Practical Machine Learning Final Project

Mariah Birgen

Executive Summary

After downloading and cleaning the data to select only acceleration predictors, two different models, random trees and generalized boosted regression, were run on 60% of the training set. The models were then run on the remaining of the 40% testing set to test for accuracy. The model with the best performance, random trees, was then chosen to run on the given testing set. The results are given below.

Introduction

Six participants were asked to perform barbell lifts correctly and incorrectly in 5 different ways. The goal of the project is to use data from accelerometers on the belt, forearm, arm, and dumbell to predict whether or not the barbell lift was perfomed correctly. The report describes the building of the model including dealing with missing data and deciding which model method to use.

Data Preparation

Data is downloaded from the given urls.

Then, data is saved into r for processing.

```r
training <- read.csv("trainingdata.csv", stringsAsFactors = FALSE)
testing <- read.csv("testingdata.csv", stringsAsFactors = FALSE)
```

Data Cleaning

The dataset is very large, so we are able to break the data into two pieces, one to train the models and the other to test the models for accuracy.

```r
set.seed(23846)
inTrain <- createDataPartition(y = training$classe, p=0.6, list = FALSE )
trainset <- training[inTrain,]
testset <- training[-inTrain,]
```

Select only predictors that indicate acceleration. After working with the first selection, it becomes clear that predictors that measure the variance of the acceleration need to be excluded.

```r
trainy <- trainset[,160]
trainuser<- trainset[,2]
trainset <- trainset[, grepl("accel", names (trainset))]
trainset <- trainset[, !grepl("var", names(trainset))]
trainset <- cbind(classe=trainy, trainset)
dim(trainset)
```

```
## [1] 11776    17
```

This leaves us with 16 predictors.

```
names(trainset)
```

```
##  [1] "classe"               "total_accel_belt"     "accel_belt_x"
##  [4] "accel_belt_y"         "accel_belt_z"         "total_accel_arm"
##  [7] "accel_arm_x"          "accel_arm_y"          "accel_arm_z"
## [10] "total_accel_dumbbell" "accel_dumbbell_x"     "accel_dumbbell_y"
## [13] "accel_dumbbell_z"     "total_accel_forearm"  "accel_forearm_x"
## [16] "accel_forearm_y"      "accel_forearm_z"
```

Remove predictors with minimal variance or missing values

Remove predictors with minimal variance (this turns out to have little effect).

```
nsv <- nearZeroVar(trainset, saveMetrics = TRUE)
trainset <- trainset[, nsv$nzv==FALSE]
dim(trainset)
```

```
## [1] 11776    17
```

Remove predictors with missing values (this turns out to have little effect).

```
trainset<- trainset[, colSums(is.na(trainset)) == 0]
dim(trainset)
```

```
## [1] 11776    17
```

Model Building

Random Forest

Training

```
controlRF <- trainControl(method="cv", number=3, verboseIter=FALSE)
model1 <- train(classe~., data = trainset, method = "rf", trControl=controlRF)
model1$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: 6.18%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3213   27   50   54    4  0.04032258
## B  104 2050   81   23   21  0.10048267
## C   38   67 1925   18    6  0.06280428
## D   42   11   87 1777   13  0.07927461
## E    6   30   24   22 2083  0.03787529
```

Validation

We then validate the model obtained model $model1$ on the test data to find out how well it performs by looking at the Accuracy variable.

```r
predrf <- predict(model1, newdata = testset)
cmrf <- confusionMatrix(predrf, factor(testset$classe))
cmrf$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   9.430283e-01   9.278910e-01   9.376690e-01   9.480555e-01   2.844762e-01
## AccuracyPValue  McnemarPValue
##   0.000000e+00   2.291977e-16
```

We see the accuracy of the “rt” model is 94%. ## Generalized Boosted Regression ### Training

```r
set.seed(23846)
controlGBM <- trainControl(method = "repeatedcv", number = 5, repeats = 1)
model2 <- train(classe ~ ., data=trainset, method = "gbm", trControl = controlGBM, verbose = FALSE)
model2$finalModel
```

```
## A gradient boosted model with multinomial loss function.
## 150 iterations were performed.
## There were 16 predictors of which 16 had non-zero influence.
```

```r
# print model summary
print(model2)
```

```
## Stochastic Gradient Boosting
##
## 11776 samples
##    16 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 1 times)
## Summary of sample sizes: 9420, 9422, 9420, 9421, 9421
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.5454308  0.4166360
##   1                  100      0.6046194  0.4948253
##   1                  150      0.6370584  0.5369123
##   2                   50      0.6560795  0.5608886
##   2                  100      0.7314882  0.6585172
##   2                  150      0.7676625  0.7050553
##   3                   50      0.7324214  0.6594103
##   3                  100      0.7896571  0.7330953
##   3                  150      0.8202283  0.7720946
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 150, interaction.depth =
##  3, shrinkage = 0.1 and n.minobsinnode = 10.
```

Validation

We then validate the model obtained model *model*1 on the test data to find out how well it performs by looking at the Accuracy variable.

```
predgbm <- predict(model2, newdata = testset)
table(predgbm, testset$classe)
```

```
##
## predgbm    A    B    C    D    E
##       A 1993  168  110  110   20
##       B   46 1076   89   33   77
##       C   81  152 1132   91   78
##       D  104   56   24 1021   53
##       E    8   66   13   31 1214
```

```
cmgbm <- confusionMatrix(predgbm, factor(testset$classe))
cmgbm$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   8.202906e-01   7.721739e-01   8.116128e-01   8.287298e-01   2.844762e-01
## AccuracyPValue  McnemarPValue
##   0.000000e+00   3.239876e-36
```

We see that the accuracy of the "gbm" model on the testset is 82%. # Running Best Model on Testing data Because the random trees model had significantly better accuracy on the testset data, we will use it to predict our answers for the quiz.

```
Results <- predict(model1, newdata = testing)
Results
```

```
##  [1] B A C A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

Bibliography

Data comes from :

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.