PML - Course project

Introduction

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.

Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. 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. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Data

The training data for this project are available here:

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

The test data are available here:

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

Libaries

```
library(caret)
```

```
## Warning: Paket 'caret' wurde unter R Version 4.1.2 erstellt

## Lade nötiges Paket: ggplot2

## Warning: Paket 'ggplot2' wurde unter R Version 4.1.2 erstellt

## Lade nötiges Paket: lattice
```

```
library(rpart)
```

```
## Warning: Paket 'rpart' wurde unter R Version 4.1.2 erstellt
```

```
library(randomForest)
```

```
## Warning: Paket 'randomForest' wurde unter R Version 4.1.2 erstellt
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attache Paket: 'randomForest'
```

```
## Das folgende Objekt ist maskiert 'package:ggplot2':
##
##     margin
```

Data Loading

```
training <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv")
testing <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv")

dim(training)
```

```
## [1] 19622   160
```

```
dim(testing)
```

```
## [1]  20 160
```

Data Cleaning

We need to clean the data set since many variables are NA and values that will not be included in the prediction. Also the training data needs to be splitted in training and validation set.

1. Removing the first 7 colums which would not be used for prediction

```
training <- training[,-c(1:7)]
testing <- testing[,-c(1:7)]
```

2. NA check for columns, removing colums with more than 70% of values as NA

```
naPercent <- colSums(is.na(training)) / nrow(training)
naCols <- naPercent < 0.7
training <- training[,naCols]
testing <- testing[,naCols]
#post-check
hasNA <- colSums(is.na(training)) > 0
names(training)[hasNA]
```

```
## character(0)
```

3. Near Zero Values

```
nearZero <- nearZeroVar(training, saveMetrics=TRUE)
training <-training[,nearZero$nzv== FALSE]
testing <- testing[,nearZero$nzv==FALSE]
```

4. Convert data type classe to factor

```
training$classe = factor(training$classe)
```

5. Splitting the training data in 70% training and 30% validation data

```
inTrain = createDataPartition(training$classe, p = 0.7, list=FALSE)
train <- training[inTrain, ]
val <- training[-inTrain, ]

dim(train)
```

```
## [1] 13737    53
```

```
dim(val)
```

```
## [1] 5885   53
```

Processing and Cross Validation

Decision tree

```
dtFit <- rpart(classe ~ ., data=train, method="class")
dtPred <- predict(dtFit, val, type = "class")
mean(dtPred == val$classe)
```

```
## [1] 0.7330501
```

Random Forrest

```
set.seed(1)
rfFit <- randomForest(classe ~ ., data=train, ntree=500)
rfPred <- predict(rfFit, val)
mean(rfPred == val$classe)
```

```
## [1] 0.9954121
```

Random Forrest worked very well on the validation set, we can use it on the testing set.

Prediction for the testing set

```r
result <- predict(rfFit, testing)
print(result)
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 
##  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
```