Activity Quality Prediction

We want to predict Classe variable based on available data.

Reading and Cleaning Data

First step is to load the data sets.

```
library(caret)
```

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

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

```
pmlTraining <- read.csv("pml-training.csv",header = TRUE, na.strings=c("NA",""))
pmlTesting <- read.csv("pml-testing.csv",header = TRUE, na.strings=c("NA",""))
```

Many columns are empty or mostly NA. We want to remove these columns from our modeling. In addition the first six columns are time and user information which we don't need in our model.

```
remove <-pmlTraining[,colSums(is.na(pmlTraining)) == 0]
build <- pmlTraining[,colSums(is.na(pmlTraining)) == 0]
build <- build[,-(1:6)]
test <- pmlTesting[,colSums(is.na(pmlTraining)) == 0]
test <- test[,-(1:6)]
```

The next step is to divide data to training and validating sets.

```
trainIndex = createDataPartition(build$classe, p = 0.7,list=FALSE)
training = build[trainIndex,]
validating = build[-trainIndex,]
```

Fitting a Model

We fit a model to the training data set using random forest method.

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

Now we use this model to predict Classe in validating data set and get the out of sample accuracy rate from the confusion matrix.

```
confusionMatrix(validating$classe, predict(rfFit, validating))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    0    0    0    1
##          B    0 1138    1    0    0
##          C    0    0 1023    3    0
##          D    0    0    5  958    1
##          E    0    0    0    7 1075
##
## Overall Statistics
##
```

```
##                Accuracy : 0.9969
##                  95% CI : (0.9952, 0.9982)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9961
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   0.9942   0.9897   0.9981
## Specificity            0.9998   0.9998   0.9994   0.9988   0.9985
## Pos Pred Value         0.9994   0.9991   0.9971   0.9938   0.9935
## Neg Pred Value         1.0000   1.0000   0.9988   0.9980   0.9996
## Prevalence             0.2843   0.1934   0.1749   0.1645   0.1830
## Detection Rate         0.2843   0.1934   0.1738   0.1628   0.1827
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9999   0.9999   0.9968   0.9942   0.9983
```

As we can see in the confusion matrix output, accuracy is 0.99.

Predicting

The final step is to use the model to predict outcome for test data set.

```
predict(rfFit,test)
```

```
##  [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
```