Practical Machine Learning Week 4 Project

XH

4/20/2020

PURPOSE

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. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. Using the measurements taken, we are asked to build a model to predict the manner they did their excercise using the outcome variable 'classe'. The 5 outcomes of 'classe' are:

1. Exactly according to the specification (Class A)
2. Throwing the elbows to the front (Class B)
3. Lifting the dumbbell only halfway (Class C)
4. Lowering the dumbbell only halfway (Class D)
5. Throwing the hips to the front (Class E)

DATA

The data for this project come from this source: [http://groupware.les.inf.puc-rio.br/har].

The training data set is here:

[https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv]

The test data set is here:

[https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv]

```
#Load data sets
training <- read.csv(url("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"))
quiz <- read.csv(url("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"))
#Look at dimensions of data
dim(training)
```

```
## [1] 19622   160
```

```
dim(quiz)
```

```
## [1]  20 160
```

The training data set has 19622 observations and 160 variables whereas the testing data set has 20 observations and 160 variables as well.

DATA CLEANING

First, we will remove any variables that have close to non-zero variance.

```
#load libraies need to do predictions
library(caret)
library(randomForest)
library(rpart)
```

```r
#Eliminate variables with close to zero variance
zero_var <- nearZeroVar(training)
training <- training[,-zero_var]
quiz <- quiz[,-zero_var]
dim(training)
```

```
## [1] 19622   100
```

```r
dim(quiz)
```

```
## [1]  20 100
```

This has removed 60 variables and we are left with 100 columns now. Next, we will remove variables where the majority of the values are N/A. We will use a threshold of 95%.

```r
#Remove variables with 95% or greather NA values
na <- sapply(training, function(x) mean(is.na(x))) > 0.95
training <- training[,na == FALSE]
quiz <- quiz[,na == FALSE]
dim(training)
```

```
## [1] 19622    59
```

```r
dim(quiz)
```

```
## [1] 20 59
```

We are now left with 59 variables. Finally, we will remove the first seven variables from the data set because these are non predictors.

```r
#Remove non-predictors
training <- training[,8:59]
quiz <- quiz[,8:59]
dim(training)
```

```
## [1] 19622    52
```

```r
dim(quiz)
```

```
## [1] 20 52
```

Thus, our cleaned data set contains only 52 variables. Next, we start to build our model.

DATA PARTITION

We will first partition our training data set into a training and test data set using a 60%/40% split.

```r
#Partition data into training and test data sets
intrain     <- createDataPartition(training$classe, p = 0.6, list = FALSE)
intraining  <- training[intrain,]
intest      <- training[-intrain,]
dim(intraining)
```

```
## [1] 11776    52
```

```r
dim(intest)
```

```
## [1] 7846   52
```

MODEL BUILDING

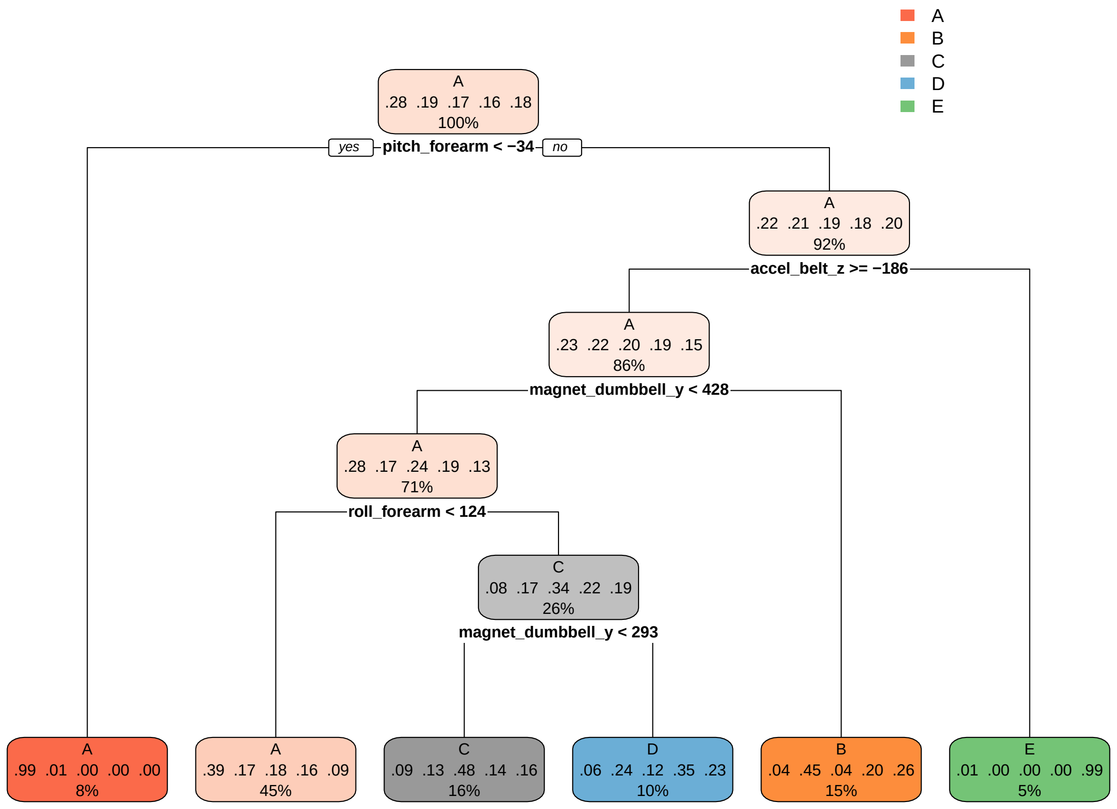
DECISION TREE

For this project, we are attempting to predict a classification. Hence, it would be a good idea to try the decision tree model and random forest models. First, we will try the decision tree model with 10-fold cross validation in an attempt to increase our accuracy.

```
#Define training control
set.seed(924)
train.control <- trainControl(method = "cv", number = 10)
#Train the model
modelDT <- train(classe ~., data = intraining, method = "rpart",trControl = train.control)
#Prediction of decision trees
predictDT <- predict(modelDT, intest)
#Confusion Matrix for Decision trees
confusionMatrix(predictDT, intest$classe)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction    A    B    C    D    E
##          A 2014  610  643  557  310
##          B   47  521   54  244  295
##          C  119  170  582  188  201
##          D   45  216   89  297  221
##          E    7    1    0    0  415
##
## Overall Statistics
##
##                Accuracy : 0.488
##                  95% CI : (0.4769, 0.4991)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3311
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9023   0.3432  0.42544  0.23095  0.28779
## Specificity            0.6224   0.8989  0.89534  0.91296  0.99875
## Pos Pred Value         0.4872   0.4488  0.46190  0.34217  0.98109
## Neg Pred Value         0.9413   0.8509  0.88066  0.85827  0.86165
## Prevalence             0.2845   0.1935  0.17436  0.16391  0.18379
## Detection Rate         0.2567   0.0664  0.07418  0.03785  0.05289
## Detection Prevalence   0.5269   0.1480  0.16059  0.11063  0.05391
## Balanced Accuracy      0.7624   0.6210  0.66039  0.57195  0.64327
```

```
#Plot decision tree
library(rpart.plot)
rpart.plot(modelDT$finalModel, roundint=FALSE)
```


A
B
C
D
E
A .28 .19 .17 .16 .18 100%
yes
pitch_forearm < −34
no
A .22 .21 .19 .18 .20 92%
accel_belt_z >= −186
A .23 .22 .20 .19 .15 86%
magnet_dumbbell_y < 428
A .28 .17 .24 .19 .13 71%
roll_forearm < 124
C .08 .17 .34 .22 .19 26%
magnet_dumbbell_y < 293
A .99 .01 .00 .00 .00 8%
A .39 .17 .18 .16 .09 45%
C .09 .13 .48 .14 .16 16%
D .06 .24 .12 .35 .23 10%
B .04 .45 .04 .20 .26 15%
E .01 .00 .00 .00 .99 5%

We see from the confusion matrix and the plot that accuracy is roughly 50% which means our out of sample error is around 50% as well. Thus, we will go on to try another model which is the random forest.

RANDOM FOREST

Due to the bad accuracy prediction of decision tree, we will now try random forest with 10-fold cross validation as well.

```
#Define training control
set.seed(924)
#Train the model
modelRF <- train(classe ~., data = intraining, method = "rf", ntree=100,trControl = train.control)
#Prediction of Random Forest
predictRF <- predict(modelRF, intest)
#Confusion matrix for Random Forest
confusionMatrix(predictRF, intest$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2225    9    0    0    0
##          B    5 1500   11    0    1
##          C    1    3 1353   24    1
##          D    0    5    4 1262    2
##          E    1    1    0    0 1438
##
```

```
## Overall Statistics
##
##                Accuracy : 0.9913
##                  95% CI : (0.989, 0.9933)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.989
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9969   0.9881   0.9890   0.9813   0.9972
## Specificity            0.9984   0.9973   0.9955   0.9983   0.9997
## Pos Pred Value         0.9960   0.9888   0.9790   0.9914   0.9986
## Neg Pred Value         0.9988   0.9972   0.9977   0.9963   0.9994
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2836   0.1912   0.1724   0.1608   0.1833
## Detection Prevalence   0.2847   0.1933   0.1761   0.1622   0.1835
## Balanced Accuracy      0.9976   0.9927   0.9923   0.9898   0.9985
```

For the random forest model, the accuracy is 99.15% which is much better than the decision tree model. Our out of sample error is just 0.85%. Hence, this is the model that we select to predict the 'classe' outcome.

QUIZ PREDICTION

Now, we will use our selected model to predict the outcome using our quiz data set.

```r
#Predict quiz data set
print(predict(modelRF, newdata=quiz))
```

```
##  [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

In this project, we looked at the decision tree algorithm with 10-fold cross validation as well as the random forest algorithm with 10-fold cross-validation. Based on the out of sample error, the model selected was random forest which had a 99.15% accuracy compared to the decision tree which had an accuracy of only around 50%.