Prediction Assignment Writeup

JC

7/31/2019

Summary

The main idea of this project was using data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants to predict the manner in which they did the exercise. To reach the goal, fist explored data and then built the model, using cross validation, expressing the expected out-of-sample error and reasons of choosing the prediction model. Finally,used the prediction model to predict 20 different test cases.

Processing Data and Exploratory Analysis

```
library(knitr)
library(caret)
```

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

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

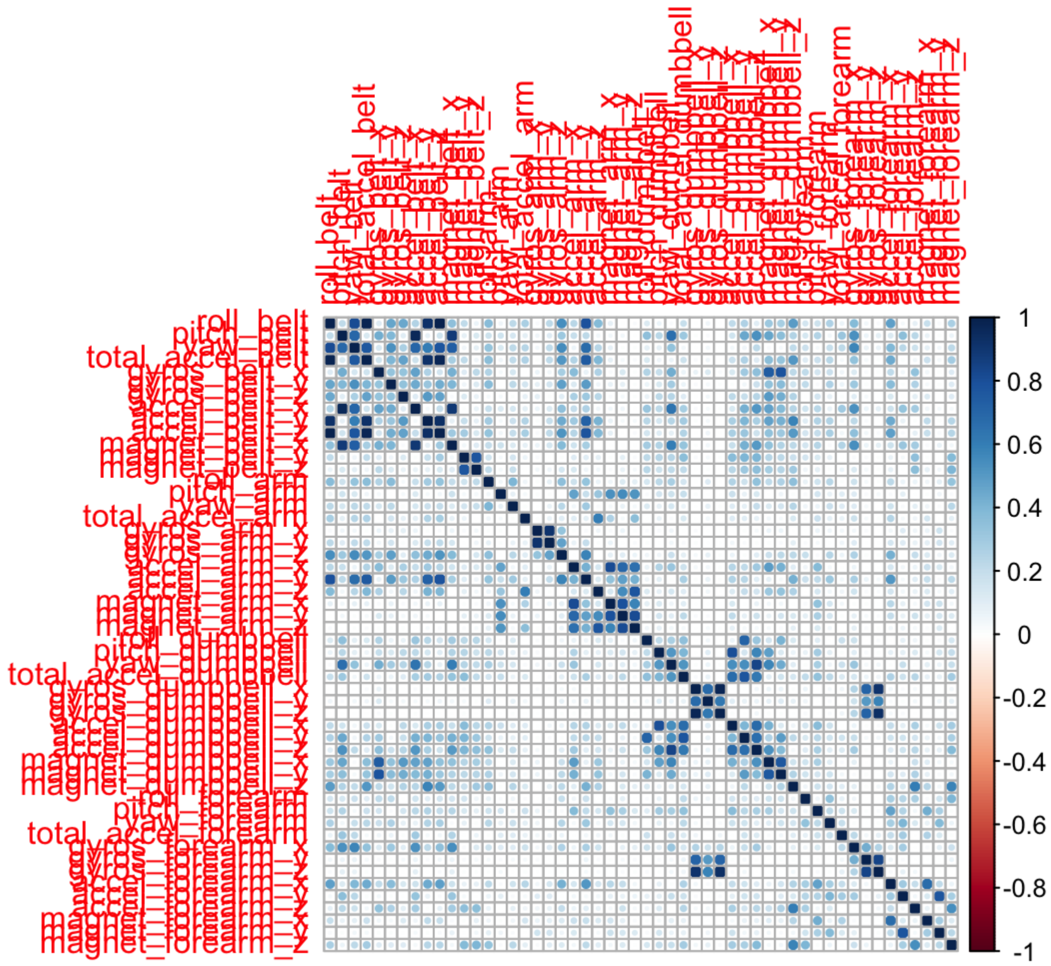
```
#Read data
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv",destfil
e = "pml-traininig.csv")
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv",destfile
= "pml-testing.csv")
traindata <- read.csv("pml-traininig.csv",na.strings=c("NA","#DIV/0!",""))
testdata <- read.csv("pml-testing.csv",na.strings=c("NA","#DIV/0!",""))
#clean NA, near zero values, and ID variables
traindata <-traindata[,colSums(is.na(traindata)) == 0]
testdata <-testdata[,colSums(is.na(testdata)) == 0]
nzv <- nearZeroVar(traindata)
traindata <- traindata[,-nzv]
traindata <- traindata[,-(1:6)]
```

[Correlation Analysis] Perform correction analysis and check if there was any highly correlated preditors. If any, deleted the variables.

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
cor <- abs(cor(traindata[,-53]))
corrplot(cor)
```



```
cor08 <- findCorrelation(cor, cutoff=0.8)
traindata <- traindata[, -cor08]
```

[Cross Validation] Create data partition in 70,30 ratio.

```
set.seed(1234)
train  <- createDataPartition(traindata$classe, p=0.7, list=FALSE)
training <- traindata[train, ]
testing  <- traindata[-train, ]
```

Building Model

```
  The target variable was a class variable. Random Forests, Decision Tree and Gradient Boost
ed were three candidate methods to build prediction model.
```

```
table(training$classe)
```

```
##
##    A    B    C    D    E
## 3906 2658 2396 2252 2525
```

1 Random Forests

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

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

```
modrf <- randomForest(classe ~ ., data=training)
# prediction on testing
predrf <- predict(modrf, testing)
RandomForestsAccuracy <- round(confusionMatrix(predrf, testing$classe)$overall[1],3)
```

2 Decision Tree

```
library(rpart)
modDT <- rpart(classe ~ ., data=training, method="class")
# prediction on Test dataset
predDT <- predict(modDT, newdata=testing,type = "class")
DecTreeAccuracy <- round(confusionMatrix(predDT, testing$classe)$overall[1],3)
```

3 Gradient Boosted Model

```
modGBM  <- train(classe ~ ., data=training, method = "gbm", verbose = FALSE,trControl =train
Control(method = "repeatedcv", number = 5, repeats = 1))
predGBM <- predict(modGBM, newdata=testing)
GBMAccuracy <-round(confusionMatrix(predGBM, testing$classe)$overall[1],3)
```

[Accuracy, Expected Out-of-Sample Error and Model Choosing] The Model from random forests method had best accuracy and lowest expected out-of sample error (1-accuracy), which meaned it could provide the best predict values. The random forests model was chosen as prediction model.

```
t <-data.frame(Model=c("RandomForest","DecisionTree","GradientBoostedModel"),Accuracy=c(Rand
omForestsAccuracy,DecTreeAccuracy,GBMAccuracy))
t$Error <-1-t$Accuracy
kable(t,caption ="Accuracy and Expected Out-of-Sample Error" )
```

Accuracy and Expected Out-of-Sample Error

Model	Accuracy	Error
RandomForest	0.995	0.005
DecisionTree	0.733	0.267
GradientBoostedModel	0.957	0.043

Predict 20 Different Test Cases

```
predTestdata <- predict(modrf, testdata)
predTestdata
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 
##  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
```