Practical machine learning course project

Background

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, the goal is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways.

Goal

The goal of the project is to predict the manner in which they did the exercise. This is given by the "classe" variable in the training set.

Loading data

```
training <- read.csv("pml-training.csv", header = TRUE);
testing <- read.csv("pml-testing.csv", header = TRUE);
```

Preprocessing

Partitioning training set

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.2.5
```

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

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

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.2.5
```

```
set.seed(123456)
train_index <- createDataPartition(training$classe, p = 0.8, list = FALSE)
train_set <- training[train_index, ]
validate_set <- training[-train_index, ]
```

Removal of non-useful predictors

Predictors which do not have too many unique values, or have too many NA values or are only descriptive statistics are eliminated.

```
zcol <- nearZeroVar(train_set)
train_set <- train_set[, -zcol]

cnt <- sapply(train_set, function(x) { sum(!(is.na(x) | x == ""))})

nullcol <- names(cnt[cnt < 0.7 * length(train_set$classe)])
descriptcol <- c("X", "user_name", "raw_timestamp_part_1", "raw_timestamp_part_2", "cvtd_timestamp", "n
excludecols <- c(descriptcol, nullcol)
train_set <- train_set[, !names(train_set) %in% excludecols]
```

Prediction model

```
m1 <- train(classe ~ ., data = train_set, method = "rf")
```

```
## Loading required package: randomForest

## Warning: package 'randomForest' was built under R version 3.2.5

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
p1 <- predict(m1, validate_set)

confusionMatrix(p1, validate_set$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1116    5    0    0    0
##          B    0  751    6    0    0
##          C    0    3  677    9    0
##          D    0    0    1  634    3
##          E    0    0    0    0  718
##
```

```
## Overall Statistics
##
##                Accuracy : 0.9931
##                  95% CI : (0.99, 0.9955)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9913
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9895   0.9898   0.9860   0.9958
## Specificity            0.9982   0.9981   0.9963   0.9988   1.0000
## Pos Pred Value         0.9955   0.9921   0.9826   0.9937   1.0000
## Neg Pred Value         1.0000   0.9975   0.9978   0.9973   0.9991
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2845   0.1914   0.1726   0.1616   0.1830
## Detection Prevalence   0.2858   0.1930   0.1756   0.1626   0.1830
## Balanced Accuracy      0.9991   0.9938   0.9930   0.9924   0.9979
```

out-off sample error

```
accuracy <- sum(p1 == validate_set$classe)/length(p1)
```