Machine Learning Course Project: Prediction Assignment

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, your goal will be 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. More information is available from the website here: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Goal

The goal of your project is to predict the manner in which they did the exercise.
1. Preict "classe" variable in the training set with any of the other variables.
2. Describe how the model is built.
3. How cross validation is applied.
4. Expected out of sample error.
5. Why you made the choices you did.
6. use prediction model to predict 20 different test cases.

Loading Data

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

In this dataset we have following categories:
exactly according to the specification (Class A)
throwing the elbows to the front (Class B)
lifting the dumbbell only halfway (Class C)
lowering the dumbbell only halfway (Class D)
throwing the hips to the front (Class E)

We will use the training data to predict the catagories in testing dataste.

Note that data contains both NA and #DIV/0!.

```
training <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", na.strings=
testing <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", na.strings=c

set.seed(33)
```

The data loaded contains some unnessary information:
1, the first 7 column are not related with prediction at all.
2, the data contained a huge amount of NAs. Those columns should be filtered out.

```
training <- training[,-c(1:7)]
testing <- testing[,-c(1:7)]
training <- training[,colSums(is.na(training))==0]
length(names(training))
```

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

Data we got now contains only 52 predictors.

Analyze

Since this problem is clearly a classification problem, random forest & KNN are certainly than regression. I chose random forest to resolve this issue.

Also, I used 70% data to build the model and 30% data to do cross validation. I except the out of sample error to be at least 95%.

```
training_index <- createDataPartition(training$classe, p=0.7, list=F)
training_data <- training[training_index,]
test_data <- training[-training_index,]
control <- trainControl(method="cv",number = 5,allowParallel = TRUE)
rf_model <- train(classe~., data=training_data, method="rf",trControl=control)
print(rf_model)
```

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10991, 10989, 10989, 10990, 10989
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9903906  0.9878419
##   27    0.9906815  0.9882104
##   52    0.9852219  0.9813019
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

The model gave ~99% Accuracy for mtry=2, which means that the model is accurate. So, I expect the out of sample error to be low (accuracy > 95%).

```
test_p <- predict(rf_model,test_data)

confusionMatrix(test_p, test_data$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1672    3    0    0    0
##          B    1 1133    7    0    1
##          C    0    3 1013   10    2
```

```
##          D    0    0    6  954    2
##          E    1    0    0    0 1077
##
## Overall Statistics
##
##                Accuracy : 0.9939
##                  95% CI : (0.9915, 0.9957)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9923
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9988   0.9947   0.9873   0.9896   0.9954
## Specificity            0.9993   0.9981   0.9969   0.9984   0.9998
## Pos Pred Value         0.9982   0.9921   0.9854   0.9917   0.9991
## Neg Pred Value         0.9995   0.9987   0.9973   0.9980   0.9990
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2841   0.1925   0.1721   0.1621   0.1830
## Detection Prevalence   0.2846   0.1941   0.1747   0.1635   0.1832
## Balanced Accuracy      0.9990   0.9964   0.9921   0.9940   0.9976
```

Result

We can see that the Accuracy is 0.9939. We can safely apply it to test data. The result for 20 tests is:

```
predict(rf_model,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
```