quiz 4

maxim

24 сентября 2019 г

library(dplyr)  
library(caret)

## Warning: package 'caret' was built under R version 3.5.2

library(rpart)  
library(rattle)  
  
#library(AppliedPredictiveModeling)  
#library(ElemStatLearn)  
#library(pgmm)  
  
#library(gbm)  
#library(forecast)  
#library(e1071)  
#library(elasticnet)

#The goal of your project One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

#Deliverable To predict the manner in which they did the exercise. This is the “classe” variable in the training set. You may use any of the other variables to predict with.

You should create a report describing - how you built your model,  
- how you used cross validation,  
- what you think the expected out of sample error is,  
- why you made the choices you did.

You will also use your prediction model to predict 20 different test cases.

# Read the data

training <- read.csv("pml-training.csv",   
 header = T,   
 na.strings = c("NA","#DIV/0!",""))  
  
testing <- read.csv("pml-testing.csv",   
 header = T,   
 na.strings=c("NA","#DIV/0!",""))  
  
dim(training)

## [1] 19622 160

dim(testing)

## [1] 20 160

# Clean the data

Remove variables that have more than 20% missing values, or are not relevant to prediction

training <- training[ , colSums(is.na(training)) <= nrow(training) \* 0.2]  
  
training <- training %>%   
 select(-X, -user\_name, -raw\_timestamp\_part\_1, -raw\_timestamp\_part\_2,  
 -cvtd\_timestamp, -new\_window, -num\_window)  
ncol(training)

## [1] 53

# Check if there are any variables with near-zero variability

nearZeroVar(training)

## integer(0)

# Remove highly correlated variables

cor\_data <- training %>% select(-classe) %>% cor  
removecor <- findCorrelation(cor\_data, cutoff = .90, verbose = TRUE)

## Compare row 10 and column 1 with corr 0.992   
## Means: 0.27 vs 0.168 so flagging column 10   
## Compare row 1 and column 9 with corr 0.925   
## Means: 0.25 vs 0.164 so flagging column 1   
## Compare row 9 and column 4 with corr 0.928   
## Means: 0.233 vs 0.161 so flagging column 9   
## Compare row 8 and column 2 with corr 0.966   
## Means: 0.245 vs 0.157 so flagging column 8   
## Compare row 19 and column 18 with corr 0.918   
## Means: 0.091 vs 0.158 so flagging column 18   
## Compare row 46 and column 31 with corr 0.914   
## Means: 0.101 vs 0.161 so flagging column 31   
## Compare row 46 and column 33 with corr 0.933   
## Means: 0.083 vs 0.164 so flagging column 33   
## All correlations <= 0.9

training <- training %>% select(-removecor)  
ncol(training)

## [1] 46

#Split data to training and testing for cross validation

in\_train <- createDataPartition(y = training$classe, p=0.7, list=FALSE)  
training\_data <- training[in\_train, ]  
testing\_data <- training[-in\_train, ]  
  
dim(training\_data)

## [1] 13737 46

dim(testing\_data)

## [1] 5885 46

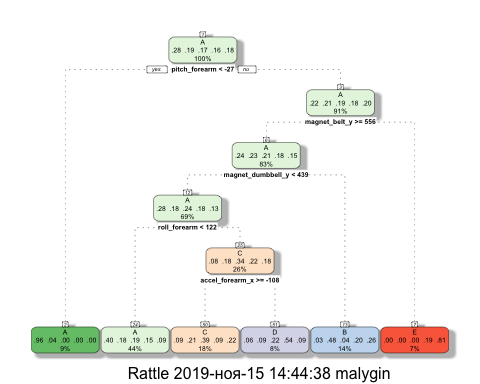
#Analysis

##rpart model

rpart\_model <- train(classe ~ ., method = "rpart", data = training\_data)  
rpart\_model$finalModel

## n= 13737   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 13737 9831 A (0.28 0.19 0.17 0.16 0.18)   
## 2) pitch\_forearm< -26.65 1246 54 A (0.96 0.043 0 0 0) \*  
## 3) pitch\_forearm>=-26.65 12491 9777 A (0.22 0.21 0.19 0.18 0.2)   
## 6) magnet\_belt\_y>=555.5 11468 8755 A (0.24 0.23 0.21 0.18 0.15)   
## 12) magnet\_dumbbell\_y< 438.5 9544 6893 A (0.28 0.18 0.24 0.18 0.13)   
## 24) roll\_forearm< 121.5 5985 3612 A (0.4 0.18 0.19 0.15 0.093) \*  
## 25) roll\_forearm>=121.5 3559 2354 C (0.078 0.18 0.34 0.22 0.18)   
## 50) accel\_forearm\_x>=-108.5 2507 1535 C (0.086 0.21 0.39 0.093 0.22) \*  
## 51) accel\_forearm\_x< -108.5 1052 487 D (0.059 0.091 0.22 0.54 0.091) \*  
## 13) magnet\_dumbbell\_y>=438.5 1924 1010 B (0.032 0.48 0.038 0.2 0.26) \*  
## 7) magnet\_belt\_y< 555.5 1023 198 E (0.00098 0.0029 0.002 0.19 0.81) \*

fancyRpartPlot(rpart\_model$finalModel)



##Cross Validation: rpart model I am going to check the performance of the rpart model on the testing data by cross validation.

rpart\_pred <- predict(rpart\_model, testing\_data)  
sum(rpart\_pred == testing\_data$classe) / length(rpart\_pred) # error rate

## [1] 0.4934579

The error rate of the rpart model is too high.

## random forest model

I am going tio apply random forest model.

model\_rf <- train(classe ~ ., data = training\_data, method = "rf")  
predict\_rf <- predict(model\_rf, newdata = testing\_data)

##Cross Validation: random forest model

cm\_rf <- confusionMatrix(predict\_rf, testing\_data$classe)  
cm\_rf$overall

## Accuracy Kappa AccuracyLower AccuracyUpper AccuracyNull   
## 0.9923534 0.9903266 0.9897815 0.9944172 0.2844520   
## AccuracyPValue McnemarPValue   
## 0.0000000 NaN

# Error

The model provides an accuracy of 99.5%/ This figure is an estimate of the out of sample error.

# Apply to final test set

Finally, I apply the random forest model to the final test data.

predict\_20 <- predict(model\_rf, newdata = testing)  
predict\_20

## [1] B A B A A E D B A A B C B A E E A B B B  
## Levels: A B C D E