Prediction Assignment Writeup

Abe1985

Sunday, December 14, 2014

Executive Summary

This is an R Markdown document that describes which method i used tp predict in which way people perform a certain exercise. In this project, the goal was 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. This was documented by the the "classe" variable, which has the following values: Exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E). For more details on the data and how it was collected see the Appendix.

The Model was obtained by removing all missing and near zero values. Moreover all variables that cannot influence in which way the exercise was performed, f.e. index number or name of the subject. Then the Model was estimated via random forest. The in sample error of the model was 0.82%. I also did a prediction on my testset: The model had an error rate of 0.75% on my testset. The out of sample error estimated via cross validation is 0.62%.

Data preperation

1st removing missing values

First, I read the training set into R and removed the missing values, that are represented by NAs:

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.2

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

```
trainraw <- read.csv("pml-training.csv")
colneeded <- c() #creates an empty vector that will contain all columns needed
k <- 0 #variable needed to count
#if the column contains more than 19 Na's it will be excluded in our count. Only the number
#of the columns with fewerer NA's will be stored
for(i in 1:length(names(trainraw))){
        if(sum(is.na(trainraw[i]))>19){next}
        else{ k <- k + 1
                colneeded[[k]] <- i}
        }
trains <- trainraw[,colneeded]
sum(is.na(trains)) #make sure no Nas are left
```

```
## [1] 0
```

2nd remove near zero values

```
library(caret)
nsv <- nearZeroVar(trains, saveMetrics=FALSE)
#all numbers of the variables that are near zero are stored in nsv
trainset <- trains[,(-nsv)]
```

The commands above will delete all columns that contain near zero values, meaning variables that nearly remain the same over all subjects and are therefore useless for prediction.

3rd split dataset

In order to evaluate the model and evaluate the error rate, I split the data in a testing and a training set:

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.1.2
```

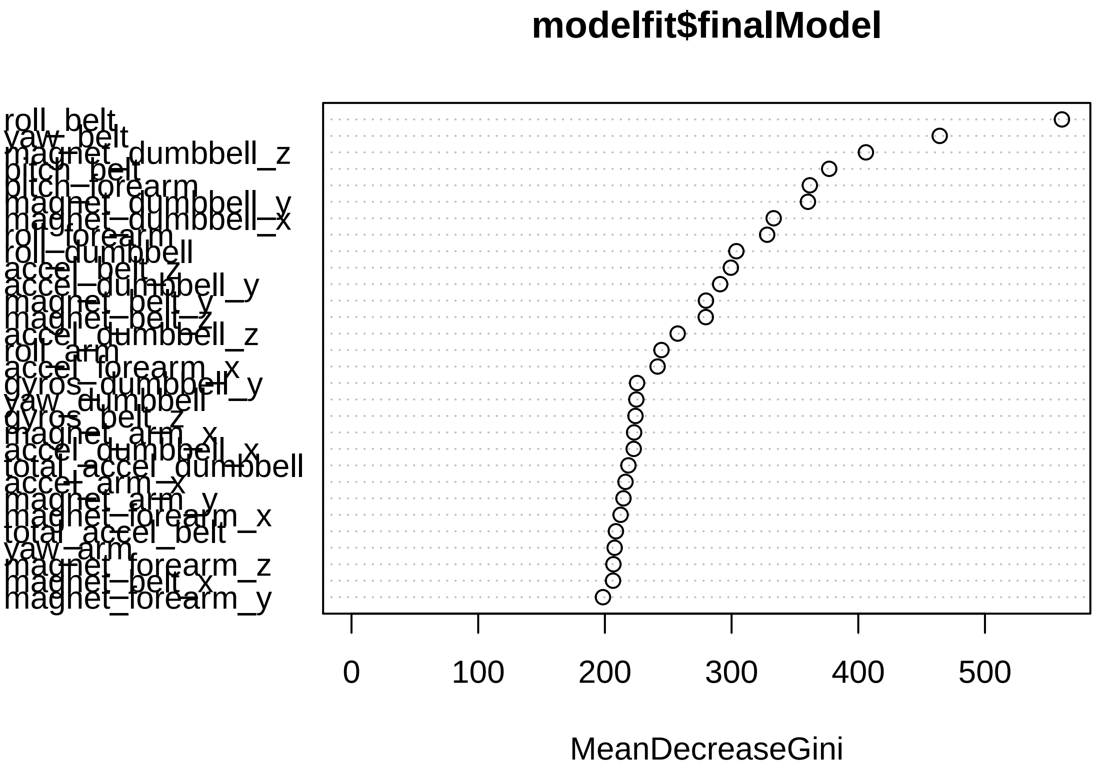
```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(444)
inTrain <- createDataPartition(y=trainset$classe, p=0.75, list=FALSE)
training <- trainset[inTrain,]
testing <- trainset[-inTrain,]
```

Prediction Model

Because with randomforest it is very likely to overfit the model it is important to use cross validation (cv). I used the trControl argument to include cv. **Exclude irrelevant variables** Moreover I have excluded the columns with variables that are useless predicting class. For example,X is simply the indexnumber. Because the variables are sorted by class, this might look as if the indexnumber has an influence on class, which of course does not make sense.

```
modelfit <- train(as.factor(classe) ~., method="rf", trControl = trainControl(method="cv", number=3), da
varImpPlot(modelfit$finalModel)
```



Error rate

```
modelfit$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.62%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 4184    1    0    0    0   0.0002389
## B   19 2825    4    0    0   0.0080758
## C    0   18 2546    3    0   0.0081808
## D    0    0   40 2371    1   0.0169983
## E    0    0    1    4 2701   0.0018477
```

The in sample error of the model was 0.82%. I also did a prediction on my testset:

```
pred <- predict(modelfit, newdata=testing)
table(pred,testing$classe)
```

```
##
## pred    A    B    C    D    E
##    A 1395    4    0    0    0
##    B    0  941   11    0    0
##    C    0    4  843   18    0
##    D    0    0    1  784    0
##    E    0    0    0    2  901
```

The model had an error rate of 0.75% on my testset.

The out of sample error estimated via cross validation is 0.62%.

Appendix

The test and the training set were provided from http://groupware.les.inf.puc-rio.br/har and are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.