Machine Learning Project

Manuel Velasco

21 de diciembre de 2016

The process of generating the model to predict the unknown test set is described in this document.

The following packages have been used:

```
library(lattice)
library(ggplot2)
library(caret)
library(randomForest)
```

The seed is set to a fixed value:

```
set.seed(32335)
```

First, the data sets are loaded and stored as data frames:

```
training_set <- read.csv("~/MachineLearningProject/data/pml-training.csv", stringsAsFactors = FALSE)
testing_set <- read.csv("~/MachineLearningProject/data/pml-testing.csv", stringsAsFactors = FALSE)
```

Once loaded, it is clear that the training data set contains a lot of unnecessary variables. Therefore, only the variables that are also present in the testing data set are considered to construct the model. This reduces the number of variables from 160 to 60.

```
testing_set <- testing_set[, colSums(is.na(testing_set)) != nrow(testing_set)]
training_set <- training_set[, colSums(is.na(training_set)) != nrow(training_set)]
training_set_skp <- training_set[, names(training_set) %in% names(testing_set)]
training_set_skp$classe <- training_set$classe
```

To construnct the model, the training data set is partitioned in a training and a testing set:

```
inTrain <- createDataPartition(y=training_set_skp$classe, p=0.7, list=FALSE)
training <- training_set_skp[inTrain, ]
testing <- training_set_skp[-inTrain, ]
```

And the first variables, which contain information about the user name or time information, are also removed because we are only interested in the information provided by the sensors:

```
training <- training[, 8:60]
testing <- testing[, 8:60]
```

Several methods have been considered for the construction of the model. The random forest method, although has the higher computational cost, provides the highest accuracy. Therefore, that method is used to construct the model.

```
modFit_rf <- train(classe ~ ., data=training, method="rf")
```

Once the model has been created, the testing set is used to measure the accuracy of the model:

```
pred_rf <- predict(modFit_rf, testing)
```

Examining the confusion matrix, it can be seen that the accuracy is of 99% approximately:

```
confusionMatrix(pred_rf, testing$classe)$overall[1]
```

```
##  Accuracy
```

```
## 0.9906542
```

Finally, the type of exercises of the unknown test set can be predicted using the generated model:

```
pred_rf_unknown <- predict(modFit_rf, testing_set)
print(pred_rf_unknown)
```

```
##  [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
```