Practical Machine Learning Course Project

Teppei Miyazaki

11/23/2021

Executive Summary

In this project, we would like to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants and to predict the manner in which they did the exercise.

Data

The participants were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

Prediction

The original training data has 19,622 observations of 160 variables, but not all of the variables look relevant. Therefore, I selected numeric columns which include no missing values.

To achieve high accuracy, I selected random forests for prediction and here is the summary of the model (R code is attached in the appendix):

Random Forest

19622 samples; 48 predictor; 5 classes: ‘A’, ‘B’, ‘C’, ‘D’, ‘E’

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 15698, 15697, 15698, 15697, 15698

Resampling results across tuning parameters:

mtry Accuracy Kappa

2 0.9945469 0.9931018

25 0.9940882 0.9925215

48 0.9884824 0.9854293

Accuracy was used to select the optimal model using the largest value.

The final value used for the model was mtry = 2.

Appendix: R Code

```
# set up
library(tidyverse)
library(caret)

# loading data
training <- read.csv('pml-training.csv')
testing <- read.csv('pml-testing.csv')
str(training)
summary(training)

# data pre-processing
x_train <- select(training,
                  ends_with(c("_x","_y","_z")),
                  starts_with(c("roll_","pitch_","yaw_")))
y_train <- as.factor(training$classe)
x_test <- select(testing,
                 ends_with(c("_x","_y","_z")),
                 starts_with(c("roll_","pitch_","yaw_")))

# random forests with a parallel implementation
library(parallel)
library(doParallel)
cluster <- makeCluster(detectCores() - 1) # convention to leave 1 core for OS
registerDoParallel(cluster)

set.seed(0)
fitControl <- trainControl(method = "cv", number = 5, allowParallel = TRUE)
modFit <- train(x_train, y_train, method="rf", data = training, trControl = fitCon
trol)

stopCluster(cluster)
registerDoSEQ()

modFit
predict(modFit, x_test)
```