Prediction Assignment Writeup

Hai Yang

December 29, 2017

Overview

This report will use model predict the manner in which a person lifted barbells using accelerometer data from individuals working out

Data

First the data will be loaded.

```
#Load data and convert blanks, divide by 0, into NA
train = read.csv('pml-training.csv', na.strings = c("", "NA", '#DIV/0!'))
test = read.csv('pml-testing.csv', na.strings = c("", "NA",'#DIV/0!'))
```

Preprocessing

The data will first need to be cleaned up. The variables containing entries, names, as well as time and trials are removed.

```
#Remove Descriptive variables
train = train[,-1:-7]
test = test[,-1:-7]

#Remove all columns but the actual gyros data with > 0.5 NAs(remove the min/max features already in dat
train = train[, colSums(is.na(train)) < nrow(train) * 0.5]
test$classe = 'dummy'
test = test[,names(train)]
test$classe = NULL
```

Before a model can be trained and tested, the data from train is split up in a 60/40 ratio for model training and validation using the caret package.

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

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

Exploratory Analysis

The goal of the model is to be able to predict which of the classes that a person's method of lifting barbells correspond to. As can be seen in the table below, there are five classes with the majority of methodology falling under category A.

```
table(train$classe)
```

```
##
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

```
avg_values = aggregate(. ~ classe,train,mean)
# print(avg_values)
```

In addition, in the data frame above, the averages per manner of barbell lifting is shown and shows some distinct differences in certain accelerometer data.

Model Training and Validation

Two models will be setup to find which is more accurate in the actual test set. Both a random forest model and cart model will be used due to their specialities in multi-class classification problems.

Random Forest

A random forest model is setup using every variable as a predictor for classe.

```
#Using a random forest model for classification
library(randomForest)
```

```
## 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
```

```
rfModel = randomForest(classe ~ .,trainset)

#Test set accuracy
predTest = predict(rfModel, newdata = testset)
print(confusionMatrix(predTest, testset$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2230    6    0    0    0
##          B    2 1510    9    0    0
##          C    0    2 1355   15    1
##          D    0    0    4 1271    6
##          E    0    0    0    0 1435
##
## Overall Statistics
##
##                Accuracy : 0.9943
##                  95% CI : (0.9923, 0.9958)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9927
##  Mcnemar's Test P-Value : NA
##
```

```
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9991   0.9947   0.9905   0.9883   0.9951
## Specificity            0.9989   0.9983   0.9972   0.9985   1.0000
## Pos Pred Value         0.9973   0.9928   0.9869   0.9922   1.0000
## Neg Pred Value         0.9996   0.9987   0.9980   0.9977   0.9989
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2842   0.1925   0.1727   0.1620   0.1829
## Detection Prevalence   0.2850   0.1939   0.1750   0.1633   0.1829
## Balanced Accuracy      0.9990   0.9965   0.9939   0.9934   0.9976
```

As can be seen from the confusion matrix, the random forest model has a accuracy of 0.9943 from the predictions using the testset data, which is very accurate. There is still a chance of overfitting which may be seen with the actual new data.

CART

A CART model is setup using every variable as a predictor for classe. The cp to help with making the model is determined using cross validation to find the optimal accruacy.

```
# Define cross-validation experiment
numFolds = trainControl( method = "cv", number = 10 )
cpGrid = expand.grid( .cp = seq(0.001,0.05,0.005))

# Perform the cross validation which shows .001 seemed to have lowest r^2
train(classe ~ ., data = trainset, method = "rpart", trControl = numFolds, tuneGrid = cpGrid )
```

```
## Loading required package: rpart
```

```
## CART
##
## 11776 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 10598, 10597, 10599, 10599, 10599, 10598, ...
## Resampling results across tuning parameters:
##
##   cp     Accuracy   Kappa
##   0.001  0.8981012  0.8710566
##   0.006  0.7886444  0.7324637
##   0.011  0.7246931  0.6505526
##   0.016  0.6648283  0.5770109
##   0.021  0.6266182  0.5301579
##   0.026  0.5562286  0.4306474
##   0.031  0.5267579  0.3841738
##   0.036  0.5205579  0.3762005
##   0.041  0.5006844  0.3495435
##   0.046  0.4812281  0.3172043
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was cp = 0.001.
```

```r
#CART regression model using all of the variables
trainCART = rpart(classe ~ ., trainset, method="class",cp=.001)

#Predictions on our test set and accuracy
predCART = predict(trainCART, newdata=testset, type="class")
print(confusionMatrix(predCART, testset$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2150   99   15   29    6
##          B   41 1282   82   27   42
##          C   13   62 1208   49   41
##          D   15   33   48 1155   50
##          E   13   42   15   26 1303
##
## Overall Statistics
##
##                Accuracy : 0.9047
##                  95% CI : (0.898, 0.9111)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8793
##  Mcnemar's Test P-Value : 4.371e-08
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9633   0.8445   0.8830   0.8981   0.9036
## Specificity            0.9735   0.9697   0.9745   0.9777   0.9850
## Pos Pred Value         0.9352   0.8697   0.8798   0.8878   0.9314
## Neg Pred Value         0.9852   0.9630   0.9753   0.9800   0.9784
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2740   0.1634   0.1540   0.1472   0.1661
## Detection Prevalence   0.2930   0.1879   0.1750   0.1658   0.1783
## Balanced Accuracy      0.9684   0.9071   0.9288   0.9379   0.9443
```

As can be seen in the confusion model above, the CART model is 0.9047 accurate which is lower than the random forest however, it may be underfitting this data set and may perform better in the actual test set.

Prediction

Both of the random forest and cart models will be used to test the predictions.

```r
#Random Forest
rfTrain = randomForest(classe ~ .,train)
predRF = predict(rfTrain, newdata = test)
predRF
```

```
##  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
```

The random forest model predicts accurately all 20 cases of the manner of activity.

```
#CART
# Perform the cross validation which shows .001 seemed to have lowest r^2
train(classe ~ ., data = train, method = "rpart", trControl = numFolds, tuneGrid = cpGrid )
```

```
## CART
##
## 19622 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 17660, 17660, 17660, 17660, 17659, 17660, ...
## Resampling results across tuning parameters:
##
##   cp     Accuracy   Kappa
##   0.001  0.9025074  0.8766125
##   0.006  0.7984422  0.7449370
##   0.011  0.7101203  0.6331275
##   0.016  0.6773029  0.5929159
##   0.021  0.6281232  0.5329479
##   0.026  0.5398540  0.4003050
##   0.031  0.5185004  0.3715896
##   0.036  0.4991343  0.3455980
##   0.041  0.4952607  0.3403221
##   0.046  0.4952607  0.3403221
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was cp = 0.001.
```

```
#CART regression model using all of the variables
CARTmodel = rpart(classe ~ ., train, method="class",cp=.001)

#Predictions on our test set and accuracy
predCART = predict(CARTmodel, newdata=test, type="class")
predCART
```

```
##  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  A  A  A  C  C  B  A  E  E  A  B  B  B
## Levels: A B C D E
```

The CART model predicts accurately 18/20 cases of manners of activity recorded.

Summary

The random forest model in this case more accurate in determining the manner of lifting dumbells that the participants did. The CART model did not predict as well as the random forest but may perform better in a different testing set.