Weight Lifting Project Code

Jennifre Richards

Friday, March 20, 2015

IN THE COMPILED HTML I reduced the number of trees in the random forest from 1000 to 10, because it took several hours to run the 1000 trees I used to build my random forest models; thus the results in this document are different from the results presented in the project write-up.

```
setwd('C:/Rrepos/Rdata/PML')
getwd()

#install.packages('caret', dependencies=c('Depends', 'Suggests'))
require(caret)
```

```
## Loading required package: caret
```

```
## Warning: package 'caret' was built under R version 3.1.3
```

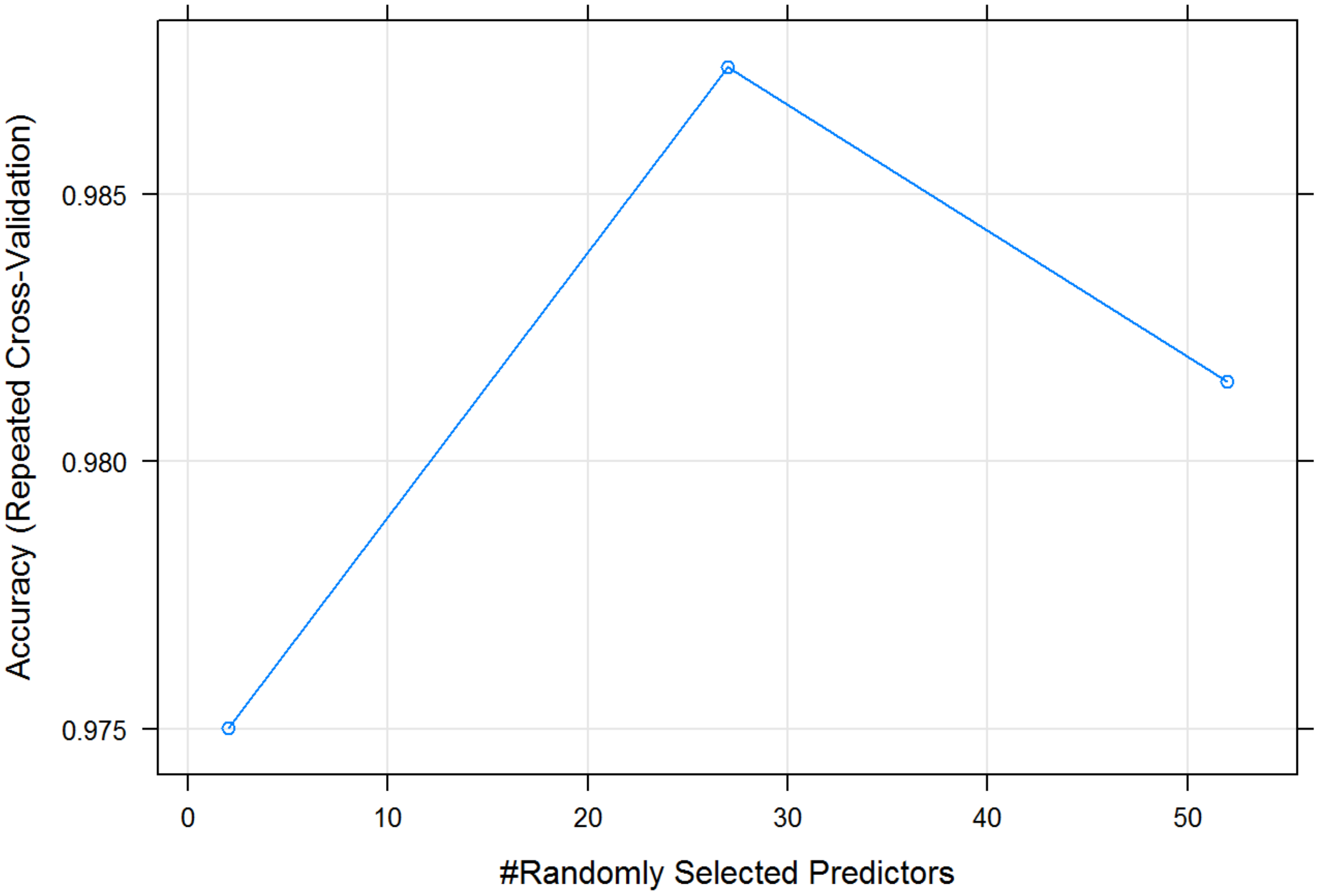
```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
# IN THE COMPILED HTML I reduced the number of trees in the random forest from
# 1000 to 10, because it took several hours to run the 1000 trees; thus the results
# in this document are different from the results presented in the project write-up.

# loading the data
test <- read.csv('C:/Rrepos/Rdata/PML/testing.csv', stringsAsFactors=T, header = T)
train <- read.csv('C:/Rrepos/Rdata/PML/training.csv', stringsAsFactors=T, header = T)

# preparing the data;since goal was to predict, eliminated data that did not seem relevan
t (some of initial
# columns, including individual names, and then data that had no values or NAs in the tes
ting dataset; this corresponded to similar but not
# completely equivalent data in the training set (i.e., were some values for kurtosis, et
c.)); note that Adelmo
# had "0" for roll, pitch and yaw forearm

train1 <- train[,-c(1, 3:7,12:36,50:59,69:83,87:101,103:112,125:139,141:150)]
test1 <- test[,-c(1, 3:7,12:36,50:59,69:83,87:101,103:112,125:139,141:150)]
train1 <- train1[,c(2:53,1,54)]
test1 <- test1[,c(2:53,1,54)]

#--------------------------------------------------------------
# subsetting the data into training and test data
```

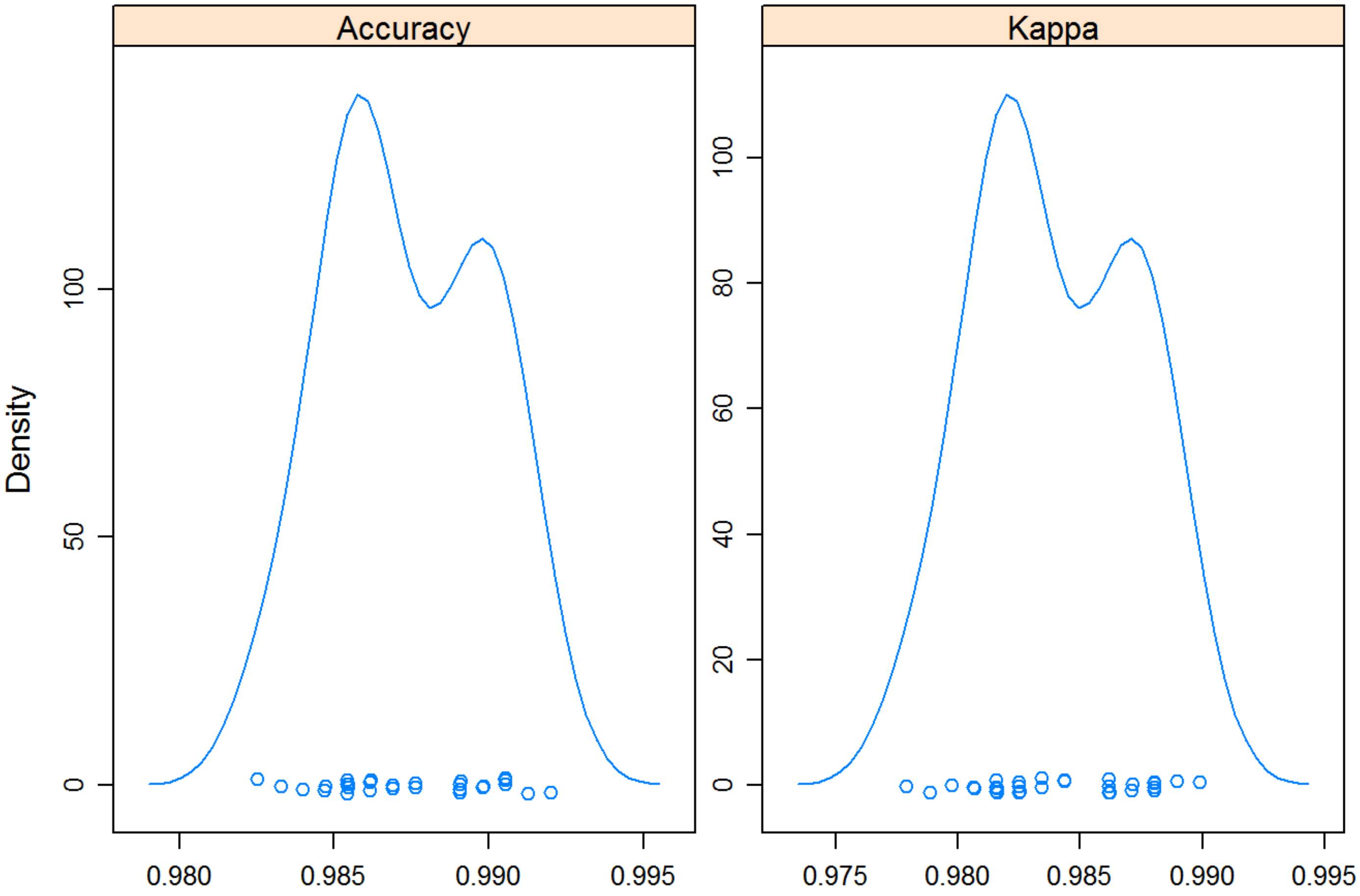
```
inTrain <- createDataPartition(y=train1$class, p=0.7, list=FALSE)

training <- train1[inTrain,]
testing <- train1[inTrain,]

#Looking for complete cases
train2 <- complete.cases(train1)
train3 <-train1[train2,]
nrow(train3)

# look at distribution of samples among classes in all three data sets
prop.orig <- prop.table(table(train1[54]))
original_Proportions <- prop.orig
prop.train <- prop.table(table(training[54]))
training_Proportions <- prop.train
prop.test <- prop.table(table(testing[54]))
testing_Proportions <- prop.test
Table1 <- rbind(original_Proportions,training_Proportions,testing_Proportions)
Table1

# check for near zero varaiance predictors
training1 <- nearZeroVar(training)
training1

# check for highly correlated variables and remove
Corr <- cor(training[-c(53:54)])
highCorr <- findCorrelation(Corr, 0.90)
names(training[highCorr])

training <- training[,-highCorr]
testing <- testing[,-highCorr]
test2 <- test1[,-highCorr]

#--------------------------------------------------------------
# fitting a model and detemining variable importance using random forest
cvCtrl <- trainControl(method='repeatedcv', repeats=3, classProbs=TRUE)
modFit <- train(class~., data=training, method='rf', ntree=10, trControl=cvCtrl)
```

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

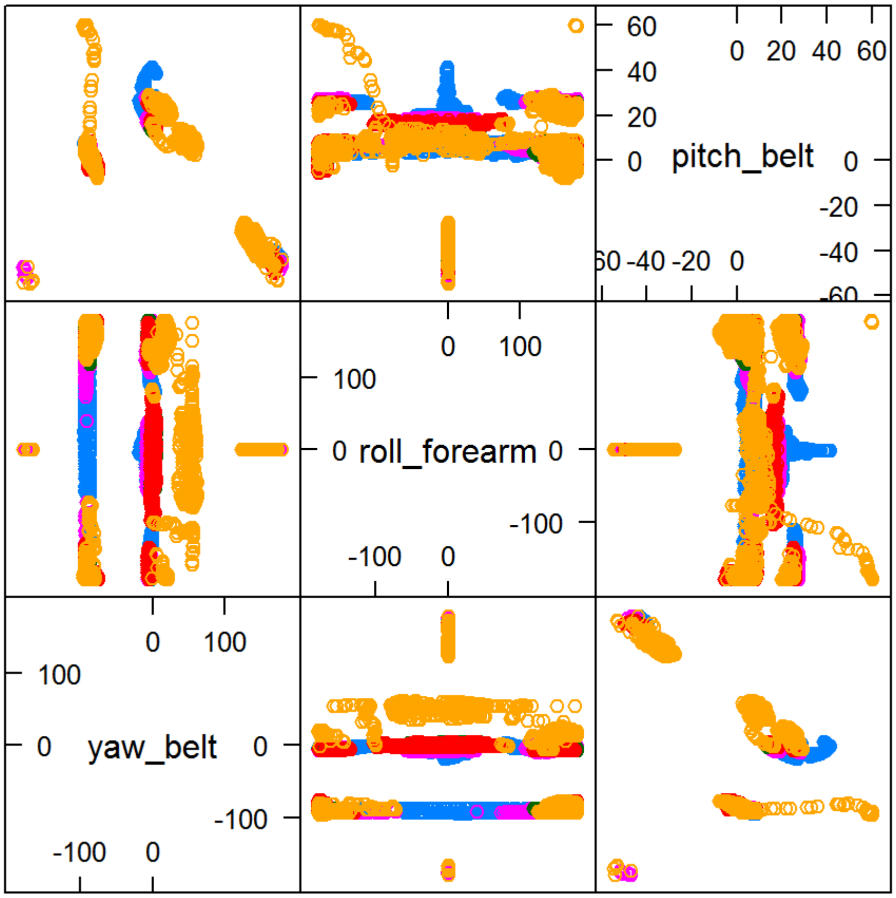
```
modFit

#Looking at plots of results and variable importance
Y <- plot(modFit)
Y
```



```
p <- resampleHist(modFit)
p
```

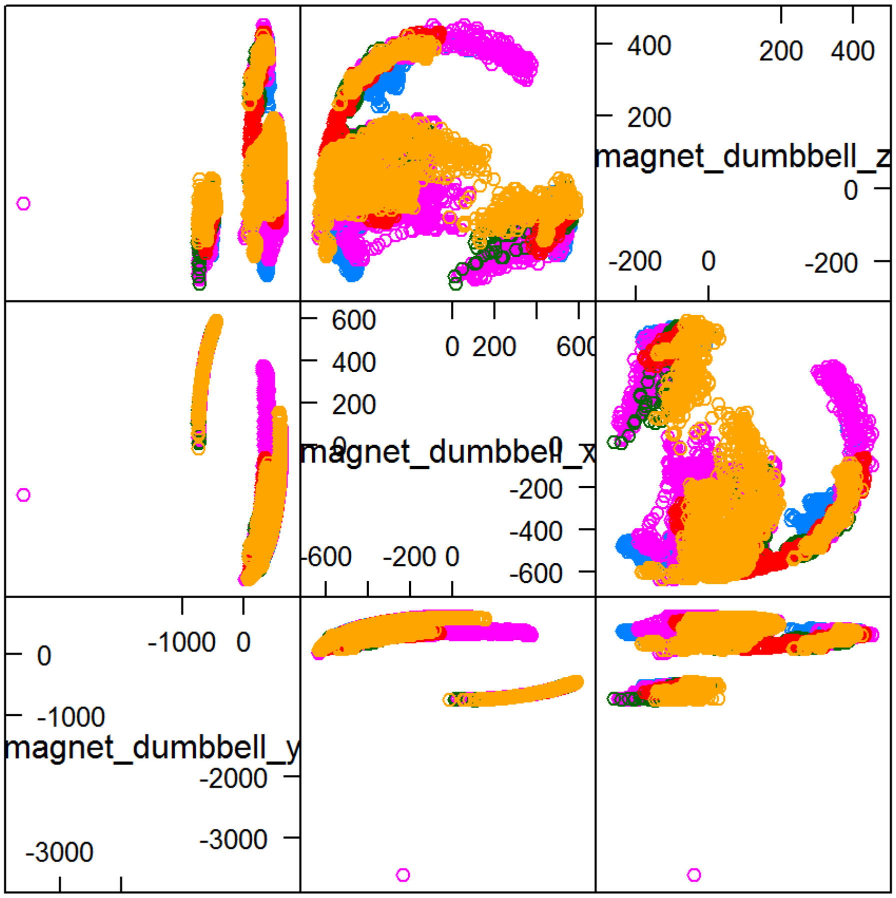


```
Imp <- varImp(modFit)
Imp

# getting column numbers for important features
which(colnames(training)=='yaw_belt')
which(colnames(training)=='pitch_forearm')
which(colnames(training)=='pitch_belt')
which(colnames(training)=='magnet_dumbbell_z')
which(colnames(training)=='magnet_dumbbell_y')
which(colnames(training)=='roll_forearm')
which(colnames(training)=='magnet_belt_y')
#-------------------------------------------------------------------
# plotting

# plotting 4 most important features against each other
featurePlot(x=training[,c(2,35,1)],
            y=training$class,
            plot='pairs')
```



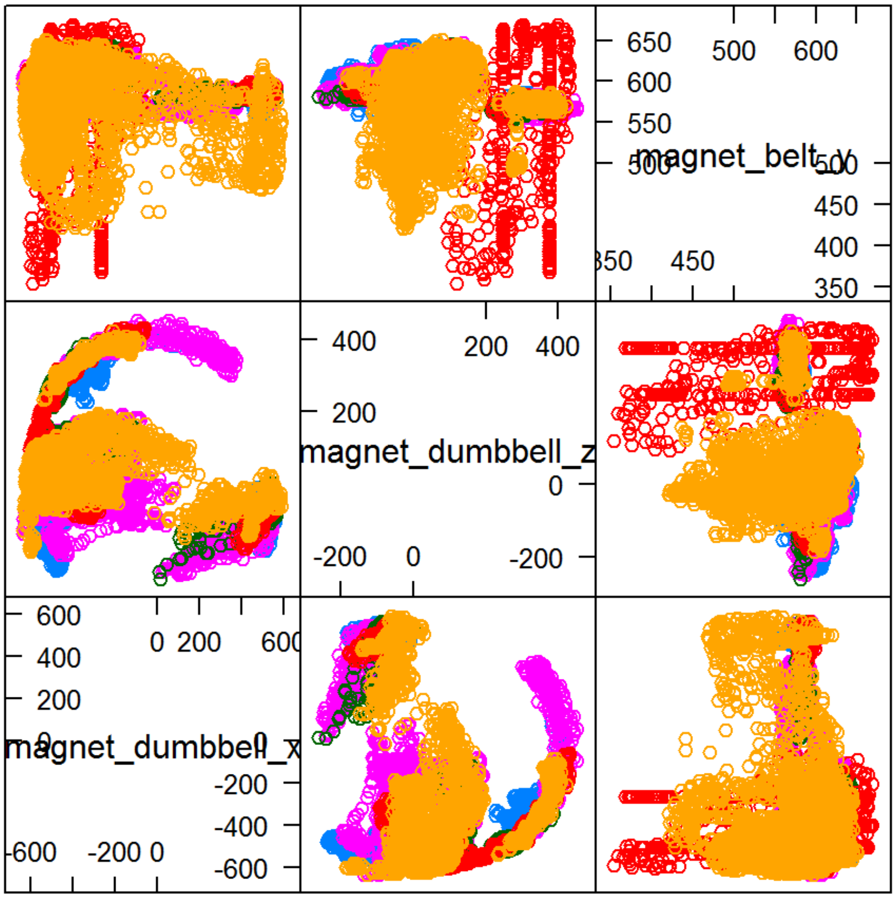
Scatter Plot Matrix

```
featurePlot(x=training[,c(33,32,34)],
            y=training$class,
            plot='pairs')
```



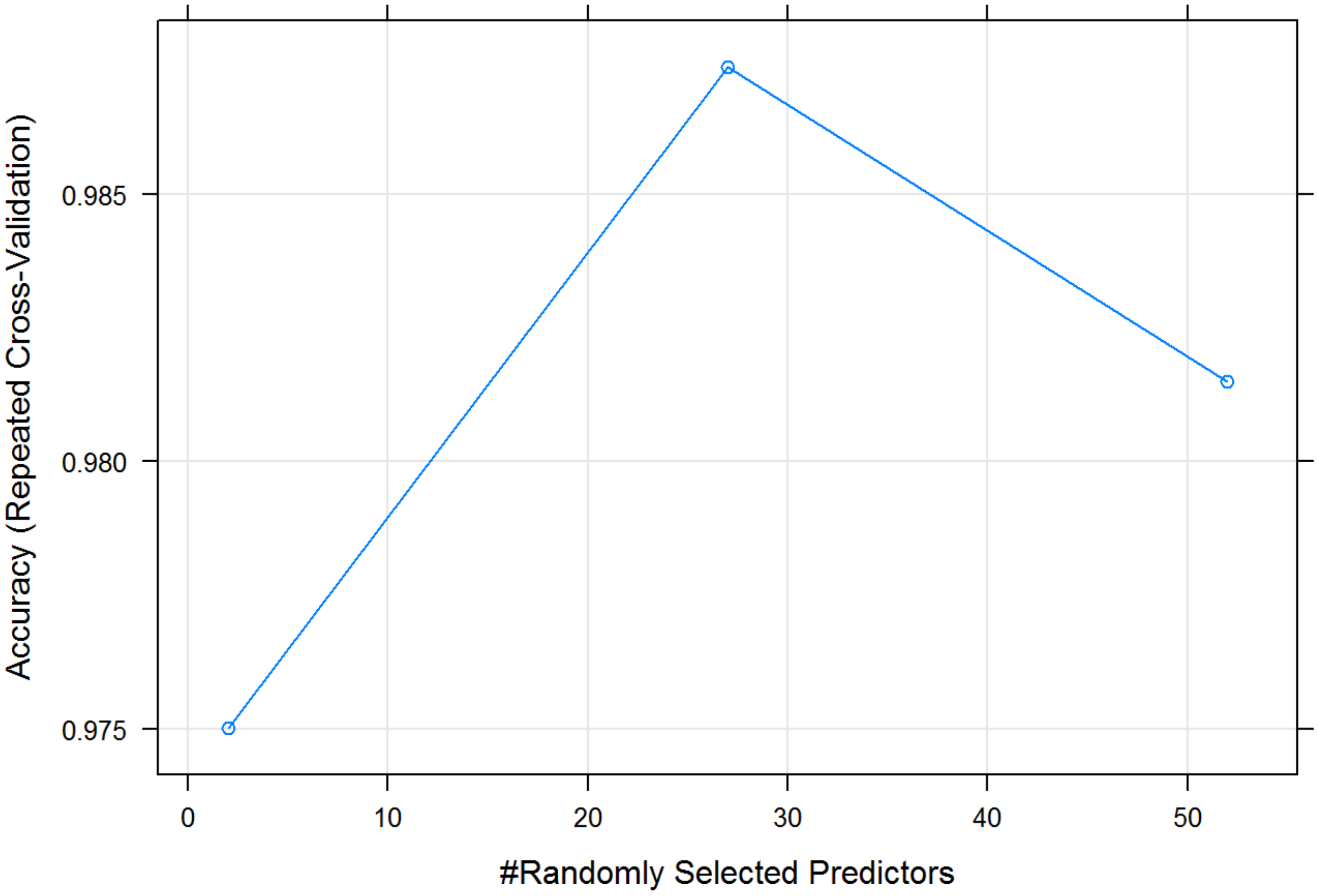
Scatter Plot Matrix

```
featurePlot(x=training[,c(32,34,8)],
            y=training$class,
            plot='pairs')
```



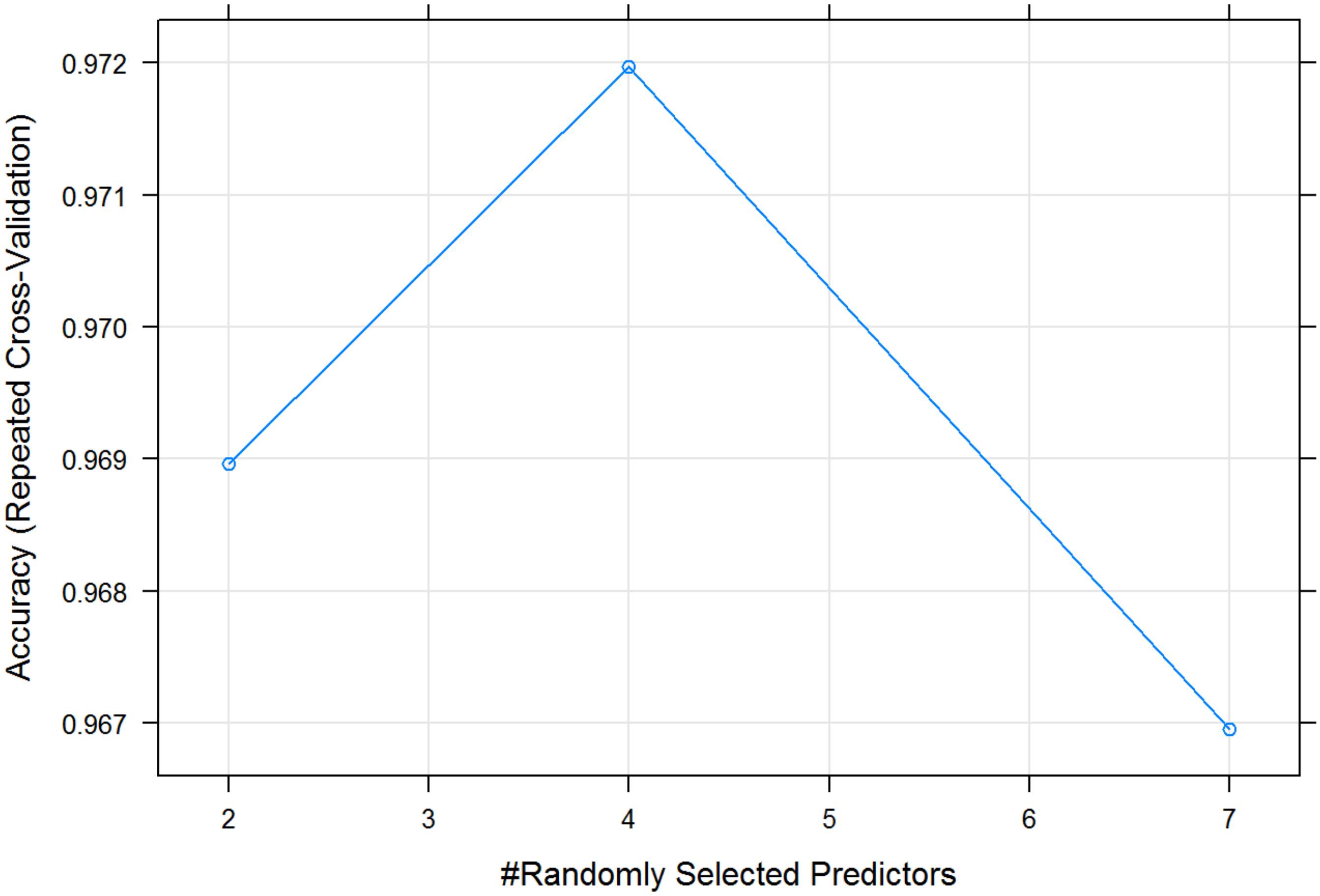
Scatter Plot Matrix

```
# plotting number of predictors vs. accuracy
plot(modFit)
```

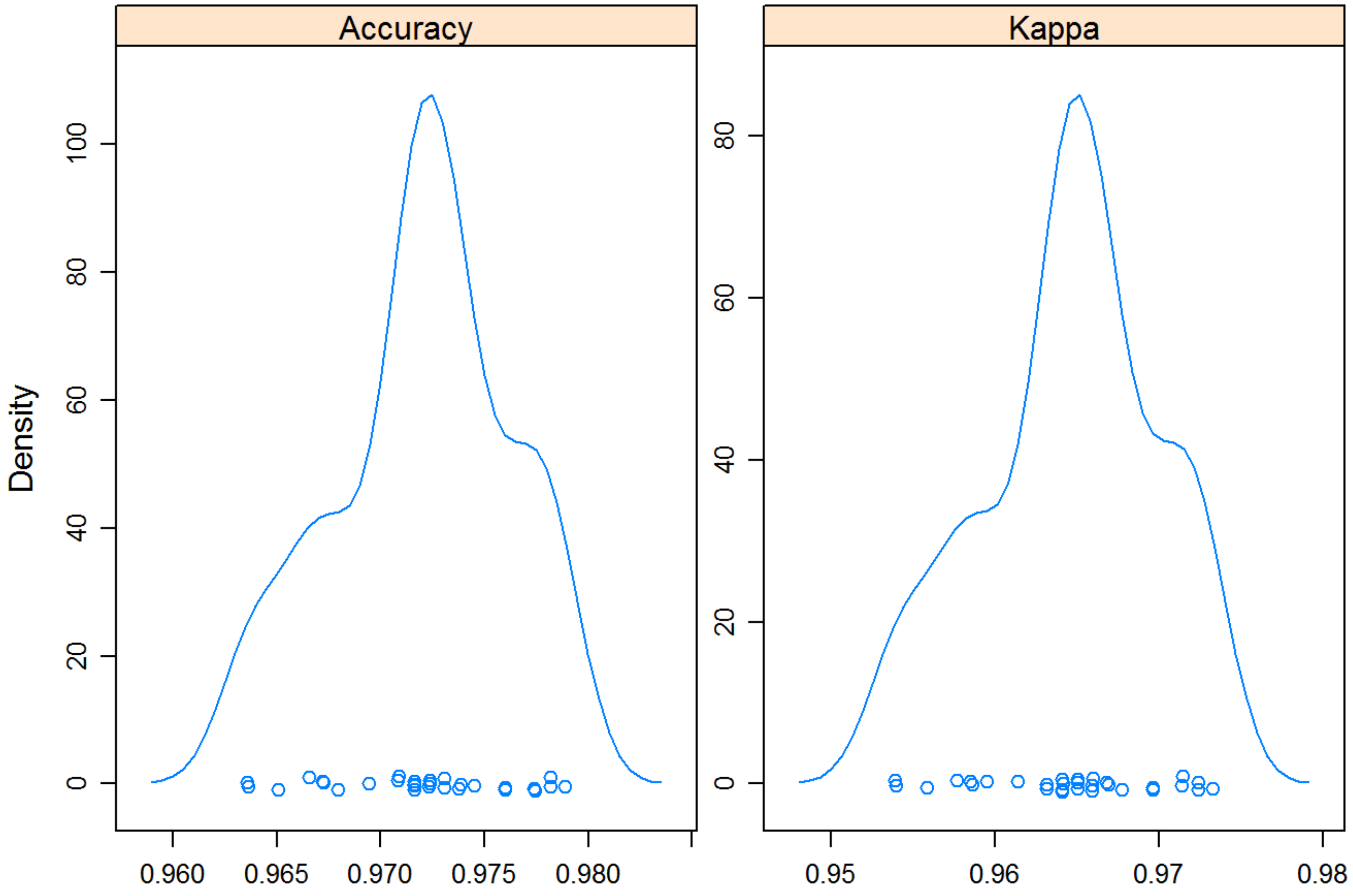


```
#----------------------------------------------------------------------
# predict weight lifing class using model and getting confusion matrix
predWL <- predict(modFit, newdata=testing)
str(predWL)

confMatrix1 <- confusionMatrix(data=predWL, testing$class)
confMatrix1

#---------------------------------------------------------------------
#predicting testing data
pred <- predict(modFit,test2)
TestPred <- as.character(pred)
TestPred

#---------------------------------------------------------------------
# repeating the model building with the 7 most important variables

# loading the data
test <- read.csv('C:/Rrepos/Rdata/PML/testing.csv', stringsAsFactors=T, header = T)
train <- read.csv('C:/Rrepos/Rdata/PML/training.csv', stringsAsFactors=T, header = T)

# preparing the data;since goal was to predict, eliminated data that did not seem relevan
t
```

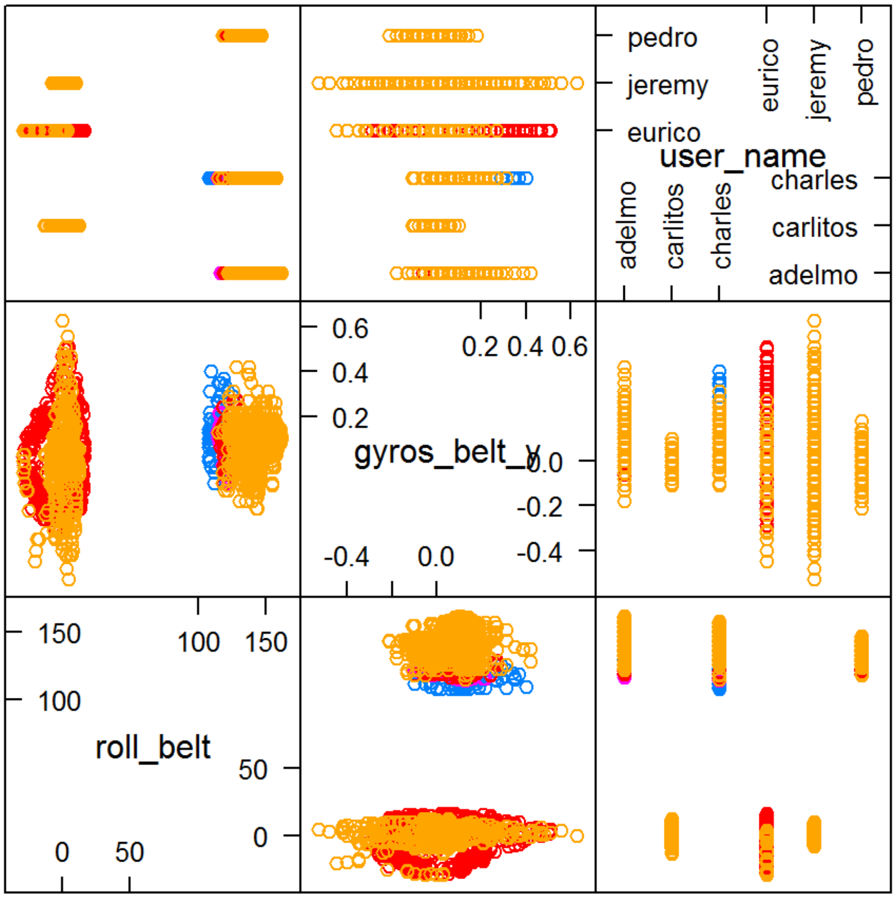
```
# (some of initialcolumns, including individual names, and then data that had no values o
r
# NAs in the testing dataset; this corresponded to similar but not completely equivalent
# data in the training set (i.e., were some values for kurtosis, etc.)); note that Adelmo
# had "0" for roll, pitch and yaw forearm

train1 <- train[,-c(1, 3:7,12:36,50:59,69:83,87:101,103:112,125:139,141:150)]
test1 <- test[,-c(1, 3:7,12:36,50:59,69:83,87:101,103:112,125:139,141:150)]

#-------------------------------------------------------------
# subsetting the data into training and test data
inTrain <- createDataPartition(y=train1$class, p=0.7, list=FALSE)

training <- train1[inTrain,]
testing <- train1[-inTrain,]

#looking for complete cases
train2 <- complete.cases(train1)
train3 <-train1[train2,]
nrow(train3)

which(colnames(training)=='yaw_belt')
which(colnames(training)=='pitch_forearm')
which(colnames(training)=='pitch_belt')
which(colnames(training)=='magnet_dumbbell_z')
which(colnames(training)=='magnet_dumbbell_y')
which(colnames(training)=='roll_forearm')
which(colnames(training)=='magnet_belt_y')

# creating subsets of data that had just the 7 most important variables
trainSub <- training[,c(3:4, 13, 39:42, 54)]
testSub <- testing[,c(3:4, 13, 39:42, 54)]
test1Sub <- test1[,c(3:4, 13, 39:42, 54)]

#-------------------------------------------------------------
# fitting a model and detemining variable importance using random forest
cvCtrl <- trainControl(method='repeatedcv', repeats=3, classProbs=TRUE)
modFit <- train(class~., data=trainSub, method='rf', ntree=10, trControl=cvCtrl)
modFit

Y <- plot(modFit)
Y
```



```
p <- resampleHist(modFit)
p
```

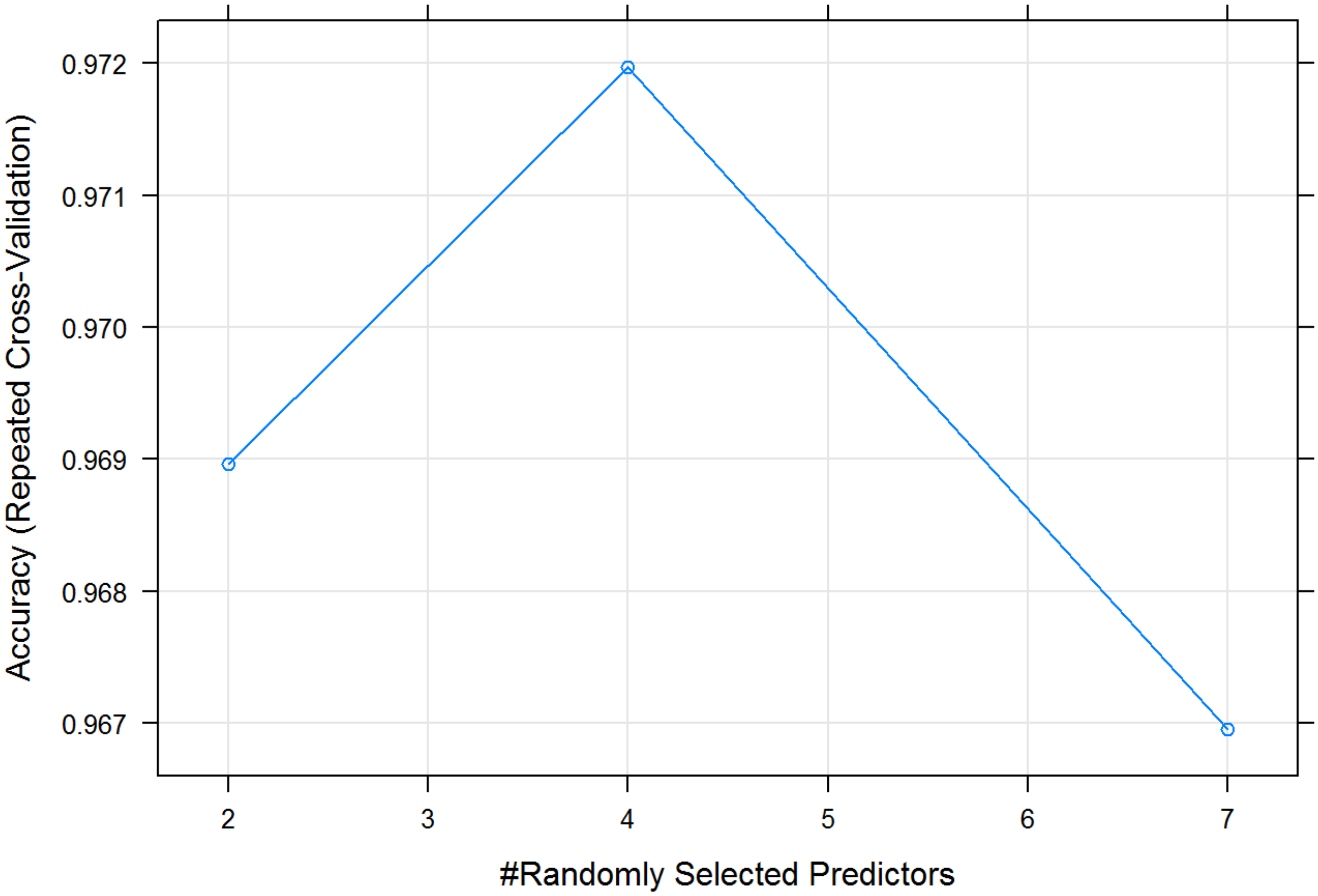


```
Imp <- varImp(modFit)
Imp

# getting column numbers for important features
which(colnames(trainSub)=='yaw_belt')
which(colnames(trainSub)=='pitch_forearm')
which(colnames(trainSub)=='pitch_belt')
which(colnames(trainSub)=='magnet_dumbbell_z')
which(colnames(trainSub)=='magnet_dumbbell_y')
which(colnames(trainSub)=='roll_forearm')
which(colnames(trainSub)=='magnet_belt_y')

#-------------------------------------------------------------------
# plotting

# plotting 4 most important features against each other
featurePlot(x=training[,c(2,7,1)],
            y=training$class,
            plot='pairs')
```



Scatter Plot Matrix

```
# plotting number of predictors vs. accuracy
plot(modFit)
```



```
#----------------------------------------------------------------------
# predict weight lifing class from testing data set using model and getting confusion mat
rix
predWL <- predict(modFit, newdata=testSub)

confMatrix1 <- confusionMatrix(data=predWL, testSub$class)
confMatrix1

#---------------------------------------------------------------------
#predicting testing data
pred <- predict(modFit,test1Sub)
TestPred <- as.character(pred)
TestPred
```