Performance Comparison of Models Trained with Decision Tree and Linear Regression Methods

Merve Topdemir

22 04 2021

CONTENTS

Introduction

In this work, regression model performance and decision trees performance will be compared using R. Data to be used here is "dragons" from “DALEX” package. The data has 8 variables and 2000 observations.

```
#install.packages ("DALEX")
library (DALEX)
data<-dragons
head(data,10)
##    year_of_birth  height     weight     scars  colour  year_of_discovery
## 1        -1291    59.40365   15.32391     7      red         1700
## 2         1589    46.21374   11.80819     5      red         1700
## 3         1528    49.17233   13.34482     6      red         1700
## 4         1645    48.29177   13.27427     5    green         1700
## 5           -8    49.99679   13.08757     1      red         1700
## 6          915    45.40876   11.48717     2      red         1700
## 7        -1557    50.73455   14.51622     3    black         1700
## 8        -1144    45.45112   11.93573     7     blue         1700
## 9          -66    65.72122   17.75933     6      red         1700
## 10        -165    54.20568   14.29586    32      red         1700
##    number_of_lost_teeth    life_length
## 1           25              1368.4331
## 2           28              1377.0474
## 3           38              1603.9632
## 4           33              1434.4222
## 5           18               985.4905
## 6           20               969.5682
## 7           28              1255.0887
## 8           29              1498.9457
## 9            2               886.8254
## 10          22              2298.9354
```

Data Structure

Dependent (target) variable:

- **life_length:** life length of the dragon.

Independent (feature) variables:

- **scars:** number of scars.
- **colour:** colour of the dragon.
- **height:** height of the dragon in yards.
- **weight:** weight of the dragon in tons.
- **year_of_birth:** year in which the dragon was born.
  (Negative year means year BC, eg: -1200 = 1201 BC)
- **year_of_discovery:** year in which the dragon was found.
- **number_of_lost_teeth:** number of teeth that the dragon lost.

```
str(data)

## 'data.frame':    2000 obs. of  8 variables:
##  $ year_of_birth       : num  -1291 1589 1528 1645 -8 ...
##  $ height              : num  59.4 46.2 49.2 48.3 50 ...
##  $ weight              : num  15.3 11.8 13.3 13.3 13.1 ...
##  $ scars               : num  7 5 6 5 1 2 3 7 6 32 ...
##  $ colour              : Factor w/ 4 levels "black","blue",..: 4 4 4 3 4 4 1 2 4 4
...
##  $ year_of_discovery   : num  1700 1700 1700 1700 1700 1700 1700 1700 1700 1700
...
##  $ number_of_lost_teeth: num  25 28 38 33 18 20 28 29 2 22 ...
##  $ life_length         : num  1368 1377 1604 1434 985 ...
```

```
summary(data)
```

```
##   year_of_birth        height           weight            scars      
##  Min.   :-1999.00   Min.   :29.94   Min.   : 7.524   Min.   : 0.00  
##  1st Qu.:-1017.25   1st Qu.:44.98   1st Qu.:12.043   1st Qu.: 3.00  
##  Median :  -66.00   Median :49.68   Median :13.385   Median : 7.00  
##  Mean   :  -79.54   Mean   :50.05   Mean   :13.493   Mean   : 9.94  
##  3rd Qu.:  858.00   3rd Qu.:54.50   3rd Qu.:14.826   3rd Qu.:14.00  
##  Max.   : 1800.00   Max.   :76.28   Max.   :22.372   Max.   :76.00  
##     colour        year_of_discovery number_of_lost_teeth  life_length    
##    black:  27       Min.   :1700       Min.   : 0           Min.   : 511.2  
##    blue : 576       1st Qu.:1724       1st Qu.:10           1st Qu.:1034.9  
##    green: 370       Median :1751       Median :20           Median :1314.8  
##    red  :1027       Mean   :1750       Mean   :20           Mean   :1371.0  
##                     3rd Qu.:1776       3rd Qu.:30           3rd Qu.:1614.4  
##                     Max.   :1800       Max.   :40           Max.   :3952.7
```

Factor variables are used to represent categorical data. Numerical variable is one that may take on any value within a finite or infinite interval.

- **colour** is a factor variable, it has 4 levels, 27 black, 576 blue , 370 green and 1027 red.
- **year_of_birth**, **height**, **weight**, **scars**, **year_of_discovery**, **number_of_lost_teeth** and **life_length** numerical variables.
- **Number_of_lost_teeth** variable minimum value is 0, maximum value is 40, median is 20, mean is 20, first quartile is 10 and third quartile is 30. **Scars** minimum value is 0, maximum value is 76, median is 7, mean is 9.94, first quartile is 3 and third quartile is 14.
- **All other variables** and their descriptive statistics are given in the output above.

LINEAR REGRESSION MODEL

Splitting of Data

The following code splits 80% of the data selected randomly into training set and the remaining 20% sample into test set.

```
set.seed(123)
index <- sample(nrow(data), nrow(data) * 0.8)
train <- data[index,]
test <- data[-index,]
nrow(train)
## [1] 1600
nrow(test)
## [1] 400
```

Training a LRM

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

##
## Call:
## lm(formula = life_length ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -107.775  -30.385    0.643   28.427  113.825
##
## Coefficients:
##                        Estimate Std. Error t value  Pr(>|t|)
## (Intercept)           6.422e+02  6.151e+01  10.440   <2e-16 ***
## year_of_birth        -2.360e-02  9.293e-04 -25.392   <2e-16 ***
## height               -4.455e-01  3.040e-01  -1.466   0.1429
## weight                1.913e+00  1.064e+00   1.798   0.0724 .
## scars                 4.004e+01  1.014e-01 395.017   <2e-16 ***
## colourblue           -1.134e+01  8.652e+00  -1.310   0.1903
## colourgreen          -1.153e+01  8.781e+00  -1.313   0.1895
## colourred            -1.087e+01  8.559e+00  -1.270   0.2041
## year_of_discovery    -3.816e-02  3.448e-02  -1.107   0.2685
## number_of_lost_teeth  2.018e+01  8.769e-02 230.140   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40.42 on 1590 degrees of freedom
## Multiple R-squared:  0.9926, Adjusted R-squared:  0.9926
## F-statistic: 2.382e+04 on 9 and 1590 DF,  p-value: < 2.2e-16
```

- In this result, intercept, year_of_birth, scars and number_of_lost_teeth variables are significant at 0.001 significance level. Weight variable is significant at 0.1 significance level but rest of the variables are not significant.
- P value is $2.2e^{-16}$ for significant of model. The model is significant because the p value is less than 0.05.
- R-squared and Adj. R-Squared values 0.99 which means our independent variables highly successful at prediction of dependent variables.

Performance of the Model on Train and Test Set

```
#Performance for test

predicted_test <- predict(model, test)
rmse_test <- sqrt(mean((predicted_test - test$life_length) ^ 2))
#Performance for train
predicted_train <- predict(model, train)
rmse_train <-sqrt(mean((predicted_train - train$life_length) ^ 2))


cat("test_rmse:", rmse_test,"\n")

## test_rmse: 42.52344

cat("train_rmse:", rmse_train)

## train_rmse: 40.2908
```

- Overfitting may have occurred but it is not certain because test RMSE value (42.52) greater than train RMSE value (40.29).
- There are several manners in which we can reduce overfitting in machine learning models. The best option is to get more training data. Other ways to reduce overfitting are using cross-validation, removing some features and regularization.

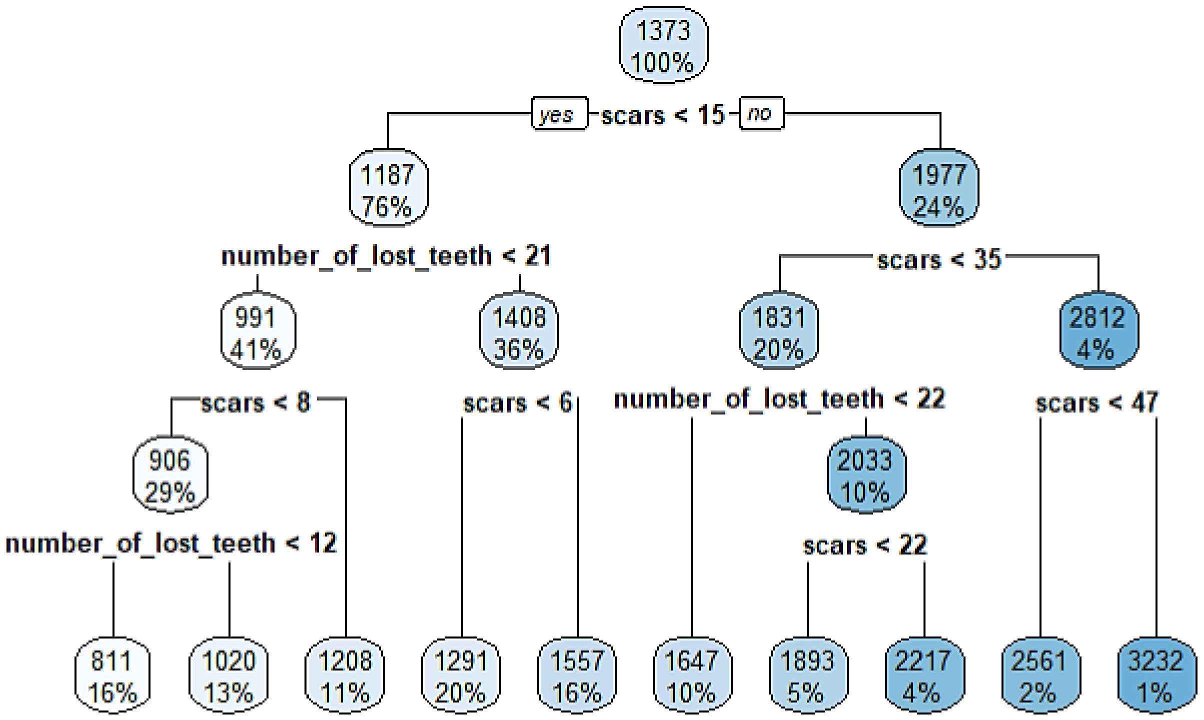
DECISION TREE (REGRESSION)

Splitting of Data

```
#install.packages("caret")
library(caret)
set.seed(123)
index <- createDataPartition(data$life_length , p = 0.8, list = FALSE, times = 1)
train <- data[index,]
test <- data[-index,]
dim(train)

## [1] 1600    8

dim(test)

## [1] 400   8
```

Training a Regression Tree

```r
# Training a regression tree on the dragons data
library(rpart)

library(rpart.plot)

model2 <- rpart(life_length  ~. , method = "anova", data = train)
rpart.plot(model2)
```



Percentages are indicated on the last line of each box. These percentages indicate what percentage of all data is in that node. In the first division, the percentages are 76% for yes (scars less than 15) and 24% for no (scars more than 15).For root node, this percentage is 100 because the data did not splitted yet. Also, the life length of dragons is given as 1373 at the root node. If the scars of dragons are less than 15, the branches go to the left (yes), if not to the right (no). Let's consider that the branches go to the left, that is, the scars of dragon are less than 15. This box has less data than the previous box (root node), and the life length of dragons is 1187. If we consider that the number of lost teeth by dragons is more than 21, the branch goes to the right. The life length of dragons in this node (interior node) is 1408. Finally, if the scars of the dragons are less than 6, the branch goes to the left. The life length of dragons in this node (leaf node) is 1291. In total, 20% of the data is located on this node. As a result, if the dragon's scars are less than 15 and the number of lost teeth less than 6, the model estimates its life length at 1291. Other nodes can be interpreted similarly.

Performance of the Trained Tree on the Train and Test Set

```
#Performance for test
predicted_test_dt <- predict(model2, test)
rmse_test_dt <- sqrt(mean((predicted_test_dt - test$life_length) ^ 2))
#Performance for train
predicted_train_dt <- predict(model2, train)
rmse_train_dt <- sqrt(mean((predicted_train_dt - train$life_length) ^ 2))


cat("test_rmse_dt:", rmse_test_dt,"\n")

## test_rmse_dt: 163.889

cat("train_rmse_dt:", rmse_train_dt)

## train_rmse_dt: 151.6945
```

- Overfitting may have occurred but it is not certain because test RMSE value (163.889) greater than train RMSE value (151.694).

- There are several manners in which we can reduce overfitting in machine learning models. The best option is to get more training data. Other ways to reduce overfitting are using cross-validation, removing some features and regularization.

Comparison

```
df<- data.frame("RMSE"=c(rmse_train,rmse_test,rmse_train_dt,rmse_test_dt),
row.names=c("Regression Model(Train)","Regression Model(Test)","Decision
Tree(Train)","Decision Tree(Test)"))
df<-round(df,2)
df

##                             RMSE
## Regression Model (Train)   40.29
## Regression Model (Test)    42.52
## Decision Tree (Train)     151.69
## Decision Tree (Test)      163.89
```

The results show that regression model outperforms the decision tree model on trains and test sets when reference to root mean square error. Because the lower the error, the higher the prediction performance of the model.