Practical Machine Learning Course: Final Project

Rakhmanova Amina

Overview

This document is the final report of the Peer Assessment project from Coursera's course Practical Machine Learning, as part of the Specialization in Data Science.

This analysis is the basis for the course quiz and a prediction assignment writeup. The main goal of the project is to predict the manner in which 6 participants performed some exercise as described below. This is the "classe" variable in the training set.

A: exactly according to the specification B: throwing the elbows to the front C: lifting the dumbbell only halfway D: lowering the dumbbell only halfway E: throwing the hips to the front

Dataset

The data for this project come from this source: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har).

The training data for this project are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

We need to get rid off the variables which have plenty of NA. In addition, we'll remove Near Zero variance and identification variables.

```
NZV = nearZeroVar(training)
training = training[, -NZV]
testing = testing[, -NZV]

na = sapply(training, function(x) mean(is.na(x))) > 0.95
training = training[, na==FALSE]
testing = testing[, na==FALSE]

training = training[, -(1:5)]
testing = testing[, -(1:5)]

dim(training)
```

```
## [1] 19622    54
```

```
dim(testing)
```

```
## [1] 20 54
```

```
set.seed(1)
intrain = createDataPartition(training$classe, p=0.8, list=FALSE)
train = training[intrain, ]
test = training[-intrain, ]

dim(train)
```

```
## [1] 15699    54
```

```
dim(test)
```

```
## [1] 3923   54
```

Prediction model building

Random Forests and Generalized Boosted Model will be applied to model the regressions and the best one will be used for the quiz predictions.

A Confusion Matrix is plotted at the end of each analysis to better visualize the accuracy of the models.

Random Forest method

```
set.seed(1)
controlRF = trainControl(method="cv", number=3, verboseIter=FALSE)
modFitRF = train(classe ~ ., data=train, method="rf", trControl=controlRF)
modFitRF$finalModel
```

```
## 
## Call:
##  randomForest(x = x, y = y, mtry = min(param$mtry, ncol(x))) 
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
## 
##         OOB estimate of  error rate: 0.22%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 4463    0    0    0    1 0.0002240143
## B    7 3027    4    0    0 0.0036208032
## C    0    5 2733    0    0 0.0018261505
## D    0    0   12 2560    1 0.0050524679
## E    0    0    0    5 2881 0.0017325017
```

```
predictRF = predict(modFitRF, newdata=test)
test$classe = as.factor(test$classe)
confMatRF = confusionMatrix(predictRF, test$classe)
confMatRF$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull 
##      0.9989804      0.9987103      0.9973914      0.9997221      0.2844762 
## AccuracyPValue  McnemarPValue 
##      0.0000000            NaN
```

Generalized Boosted Model

```
set.seed(1)
controlGBM = trainControl(method = "repeatedcv", number = 5, repeats = 1)
modFitGBM  = train(classe ~ ., data=train, method = "gbm", trControl = controlGBM, ve
rbose = FALSE)
modFitGBM$finalModel
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 53 predictors of which 52 had non-zero influence.
```

```
predictGBM = predict(modFitGBM, newdata=test)
confMatGBM = confusionMatrix(predictGBM, test$classe)
confMatGBM$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull 
##      0.9895488      0.9867776      0.9858482      0.9924899      0.2844762 
## AccuracyPValue  McnemarPValue 
##      0.0000000            NaN
```

Comparison

We have analyzed two representation models. The first one proved to be the best with accuracy 0,999. That’s why we will use Random forest to predict the 20 quiz results.

Prediction Assignment

```
pred_test = predict(modFitRF, newdata=testing)
pred_test
```

```
##  [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
```