Practical Machine Learning: Course Project Report

```
Data for this project comes from: "http://groupware.les.inf.puc-rio.br/har"
```

Synopsis:

The goal of this project is to use the data from accelerometers on the belt,forearm, arm and dumbbells of 6 participants and to quantify how well they did on their exercises. We will attempt to achieve this goal by training a prediction model on the accelerometer data. After data analysis and selection of variables, we will select an algorithm to be used for building our prediction model. The prediction model will be rerun on the test data to predict the outcome of 20 different tests. The following sections of this report will attempt to describe:

```
- the reason for our selections.
- how we build our model.
- how the cross validation was used.
- what could be the expected out of sample error.
```

Data Analysis and Variable Selection:

We start by loading the required R libraries and read in the provided data as R data sets. Then we evaluate and select our training and testing subsets.

```
library(AppliedPredictiveModeling)
library(caret)
library(rattle)
library(rpart.plot)
library(randomForest)
inTest <- read.csv("pml-testing.csv")
inTrain <- read.csv("pml-training.csv")
```

As we want to be able to estimate the out of sample error, we randomly split the large training data set into a smaller training set, subTrain1, and a validation set, subTrain2.

```
set.seed(1234)
subTrain <- createDataPartition(y=inTrain$classe, p=0.6, list=F)
subTrain1 <- inTrain[subTrain,]
subTrain2 <- inTrain[-subTrain,]
```

There seems to be a large number of NAs, with nearly zero variance. Some of the variables seem to be metadata and not useful for predictive analysis. So we clean up our working set by removing them.

```
# remove near zeros
nzVar <- nearZeroVar(inTrain)
subTrain1 <- subTrain1[, -nzVar]
subTrain2 <- subTrain2[, -nzVar]
# remove NAs
xNA <- sapply(subTrain1, function(x) mean(is.na(x))) > 0.95
subTrain1 <- subTrain1[,xNA==F]
subTrain2 <- subTrain2[,xNA==F]
```

```
# remove non-usable variables
subTrain1 <- subTrain1[, -(1:5)]
subTrain2 <- subTrain2[, -(1:5)]
```

Algorithm for Model Building:

After cleaning the data we are left with a fairly large set of data which seems to be a good candidate for fitting a prediction model using Random Forest classification with significant measure of fit being accuracy. We will fit the model on subTrain1 and call train(),to use 3-way cross validation which will select the optimally tuned parameters for our model.

```
# calling train to use 3-fold CV to use optimal tuning parameters.
fitCntl <- trainControl(method = "cv", number=3, verboseIter=F)
# calling train to fit the model on subTrain1
fit <- train(classe ~., data=subTrain1, method="rf", trControl=fitCntl)
# print the final model to see tunning parameters it selected.
fit$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.33%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3346    2    0    0    0 0.0005973716
## B    9 2268    1    1    0 0.0048266784
## C    0    7 2046    1    0 0.0038948393
## D    0    0   10 1919    1 0.0056994819
## E    0    1    1    5 2158 0.0032332564
```

We see that total of 500 trees and 27 variables were selected at each split.

Model Selection and Validation:

Here we use our fitted model to predict the accuracy of the variable "classe" in subTrain2 and show the confusion matrix to compare the predicted versus actual statistics.

```
predx <- predict(fit, newdata=subTrain2)
confusionMatrix(subTrain2$classe,predx)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2232    0    0    0    0
##          B    4 1512    2    0    0
##          C    0    3 1364    1    0
```

```
##          D    0    0    5 1281    0
##          E    0    3    0    2 1437
##
## Overall Statistics
##
##                Accuracy : 0.9975
##                  95% CI : (0.9961, 0.9984)
##     No Information Rate : 0.285
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9968
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9982   0.9960   0.9949   0.9977   1.0000
## Specificity            1.0000   0.9991   0.9994   0.9992   0.9992
## Pos Pred Value         1.0000   0.9960   0.9971   0.9961   0.9965
## Neg Pred Value         0.9993   0.9991   0.9989   0.9995   1.0000
## Prevalence             0.2850   0.1935   0.1747   0.1637   0.1832
## Detection Rate         0.2845   0.1927   0.1738   0.1633   0.1832
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9991   0.9975   0.9971   0.9985   0.9996
```

We see the accuracy rate is 99.7% thus accuracy for the out of sample error is 0.3%. This is a pretty good statistic. It encourages us to use Random Forest method on the test set as well. So we prep the test set by repeating the same transformations as we did on the training set.

```
# remove near zeros
nzVar <- nearZeroVar(inTrain)
inTrain <- inTrain[, -nzVar]
inTest <- inTest[, -nzVar]
# remove NAs
xNA <- sapply(inTrain, function(x) mean(is.na(x))) > 0.95
inTrain <- inTrain[,xNA==F]
inTest <- inTest[, xNA==F]
# remove non-usable variables
inTrain <- inTrain[, -(1:5)]
inTest <- inTest[, -(1:5)]
# Now we re-fit the model using full trainig set
fitCntl <- trainControl(method="cv", number=3, verboseIter=F)
fit <- train(classe ~., data=inTrain, method="rf", trControl=fitCntl)
```

Test Set Predictions:

In this section, we fit the model on inTrain to predict accuracy rate for "classe" in inTest and write out predicted outcomes as a character vectors to individual files:

```
predx <- predict(fit,newdata=inTest)
predx <- as.character(predx)
write_pml_files <- function(x){
      n <- length(x)
```

```
    for (i in 1:n){
        filename <- paste0("problem_id_", i, ".txt")
        write.table(x[i], file=filename, quote=F,row.names=F,col.names=F)
    }
}
  write_pml_files(predx)
```