Practical Machine Learning Project

Rohit Singh

May 29, 2020

Download files

```
#install.packages("caret")
#install.packages("randomForest")
#install.packages("rpart")
#install.packages("rpart.plot")
library("rpart")
library("randomForest")
```

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

```
url="https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
download.file(url, "pml-training.csv",method='curl')
url="https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(url, "pml-testing.csv",method='curl')
```

Read,view and clean datasets

```
# Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes. Participants were supervised by an experienced weightlifter to make sure the execution complied to the manner they were supposed to simulate. The exercises were performed by six male participants aged between 20-28 years, with little weightlifting experience. We made sure that all participants could easily simulate the mistakes in a safe and controlled manner by using a relatively light dumbbell (1.25kg).
#
# Read more: http://groupware.les.inf.puc-rio.br/har#ixzz3isa6f0Ds

training=read.csv("pml-training.csv",head=T, na.string=c("NA","#DIV/0!", ""))
testing=read.csv("pml-testing.csv",head=T,na.string=c("NA","#DIV/0!", ""))

# Delete columns with more than 20% missing values

training<-training[ , apply(training , 2 ,
function (x) sum(is.na(x)) < 0.2 *nrow(training)) ]
dim(training)
```

```
## [1] 19622 60
```

```
Testing<-testing[ , apply(testing , 2 ,
function (x) sum(is.na(x)) < 0.2 *nrow(testing)) ]
dim(testing)
```

```
## [1] 20 60
```

```
head(training)[1:10]
```

```
## X user_name raw_timestamp_part_1 raw_timestamp_part_2 cvtd_timestamp
## 1 1 carlitos 1323084231 788290 05/12/2011 11:23
## 2 2 carlitos 1323084231 808298 05/12/2011 11:23
## 3 3 carlitos 1323084231 820366 05/12/2011 11:23
## 4 4 carlitos 1323084232 120339 05/12/2011 11:23
## 5 5 carlitos 1323084232 196328 05/12/2011 11:23
## 6 6 carlitos 1323084232 304277 05/12/2011 11:23
## new_window num_window roll_belt pitch_belt yaw_belt
## 1 no 11 1.41 8.07 -94.4
## 2 no 11 1.41 8.07 -94.4
## 3 no 11 1.42 8.07 -94.4
## 4 no 12 1.48 8.05 -94.4
## 5 no 12 1.48 8.07 -94.4
## 6 no 12 1.45 8.06 -94.4
```

```
head(testing)[1:10]
```

```
## X user_name raw_timestamp_part_1 raw_timestamp_part_2 cvtd_timestamp
## 1 1 pedro 1323095002 868349 05/12/2011 14:23
## 2 2 jeremy 1322673067 778725 30/11/2011 17:11
## 3 3 jeremy 1322673075 342967 30/11/2011 17:11
## 4 4 adelmo 1322832789 560311 02/12/2011 13:33
## 5 5 eurico 1322489635 814776 28/11/2011 14:13
## 6 6 jeremy 1322673149 510661 30/11/2011 17:12
## new_window num_window roll_belt pitch_belt yaw_belt
## 1 no 74 123.00 27.00 -4.75
## 2 no 431 1.02 4.87 -88.90
## 3 no 439 0.87 1.82 -88.50
## 4 no 194 125.00 -41.60 162.00
## 5 no 235 1.35 3.33 -88.60
## 6 no 504 -5.92 1.59 -87.70
# Columns 1-7 we can delete too
```

```
training <-training[,-c(1:7)]
testing <-testing[,-c(1:7)]
set.seed(848)
# Random subsampling without replacement (60%)
subsamples= sample(1:nrow(training),size=nrow(training)*0.6,replace=F)
subTraining <- training[subsamples, ]
subTesting <- training[-subsamples, ]
dim(subTraining)
```

```
## [1] 11773 53
```

```
dim(subTesting)
```

```
## [1] 7849 53
```

Frequency of levels (A, B, C, D, E) in the subTraining dataset for variable “classe”

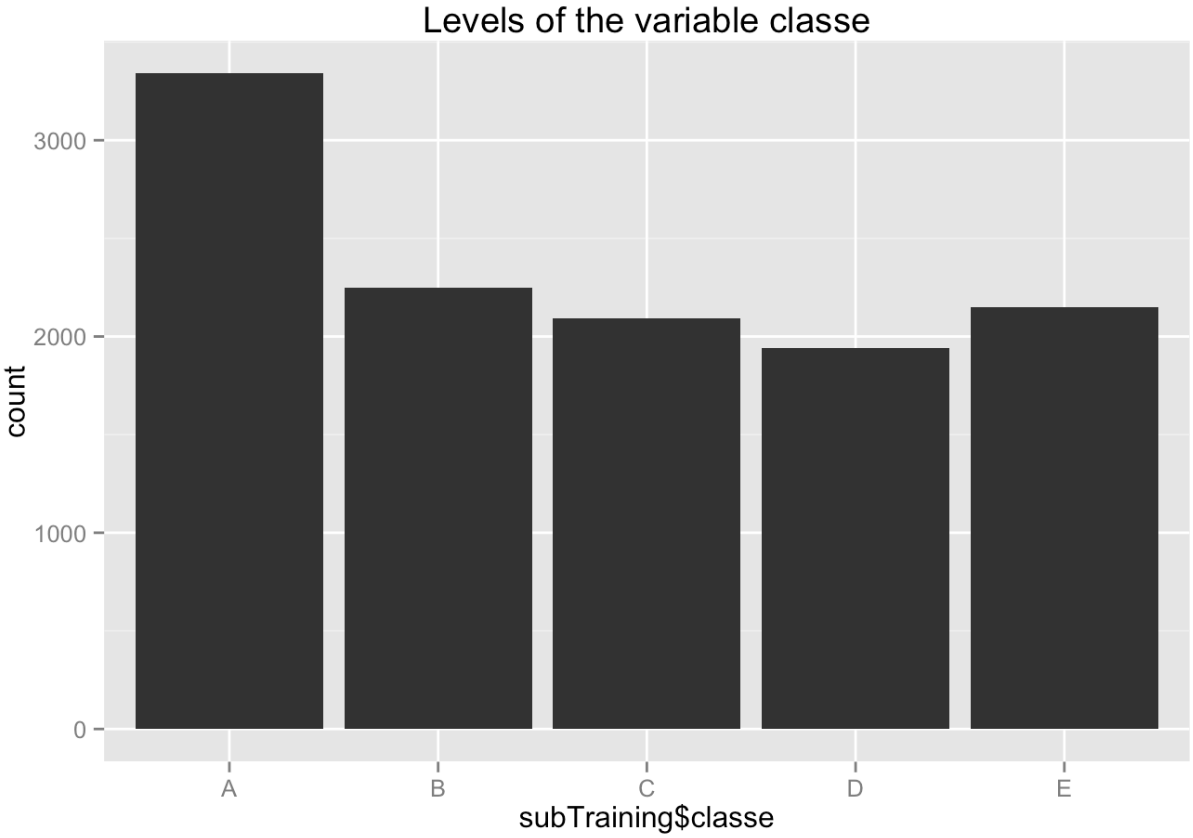
```
library("caret")
```

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

```
summary(subTraining$classe)
```

```
## A B C D E
## 3340 2249 2092 1943 2149
```

```
qplot(subTraining$classe,
      main="Levels of the variable classe")
```



Correlation Analysis

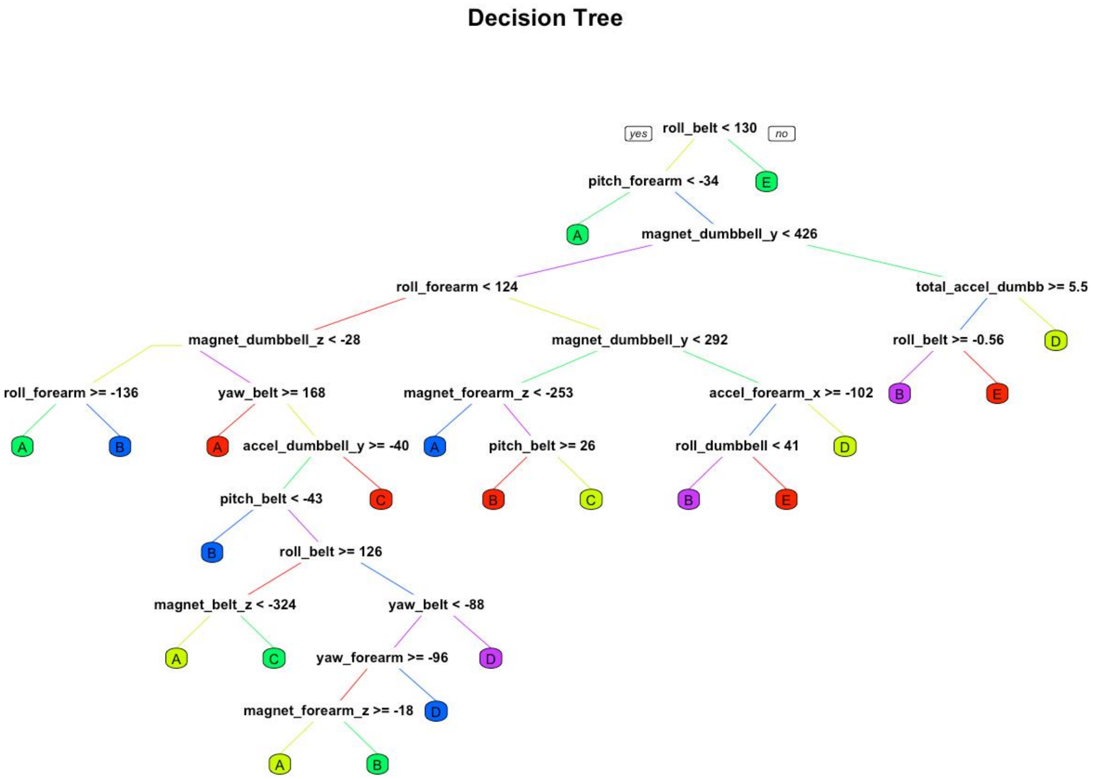
```
correlation <- findCorrelation(cor(subTraining[, 1:ncol(subTraining)-1]), cutoff=0.8)
names(subTraining)[correlation]
```

```
## [1] "accel_belt_z" "roll_belt" "accel_belt_y"
## [4] "accel_dumbbell_z" "accel_belt_x" "pitch_belt"
## [7] "accel_dumbbell_x" "accel_arm_x" "magnet_arm_y"
## [10] "gyros_arm_x"
```

Prediction model 1: Decision Tree

```
library("rpart")
model1 <- rpart(classe ~ .,
subTraining,
method="class")
#model1
predictions1<-predict(model1, subTesting,type ="class")
cols=rainbow(5)
library("rpart.plot")
```

```
rpart.plot(model1, main="Decision Tree",box.col=cols, branch.col=cols)
```



```
confusionMatrix(predictions1, subTesting$classe)
```

```
## Confusion Matrix and Statistics
##
## Reference
## Prediction A B C D E
## A 1956 246 23 57 31
## B 103 982 131 103 193
## C 67 122 1016 184 167
## D 91 107 87 856 120
## E 23 91 73 73 947
##
## Overall Statistics
##
## Accuracy : 0.7335
## 95% CI : (0.7235, 0.7432)
## No Information Rate : 0.2854
## P-Value [Acc > NIR] : < 2.2e-16
##
## Kappa : 0.6625
## Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
## Class: A Class: B Class: C Class: D Class: E
## Sensitivity 0.8732 0.6344 0.7639 0.6724 0.6495
## Specificity 0.9364 0.9159 0.9172 0.9384 0.9593
## Pos Pred Value 0.8457 0.6495 0.6530 0.6788 0.7846
## Neg Pred Value 0.9487 0.9107 0.9501 0.9367 0.9231
## Prevalence 0.2854 0.1972 0.1694 0.1622 0.1858
## Detection Rate 0.2492 0.1251 0.1294 0.1091 0.1207
## Detection Prevalence 0.2947 0.1926 0.1982 0.1607 0.1538
## Balanced Accuracy 0.9048 0.7751 0.8405 0.8054 0.8044
```

Prediction model 2: Random Forest

```
library("randomForest")
model2<- randomForest(classe ~ .,
subTraining,
method="class")
model2
```

```
##
## Call:
## randomForest(formula = classe ~ ., data = subTraining, method = "class")
## Type of random forest: classification
## Number of trees: 500
## No. of variables tried at each split: 7
##
## OOB estimate of error rate: 0.59%
## Confusion matrix:
## A B C D E class.error
## A 3335 5 0 0 0 0.001497006
## B 13 2231 5 0 0 0.008003557
## C 0 9 2081 2 0 0.005258126
## D 0 0 24 1918 1 0.012866701
## E 0 0 2 8 2139 0.004653327
```

```
predictions2<-predict(model2, subTesting,type="class")
confusionMatrix(predictions2, subTesting$classe)
```

```
## Confusion Matrix and Statistics
##
## Reference
## Prediction A B C D E
## A 2239 8 0 0 0
## B 0 1535 10 0 0
## C 0 5 1319 14 0
## D 0 0 1 1259 6
## E 1 0 0 0 1452
##
## Overall Statistics
##
## Accuracy : 0.9943
## 95% CI : (0.9923, 0.9958)
## No Information Rate : 0.2854
## P-Value [Acc > NIR] : < 2.2e-16
##
## Kappa : 0.9927
## Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
## Class: A Class: B Class: C Class: D Class: E
## Sensitivity 0.9996 0.9916 0.9917 0.9890 0.9959
## Specificity 0.9986 0.9984 0.9971 0.9989 0.9998
## Pos Pred Value 0.9964 0.9935 0.9858 0.9945 0.9993
## Neg Pred Value 0.9998 0.9979 0.9983 0.9979 0.9991
## Prevalence 0.2854 0.1972 0.1694 0.1622 0.1858
## Detection Rate 0.2853 0.1956 0.1680 0.1604 0.1850
## Detection Prevalence 0.2863 0.1968 0.1705 0.1613 0.1851
## Balanced Accuracy 0.9991 0.9950 0.9944 0.9940 0.9979
```

Predictions for both models

```
predict(model1, testing,type="class")
```

```
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
## B A E D A C D B A A C E C A E D A B B B
## Levels: A B C D E
```

```
predict(model2, testing,type="class")
```

```
## 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
```

Decision about one of the two prediction model

Accuracy for Random Forest model - 0.9943 (95% CI : (0.9923, 0.9958)).

Accuracy for Decision Tree model - 0.7335 (95% CI: (0.7235, 0.7432)).

The Random Forests model is better.