 $ has_accidents   : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ is_cpo          : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 2 1 1 ...
 $ is_new          : Factor w/ 2 levels "False","True": 2 2 2 2 2 2 1 1 1 2 ...
 $ is_oemcpo       : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ salvage         : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ theft_title     : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ back_legroom    : num  38.2 34.6 40.4 34.6 39.6 43.6 43.6 38.1 39.8 39.9 ...
 $ fuel_tank_volume: num  16.4 12.2 14 13.2 19.5 26 34 18.5 18.5 15.6 ...
 $ height          : num  65 59.7 66.5 61.6 72.2 77.2 81.5 66.1 58.9 65.4 ...
 $ horsepower      : int  181 138 190 147 280 375 450 175 305 252 ...
 $ length          : num  176 160 182 164 190 ...
 $ maximum_seating : int  5 5 5 5 5 6 6 5 5 5 ...
 $ owner_count     : int  0 0 0 0 0 0 1 1 1 0 ...
 $ price_nom       : Factor w/ 2 levels "low","high": 1 1 1 1 1 1 1 1 1 1 ...
```

```
# levels of each variables inside the training data
```

```
levels(car_test_kNN$fleet)
```

```
[1] ""      "False"
```

```
levels(car_test_kNN$frame_damaged)
```

```
[1] ""      "False"
```

```
levels(car_test_kNN$is_cpo)
```

```
[1] "False"
```

```
levels(car_test_kNN$is_oemcpo)
```

```
[1] "False"
```

```
levels(car_test_kNN$salvage)
```

```
[1] ""      "False"
```

```
levels(car_test_kNN$theft_title)
```

```
[1] ""      "False"
```

B. After the adding the levels to the new data

```
levels(car_test_kNN$fleet) <- levels(cars_kNN$fleet)
levels(car_test_kNN$frame_damaged) <- levels(cars_kNN$frame_damaged)
levels(car_test_kNN$is_cpo) <- levels(cars_kNN$is_cpo)
levels(car_test_kNN$is_oemcpo) <- levels(cars_kNN$is_oemcpo)
levels(car_test_kNN$salvage) <- levels(cars_kNN$salvage)
levels(car_test_kNN$theft_title) <- levels(cars_kNN$theft_title)

head(car_test_kNN)
```

```
  fleet frame_damaged franchise_dealer has_accidents is_cpo is_new is_oemcpo
1                              True               False   True     False
2 False         False           False          True  False  False     False
3                              True               False   True     False
4 False         False            True         False  False  False     False
5 False         False            True         False  False  False     False
6 False         False            True         False  False  False     False
  salvage theft_title back_legroom fuel_tank_volume height horsepower length
1                           43.4             24.0   75.5        355  231.7
2   False       False         36.8             18.5   57.6        197  191.5
3                           43.6             26.0   77.2        375  231.9
4   False       False         38.0             15.8   57.5        310  197.5
5   False       False         38.3             18.5   57.9        158  189.2
6   False       False         38.4             24.6   70.7        302  189.7
  maximum_seating owner_count
1               5           0
2               5           2
3               6           0
4               5           1
5               5           1
6               5           1
```

```
str(cars_kNN)
```

```
'data.frame':   26732 obs. of  17 variables:
 $ fleet           : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 3 3 3 1 ...
 $ frame_damaged   : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ franchise_dealer: Factor w/ 2 levels "False","True": 2 2 2 2 2 2 1 2 2 2 ...
 $ has_accidents   : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ is_cpo          : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 2 1 1 ...
 $ is_new          : Factor w/ 2 levels "False","True": 2 2 2 2 2 2 1 1 1 2 ...
 $ is_oemcpo       : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ salvage         : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ theft_title     : Factor w/ 3 levels "","False","True": 1 1 1 1 1 2 2 2 2 1 ...
 $ back_legroom    : num  38.2 34.6 40.4 34.6 39.6 43.6 43.6 38.1 39.8 39.9 ...
 $ fuel_tank_volume: num  16.4 12.2 14 13.2 19.5 26 34 18.5 18.5 15.6 ...
 $ height          : num  65 59.7 66.5 61.6 72.2 77.2 81.5 66.1 58.9 65.4 ...
 $ horsepower      : int  181 138 190 147 280 375 450 175 305 252 ...
 $ length          : num  176 160 182 164 190 ...
 $ maximum_seating : int  5 5 5 5 5 6 6 5 5 5 ...
 $ owner_count     : int  0 0 0 0 0 0 1 1 1 0 ...
 $ price_nom       : Factor w/ 2 levels "low","high": 1 1 1 1 1 1 1 1 1 1 ...
```

```
str(car_test_kNN)
```

```
'data.frame':   6 obs. of  16 variables:
 $ fleet           : Factor w/ 3 levels "","False","True": 1 2 1 2 2 2
 $ frame_damaged   : Factor w/ 3 levels "","False","True": 1 2 1 2 2 2
 $ franchise_dealer: Factor w/ 2 levels "False","True": 2 1 2 2 2 2
 $ has_accidents   : Factor w/ 3 levels "","False","True": 1 3 1 2 2 2
 $ is_cpo          : Factor w/ 2 levels "False","True": 1 1 1 1 1 1
 $ is_new          : Factor w/ 2 levels "False","True": 2 1 2 1 1 1
 $ is_oemcpo       : Factor w/ 2 levels "False","True": 1 1 1 1 1 1
 $ salvage         : Factor w/ 3 levels "","False","True": 1 2 1 2 2 2
 $ theft_title     : Factor w/ 3 levels "","False","True": 1 2 1 2 2 2
 $ back_legroom    : num  43.4 36.8 43.6 38 38.3 38.4
 $ fuel_tank_volume: num  24 18.5 26 15.8 18.5 24.6
 $ height          : num  75.5 57.6 77.2 57.5 57.9 70.7
 $ horsepower      : int  355 197 375 310 158 302
 $ length          : num  232 192 232 198 189 ...
 $ maximum_seating : int  5 5 6 5 5 5
 $ owner_count     : int  0 2 0 1 1 1
```

2. Prepare for the kNN

Here we set our training and validation sets for the KNN Model:

training_index_kNN will be the training index for the kNN Model.

valid_index_kNN will be the validation index for the kNN Model.

train_kNN will be the data frame for the training data for the kNN model after splitting.

valid_kNN will be data frame for the validation data for the kNN model after splitting

```
# Set training and validation sets for kNN model
set.seed(666)

train_index_kNN <- sample(1:nrow(cars_kNN), 0.6 * nrow(cars_kNN))
valid_index_kNN <- setdiff(1:nrow(cars_kNN), train_index_kNN)

train_kNN <- cars_kNN[train_index_kNN,]
valid_kNN <- cars_kNN[valid_index_kNN,]
```

```
nrow(train_kNN)
```

```
[1] 16039
```

```
nrow(valid_kNN)
```

```
[1] 10693
```

train_norm_kNN will be the data frame for the normalization of the training data of the kNN model.

valid_norm_kNN will be the data frame for the validation of the validation data of the kNN model

```
train_norm_kNN <- train_kNN
valid_norm_kNN <- valid_kNN
```

```
t(t(names(cars_kNN)))
```

```
      [,1]
 [1,] "fleet"
 [2,] "frame_damaged"
 [3,] "franchise_dealer"
 [4,] "has_accidents"
 [5,] "is_cpo"
 [6,] "is_new"
 [7,] "is_oemcpo"
 [8,] "salvage"
 [9,] "theft_title"
[10,] "back_legroom"
[11,] "fuel_tank_volume"
[12,] "height"
[13,] "horsepower"
[14,] "length"
[15,] "maximum_seating"
[16,] "owner_count"
[17,] "price_nom"
```

```
str(train_kNN)
```

```
'data.frame':   16039 obs. of  17 variables:
 $ fleet           : Factor w/ 3 levels "","False","True": 3 1 2 2 2 1 1 2 1 3 ...
 $ frame_damaged   : Factor w/ 3 levels "","False","True": 2 1 2 2 2 1 1 2 1 2 ...
 $ franchise_dealer: Factor w/ 2 levels "False","True": 2 2 1 1 2 2 2 2 2 2 ...
 $ has_accidents   : Factor w/ 3 levels "","False","True": 2 1 3 3 2 1 1 2 1 3 ...
 $ is_cpo          : Factor w/ 2 levels "False","True": 1 1 1 1 2 1 1 1 1 1 ...
 $ is_new          : Factor w/ 2 levels "False","True": 1 2 1 1 1 2 2 1 2 1 ...
 $ is_oemcpo       : Factor w/ 2 levels "False","True": 1 1 1 1 2 1 1 1 1 1 ...
 $ salvage         : Factor w/ 3 levels "","False","True": 2 1 2 2 2 1 1 2 1 2 ...
 $ theft_title     : Factor w/ 3 levels "","False","True": 2 1 2 2 2 1 1 2 1 2 ...
 $ back_legroom    : num  40 38.2 38.4 35.9 38.9 38.5 37.8 36 38.3 40 ...
 $ fuel_tank_volume: num  19 16.4 26 18.5 17 26 18.5 12.4 21.5 25.1 ...
 $ height          : num  59.4 65.2 75.8 57.1 57.9 77.2 65.3 56.5 73.6 71.5 ...
 $ horsepower      : int  304 181 240 272 178 400 248 174 270 215 ...
 $ length          : num  201 176 204 191 191 ...
 $ maximum_seating : int  5 5 8 5 5 5 5 5 5 5 ...
 $ owner_count     : int  1 0 4 2 1 0 0 1 0 4 ...
 $ price_nom       : Factor w/ 2 levels "low","high": 1 1 1 1 1 1 2 1 1 1 ...
```

Here we will prepare the data for analysis by creating pre-process model. We also normalized the selected columns in the train_norm data set based on the transformations learned from the norm values model.

norm_values_kNN will be the data frame that will prepare for the analysis for the kNN model.

```
# preProcess: Prepare the data for the analysis / create preprocessing model
norm_values_kNN <- preProcess(train_kNN[, -c(1:9,17)],
                           method = c("center",
                                      "scale"))

# Normalize the selected columns in the train_norm dataset based on the transformations learned from "norm_values model"
# predict(model, dataset)
train_norm_kNN[, -c(1:9, 17)] <- predict(norm_values_kNN,
                                    train_kNN[, -c(1:9, 17)])
```

```
head(train_norm_kNN)
```

```
      fleet frame_damaged franchise_dealer has_accidents is_cpo is_new
17982  True         False             True         False  False  False
12926                                 True                False   True
13195 False         False            False          True  False  False
23675 False         False            False          True  False  False
15900 False         False             True         False   True  False
873                                   True                False   True
      is_oemcpo salvage theft_title back_legroom fuel_tank_volume      height
17982     False   False       False   0.64311937       0.11767538 -0.89323874
12926     False                         0.07772847      -0.38966276 -0.06143842
13195     False   False       False   0.14054968       1.48358578  1.45874836
23675     False   False       False  -0.64471546       0.02011035 -1.22309059
15900      True   False       False   0.29760271      -0.27258473 -1.10835951
873       False                         0.17196029       1.48358578  1.65952775
       horsepower      length maximum_seating owner_count price_nom
17982  0.68578482  0.40667868       -0.497447   0.2389365       low
12926 -0.72440277 -0.85592908       -0.497447  -0.7802928       low
13195 -0.04797133  0.56128371        2.232441   3.2966242       low
23675  0.31890675 -0.09836443       -0.497447   1.2581657       low
```

```
15900 -0.75879759 -0.10867143       -0.497447   0.2389365       low
873    1.78641904  1.88057997       -0.497447  -0.7802928       low

valid_norm_kNN[, -c(1:9, 17)] <- predict(norm_values_kNN,
                                valid_kNN[, -c(1:9, 17)])

head(valid_norm_kNN)
```

```
   fleet frame_damaged franchise_dealer has_accidents is_cpo is_new is_oemcpo
2                                 True               False   True     False
3                                 True               False   True     False
6  False         False             True         False  False   True     False
9   True         False             True         False  False  False     False
10                                True               False   True     False
11                                True               False   True     False
   salvage theft_title back_legroom fuel_tank_volume      height  horsepower
2                       -1.0530533      -1.20920900 -0.85021459 -1.21739518
3                        0.7687618      -0.85797490  0.12499958 -0.62121832
6    False       False    1.7739012       1.48358578  1.65952775  1.49979554
9    False       False    0.5802982       0.02011035 -0.96494566  0.69724976
10                       0.6117088      -0.54576681 -0.03275565  0.08960795
11                      -0.8960003      -1.11164397  0.12499958 -0.77026253
      length maximum_seating owner_count price_nom
2  -1.7114103      -0.4974470  -0.7802928       low
3  -0.5621795      -0.4974470  -0.7802928       low
6   2.0042640       0.4125156  -0.7802928       low
9   0.4272927      -0.4974470   0.2389365       low
10 -0.5106445      -0.4974470  -0.7802928       low
11 -1.3609722      -0.4974470  -0.7802928       low
```

Next we need to normalize car test data.

car_test_norm_kNN will be the data frame of the car test data frame that will be normalized for the kNN model.

```
# Normalize Car Test data for kNN
car_test_norm_kNN <- predict(norm_values_kNN, car_test_kNN)
car_test_norm_kNN
```

```
  fleet frame_damaged franchise_dealer has_accidents is_cpo is_new is_oemcpo
1                                 True               False   True     False
2 False         False            False          True  False  False     False
3                                 True               False   True     False
4 False         False             True         False  False  False     False
5 False         False             True         False  False  False     False
6 False         False             True         False  False  False     False
  salvage theft_title back_legroom fuel_tank_volume     height horsepower
1                      1.71107996       1.09332566  1.4157242  1.2704967
2   False       False  -0.36202001       0.02011035 -1.1513837 -0.5409637
3                      1.77390117       1.48358578  1.6595277  1.4997955
4   False       False   0.01490726      -0.50674080 -1.1657251  0.7545745
5   False       False   0.10913908       0.02011035 -1.1083595 -0.9880964
6   False       False   0.14054968       1.21040370  0.7273377  0.6628549
      length maximum_seating owner_count
1  1.99395700      -0.4974470  -0.7802928
2 -0.07775042      -0.4974470   1.2581657
3  2.00426400       0.4125156  -0.7802928
4  0.23145964      -0.4974470   0.2389365
5 -0.19628095      -0.4974470   0.2389365
6 -0.17051344      -0.4974470   0.2389365
```

3. kNN model

3.1 k = 3

A. The Training

kNN_model_k3 is the kNN model for k = 3.

```
knn_model_k3 <- caret::knn3(price_nom ~.,
                            data = train_norm_kNN, k = 3)
knn_model_k3
```

```
3-nearest neighbor model
Training set outcome distribution:

  low  high
14135  1904
```

B. The Prediction

Prediction on training Set

kNN_pred_k3_train will be the prediction for the training set for the kNN model for k = 3.

```
knn_pred_k3_train <- predict(knn_model_k3, newdata = train_norm_kNN[, -c(17)],
                             type = "class")
head(knn_pred_k3_train)
```

```
[1] low  low  low  low  low  high
Levels: low high
```

```
confusionMatrix(knn_pred_k3_train, as.factor(train_norm_kNN[, 17]))
```

```
Confusion Matrix and Statistics

          Reference
Prediction   low  high
      low  13843   466
      high   292  1438

               Accuracy : 0.9527
                 95% CI : (0.9493, 0.956)
    No Information Rate : 0.8813
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7648

 Mcnemar's Test P-Value : 3.307e-10

            Sensitivity : 0.9793
            Specificity : 0.7553
         Pos Pred Value : 0.9674
         Neg Pred Value : 0.8312
             Prevalence : 0.8813
         Detection Rate : 0.8631
   Detection Prevalence : 0.8921
      Balanced Accuracy : 0.8673

       'Positive' Class : low
```

Prediction on Validation Set

kNN_pred_k3_valid will be the prediction for the validation set for the kNN model for k = 3.

```
knn_pred_k3_valid <- predict(knn_model_k3, newdata = valid_norm_kNN[, -c(17)],
                             type = "class")
head(knn_pred_k3_valid)
```

```
[1] low  low  high low  low  low
Levels: low high
```

```
confusionMatrix(knn_pred_k3_valid, as.factor(valid_norm_kNN[, 17]))
```

```
Confusion Matrix and Statistics

          Reference
Prediction  low high
      low  9266  349
      high  211  867

               Accuracy : 0.9476
                 95% CI : (0.9432, 0.9518)
    No Information Rate : 0.8863
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7267

 Mcnemar's Test P-Value : 7.068e-09

            Sensitivity : 0.9777
            Specificity : 0.7130
         Pos Pred Value : 0.9637
         Neg Pred Value : 0.8043
             Prevalence : 0.8863
         Detection Rate : 0.8665
   Detection Prevalence : 0.8992
      Balanced Accuracy : 0.8454

       'Positive' Class : low
```

***Predicting the Car Test

car_test_predict_kNN_k3 will be the values of the car test prediction from kNN model with k = 3.

```
car_test_predict_kNN_k3 <- predict(knn_model_k3,
                            newdata = car_test_norm_kNN,
                            type = "class")
car_test_predict_kNN_k3
```

```
[1] high low  high low  low  low
Levels: low high
```

***Probabilities

```
knn_pred_k3_prob <- predict(knn_model_k3, newdata = valid_norm_kNN[, -c(17)],
                            type = "prob")
head(knn_pred_k3_prob)
```

```
      low high
[1,] 1.00 0.00
[2,] 1.00 0.00
[3,] 0.45 0.55
```

```
[4,] 1.00 0.00
[5,] 1.00 0.00
[6,] 1.00 0.00
```

3.2 k = 5

A. The Training

kNN_model_k5 is kNN model for k = 5.

```
knn_model_k5 <- caret::knn3(price_nom ~.,
                            data = train_norm_kNN, k = 5)
knn_model_k5
```

```
5-nearest neighbor model
Training set outcome distribution:

  low  high
14135  1904
```

B. The Prediction

Prediction on Training Set

kNN_pred_k5_train is the prediction for the training set for the kNN model for k = 5.

```
knn_pred_k5_train <- predict(knn_model_k5, newdata = train_norm_kNN[, -c(17)],
                             type = "class")
head(knn_pred_k5_train)
```

```
[1] low  low  low  low  low  high
Levels: low high
```

```
confusionMatrix(knn_pred_k5_train, as.factor(train_norm_kNN[, 17]))
```

```
Confusion Matrix and Statistics

          Reference
Prediction   low  high
      low  13845   523
      high   290  1381

               Accuracy : 0.9493
                 95% CI : (0.9458, 0.9527)
    No Information Rate : 0.8813
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7442

 Mcnemar's Test P-Value : 4.065e-16

            Sensitivity : 0.9795
            Specificity : 0.7253
         Pos Pred Value : 0.9636
         Neg Pred Value : 0.8265
             Prevalence : 0.8813
         Detection Rate : 0.8632
   Detection Prevalence : 0.8958
      Balanced Accuracy : 0.8524

       'Positive' Class : low
```

Prediction on Validation Set

kNN_pred_k5_valid is the prediction for the validation set for the kNN model for k = 5.

```
knn_pred_k5_valid <- predict(knn_model_k5, newdata = valid_norm_kNN[, -c(17)],
                             type = "class")
head(knn_pred_k5_valid)
```

```
[1] low  low  high low  low  low
Levels: low high
```

```
confusionMatrix(knn_pred_k5_valid, as.factor(valid_norm_kNN[, 17]))
```

```
Confusion Matrix and Statistics

          Reference
Prediction  low high
      low  9295  391
      high  182  825

               Accuracy : 0.9464
                 95% CI : (0.942, 0.9506)
    No Information Rate : 0.8863
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7126

 Mcnemar's Test P-Value : < 2.2e-16
```

```
            Sensitivity : 0.9808
            Specificity : 0.6785
         Pos Pred Value : 0.9596
         Neg Pred Value : 0.8193
             Prevalence : 0.8863
         Detection Rate : 0.8693
   Detection Prevalence : 0.9058
      Balanced Accuracy : 0.8296

       'Positive' Class : low
```

***Predicting the Car Test

car_test_predict_kNN_k5 will be the values of the car test prediction from kNN model with k = 5.

```
car_test_predict_kNN_k5 <- predict(knn_model_k5,
                           newdata = car_test_norm_kNN,
                           type = "class")
car_test_predict_kNN_k5
```

```
[1] high low  high low  low  low
Levels: low high
```

***Probabilities

```
knn_pred_k5_prob <- predict(knn_model_k5, newdata = valid_norm_kNN[, -c(17)],
                           type = "prob")
head(knn_pred_k5_prob)
```

```
      low high
[1,] 1.00 0.00
[2,] 1.00 0.00
[3,] 0.45 0.55
[4,] 1.00 0.00
[5,] 1.00 0.00
[6,] 1.00 0.00
```

4. Model Evaluation

```
library(ROSE)
```

```
Loaded ROSE 0.0-4
```

4.1 k = 3

```
ROSE::roc.curve(valid_norm_kNN$price_nom, knn_pred_k3_valid)
```

```
Area under the curve (AUC): 0.845
```

4.2 k = 5

```
ROSE::roc.curve(valid_norm_kNN$price_nom, knn_pred_k5_valid)
```

```
Area under the curve (AUC): 0.830
```

5. Weighted Data kNN

train_kNN_df_rose is the data frame for the training data after balancing the data for the kNN model.

```
train_kNN_df_rose <- ROSE(price_nom ~., data = train_kNN,
                     seed = 666)$data

table(train_kNN_df_rose$price_nom)
```

```
 low high
7953 8086
```

train_norm_kNN_2 is the 2nd data frame for the normalization of the training data of the kNN model for the balance data.

valid_norm_kNN_2 is the 2nd data frame for the validation of the validation data of the kNN model for balance data.

```
train_norm_kNN_2 <- train_kNN_df_rose
valid_norm_kNN_2 <- valid_kNN

names(train_norm_kNN_2)
```

```
 [1] "fleet"           "frame_damaged"    "franchise_dealer" "has_accidents"
 [5] "is_cpo"          "is_new"           "is_oemcpo"        "salvage"
 [9] "theft_title"     "back_legroom"     "fuel_tank_volume" "height"
```

```
[13] "horsepower"       "length"           "maximum_seating"  "owner_count"
[17] "price_nom"
```

norm_values_kNN_2 is the 2nd data frame that was prepared for the analysis for the kNN model for the balance data.

```
norm_values_kNN_2 <- preProcess(train_kNN[, -c(1:9, 17)],
                                method = c("center",
                                           "scale"))

train_norm_kNN_2[, -c(1:9, 17)] <- predict(norm_values_kNN_2,
                                     train_kNN[, -c(1:9, 17)])
head(train_norm_kNN_2)
```

```
  fleet frame_damaged franchise_dealer has_accidents is_cpo is_new is_oemcpo
1 False         False             True         False   True  False      True
2 False         False             True         False  False  False     False
3  True         False             True         False  False  False     False
4                                 True                False   True     False
5                                 True                False   True     False
6                                 True                False   True     False
  salvage theft_title back_legroom fuel_tank_volume      height  horsepower
1   False       False   0.64311937       0.11767538 -0.89323874  0.68578482
2   False       False   0.07772847      -0.38966276 -0.06143842 -0.72440277
3   False       False   0.14054968       1.48358578  1.45874836 -0.04797133
4                      -0.64471546       0.02011035 -1.22309059  0.31890675
5                       0.29760271      -0.27258473 -1.10835951 -0.75879759
6                       0.17196029       1.48358578  1.65952775  1.78641904
       length maximum_seating owner_count price_nom
1  0.40667868       -0.497447   0.2389365       low
2 -0.85592908       -0.497447  -0.7802928       low
3  0.56128371        2.232441   3.2966242       low
4 -0.09836443       -0.497447   1.2581657       low
5 -0.10867143       -0.497447   0.2389365       low
6  1.88057997       -0.497447  -0.7802928       low
```

```
valid_norm_kNN_2[, -c(1:9, 17)] <- predict(norm_values_kNN_2,
                                     valid_kNN[, -c(1:9, 17)])
head(valid_norm_kNN_2)
```

```
   fleet frame_damaged franchise_dealer has_accidents is_cpo is_new is_oemcpo
2                                 True                False   True     False
3                                 True                False   True     False
6  False         False             True         False  False   True     False
9   True         False             True         False  False  False     False
10                                True                False   True     False
11                                True                False   True     False
   salvage theft_title back_legroom fuel_tank_volume      height  horsepower
2                      -1.0530533      -1.20920900 -0.85021459 -1.21739518
3                       0.7687618      -0.85797490  0.12499958 -0.62121832
6    False       False    1.7739012       1.48358578  1.65952775  1.49979554
9    False       False    0.5802982       0.02011035 -0.96494566  0.69724976
10                      0.6117088      -0.54576681 -0.03275565  0.08960795
11                     -0.8960003      -1.11164397  0.12499958 -0.77026253
       length maximum_seating owner_count price_nom
2  -1.7114103      -0.4974470  -0.7802928       low
3  -0.5621795      -0.4974470  -0.7802928       low
6   2.0042640       0.4125156  -0.7802928       low
9   0.4272927      -0.4974470   0.2389365       low
10 -0.5106445      -0.4974470  -0.7802928       low
11 -1.3609722      -0.4974470  -0.7802928       low
```

knn_model_2 is the kNN model 2 for the balance data.

```
knn_model_2 <- caret::knn3(price_nom ~ ., data = train_norm_kNN_2, k = 15)
knn_model_2
```

```
15-nearest neighbor model
Training set outcome distribution:

 low high
7953 8086
```

5.1 Predict training set

knn_pred_train_2 is the prediction for the training set for the kNN model 2.

```
knn_pred_train_2 <- predict(knn_model_2, newdata =
                              train_norm_kNN_2[, -c(17)],
                            type = "class")
head(knn_pred_train_2)
```

```
[1] low  low  low  high high high
Levels: low high
```

```
confusionMatrix(knn_pred_train_2, as.factor(train_norm_kNN_2[, 17]),
                positive = "low")
```

```
Confusion Matrix and Statistics

          Reference
Prediction  low high
      low  4884 1565
      high 3069 6521
```

```
               Accuracy : 0.7111
                 95% CI : (0.704, 0.7181)
    No Information Rate : 0.5041
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4212

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.6141
            Specificity : 0.8065
         Pos Pred Value : 0.7573
         Neg Pred Value : 0.6800
             Prevalence : 0.4959
         Detection Rate : 0.3045
   Detection Prevalence : 0.4021
      Balanced Accuracy : 0.7103

       'Positive' Class : low
```

5.2 Predict Validation set

knn_pred_valid_2 is the prediction for the validation set for the kNN model 2.

```
knn_pred_valid_2 <- predict(knn_model_2,
                            newdata = valid_norm_kNN_2[, -c(17)],
                            type = "class")
head(knn_pred_valid_2)
```

```
[1] high high low  low  high high
Levels: low high
```

```
confusionMatrix(knn_pred_valid_2, as.factor(valid_norm_kNN_2[, 17]),
                positive = "low")
```

```
Confusion Matrix and Statistics

          Reference
Prediction  low high
      low  5418  257
      high 4059  959

               Accuracy : 0.5964
                 95% CI : (0.587, 0.6057)
    No Information Rate : 0.8863
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1525

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.5717
            Specificity : 0.7887
         Pos Pred Value : 0.9547
         Neg Pred Value : 0.1911
             Prevalence : 0.8863
         Detection Rate : 0.5067
   Detection Prevalence : 0.5307
      Balanced Accuracy : 0.6802

       'Positive' Class : low
```

5.3 Model Evaluation

```
ROSE::roc.curve(valid_norm_kNN_2$price_nom, knn_pred_valid_2)
```

```
Area under the curve (AUC): 0.680
```

***Predicting the Car Test

car_test_predict_kNN_model_2 is the values of the car test prediction from kNN model 2.

```
car_test_predict_kNN_model_2 <- predict(knn_model_2,
                            newdata = car_test_norm_kNN,
                            type = "class")
car_test_predict_kNN_model_2
```

```
[1] high low  high low  low  low
Levels: low high
```

***Probabilities

```
knn_pred_model_2_prob <- predict(knn_model_2, newdata = valid_norm_kNN_2[, -c(17)],
                                 type = "prob")
head(knn_pred_model_2_prob)
```

```
          low      high
[1,] 0.4615385 0.5384615
[2,] 0.3684211 0.6315789
[3,] 0.5200000 0.4800000
[4,] 0.8666667 0.1333333
[5,] 0.3846154 0.6153846
[6,] 0.3783784 0.6216216
```

Conclusion About the Probabilities

We can see that the probabilities for k = 3 and k = 5 is not balance and accurate due to the imbalance in the data but after we create a new model for kNN and balance the data, the probabilities has become more accurate. That means we can be sure that the weighted data is eligible to use to predict the outcome.

III. Classification Tree model

1. Data for Classification Tree model

Next we will create a classification tree. We use a classification tree to analyze both numerical and categorical data, while kNN can analyze only int and numerical data.

cars_class_tr is the data frame for the Classification Tree model.

```
cars_class_tr <- cars
str(cars_class_tr)
```

```
'data.frame':   29233 obs. of  17 variables:
 $ fleet           : chr  "" "" "" "" ...
 $ frame_damaged   : chr  "" "" "" "" ...
 $ franchise_dealer: chr  "True" "True" "True" "True" ...
 $ has_accidents   : chr  "" "" "" "" ...
 $ is_cpo          : chr  "False" "False" "False" "False" ...
 $ is_new          : chr  "True" "True" "True" "True" ...
 $ is_oemcpo       : chr  "False" "False" "False" "False" ...
 $ salvage         : chr  "" "" "" "" ...
 $ theft_title     : chr  "" "" "" "" ...
 $ back_legroom    : num  38.2 34.6 40.4 34.6 39.6 43.6 43.6 38.1 39.8 NA ...
 $ fuel_tank_volume: num  16.4 12.2 14 13.2 19.5 26 34 18.5 18.5 NA ...
 $ height          : num  65 59.7 66.5 61.6 72.2 77.2 81.5 66.1 58.9 NA ...
 $ horsepower      : int  181 138 190 147 280 375 450 175 305 NA ...
 $ length          : num  176 160 182 164 190 ...
 $ maximum_seating : int  5 5 5 5 5 6 6 5 5 NA ...
 $ owner_count     : int  0 0 0 0 0 0 1 1 1 0 ...
 $ price_nom       : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
cars_class_tr$price_nom <- factor(cars_class_tr$price_nom,
                            levels = c('0', '1'),
                            labels = c('low', 'high'))
```

```
head(cars_class_tr)
```

```
  fleet frame_damaged franchise_dealer has_accidents is_cpo is_new is_oemcpo
1                                 True               False   True     False
2                                 True               False   True     False
3                                 True               False   True     False
4                                 True               False   True     False
5                                 True               False   True     False
6 False         False             True         False  False   True     False
  salvage theft_title back_legroom fuel_tank_volume height horsepower length
1                            38.2             16.4   65.0        181  176.4
2                            34.6             12.2   59.7        138  159.8
3                            40.4             14.0   66.5        190  182.1
4                            34.6             13.2   61.6        147  164.0
5                            39.6             19.5   72.2        280  190.5
6   False       False        43.6             26.0   77.2        375  231.9
  maximum_seating owner_count price_nom
1               5           0       low
2               5           0       low
3               5           0       low
4               5           0       low
5               5           0       low
6               6           0       low
```

```
table(cars_class_tr$price_nom)
```

```
  low  high
25575  3658
```

```
str(cars_class_tr)
```

```
'data.frame':   29233 obs. of  17 variables:
 $ fleet           : chr  "" "" "" "" ...
 $ frame_damaged   : chr  "" "" "" "" ...
 $ franchise_dealer: chr  "True" "True" "True" "True" ...
 $ has_accidents   : chr  "" "" "" "" ...
 $ is_cpo          : chr  "False" "False" "False" "False" ...
 $ is_new          : chr  "True" "True" "True" "True" ...
 $ is_oemcpo       : chr  "False" "False" "False" "False" ...
 $ salvage         : chr  "" "" "" "" ...
 $ theft_title     : chr  "" "" "" "" ...
 $ back_legroom    : num  38.2 34.6 40.4 34.6 39.6 43.6 43.6 38.1 39.8 NA ...
 $ fuel_tank_volume: num  16.4 12.2 14 13.2 19.5 26 34 18.5 18.5 NA ...
 $ height          : num  65 59.7 66.5 61.6 72.2 77.2 81.5 66.1 58.9 NA ...
```

```
 $ horsepower      : int  181 138 190 147 280 375 450 175 305 NA ...
 $ length          : num  176 160 182 164 190 ...
 $ maximum_seating : int  5 5 5 5 5 6 6 5 5 NA ...
 $ owner_count     : int  0 0 0 0 0 0 1 1 1 0 ...
 $ price_nom       : Factor w/ 2 levels "low","high": 1 1 1 1 1 1 1 1 1 1 ...
```

Here we set the training and validation set for the classification tree model:

train_index_class_tr is the training index for the Classification Tree model.

valid_index_class_tr is the validation index for the Classification Tree model.

train_class is the data frame for the training data for the Classification Tree model after splitting.

valid_class is the data frame for the validation data for the Classification Tree model after splitting.

```
# Set Training and Validation set for Classification Tree model

set.seed(666)

train_index_class_tr <- sample(1:nrow(cars_class_tr), 0.7 * nrow(cars_class_tr))
valid_index_class_tr <- setdiff(1:nrow(cars_class_tr), train_index_class_tr)

train_class <- cars_class_tr[train_index_class_tr,]
valid_class <- cars_class_tr[valid_index_class_tr,]
```

```
nrow(train_class)
```

```
[1] 20463
```

```
nrow(valid_class)
```

```
[1] 8770
```

```
head(train_class,10)
```

```
      fleet frame_damaged franchise_dealer has_accidents is_cpo is_new
17982  True         False             True         False  False  False
12926                                True                False   True
13195                                True                False   True
23675  True         False             True         False  False  False
15900 False         False            False         False  False  False
873   False         False             True         False   True  False
17036                                True                False   True
18081 False         False            False         False  False  False
1074  False         False             True         False  False  False
6275                                 True                False   True
      is_oemcpo salvage theft_title back_legroom fuel_tank_volume height
17982     False   False       False         41.5             23.3   76.6
12926     False                             38.6             24.6   69.3
13195     False                               NA               NA     NA
23675     False   False       False         39.8             18.5   58.9
15900     False   False       False         36.3             21.0   65.7
873       False   False       False         34.8             17.7   63.9
17036     False                             39.0             19.2   69.9
18081     False   False       False         38.5             15.3   66.1
1074      False   False       False         37.2             15.9   67.1
6275      False                             34.5             18.0   71.8
      horsepower length maximum_seating owner_count price_nom
17982        375  210.0               8           1      high
12926        295  189.8               5           0       low
13195         NA     NA              NA           0       low
23675        305  201.3               5           1       low
15900        308  190.3               5           2       low
873          240  184.5               5           2       low
17036        300  198.8               7           0      high
18081        166  177.9               5           1       low
1074         176  183.5               5           1       low
6275         270  210.8               5           0       low
```

```
head(valid_class, 10)
```

```
   fleet frame_damaged franchise_dealer has_accidents is_cpo is_new is_oemcpo
2                                 True                False   True     False
3                                 True                False   True     False
6  False         False             True         False  False   True     False
9   True         False             True         False  False  False     False
10                                True                False   True     False
12                                True                False   True     False
19 False         False            False         False  False  False     False
21                                True                False   True     False
31  True         False            False         False  False  False     False
38 False         False             True         False  False  False     False
   salvage theft_title back_legroom fuel_tank_volume height horsepower length
2                            34.6             12.2   59.7        138  159.8
3                            40.4             14.0   66.5        190  182.1
6    False       False         43.6             26.0   77.2        375  231.9
9    False       False         39.8             18.5   58.9        305  201.3
10                             NA               NA     NA         NA     NA
12                             NA               NA     NA         NA     NA
19   False       False         34.1             17.4   56.2        329  190.0
21                           37.8             18.5   65.3        248  183.6
31   False       False         36.5             20.0   69.0        283  203.7
38   False       False         29.8             16.0   56.1        315  188.1
   maximum_seating owner_count price_nom
```

```
2                5           0       low
3                5           0       low
6                6           0       low
9                5           1       low
10              NA           0       low
12              NA           0       low
19               4           1      high
21               5           0      high
31               7           1       low
38               4           1       low
```

```
str(train_class)
```

```
'data.frame':   20463 obs. of  17 variables:
 $ fleet           : chr  "True" "" "" "True" ...
 $ frame_damaged   : chr  "False" "" "" "False" ...
 $ franchise_dealer: chr  "True" "True" "True" "True" ...
 $ has_accidents   : chr  "False" "" "" "False" ...
 $ is_cpo          : chr  "False" "False" "False" "False" ...
 $ is_new          : chr  "False" "True" "True" "False" ...
 $ is_oemcpo       : chr  "False" "False" "False" "False" ...
 $ salvage         : chr  "False" "" "" "False" ...
 $ theft_title     : chr  "False" "" "" "False" ...
 $ back_legroom    : num  41.5 38.6 NA 39.8 36.3 34.8 39 38.5 37.2 34.5 ...
 $ fuel_tank_volume: num  23.3 24.6 NA 18.5 21 17.7 19.2 15.3 15.9 18 ...
 $ height          : num  76.6 69.3 NA 58.9 65.7 63.9 69.9 66.1 67.1 71.8 ...
 $ horsepower      : int  375 295 NA 305 308 240 300 166 176 270 ...
 $ length          : num  210 190 NA 201 190 ...
 $ maximum_seating : int  8 5 NA 5 5 5 7 5 5 5 ...
 $ owner_count     : int  1 0 0 1 2 2 0 1 1 0 ...
 $ price_nom       : Factor w/ 2 levels "low","high": 2 1 1 1 1 1 2 1 1 1 ...
```

```
str(valid_class)
```

```
'data.frame':   8770 obs. of  17 variables:
 $ fleet           : chr  "" "" "False" "True" ...
 $ frame_damaged   : chr  "" "" "False" "False" ...
 $ franchise_dealer: chr  "True" "True" "True" "True" ...
 $ has_accidents   : chr  "" "" "False" "False" ...
 $ is_cpo          : chr  "False" "False" "False" "False" ...
 $ is_new          : chr  "True" "True" "True" "False" ...
 $ is_oemcpo       : chr  "False" "False" "False" "False" ...
 $ salvage         : chr  "" "" "False" "False" ...
 $ theft_title     : chr  "" "" "False" "False" ...
 $ back_legroom    : num  34.6 40.4 43.6 39.8 NA NA 34.1 37.8 36.5 29.8 ...
 $ fuel_tank_volume: num  12.2 14 26 18.5 NA NA 17.4 18.5 20 16 ...
 $ height          : num  59.7 66.5 77.2 58.9 NA NA 56.2 65.3 69 56.1 ...
 $ horsepower      : int  138 190 375 305 NA NA 329 248 283 315 ...
 $ length          : num  160 182 232 201 NA ...
 $ maximum_seating : int  5 5 6 5 NA NA 4 5 7 4 ...
 $ owner_count     : int  0 0 0 1 0 0 1 0 1 1 ...
 $ price_nom       : Factor w/ 2 levels "low","high": 1 1 1 1 1 1 2 2 1 1 ...
```

2. Classification Tree

2.1 The Tree

```
names(train_class)
```

```
 [1] "fleet"            "frame_damaged"    "franchise_dealer" "has_accidents"
 [5] "is_cpo"           "is_new"           "is_oemcpo"        "salvage"
 [9] "theft_title"      "back_legroom"     "fuel_tank_volume" "height"
[13] "horsepower"       "length"           "maximum_seating"  "owner_count"
[17] "price_nom"
```

class_tr is what we used to create a classification decision tree model.

```
class_tr <- rpart(price_nom ~., data = train_class, method = "class",
                  maxdepth = 5)
```

```
prp(class_tr, cex = 0.8, tweak = 1)
```

3. Model Evaluation

3.1 ConfusionMatrix

A. Training Set

Class_tr_train_predict is the data frame on the training data using the previously created classification decision tree model.

```
class_tr_train_predict <- predict(class_tr, train_class,
                                  type = "class")
```

```
summary(class_tr_train_predict)
```

```
  low  high
18659  1804
```

```
# In this case, we have the data imbalance
```

Here we used data imbalance.

```
confusionMatrix(class_tr_train_predict, train_class$price_nom)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   low  high
      low  17384  1275
      high   538  1266

               Accuracy : 0.9114
                 95% CI : (0.9074, 0.9153)
    No Information Rate : 0.8758
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5348

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9700
            Specificity : 0.4982
         Pos Pred Value : 0.9317
         Neg Pred Value : 0.7018
             Prevalence : 0.8758
         Detection Rate : 0.8495
   Detection Prevalence : 0.9118
      Balanced Accuracy : 0.7341

       'Positive' Class : low
```

B. Validation Set

class_tr_valid_predict is the data frame on the validation data using the previously created classification decision tree model.

```
class_tr_valid_predict <- predict(class_tr, valid_class,
                                  type = "class")
summary(class_tr_valid_predict)
```

```
 low high
7997  773
```

```
confusionMatrix(class_tr_valid_predict, valid_class$price_nom)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  low high
      low  7422  575
      high  231  542

               Accuracy : 0.9081
                 95% CI : (0.9019, 0.9141)
    No Information Rate : 0.8726
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5239

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9698
            Specificity : 0.4852
         Pos Pred Value : 0.9281
         Neg Pred Value : 0.7012
             Prevalence : 0.8726
         Detection Rate : 0.8463
   Detection Prevalence : 0.9119
      Balanced Accuracy : 0.7275

       'Positive' Class : low
```

C. Model Evaluation

```
ROSE::roc.curve(valid_class$price_nom, class_tr_valid_predict)
```

```
Area under the curve (AUC): 0.728
```

4. Weighted Sampling

The purpose of the ROSE package is to generate new synthetic examples. Since our data is imbalanced we need to increase the accuracy. So using our weighted sampling methods we learned, we were able to make the data more accurate.

```
library(ROSE)
```

```
names(train_class)
```

```
 [1] "fleet"           "frame_damaged"    "franchise_dealer" "has_accidents"
 [5] "is_cpo"          "is_new"           "is_oemcpo"        "salvage"
 [9] "theft_title"     "back_legroom"     "fuel_tank_volume" "height"
[13] "horsepower"      "length"           "maximum_seating"  "owner_count"
[17] "price_nom"
```

Before the factor

```
# train_class before the factor
str(train_class)
```

```
'data.frame':   20463 obs. of  17 variables:
 $ fleet           : chr  "True" "" "" "True" ...
 $ frame_damaged   : chr  "False" "" "" "False" ...
 $ franchise_dealer: chr  "True" "True" "True" "True" ...
 $ has_accidents   : chr  "False" "" "" "False" ...
 $ is_cpo          : chr  "False" "False" "False" "False" ...
 $ is_new          : chr  "False" "True" "True" "False" ...
 $ is_oemcpo       : chr  "False" "False" "False" "False" ...
 $ salvage         : chr  "False" "" "" "False" ...
 $ theft_title     : chr  "False" "" "" "False" ...
 $ back_legroom    : num  41.5 38.6 NA 39.8 36.3 34.8 39 38.5 37.2 34.5 ...
 $ fuel_tank_volume: num  23.3 24.6 NA 18.5 21 17.7 19.2 15.3 15.9 18 ...
 $ height          : num  76.6 69.3 NA 58.9 65.7 63.9 69.9 66.1 67.1 71.8 ...
 $ horsepower      : int  375 295 NA 305 308 240 300 166 176 270 ...
 $ length          : num  210 190 NA 201 190 ...
 $ maximum_seating : int  8 5 NA 5 5 5 7 5 5 5 ...
 $ owner_count     : int  1 0 0 1 2 2 0 1 1 0 ...
 $ price_nom       : Factor w/ 2 levels "low","high": 2 1 1 1 1 1 2 1 1 1 ...
```

```
train_class$fleet <- as.factor(train_class$fleet)
train_class$frame_damaged <- as.factor(train_class$frame_damaged)
train_class$franchise_dealer <- as.factor(train_class$franchise_dealer)
train_class$has_accidents <- as.factor(train_class$has_accidents)
train_class$is_cpo <- as.factor(train_class$is_cpo)
train_class$is_new <- as.factor(train_class$is_new)
train_class$is_oemcpo <- as.factor(train_class$is_oemcpo)
train_class$salvage <- as.factor(train_class$salvage)
train_class$theft_title <- as.factor(train_class$theft_title)
```

After the factor

```
# train_class after the factor
str(train_class)
```

```
'data.frame':   20463 obs. of  17 variables:
 $ fleet           : Factor w/ 3 levels "","False","True": 3 1 1 3 2 2 1 2 2 1 ...
 $ frame_damaged   : Factor w/ 3 levels "","False","True": 2 1 1 2 2 2 1 2 2 1 ...
 $ franchise_dealer: Factor w/ 2 levels "False","True": 2 2 2 2 1 2 2 1 2 2 ...
 $ has_accidents   : Factor w/ 3 levels "","False","True": 2 1 1 2 2 2 1 2 2 1 ...
 $ is_cpo          : Factor w/ 2 levels "False","True": 1 1 1 1 1 2 1 1 1 1 ...
 $ is_new          : Factor w/ 2 levels "False","True": 1 2 2 1 1 1 2 1 1 2 ...
 $ is_oemcpo       : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ salvage         : Factor w/ 3 levels "","False","True": 2 1 1 2 2 2 1 2 2 1 ...
 $ theft_title     : Factor w/ 3 levels "","False","True": 2 1 1 2 2 2 1 2 2 1 ...
 $ back_legroom    : num  41.5 38.6 NA 39.8 36.3 34.8 39 38.5 37.2 34.5 ...
 $ fuel_tank_volume: num  23.3 24.6 NA 18.5 21 17.7 19.2 15.3 15.9 18 ...
 $ height          : num  76.6 69.3 NA 58.9 65.7 63.9 69.9 66.1 67.1 71.8 ...
 $ horsepower      : int  375 295 NA 305 308 240 300 166 176 270 ...
 $ length          : num  210 190 NA 201 190 ...
 $ maximum_seating : int  8 5 NA 5 5 5 7 5 5 5 ...
 $ owner_count     : int  1 0 0 1 2 2 0 1 1 0 ...
 $ price_nom       : Factor w/ 2 levels "low","high": 2 1 1 1 1 1 2 1 1 1 ...
```

Before the factor for validation set.

```
# valid_class before the factor
str(valid_class)
```

```
'data.frame':   8770 obs. of  17 variables:
 $ fleet           : chr  "" "" "False" "True" ...
 $ frame_damaged   : chr  "" "" "False" "False" ...
 $ franchise_dealer: chr  "True" "True" "True" "True" ...
 $ has_accidents   : chr  "" "" "False" "False" ...
 $ is_cpo          : chr  "False" "False" "False" "False" ...
 $ is_new          : chr  "True" "True" "True" "False" ...
 $ is_oemcpo       : chr  "False" "False" "False" "False" ...
 $ salvage         : chr  "" "" "False" "False" ...
 $ theft_title     : chr  "" "" "False" "False" ...
 $ back_legroom    : num  34.6 40.4 43.6 39.8 NA NA 34.1 37.8 36.5 29.8 ...
 $ fuel_tank_volume: num  12.2 14 26 18.5 NA NA 17.4 18.5 20 16 ...
 $ height          : num  59.7 66.5 77.2 58.9 NA NA 56.2 65.3 69 56.1 ...
 $ horsepower      : int  138 190 375 305 NA NA 329 248 283 315 ...
 $ length          : num  160 182 232 201 NA ...
 $ maximum_seating : int  5 5 6 5 NA NA 4 5 7 4 ...
 $ owner_count     : int  0 0 0 1 0 0 1 0 1 1 ...
 $ price_nom       : Factor w/ 2 levels "low","high": 1 1 1 1 1 1 2 2 1 1 ...
```

```
valid_class$fleet <- as.factor(valid_class$fleet)
valid_class$frame_damaged <- as.factor(valid_class$frame_damaged)
valid_class$franchise_dealer <- as.factor(valid_class$franchise_dealer)
valid_class$has_accidents <- as.factor(valid_class$has_accidents)
valid_class$is_cpo <- as.factor(valid_class$is_cpo)
valid_class$is_new <- as.factor(valid_class$is_new)
```

```
valid_class$is_oemcpo <- as.factor(valid_class$is_oemcpo)
valid_class$salvage <- as.factor(valid_class$salvage)
valid_class$theft_title <- as.factor(valid_class$theft_title)
```

After the factor for validation set.

```
# valid_class after the factor
str(valid_class)
```

```
'data.frame':   8770 obs. of  17 variables:
 $ fleet           : Factor w/ 3 levels "","False","True": 1 1 2 3 1 1 2 1 3 2 ...
 $ frame_damaged   : Factor w/ 3 levels "","False","True": 1 1 2 2 1 1 2 1 2 2 ...
 $ franchise_dealer: Factor w/ 2 levels "False","True": 2 2 2 2 2 2 1 2 1 2 ...
 $ has_accidents   : Factor w/ 3 levels "","False","True": 1 1 2 2 1 1 2 1 2 2 ...
 $ is_cpo          : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ is_new          : Factor w/ 2 levels "False","True": 2 2 2 1 2 2 1 2 1 1 ...
 $ is_oemcpo       : Factor w/ 2 levels "False","True": 1 1 1 1 1 1 1 1 1 1 ...
 $ salvage         : Factor w/ 3 levels "","False","True": 1 1 2 2 1 1 2 1 2 2 ...
 $ theft_title     : Factor w/ 3 levels "","False","True": 1 1 2 2 1 1 2 1 2 2 ...
 $ back_legroom    : num  34.6 40.4 43.6 39.8 NA NA 34.1 37.8 36.5 29.8 ...
 $ fuel_tank_volume: num  12.2 14 26 18.5 NA NA 17.4 18.5 20 16 ...
 $ height          : num  59.7 66.5 77.2 58.9 NA NA 56.2 65.3 69 56.1 ...
 $ horsepower      : int  138 190 375 305 NA NA 329 248 283 315 ...
 $ length          : num  160 182 232 201 NA ...
 $ maximum_seating : int  5 5 6 5 NA NA 4 5 7 4 ...
 $ owner_count     : int  0 0 0 1 0 0 1 0 1 1 ...
 $ price_nom       : Factor w/ 2 levels "low","high": 1 1 1 1 1 1 2 2 1 1 ...
```

```
train_class_df_rose <- ROSE(price_nom ~., data = train_class,
                      seed = 666)$data
```

```
table(train_class_df_rose$price_nom)
```

```
 low high
9265 9484
```

```
# Now we have balance data and ready for the tree
```

Now we have balanced data and we are ready for the tree.

5. Weighted Data Decision Tree

class_tr_2 is used to create a 2nd classification decision tree model for the balance data.

```
class_tr_2 <- rpart(price_nom ~., data = train_class_df_rose,
                    method = "class",
                    maxdepth = 10)
```

```
rpart.plot(class_tr_2, type = 5)
```

5.1 Predict Training Set

class_tr_2_train_class_predict is the 2nd predictions data frame on the training data using the previously created classification decision tree model for the balance data.

```
class_tr_2_train_class_predict <- predict(class_tr_2, train_class_df_rose,
                                  type = "class")
summary(class_tr_2_train_class_predict)
```

```
 low high
8791 9958
```

```
class_tr_2_train_class_predict <- as.factor(class_tr_2_train_class_predict)
train_class_df_rose$price_nom <- as.factor(train_class_df_rose$price_nom)
```

```
confusionMatrix(class_tr_2_train_class_predict, train_class_df_rose$price_nom)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  low high
      low  7682 1109
      high 1583 8375

               Accuracy : 0.8564
                 95% CI : (0.8513, 0.8614)
    No Information Rate : 0.5058
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7126

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8291
            Specificity : 0.8831
         Pos Pred Value : 0.8738
         Neg Pred Value : 0.8410
             Prevalence : 0.4942
         Detection Rate : 0.4097
   Detection Prevalence : 0.4689
```

```
      Balanced Accuracy : 0.8561

       'Positive' Class : low
```

5.2 Predict Validation Set

class_tr_2_valid_class_predict is the 2nd predictions data frame on the validation data using the previously created classification decision tree model for the balance data.

```
class_tr_2_valid_class_predict <- predict(class_tr_2, valid_class,
                                  type = "class")
summary(class_tr_2_valid_class_predict)
```

```
 low high
5983 2787
```

```
class_tr_2_valid_class_predict <- as.factor(class_tr_2_valid_class_predict)
valid_class$price_nom <- as.factor(valid_class$price_nom)
confusionMatrix(class_tr_2_valid_class_predict, valid_class$price_nom)
```

```
Confusion Matrix and Statistics

          Reference
Prediction  low high
      low  5893   90
      high 1760 1027

               Accuracy : 0.7891
                 95% CI : (0.7804, 0.7976)
    No Information Rate : 0.8726
    P-Value [Acc > NIR] : 1

                  Kappa : 0.4208

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.7700
            Specificity : 0.9194
         Pos Pred Value : 0.9850
         Neg Pred Value : 0.3685
             Prevalence : 0.8726
         Detection Rate : 0.6719
   Detection Prevalence : 0.6822
      Balanced Accuracy : 0.8447

       'Positive' Class : low
```

5.3 Model Evaluation

```
ROSE::roc.curve(valid_class$price_nom, class_tr_2_valid_class_predict)
```

```
Area under the curve (AUC): 0.845
```

6. Predict New Record

```
car_test_class_tr <- car_test
car_test_class_tr
```

```
  fleet frame_damaged franchise_dealer has_accidents is_cpo is_new is_oemcpo
1                                True               False   True     False
2 False         False            False          True  False  False     False
3                                 True               False   True     False
4 False         False             True         False  False  False     False
5 False         False             True         False  False  False     False
6 False         False             True         False  False  False     False
  salvage theft_title back_legroom fuel_tank_volume height horsepower length
1                          43.4             24.0   75.5        355  231.7
2   False       False         36.8             18.5   57.6        197  191.5
3                             43.6             26.0   77.2        375  231.9
4   False       False         38.0             15.8   57.5        310  197.5
5   False       False         38.3             18.5   57.9        158  189.2
6   False       False         38.4             24.6   70.7        302  189.7
  maximum_seating owner_count
1               5           0
2               5           2
3               6           0
4               5           1
5               5           1
6               5           1
```

```
car_test_predict_class_tr <- predict(class_tr_2, newdata = car_test_class_tr,
                                     type = "class")
car_test_predict_class_tr
```

```
   1    2    3    4    5    6
high  low high  low  low  low
Levels: low high
```

IV. Best model:

We used two models in our project to understand which we should use to determine what will help Mator understand how to set a price to a car. We used a kNN Model and a Classification tree. Using a kNN model is the most popular when it comes to machine learning since it can be used for both classification and regression tasks. But the purpose of the classification tree is to easily interpretable and handle nonlinear relationships.

I can see that the model Classification Tree has the highest accuracy of both training and validation set it means that the classification tree model is the best model compare to the others.

Overfitting

We have no over fitting in our data. This is because when comparing the model's performance on the training data versus its performance on the test data set, a significant difference was not present.

kNN model

The accuracy of the training set: 0.7111 = 71.11% -> Bad

The accuracy of the validation set: 0.5964 = 59.64% -> Bad

Area under the curve (AUC): 0.680 -> Bad

Decision Tree model

The accuracy of the training set: 0.8564 = 85.64% -> Good

The accuracy of the validation set: 0.7891 = 78.91% -> Good

Area under the curve (AUC): 0.845 -> Good

=> We will use the classification tree model to predict the car test price

We are using the classification tree model because as you can see from above the accuracy for our training, validation and area under the curve is much higher for the decision tree model. Because we see the percentage of the training and validation set is more near to 100% than the kNN model. We know that having a percentage above 80% is accurate. When the AUC is near to 1, that means it would be more accurate. In our case the decision tree AUC is closer to 1 than the kNN model.

V. Predict New Record Based on Best Model

We will use the Decision Tree model to predict the new used car price because of the previous information. Here are the prediction of 6 used cars that Mator need help to predict:

```
car_test_predict_class_tr <- predict(class_tr_2, newdata = car_test_class_tr,
                                     type = "class")
car_test_predict_class_tr
```

```
   1    2    3    4    5    6
high  low high  low  low  low
Levels: low high
```

Solution for predicting new used car price for Mator:

- Car #1: High price
- Car #2: Low price
- Car #3: High price
- Car #4: Low price
- Car #5: Low price
- Car #6: Low price