Practical Machine Learning: Prediction Assignment

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 will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. The goal of the project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. This report describes how I built my model, including many details. I will also use my prediction model to predict 20 different test cases.

To begin, I load in the data. Glancing at a summary of the data, I notice there are some columns with very little data, and some columns that are not relevant to the task at hand. I clean the data accordingly.

```
# Load in the data
TrainingData<-read.csv('pml-training.csv',header=TRUE,na.strings=c("NA","#DIV/0!",""))
TestingData<-read.csv('pml-testing.csv',header=TRUE,na.strings=c("NA","#DIV/0!",""))

# Remove columns with blank or NA data
TrainingData<-TrainingData[,colSums(is.na(TrainingData))==0]
TestingData<-TestingData[,colSums(is.na(TestingData))==0]

# Remove columns containing time stamps or index data
TrainingData<-TrainingData[,c(2,8:60)]
TestingData<-TestingData[,c(2,8:60)]
```

We can partition the data so that we have a validation set in addition to a training set. This allows us to check our model for over-fit before incorporating our testing set.

```
# Split into training and validation sets
set.seed(1)
TrainPartition<-createDataPartition(TrainingData$classe,p=0.8,list=FALSE)
TrainingPart<-TrainingData[TrainPartition,]
ValidationPart<-TrainingData[-TrainPartition,]
```

Finally we use a random forest with built-in cross validation. We train the forest on our training partition. We then validate the model on our validation set. Lastly, we generate predictions on our test set.

```
# Program a random forest model with cross validation and confirm on validation data
CrossVal<-trainControl(method='cv',number=2)
RandFor<-
train(classe~.,data=TrainingPart,method="rf",trControl=CrossVal,verbose=FALSE)
RandForValidate<-predict(RandFor,newdata=ValidationPart)
RandForValidResult<-confusionMatrix(ValidationPart$classe,RandForValidate)
RandForValidResult
```

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1114    1    0    0    1
         B    4  753    2    0    0
         C    0    2  679    3    0
         D    0    0    6  635    2
         E    0    0    0    0  721

Overall Statistics

               Accuracy : 0.9946
                 95% CI : (0.9918, 0.9967)
    No Information Rate : 0.285
    P-Value [Acc > NIR] : < 2.2e-16

# Finally predict the results for the testing set
RandForTest<-predict(RandFor,newdata=TestingData)
RandForTest

[1] B A B A A E D B A A B C B A E E A B B B
```

The model validation suggests accuracy greater than 99% for this model. I believe these results to be good enough to accept the predictions for the final test.