

Machine Learning Course Project

Zelepukin Dmitriy

12 august 2017

The goal of this project is to predict the manner in which they did the exercise. This is the “classe” variable in the training set.

Load libraries and data sets

There is need to prepare data for a random forest method

```
train <- train[,-c(1:7)]  
table(train$classe)
```

```
##  
##      A      B      C      D      E  
## 5580 3797 3422 3216 3607
```

```
new.names <- c("classe")  
  
for(i in names(train)[1:152]){  
  
  #--- step 1  
  share.na <- round(sum(is.na(train[,i]))*100/19622,2)  
  print(paste0(i, " has NAs: ", share.na, "%"))  
  
  #--- step 2  
  if(share.na==0){  
    new.names<-c(new.names,i)  
  }  
}
```

```
## [1] "roll_belt has NAs: 0%"
## [1] "pitch_belt has NAs: 0%"
## [1] "yaw_belt has NAs: 0%"
## [1] "total_accel_belt has NAs: 0%"
## [1] "kurtosis_roll_belt has NAs: 97.98%"
## [1] "kurtosis_pitch_belt has NAs: 98.09%"
## [1] "kurtosis_yaw_belt has NAs: 100%"
## [1] "skewness_roll_belt has NAs: 97.98%"
## [1] "skewness_roll_belt.1 has NAs: 98.09%"
## [1] "skewness_yaw_belt has NAs: 100%"
## [1] "max_roll_belt has NAs: 97.93%"
## [1] "max_pitch_belt has NAs: 97.93%"
## [1] "max_yaw_belt has NAs: 97.98%"
## [1] "min_roll_belt has NAs: 97.93%"
## [1] "min_pitch_belt has NAs: 97.93%"
## [1] "min_yaw_belt has NAs: 97.98%"
## [1] "amplitude_roll_belt has NAs: 97.93%"
## [1] "amplitude_pitch_belt has NAs: 97.93%"
## [1] "amplitude_yaw_belt has NAs: 97.98%"
## [1] "var_total_accel_belt has NAs: 97.93%"
## [1] "avg_roll_belt has NAs: 97.93%"
## [1] "stddev_roll_belt has NAs: 97.93%"
## [1] "var_roll_belt has NAs: 97.93%"
## [1] "avg_pitch_belt has NAs: 97.93%"
## [1] "stddev_pitch_belt has NAs: 97.93%"
## [1] "var_pitch_belt has NAs: 97.93%"
## [1] "avg_yaw_belt has NAs: 97.93%"
## [1] "stddev_yaw_belt has NAs: 97.93%"
## [1] "var_yaw_belt has NAs: 97.93%"
## [1] "gyros_belt_x has NAs: 0%"
## [1] "gyros_belt_y has NAs: 0%"
## [1] "gyros_belt_z has NAs: 0%"
## [1] "accel_belt_x has NAs: 0%"
## [1] "accel_belt_y has NAs: 0%"
## [1] "accel_belt_z has NAs: 0%"
## [1] "magnet_belt_x has NAs: 0%"
## [1] "magnet_belt_y has NAs: 0%"
## [1] "magnet_belt_z has NAs: 0%"
## [1] "roll_arm has NAs: 0%"
## [1] "pitch_arm has NAs: 0%"
## [1] "yaw_arm has NAs: 0%"
## [1] "total_accel_arm has NAs: 0%"
## [1] "var_accel_arm has NAs: 97.93%"
## [1] "avg_roll_arm has NAs: 97.93%"
## [1] "stddev_roll_arm has NAs: 97.93%"
## [1] "var_roll_arm has NAs: 97.93%"
## [1] "avg_pitch_arm has NAs: 97.93%"
## [1] "stddev_pitch_arm has NAs: 97.93%"
## [1] "var_pitch_arm has NAs: 97.93%"
## [1] "avg_yaw_arm has NAs: 97.93%"
## [1] "stddev_yaw_arm has NAs: 97.93%"
## [1] "var_yaw_arm has NAs: 97.93%"
## [1] "gyros_arm_x has NAs: 0%"
## [1] "gyros_arm_y has NAs: 0%"
## [1] "gyros_arm_z has NAs: 0%"
## [1] "accel_arm_x has NAs: 0%"
## [1] "accel_arm_y has NAs: 0%"
```

```
## [1] "accel_arm_z has NAs: 0%"
## [1] "magnet_arm_x has NAs: 0%"
## [1] "magnet_arm_y has NAs: 0%"
## [1] "magnet_arm_z has NAs: 0%"
## [1] "kurtosis_roll_arm has NAs: 98.33%"
## [1] "kurtosis_pitch_arm has NAs: 98.34%"
## [1] "kurtosis_yaw_arm has NAs: 97.99%"
## [1] "skewness_roll_arm has NAs: 98.32%"
## [1] "skewness_pitch_arm has NAs: 98.34%"
## [1] "skewness_yaw_arm has NAs: 97.99%"
## [1] "max_roll_arm has NAs: 97.93%"
## [1] "max_pitch_arm has NAs: 97.93%"
## [1] "max_yaw_arm has NAs: 97.93%"
## [1] "min_roll_arm has NAs: 97.93%"
## [1] "min_pitch_arm has NAs: 97.93%"
## [1] "min_yaw_arm has NAs: 97.93%"
## [1] "amplitude_roll_arm has NAs: 97.93%"
## [1] "amplitude_pitch_arm has NAs: 97.93%"
## [1] "amplitude_yaw_arm has NAs: 97.93%"
## [1] "roll_dumbbell has NAs: 0%"
## [1] "pitch_dumbbell has NAs: 0%"
## [1] "yaw_dumbbell has NAs: 0%"
## [1] "kurtosis_roll_dumbbell has NAs: 97.96%"
## [1] "kurtosis_pitch_dumbbell has NAs: 97.94%"
## [1] "kurtosis_yaw_dumbbell has NAs: 100%"
## [1] "skewness_roll_dumbbell has NAs: 97.95%"
## [1] "skewness_pitch_dumbbell has NAs: 97.94%"
## [1] "skewness_yaw_dumbbell has NAs: 100%"
## [1] "max_roll_dumbbell has NAs: 97.93%"
## [1] "max_pitch_dumbbell has NAs: 97.93%"
## [1] "max_yaw_dumbbell has NAs: 97.96%"
## [1] "min_roll_dumbbell has NAs: 97.93%"
## [1] "min_pitch_dumbbell has NAs: 97.93%"
## [1] "min_yaw_dumbbell has NAs: 97.96%"
## [1] "amplitude_roll_dumbbell has NAs: 97.93%"
## [1] "amplitude_pitch_dumbbell has NAs: 97.93%"
## [1] "amplitude_yaw_dumbbell has NAs: 97.96%"
## [1] "total_accel_dumbbell has NAs: 0%"
## [1] "var_accel_dumbbell has NAs: 97.93%"
## [1] "avg_roll_dumbbell has NAs: 97.93%"
## [1] "stddev_roll_dumbbell has NAs: 97.93%"
## [1] "var_roll_dumbbell has NAs: 97.93%"
## [1] "avg_pitch_dumbbell has NAs: 97.93%"
## [1] "stddev_pitch_dumbbell has NAs: 97.93%"
## [1] "var_pitch_dumbbell has NAs: 97.93%"
## [1] "avg_yaw_dumbbell has NAs: 97.93%"
## [1] "stddev_yaw_dumbbell has NAs: 97.93%"
## [1] "var_yaw_dumbbell has NAs: 97.93%"
## [1] "gyros_dumbbell_x has NAs: 0%"
## [1] "gyros_dumbbell_y has NAs: 0%"
## [1] "gyros_dumbbell_z has NAs: 0%"
## [1] "accel_dumbbell_x has NAs: 0%"
## [1] "accel_dumbbell_y has NAs: 0%"
## [1] "accel_dumbbell_z has NAs: 0%"
## [1] "magnet_dumbbell_x has NAs: 0%"
## [1] "magnet_dumbbell_y has NAs: 0%"
## [1] "magnet_dumbbell_z has NAs: 0%"
## [1] "roll_forearm has NAs: 0%"
```

```
## [1] "pitch_forearm has NAs: 0%"
## [1] "yaw_forearm has NAs: 0%"
## [1] "kurtosis_roll_forearm has NAs: 98.36%"
## [1] "kurtosis_pitch_forearm has NAs: 98.36%"
## [1] "kurtosis_yaw_forearm has NAs: 100%"
## [1] "skewness_roll_forearm has NAs: 98.35%"
## [1] "skewness_pitch_forearm has NAs: 98.36%"
## [1] "skewness_yaw_forearm has NAs: 100%"
## [1] "max_roll_forearm has NAs: 97.93%"
## [1] "max_pitch_forearm has NAs: 97.93%"
## [1] "max_yaw_forearm has NAs: 98.36%"
## [1] "min_roll_forearm has NAs: 97.93%"
## [1] "min_pitch_forearm has NAs: 97.93%"
## [1] "min_yaw_forearm has NAs: 98.36%"
## [1] "amplitude_roll_forearm has NAs: 97.93%"
## [1] "amplitude_pitch_forearm has NAs: 97.93%"
## [1] "amplitude_yaw_forearm has NAs: 98.36%"
## [1] "total_accel_forearm has NAs: 0%"
## [1] "var_accel_forearm has NAs: 97.93%"
## [1] "avg_roll_forearm has NAs: 97.93%"
## [1] "stddev_roll_forearm has NAs: 97.93%"
## [1] "var_roll_forearm has NAs: 97.93%"
## [1] "avg_pitch_forearm has NAs: 97.93%"
## [1] "stddev_pitch_forearm has NAs: 97.93%"
## [1] "var_pitch_forearm has NAs: 97.93%"
## [1] "avg_yaw_forearm has NAs: 97.93%"
## [1] "stddev_yaw_forearm has NAs: 97.93%"
## [1] "var_yaw_forearm has NAs: 97.93%"
## [1] "gyros_forearm_x has NAs: 0%"
## [1] "gyros_forearm_y has NAs: 0%"
## [1] "gyros_forearm_z has NAs: 0%"
## [1] "accel_forearm_x has NAs: 0%"
## [1] "accel_forearm_y has NAs: 0%"
## [1] "accel_forearm_z has NAs: 0%"
## [1] "magnet_forearm_x has NAs: 0%"
## [1] "magnet_forearm_y has NAs: 0%"
## [1] "magnet_forearm_z has NAs: 0%"
```

```
new.train <- train[,new.names]
new.train$classe <- as.factor(new.train$classe)
table(new.train$classe)
```

```
##
##      A      B      C      D      E
## 5580 3797 3422 3216 3607
```

Random Forest method prediction - training and cross-validation checking:

```
set.seed(1234)
cv3 = trainControl(method="cv",number=3,allowParallel=TRUE,verboseIter=TRUE)
modrf = train(classe~., data=new.train, method="rf",trControl=cv3)
```

```
## + Fold1: mtry= 2
## - Fold1: mtry= 2
## + Fold1: mtry=27
## - Fold1: mtry=27
## + Fold1: mtry=52
## - Fold1: mtry=52
## + Fold2: mtry= 2
## - Fold2: mtry= 2
## + Fold2: mtry=27
## - Fold2: mtry=27
## + Fold2: mtry=52
## - Fold2: mtry=52
## + Fold3: mtry= 2
## - Fold3: mtry= 2
## + Fold3: mtry=27
## - Fold3: mtry=27
## + Fold3: mtry=52
## - Fold3: mtry=52
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 2 on full training set
```

And add a decision tree model:

```
modtree = train(classe~.,data=new.train,method="rpart",trControl=cv3)
```

```
## + Fold1: cp=0.03568
## - Fold1: cp=0.03568
## + Fold2: cp=0.03568
## - Fold2: cp=0.03568
## + Fold3: cp=0.03568
## - Fold3: cp=0.03568
## Aggregating results
## Selecting tuning parameters
## Fitting cp = 0.0357 on full training set
```

```
prf = predict(modrf,new.train)
ptree = predict(modtree,new.train)
```

Let's check models:

```
table(prf,new.train$classe)
```

```
##
## prf    A    B    C    D    E
##  A 5580    0    0    0    0
##  B    0 3797    0    0    0
##  C    0    0 3422    0    0
##  D    0    0    0 3216    0
##  E    0    0    0    0 3607
```

```
table(ptree,new.train$classe)
```

```
##  
## ptree      A      B      C      D      E  
##      A 5080 1581 1587 1449 524  
##      B   81 1286  108  568 486  
##      C  405  930 1727 1199 966  
##      D    0    0    0    0    0  
##      E   14    0    0    0 1631
```

```
prf=predict(modrf,test)  
ptree=predict(modtree,test)  
table(prf,ptree)
```

```
##      ptree  
## prf A B C D E  
##   A 7 0 0 0 0  
##   B 3 0 5 0 0  
##   C 0 0 1 0 0  
##   D 0 0 1 0 0  
##   E 1 0 2 0 0
```

Finally, the random forest model runs with a better accuracy.