Prediction Assignment

5/16/2021

Executive Summary

In this project we will build a prediction model from data on barbell lifts done correctly and incorrectly in 5 different ways by 6 participants. The goal of the project is to predict the manner in which they did the exercise with the prediction outcome being the variable "classe" in the training set.This project will determine which of the other variables to predict with and will then use the prediction model to predict 20 different test cases.

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(corrplot)
```

```
## corrplot 0.88 loaded
```

```
training<-read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv",na.strings=c(

quiz_set<-read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv",na.strings=c("
```

Cleaning the Data

Here we will remove informational variables as well as those with zero variance across samples as they are not useful for prediction.

```
training2<-training[,-c(1:7)] #filter out informational cols since they are not needed for model predic

#remove near zero variance variables. that is, variables that are constant/near constant across samples
nzv <- nearZeroVar(training2)
training3 <- training2[ , -nzv]
```

Here, our standard treatment for NA's is to remove variables that have them. However, this may not be the best method with a better method being imputation.

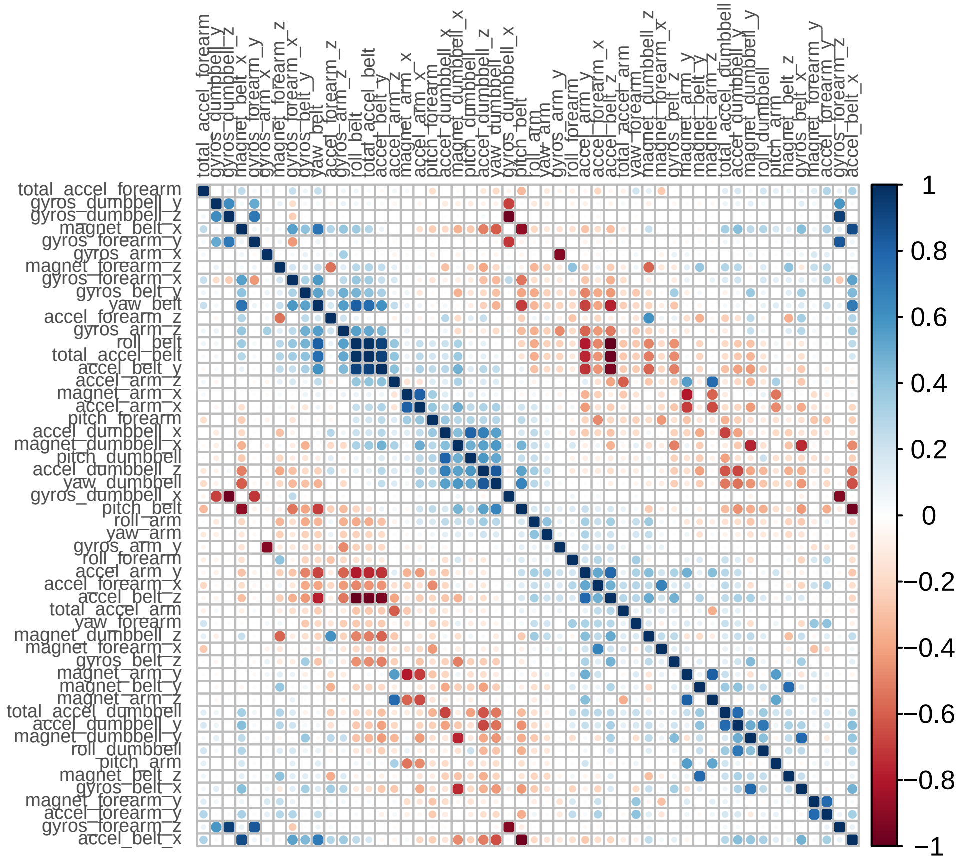
```
#Remove variables that are  NA.
na_var <- sapply(training3, function(x) sum(is.na(x)))
# alt. apply(training3,2, function(x) sum(is.na(x)))>0 and then do == FALSE, or do it like with ==0

training4 <- training3[ , na_var == 0]  #alt. training3[,colSums(is.na(training2)) == 0] . NOTE : compl
```

Correlation Matrix

A quick visualization is done to see how the variables correlate. The AOE (Angular Order of Eigenvectors) method is used, which orders based on variable similarity calculated as the angular distance between variables on an eigenvector plot.

```
corr_mat <- cor(training4[ , -53])
#AOE : angles between vectors on an eigenvector plot approximate correlations
#so an ordering based on the angular positions of these vectors naturally places the most similar varia
corrplot(corr_mat, order = "AOE", method = "circle", type = "full", tl.cex = 0.55, tl.col = rgb(.3, .3,
```



Model Fit

Here we fit the model using random forest since these models are known for their very high accuracy. We use k-fold cross-validation method as a resampling method

```
#fit a random forest model with cross-validation

fit<-train(classe ~ .,method="rf",data=training4, trControl = trainControl(method = "cv"))

fit$finalModel
```

```
##
## Call:
```

```
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0.42%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 5578    1    0    0    1 0.0003584229
## B   13 3780    4    0    0 0.0044772189
## C    0   14 3406    2    0 0.0046756283
## D    0    0   41 3173    2 0.0133706468
## E    0    0    0    5 3602 0.0013861935
```

```
varImp(fit)
```

```
## rf variable importance
##
##   only 20 most important variables shown (out of 52)
##
##                      Overall
## roll_belt             100.00
## yaw_belt               81.98
## magnet_dumbbell_z      74.61
## pitch_belt             66.42
## magnet_dumbbell_y      62.52
## pitch_forearm          62.42
## magnet_dumbbell_x      53.11
## roll_forearm           48.39
## accel_belt_z           46.20
## magnet_belt_z          44.70
## accel_dumbbell_y       42.67
## magnet_belt_y          42.25
## roll_dumbbell          41.06
## accel_dumbbell_z       39.37
## roll_arm               36.83
## accel_forearm_x        33.67
## total_accel_dumbbell   30.50
## gyros_belt_z           30.17
## yaw_dumbbell           30.13
## accel_dumbbell_x       29.57
```

```
pred<-predict(fit,quiz_set) #note to self: no need to do anything to quiz DS because model only uses va
pred
```

```
##  [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
```

Conclusion

The random forest model produces a very high accuracy prediction. The OOB is very low.