

```
knitr::opts_chunk$set(echo = TRUE)
```

I was able to predict the levels of the test set by applying the Random Forest and Decision Tree algorithms.

#Summary of the process:

Of the data, 25% was used for CrossValidation and the remaining 75% was used for training.

The two separate algorithms of Random Forest and Decision Tree were used.

With the decision tree, the accuracy of evaluating CrossValidation was at 75%, while the accuracy of evaluating the data with Random Forest was at 99.8%.

I then predicted the levels for the test set, and determined how much influence each of the 53 variables gave in determining the machine learning algorithm's evaluation.

【Download packages】

```
install.packages("caret");require(caret)
install.packages("randomForest")
require(randomForest)
install.packages("raprt"); require(rpart)
```

【Data loading, remove Na columns for the training and testing sets】

```
## DATA Loading#
traindata <- read.csv("pml-training.csv", na.strings = c("NA", "#DIV/0!", ""))
testdata <- read.csv("pml-testing.csv", na.strings = c("NA", "#DIV/0!", ""))
# remove NA columns for the training and testing data
comps <- complete.cases(t(traindata)) & complete.cases(t(testdata))
traindata1 <- traindata[,comps]
testdata1 <- testdata[,comps]
# Drop the first 7 columns as they're unnecessary for predicting.
traindata2<- traindata1[,8:length(colnames(traindata1))]
testdata2 <- testdata1[,8:length(colnames(testdata1))]
```

【Corss Validation】

```
#Cross Validation#
set.seed(32323)
in.training <- createDataPartition(traindata2$classe, p=0.75, list=F)
Train <- traindata2[in.training, ]
Train$classe
Cross <- traindata2[-in.training, ]
Train$classe = as.factor(Train$classe)
```

【Decision Tree】

```
## Decision Tree #
require(rpart);require(rpart.plot)
Dtree <- rpart(Train$classe ~ ., data = Train)
pdtree <- predict(Dtree, newdata = testdata2, type = "class")
val_DT <- predict(Dtree, newdata = Cross, type = "class")
table(pdtree) ; print(pdtree)
table_mat <- table(Cross$classe, val_DT)
library(RColorBrewer);library(rattle)
fancyRpartPlot(Dtree)
accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
print(paste('Accuracy for test', accuracy_Test)) #Accuracy : 0.7488
Cross$classe = as.factor(Cross$classe)
confusionMatrix(val_DT,Cross$classe)
```

【Final modeling, Random Forest】

```
modFit <- randomForest(Train$classe~., data = Train, importance=T) #Please, take
time. It may take a few minutes:) #
modFit
pred1 <-predict(modFit,Train)
table(pred1) ; table(Train$classe)
confusionMatrix(pred1,Train$classe)
#the results on the validation set
Cross$classe = as.factor(Cross$classe)
valid <- predict(modFit, newdata=Cross)
confusionMatrix(valid,Cross$classe) #Accuracy : 0.9951
```

【The results on the validation and the test set】

```
#the results on the validation set
Cross$classe = as.factor(Cross$classe)
valid <- predict(modFit, newdata=Cross)
confusionMatrix(valid,Cross$classe) #Accuracy : 0.9951
# the results on the test set
test <- predict(modFit, newdata=testdata2)
print("Classification"); test
## [1] "Classification"
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
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
```

[Importance of the variables]

```
varImp <- importance(modFit)
head(varImp)
```

Including Plots

```
varImpPlot(modFit)
```

In conclusion, the Random Forest model showed very high accuracy, able to predict the answers to the Coursera Quiz 4 questions with 100% accuracy. The variables of yaw_belt, rell_belt, and pitch_belt were among the variables identified as the most essential as seen by the plots.

