Regression Task: Predicting Second Hand Car Prices

Buğracan Tanrıverdi

1. Task

The task we have given is to predict the price of second hand cars. To do this, we will use this data set Kaggle. Since price is a continuous variable we will use regression for this task.

2. Description of the Data Set

First of all, we must import the data set.

```
cars <- read.csv("cars.csv")
cars <- cars[,2:12]
head(cars)
```

```
  on.road.old on.road.now years     km rating condition economy top.speed hp
1      535651      798186     3  78945      1         2      14       177 73
2      591911      861056     6 117220      5         9       9       148 74
3      686990      770762     2 132538      2         8      15       181 53
4      573999      722381     4 101065      4         3      11       197 54
5      691388      811335     6  61559      3         9      12       160 53
6      650007      844846     6 148846      2         9      13       138 61
  torque current.price
1    123      351318.0
2     95      285001.5
3     97      215386.0
4    116      244295.5
5    105      531114.5
6    109      177933.5
```

The data set consists of 12 columns (one of them is just ids, so we deleted that column) and 1000 rows. Since `current.price` is our *target* this means we have 10 *features*. Also all of our *features* seems like they are all `numeric`.

3. Model Training

3.1 Splitting the Data Set

Before training a regression model we should split the data set into two parts; `train` and `test`.

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

3.2 Training

We will train the model on the `train` subset of our data set using the `lm` function.

```
lm.model <- lm(current.price ~ ., data = train)
summary(lm.model)
```

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

Residuals:
   Min     1Q Median     3Q    Max 
-12415  -7382  -1864   5132  21921 

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)    
(Intercept) -1.507e+04  6.722e+03   -2.242   0.0252 *  
on.road.old  5.057e-01  5.231e-03   96.685   <2e-16 ***
on.road.now  5.000e-01  5.411e-03   92.401   <2e-16 ***
years       -1.584e+03  1.787e+02   -8.865   <2e-16 ***
km          -3.991e+00  1.058e-02 -377.268   <2e-16 ***
rating       1.385e+02  2.187e+02    0.633   0.5268    
condition    4.531e+03  1.088e+02   41.646   <2e-16 ***
economy      6.017e+01  1.385e+02    0.434   0.6642    
```

```
top.speed   -1.323e+01  1.603e+01   -0.825   0.4096
hp           1.478e+01  1.508e+01    0.980   0.3275
torque       1.611e+01  1.476e+01    1.091   0.2755
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8660 on 789 degrees of freedom
Multiple R-squared:  0.9953,    Adjusted R-squared:  0.9953
F-statistic: 1.683e+04 on 10 and 789 DF,  p-value: < 2.2e-16
```

It seems that the only significant features are `on.road.old`, `on.road.new`, `years`, `km`, `condition`.

3.3 Prediction

We can easily calculate the predicted prices using the `predict` function.

```
predicted.price <- predict(lm.model, test)
head(predicted.price)
```

```
      18       23       26       32       38       46
281916.2 333030.9 379000.9 464309.6 429237.8 370423.1
```

4. Performance

In order to measure the performance of the regression model that we have trained we will use the *root mean squared error (RMSE)* metric, since it is easy to calculate and gives the error in the same unit as our *target*.

```
error <- test$current.price - predicted.price
rmse.model <- sqrt(mean(error ^ 2))
rmse.model
```

```
[1] 9091.345
```

Intuitively this means that on average our predictions differ 9000 from the actual price of the cars. Since the average price of a car in our data set is 308520.2 this error looks acceptable.

5. Overfitting

We must also check if there is an overfitting problem.

```
rmse.test <- sqrt(mean(lm.model$residuals^2))
rmse.model - rmse.test
```

```
[1] 491.4751
```

Since the difference between the model and test RMSE is a large positive number, most of the learning comes from the model. We can conclude that we don't have any overfitting problems.

6. Predicting New Observations

We can also create our own observations to predict. We will create 5 new observations.

```
new.observations
```

```
  on.road.now on.road.old years     km rating condition economy top.speed  hp
1      700000      550000     4  90000      1         8       7       150  70
2      750000      400000     6  60000      3         7      10       170  80
3      750000      500000     8 100000      5         6      11       140 100
4      800000      650000     7  70000      5         8      12       150  60
5      900000      600000     3  80000      4         5       9       180  50
  torque
1     90
2     80
3     80
4    110
5    100
```

Now we will predict the price of each of these cars using our model.

```
predicted.price.new <- predict(lm.model, new.observations)
predicted.price.new
```

```
       1        2        3        4        5 
284864.2 346218.1 230482.2 461534.8 438060.3 
```