course project

m.pryidun

2023-02-08

Load packages:

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: ’randomForest’

## The following object is masked from ’package:ggplot2’:
##
##     margin
```

```
library(dplyr)
```

```
##
## Attaching package: ’dplyr’

## The following object is masked from ’package:randomForest’:
##
##     combine

## The following objects are masked from ’package:stats’:
##
##     filter, lag

## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

Set urls:

```
trainURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
```

Read in the data

```
download.file(trainURL, destfile = "pml-training.csv", method = "curl")
download.file(testURL, destfile = "pml-testing.csv", method = "curl")
training <- read.csv("pml-training.csv")
testing <- read.csv("pml-testing.csv")
```

Remove na and zero values:

```
NAChecker <- function(x){unlist(apply(x, 2, function(x){length(which(!is.na(x)))}))}
NDataPoints <- NAChecker(training)

CompleteVariable <- c()
for(i in 1:length(NDataPoints)){
  if(NDataPoints[[i]]==nrow(training)){
    CompleteVariable <- c(CompleteVariable, names(training)[i])
  }
}

trainingSet <- training[, names(training) %in% CompleteVariable]

nzv <- nearZeroVar(trainingSet, saveMetrics = TRUE)

myVar <- rownames(subset(nzv, nzv==FALSE))
print(myVar)
```

```
##  [1] "X"                    "user_name"            "raw_timestamp_part_1"
##  [4] "raw_timestamp_part_2" "cvtd_timestamp"       "num_window"
##  [7] "roll_belt"            "pitch_belt"           "yaw_belt"
## [10] "total_accel_belt"     "gyros_belt_x"         "gyros_belt_y"
## [13] "gyros_belt_z"         "accel_belt_x"         "accel_belt_y"
## [16] "accel_belt_z"         "magnet_belt_x"        "magnet_belt_y"
## [19] "magnet_belt_z"        "roll_arm"             "pitch_arm"
## [22] "yaw_arm"              "total_accel_arm"      "gyros_arm_x"
## [25] "gyros_arm_y"          "gyros_arm_z"          "accel_arm_x"
## [28] "accel_arm_y"          "accel_arm_z"          "magnet_arm_x"
## [31] "magnet_arm_y"         "magnet_arm_z"         "roll_dumbbell"
## [34] "pitch_dumbbell"       "yaw_dumbbell"         "total_accel_dumbbell"
## [37] "gyros_dumbbell_x"     "gyros_dumbbell_y"     "gyros_dumbbell_z"
## [40] "accel_dumbbell_x"     "accel_dumbbell_y"     "accel_dumbbell_z"
## [43] "magnet_dumbbell_x"    "magnet_dumbbell_y"    "magnet_dumbbell_z"
## [46] "roll_forearm"         "pitch_forearm"        "yaw_forearm"
## [49] "total_accel_forearm"  "gyros_forearm_x"      "gyros_forearm_y"
## [52] "gyros_forearm_z"      "accel_forearm_x"      "accel_forearm_y"
## [55] "accel_forearm_z"      "magnet_forearm_x"     "magnet_forearm_y"
## [58] "magnet_forearm_z"     "classe"
```

Create a new data set with newly identified set of variables and remove the first 6 columns which would not be used for prediction:

```r
myVar <- myVar[-(1:6)]
trainingData <- select(trainingSet, one_of(myVar))
```

Slice dataset for validationt:

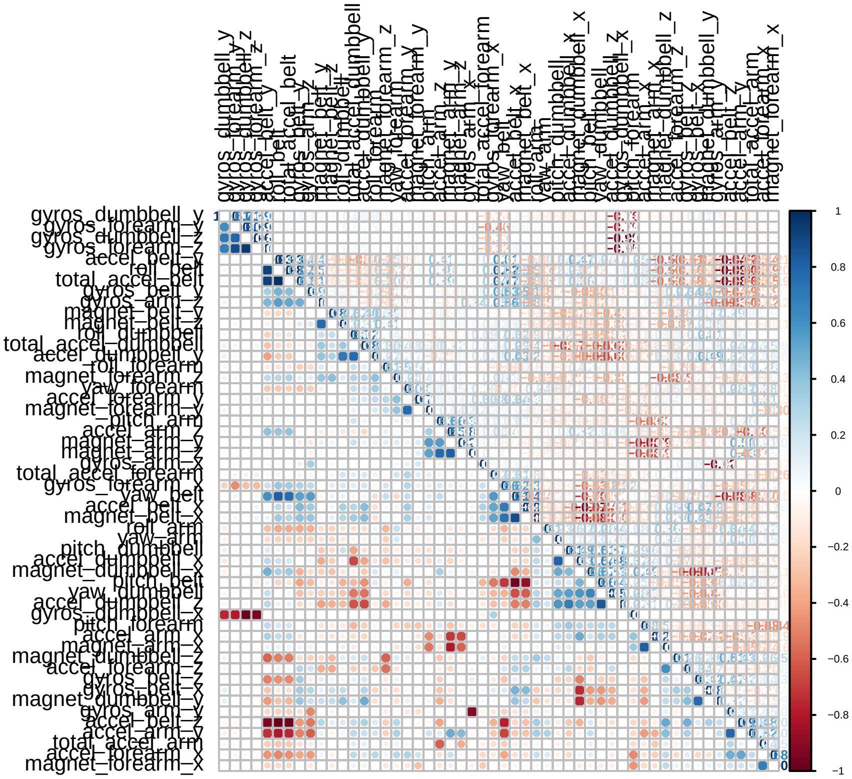
```r
inTrain <- createDataPartition(y=trainingData$classe, p=0.6, list=FALSE)


trainingPart <- trainingData[inTrain,]
validationPart <- trainingData[-inTrain,]
```

Check Relationahsips Among Variables:

```r
varCorr <- round(cor(trainingPart[sapply(trainingPart, is.numeric)]), 4)

par(ps=5)
corrplot.mixed(varCorr, order="hclust", tl.col="black", diag="n", tl.pos="lt", lower="circle", upper =
```



# Principal Component Analysis

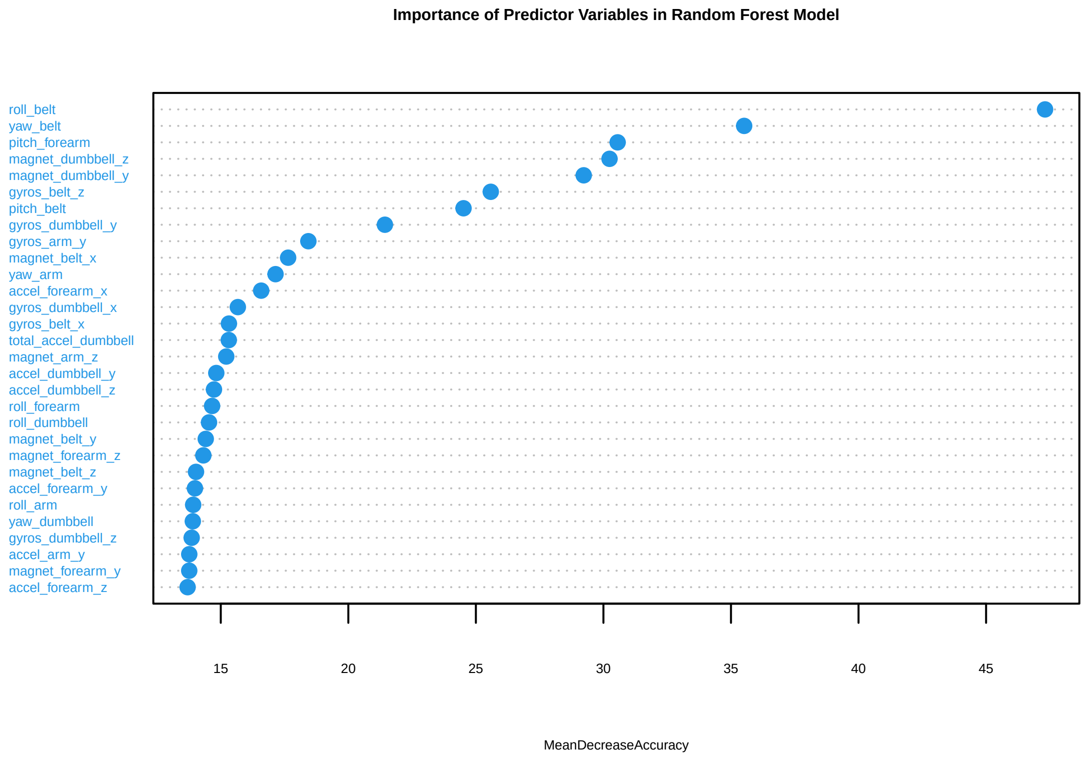
```r
reduced <- preProcess(trainingPart[,-53], method = "pca")
trainingPCA <- predict(reduced, trainingPart[,-53])
validationPCA <- predict(reduced, validationPart[,-53])
print(reduced)
```

```
## Created from 11776 samples and 52 variables
##
## Pre-processing:
##   - centered (52)
##   - ignored (0)
##   - principal component signal extraction (52)
##   - scaled (52)
##
## PCA needed 24 components to capture 95 percent of the variance
```

Build a Random Forest Model Without PCA

```r
modelRF2 <- train(classe ~., method="rf", data=trainingPart, trControl = trainControl(method="cv", numbe

par(ps=5)
varImpPlot(modelRF2$finalModel, sort = TRUE, type = 1, pch=19, col=12, cex=1, main="Importance of Predic
```



Caculate the Accuracy of the Model

```
modelRF2Val <- predict(modelRF2, validationPart)
modelRF2Acc <- round(postResample(validationPart$classe, modelRF2Val)[[1]], 4)
modelRF2Acc
```

```
## [1] 0.9907
```

#Final Test

```
modelRF2Test <- predict(modelRF2, testing)
modelRF2Test
```

```
##  [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
```