Homework #1: Regression task

The problem about this project is making a price prediction about second hand price with second hand cars dataset. In the dataset we have 12 features and our target is price.

```
second <- read.csv("second.csv")
```

DATASET

```
str(second)
```

```
'data.frame':   1000 obs. of  12 variables:
 $ v.id         : int  1 2 3 4 5 6 7 8 9 10 ...
 $ on.road.old  : int  535651 591911 686990 573999 691388 650007 633344 662990 543184 573043
 $ on.road.now  : int  798186 861056 770762 722381 811335 844846 756063 891569 841354 879481
 $ years        : int  3 6 2 4 6 6 5 6 7 2 ...
 $ km           : int  78945 117220 132538 101065 61559 148846 78025 76546 57662 132347 ...
 $ rating       : int  1 5 2 4 3 2 1 1 4 2 ...
 $ condition    : int  2 9 8 3 9 9 9 2 7 3 ...
 $ economy      : int  14 9 15 11 12 13 15 12 14 12 ...
 $ top.speed    : int  177 148 181 197 160 138 171 146 151 200 ...
 $ hp           : int  73 74 53 54 53 61 94 109 50 115 ...
 $ torque       : int  123 95 97 116 105 109 132 96 132 82 ...
 $ current.price: num  351318 285002 215386 244296 531114 ...
```

In the dataset we have 12 features these are and 1000 observation. All features are string and all observations are numeric.

Splitting The Dataset

```
set.seed(1)
index <- sample(1 : nrow(second), round(nrow(second) * 0.80))
train <- second[index, ]
```

```r
test  <- second[-index, ]
```

Train a Linear Regression Model

```r
lrm_model <- lm(`current.price` ~ ., data = train)
```

```r
lrm_model
```

```

Call:
lm(formula = current.price ~ ., data = train)

Coefficients:
(Intercept)         v.id  on.road.old  on.road.now        years           km
 -1.563e+04    1.176e+00    5.057e-01    5.002e-01   -1.574e+03   -3.992e+00
     rating    condition      economy    top.speed           hp       torque
  1.352e+02    4.532e+03    5.199e+01   -1.339e+01    1.452e+01    1.602e+01
```

```r
summary(lrm_model)
```

```

Call:
lm(formula = current.price ~ ., data = train)

Residuals:
   Min     1Q Median     3Q    Max
-13012  -7373  -1668   5201  21714

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)
(Intercept) -1.563e+04  6.740e+03   -2.318   0.0207 *
v.id         1.176e+00  1.068e+00    1.101   0.2713
on.road.old  5.057e-01  5.230e-03   96.701   <2e-16 ***
on.road.now  5.002e-01  5.413e-03   92.404   <2e-16 ***
years       -1.574e+03  1.788e+02   -8.803   <2e-16 ***
km          -3.992e+00  1.061e-02 -376.295   <2e-16 ***
rating       1.352e+02  2.187e+02    0.618   0.5367
condition    4.532e+03  1.088e+02   41.663   <2e-16 ***
economy      5.199e+01  1.387e+02    0.375   0.7080
```

```
top.speed   -1.339e+01  1.603e+01  -0.835   0.4039
hp           1.452e+01  1.508e+01   0.962   0.3362
torque       1.602e+01  1.476e+01   1.086   0.2780
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8658 on 788 degrees of freedom
Multiple R-squared:  0.9953,    Adjusted R-squared:  0.9953
F-statistic: 1.53e+04 on 11 and 788 DF,  p-value: < 2.2e-16
```

Measuring Model Performance

```r
predicted_y <- predict(lrm_model, test[,-12])
head(predicted_y)
```

```
      18       23       26       32       38       46
281326.3 332443.6 378467.3 463773.5 428744.2 369894.2
```

```r
error <- test$`current.price` - predicted_y
head(error)
```

```
       18         23         26         32         38         46
-6968.814  4883.420  3379.170 -3858.993 -6963.156 20031.278
```

```r
mse_model  <- mean(error ^ 2)
rmse_model <- sqrt(mean(error ^ 2))
mae_model  <- mean(abs(error))
```

```r
mse_model
```

```
[1] 83026359
```

```r
rmse_model
```

```
[1] 9111.88
```

```
mae_model
```

```
[1] 7493.416
```

To measure the performance of regression model, I will use MSE, RMSE, and MAE metrics. The lower value of MAE, MSE, and RMSE implies higher accuracy of a regression model. For the model performance we have to choose lowest error. Because the errors represent how much the model is making mistakes in its prediction.

Checking Overfitting and Underfitting Problem

```
rmse_train <- sqrt(mean((lrm_model$residuals) ^ 2))
rmse_test  <- rmse_model
```

```
rmse_train - rmse_test
```

```
[1] -518.615
```

Here RMSE train set less than RMSE test set. This means model learns more from the train set. This may be sign of overfitting problem.

Adding New Observations

```
new_row <- c(1001, 397631, 550289, 3, 12651, 3, 8, 14, 250, 150, 230, 659625.0)
newdata <- rbind(second,new_row)
```

```
str(newdata)
```

```
'data.frame':   1001 obs. of  12 variables:
 $ v.id         : num  1 2 3 4 5 6 7 8 9 10 ...
 $ on.road.old  : num  535651 591911 686990 573999 691388 ...
 $ on.road.now  : num  798186 861056 770762 722381 811335 ...
 $ years        : num  3 6 2 4 6 6 5 6 7 2 ...
 $ km           : num  78945 117220 132538 101065 61559 ...
 $ rating       : num  1 5 2 4 3 2 1 1 4 2 ...
 $ condition    : num  2 9 8 3 9 9 9 2 7 3 ...
 $ economy      : num  14 9 15 11 12 13 15 12 14 12 ...
 $ top.speed    : num  177 148 181 197 160 138 171 146 151 200 ...
 $ hp           : num  73 74 53 54 53 61 94 109 50 115 ...
 $ torque       : num  123 95 97 116 105 109 132 96 132 82 ...
 $ current.price: num  351318 285002 215386 244296 531114 ...
```

Now we have 1001 observation.

```r
nrow(newdata)
```

```
[1] 1001
```

```r
set.seed(2)
indexx <- sample(1 : nrow(newdata), round(nrow(newdata) * 0.80))
trainn <- newdata[indexx, ]
testt  <- newdata[-indexx, ]
lrm_modell <- lm(`current.price` ~ ., data = trainn)
```

```r
lrm_modell <- lm(`current.price` ~ ., data = trainn)
```

```r
lrm_modell
```

```

Call:
lm(formula = current.price ~ ., data = trainn)

Coefficients:
(Intercept)         v.id  on.road.old  on.road.now        years           km
 -1.192e+04    1.132e+00    4.963e-01    4.810e-01   -1.811e+03   -4.021e+00
     rating    condition      economy    top.speed           hp       torque
  2.231e+02    4.733e+03    2.884e+02    3.746e+01    5.586e+01    6.981e+01
```

```r
summary(lrm_modell)
```

```

Call:
lm(formula = current.price ~ ., data = trainn)

Residuals:
   Min     1Q Median     3Q    Max
```

```
-17988  -7772  -1858   5702 188308

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)    
(Intercept) -1.192e+04  8.808e+03   -1.353 0.176409    
v.id         1.132e+00  1.388e+00    0.816 0.414766    
on.road.old  4.963e-01  6.863e-03   72.319  < 2e-16 ***
on.road.now  4.810e-01  7.045e-03   68.280  < 2e-16 ***
years       -1.811e+03  2.314e+02   -7.826 1.61e-14 ***
km          -4.021e+00  1.370e-02 -293.570  < 2e-16 ***
rating       2.231e+02  2.873e+02    0.777 0.437606    
condition    4.733e+03  1.425e+02   33.216  < 2e-16 ***
economy      2.884e+02  1.804e+02    1.599 0.110201    
top.speed    3.746e+01  2.064e+01    1.815 0.069929 .  
hp           5.586e+01  1.966e+01    2.841 0.004614 ** 
torque       6.981e+01  1.852e+01    3.769 0.000176 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11290 on 789 degrees of freedom
Multiple R-squared:  0.9922,	Adjusted R-squared:  0.9921 
F-statistic:  9142 on 11 and 789 DF,  p-value: < 2.2e-16
```

Measuring New Model Performance

```
predicted_yy <- predict(lrm_modell, testt[,-12])
head(predicted_yy)
```

```
       6        7       15       18       19       24 
170680.4 411316.9 462410.8 280744.1 427878.0 129963.2 
```

```
errorr <- testt$`current.price` - predicted_yy
head(errorr)
```

```
         6          7         15         18         19         24 
 7253.1432  -439.8533 11870.6914 -6386.5617 11761.4571 10253.3321 
```

```
mse_modell  <- mean(errorr ^ 2)
rmse_modell <- sqrt(mean(errorr ^ 2))
```

```r
mae_modell  <- mean(abs(errorr))
```

```r
mse_modell
```

```
[1] 82814110
```

```r
rmse_modell
```

```
[1] 9100.226
```

```r
mae_modell
```

```
[1] 7662.841
```