RegressionModelsCourseProject_ElKerns

Ellen Kerns

May 7, 2018

Clasifying Weightlifting Technique Using Accelerometer Data

Executive Summary:

This analysis aimed to build a model that predicts the quality of the weightlifting technique used by participants wearing devices that record accelerometer data. The data were loaded from this project " http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har" which had 6 particpants perform Unilateral Dumbbell Biceps Curs in 5 different fashions with only one being correct and the other4 representing common mistakes. After features unsuitable for predcition were removed (timestamps, id, etc), the test data were split into a testing and training set in order to take adavantage of cross validation. Random forest and decision tree models were trained on the training set and tested on the train based test set. Due to it's higher accuracy (99% versus 52%), the random forest model was chosen. ##Preparing the Datasets:

```
#download and import dataset
download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv","train.csv")
train<-read.csv("train.csv")

#Remove variables that are not relevant to the analysis
train<-train[, -c(1:5)]

#Remove variables with 90+% missing values
mostlyNA <- sapply(train, function(x) mean(is.na(x))) > 0.95
train<-train[,mostlyNA == F]

# remove variables with nearly zero variance -> poor predictors
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.4
```

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

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

```
nzv <- nearZeroVar(train)
train <- train[, -nzv]

#make outcome a factor variable
train$classe<-as.factor(train$classe)

#Partition into test and train for cross validation
inTrain <- createDataPartition(y=train$classe,
                               p=0.75, list=FALSE)#75% train
trainTrain <- train[inTrain,]
testTrain <- train[-inTrain,]
```

Model Creation

```r
#3-fold cross validation
fitControl <- trainControl(method="cv", number=3, verboseIter=F)

#Random Forest
m1 <- train(y = trainTrain$classe, method="rf", x = trainTrain[,-54],
                    trControl=fitControl)
#Decision Tree
m2 <- train(y = trainTrain$classe, method="rpart", x = trainTrain[,-54],
                    trControl=fitControl)
```

Model Comparison

```r
    confusionMatrix(testTrain$classe,predict(m1,testTrain))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1393    1    0    0    1
##          B    0  947    1    1    0
##          C    0    1  854    0    0
##          D    0    0    2  802    0
##          E    0    0    0    0  901
##
## Overall Statistics
##
##                Accuracy : 0.9986
##                  95% CI : (0.9971, 0.9994)
##     No Information Rate : 0.2841
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9982
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9979   0.9965   0.9988   0.9989
## Specificity            0.9994   0.9995   0.9998   0.9995   1.0000
## Pos Pred Value         0.9986   0.9979   0.9988   0.9975   1.0000
## Neg Pred Value         1.0000   0.9995   0.9993   0.9998   0.9998
## Prevalence             0.2841   0.1935   0.1748   0.1637   0.1839
## Detection Rate         0.2841   0.1931   0.1741   0.1635   0.1837
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      0.9997   0.9987   0.9981   0.9991   0.9994
```

```r
    confusionMatrix(testTrain$classe,predict(m2,testTrain))
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1266   18  108    0    3
##          B  384  342  223    0    0
##          C  390   28  437    0    0
##          D  321  153  293    0   37
##          E   89   70  190    0  552
##
## Overall Statistics
##
##                Accuracy : 0.5296
##                  95% CI : (0.5155, 0.5436)
##     No Information Rate : 0.4996
##     P-Value [Acc > NIR] : 1.425e-05
##
##                   Kappa : 0.3868
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.5167  0.55974  0.34932       NA   0.9324
## Specificity            0.9474  0.85861  0.88557   0.8361   0.9191
## Pos Pred Value         0.9075  0.36038  0.51111       NA   0.6127
## Neg Pred Value         0.6626  0.93198  0.79896       NA   0.9900
## Prevalence             0.4996  0.12459  0.25510   0.0000   0.1207
## Detection Rate         0.2582  0.06974  0.08911   0.0000   0.1126
## Detection Prevalence   0.2845  0.19352  0.17435   0.1639   0.1837
## Balanced Accuracy      0.7321  0.70917  0.61745       NA   0.9257
```