Practical Machine Learning Write Up

This report is using Human Activity Recognition (HAR) data. The report attempts to predict the Classe variable, which is how the those involved in collecting the HAR data exercised.

First step is to pre process data by creating traning and test sets and removing columns with NA values.

```
setwd("~/Data Science/Practical Machine Learning")
library(caret)
```

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

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.1.3
```

```
Train<-read.csv("pml-training.csv", header=T, na.strings=c("NA", "#DIV/0!"))
Test<-read.csv("pml-testing.csv", header=T, na.string=c("NA", "#DIV/0!"))

processedTrain<-Train[, apply(Train, 2, function(x) !any(is.na(x)))]
cleanTrain<-processedTrain[,-c(1:8)]
cleanTest<-Test[,names(cleanTrain[,-52])]
```

The data was broken up into two chunks: 75% went into a traning set and the rest of the 25% went into the test set.

```
inTrain<-createDataPartition(y=cleanTrain$classe, p=0.75,list=F)
training<-cleanTrain[inTrain,]
test<-cleanTrain[-inTrain,]
```

Using the Random Forest Trees algorithm on the training set, the estimated error of prediction and accuracy of the data was predicted using 51 predictors. Accuracy was predicted at 99.2% with a 95% confidence interval (0.989-0.994) with a 0.99 Kappa value.

```
fit<-trainControl(method="cv", number=5, allowParallel=T, verbose=T)
fit2<-train(classe~.,data=training, method="rf", trControl=fit, verbose=F)
```

```
## Loading required package: randomForest
```

```
## Warning: package 'randomForest' was built under R version 3.1.3
```

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

```
## + Fold1: mtry= 2
## - Fold1: mtry= 2
## + Fold1: mtry=26
## - Fold1: mtry=26
## + Fold1: mtry=51
## - Fold1: mtry=51
## + Fold2: mtry= 2
## - Fold2: mtry= 2
## + Fold2: mtry=26
## - Fold2: mtry=26
## + Fold2: mtry=51
## - Fold2: mtry=51
## + Fold3: mtry= 2
## - Fold3: mtry= 2
## + Fold3: mtry=26
## - Fold3: mtry=26
## + Fold3: mtry=51
## - Fold3: mtry=51
## + Fold4: mtry= 2
## - Fold4: mtry= 2
## + Fold4: mtry=26
## - Fold4: mtry=26
## + Fold4: mtry=51
## - Fold4: mtry=51
## + Fold5: mtry= 2
## - Fold5: mtry= 2
## + Fold5: mtry=26
## - Fold5: mtry=26
## + Fold5: mtry=51
## - Fold5: mtry=51
## Aggregating results
## Selecting tuning parameters
## Fitting mtry = 26 on full training set
```

```
predicttest<-predict(fit2, newdata=test)
confusionMatrix(predicttest, test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    8    0    0    0
##          B    0  940    5    1    0
##          C    0    1  845   12    0
##          D    0    0    5  790    2
##          E    0    0    0    1  899
##
## Overall Statistics
##
##                Accuracy : 0.9929
##                  95% CI : (0.9901, 0.995)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.991
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9905   0.9883   0.9826   0.9978
## Specificity            0.9977   0.9985   0.9968   0.9983   0.9998
## Pos Pred Value         0.9943   0.9937   0.9848   0.9912   0.9989
## Neg Pred Value         1.0000   0.9977   0.9975   0.9966   0.9995
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2845   0.1917   0.1723   0.1611   0.1833
## Detection Prevalence   0.2861   0.1929   0.1750   0.1625   0.1835
## Balanced Accuracy      0.9989   0.9945   0.9925   0.9904   0.9988
```

```
testCases<-predict(fit2, newdata=cleanTest)
testCases
```

```
##  [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
```

Upon completion of prediction algorithm for 20 test cases, the function below creates single text files to upload to coursera.

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}
pml_write_files(testCases)
```