Machine Learning to Predict which Weight Lifting Exercise is Being Performed

LyndaFinn

Thursday, October 23, 2014

Data Processing

Load the data, add the necessary libraries, select columns for modeling

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. http://groupware.les.inf.puc-rio.br/public/papers/2013.Velloso.QAR-WLE.pdf (http://groupware.les.inf.puc-rio.br/public/papers/2013.Velloso.QAR-WLE.pdf)

```
## Add necessary libraries

library(AppliedPredictiveModeling)
library(caret)
```

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

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

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

```
## Read Data from web
y <- read.csv(url("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"))

## remove non-numeric variables and variables with missing values
## thes variables vary very little or are predominantly missing
nums <- sapply(y, is.numeric)
miss <- sapply(y,function(x) any(is.na(x)))
cols <- nums & !miss

##include classe (response variable)
cols[160]<-TRUE

## also remove case number and time stamp and num window
cols[1:7]<-FALSE

## subset data to only important predictor columns and the response
suby<-y[,cols]
```

Model Fitting

Use Random Tree Model on Remaining Explanatory Variables. Random Forests are good at prediction and tend not to overfit, while still giving some insight into what are the important factors

```
## partition data to training and testing
set.seed(666)
inTrain<-createDataPartition(suby$classe, p = 3/4)[[1]]
training<-suby[ inTrain,]
testing<-suby[-inTrain,]

## Fit Random Forest model to training set
modelFit<-randomForest(training$classe ~ ., data=training, importance = TRUE)
print(modelFit)
```

```
##
## 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: 0.49%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 4182    3    0    0    0   0.0007168
## B   12 2831    4    1    0   0.0059691
## C    0   11 2552    4    0   0.0058434
## D    0    0   23 2386    3   0.0107794
## E    0    0    1   10 2695   0.0040650
```

```
## Evaluate Model Perfomance on Testing Data
confusionMatrix(testing$classe,predict(modelFit,testing))
```

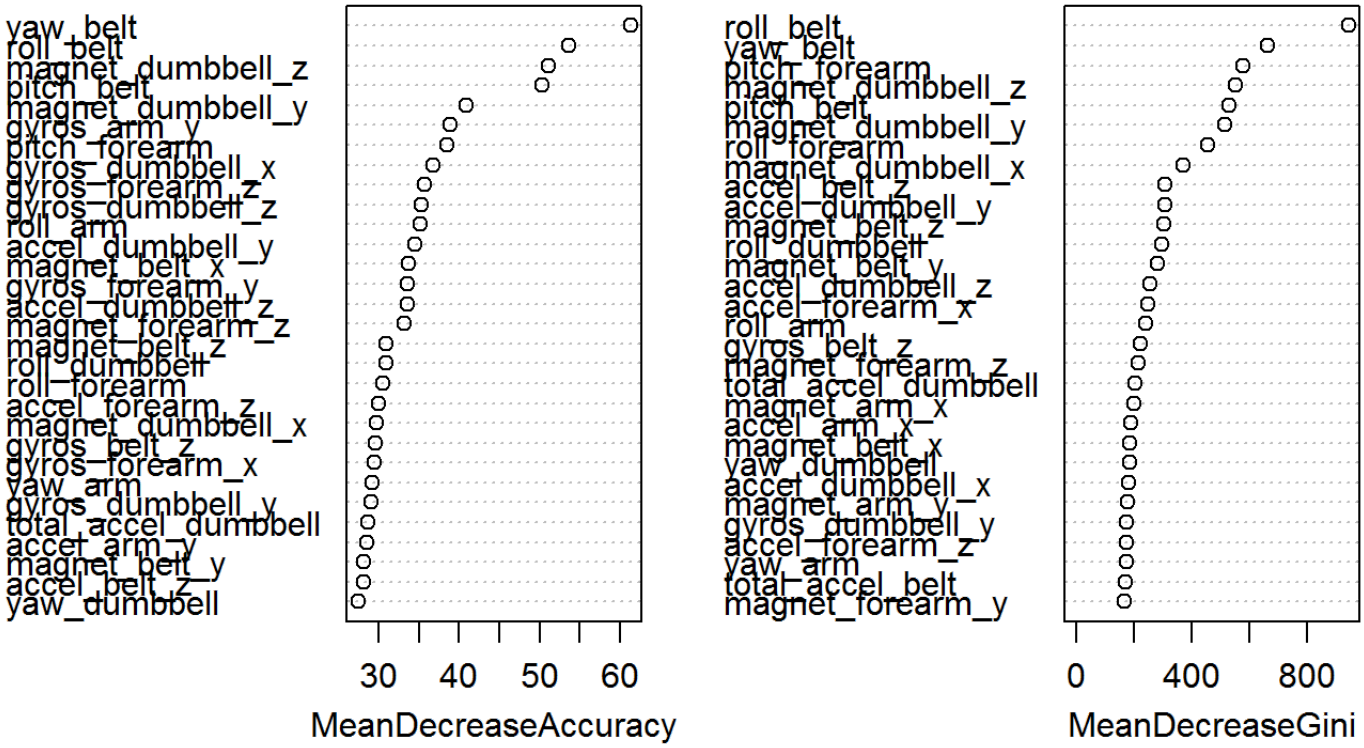
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1394    1    0    0    0
##          B    3  946    0    0    0
##          C    0    2  853    0    0
##          D    0    0    4  799    1
##          E    0    0    0    2  899
##
## Overall Statistics
##
##                Accuracy : 0.997
##                  95% CI : (0.995, 0.999)
##     No Information Rate : 0.285
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.997
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity             0.998    0.997    0.995    0.998    0.999
## Specificity             1.000    0.999    1.000    0.999    1.000
## Pos Pred Value          0.999    0.997    0.998    0.994    0.998
## Neg Pred Value          0.999    0.999    0.999    1.000    1.000
## Prevalence              0.285    0.194    0.175    0.163    0.184
## Detection Rate          0.284    0.193    0.174    0.163    0.183
## Detection Prevalence    0.284    0.194    0.174    0.164    0.184
## Balanced Accuracy       0.999    0.998    0.997    0.998    0.999
```

Model Interpretation

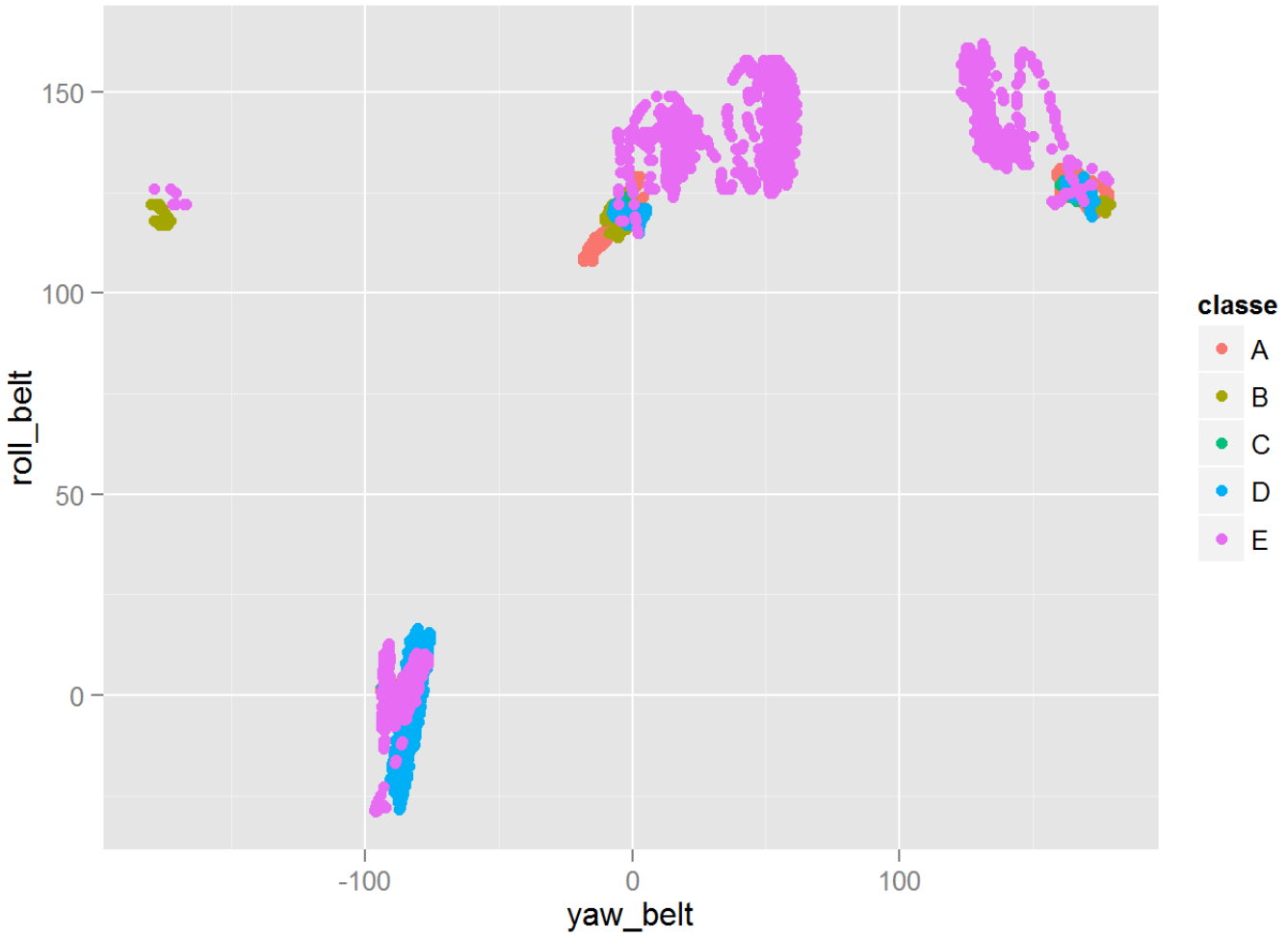
Model evaluation shows good accuracy. The out of sample error rate is .49% which is quite low. It is obtained within the Random Forest Algorithm through cross-validation, and matches closely the accuracy obtained from the testing sample (99.7%).

Next, Use Random Tree tools to determine most influential factors. Plot response against some of these factors

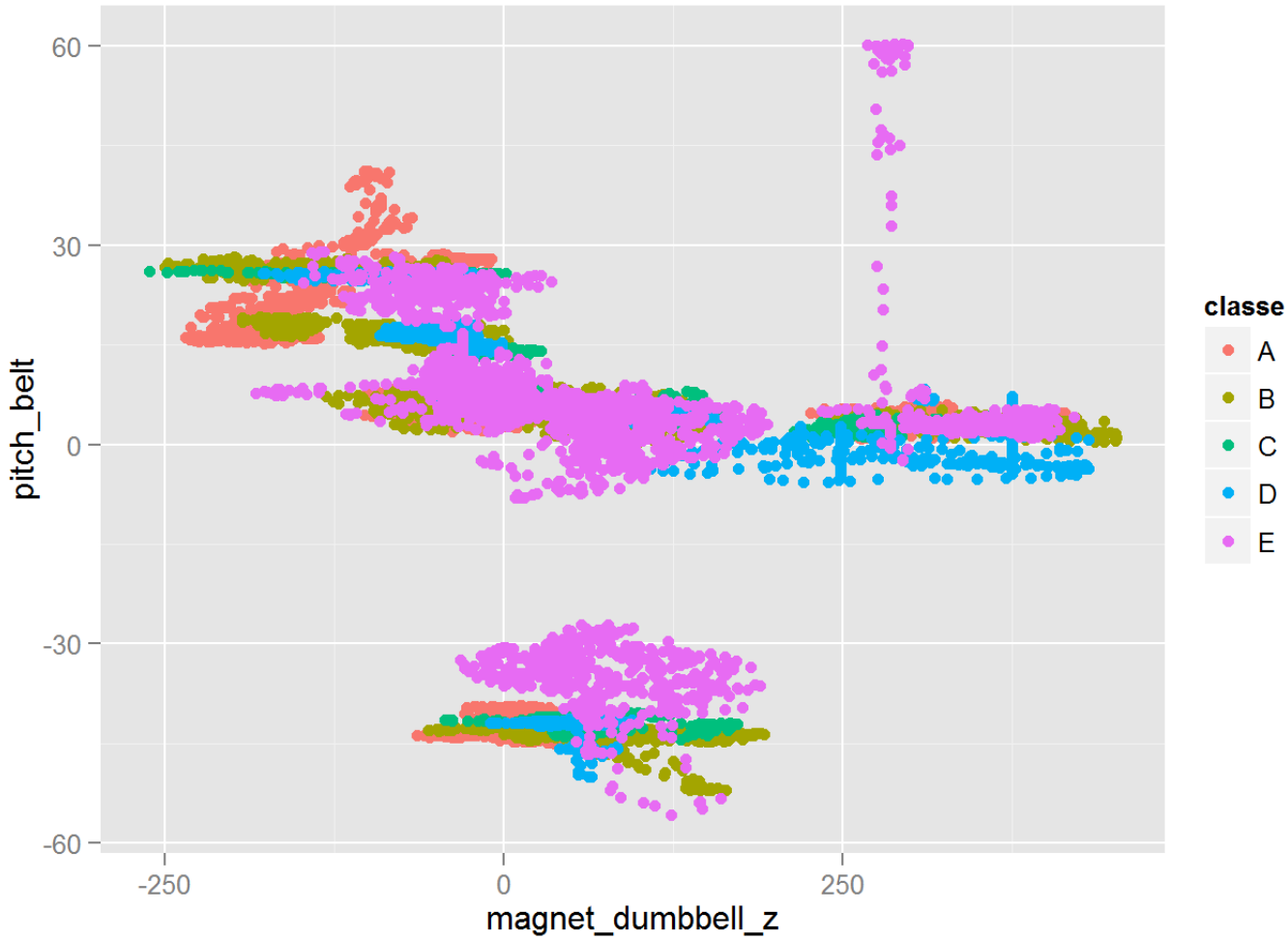
```
varImpPlot(modelFit)
```




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




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



Clearly the top four variables do help to visually cluster the response. The coding for 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

Generate Answer files

```
tt<- read.csv(url("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"))
subtt<-tt[,cols]
answers<-predict(modelFit,tt)
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)
```