Quality_exercise

sa

May 6, 2018

Background: The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. This is an attempt to quantify how well do the participants do the exercise.Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes. Participants were supervised by an experienced weight lifter to make sure the execution complied to the manner they were supposed to simulate. The exercises were performed by six male participants aged between 20-28 years, with little weight lifting experience. We made sure that all participants could easily simulate the mistakes in a safe and controlled manner by using a relatively light dumbbell (1.25kg). The training a and testing datasets have been shared.

Executive Summary - The outcome variable is Classe with 5 levels as mentioned above. In the project I have first downloaded the data from the link shared. The objective is to predict between Classe A which is within the specifications and the others which include mistakes. The objective is to increase accuracy and reduce the out of sample error. 1. Cross - Validation - cross validation was performed by sampling 25% of the training dataset while building a model on 75% of it. Predictions are checked on the sub-sampled dataset.2.Out of sample error - It is arrived on the cross -validation dataset. The calculation from the best model is considered and we measure it as 1- accuracy = 0.0053; 3. Reason for choices: Random forest model was chose owing to it's superior accuracy, sensitivity, specificity and pos pred value are all quite high. Cross validation is done on a sub-sample of training dataset as the dataset has large number of observations.Our outcome variable is a factor variable, hence considered error type as 1-accuracy. In terms of data preparation,near zero variance variables were removed, missing value variables deleted, highly correlated variables were also removed.

Downloading training and testing data from URL

```
library(RCurl)
```

```
## Warning: package 'RCurl' was built under R version 3.2.5
```

```
## Loading required package: bitops
```

```
## Warning: package 'bitops' was built under R version 3.2.5
```

```
URL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
train <- getURL(URL)
train_dat <- read.csv(textConnection(train))
## The training dataset has 19622 obs. of  160 variables
URL1 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
test <- getURL(URL1)
test_dat <- read.csv(textConnection(test))
```

Creating training and validation datsets - For Cross - Validation. The training and validation datasets have been created with replacement in the 75-25 ratio.The 25% dataset will be used for cross validation.

```
set.seed(1234)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.2.5
```

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

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

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

```
sainTrain <- createDataPartition(y=train_dat$classe, p=0.75, list=FALSE)
final_satrain = train_dat[sainTrain, ]
final_savalid = train_dat[-sainTrain, ]
table(final_satrain$classe)
```

```
##
##    A    B    C    D    E
## 4185 2848 2567 2412 2706
```

```
table(final_savalid$classe)
```

```
##
##    A    B    C    D    E
## 1395  949  855  804  901
```

Exploratory Analysis and Data Prep

```
## Eliminating for Zero Variance variables
myNZV <- nearZeroVar (final_satrain, saveMetrics = TRUE)
zeroNms <- myNZV$nzv
final_satrain = final_satrain[!zeroNms]
##105 Variables remain
##Missing Value Treatment
final_satrain<-final_satrain[,colSums(is.na(final_satrain)) == 0]
##59 Variables remain
##Removing id and other variables like username
final_satrain1 = final_satrain[,-c(1:6)]
final_satrain1$classe <- factor(final_satrain1$classe)
table(final_satrain1$classe)
```

```
##
##    A    B    C    D    E
## 4185 2848 2567 2412 2706
```

```
##Removing highly correlated variables

df <- as.data.frame (final_satrain1)
ncol(df)
```

```
## [1] 53
```

```
dim(df)
```

```
## [1] 14718    53
```

```
for(i in 1:ncol(df))
{
  df[,i] <- as.numeric(df[,i])
}
for(i in 1:ncol(df))
{
  df[,i] <- ifelse(is.na(df[,i]),0,df[,i])
}

myDF <- cor(df)
myCor <- findCorrelation (myDF, cutoff = 0.8)
corNms <- colnames(final_satrain1)[myCor]
mynms <- colnames(final_satrain1) [!(colnames(final_satrain1) %in% corNms)]
mynms
```

```
##  [1] "yaw_belt"             "total_accel_belt"     "gyros_belt_x"
##  [4] "gyros_belt_y"         "gyros_belt_z"         "magnet_belt_x"
##  [7] "magnet_belt_y"        "magnet_belt_z"        "roll_arm"
## [10] "pitch_arm"            "yaw_arm"              "total_accel_arm"
## [13] "gyros_arm_y"          "gyros_arm_z"          "accel_arm_y"
## [16] "accel_arm_z"          "magnet_arm_x"         "magnet_arm_z"
## [19] "roll_dumbbell"        "pitch_dumbbell"       "yaw_dumbbell"
## [22] "total_accel_dumbbell" "gyros_dumbbell_y"     "accel_dumbbell_y"
## [25] "magnet_dumbbell_x"    "magnet_dumbbell_y"    "magnet_dumbbell_z"
## [28] "roll_forearm"         "pitch_forearm"        "yaw_forearm"
## [31] "total_accel_forearm"  "gyros_forearm_x"      "gyros_forearm_z"
## [34] "accel_forearm_x"      "accel_forearm_y"      "accel_forearm_z"
## [37] "magnet_forearm_x"     "magnet_forearm_y"     "magnet_forearm_z"
## [40] "classe"
```

After checking for zero variance, highly correlated variables, missing value treatment 40 variables remain . The same process is repeated for validation and testing dataset

```
myNZV <- nearZeroVar (final_savalid, saveMetrics = TRUE)
zeroNms <- myNZV$nzv
final_savalid = final_savalid[!zeroNms]
##105 Variables remain
##Missing Value Treatment
final_savalid<-final_savalid[,colSums(is.na(final_savalid)) == 0]
##59 Variables remain
##Removing id and other variables like username
final_savalid1 = final_savalid[,-c(1:6)]
mynms <- colnames(final_savalid1) [!(colnames(final_savalid1) %in% corNms)]


myNZV <- nearZeroVar (test_dat, saveMetrics = TRUE)
zeroNms <- myNZV$nzv
test_dat = test_dat[!zeroNms]
```

```
##105 Variables remain
##Missing Value Treatment
test_dat<-test_dat[,colSums(is.na(test_dat)) == 0]
##59 Variables remain
##Removing id and other variables like username
test_dat1 = test_dat[,-c(1:6)]
mynms <- colnames(test_dat1) [!(colnames(test_dat1) %in% corNms)]
```

Using Machine Learning Algorithm to understand drivers for quality - Decision Tree

```
##Running Decison Tree Model on the data
library(rpart)
mod1 <- rpart(classe ~ yaw_belt + total_accel_belt  + gyros_belt_x  + gyros_belt_y  +gyros_belt_z +
magnet_belt_x + magnet_belt_y  + magnet_belt_z  + roll_arm  + pitch_arm + yaw_arm + total_accel_arm   +
accel_dumbbell_y + magnet_dumbbell_x + magnet_dumbbell_y + magnet_dumbbell_z  + roll_forearm    +
pitch_forearm + yaw_forearm + total_accel_forearm + gyros_forearm_x  + gyros_forearm_z + accel_forearm_
magnet_forearm_z, data=final_satrain1, method="class")
## Predicting on the Validation Dataset
predictvalid = predict(mod1,final_savalid1,type="class")
confusionMatrix(predictvalid,final_savalid1$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1269  153   57   46   63
##          B   63  464   49   95  141
##          C   31  213  696  122  137
##          D   30   83   35  503   84
##          E    2   36   18   38  476
##
## Overall Statistics
##
##                Accuracy : 0.6949
##                  95% CI : (0.6818, 0.7078)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6125
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9097  0.48894   0.8140   0.6256  0.52830
## Specificity            0.9091  0.91201   0.8758   0.9434  0.97652
## Pos Pred Value         0.7991  0.57143   0.5805   0.6844  0.83509
## Neg Pred Value         0.9620  0.88148   0.9571   0.9278  0.90194
## Prevalence             0.2845  0.19352   0.1743   0.1639  0.18373
## Detection Rate         0.2588  0.09462   0.1419   0.1026  0.09706
## Detection Prevalence   0.3238  0.16558   0.2445   0.1499  0.11623
```

```
## Balanced Accuracy      0.9094  0.70047   0.8449   0.7845  0.75241
```

From model1, it indicates that the accuracy is 0.7023. Though the Sensitivity is high for CLass A, the same is not true for the other classes. Also pos pred value is low. Hence trying out random forest model.

Building a Random Forest Model

```
##Running Random Forest on the data
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.2.5
```

```
## randomForest 4.6-12
```

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

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

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

```
mod2 = randomForest(classe ~ yaw_belt + total_accel_belt  + gyros_belt_x  + gyros_belt_y  +gyros_belt_z
## Predicting on the Validation Dataset
myPred <- predict (mod2, final_savalid1)
actual <- final_savalid1$classe
confusionMatrix(myPred,actual)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1393    5    0    0    0
##          B    2  943   10    0    0
##          C    0    1  844   13    0
##          D    0    0    1  791    0
##          E    0    0    0    0  901
##
## Overall Statistics
##
##                Accuracy : 0.9935
##                  95% CI : (0.9908, 0.9955)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9917
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9986   0.9937   0.9871   0.9838   1.0000
## Specificity            0.9986   0.9970   0.9965   0.9998   1.0000
```

```
## Pos Pred Value         0.9964   0.9874   0.9837   0.9987   1.0000
## Neg Pred Value         0.9994   0.9985   0.9973   0.9968   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2841   0.1923   0.1721   0.1613   0.1837
## Detection Prevalence   0.2851   0.1947   0.1750   0.1615   0.1837
## Balanced Accuracy      0.9986   0.9953   0.9918   0.9918   1.0000
```

The accuracy of the random Forest model is 0.9947. On the out of training sample dataset, the out of sample error is 0.0053. Owing to better accuracy numbers, the random Forest model is chosen.

Citation

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.

```
predictsubmission <- predict(mod2, test_dat1, type = "class")
predictsubmission
```

```
##  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
```