In my analysis regarding the quantified self movement data yielded a relatively simple model that provided fantastic results. I will describe the data transformations implemented, packages and parameters used, as well as expected sample error and cross validation.

Data Transformation

The first step I took was to set all blank vlaues to NA, then remove columns of all NA values,and did so using the below code.

```
train2 = pml.training
train2[pml.training==""]=NA
train = train2[,colSums(is.na(train2))<.8*nrow(train2)]
```

This highlighed any columns that contained 80% NA values and removed them. Due to the large number of observations and predictor variables, I was aggressive in this manner. I also removed the X variable , timestamps (raw, cvtd) as well as new_window and num_window. This left 53 feature variables remaining.

Packages and Parameters

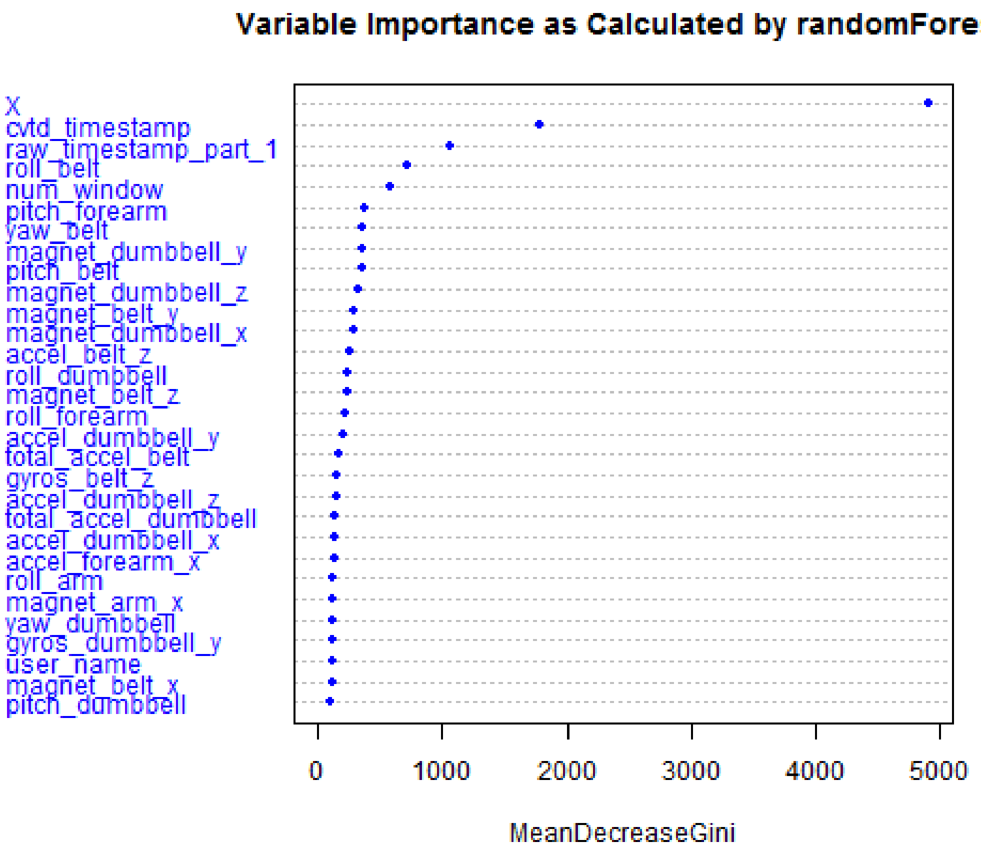
My initial use of the caret package proved frustrating due to very long run times and I looked elsewhere, eventually using randomForest. The number of trees used in the voting was 500. This initial model provided very high accuracy of 0.9972.

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

```
library("randomForest")
RFfit = randomForest(classe~., data=train, ntree=500)
accuracy = sum(RFfit$predicted==train$classe)/nrow(train)
print(accuracy)
```

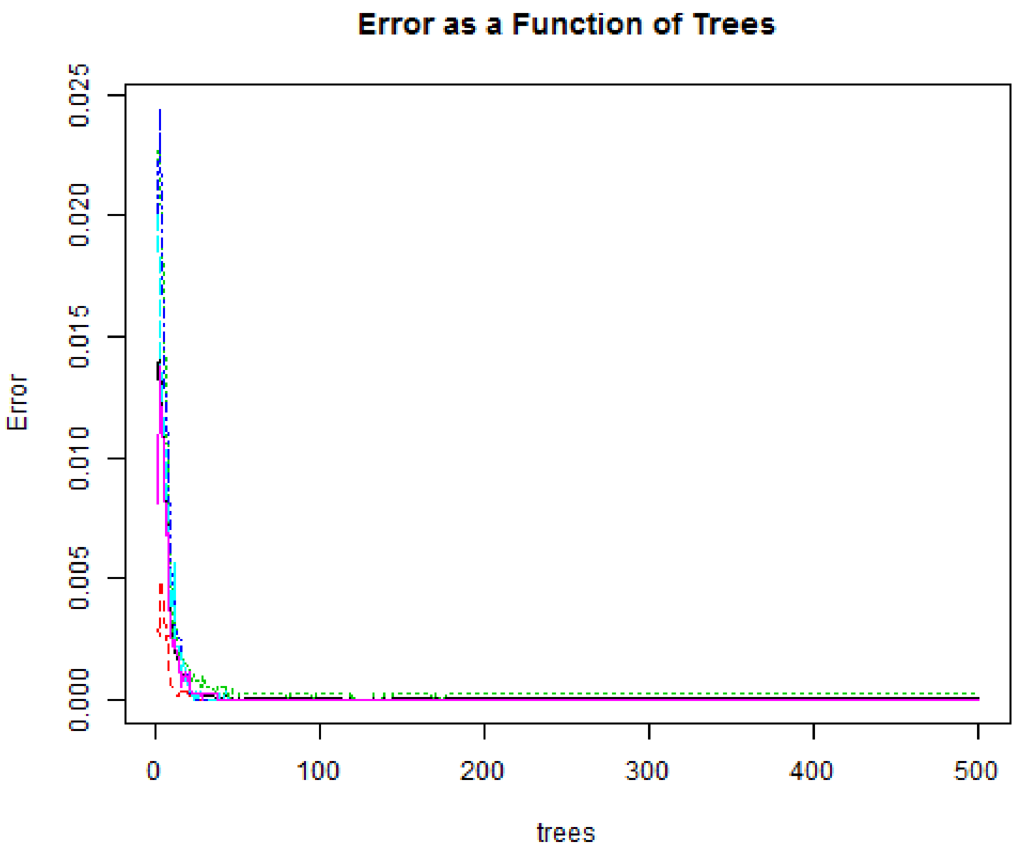
```
## [1] 0.999949
```

This high accuracy led to concerns of over-fitting. Thus, I looked at the Importance of each predictor as computed by randomForest. I then removed the "less" important variables and refir the model. This made very little difference in accuracy and predicted test results.



Expected Sample Error and Cross Validation

Since the in-sample error was very low, I had a small lower bound on the out of sample error of 0.28%.



The package randomForest lets you look at average cross validation error with different numbers of predictor variables. I chose a 4-fold cross validation (mainly because computing time).

```
rfcv.fit = rfcv(train[,1:53],train[,54],cv.fold = 4)
rfcv.fit$error.cv
```

```
##         53         26         13          7          3          1
##   933.6774   775.3896  1069.8067  3034.7335  9426.3456 16634.6613
```

Thus, randomForest provided a great model with minimal tweaking of parameters. (it also correctly predicted all 20 of the test cases!)