

Final Project

2023-11-14

Title: "Final Project" author: Sylvie Michaela Essongue

```
options(repos = c(CRAN = "https://cran.r-project.org"))
```

```
library(readr)
Electric_cars <- read_csv("D:/ElectricCars (1).csv" )
```

```
## New names:
## Rows: 180 Columns: 10
## — Column specification
## _____ Delimiter: "," chr
## (2): Name, Battery_life dbl (8): ...1, Acceleration, TopSpeed, Range,
## Efficiency, FastChargeSpeed, N...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## • `` -> `...1`
```

```
head(Electric_cars )
```

...1	Name	Battery_life	Acceleration	TopSp...	Ra...	Efficiency	FastChargeSpeed	Numk
<dbl>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
1	Opel	58 kWh	7.3	150	335	173	210	
2	Renault	31 kWh	22.4	130	160	194	NA	
3	Nissan	36 kWh	7.9	144	220	164	230	
4	Audi	86.5 kWh	5.7	200	375	231	600	
5	Porsche	83.7 kWh	2.8	260	390	215	860	
6	Nissan	36 kWh	14.0	123	165	218	170	

6 rows | 1-9 of 10 columns

```
#Handle missing values :mean, median, or other imputation methods
Electric_cars$Price[is.na(Electric_cars$Price)] <- mean(Electric_cars$Price, na.rm = TRUE)
```

```
Electric_cars$Battery_life <- parse_number(Electric_cars$Battery_life)
```

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.3.3
```

```
##  
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':  
##  
##      format.pval, units
```

```
describe(Electric_cars)
```

```

## Electric_cars
##
## 10 Variables      180 Observations
## -----
## ...1
##      n missing distinct      Info      Mean  pMedian      Gmd      .05
##      180      0      180      1      90.5      90.5      60.33      9.95
##      .10      .25      .50      .75      .90      .95
##      18.90      45.75      90.50      135.25      162.10      171.05
##
## lowest : 1 2 3 4 5, highest: 176 177 178 179 180
## -----
## Name
##      n missing distinct
##      180      0      37
##
## lowest : Aalways      Audi      BMW      Byton      Citroen
## highest: Sono      Tesla      Toyota      Volkswagen      Volvo
## -----
## Battery_life
##      n missing distinct      Info      Mean  pMedian      Gmd      .05
##      180      0      61      0.998      65.26      64.35      25.89      29.95
##      .10      .25      .50      .75      .90      .95
##      37.84      45.00      66.50      77.40      88.00      95.00
##
## lowest : 16.7 21.3 23.8 26.8 28.5 , highest: 105.2 107.8 110 120 200
## -----
## Acceleration
##      n missing distinct      Info      Mean  pMedian      Gmd      .05
##      180      0      77      0.999      7.854      7.7      3.529      3.295
##      .10      .25      .50      .75      .90      .95
##      4.000      5.575      7.500      9.625      12.100      13.110
##
## lowest : 2.1 2.6 2.8 2.9 3 , highest: 13.1 13.3 14 15 22.4
## -----
## TopSpeed
##      n missing distinct      Info      Mean  pMedian      Gmd      .05
##      180      0      35      0.989      176.4      171      44.16      130.0
##      .10      .25      .50      .75      .90      .95
##      130.0      150.0      160.0      200.0      240.1      250.0
##
## lowest : 123 125 130 132 135, highest: 261 262 270 322 410
## -----
## Range
##      n missing distinct      Info      Mean  pMedian      Gmd      .05
##      180      0      68      0.999      337.4      332.5      126.8      170.0
##      .10      .25      .50      .75      .90      .95
##      185.0      258.8      340.0      400.0      455.0      530.2
##
## lowest : 95 100 130 160 165, highest: 610 640 660 750 970
## -----
## Efficiency

```

```
##          n missing distinct      Info      Mean  pMedian      Gmd      .05
##        180         0        80      0.999     194.3     191.5     36.32     155.0
##        .10        .25        .50        .75        .90        .95
##       160.9     168.0     188.5     215.2     244.6     252.0
##
## lowest : 104 144 147 149 150, highest: 258 261 267 270 281
## -----
## FastChargeSpeed
##          n missing distinct      Info      Mean  pMedian      Gmd      .05
##        175         5        68      0.999     481.3     455     261.2     204
##        .10        .25        .50        .75        .90        .95
##       230        290        430        595        836        896
##
## lowest : 120 150 160 170 180, highest: 960 980 1380 1390 1410
## -----
## NumberofSeats
##          n missing distinct      Info      Mean  pMedian      Gmd
##        180         0         4      0.709     5.167         5     0.9115
##
## Value          2      4      5      7
## Frequency        2     28    118    32
## Proportion 0.011 0.156 0.656 0.178
##
## For the frequency table, variable is rounded to the nearest 0
## -----
## Price
##          n missing distinct      Info      Mean  pMedian      Gmd      .05
##        180         0        124      0.985     52939     49795     24575     26043
##        .10        .25        .50        .75        .90        .95
##       28373     36754     52939     55913     84282     110952
##
## lowest : 19200 19795 20495 21620 23995, highest: 118980 125000 138830 139910 189000
## -----
```

```
multiple.regression <- lm(Price ~ .-Name, data = Electric_cars)
summary(multiple.regression)
```

```
##
## Call:
## lm(formula = Price ~ . - Name, data = Electric_cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27439  -8346  -2004   5365  48311
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.524e+05  1.915e+04  -7.959 2.59e-13 ***
## ...1          -2.135e+01  2.048e+01  -1.042 0.298736
## Battery_life   -8.469e+02  2.738e+02  -3.093 0.002323 **
## Acceleration    2.904e+03  7.347e+02   3.952 0.000114 ***
## TopSpeed        5.000e+02  5.183e+01   9.647 < 2e-16 ***
## Range           2.100e+02  5.668e+01   3.704 0.000288 ***
## Efficiency       5.972e+02  1.076e+02   5.552 1.10e-07 ***
## FastChargeSpeed  6.685e-01  7.598e+00   0.088 0.929994
## NumberofSeats   -6.826e+03  1.707e+03  -3.999 9.57e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13200 on 166 degrees of freedom
## (5 observations deleted due to missingness)
## Multiple R-squared:  0.7303, Adjusted R-squared:  0.7173
## F-statistic: 56.2 on 8 and 166 DF,  p-value: < 2.2e-16
```

```
## Random Forest Model
install.packages("dplyr")
```

```
## Installing package into 'C:/Users/Karthi/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'dplyr' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'dplyr'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\Karthi\AppData\Local\R\win-library\4.3\00LOCK\dplyr\libs\x64\dplyr.dll
## to C:\Users\Karthi\AppData\Local\R\win-library\4.3\dplyr\libs\x64\dplyr.dll:
## Permission denied
```

```
## Warning: restored 'dplyr'
```

```
##
## The downloaded binary packages are in
## C:\Users\Karthi\AppData\Local\Temp\RtmpyYRvqO\downloaded_packages
```

```
install.packages("randomForest")
```

```
## Installing package into 'C:/Users/Karthi/AppData/Local/R/win-library/4.3'  
## (as 'lib' is unspecified)
```

```
## package 'randomForest' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'randomForest'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying  
## C:\Users\Karthi\AppData\Local\R\win-library\4.3\00LOCK\randomForest\libs\x64\randomForest.dll  
## to  
## C:\Users\Karthi\AppData\Local\R\win-library\4.3\randomForest\libs\x64\randomForest.dll:  
## Permission denied
```

```
## Warning: restored 'randomForest'
```

```
##  
## The downloaded binary packages are in  
## C:\Users\Karthi\AppData\Local\Temp\RtmpyYRvqO\downloaded_packages
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.3
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
# import any character vectors as a factor (as a categorical variable)  
  
#str(Electric_cars_F2)  
#head(Electric_cars_F2)  
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:randomForest':  
##  
##   combine
```

```
## The following objects are masked from 'package:Hmisc':  
##  
##   src, summarize
```

```
## The following objects are masked from 'package:stats':  
##  
##   filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
# Create a binary target variable  
Electric_cars_F2 <- Electric_cars %>%  
  mutate(Affordability = if_else(Price>=53000, 1,0))
```

```
# Train-Test Split
```

```
library(rsample)
```

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

```
set.seed(645)
```

```
Car_split <- initial_split(Electric_cars_F2, prop = 0.7)
```

```
Car_train <- training(Car_split)
```

```
Car_test <- testing(Car_split)
```

```
## Ensemble of Trees
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.3.3
```

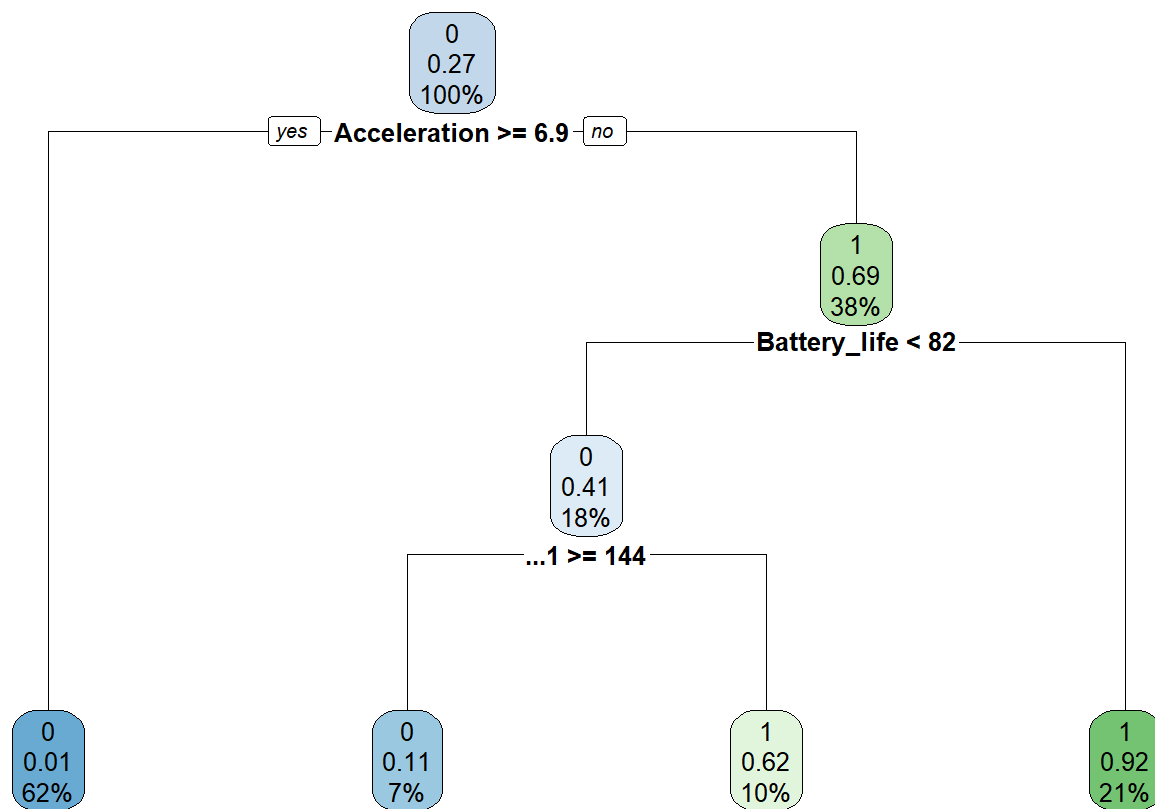
```
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.3.3
```

```
# Classification tree (Single Decision Tree)
```

```
Car_dtree <- rpart(as.factor(Affordability)~. -Price -Name, data = Car_train, method = "class")
```

```
rpart.plot(Car_dtree, cex= 0.8)
```



```
print(Car_dtree)
```



```
## n= 125
##
## node), split, n, loss, yval, (yprob)
##      * denotes terminal node
##
## 1) root 125 34 0 (0.72800000 0.27200000)
##    2) Acceleration>=6.85 77 1 0 (0.98701299 0.01298701) *
##    3) Acceleration< 6.85 48 15 1 (0.31250000 0.68750000)
##      6) Battery_life< 82.2 22 9 0 (0.59090909 0.40909091)
##        12) ...1>=143.5 9 1 0 (0.88888889 0.11111111) *
##        13) ...1< 143.5 13 5 1 (0.38461538 0.61538462) *
##      7) Battery_life>=82.2 26 2 1 (0.07692308 0.92307692) *
```

```
## Random Forest Model
```

```
install.packages("randomForest")
```

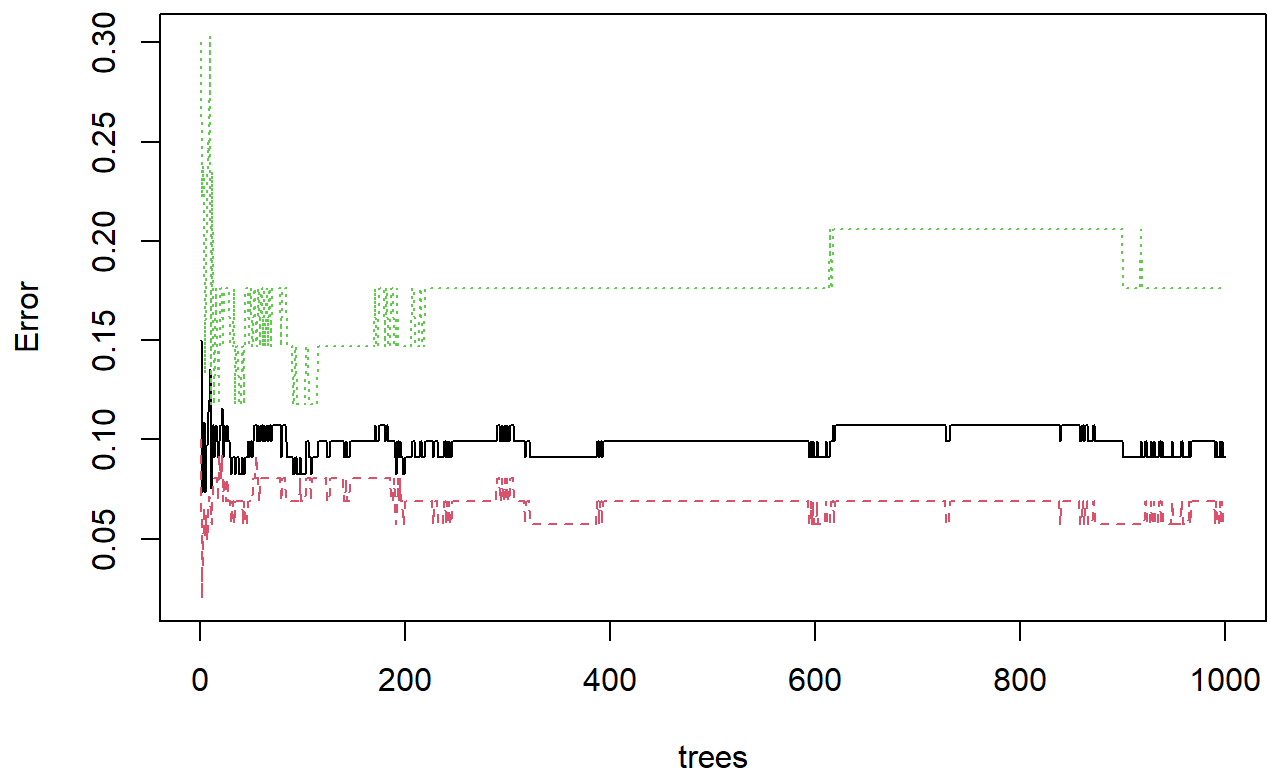
```
## Warning: package 'randomForest' is in use and will not be installed
```

```
library(randomForest)
```

```
rf_car <- randomForest(as.factor(Affordability) ~ . - Price -Name, data = Car_train, ntree = 100
0, importance=TRUE,na.action=na.omit)
```

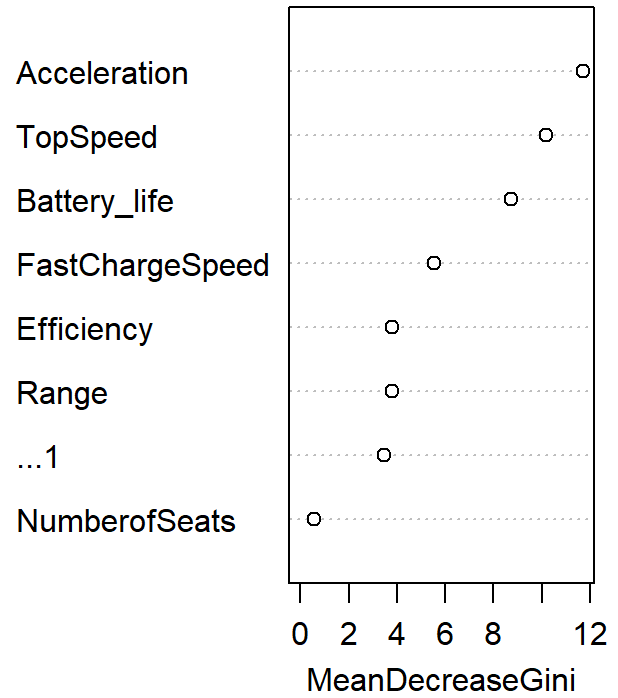
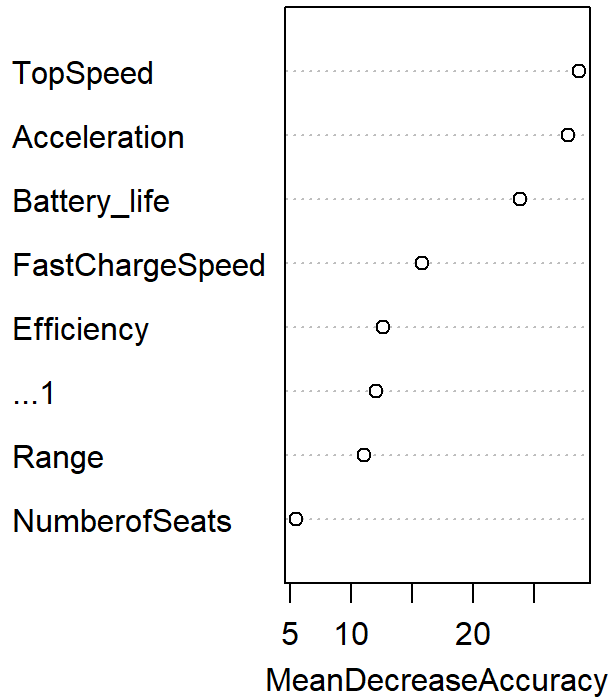
```
plot(rf_car)
```

rf_car



```
varImpPlot(rf_car)
```

rf_car



rf_car

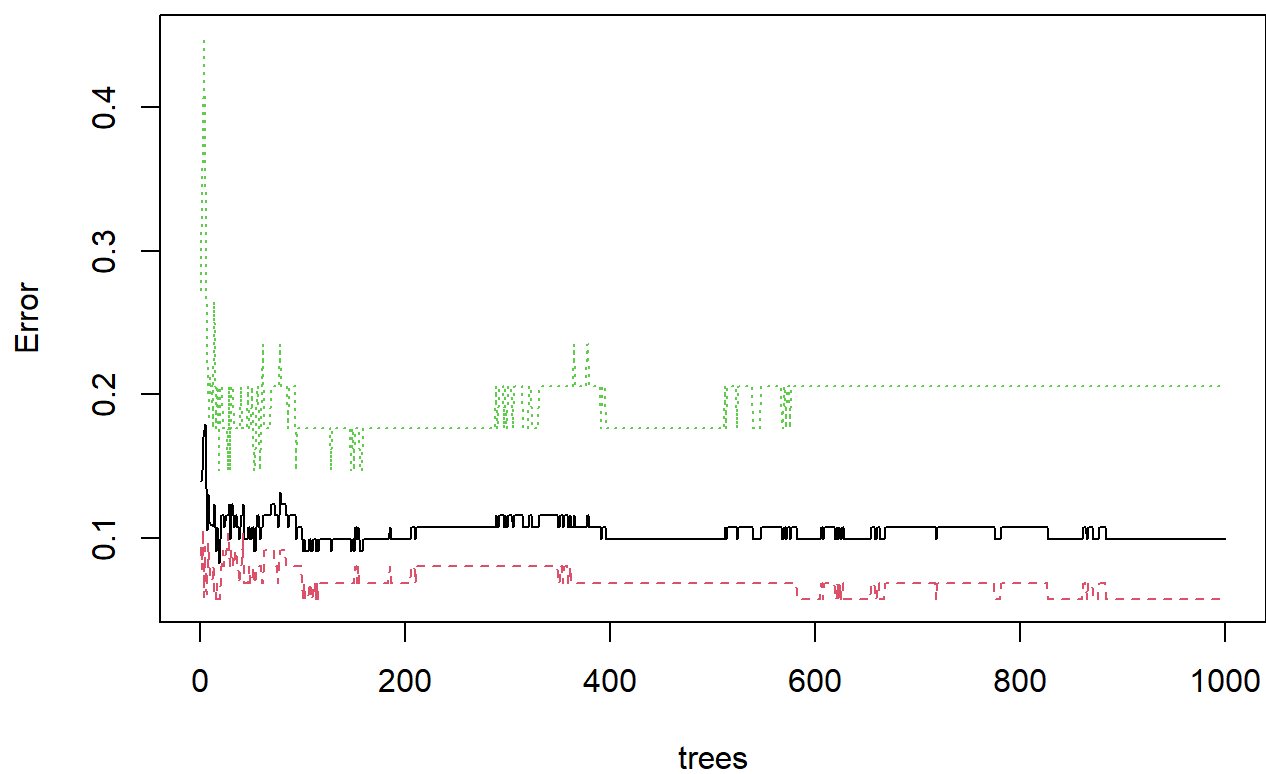
```
##
## Call:
## randomForest(formula = as.factor(Affordability) ~ . - Price -      Name, data = Car_train, n
tree = 1000, importance = TRUE,      na.action = na.omit)
##           Type of random forest: classification
##           Number of trees: 1000
## No. of variables tried at each split: 2
##
##           OOB estimate of  error rate: 9.09%
## Confusion matrix:
##    0  1 class.error
## 0 82  5  0.05747126
## 1  6 28  0.17647059
```

```
# For numeric (not factor) target variables, regression tree is assumed.
```

```
rf_model <- randomForest(as.factor(Affordability)~.-Price -Name , data = Car_train, ntree = 1000, importance=TRUE, na.action=na.omit )
```

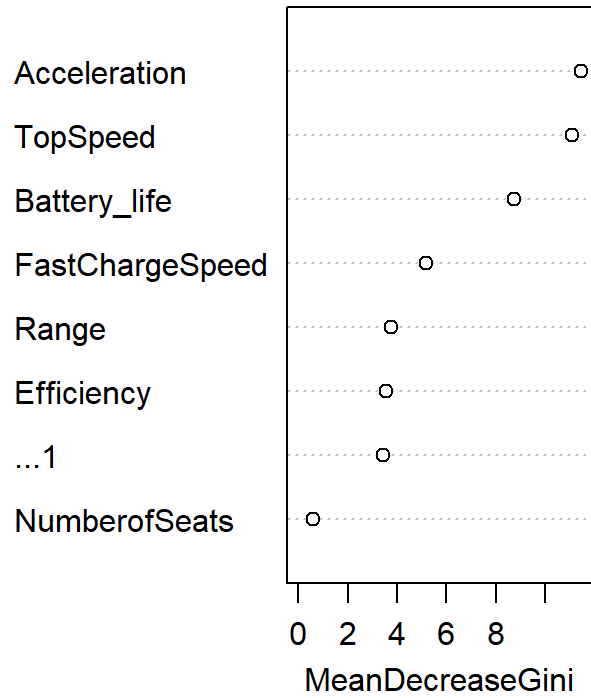
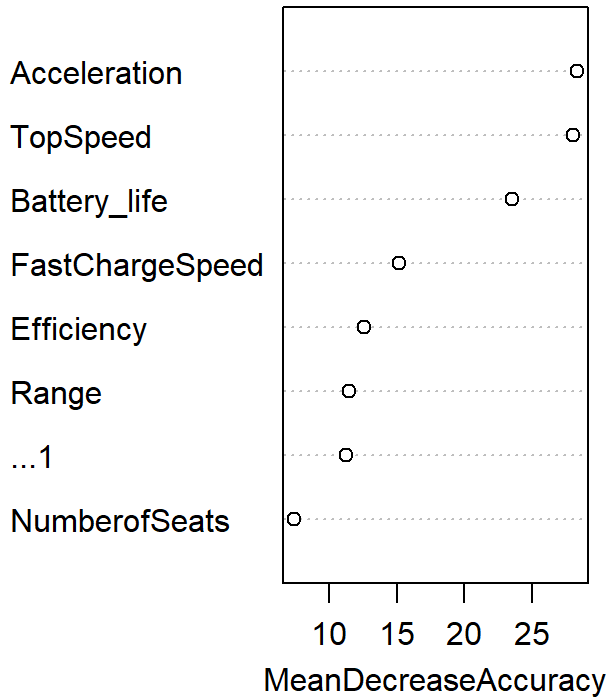
```
plot(rf_model)
```

rf_model



```
varImpPlot(rf_model)
```

rf_model



```
importance(rf_model)
```

##		0	1	MeanDecreaseAccuracy	MeanDecreaseGini
##	...1	7.102406	10.224438	11.234973	3.4414207
##	Battery_life	13.656324	21.147720	23.544460	8.7639416
##	Acceleration	16.341716	27.359468	28.362443	11.4505847
##	TopSpeed	16.689492	26.126362	28.107775	11.1101572
##	Range	7.934924	8.153357	11.454239	3.7770162
##	Efficiency	7.915354	10.259894	12.526668	3.5670279
##	FastChargeSpeed	9.152012	12.970963	15.189831	5.1663445
##	NumberofSeats	7.668060	1.761661	7.379907	0.5801153

```
# Gradient Boosting Model
```

```
install.packages("gbm")
```

```
## Installing package into 'C:/Users/Karthi/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'gbm' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'gbm'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying  
## C:\Users\Karthi\AppData\Local\R\win-library\4.3\00LOCK\gbm\libs\x64\gbm.dll to  
## C:\Users\Karthi\AppData\Local\R\win-library\4.3\gbm\libs\x64\gbm.dll:  
## Permission denied
```

```
## Warning: restored 'gbm'
```

```
##  
## The downloaded binary packages are in  
## C:\Users\Karthi\AppData\Local\Temp\RtmpyYRvq0\downloaded_packages
```

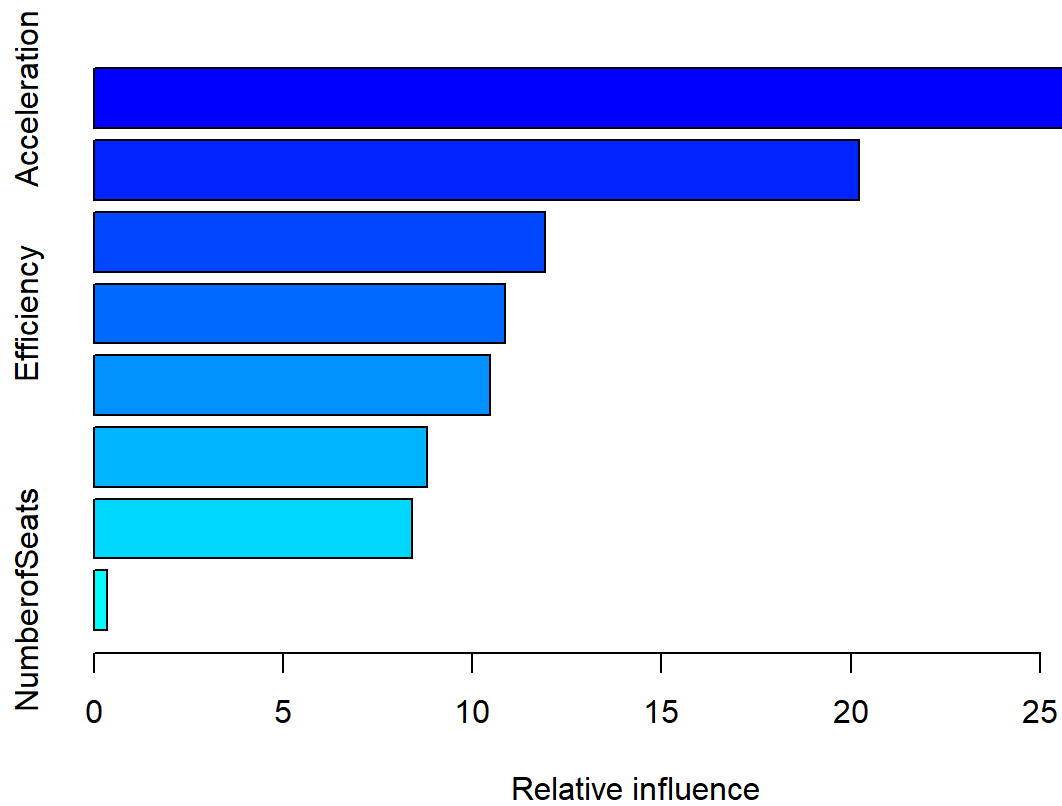
```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.3.3
```

```
## Loaded gbm 2.2.2
```

```
## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.com/gbm-developers/gbm3
```

```
grade_gbm_reg <- gbm(as.factor(Affordability) ~ .-Price -Name, data = Car_train, distribution =  
"gaussian", n.trees = 1000)  
  
summary(grade_gbm_reg)
```



	var <chr>	rel.inf <dbl>
Acceleration	Acceleration	28.9383359
Battery_life	Battery_life	20.2270577
...1	...1	11.9310243
Efficiency	Efficiency	10.8720275
FastChargeSpeed	FastChargeSpeed	10.4732096
Range	Range	8.8055762
TopSpeed	TopSpeed	8.4182771
NumberofSeats	NumberofSeats	0.3344918
8 rows		

```
Electric_cars_F3 <- Electric_cars_F2[Electric_cars_F2$Name %in% c("BMW", "Tesla"), ]
LogisticRegression <- glm(formula = Price ~ . - Name, data = Electric_cars_F3)
summary(LogisticRegression)
```

```
##
## Call:
## glm(formula = Price ~ . - Name, data = Electric_cars_F3)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -230606.32    78608.20  -2.934 0.012520 *
## ...1             63.07      67.28   0.937 0.367017
## Battery_life   -1546.96     776.32  -1.993 0.069539 .
## Acceleration    2504.75    4621.66   0.542 0.597767
## TopSpeed        532.73     120.22   4.431 0.000819 ***
## Range           423.68     191.62   2.211 0.047189 *
## Efficiency       874.09     355.89   2.456 0.030254 *
## FastChargeSpeed -116.78      46.34  -2.520 0.026906 *
## NumberofSeats   1138.90    5031.80   0.226 0.824746
## Affordability   13987.47    7318.18   1.911 0.080136 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 154811763)
##
##    Null deviance: 2.7053e+10  on 21  degrees of freedom
## Residual deviance: 1.8577e+09  on 12  degrees of freedom
## AIC: 485.97
##
## Number of Fisher Scoring iterations: 2
```

```
library(ggplot2)
```

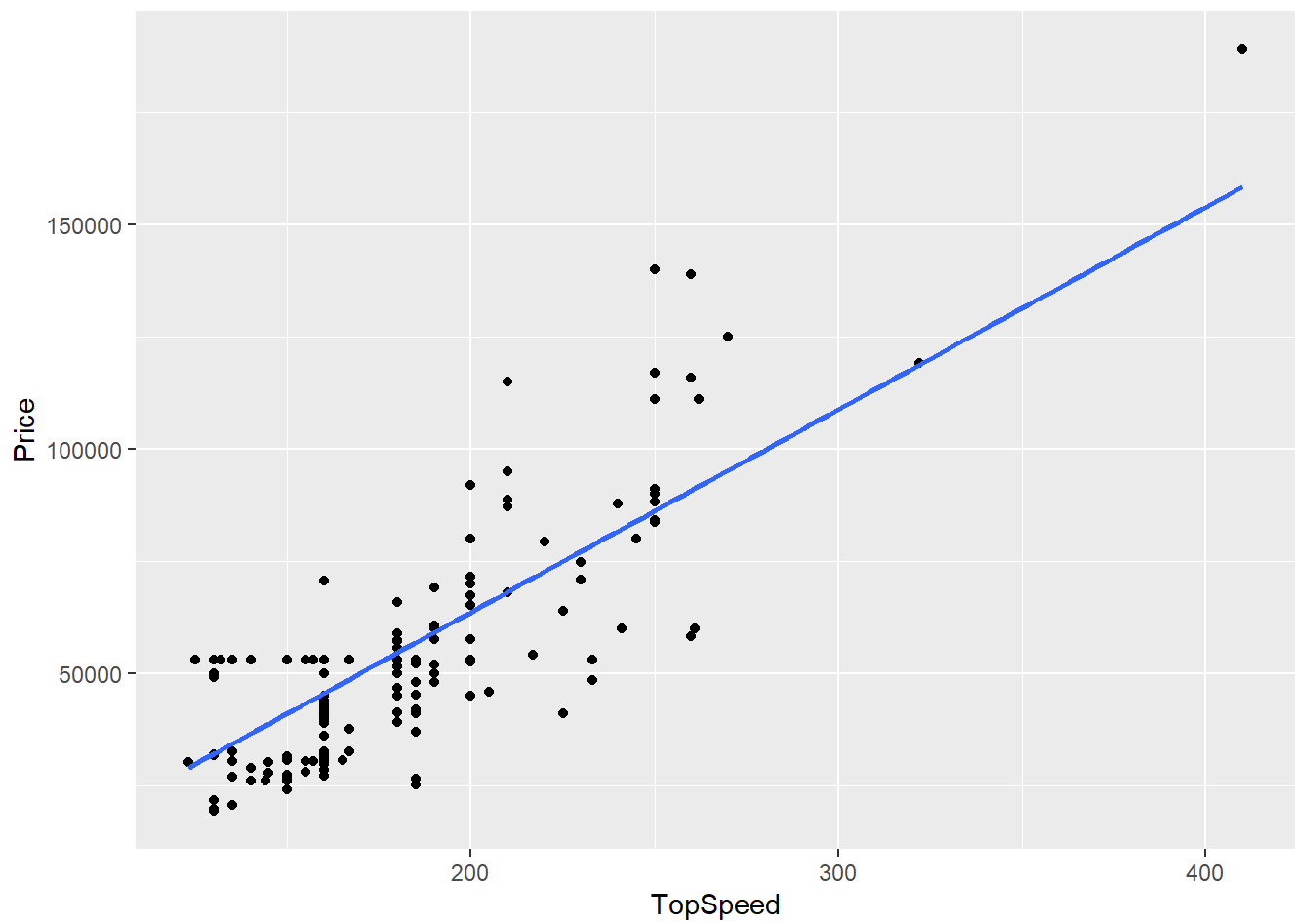
```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##    margin
```

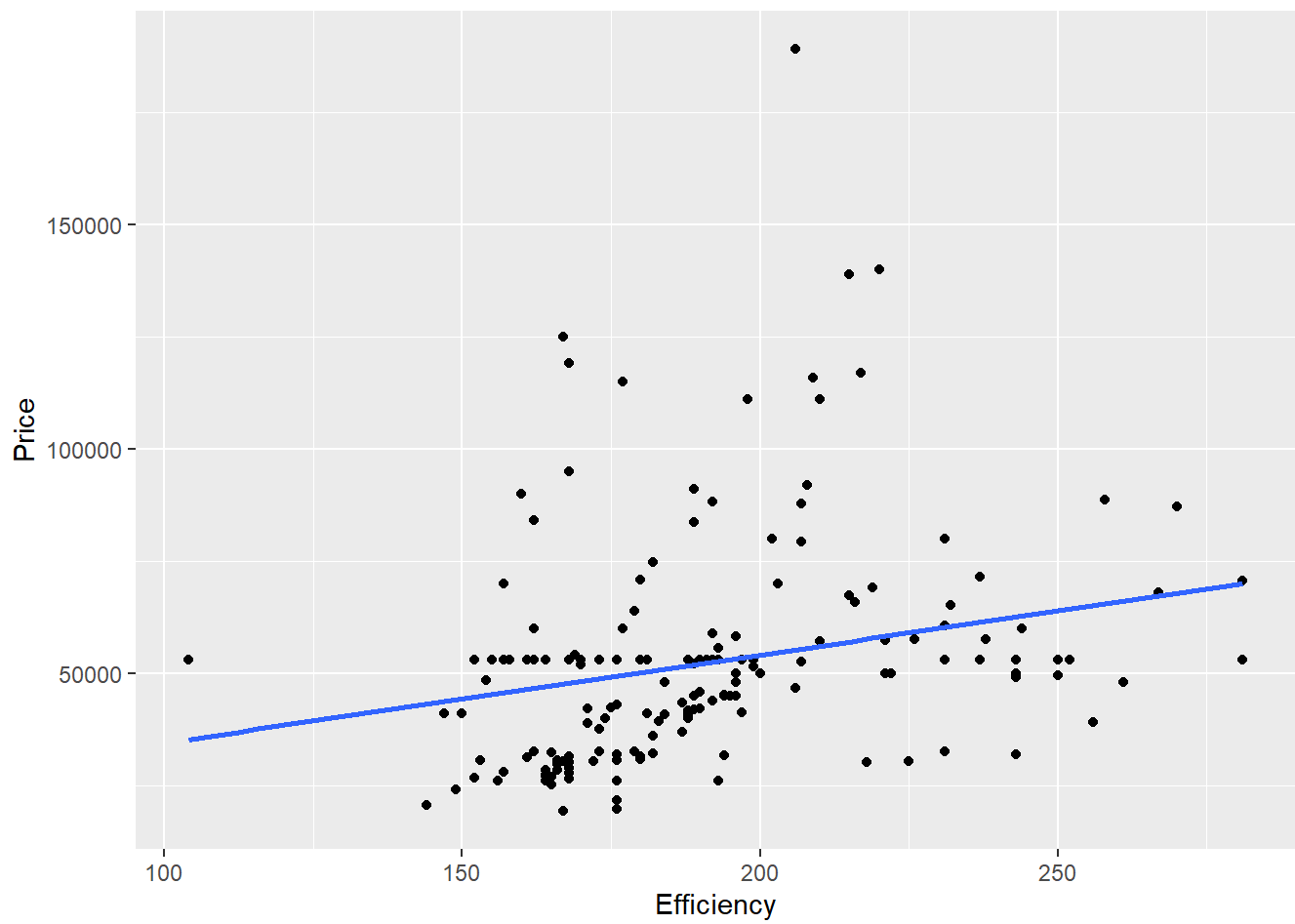
```
ggplot(Electric_cars, aes(x= TopSpeed, y=Price)) +
geom_point() +
geom_smooth(method="lm", se=FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

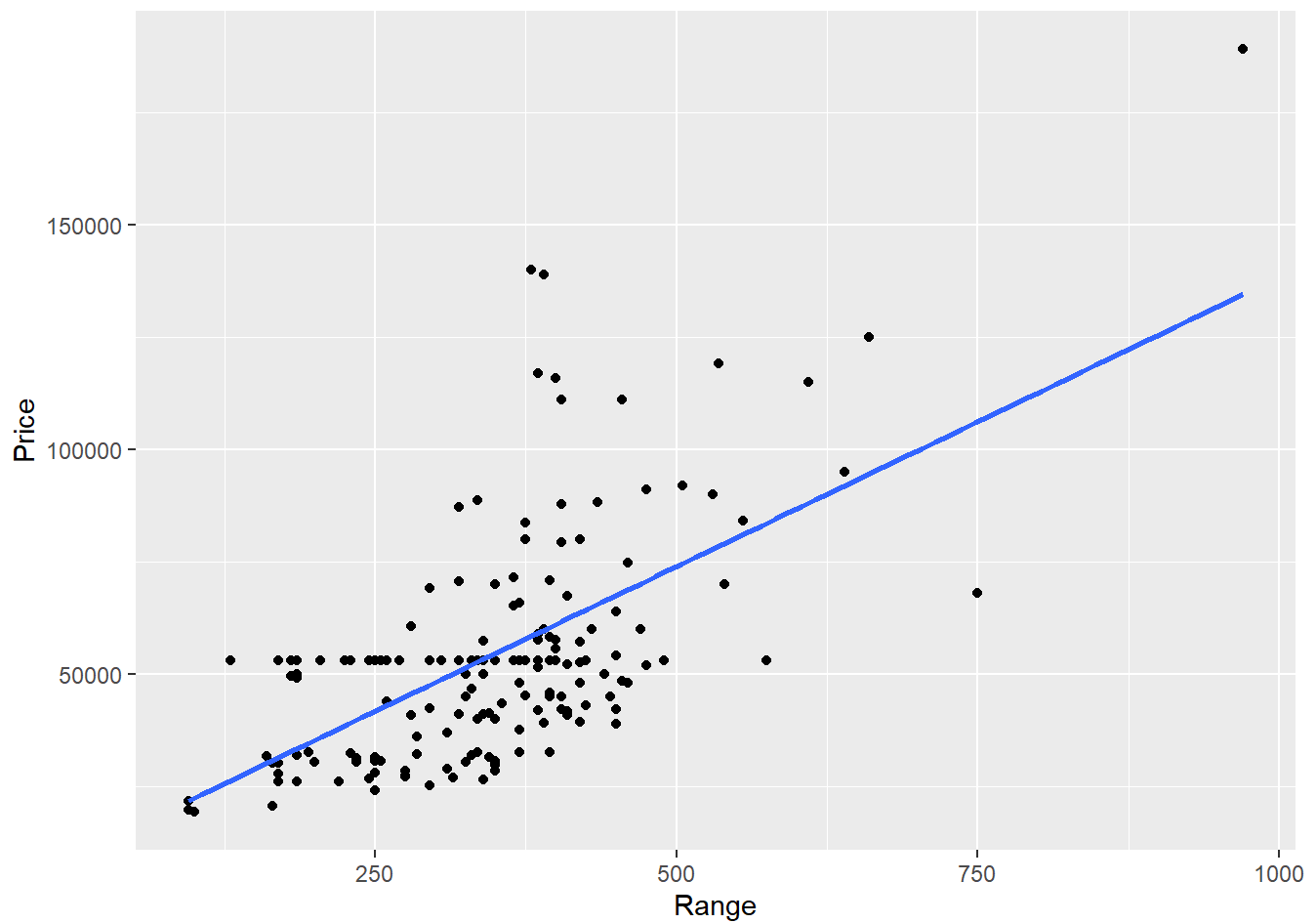
```
ggplot(Electric_cars, aes(x= Efficiency, y=Price)) +  
geom_point() +  
geom_smooth(method="lm", se=FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
ggplot(Electric_cars, aes(x= Range, y=Price)) +  
  geom_point() +  
  geom_smooth(method="lm", se=FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
ggplot(Electric_cars, aes(x= FastChargeSpeed, y=Price)) +  
  geom_point() +  
  geom_smooth(method="lm", se=FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 5 rows containing non-finite outside the scale range  
## (`stat_smooth()`).
```

```
## Warning: Removed 5 rows containing missing values or values outside the scale range  
## (`geom_point()`).
```

