Assignment-4: Predicting Automobile Prices using Linear Regression and ANN

Mithilesh Bhutada-677760107, Vineet Srivastava-671461219, Taksh Ahuja-670941620

```
options(warn=-1)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
library(dplyr)
library(performance)
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(nnet)
library(NeuralNetTools)
library(Metrics)
```

```
##
## Attaching package: 'Metrics'
##
## The following objects are masked from 'package:performance':
##
##     mae, mse, rmse
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

Q1)After your EDA, what factors do you think influence a customer's decision to buy a care? What are the objectives of the model that Farid plans to build?

```
cars <- read_excel('C:/Users/lenovo/Downloads/Cars_r.xlsx')

head(cars)
```

```
## # A tibble: 6 x 28
##   Price Age   KM    Fuel  HP    MC    Colour Auto  CC      Drs   Cyl   Grs Wght
##   <chr> <chr> <chr> <chr> <chr> <chr> <chr>  <chr> <chr> <dbl> <dbl> <dbl> <chr>
## 1 21,0~ 26    31463 Petr~ 195   0     Silver 0     1800      3     3     6 1189
## 2 20,0~ 23    43612 Petr~ 195   0     Red    0     1800      3     3     6 1189
## 3 19,6~ 26    32191 Petr~ 195   0     Red    0     1800      3     3     6 1189
## 4 21,5~ 32    23002 Petr~ 195   1     Black  0     1800      3     3     6 1189
## 5 22,5~ 33    34133 Petr~ 195   1     Grey   0     1800      3     3     6 1189
## 6 22,0~ 29    18741 Petr~ 195   0     Grey   0     1800      3     3     6 1189
## # ... with 15 more variables: `G P` <chr>, Mfr_G <chr>, ABS <dbl>,
## #   Abag_1 <dbl>, Abag_2 <dbl>, AC <dbl>, Comp <dbl>, CD <dbl>, Clock <dbl>,
## #   Pwi <dbl>, PStr <dbl>, Radio <dbl>, SpM <chr>, `M Rim` <chr>, Tow_Bar <chr>
```

```
dim(cars) #31 rows and 28 features.
```

```
## [1] 31 28
```

```
sum(is.na(cars)) #0 which mean no null values are present.
```

```
## [1] 0
```

```
str(cars)
```

```
## tibble [31 x 28] (S3: tbl_df/tbl/data.frame)
##  $ Price  : chr [1:31] "21,000" "20,000" "19,650" "21,550" ...
##  $ Age    : chr [1:31] "26" "23" "26" "32" ...
##  $ KM     : chr [1:31] "31463" "43612" "32191" "23002" ...
##  $ Fuel   : chr [1:31] "Petrol" "Petrol" "Petrol" "Petrol" ...
##  $ HP     : chr [1:31] "195" "195" "195" "195" ...
##  $ MC     : chr [1:31] "0" "0" "0" "1" ...
##  $ Colour : chr [1:31] "Silver" "Red" "Red" "Black" ...
##  $ Auto   : chr [1:31] "0" "0" "0" "0" ...
##  $ CC     : chr [1:31] "1800" "1800" "1800" "1800" ...
##  $ Drs    : num [1:31] 3 3 3 3 3 3 3 3 3 3 ...
##  $ Cyl    : num [1:31] 3 3 3 3 3 3 3 3 3 3 ...
##  $ Grs    : num [1:31] 6 6 6 6 6 6 5 5 5 5 ...
##  $ Wght   : chr [1:31] "1189" "1189" "1189" "1189" ...
##  $ G P    : chr [1:31] "10" "4" "4" "4" ...
##  $ Mfr_G  : chr [1:31] "1" "1" "1" "1" ...
##  $ ABS    : num [1:31] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Abag_1 : num [1:31] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Abag_2 : num [1:31] 1 1 1 1 1 1 1 0 1 1 ...
```

```
##  $ AC     : num [1:31] 1 1 1 1 1 1 1 0 1 0 ...
##  $ Comp   : num [1:31] 0 1 1 1 1 1 1 0 1 1 ...
##  $ CD     : num [1:31] 1 0 0 1 1 0 1 0 1 1 ...
##  $ Clock  : num [1:31] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Pwi    : num [1:31] 1 1 1 1 1 1 1 1 1 1 ...
##  $ PStr   : num [1:31] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Radio  : num [1:31] 0 0 0 0 0 0 0 1 0 0 ...
##  $ SpM    : chr [1:31] "0" "1" "1" "1" ...
##  $ M Rim  : chr [1:31] "1" "1" "1" "1" ...
##  $ Tow_Bar: chr [1:31] "0" "0" "0" "0" ...
```

```
cars$Mfr_G <- ifelse(cars$Mfr_G == "1.0", "1", cars$Mfr_G)

cols <- c("Fuel", "MC", "Auto", "Cyl", "Drs", "Grs", "ABS", "Abag_1", "Abag_2",
          "AC", "Comp", "CD", "Clock", "Pwi", "PStr", "Radio", "SpM", "M Rim",
          "Tow_Bar")
cars[cols] <- lapply(cars[cols], factor)
str(cars)
```

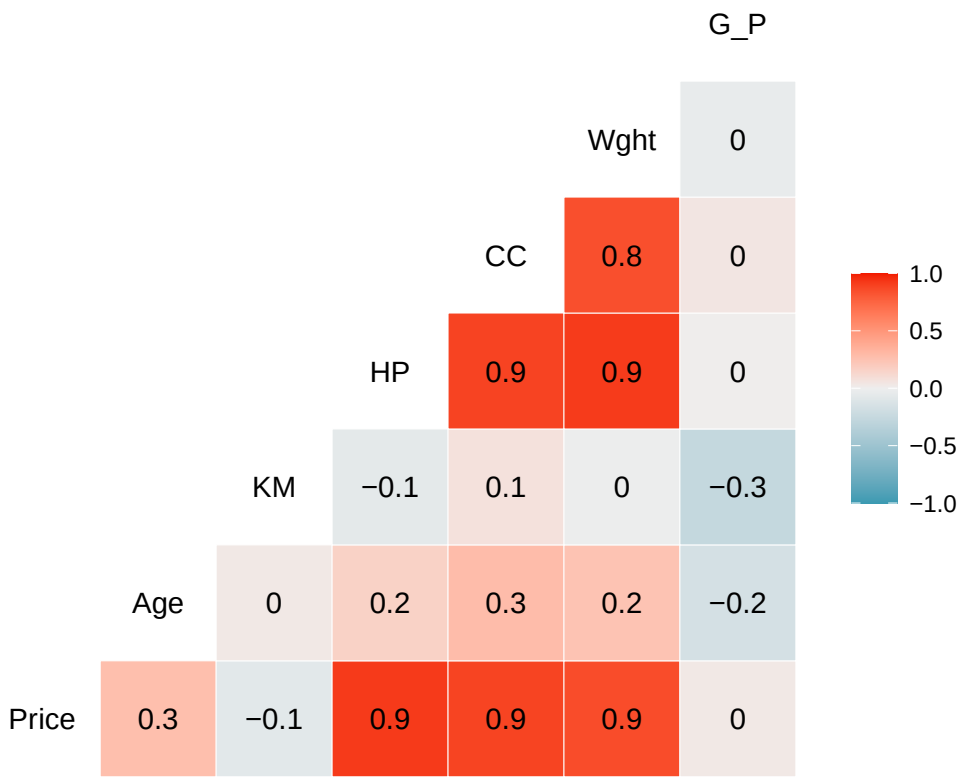
```
## tibble [31 x 28] (S3: tbl_df/tbl/data.frame)
##  $ Price  : chr [1:31] "21,000" "20,000" "19,650" "21,550" ...
##  $ Age    : chr [1:31] "26" "23" "26" "32" ...
##  $ KM     : chr [1:31] "31463" "43612" "32191" "23002" ...
##  $ Fuel   : Factor w/ 1 level "Petrol": 1 1 1 1 1 1 1 1 1 1 ...
##  $ HP     : chr [1:31] "195" "195" "195" "195" ...
##  $ MC     : Factor w/ 2 levels "0","1": 1 1 1 2 2 1 2 2 1 2 ...
##  $ Colour : chr [1:31] "Silver" "Red" "Red" "Black" ...
##  $ Auto   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CC     : chr [1:31] "1800" "1800" "1800" "1800" ...
##  $ Drs    : Factor w/ 1 level "3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Cyl    : Factor w/ 1 level "3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Grs    : Factor w/ 2 levels "5","6": 2 2 2 2 2 2 1 1 1 1 ...
##  $ Wght   : chr [1:31] "1189" "1189" "1189" "1189" ...
##  $ G P    : chr [1:31] "10" "4" "4" "4" ...
##  $ Mfr_G  : chr [1:31] "1" "1" "1" "1" ...
##  $ ABS    : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Abag_1 : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Abag_2 : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 2 2 ...
##  $ AC     : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 2 1 ...
##  $ Comp   : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 1 2 2 ...
##  $ CD     : Factor w/ 2 levels "0","1": 2 1 1 2 2 1 2 1 2 2 ...
##  $ Clock  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Pwi    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PStr   : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Radio  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ SpM    : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 1 1 1 2 ...
##  $ M Rim  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 1 1 ...
##  $ Tow_Bar: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
```

```
#Categorical variables converted from numerical and character to factors

cars$Price <- gsub(",", "", cars$Price)
# Applying gsub function on Price variable as some of the input values were not
```

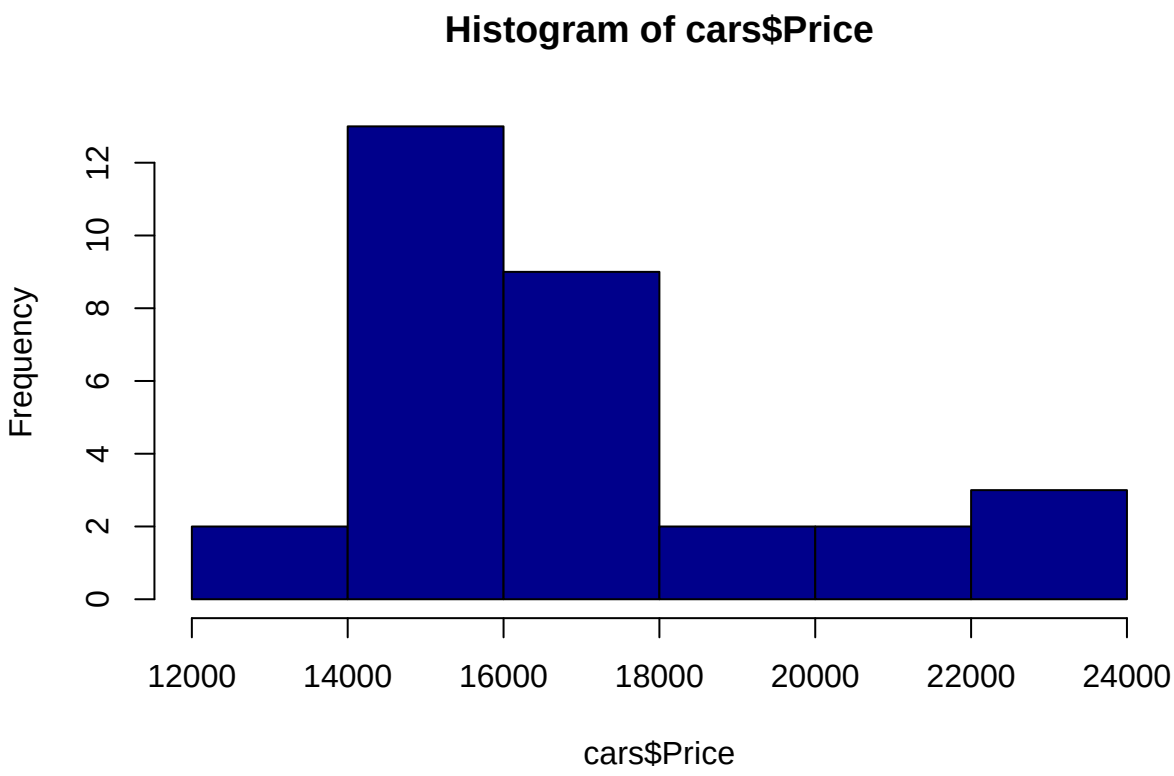
```r
#properly formatted and were showing a warning message.

num_cols <- c("Price", "Age", "KM", "HP", "CC", "Wght", "G P")
cars[num_cols] <- apply(cars[num_cols], 2,  as.numeric)
str(cars)
```

```
## tibble [31 x 28] (S3: tbl_df/tbl/data.frame)
##  $ Price  : num [1:31] 21000 20000 19650 21550 22550 ...
##  $ Age    : num [1:31] 26 23 26 32 33 29 31 25 25 31 ...
##  $ KM     : num [1:31] 31463 43612 32191 23002 34133 ...
##  $ Fuel   : Factor w/ 1 level "Petrol": 1 1 1 1 1 1 1 1 1 1 ...
##  $ HP     : num [1:31] 195 195 195 195 195 195 195 113 113 113 ...
##  $ MC     : Factor w/ 2 levels "0","1": 1 1 1 2 2 1 2 2 1 2 ...
##  $ Colour : chr [1:31] "Silver" "Red" "Red" "Black" ...
##  $ Auto   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CC     : num [1:31] 1800 1800 1800 1800 1800 1800 1800 1600 1600 1600 ...
##  $ Drs    : Factor w/ 1 level "3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Cyl    : Factor w/ 1 level "3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Grs    : Factor w/ 2 levels "5","6": 2 2 2 2 2 2 1 1 1 1 ...
##  $ Wght   : num [1:31] 1189 1189 1189 1189 1189 ...
##  $ G P    : num [1:31] 10 4 4 4 4 4 4 20 4 4 ...
##  $ Mfr_G  : chr [1:31] "1" "1" "1" "1" ...
##  $ ABS    : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Abag_1 : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Abag_2 : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 2 2 ...
##  $ AC     : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 2 1 ...
##  $ Comp   : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 1 2 2 ...
##  $ CD     : Factor w/ 2 levels "0","1": 2 1 1 2 2 1 2 1 2 2 ...
##  $ Clock  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Pwi    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ PStr   : Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Radio  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ SpM    : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 1 1 1 2 ...
##  $ M Rim  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 1 1 ...
##  $ Tow_Bar: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
```

```r
cars$Silver <- as.factor(ifelse(cars$Colour == "Silver", 1, 0))
cars$Red <- as.factor(ifelse(cars$Colour == "Red", 1, 0))
cars$Black <- as.factor(ifelse(cars$Colour == "Black", 1, 0))
cars$Grey <- as.factor(ifelse(cars$Colour == "Grey", 1, 0))
cars$Blue <- as.factor(ifelse(cars$Colour == "Blue", 1, 0))
cars$Green <- as.factor(ifelse(cars$Colour == "Green", 1, 0))
cars$Mfr_G <- ifelse(cars$Mfr_G == "1.0", "1", cars$Mfr_G)
cars$Mfr_G <- as.factor(cars$Mfr_G)
#There is no need to check for outliers in the dataset as each car may have
#separate and unique features.

ggcorr(cars, label = T)
```




```
#Hence as we can see that the Price has a strong positive correlation with HP,
#CC and Weight. However, price is negatively correlated with KM.
#Also we can see strong positive correlation between HP and CC, HP and weight,
#CC and Weight.

cars <- subset(cars, select = -c(Fuel, Drs, Cyl, ABS, Abag_1, PStr, Colour))

hist(cars$Price, col="darkblue")
```




```
#As we can see, the majority of the car prices are in the range of 14000 to
#18000

#The data has been completely cleaned and EDA has been performed.

head(cars)
```

```
## # A tibble: 6 x 27
##   Price   Age    KM    HP MC    Auto     CC Grs    Wght `G P` Mfr_G Abag_2 AC
##   <dbl> <dbl> <dbl> <dbl> <fct> <fct> <dbl> <fct> <dbl> <dbl> <fct> <fct>  <fct>
## 1 21000    26 31463   195 0     0      1800 6      1189    10 1     1      1
## 2 20000    23 43612   195 0     0      1800 6      1189     4 1     1      1
## 3 19650    26 32191   195 0     0      1800 6      1189     4 1     1      1
## 4 21550    32 23002   195 1     0      1800 6      1189     4 1     1      1
## 5 22550    33 34133   195 1     0      1800 6      1189     4 1     1      1
## 6 22050    29 18741   195 0     0      1800 6      1189     4 1     1      1
## # ... with 14 more variables: Comp <fct>, CD <fct>, Clock <fct>, Pwi <fct>,
## #   Radio <fct>, SpM <fct>, `M Rim` <fct>, Tow_Bar <fct>, Silver <fct>,
## #   Red <fct>, Black <fct>, Grey <fct>, Blue <fct>, Green <fct>
```

```
#Forward Variable Selection
null <- lm(Price~1, data=cars)
full <- lm(Price~., data=cars)

variable_sel <- step(null, scope=list(lower=null, upper=full),
                     direction="forward")
```

```
## Start:  AIC=487.17
## Price ~ 1
##
##           Df Sum of Sq       RSS    AIC
## + HP       1 165304301  28935159 430.14
## + CC       1 153800304  40439156 440.52
## + Wght     1 142367468  51871991 448.24
## + Grs      1 112113402  82126057 462.48
## + `M Rim`  1  76547985 117691475 473.64
## + AC       1  73596115 120643345 474.41
## + Clock    1  36568626 157670834 482.70
## + Pwi      1  36568626 157670834 482.70
## + Blue     1  16028445 178211015 486.50
## + Age      1  14424630 179814829 486.78
## + Red      1  14159841 180079619 486.82
## <none>                 194239460 487.17
## + SpM      1   6565339 187674121 488.10
## + Abag_2   1   6558280 187681179 488.10
## + Radio    1   6558280 187681179 488.10
## + Grey     1   6347407 187892052 488.14
## + Tow_Bar  1   4737551 189501909 488.40
## + CD       1   4139032 190100427 488.50
## + MC       1   2646813 191592646 488.74
## + Green    1   2176936 192062524 488.82
## + KM       1   1044577 193194883 489.00
## + Mfr_G    1    727840 193511619 489.05
## + `G P`    1    265172 193974288 489.13
## + Silver   1    135177 194104283 489.15
## + Auto     1     65336 194174124 489.16
## + Comp     1     22269 194217191 489.17
## + Black    1      2051 194237409 489.17
##
## Step:  AIC=430.14
## Price ~ HP
##
##           Df Sum of Sq      RSS    AIC
## + Clock    1   9564869 19370290 419.70
## + Pwi      1   9564869 19370290 419.70
## + Red      1   7584718 21350441 422.72
## + Silver   1   5109142 23826017 426.12
## + Grs      1   4468418 24466741 426.94
## + CC       1   4281913 24653246 427.18
## + CD       1   4247868 24687291 427.22
## + Grey     1   3610698 25324461 428.01
## + Age      1   2914741 26020418 428.85
## <none>                 28935159 430.14
## + MC       1   1616769 27318390 430.36
## + AC       1   1583060 27352099 430.40
## + Mfr_G    1   1500586 27434573 430.49
## + Comp     1    778451 28156708 431.30
## + Abag_2   1    514479 28420680 431.59
## + Radio    1    514479 28420680 431.59
## + SpM      1    439974 28495185 431.67
## + Auto     1    387867 28547292 431.73
```

```
## + `G P`    1    182188 28752971 431.95
## + `M Rim`  1    142592 28792567 431.99
## + Blue     1    134644 28800515 432.00
## + Tow_Bar  1    108881 28826278 432.03
## + Wght     1    106231 28828927 432.03
## + KM       1     62465 28872694 432.08
## + Green    1     49923 28885236 432.09
## + Black    1      3712 28931447 432.14
##
## Step:  AIC=419.7
## Price ~ HP + Clock
##
##           Df Sum of Sq      RSS    AIC
## + Red      1   7047640 12322649 407.68
## + Grs      1   3079556 16290734 416.34
## + Age      1   2261200 17109089 417.86
## + SpM      1   1353579 18016711 419.46
## + Grey     1   1313079 18057210 419.53
## <none>                 19370290 419.70
## + Silver   1   1193839 18176450 419.73
## + Blue     1   1075689 18294600 419.93
## + CC       1    914780 18455509 420.20
## + CD       1    683209 18687080 420.59
## + `M Rim`  1    648993 18721297 420.65
## + `G P`    1    645610 18724680 420.65
## + Mfr_G    1    586574 18783716 420.75
## + KM       1    449179 18921111 420.98
## + MC       1    176050 19194240 421.42
## + AC       1    162116 19208174 421.44
## + Black    1    161059 19209230 421.44
## + Auto     1    145769 19224521 421.47
## + Abag_2   1    119336 19250953 421.51
## + Radio    1    119336 19250953 421.51
## + Tow_Bar  1     84663 19285627 421.57
## + Wght     1     21890 19348399 421.67
## + Green    1      4771 19365519 421.70
## + Comp     1       230 19370060 421.70
##
## Step:  AIC=407.68
## Price ~ HP + Clock + Red
##
##           Df Sum of Sq      RSS    AIC
## + Silver   1   2645283  9677366 402.19
## + Grs      1   1167123 11155526 406.60
## + Blue     1   1001729 11320920 407.05
## <none>                 12322649 407.68
## + `M Rim`  1    675839 11646810 407.93
## + Mfr_G    1    564831 11757818 408.23
## + CC       1    563176 11759473 408.23
## + Black    1    555527 11767122 408.25
## + `G P`    1    306836 12015814 408.90
## + SpM      1    297613 12025036 408.92
## + Grey     1    209460 12113189 409.15
## + Comp     1    169111 12153538 409.25
```

```
## + MC       1    137460 12185189 409.33
## + Age      1    137101 12185548 409.33
## + Auto     1    117258 12205391 409.39
## + Tow_Bar  1    116896 12205753 409.39
## + KM       1    104218 12218431 409.42
## + Abag_2   1     99362 12223287 409.43
## + Radio    1     99362 12223287 409.43
## + CD       1     65683 12256966 409.52
## + AC       1     65339 12257310 409.52
## + Wght     1     14161 12308488 409.65
## + Green    1       179 12322470 409.68
##
## Step:  AIC=402.19
## Price ~ HP + Clock + Red + Silver
##
##           Df Sum of Sq     RSS    AIC
## + SpM      1   2368527 7308839 395.49
## + `G P`    1   1817239 7860128 397.74
## + Comp     1   1817239 7860128 397.74
## + Abag_2   1   1228468 8448898 399.98
## + Radio    1   1228468 8448898 399.98
## + Black    1    796333 8881033 401.53
## + `M Rim`  1    687314 8990052 401.91
## + Grs      1    612147 9065219 402.17
## <none>                 9677366 402.19
## + Mfr_G    1    555793 9121573 402.36
## + CC       1    440082 9237284 402.75
## + KM       1    410299 9267068 402.85
## + Blue     1    384533 9292833 402.93
## + Tow_Bar  1    132037 9545329 403.76
## + MC       1    113468 9563898 403.83
## + Auto     1    106174 9571192 403.85
## + Age      1     54743 9622623 404.01
## + AC       1     37462 9639904 404.07
## + Grey     1     23255 9654111 404.12
## + CD       1     22123 9655243 404.12
## + Wght     1     11407 9665959 404.15
## + Green    1       102 9677264 404.19
##
## Step:  AIC=395.49
## Price ~ HP + Clock + Red + Silver + SpM
##
##           Df Sum of Sq     RSS    AIC
## + `M Rim`  1    802295 6506544 393.88
## + Wght     1    542117 6766723 395.10
## + Comp     1    479196 6829643 395.39
## + `G P`    1    479196 6829643 395.39
## <none>                 7308839 395.49
## + Age      1    324057 6984782 396.08
## + Black    1    316146 6992693 396.12
## + Tow_Bar  1    246647 7062193 396.42
## + Abag_2   1    205550 7103290 396.60
## + Radio    1    205550 7103290 396.60
## + Auto     1    203097 7105742 396.62
```

```
## + KM       1    171029 7137810 396.75
## + MC       1    151898 7156941 396.84
## + Blue     1     76555 7232284 397.16
## + CC       1     71940 7236899 397.18
## + CD       1     29577 7279262 397.36
## + Grey     1     22875 7285964 397.39
## + Green    1     13958 7294881 397.43
## + Mfr_G    1     12197 7296642 397.44
## + Grs      1      2751 7306088 397.48
## + AC       1       863 7307976 397.49
##
## Step:  AIC=393.88
## Price ~ HP + Clock + Red + Silver + SpM + `M Rim`
##
##           Df Sum of Sq     RSS    AIC
## + Wght     1    757071 5749472 392.05
## <none>                 6506544 393.88
## + Blue     1    316647 6189897 394.34
## + `G P`    1    279147 6227396 394.52
## + Comp     1    279147 6227396 394.52
## + Age      1    181587 6324956 395.01
## + Mfr_G    1    165471 6341073 395.09
## + CC       1    110058 6396486 395.36
## + Black    1    108623 6397921 395.36
## + Auto     1    104229 6402315 395.38
## + Abag_2   1     97377 6409166 395.42
## + Radio    1     97377 6409166 395.42
## + KM       1     95711 6410832 395.42
## + AC       1     83742 6422802 395.48
## + Grey     1     73493 6433051 395.53
## + MC       1     26622 6479922 395.76
## + CD       1     12783 6493761 395.82
## + Tow_Bar  1     10227 6496316 395.84
## + Grs      1      2885 6503659 395.87
## + Green    1      1738 6504805 395.88
##
## Step:  AIC=392.05
## Price ~ HP + Clock + Red + Silver + SpM + `M Rim` + Wght
##
##           Df Sum of Sq     RSS    AIC
## + Auto     1    364462 5385010 392.02
## <none>                 5749472 392.05
## + Blue     1    304929 5444544 392.36
## + `G P`    1    193196 5556276 392.99
## + Comp     1    193196 5556276 392.99
## + CD       1    172921 5576552 393.10
## + KM       1    116638 5632834 393.41
## + Grs      1    115185 5634287 393.42
## + Grey     1     94871 5654601 393.53
## + Black    1     89980 5659493 393.56
## + Age      1     50529 5698943 393.78
## + Abag_2   1     45607 5703865 393.80
## + Radio    1     45607 5703865 393.80
## + Mfr_G    1     13979 5735493 393.97
```

```
## + MC       1     13025 5736448 393.98
## + CC       1     12320 5737152 393.98
## + Green    1      8356 5741116 394.00
## + AC       1       101 5749372 394.05
## + Tow_Bar  1        54 5749418 394.05
##
## Step:  AIC=392.02
## Price ~ HP + Clock + Red + Silver + SpM + `M Rim` + Wght + Auto
##
##           Df Sum of Sq     RSS    AIC
## <none>                 5385010 392.02
## + Grs      1    238905 5146106 392.61
## + Blue     1    210632 5174378 392.78
## + MC       1    159202 5225808 393.09
## + CD       1    139438 5245573 393.21
## + Black    1    126897 5258113 393.28
## + Comp     1    106535 5278475 393.40
## + `G P`    1    106535 5278475 393.40
## + KM       1     56758 5328253 393.69
## + Grey     1     26398 5358612 393.87
## + Age      1     20531 5364479 393.90
## + CC       1     19095 5365916 393.91
## + Abag_2   1     10843 5374167 393.96
## + Radio    1     10843 5374167 393.96
## + Green    1      4516 5380494 393.99
## + Mfr_G    1      3919 5381091 394.00
## + Tow_Bar  1       976 5384034 394.01
## + AC       1       871 5384139 394.01
```

```
summary(variable_sel)
```

```
##
## Call:
## lm(formula = Price ~ HP + Clock + Red + Silver + SpM + `M Rim` +
##     Wght + Auto, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -735.34 -262.83   61.58  219.83 1124.70
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -15156.58   11412.19  -1.328 0.197759
## HP              52.22      10.10   5.172 3.47e-05 ***
## Clock1        1417.65     345.33   4.105 0.000467 ***
## Red1         -2163.42     428.44  -5.050 4.67e-05 ***
## Silver1      -2356.07     486.97  -4.838 7.80e-05 ***
## SpM1         -1203.01     315.53  -3.813 0.000951 ***
## `M Rim`1      -667.32     283.60  -2.353 0.027975 *
## Wght            23.06      11.31   2.039 0.053680 .
## Auto1        -1028.18     842.61  -1.220 0.235294
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 494.7 on 22 degrees of freedom
## Multiple R-squared:  0.9723, Adjusted R-squared:  0.9622
## F-statistic: 96.44 on 8 and 22 DF,  p-value: 2.506e-15
```

```
#Therefore, we can conclude that factors such as HP, Clock, Red, Silver, SpM,
#MRim, Wght, Auto strongly influence a customers decision in buying a car.
#Following are the objectives of Farid's Model:
#1. He plans to build a robust model to accurately predict the prices of cars
#using the car features list.
#2. He plans to use and compare various ML models such as Linear Regression and
#compare the same with Neural-Network.
#3. He plans to identify important features that play a key role in predicting
#the prices of cars and deploy and use a single model after proper comparison
#for computing the prices of cars.
#4. This will help Adam Ebraham to provide the optimal MSRPs to the dealers, so
#that the dealers could decide their final price.
```

Q2) Construct a neural network model. Validate and interpret the model using a different number of hidden neurons.

```
#Normalizing the dataset
scale <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}
cars_normalize <- cars %>% mutate_if(is.numeric, scale)
cars_normalize
```

```
## # A tibble: 31 x 27
##    Price   Age    KM    HP MC    Auto     CC Grs    Wght `G P` Mfr_G Abag_2
##    <dbl> <dbl> <dbl> <dbl> <fct> <fct> <dbl> <fct> <dbl> <dbl> <fct> <fct>
##  1 0.816   0.3 0.375 1     0     0       1   6     1     0.375 1     1
##  2 0.714   0   0.585 1     0     0       1   6     1     0     1     1
##  3 0.679   0.3 0.387 1     0     0       1   6     1     0     1     1
##  4 0.872   0.9 0.229 1     1     0       1   6     1     0     1     1
##  5 0.974   1   0.421 1     1     0       1   6     1     0     1     1
##  6 0.923   0.6 0.155 1     0     0       1   6     1     0     1     1
##  7 1       0.8 0.419 1     1     0       1   5     1     0     1     1
##  8 0.510   0.2 0.207 0.137 1     0       0.5 5     0.333 1     0     0
##  9 0.388   0.2 0.273 0.137 0     0       0.5 5     0     0     0     1
## 10 0.408   0.8 0.943 0.137 1     0       0.5 5     0.333 0     1     1
## # ... with 21 more rows, and 15 more variables: AC <fct>, Comp <fct>, CD <fct>,
## #   Clock <fct>, Pwi <fct>, Radio <fct>, SpM <fct>, `M Rim` <fct>,
## #   Tow_Bar <fct>, Silver <fct>, Red <fct>, Black <fct>, Grey <fct>,
## #   Blue <fct>, Green <fct>
```

```
str(cars_normalize)
```

```
## tibble [31 x 27] (S3: tbl_df/tbl/data.frame)
##  $ Price  : num [1:31] 0.816 0.714 0.679 0.872 0.974 ...
##  $ Age    : num [1:31] 0.3 0 0.3 0.9 1 0.6 0.8 0.2 0.2 0.8 ...
```

```
##  $ KM     : num [1:31] 0.375 0.585 0.387 0.229 0.421 ...
##  $ HP     : num [1:31] 1 1 1 1 1 ...
##  $ MC     : Factor w/ 2 levels "0","1": 1 1 1 2 2 1 2 2 1 2 ...
##  $ Auto   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ CC     : num [1:31] 1 1 1 1 1 1 1 0.5 0.5 0.5 ...
##  $ Grs    : Factor w/ 2 levels "5","6": 2 2 2 2 2 2 1 1 1 1 ...
##  $ Wght   : num [1:31] 1 1 1 1 1 ...
##  $ G P    : num [1:31] 0.375 0 0 0 0 0 0 1 0 0 ...
##  $ Mfr_G  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 1 2 ...
##  $ Abag_2 : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 2 2 ...
##  $ AC     : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 2 1 ...
##  $ Comp   : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 2 1 2 2 ...
##  $ CD     : Factor w/ 2 levels "0","1": 2 1 1 2 2 1 2 1 2 2 ...
##  $ Clock  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Pwi    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Radio  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ SpM    : Factor w/ 2 levels "0","1": 1 2 2 2 2 2 1 1 1 2 ...
##  $ M Rim  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 1 1 ...
##  $ Tow_Bar: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ Silver : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
##  $ Red    : Factor w/ 2 levels "0","1": 1 2 2 1 1 1 1 1 1 1 ...
##  $ Black  : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
##  $ Grey   : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 2 1 2 2 ...
##  $ Blue   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 1 ...
##  $ Green  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

```r
#Splitting the data into training and testing.
set.seed(1234)
ind <- sample(2, nrow(cars_normalize), replace = T, prob = c(0.6, 0.4))
train <- cars_normalize[ind == 1, ]
test <- cars_normalize[ind == 2, ]

#Building the neural network model.
dec <- seq(0.0001,1,length.out=20)
i <- 1
table_col_names <- c("Decay", "Size", "Error_Percentage")
table_matrix <- data.frame(matrix(nrow=1, ncol = length(table_col_names)))
colnames(table_matrix) <- table_col_names
table_matrix <- na.omit(table_matrix)
for (j in dec)
{
  x <- 1
  while (x <= 20)
  {
      nn_cars_model <- nnet(Price ~ HP + Clock + Red + Silver + SpM + `M Rim` +
                              Wght + Auto, data = train, linout = TRUE,
                            size = x, decay = j, maxit = 1000, trace = FALSE)
      nn_cars_pred <- predict(nn_cars_model, test)
      error <- rmse(test$Price, nn_cars_pred)
      error_percent <- error/mean(test$Price)
      table_matrix[nrow(table_matrix) + 1,] <- c(j, x, error_percent)
      x <- x + 1
      i <- i + 1
```

```
  }
}

#In the above code, we have created a table called "table_matrix" of different
#neural network models with different decay and size parameters along with their
#respective Error%.
#From the "table_matrix", we would be selecting that model, that has the lowest
#Error% as that would be giving us the best performance.

head(table_matrix)
```

```
##   Decay Size Error_Percentage
## 1 1e-04    1        0.2148340
## 2 1e-04    2        0.3029105
## 3 1e-04    3        0.2170246
## 4 1e-04    4        0.1721483
## 5 1e-04    5        0.1711431
## 6 1e-04    6        0.1721076
```

```
minimum_err <- min(table_matrix$Error_Percentage)
best_nn_index <- which(table_matrix$Error_Percentage == minimum_err)
best_nn_model <- table_matrix[best_nn_index,]
print(best_nn_model)
```

```
##         Decay Size Error_Percentage
## 40 0.05272632   20        0.1707631
```

```
#Thus, from the above code, we got that when we set the decay to 0.05272 with a
#size of 20, we get the best neural network model having Error% = 17.08%
```

Q3)Compare your neural network models with linear regression model. Which one is better?

```
#Building the Linear Regression Model
#Dividing the data into training and testing
set.seed(123)
ind_reg <- sample(2, nrow(cars), replace = T, prob = c(0.6, 0.4))
train_reg <- cars[ind_reg == 1, ]
test_reg <- cars[ind_reg == 2, ]

linear_model <- lm(formula = Price ~ HP + Clock + Red + Silver + SpM + `M Rim` +
                     Wght + Auto, data = train_reg)
linear_model_pred <- predict(linear_model, test_reg)
error_reg <- rmse(test_reg$Price, linear_model_pred)
error_percent_reg <- error_reg/mean(test_reg$Price)
print(error_percent_reg)
```

```
## [1] 0.03528415
```

```
#Therefore, the Error% for the Linear Regression Model is 3.53%
#So on comparing the best Neural Network Model along with the Linear Regression
#Model, we find that the Linear Regression Model is way better than the Neural
#Network Model as the Error% in case of Linear Regression Model is significantly
#lower than the Error% in the best Neural Network Model.
```

Q4)Make a decision and offer your recommendations.

```
#Therefore, as concluded previously, the Linear model is better than the Neural
#Network Model because of
#lower Error%. Therefore, we got our important variable as HP, Clock, Red,
#Silver, SpM, MRim, Wght, Auto.
#On training the Linear Regression Model using these variables, we got the
#following co-efficients.

summary(linear_model)
```

```
##
## Call:
## lm(formula = Price ~ HP + Clock + Red + Silver + SpM + `M Rim` +
##     Wght + Auto, data = train_reg)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -695.7 -119.8    0.0  193.3  490.1
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11378.09   12380.58  -0.919  0.38865
## HP              55.03      11.17   4.928  0.00170 **
## Clock1         895.67     481.73   1.859  0.10532
## Red1         -2215.53     593.80  -3.731  0.00735 **
## Silver1      -2029.97     588.39  -3.450  0.01069 *
## SpM1         -1164.44     432.60  -2.692  0.03101 *
## `M Rim`1      -739.13     464.74  -1.590  0.15576
## Wght            19.78      12.33   1.604  0.15271
## Auto1         -795.79     877.62  -0.907  0.39467
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 450.8 on 7 degrees of freedom
## Multiple R-squared:  0.9842, Adjusted R-squared:  0.9661
## F-statistic:  54.5 on 8 and 7 DF,  p-value: 1.282e-05
```

```
#Intercept: -11378.09
#HP: 55.03
#Clock1: 895.67
#Red1: -2215.53
#Silver1: -2029.97
#SpM1: -1164.44
#MRim1: -739.13
```

```
#Wght: 19.78
#Auto1: -795.79

#Therefore, our Linear Regression Model is:
#Price = -11378.09 + 55.03HP + 895.67Clock1 - 2215.53Red1 - 2029.97Silver1 -
#1164.44SpM1 - 739.13MRim1 + 19.78Wght - 795.79Auto1

#Training the Neural Network Model by taking all the variables as predictors.
set.seed(1234)
ind <- sample(2, nrow(cars_normalize), replace = T, prob = c(0.6, 0.4))
train <- cars_normalize[ind == 1, ]
test <- cars_normalize[ind == 2, ]
dec <- seq(0.0001,1,length.out=20)
i <- 1
table_col_names <- c("Decay", "Size", "Error_Percentage")
table_matrix <- data.frame(matrix(nrow=1, ncol = length(table_col_names)))
colnames(table_matrix) <- table_col_names
table_matrix <- na.omit(table_matrix)
for (j in dec)
{
  x <- 1
  while (x <= 20)
  {
    nn_cars_model <- nnet(Price ~ ., data = train, linout = TRUE,
                          size = x, decay = j, maxit = 1000, trace = FALSE)
    nn_cars_pred <- predict(nn_cars_model, test)
    error <- rmse(test$Price, nn_cars_pred)
    error_percent <- error/mean(test$Price)
    table_matrix[nrow(table_matrix) + 1,] <- c(j, x, error_percent)
    x <- x + 1
    i <- i + 1

  }
}

#In the above code, we have created a table called "table_matrix" of different
#neural network models with different decay and size parameters along with their
#respective Error%.
#From the "table_matrix", we would be selecting that model, that has the lowest
#Error% as that would be giving us the best performance.

head(table_matrix)
```

```
##   Decay Size Error_Percentage
## 1 1e-04    1        0.3264059
## 2 1e-04    2        0.2720755
## 3 1e-04    3        0.2773636
## 4 1e-04    4        0.2758099
## 5 1e-04    5        0.2760842
## 6 1e-04    6        0.2793845
```

```
minimum_err <- min(table_matrix$Error_Percentage)
best_nn_index <- which(table_matrix$Error_Percentage == minimum_err)
```

```r
best_nn_model <- table_matrix[best_nn_index,]
print(best_nn_model)
```

```
##         Decay Size Error_Percentage
## 40 0.05272632   20        0.1866575
```

```r
#Essentially, as we can see that training the Neural Network Model again using
#all variables as predictors, we get the least Error%
# of 18.67% for the same Decay and Size parameter as the previous one. However,
#we would recommend using the previous Neural Network Model
#as compared to this one because the previous one had lesser Error%.
```