Practical Machine Learning Assignment

Mary0523

September 19, 2018

Project goal

The goal of this project is to predict the manner in which people did barbell lifts. This is the “classe” variable in the training set. May use any of the other variables to predict with.
Use the prediction model to predict 20 different test cases.

Download training dataset

```
url<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
download.file(url,destfile ="C:/Users/may/Desktop/DataScience/Practical_Machine_Learning/Projec
t/pml-training.csv")
pml_training<-read.csv("C:/Users/may/Desktop/DataScience/Practical_Machine_Learning/Project/pml-
training.csv")
```

Cross validation by further divide the training dataset into subtraining(70%) and subtesting(30%) dataset

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.5.1
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
set.seed(12345)
inTrain<-createDataPartition(y=pml_training$classe,p=0.7,list=F)
subtraining<-pml_training[inTrain,]
subtesting<-pml_training[-inTrain,]
```

Data cleaning for subtraining and subtesting dataset

```
#remove the first 7 columns which are not variables can be used to predict classe
subtraining<-subset(subtraining,select=-c(1,2,3,4,5,6,7))
#After check each variables by str() function (not show up here) we found several variables cont
ains NA only.
#So delete those variables
subtraining<-subset(subtraining,select=c(colSums(is.na(subtraining))==0))
#Check near zero variance
NZV<-nearZeroVar(subtraining, saveMetrics=T)
#Delete variable with nzv=T
subtraining<-subtraining[,!NZV$nzv]

subtesting<-subset(subtesting,select=-c(1,2,3,4,5,6,7))
subtesting<-subset(subtesting,select=c(colSums(is.na(subtesting))==0))
subtesting<-subtesting[,!NZV$nzv]
```

Prediction models

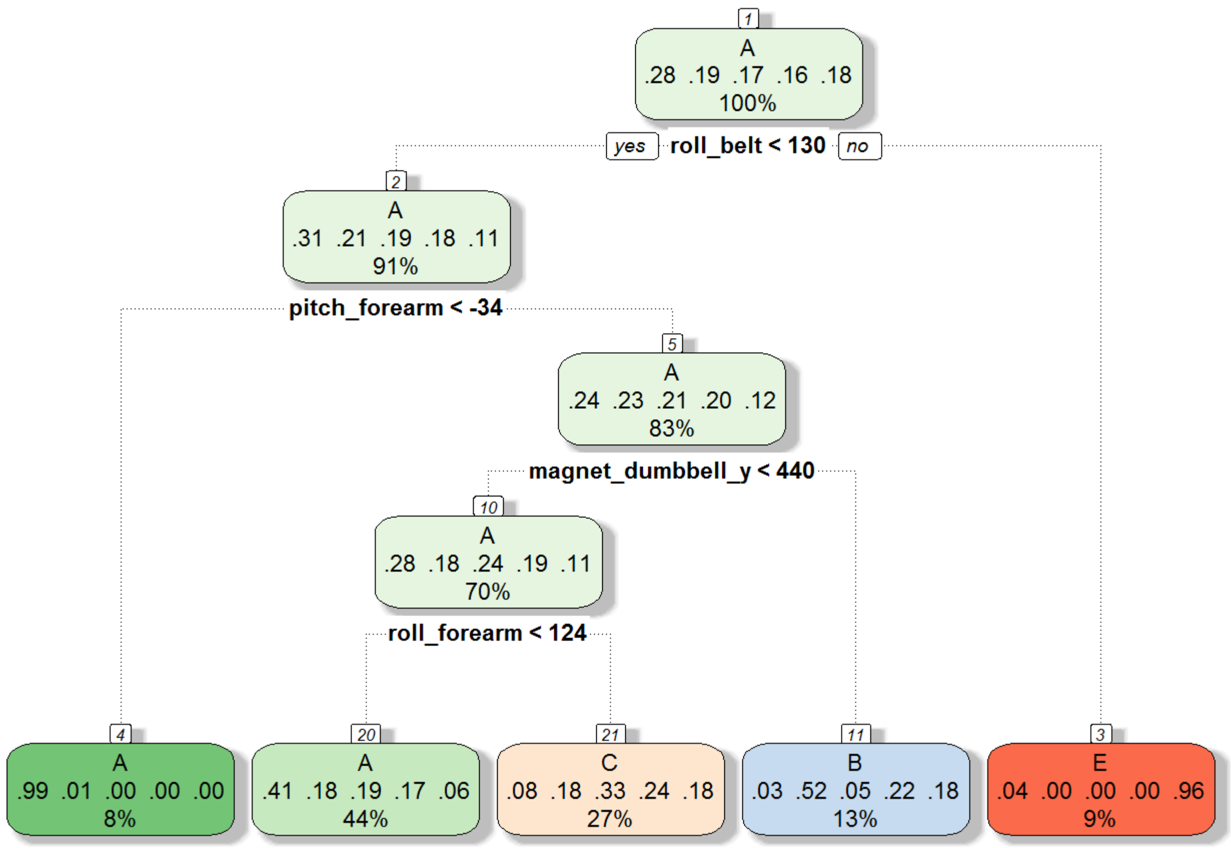
Predicting with decision tree: use a sample prediction model first

```
set.seed(12345)
modFit<-train(classe~.,method="rpart",data=subtraining)
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 3.5.1
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.2.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
fancyRpartPlot(modFit$finalModel)
```



Rattle 2018-Sep-21 11:54:26 may

```
pred<-predict(modFit,newdata=subtesting)
#Accuracy of the model
confusionMatrix(pred,subtesting$classe)$overall['Accuracy']
```

```
##  Accuracy
## 0.4963466
```

The decision tree is esay to interpret but this method could not have pruning or cross-validation which can lead to overfitting and the model can not estimate uncertainty well and its results may be variable.
From out results we can see the overall accuracy is 0.4963466, which is not very good.

Predict with more advanced models
Predicting with random forests
The default "rf" method runs very slow so use method in:
https://github.com/lgreski/datasciencectacontent/blob/master/markdown/pml-randomForestPerformance.md
(https://github.com/lgreski/datasciencectacontent/blob/master/markdown/pml-randomForestPerformance.md)

```
set.seed(12345)
#Step 1: Configure parallel processing
library(parallel)
library(doParallel)
```

```
## Warning: package 'doParallel' was built under R version 3.5.1
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.5.1
```

```
## Loading required package: iterators
```

```
## Warning: package 'iterators' was built under R version 3.5.1
```

```
cluster <- makeCluster(detectCores() - 1)
registerDoParallel(cluster)
#Step 2: Configure trainControl object
fitControl <- trainControl(method = "cv",
                           number = 5,
                           allowParallel = TRUE)
#Step 3: Develop training model
modFit_rf <- train(classe~., method="rf",data=subtraining,trControl = fitControl)
#Step 4: De-register parallel processing cluster
stopCluster(cluster)
registerDoSEQ()

pred_rf<-predict(modFit_rf,newdata=subtesting)
#Accuracy of the model
confusionMatrix(pred_rf,subtesting$classe)$overall['Accuracy']
```

```
##  Accuracy
## 0.9891249
```

The accuracy of this model is 0.9896347

Predicting with boosting

```
set.seed(12345)
modFit_gbm<-train(classe~.,method="gbm",data=subtraining,verbose=FALSE)
pred_gbm<-predict(modFit_gbm,newdata=subtesting)
#Accuracy of the model
confusionMatrix(pred_gbm,subtesting$classe)$overall['Accuracy']
```

```
##  Accuracy
## 0.9575191
```

The accuracy of this model is 0.9575191, worse than random forest

Combining predictors (random forest and boosting)

```
predcomb<-data.frame(pred_rf,pred_gbm,classe=subtesting$classe)
combModFit<-train(classe~.,method="rf",data=predcomb)
combPred<-predict(combModFit,newdata=predcomb)
#Accuracy of the model
confusionMatrix(combPred,subtesting$classe)$overall['Accuracy']
```

```
##  Accuracy
## 0.9891249
```

The accutacy of the combined model is 0.9891249, the same as random forest. So use random forest as the final prediction model.

Download testing dataset and apply random forest prediction on testing dataset

```
url<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(url,destfile ="C:/Users/may/Desktop/DataScience/Practical_Machine_Learning/Projec
t/pml-testing.csv")
pml_testing<-read.csv("C:/Users/may/Desktop/DataScience/Practical_Machine_Learning/Project/pml-t
esting.csv")

pml_testing<-as.data.frame(pml_testing)

final_pred<-predict(modFit_rf,newdata=pml_testing)
final_pred
```

```
##  [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
```

```
pml_testing$pred_classe<-as.factor(final_pred)
```