Machine Learning Assignment

Emily Jones

Project Goal

I will apply Machine Learning techniques to predict the manner in which a study participant performed weight lifting exercises.

I am using data from an existing study. Citation for this dataset is

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human ’13) . Stuttgart, Germany: ACM SIGCHI, 2013.

Read more: http://groupware.les.inf.puc-rio.br/har#ixzz3HCvitx6t (http://groupware.les.inf.puc-rio.br/har#ixzz3HCvitx6t)

Data Processing

First load the data, and any libraries needed for analysis

```
library(caret)
```

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

```
library(AppliedPredictiveModeling)
library(ggplot2)
library(randomForest)
```

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

```
library(RCurl)
```

```
## Loading required package: bitops
```

```
#myCsv <- getURL("https://gist.github.com/raw/667867/c47ec2d72801cfd84c6320e1fe37055ffe600c87/test.csv")
#WhatJDwants <- read.csv(textConnection(myCsv))

#csvtest<-getUrL("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv")
#pml.testing <- read.csv(textConnection(csvtest))
pml.testing <- read.csv(url("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"))
pml.training <- read.csv(url("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"))
```

Reduce the variables I am looking at.

Remove

- non-numeic columns
- columns with missing data
- first 7 columns with timesstamp and other data not pertinent

Add the classe column back in (It is at column 160)

```
# Table is sparse, choose only numeric with no missing values, exclude for 7 columns with datestamps and # windows. They will not affect the result

nums <- sapply(pml.training, is.numeric)
miss<-sapply(pml.training,function(x) any(is.na(x)))

which(colnames(training)=="classe")
```

```
## [1] 54
```

```
cols<-nums & !miss
cols[160]<-TRUE
cols[1:7]<-FALSE

# subtrain is now the columns from pml.training I am interested
subtrain <-pml.training[,cols]
```

Fitting a model

I used a Random Tree Model on the remaining columns. Random Forests do well with prediction and give estimates of what variables are of most importance. They are also not sensitive to overfitting

Create training and testing subsets from reduced training data

In order to see how accurately the proposed model will generalize to the independent testing data set (cross-validation), the data is partitioned into a training subset and validation subset, using a 75/25 split.

I fit the Random Forest model against the training subset

```
inTrain = createDataPartition(subtrain$classe, p = 3/4)[[1]]
training = subtrain[ inTrain,]
testing = subtrain[-inTrain,]

fit <- randomForest(training$classe ~ ., data=training, importance = TRUE)

print(fit)
```

```
##
## Call:
##  randomForest(formula = training$classe ~ ., data = training,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.43%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 4180    3    1    0    1    0.001195
## B   13 2833    2    0    0    0.005267
## C    0   14 2553    0    0    0.005454
## D    0    0   25 2386    1    0.010779
## E    0    0    1    3 2702    0.001478
```

Evaluate model against the testing subset

```
confusionMatrix(testing$classe,predict(fit,testing))
```

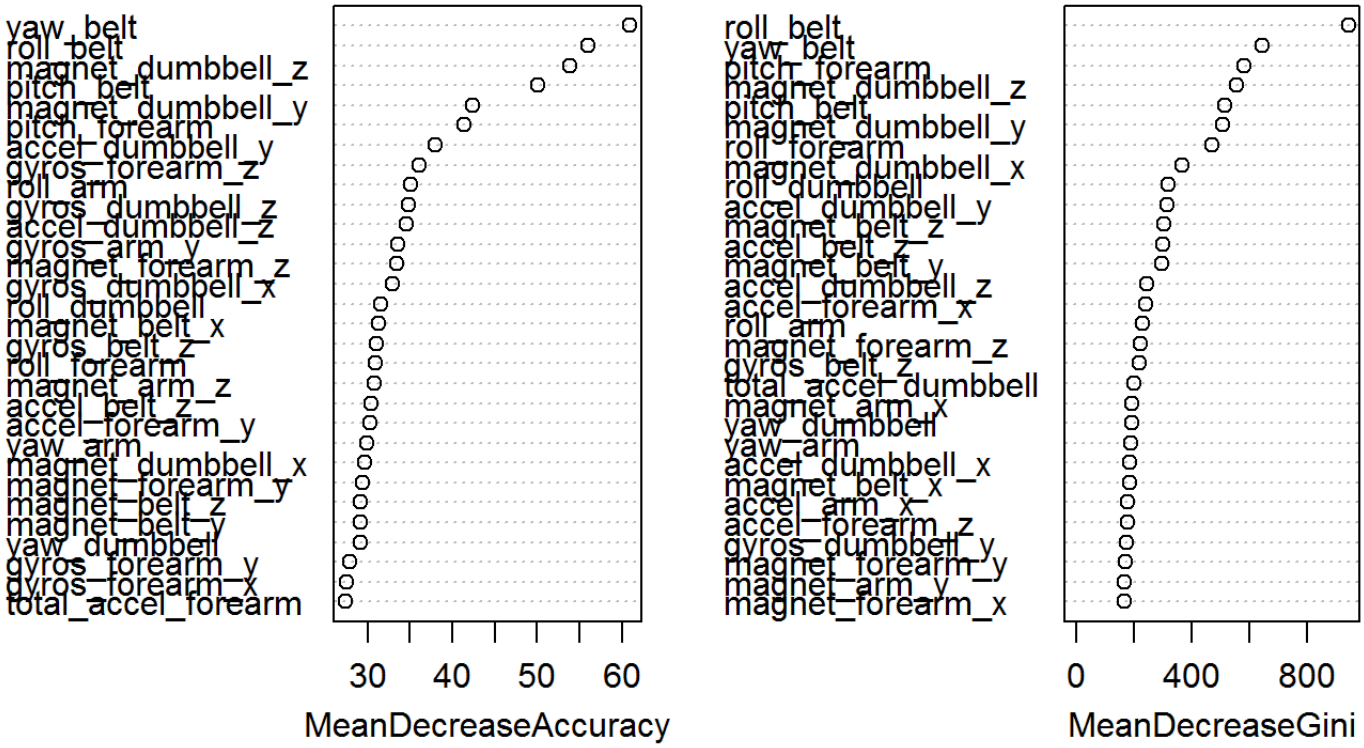
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    0    0    0    0
##          B    2  943    4    0    0
##          C    0    0  855    0    0
##          D    0    0    7  797    0
##          E    0    0    0    1  900
##
## Overall Statistics
##
##                Accuracy : 0.997
##                  95% CI : (0.995, 0.998)
##     No Information Rate : 0.285
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.996
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.999    1.000    0.987    0.999    1.000
## Specificity             1.000    0.998    1.000    0.998    1.000
## Pos Pred Value          1.000    0.994    1.000    0.991    0.999
## Neg Pred Value          0.999    1.000    0.997    1.000    1.000
## Prevalence              0.285    0.192    0.177    0.163    0.184
## Detection Rate          0.284    0.192    0.174    0.163    0.184
## Detection Prevalence    0.284    0.194    0.174    0.164    0.184
## Balanced Accuracy       0.999    0.999    0.994    0.999    1.000
```

Interpretation of the Model

The model seems to be quite accurate. The out of sample error rate is .49% (see OOB estimate of error rate above) and the accuracy of the testing subset (99.5%)

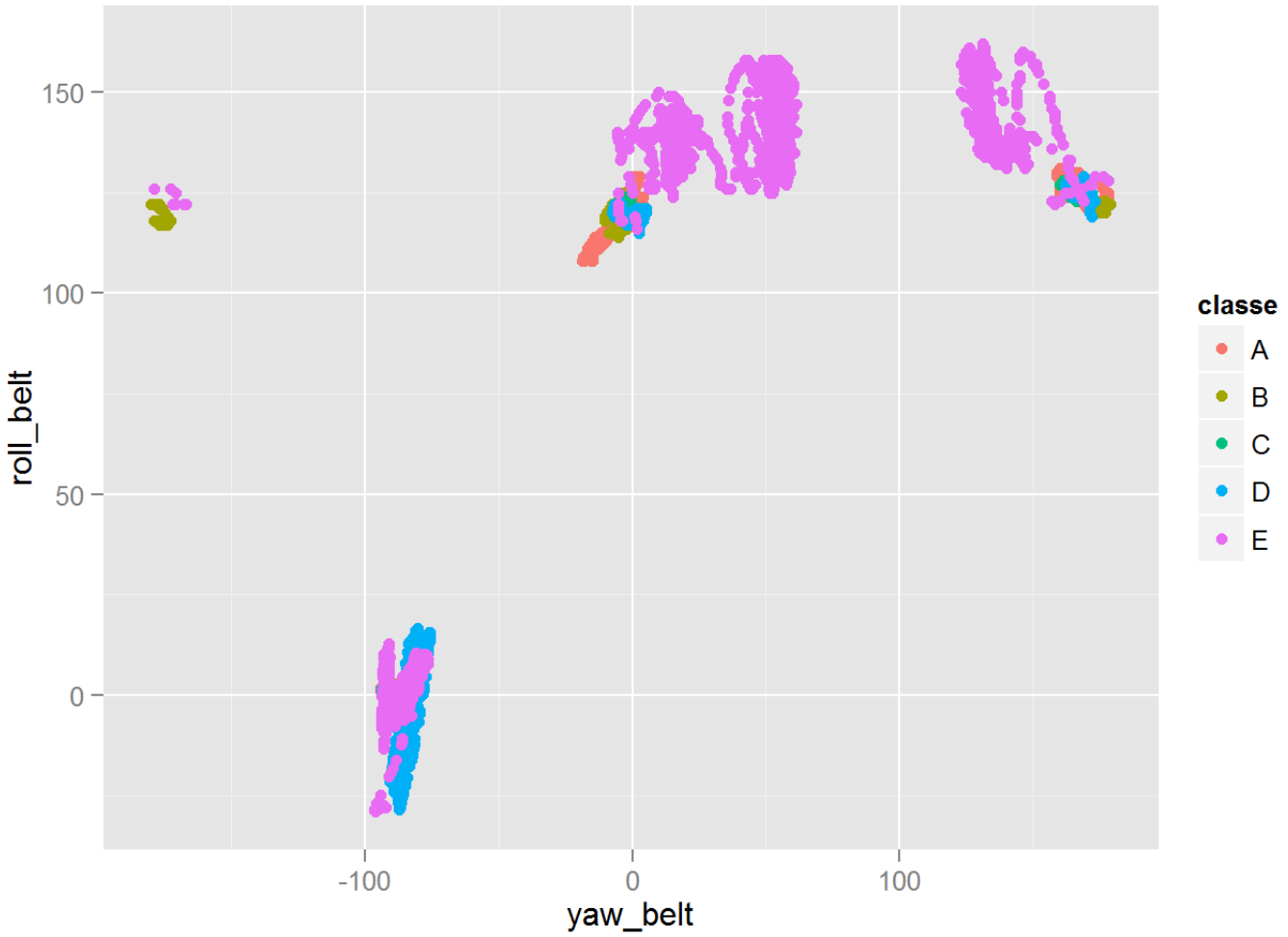
Now use Random Tree tools to evaluate most influential factor.

```
varImpPlot(fit)
```

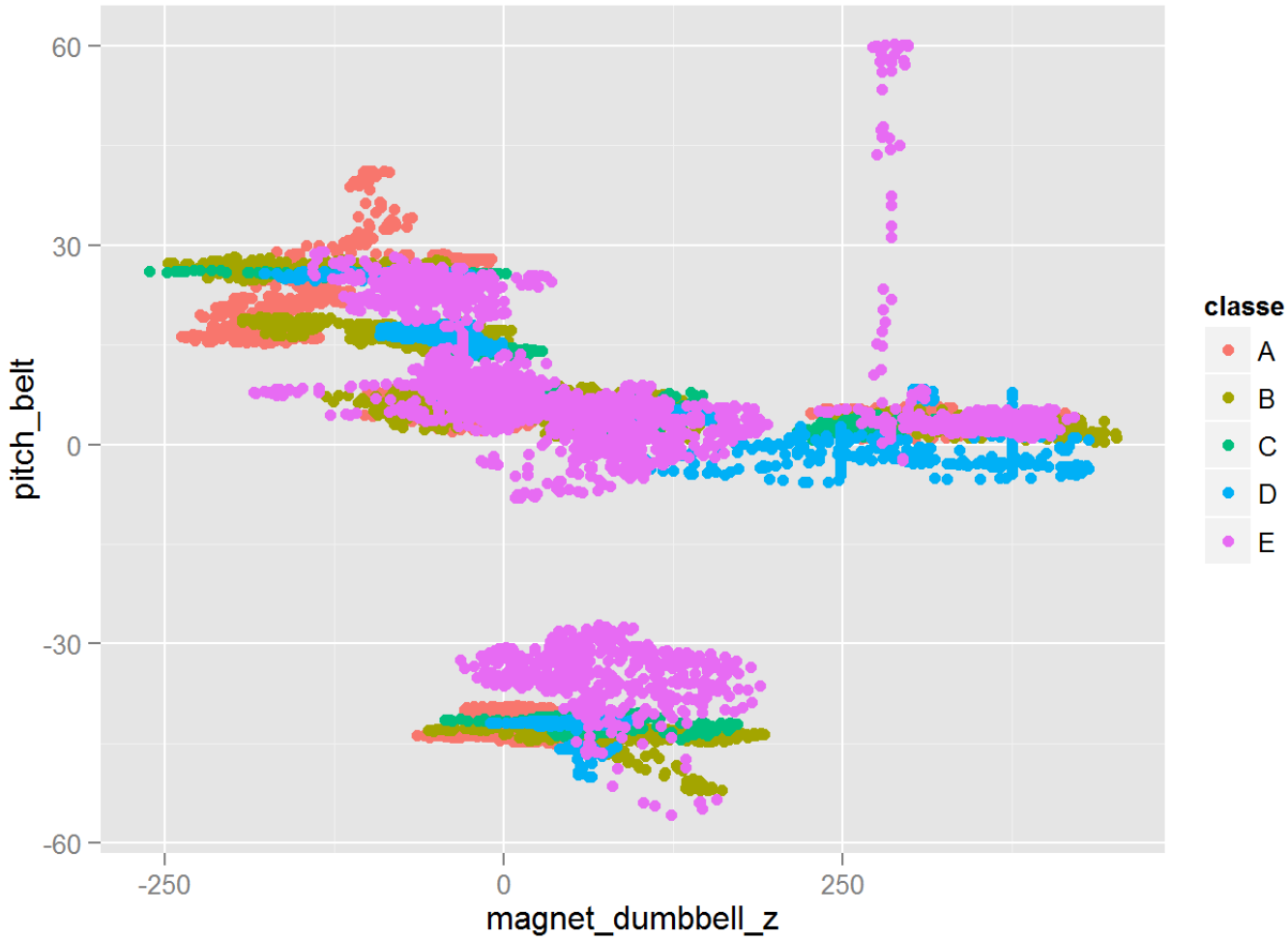


The top factors seem to be roll_belt, yaw_belt, pitch_forearm adn magnet_dumbell_z

```
qplot(yaw_belt,roll_belt,colour=classe,data=training)
```




```
qplot(magnet_dumbbell_z,pitch_belt,colour=classe,data=training)
```



I see a clear clustering of the response with these factors.

The coding for the responses: A-E is as follows:

Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions:

A: exactly according to the specification B: throwing the elbows to the front C: lifting the dumbbell only halfway D: lowering the dumbbell only halfway E: throwing the hips to the front

Make predictions against the test set

```
subtesting <- pml.testing[,cols]

answers = predict(fit, subtesting)

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(answers)
```