predictive_project

2024-03-18

Predictive Project

Read data

```r
vehicles <- read.csv("~/Downloads/vehicles.csv")
tail(vehicles)
```

```
##                id
## 426875 7301591199
## 426876 7301591192
## 426877 7301591187
## 426878 7301591147
## 426879 7301591140
## 426880 7301591129
##                                                                                                  url
## 426875   https://wyoming.craigslist.org/ctd/d/atlanta-2018-lexus-gs-gs-350-sedan-4d/7301591199.html
## 426876     https://wyoming.craigslist.org/ctd/d/atlanta-2019-nissan-maxima-sedan-4d/7301591192.html
## 426877      https://wyoming.craigslist.org/ctd/d/atlanta-2020-volvo-s60-t5-momentum/7301591187.html
## 426878   https://wyoming.craigslist.org/ctd/d/atlanta-2020-caddy-cadillac-xt4-sport/7301591147.html
## 426879   https://wyoming.craigslist.org/ctd/d/atlanta-2018-lexus-es-es-350-sedan-4d/7301591140.html
## 426880 https://wyoming.craigslist.org/ctd/d/atlanta-2019-bmw-series-430i-gran-coupe/7301591129.html
##         region                     region_url price year manufacturer
## 426875 wyoming https://wyoming.craigslist.org 33590 2018        lexus
## 426876 wyoming https://wyoming.craigslist.org 23590 2019       nissan
## 426877 wyoming https://wyoming.craigslist.org 30590 2020        volvo
## 426878 wyoming https://wyoming.craigslist.org 34990 2020     cadillac
## 426879 wyoming https://wyoming.craigslist.org 28990 2018        lexus
## 426880 wyoming https://wyoming.craigslist.org 30590 2019          bmw
##                           model condition   cylinders   fuel odometer
## 426875          gs 350 sedan 4d      good 6 cylinders    gas    30814
## 426876        maxima s sedan 4d      good 6 cylinders    gas    32226
## 426877 s60 t5 momentum sedan 4d      good                gas    12029
## 426878         xt4 sport suv 4d      good             diesel     4174
## 426879          es 350 sedan 4d      good 6 cylinders    gas    30112
## 426880 4 series 430i gran coupe      good                gas    22716
##        title_status transmission               VIN drive size      type
## 426875        clean    automatic JTHBZ1BLXJA012999   rwd          sedan
## 426876        clean        other 1N4AA6AV6KC367801   fwd          sedan
## 426877        clean        other 7JR102FKXLG042696   fwd          sedan
## 426878        clean        other 1GYFZFR46LF088296            hatchback
## 426879        clean        other 58ABK1GG4JU103853   fwd          sedan
## 426880        clean        other WBA4J1C58KBM14708   rwd          coupe
##        paint_color
## 426875       white
## 426876
## 426877         red
## 426878       white
## 426879      silver
## 426880
##                                                                   image_url
## 426875 https://images.craigslist.org/00I0I_hJHfjCUppaEz_0gw0co_600x450.jpg
## 426876 https://images.craigslist.org/00o0o_iiraFnHg8qUz_0gw0co_600x450.jpg
## 426877 https://images.craigslist.org/00x0x_15sbgnxCISvz_0gw0co_600x450.jpg
## 426878 https://images.craigslist.org/00L0L_farM7bxnxRiz_0gw0co_600x450.jpg
## 426879 https://images.craigslist.org/00z0z_bKnIVGLkDTcz_0gw0co_600x450.jpg
## 426880 https://images.craigslist.org/00Y0Y_lEUocjyRxaJz_0gw0co_600x450.jpg
##
## 426875                                                                              Carvana is the sa
## 426876
## 426877                                                                    Carvana is the safer way to
## 426878
## 426879 Carvana is the safer way to buy a car During these uncertain times, Carvana is dedicated to en
## 426880
##         county state      lat      long             posting_date
## 426875     NA    wy 33.77921 -84.41181 2021-04-04T03:21:34-0600
## 426876     NA    wy 33.78650 -84.44540 2021-04-04T03:21:31-0600
## 426877     NA    wy 33.78650 -84.44540 2021-04-04T03:21:29-0600
## 426878     NA    wy 33.77921 -84.41181 2021-04-04T03:21:17-0600
## 426879     NA    wy 33.78650 -84.44540 2021-04-04T03:21:11-0600
## 426880     NA    wy 33.77921 -84.41181 2021-04-04T03:21:07-0600
```

```r
# check number of rows and columns
rows <- nrow(vehicles)
col <- ncol(vehicles)
print(paste("rows",rows))
```

```
## [1] "rows 426880"
```

```r
print(paste("col",col))
```

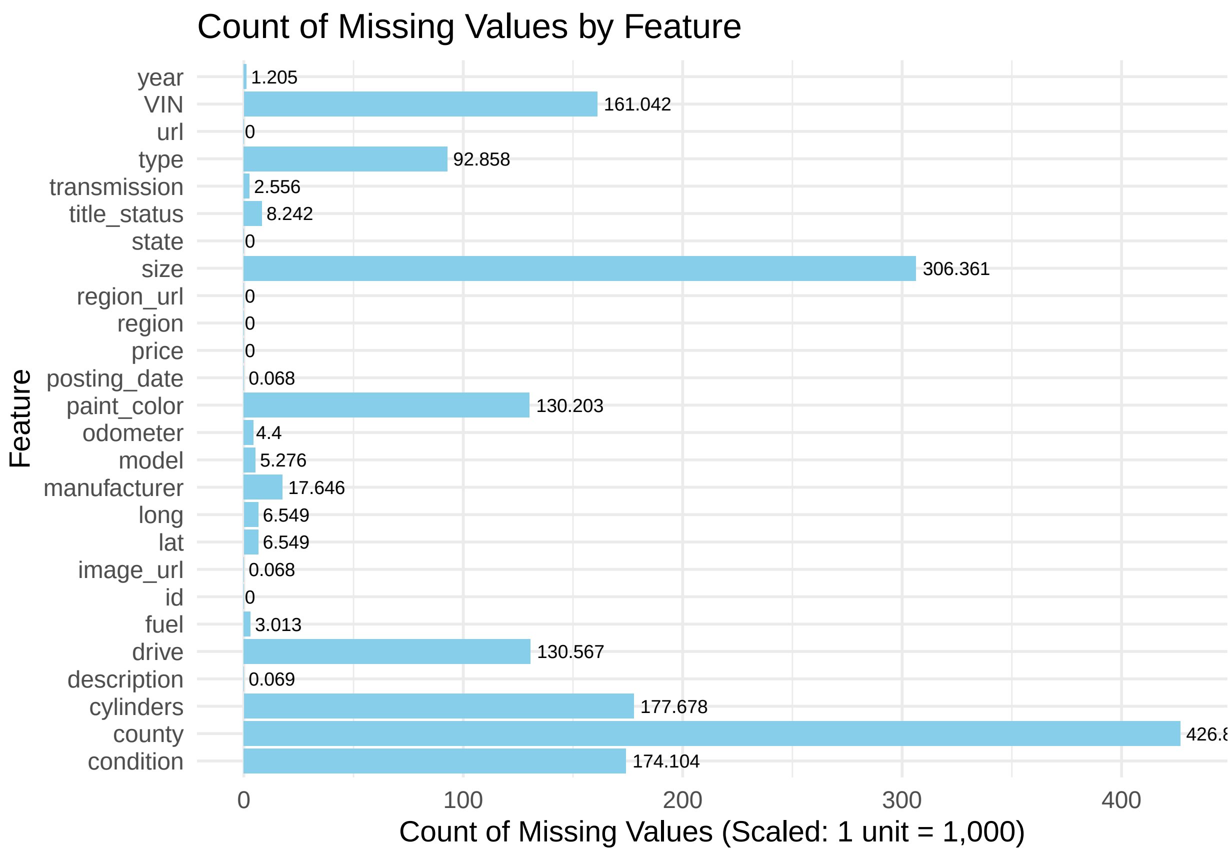
```
## [1] "col 26"
```

step.1 : Check for null or empty cells in dataset

```r
# Checking the null values of each feature by plotting
library(ggplot2)

feature_miss_count <- colSums(is.na(vehicles) | vehicles == "")
print(data.frame(feature_miss_count))
```

```
##              feature_miss_count
## id                            0
## url                           0
## region                        0
## region_url                    0
## price                         0
## year                       1205
## manufacturer              17646
## model                      5276
## condition                174104
## cylinders                177678
## fuel                       3013
## odometer                   4400
## title_status               8242
## transmission               2556
## VIN                      161042
## drive                    130567
## size                     306361
## type                      92858
## paint_color              130203
## image_url                    68
## description                  69
## county                   426880
## state                         0
## lat                        6549
## long                       6549
## posting_date                 68
```

```
scaling <- 1000  # 1 unit represents 1,000 missing values
scaled_miss <- feature_miss_count / scaling

missing_df <- data.frame(
  Feature = names(scaled_miss),
  Scaled_Missing_Count = scaled_miss
)

missing_df <- missing_df[order(-missing_df$Scaled_Missing_Count), ]

ggplot(missing_df, aes(x = Scaled_Missing_Count, y = Feature)) +
  geom_bar(stat = "identity", fill = "skyblue") +
    geom_text(aes(label = Scaled_Missing_Count), vjust = 0.5, color = "black",size=2.6,hjust = - 0.1) +
  labs(
    title = "Count of Missing Values by Feature",
    x = "Count of Missing Values (Scaled: 1 unit = 1,000)",
    y = "Feature"
  ) +
  theme_minimal()
```



```r
# Step2 : Dropping unwanted columns

# We can see that few columns are unwanted for our use case. So we are dropping them
# columns are 'county','url', 'region_url', 'VIN', 'image_url','region','description','model'

vehicles <- vehicles[, -c(2,3,4,8,15,20,21,22)]
```

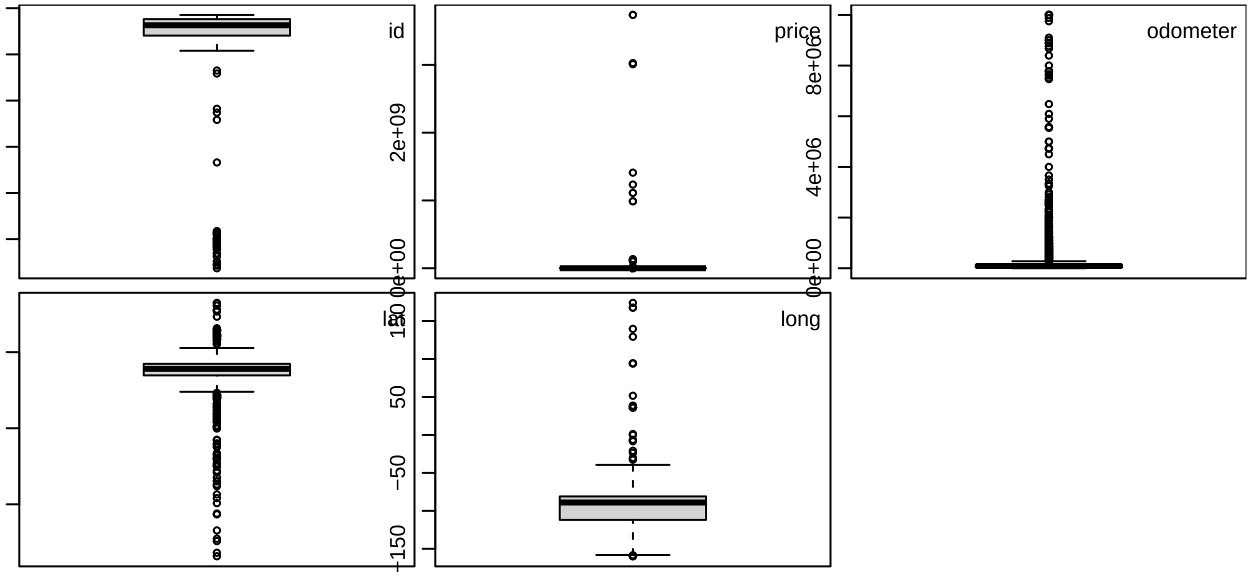
```r
# Step3 : Filling missing values

# If there is any feature with numeric as its datatype, we are replacing with mean of that feature
# Else if there is any feature with character as its datatype, we are replacing with mode of that featu

for (i in names(vehicles)[!names(vehicles) %in% c('year','manufacturer', 'paint_color')]) {
  if (class(vehicles[[i]]) == "numeric" || class(vehicles[[i]]) == "integer") {
    non_empty_values <- as.numeric(vehicles[[i]][vehicles[[i]] != ""])
    mean_val <- mean(non_empty_values, na.rm = TRUE)
    vehicles[[i]][is.na(vehicles[[i]]) | vehicles[[i]] == ""] <- mean_val
  }
  if (class(vehicles[[i]]) == "character") {
    non_empty_values <- vehicles[[i]][vehicles[[i]] != ""]
    mode_val <- names(sort(table(non_empty_values), decreasing = TRUE))[1]  # Find the mode
    vehicles[[i]][is.na(vehicles[[i]]) | vehicles[[i]] == ""] <- mode_val
  }
}
```

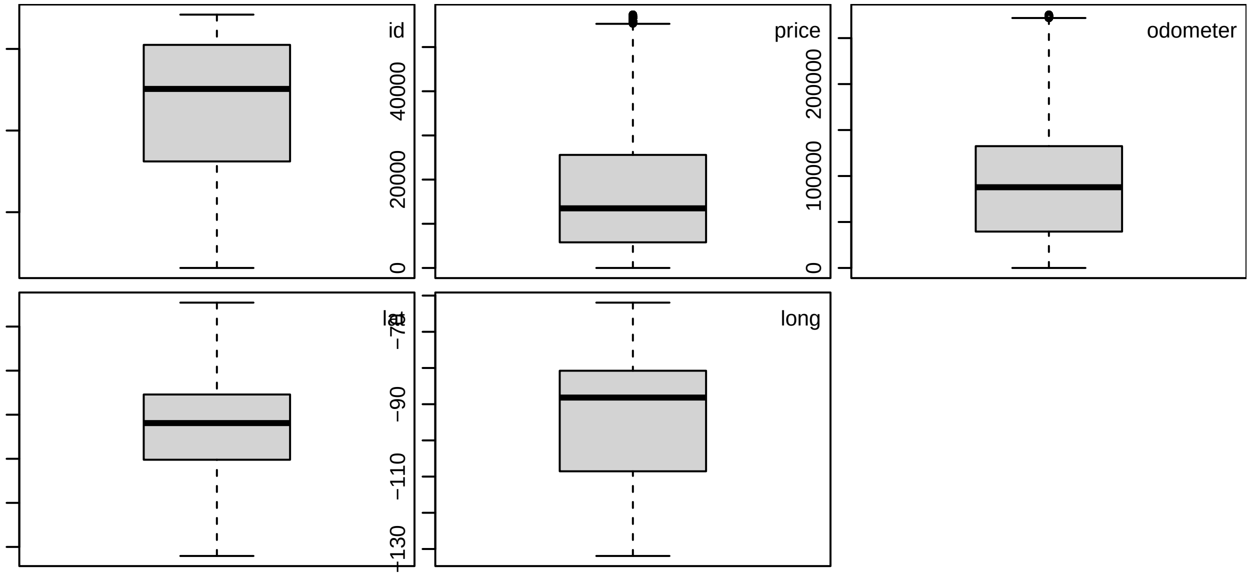
```
# Handling missing values for specific columns (i.e year,manufacturer,paint_color)
vehicles$year[is.na(vehicles$year) | vehicles$year == ""] <- names(sort(table(vehicles$year), decreasing
vehicles$manufacturer[is.na(vehicles$manufacturer) | vehicles$manufacturer == ""] <- "Unknown"
vehicles$paint_color[is.na(vehicles$paint_color) | vehicles$paint_color == ""] <- "Unknown"
```

```
head(vehicles)
```

```
##           id price year manufacturer condition   cylinders fuel odometer
## 1 7222695916  6000 2017      Unknown      good 6 cylinders  gas 98043.33
## 2 7218891961 11900 2017      Unknown      good 6 cylinders  gas 98043.33
## 3 7221797935 21000 2017      Unknown      good 6 cylinders  gas 98043.33
## 4 7222270760  1500 2017      Unknown      good 6 cylinders  gas 98043.33
## 5 7210384030  4900 2017      Unknown      good 6 cylinders  gas 98043.33
## 6 7222379453  1600 2017      Unknown      good 6 cylinders  gas 98043.33
##   title_status transmission drive      size  type paint_color state      lat
## 1        clean    automatic   4wd full-size sedan     Unknown    az 38.49394
## 2        clean    automatic   4wd full-size sedan     Unknown    ar 38.49394
## 3        clean    automatic   4wd full-size sedan     Unknown    fl 38.49394
## 4        clean    automatic   4wd full-size sedan     Unknown    ma 38.49394
## 5        clean    automatic   4wd full-size sedan     Unknown    nc 38.49394
## 6        clean    automatic   4wd full-size sedan     Unknown    ny 38.49394
##       long             posting_date
## 1 -94.7486 2021-04-23T22:13:05-0400
## 2 -94.7486 2021-04-23T22:13:05-0400
## 3 -94.7486 2021-04-23T22:13:05-0400
## 4 -94.7486 2021-04-23T22:13:05-0400
## 5 -94.7486 2021-04-23T22:13:05-0400
## 6 -94.7486 2021-04-23T22:13:05-0400
```

```
# Box plots before and after cleaning the data
plot_boxplots <- function(data) {
  numeric_cols <- sapply(data, is.numeric)
  par(mfrow = c(3, 3), plt = c(0.05, 1, 0.05, 1))  # Adjust rows, columns, and margins as needed
  for (col in names(data)[numeric_cols]) {
    boxplot(data[[col]], main = col)
    legend("topright", legend = col, bty = "n")
  }
}

# Plot boxplots for all numeric features
plot_boxplots(vehicles)



# Function to remove outliers across all features
remove_outliers_all <- function(data) {
  for (col in names(data)) {
    if (is.numeric(data[[col]])) {
      q1 <- quantile(data[[col]], 0.25)
      q3 <- quantile(data[[col]], 0.75)
      iqr <- q3 - q1
      lower_bound <- q1 - 1.5 * iqr
      upper_bound <- q3 + 1.5 * iqr
      data <- data[data[[col]] >= lower_bound & data[[col]] <= upper_bound, ]
    }
  }
  return(data)
}


# Remove outliers across all features
vehicles_no_outliers <- remove_outliers_all(vehicles)

# Plot boxplots again to check for outliers removal
plot_boxplots(vehicles_no_outliers)
```



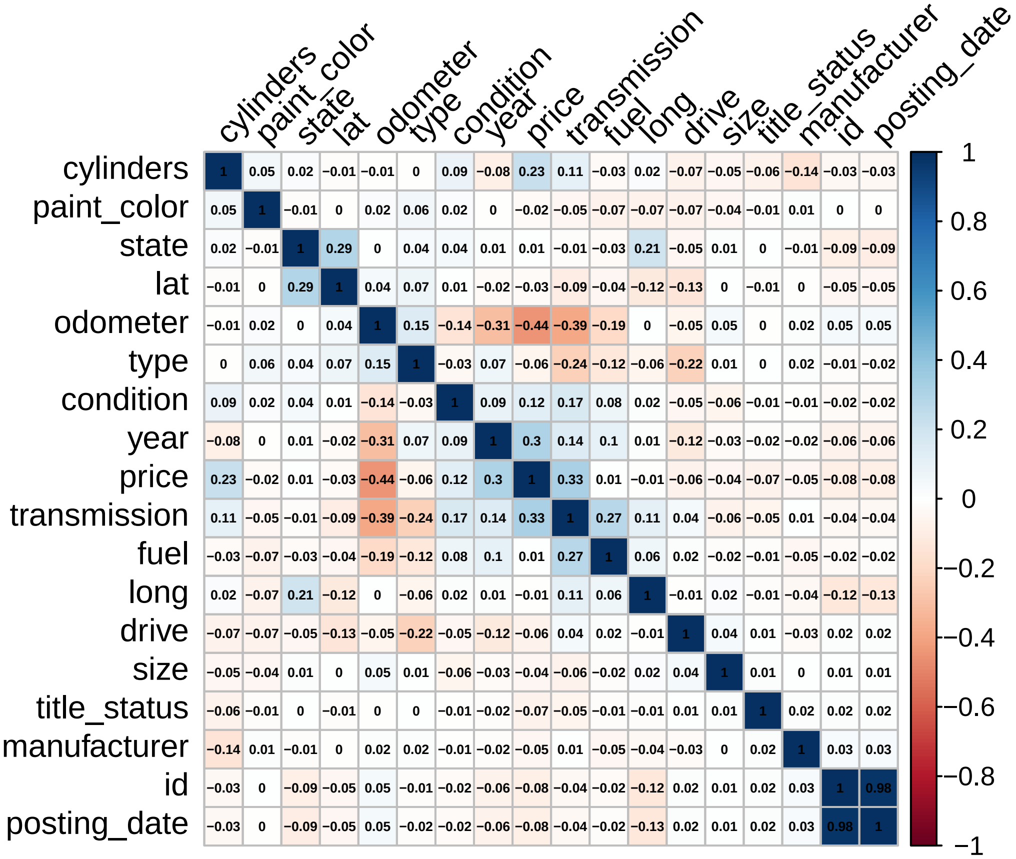


```
# Correlation plot to know about the feature importance

# Remove outliers from the "price" column
vehicles <- remove_outliers_all(vehicles)

# Now, proceed with the label encoding and correlation plot generation as before
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
label_encode_df <- function(dataframe) {
  for (col in names(dataframe)) {
    if (is.factor(dataframe[[col]]) || is.character(dataframe[[col]])) {
      dataframe[[col]] <- as.integer(factor(dataframe[[col]]))
    }
  }
  return(dataframe)
}

# Apply label encoding
df_encoded <- label_encode_df(vehicles)

# Plot correlation matrix
par(mfrow = c(1, 1), mar = c(5, 5, 5, 5))
options(repr.plot.width = 15, repr.plot.height = 15)
```

```r
correlation_matrix <- cor(df_encoded)
corrplot(correlation_matrix, method = "color",
         order = "hclust", tl.col = "black", tl.srt = 45,
         addCoef.col = "black", number.cex = 0.45, mar = c(0, 0, 0, 0), addgrid.col = "gray")
```



```r
head(vehicles)
```

```
##            id price year manufacturer condition   cylinders fuel odometer
## 28 7316814884 33590 2014          gmc      good 8 cylinders  gas    57923
## 29 7316814758 22590 2010    chevrolet      good 8 cylinders  gas    71229
## 30 7316814989 39590 2020    chevrolet      good 8 cylinders  gas    19160
## 31 7316743432 30990 2017       toyota      good 8 cylinders  gas    41124
## 32 7316356412 15000 2013         ford excellent 6 cylinders  gas   128000
## 33 7316343444 27990 2012          gmc      good 8 cylinders  gas    68696
##    title_status transmission drive      size   type paint_color state    lat
## 28        clean        other   4wd full-size pickup       white    al 32.590
## 29        clean        other   4wd full-size pickup        blue    al 32.590
## 30        clean        other   4wd full-size pickup         red    al 32.590
## 31        clean        other   4wd full-size pickup         red    al 32.590
## 32        clean    automatic   rwd full-size  truck       black    al 32.592
## 33        clean        other   4wd full-size pickup       black    al 32.590
##        long             posting_date
## 28 -85.4800 2021-05-04T12:31:18-0500
## 29 -85.4800 2021-05-04T12:31:08-0500
## 30 -85.4800 2021-05-04T12:31:25-0500
## 31 -85.4800 2021-05-04T10:41:31-0500
## 32 -85.5189 2021-05-03T14:02:03-0500
## 33 -85.4800 2021-05-03T13:41:25-0500
```

```
# Step4 : Checking duplicated records

# Checking duplicated rows
duplicates <- vehicles[duplicated(vehicles), ]

# Printing duplicated rows if any
print(duplicates)
```

```
##  [1] id           price        year         manufacturer condition
##  [6] cylinders    fuel         odometer     title_status transmission
## [11] drive        size         type         paint_color  state
## [16] lat          long         posting_date
## <0 rows> (or 0-length row.names)
```

The above value of 0 concludes that there are no duplicated records.

```
# Step5 : Transforming "Odometer" feature into categories of (low,medium,high)

quan25 <- quantile(vehicles$odometer,0.25)
quan50 <- quantile(vehicles$odometer,0.50)

a <- function(val){if(val< quan25){
  return('Low')
} else if(val> quan25 & val< quan50){
  return('Medium')
} else{
  return('High')
}}

vehicles$odometer_status <- sapply(vehicles$odometer,a)
```

```
# Step6 : Transforming "Postingdate" feature by removing the timeframe and just keeping the date as dat
b <- function(val){
  substring(val,1,10)
}
vehicles$posting_date <- sapply(vehicles$posting_date,b)
```

```
# Step7 : Transforming "cylinders" feature by removing the non-numeric characters and considering only
vehicles$cylinders <- as.integer(substr(gsub("^\\D*", "", vehicles$cylinders), 1, 1))
mode <- as.integer(names(sort(table(vehicles$cylinders), decreasing = TRUE))[1])
vehicles$cylinders[is.na(vehicles$cylinders)] <- mode
```

```
head(vehicles)
```

```
##            id price year manufacturer condition cylinders fuel odometer
## 28 7316814884 33590 2014          gmc      good         8  gas    57923
## 29 7316814758 22590 2010    chevrolet      good         8  gas    71229
## 30 7316814989 39590 2020    chevrolet      good         8  gas    19160
## 31 7316743432 30990 2017       toyota      good         8  gas    41124
## 32 7316356412 15000 2013         ford excellent         6  gas   128000
## 33 7316343444 27990 2012          gmc      good         8  gas    68696
##    title_status transmission drive      size   type paint_color state    lat
## 28        clean        other   4wd full-size pickup       white    al 32.590
## 29        clean        other   4wd full-size pickup        blue    al 32.590
## 30        clean        other   4wd full-size pickup         red    al 32.590
## 31        clean        other   4wd full-size pickup         red    al 32.590
## 32        clean    automatic   rwd full-size  truck       black    al 32.592
## 33        clean        other   4wd full-size pickup       black    al 32.590
##        long posting_date odometer_status
## 28 -85.4800   2021-05-04          Medium
## 29 -85.4800   2021-05-04          Medium
## 30 -85.4800   2021-05-04             Low
## 31 -85.4800   2021-05-04          Medium
## 32 -85.5189   2021-05-03            High
## 33 -85.4800   2021-05-03          Medium
```

```
colSums(is.na(vehicles)==TRUE)
```

```
##              id           price            year    manufacturer       condition
##               0               0               0               0               0
##       cylinders            fuel        odometer    title_status    transmission
##               0               0               0               0               0
##           drive            size            type     paint_color           state
##               0               0               0               0               0
##             lat            long    posting_date odometer_status
##               0               0               0               0
```

```
# Install and load necessary packages
options(repos = "https://cran.r-project.org/")
install.packages("ranger")
```

```
## Installing package into '/Users/narayanaroyal/Library/R/x86_64/4.3/library'
## (as 'lib' is unspecified)
```

```
##
## The downloaded binary packages are in
##  /var/folders/w8/s5t40zys2lg2hxnvvxxnyqdm0000gn/T//RtmpFwVP6P/downloaded_packages
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
library(ranger)
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 4.3.2
```

```r
library(rpart)

# Split the data into training and testing sets
index <- createDataPartition(vehicles$price, p = 0.8, list = FALSE)
train_data <- vehicles[index, ]
test_data <- vehicles[-index, ]

# Prepare the data for modeling
x_train <- train_data[, -1]  # Exclude 'price' column
y_train <- train_data$price
x_test <- test_data[, -1]  # Exclude 'price' column
y_test <- test_data$price

# Convert to data frames
x_train <- as.data.frame(x_train)
x_test <- as.data.frame(x_test)

# Convert factor columns to numeric
convert_to_numeric <- function(df) {
  df[] <- lapply(df, function(x) {
    if (is.factor(x)) as.numeric(as.character(x)) else x
  })
  return(df)
}

x_train <- convert_to_numeric(x_train)
x_test <- convert_to_numeric(x_test)

# Remove non-numeric columns
x_train <- x_train[, sapply(x_train, is.numeric)]
x_test <- x_test[, sapply(x_test, is.numeric)]

# Ensure y_train and y_test are numeric
y_train <- as.numeric(y_train)
y_test <- as.numeric(y_test)

# Remove rows with NA values
train_complete_cases <- complete.cases(x_train) & !is.na(y_train)
test_complete_cases <- complete.cases(x_test) & !is.na(y_test)

x_train <- x_train[train_complete_cases, ]
y_train <- y_train[train_complete_cases]

x_test <- x_test[test_complete_cases, ]
y_test <- y_test[test_complete_cases]

# Remove rows with infinite values
x_train <- x_train[apply(x_train, 1, function(row) all(is.finite(row))), ]
x_test <- x_test[apply(x_test, 1, function(row) all(is.finite(row))), ]

# Combine x_train and y_train for ranger
train_data <- cbind(x_train, price = y_train)
```

```r
# Train the random forest model using ranger
model <- ranger(
  formula = price ~ .,
  data = train_data,
  num.trees = 100,
  num.threads = 12,  # Use all 12 cores
  importance = 'impurity'  # Optional: to calculate variable importance
)
# XGBoost model
xgb_model <- xgboost(
  data = as.matrix(x_train),
  label = y_train,
  nrounds = 100,
  nthread = 12,  # Use all 12 cores
  verbose = 0
)

# Decision Tree model
dt_model <- rpart(
  formula = price ~ .,
  data = train_data
)

# Make predictions
# Make predictions
rf_predictions <- predict(model, data = x_test)$predictions
rf_rmse <- sqrt(mean((rf_predictions - y_test)^2))
print(paste("Random Forest RMSE:", rf_rmse))
```

```
## [1] "Random Forest RMSE: 6782.20392081608"
```

```r
# Make predictions
xgb_predictions <- predict(xgb_model, as.matrix(x_test))
xgb_rmse <- sqrt(mean((xgb_predictions - y_test)^2))
print(paste("XGBoost RMSE:", xgb_rmse))
```

```
## [1] "XGBoost RMSE: 11.5523166704152"
```

```r
# Make predictions
dt_predictions <- predict(dt_model, newdata = x_test)
dt_rmse <- sqrt(mean((dt_predictions - y_test)^2))
print(paste("Decision Tree RMSE:", dt_rmse))
```

```
## [1] "Decision Tree RMSE: 11172.610012344"
```

XGboost model is preferred to use in order to make predictions of the secondhand car price.