Performance comparison of models trained with Decision Trees and Linear Regression methods

Machine Learning and Applications

Tufan Bostan

24 04 2021

In this work, linear regression model performance and decision trees performance will be compared using R. Data to be used here is "Salary" from “carData” package. The wine dataset contains the results of a chemical analysis of wines grown in a specific area of Italy. The 2008-09 nine-month academic salary for Assistant Professors, Associate Professors and Professors in a college in the U.S. The data were collected as part of the on-going effort of the college's administration to monitor salary differences between male and female faculty members. The data contains no missing values. First, some descriptive statistics will be obtained and interpreted for the data. Then, the data will be split into two parts which are “Train” and “Test”. Then using the “Train” part of data models will be generated using both methods. Finally, performance of the models on train and test sets will be interpreted and Decision Trees and Linear Regression methods performances will be compared for this data.

Data Structure

First of all, “catData” package installed to access “Salary” data (`install.packages("catData")`). Now “Salary” data introduced to R. The “Salary” data assigned to “data”. All variables were correctly classified. First 6 observations of the data are listed below. Then, structure of the data displayed. (`str(data)`) After that, data summary displayed to gather more information about variables. (`summary(data)`)

```
data <- carData::Salaries
head(data)

##        rank discipline yrs.since.phd yrs.service  sex salary
## 1      Prof          B            19          18 Male 139750
## 2      Prof          B            20          16 Male 173200
## 3  AsstProf          B             4           3 Male  79750
## 4      Prof          B            45          39 Male 115000
## 5      Prof          B            40          41 Male 141500
## 6 AssocProf          B             6           6 Male  97000

str(data)

## 'data.frame':    397 obs. of  6 variables:
##  $ rank         : Factor w/ 3 levels "AsstProf","AssocProf",..: 3 3 1 3 3 2 3 3 3 3 ...
##  $ discipline   : Factor w/ 2 levels "A","B": 2 2 2 2 2 2 2 2 2 2 ...
##  $ yrs.since.phd: int  19 20 4 45 40 6 30 45 21 18 ...
##  $ yrs.service  : int  18 16 3 39 41 6 23 45 20 18 ...
##  $ sex          : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 1 ...
##  $ salary       : int  139750 173200 79750 115000 141500 97000 175000 147765 119250 1290
00 ...

summary(data)

##         rank     discipline yrs.since.phd    yrs.service        sex
##  AsstProf : 67   A:181      Min.   : 1.00   Min.   : 0.00   Female: 39
```

```
##  AssocProf: 64   B:216      1st Qu.:12.00   1st Qu.: 7.00   Male  :358
##  Prof     :266             Median :21.00   Median :16.00
##                            Mean   :22.31   Mean   :17.61
##                            3rd Qu.:32.00   3rd Qu.:27.00
##                            Max.   :56.00   Max.   :60.00
##      salary
##  Min.   : 57800
##  1st Qu.: 91000
##  Median :107300
##  Mean   :113706
##  3rd Qu.:134185
##  Max.   :231545
```

The data has 6 variables and 397 observations. There are 3 factor and 3 numeric variables. Here, the target variable is “salary”. The other variables are the independent variables, to be used to predict the “salary”.

- rank is a factor with levels AssocProf, AsstProf, Prof
- discipline is a factor with levels A (“theoretical” departments) or B (“applied” departments).
- yrs.since.phd represents years since PhD.
- yrs.service represents years of service.
- sex is a factor with levels Female Male
- salary represents nine-month salary, in dollars

Here, the information about the variables in the data is listed as found in its source. Descriptive statistics of 6 variables are given. rank is a factor and it has 3 levels which are AssocProf, AsstProf, Prof and levels observed 67 times, 64 times and 266 times respectively. discipline also a factor which has 2 levels. These levels are A (“theoretical” departments) or B (“applied” departments). They observed 181 times and 216 times, respectively. yrs.since.pdh has 22.31 mean and ranges 1 to 56. yrs.service has 17.61 mean and ranges 0 to 60. sex is a factor which has 2 levels. These levels are Female or Male. They observed 39 times and 358 times, respectively. Finally, salary is the target variable and it has 113706 mean and ranges 57800 to 231545.

Splitting Data

A sample were created using "Salary" data. This sample has two parts one of them is going to be used for training to model and the other part for testing to the model. These are assigned to variables named `train` and `test`, respectively. Also, using `nrow(train)` and `nrow(test)` how many observations they contain is shown.

```
library(caret)
set.seed(2380)
index <- createDataPartition(data$salary, p = 0.75,
list = FALSE, times = 1)
train <- data[index,]
test <- data[-index,]
nrow(train)

## [1] 300

nrow(test)

## [1] 97
```

`train` has 300 and observations and `test` has 97 observations. After this process, the LRM and DTM is ready to be trained.

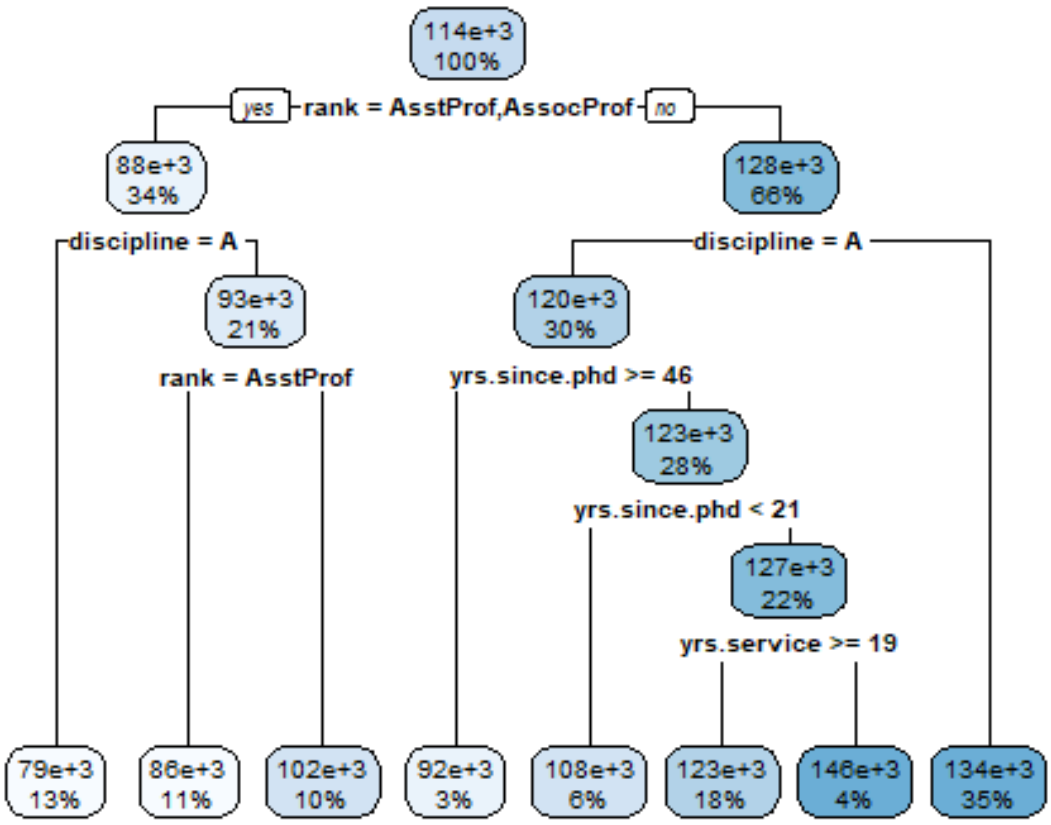
Linear Regression Model

```
#Traing Model
model <- lm(salary~.,data=train)
#Performance for Train
predicted_train <- predict(model,train)
modelEvaluation_train <- data.frame(train$salary,predicted_train)
colnames(modelEvaluation_train) <- c("Actual","Predicted_train")
mse_train <- mean((modelEvaluation_train$Actual - modelEvaluation_train$Predicted_train)^2)
print(c("Train:",sqrt(mse_train)))

## [1] "Train:"           "22673.4603176509"

#Performance for Test
predicted_test <- predict(model,test)
modelEvaluation_test <- data.frame(test$salary,predicted_test)
colnames(modelEvaluation_test) <- c("Actual","Predicted_test")
mse_test <- mean((modelEvaluation_test$Actual - modelEvaluation_test$Predicted_test)^2)
print(c("Test:",sqrt(mse_test)))

## [1] "Test:"            "21399.3185918163"
```

In this part, the Linear Regression Model were trained and the model error were calculated. This error is square root of mean square error. RMSE of Train is 22673,46 and for Test this value 21399,32. Test RMSE is smaller than Train RMSE. According to this there might be underfitting problem. Getting more training data, increasing the size or number of parameters in the model or increasing the complexity of the model may decrease or fix this problem.

Decision Trees

```
library(rpart)
library(rpart.plot)
#Training Model
model_dt <- rpart(salary ~. , method = "anova", data = train)
#Decision Tree
rpart.plot(model_dt)
```



Here the nodes take the color according to range of the variable. The higher the value of the salary variable, the darker the color of the nodes. For example, in the first node (root node), %40 of them is 2 so the node has grey color. Percentages are indicated on the last line of each box. These percentages indicate what percentage of all data is in that node. For root node, this percentage is 100 because the data did not splitted yet. If the rank is AsstProf. or AssocProf. then it goes left(yes). If it's not one of them then it goes right(no). Let's consider rank is AsstProf. Then it goes to left. That box colored lighter blue than previous node (root node), as mentioned before, new node (88000) is smaller than the root node (114000). And the percentage that written of last row of this box %34 so the main data splitted into two parts and %34 of them on the left and %66 of them on the right. If we consider discipline is A ("theoretical") then it goes left again. This node (leaf node) and the salary on this node is 79000. As a result, if the person has AsstProf. or AssocProf. rank and has "theoretical" discipline then the model predicts her/his nine-month salary around 79000$.

```r
#Performance on Test
preds_test <- predict(model_dt, test)
rmse_test <- sqrt(mean((preds_test - test$salary) ^ 2))
cat("test_rmse:", rmse_test,"\n")

## test_rmse: 21405.14

#Performance on Train
preds_train <- predict(model_dt, train)
rmse_train <- sqrt(mean((preds_train - train$salary) ^ 2))
cat("train_mse:", rmse_train)

## train_mse: 21539.73
```

Here, RMSE were calculated to see performance of the model on train and test sets. RMSE of Train is 21539,73 and for Test this value 21405,14. Test RMSE is smaller than Train RMSE. According to this there might be underfitting problem. Getting more training data, increasing the size or number of parameters in the model or increasing the complexity of the model may decrease or fix this problem. Also, this value can be used to compare models. Let's compare the LRM and DT;

```r
#Comparing models performances
data.frame(
  "LRM" = c(sqrt(mse_train),sqrt(mse_test)),
  "Decision Tree" = c(rmse_train,rmse_test),
  row.names = c("Train Root Mean S. Error", "Test Root Mean S. Error")
)

##                              LRM Decision.Tree
## Train Root Mean S. Error 22673.46      21539.73
## Test Root Mean S. Error  21399.32      21405.14
```

For better interpretations Errors of both models were displayed on a table. It is possible to say that the lower the error, the higher the predictive performance of the model. Based on this, if the error of the model is lower than the other model, it can be concluded that the model with low error is better. According to test errors LRM seems better than Decision Tree with a small difference. Also, according to the train error, it is possible to say that Decision Tree is better. Also, Decision Tree's test and train errors quite close each other. In this way, the Decision Tree model is expected to be more successful on new data but it should not be ignored that the success of LM on the test set is better than DT, so this inference can also be made for 004CM. According to the results of this study, unfortunately it is difficult to decide clearly which of the models is better.