Practical Machine Learning Assignment: Writeup

1. Synopsis

The goal of this project is to predict the manner in which people did the exercise, includes:

- Create a report describing how to built the model;
- how to use cross validation;
- what I think the expected out of sample error is;
- why I made these choices;
- Finally use the prediction model to predict 20 different test cases.

2. Load the Caret Library and Set the Random Number Generator's Seed

The purpose of setting the random number generator's seed is to ensure reproducibility.

```
library(caret)
```

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

```
set.seed(1234)
```

3. Pre-Processing of Data

```
rawData <- read.csv("pml-training.csv", na.strings=c("NA",""), strip.white=T)
Totalna <- apply(rawData, 2, function(x) { sum(is.na(x)) })
cleandata <- subset(rawData[, which(Totalna == 0)], select=c(roll_belt, pitch_forearm, yaw_belt, magnet
```

4. Partition

Set the training/testing partition using the training data set.

```
inTrain <- createDataPartition(cleandata$classe, p=0.7, list=F)
training <- cleandata[inTrain,]
testing <- cleandata[-inTrain,]
```

5. Learning the Clasification Hypothesis using the Training Data

5.1 Training a Random Forest model.

```
#Fitting the Model
ctrl <- trainControl(allowParallel=T, method="cv", number=4)
modFit<-train(classe~.,data=training,method="rf",trControl=ctrl)
```

```
## Loading required package: randomForest
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
#Visualizing the Model Results
modFit
```

```
## Random Forest
##
## 13737 samples
##    10 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
##
## Summary of sample sizes: 10302, 10302, 10304, 10303
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy  Kappa  Accuracy SD  Kappa SD
##    2    0.980     0.975  0.00188      0.00239
##    6    0.981     0.977  0.00246      0.00311
##   10    0.979     0.973  0.00342      0.00433
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 6.
```

5.2 Confusion matrix.

```
predictions<-predict(modFit,newdata=testing)
confusionMatrix(predictions,testing$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1662   13    1    0    0
##          B    2 1099   10    4    7
##          C    9   11 1008    2    5
##          D    1   13    7  956    4
##          E    0    3    0    2 1066
##
## Overall Statistics
##
##                Accuracy : 0.984
##                  95% CI : (0.9805, 0.9871)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9798
##  Mcnemar's Test P-Value : NA
```

```
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9928   0.9649   0.9825   0.9917   0.9852
## Specificity            0.9967   0.9952   0.9944   0.9949   0.9990
## Pos Pred Value         0.9916   0.9795   0.9739   0.9745   0.9953
## Neg Pred Value         0.9971   0.9916   0.9963   0.9984   0.9967
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2824   0.1867   0.1713   0.1624   0.1811
## Detection Prevalence   0.2848   0.1907   0.1759   0.1667   0.1820
## Balanced Accuracy      0.9948   0.9800   0.9884   0.9933   0.9921
```

Its accuracy on the test set is 98.4%

6. Predictions

```
rawdata2 <- read.csv("pml-testing.csv", na.strings=c("NA",""), strip.white=T)
predictions2 <- predict(modFit, newdata=rawdata2)
predictions2
```

```
##  [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
```