Course-Project

Introduction

The goal of this project is to use the HAR data set provided in the assignment to train and test a machine learning model to predict the type of activity performed. First the data is loaded, cleaned, and explored. Next, three candidate models are trained, and their in-sample error rates are compared to determine the best predictor. Finally, the best predictor is applied to the test set.

```
library(caret)
```

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

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

Pre-Analysis Preparation

Loading the Data

Read the data straight into R from the links given in the assignment.

```
train <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv")
test <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv")
```

Cleaning and Pre-processing

First, check for for NA values and blank character values, since they need to be dealt with before most machine learning tools can be applied.

```
checkContent <- function(x) {
  if (is.character(x)) {
    return(mean(x==""))
  } else {
    return(mean(is.na(x)))
  }
}
table(sapply(train, checkContent))
```

```
##
##                  0 0.979308938946081
##                 60                 100
```

Since some fields are almost entirely NA or blank but most fields have no NA or blank values, simply remove all columns in which an NA or blank value is found. Little information is lost, and imputing these values would be unreliable.

```
train <- train[, sapply(train, function(x) checkContent(x)==0)]
```

Next, check the types of the features remaining, and look at the head of the character variables to see if they still hold any useful information.

```
table(sapply(train, class))
```

```
##
## character   integer   numeric
##         4        29        27
```

```
head(train[, sapply(train, function(x) !is.numeric(x))], n=3)
```

```
##   user_name   cvtd_timestamp new_window classe
## 1  carlitos 05/12/2011 11:23         no      A
## 2  carlitos 05/12/2011 11:23         no      A
## 3  carlitos 05/12/2011 11:23         no      A
```

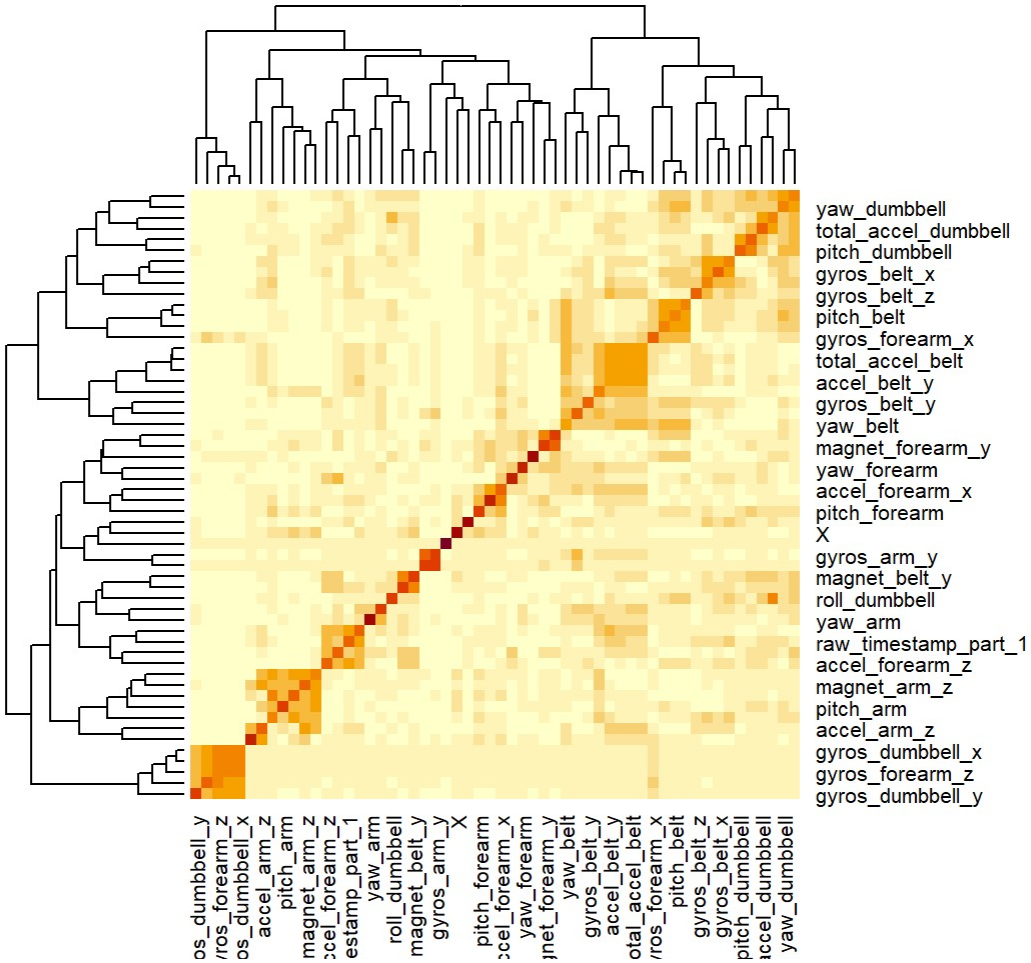
There's the "classe" feature to be predicted, which is extremely important, but the others (user_name, cvtd_timestamp, new_window) seem unlikely to be of predictive value. Remove from data set.

```
train <- train[, -which(names(train) %in% c("user_name", "cvtd_timestamp", "new_windo
w"))]
```

Exploratory Analysis

Examine the correlation matrix for the predictor features, which are now all numeric, to check for patterns or major groups of correlated variables.

```
heatmap(abs(cor(train[, sapply(train, is.numeric)])))
```



There are some small, somewhat correlated groups, but no major trends.

Modeling

Training

This is a supervised classification problem, so candidates include methods such as random forest, linear discriminant, and gradient boosting models. Train candidate models using 10-fold cross-validation to further improve accuracy, and excluding the “X” feature, which simply indexes all entries.

Broken up into separate chunks because oh boy does this take a while.

```
trctrl <- trainControl(method="cv", number=10)
rf <- train(classe~.-X, data=train, method="rf", trControl=trctrl)
```

```
lda <- train(classe~.-X, data=train, method="lda", trControl=trctrl)
```

```
gbm <- train(classe~.-X, data=train, method="gbm", trControl=trctrl)
```

In Sample Accuracy

Check in-sample accuracy rates, to determine which candidate model is best.

```
confusionMatrix(predict(rf, train), factor(train$classe))$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      1.0000000      1.0000000      0.9998120      1.0000000      0.2843747
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

```
confusionMatrix(predict(lda, train), factor(train$classe))$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   7.196004e-01   6.450737e-01   7.132577e-01   7.258781e-01   2.843747e-01
## AccuracyPValue  McnemarPValue
##   0.000000e+00  3.613112e-185
```

```
confusionMatrix(predict(gbm, train), factor(train$classe))$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.9987769      0.9984529      0.9981806      0.9992162      0.2843747
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

Testing

Test Cases

Use the most accurate model to predict the “classe” for the test cases in the “test” dataset. Given how high the in-sample accuracy is, it seems possible that the model is somewhat overtrained. Out-of-sample accuracy is also usually lower in general, so 90-95% seems like a reasonable estimate.

```
predict(rf, test)
```

```
##  [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
```