Course project: Practical Machine Learning

Anatoli Kraev

10/24/2019

About

This is project for the **Practical Machine Learning** course in Coursera's Data Science specialization. The aim of the course project is to create a model of behavior for a group of people involved in weightlifting. It is necessary to predict how they did the exercises (the "Classe" variable in the training set). Further information is available on the website: http://groupware.les.inf.puc-rio.br/har

Loading and preprocessing the data

```
library(caret)
```

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

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

```
library(rpart)
if(file.exists("Data/pml-training.csv")){
  print("File already downloaded")
} else{
  train.URL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
  download.file(train.URL, "Data/pml-training.csv")
}
```

```
## [1] "File already downloaded"
```

```
if(file.exists("Data/pml-testing.csv")){
  print("File already downloaded")
} else{
  test.URL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
  download.file(test.URL, "Data/pml-testing.csv")
}
```

```
## [1] "File already downloaded"
```

```
Training <- read.csv("Data/pml-training.csv")
Testing <- read.csv("Data/pml-testing.csv")
dim(Training); dim(Testing)
```

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

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

Identification and delition Of Near Zero Variance Predictors

```
 NZV <- nearZeroVar(Training)
Training <- Training[, -NZV]
```

Removing predictors containing more than 95% NA

```
ClearColNum <- colSums(is.na(Training))/dim(Training)[1]
Training <- Training[, ClearColNum < 0.05]
```

Removing predictors that do not make sense for prediction

```
Training <- Training[, -(1 : 6)]
dim(Training)
```

```
## [1] 19622    53
```

```
names(Training)
```

```
##  [1] "roll_belt"            "pitch_belt"           "yaw_belt"
##  [4] "total_accel_belt"     "gyros_belt_x"         "gyros_belt_y"
##  [7] "gyros_belt_z"         "accel_belt_x"         "accel_belt_y"
## [10] "accel_belt_z"         "magnet_belt_x"        "magnet_belt_y"
## [13] "magnet_belt_z"        "roll_arm"             "pitch_arm"
## [16] "yaw_arm"              "total_accel_arm"      "gyros_arm_x"
## [19] "gyros_arm_y"          "gyros_arm_z"          "accel_arm_x"
## [22] "accel_arm_y"          "accel_arm_z"          "magnet_arm_x"
## [25] "magnet_arm_y"         "magnet_arm_z"         "roll_dumbbell"
## [28] "pitch_dumbbell"       "yaw_dumbbell"         "total_accel_dumbbell"
## [31] "gyros_dumbbell_x"     "gyros_dumbbell_y"     "gyros_dumbbell_z"
## [34] "accel_dumbbell_x"     "accel_dumbbell_y"     "accel_dumbbell_z"
## [37] "magnet_dumbbell_x"    "magnet_dumbbell_y"    "magnet_dumbbell_z"
## [40] "roll_forearm"         "pitch_forearm"        "yaw_forearm"
## [43] "total_accel_forearm"  "gyros_forearm_x"      "gyros_forearm_y"
## [46] "gyros_forearm_z"      "accel_forearm_x"      "accel_forearm_y"
## [49] "accel_forearm_z"      "magnet_forearm_x"     "magnet_forearm_y"
## [52] "magnet_forearm_z"     "classe"
```

To assess the error of the created model, we will divide the basic training data into training and verification parts

```
set.seed(1234)
inTrain <- createDataPartition(y=Training$classe, p = 0.7, list = FALSE)
Ttrain <- Training[inTrain, ]
Ttest <- Training[-inTrain, ]
dim(Ttrain); dim(Ttest)
```

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

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

Model Building

Consider four different model-building algorithms:

1.Recursive Partitioning And Regression Trees

2. Bagging

3. Boosted trees

4. Random forest

Cross-validation is performed for each Ensemble models (2 ,3, 4) with K = 3.

```
fitControl <- trainControl(method="cv", number=3, verboseIter=F)
TreeModel <- rpart(classe ~ ., data=Ttrain, method="class")
BaggModel <- train(classe ~ ., data=Ttrain, method="treebag", trControl=fitControl)
GbmModel <- train(classe ~ ., data=Ttrain, method="gbm", trControl=fitControl, verbose = FALSE)
RfModel <- train(classe ~ ., data=Ttrain, method="rf", trControl=fitControl, ntree=100)
```

Model Evaluation

```
predTREE <- predict(TreeModel, newdata = Ttest, type = "class")
cmTree <- confusionMatrix(predTREE, Ttest$classe)
predBAGG <- predict(BaggModel, newdata = Ttest)
cmBagg <- confusionMatrix(predBAGG, Ttest$classe)
predGBM <- predict(GbmModel, newdata = Ttest)
cmGBM <- confusionMatrix(predGBM, Ttest$classe)
predRF <- predict(RfModel, newdata = Ttest)
cmRF <- confusionMatrix(predRF, Ttest$classe)
ResultsAccuracy <- data.frame(Model = c("RPART","BAGGING", "GBM", "RF"),
                   Accurasy = rbind(cmTree$overall[1], cmBagg$overall[1], cmGBM$overall[1], cmRF$overall[1]
))
print(ResultsAccuracy)
```

```
##     Model  Accuracy
## 1   RPART 0.7541206
## 2 BAGGING 0.9833475
## 3     GBM 0.9660153
## 4      RF 0.9942226
```

In our case, it is clear that Ensemble models are superior then Recursive Partitioning And Regression Trees. The best model is Random Forest. The confusion matrix Random Forest model is below:

```
cmRF$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    4    0    0    0
##          B    0 1131   13    0    0
##          C    0    4 1011    9    1
##          D    0    0    2  954    0
##          E    0    0    0    1 1081
```

Prediction

Apply the Random Forest model to the validation data: “pml-testing.csv”

```
predTesting <- predict(RfModel, newdata = Testing)
Resut <- data.frame(problem_id = Testing$problem_id, predicted = predTesting)
print(Resut)
```

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

Conclusion

Based on the data provided for the project, it was possible to select a model with high accuracy - Random Forest.

In principle, all the investigated models belonging to the Ensemble showed very good results.