Prediction Movement Data

Karl Andersson

Friday, April 24, 2015

Preparation and Method

Load Libraries and data

```
require(dplyr)
require(caret)
require(randomForest)
training = read.csv("pml-training.csv", header = T)
test = read.csv("pml-testing.csv", header = T)
```

Data Cleaning

Remove Outcome variable before transformation

```
classe <- training$classe
```

remove columns with all NA-values and X-column

```
training = training[,colSums(is.na(training)) ==0]
training = select(training,-X)
```

Select the same columns from test as from training

```
test <- test[,which(names(test) %in% names(training))]
```

Convert columns to numeric

```
training <- training[, sapply(training, is.numeric)]
test <- test[,sapply(test, is.numeric)]
```

add the outcome variable which was removed before cleaning

```
training$classe <- classe
```

Splitting Data

Set seed and aprtition data into training and validation sets

```
set.seed(12312)
trainIndex <- createDataPartition(training$classe, p=0.70, list=F)
trainData = training[trainIndex,]
validationData = training[-trainIndex,]
```

Train Model

Train a randomForest predictor with classe as outcome and all other columns as predictors.

```
train_control <- trainControl(method="boot", number=3,allowParallel=T)
```

```
model <- train(as.factor(classe) ~., data = trainData, method = "rf",trControl=train_control)
```

Make Predictions

Make Predictions on Training and Validation set using the trained model

```
predictTrain <- predict(model,trainData)
predictValidation <- predict(model,validationData)
```

Analysis and results

Confusion Matrix

A confusionmatrix on the validation predictions and validation set shows what what prediction the model makes correct and which not. The accurancy is very high 0.9997.

```
confusionMatrix(predictValidation,validationData$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    1    0    0    0
##          B    0 1138    0    0    0
##          C    0    0 1026    1    0
##          D    0    0    0  963    0
```

```
##          E    0    0    0    0 1082
##
## Overall Statistics
##
##                Accuracy : 0.9997
##                  95% CI : (0.9988, 1)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9996
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9991   1.0000   0.9990   1.0000
## Specificity            0.9998   1.0000   0.9998   1.0000   1.0000
## Pos Pred Value         0.9994   1.0000   0.9990   1.0000   1.0000
## Neg Pred Value         1.0000   0.9998   1.0000   0.9998   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1934   0.1743   0.1636   0.1839
## Detection Prevalence   0.2846   0.1934   0.1745   0.1636   0.1839
## Balanced Accuracy      0.9999   0.9996   0.9999   0.9995   1.0000
```

Our out-of test was 0.00034 = 0.034%

```
valInd <- validationData$classe == predictValidation
ErrorVal = length(valInd[valInd == FALSE])/ length(valInd)
```

```
ErrorVal
```

```
## [1] 0.0003398471
```

Prediction of test Dataset

```
predictTest = predict(model,newdata=test)
predictTest
```

```
##  [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
```