Machine Learning - Final Project

MJ

3 April 2016

```r
library(dplyr)
library(caret)
library(rattle)
library(randomForest)
library(rpart)
```

Data processing

Load data

```r
training <- read.csv("./data/pml-training.csv", na.strings = c("NA", ""))
testing <- read.csv("./data/pml-testing.csv", na.strings = c("NA", ""))
```

Edit data

We get rid of columns with missing data.

```r
training <- training[, colSums(is.na(training)) == 0]
testing <- testing[, colSums(is.na(testing)) == 0]
```

We also remove columns predictors that are not relevant to predict the classe (first seven columns)

```r
training <- training[, -c(1:7)]
testing <- testing[, -c(1:7)]
```

To be able to get out-of-sample errors we need to split the training data into training and validation sets.

```r
set.seed(100)
inTrain <- createDataPartition(training$classe, p = 0.7, list = FALSE)
train <- training[inTrain, ]
valid <- training[-inTrain, ]
```

Analysis

We will compare results from two different models - a classifiacation tree and a random forest.

Classification Tree

```
control <- trainControl(method = "cv", number = 5)
fit_rpart <- train(classe ~ ., data = train, method = "rpart",
                   trControl = control)
modFitRPART <- train(classe ~ ., method="rpart", data=train,
trControl=control)
predRPART <- predict(modFitRPART, valid)
confusionMatrix(predRPART, valid$classe)$overall[1]

## Accuracy
## 0.484452
```

Random Forest

```
modFitRF <- train(classe ~ ., data = train, method = "rf",trControl =
control)
predRF <- predict(modFitRF, valid)
confusionMatrix(predRF, valid$classe)$overall[1]

##  Accuracy
## 0.9940527
```

Random forest appears most accurate and so we use it to predict the test data. The predicted values are then:

```
(predict(modFitRF, testing))

##  [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
```