Performance comparison of Regression Model, Regression Tree, Bagging Tree, Random Forest and Gradient Boosted Model

Machine Learning and Applications

Tufan Bostan

08/05/2021

In this work, bunch of model performance will be compared using R. Data to be used here is "penguins" from "`palmerpenguins`" package. Here, Includes measurements for penguin species, island in Palmer Archipelago, size (flipper length, body mass, bill dimensions), and sex. The data contains some missing values. They will be removed before training models. 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 trained. Finally, performance of the models on train and test sets will be interpreted for all model and their performances will be compared for this data.

Data Structure

First of all, "penguins" data introduced to R. The "penguins" data assigned to "data". All variables were correctly classified. But there are some missing values in the data set. First, these missing values removed (`na.omit(data)`) and then firs 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 <- palmerpenguins::penguins

#checking missing values
anyNA(data)

## [1] TRUE

dim(data)

## [1] 344   8

#removig missing values
data <- na.omit(data)

#checking missing values again
anyNA(data)

## [1] FALSE

dim(data)

## [1] 333   8

#DATA structure
head(data)

## # A tibble: 6 x 8
##   species island bill_length_mm bill_depth_mm flipper_length_~ body_mass_g sex
##   <fct>   <fct>           <dbl>         <dbl>            <int>       <int> <fct>
## 1 Adelie  Torge~           39.1          18.7              181        3750 male
## 2 Adelie  Torge~           39.5          17.4              186        3800 fema~
```

```
## 3 Adelie  Torge~           40.3          18                 195        3250 fema~
## 4 Adelie  Torge~           36.7          19.3               193        3450 fema~
## 5 Adelie  Torge~           39.3          20.6               190        3650 male
## 6 Adelie  Torge~           38.9          17.8               181        3625 fema~
## # ... with 1 more variable: year <int>
```

```
str(data)
```

```
## tibble [333 x 8] (S3: tbl_df/tbl/data.frame)
##  $ species          : Factor w/ 3 levels "Adelie","Chinstrap",..: 1 1 1 1 1 1 1 1 1 1 ..
.
##  $ island           : Factor w/ 3 levels "Biscoe","Dream",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ bill_length_mm   : num [1:333] 39.1 39.5 40.3 36.7 39.3 38.9 39.2 41.1 38.6 34.6 ...
##  $ bill_depth_mm    : num [1:333] 18.7 17.4 18 19.3 20.6 17.8 19.6 17.6 21.2 21.1 ...
##  $ flipper_length_mm: int [1:333] 181 186 195 193 190 181 195 182 191 198 ...
##  $ body_mass_g      : int [1:333] 3750 3800 3250 3450 3650 3625 4675 3200 3800 4400 ...
##  $ sex              : Factor w/ 2 levels "female","male": 2 1 1 1 2 1 2 1 2 2 ...
##  $ year             : int [1:333] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
##  - attr(*, "na.action")= 'omit' Named int [1:11] 4 9 10 11 12 48 179 219 257 269 ...
##   ..- attr(*, "names")= chr [1:11] "4" "9" "10" "11" ...
```

```
summary(data)
```

```
##       species          island    bill_length_mm  bill_depth_mm
##  Adelie   :146   Biscoe   :163   Min.   :32.10   Min.   :13.10
##  Chinstrap: 68   Dream    :123   1st Qu.:39.50   1st Qu.:15.60
##  Gentoo   :119   Torgersen: 47   Median :44.50   Median :17.30
##                                  Mean   :43.99   Mean   :17.16
##                                  3rd Qu.:48.60   3rd Qu.:18.70
##                                  Max.   :59.60   Max.   :21.50
##  flipper_length_mm  body_mass_g       sex           year
##  Min.   :172       Min.   :2700   female:165   Min.   :2007
##  1st Qu.:190       1st Qu.:3550   male  :168   1st Qu.:2007
##  Median :197       Median :4050                Median :2008
##  Mean   :201       Mean   :4207                Mean   :2008
##  3rd Qu.:213       3rd Qu.:4775                3rd Qu.:2009
##  Max.   :231       Max.   :6300                Max.   :2009
```

The data has 8 variables and 333 observations. There are 3 factor and 5 numeric variables. Here, the target variable is “body_mass_g”. The other variables are the independent variables, to be used to predict the “body_mass_g”.

- species represents a factor denoting penguin species (Adélie, Chinstrap and Gentoo).
- island represents a factor denoting island in Palmer Archipelago, Antarctica (Biscoe, Dream or Torgersen).
- bill_length_mm represents a number denoting bill length (millimeters).
- bill_depth_mm represents a number denoting bill depth (millimeters).
- flipper_length_mm represents an integer denoting flipper length (millimeters).
- body_mass_g represents an integer denoting body mass (grams).
- sex represents a factor denoting penguin sex (female, male).
- year represents an integer denoting the study year (2007, 2008, or 2009).

Here, the information about the variables in the data is listed as found in its source. Descriptive statistics of all variables are given. species is a factor and it has 3 levels which are Adelie, Chinstrap and Gento and levels observed 146 times, 68 times and 119 times respectively. island is a factor and it has 3 levels which are Biscoe, Dream and Torgersen and levels observed 163 times, 123 times and 47 times respectively. bill_length_mm has 43,99 mean and ranges 32,10 to 59,6. bill_depth_mm has 17,16 mean and ranges 13,10 to 21,5. flipper_length_mm has 201 mean and ranges 172 to 231. sex is a factor and it has 2 levels which are female and male and levels observed 165 times and 168 times

respectively. year has 2008 mean and ranges 2007 to 2009. Finally, body_mass_g is the target variable and it has 4207 mean and ranges 2700 to 6300.

Splitting Data

A sample were created using "penguins" 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.

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

## [1] 252

nrow(test)

## [1] 81
```

`train` has 252 and observations and `test` has 81 observations. After this process, models are ready to be trained.

Linear Regression Model

```
#Traing the Model
model_LM <- lm(body_mass_g~. ,data=train)
#Performance for Train
predicted_train_LM <- predict(model_LM, train)
rmse_train_LM <-sqrt(mean((predicted_train_LM - train$body_mass_g) ^ 2))

#Performance for Test
predicted_test_LM <- predict(model_LM, test)
rmse_test_LM <- sqrt(mean((predicted_test_LM - test$body_mass_g) ^ 2))

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

## test_rmse: 291.0292

cat("train_rmse:", rmse_train_LM)

## train_rmse: 281.4551
```

In this part, the Linear Regression Model were trained and the model errors were calculated. This is square root of mean square error (RMSE). RMSE of Train is 281,4551 and for Test this value 291,0292. Train RMSE is smaller than Test RMSE. According to RMSE, there might be overfitting problem. Training whit more data, removing some features or training an ensemble model can be applied to remove overfitting problem.

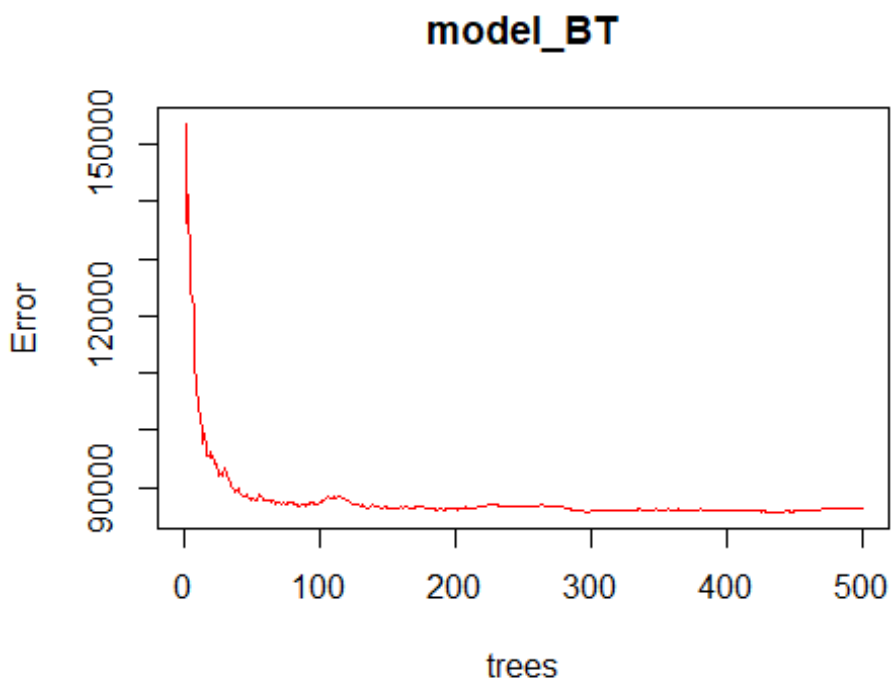
Regression Tree

```
#Training the Model
library(rpart)
model_DT <- rpart(body_mass_g ~. , method = "anova", data = train)

#Performance for test
predicted_test_DT <- predict(model_DT, test)
rmse_test_DT <- sqrt(mean((predicted_test_DT - test$body_mass_g) ^ 2))

#Performance for train
predicted_train_DT <- predict(model_DT, train)
rmse_train_DT <- sqrt(mean((predicted_train_DT - train$body_mass_g) ^ 2))

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

## test_rmse: 310.8597

cat("train_rmse:", rmse_train_DT)

## train_rmse: 310.9991
```

In this part, the model was trained by Regression Tree method and the model errors were calculated. RMSE of Train is 310,9991 and for Test this value 310,8597. Train RMSE and Test RMSE are almost equal each other. Thus, we might not be suspect for overfitting or underfitting problems.

Bagging Tree

```
#Training the Model
set.seed(2380)
library(randomForest)
model_BT <- randomForest(body_mass_g~. , data=train, mtry= 7)
plot(model_BT,col="red")
```

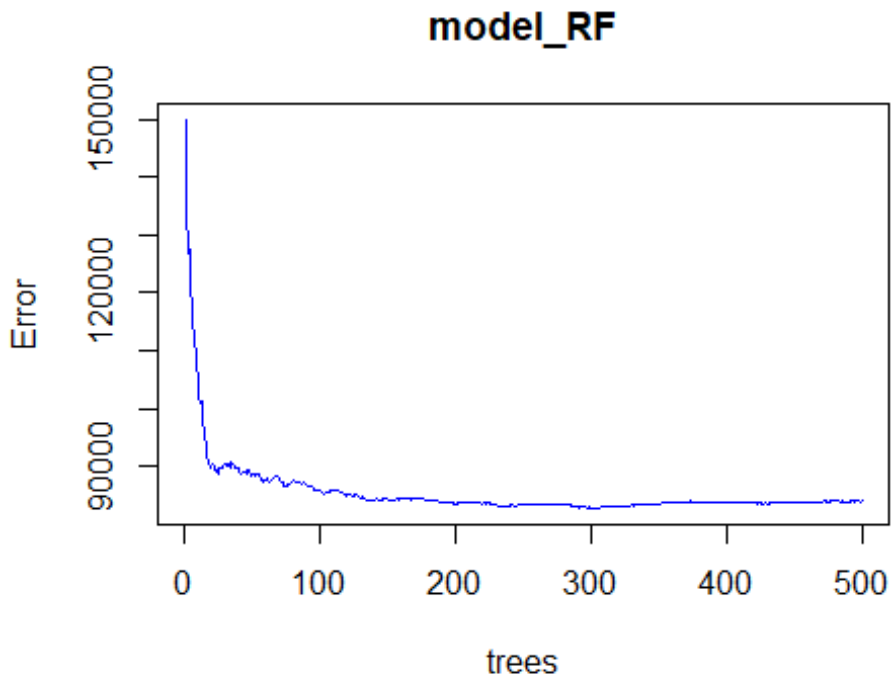


```r
#obtaining ntrees which has min. mse
which.min(model_BT$mse)

## [1] 435

#Performance for test
predicted_test_BT <- predict(model_BT, test)
rmse_test_BT <- sqrt(mean((predicted_test_BT - test$body_mass_g) ^ 2))

#Performance for train
predicted_train_BT <- predict(model_BT, train)
rmse_train_BT <- sqrt(mean((predicted_train_BT - train$body_mass_g) ^ 2))

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

## test_rmse: 306.7146

cat("train_rmse:", rmse_train_BT)

## train_rmse: 132.5069
```

In this part, the model was trained by Bagging Tree method and the model errors were calculated. RMSE of Train is 132,5069 and for Test this value 306,7146. Train RMSE is smaller than Test RMSE. According to RMSE, there might be overfitting problem. Training whit more data, removing some features or training an ensemble model can be applied to remove overfitting problem. Also, in this model optimal number of trees were determined as 435.

Random Forest

```r
#Training Model
set.seed(2380)
model_RF <- randomForest(body_mass_g ~ ., data = train)
plot(model_RF,col  = "blue")
```

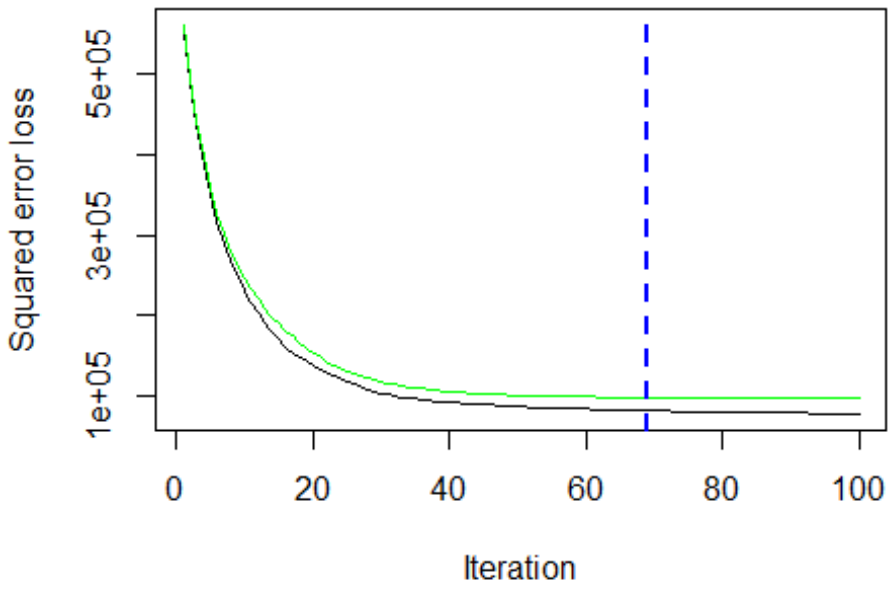


```
#obtaining ntrees which has min. mse
which.min(model_RF$mse)

## [1] 302

#Performance for test
predicted_test_RF <- predict(model_RF, test)
rmse_test_RF <- sqrt(mean((predicted_test_RF - test$body_mass_g) ^ 2))

#Performance for train
predicted_train_RF <- predict(model_RF, train)
rmse_train_RF <- sqrt(mean((predicted_train_RF - train$body_mass_g) ^ 2))

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

## test_rmse: 290.5424

cat("train_rmse:", rmse_train_RF)

## train_rmse: 173.4199
```

In this part, RMSE were calculated for Random Forest to see performance of the model on train and test sets. RMSE of Train is 173,4199 and for Test this value 290,5424. Train RMSE is way smaller than Test RMSE so probably there is overfitting problem. Training whit more data, removing some features or training an ensemble model can be applied to fix or decrease differences between train and test error. Also, in this model optimal number of trees were determined as 435.

GRADIENT BOOSTED MODEL

```
library(gbm)
#Training Model
set.seed(2380)
model_GBM <- gbm(body_mass_g ~ ., data = train, distribution = "gaussian", n.trees = 100, c
v.folds = 5)
gbm.perf(model_GBM)
```



```
## [1] 69

#Performance for test
predicted_test_GBM <- predict(model_GBM, test)

## Using 69 trees...

rmse_test_GBM <- sqrt(mean((predicted_test_GBM - test$body_mass_g) ^ 2))

#Performance for train
predicted_train_GBM <- predict(model_GBM, train)

## Using 69 trees...

rmse_train_GBM <- sqrt(mean((predicted_train_GBM - train$body_mass_g) ^ 2))

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

## test_rmse: 295.3791

cat("train_rmse:", rmse_train_GBM)

## train_rmse: 285.6021
```

In this part, RMSE were calculated for Gradient Boosted Model. RMSE of Train is 285,6021 and for Test this value 295,3791. Train RMSE is smaller than Test RMSE so probably there might be overfitting problem. Training whit more data, removing some features or training an ensemble model can be applied to fix or decrease differences between train and test error. Also, in this model optimal number of trees were determined as 435. Also, in this model optimal number of trees were determined as 69.

Comparing Models Performances

```
#Comparing models performances
data.frame(
  "TEST" = c(rmse_test_BT,rmse_test_DT,rmse_test_LM,rmse_test_RF,rmse_test_GBM),
  "TRAIN" = c(rmse_train_BT,rmse_train_DT,rmse_train_LM,rmse_train_RF,rmse_train_GBM),
  row.names = c("BAGGING TREE", "DECISION TREE","LINEAR MODEL","RANDOM FOREST","GRADIENT BO
OSTED")
)

##                       TEST    TRAIN
## BAGGING TREE     306.7146 132.5069
## DECISION TREE    310.8597 310.9991
## LINEAR MODEL     291.0292 281.4551
## RANDOM FOREST    290.5424 173.4199
## GRADIENT BOOSTED 295.3791 285.6021
```

For better interpretations Errors of all 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. It is not clear to say which model is best for all. In addition, some of the models' train errors way lower than the other models but differences between their test and train errors very high. So, there might be overfitting problem for these models which is not good. I could not say which is the best of all. Begging Tree and Random Forest might have low errors on train sets but for test these errors really far away from train so there might be overfitting problem here. On the other hand, Linear Model and Gradient Boosted models have a little bit smaller differences between test and train errors and Decision Tree have almost equal error scores but overall, all of the tree models have high RMSE scores.