Prediction Assignment

Rosina Plomp

April 4, 2018

Summary

In this study, movement data from six participants was registered using accelerometers while they performed barbell lifts either correctly and incorrectly. The data was subdivided into training and testing sets, which were preprocessed with PCA. A random forest model was built on the training data, and was shown to have an accuracy of 97.88% for the testing data (out-of-sample error: 2.12%).

Data preprocessing

The training data was obtained from cloudfront.net. Columns which contained missing values ("NA") were removed. The data was suqsequently subdivided into a training set (70%) and a test set (30%). Principle component analysis (PCA) was used to combine correlated variables in the training dataset, leaving us with 26 predictor variables. The testing dataset was treated in the same way.

```
#download file
  setwd("G:/Coursera - Data Science/Course 8 - Machine Learning")
  ifelse(file.exists("predmachlearn.csv"),
      print("file exists"),
      download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv",
      destfile="predmachlearn.csv")
    )

#read in data
  data <- read.csv("predmachlearn.csv", na.strings = c("NA", ""), header=TRUE)

#remove variables with missing data
  col_NA <- vector(length=160) #160 is nr of columns
  for(i in 1:160){
    col_NA[i]<-sum(is.na(data[,i]))
  }
  col_remove <- col_NA==0
  data <- data[,col_remove]

#remove user name, timestamp and window variables
  data <- data[,c(8:60)]

#subdivide the training data into a training (70%) and test (30%) set.
    library(caret)
  inTrain <- createDataPartition(y=data$classe, p=0.7, list=FALSE)
    training <- data[inTrain,]
    testing <- data[-inTrain,]

#use PCA to merge highly correlated variables in training set
preProc <- preProcess(training[, -53], method = "pca")
training_PCA <- cbind(training[,53],predict(preProc, training[, -53]))
names(training_PCA)[1] <- "classe"
```

```
#do same for testing dataset
testing_PCA <- cbind(testing[,53],predict(preProc, testing[, -53]))
names(testing_PCA)[1] <- "classe"
```

Prediction model building with training set

Both a random forest model and a boosting model were built on the training data, and used to predict the classification of the testing data. Both models yielded an accuracy of 0.9963 (in-sample-error) within the training dataset. I chose to continue with the random forest model.

```
#random forest
  mod_rf<- train(classe~., method="rf", data=training_PCA,
  trControl=trainControl(method="cv", number=4), importance=TRUE)
  trainpred_rf <- predict(mod_rf, training_PCA)
  confusionMatrix(training$classe, trainpred_rf) #accuracy was 100%

#mod_gbm
    mod_gbm <- train(classe~., method="gbm", data=training_PCA, verbose=FALSE)
  trainpred_gbm <- predict(mod_gbm, training_PCA)
  confusionMatrix(training$classe, trainpred_gbm) #accuracy was 85.4%
```

Assessing out-of-sample error with testing set

Next, the random forest model was applied to the testing dataset, and this was revealed to have an accuracy of 97.88% (out-of-sample error: 2.18%).

```
#random forest
  testpred_rf <- predict(mod_rf, testing_PCA)
  confusionMatrix(testing$classe, testpred_rf) #accuracy was 97.88%
```

Quiz predictions

The data for the quiz prediction questions was downloaded from cloudfront.net (named "testing dataset"), and preprocessed in the same way as the training and testing data - variables with missing data were removed and PCA was used to combine correlated variables. Next, the random forest model was applied to predict the classification of these samples, yielding the following answers: B A A A A E D B A A B C B A E E A B B B.

```
#download file
  ifelse(file.exists("predmachlearn_quiz.csv"),
      print("file exists"),
      download.file("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv",
      destfile="predmachlearn_quiz.csv")
    )

#read in data
  data_quiz <- read.csv("predmachlearn_quiz.csv", header=TRUE)

#remove variables with missing data
  data_quiz <- data_quiz[,col_remove]

#remove user name, timestamp and window variables
  data_quiz <- data_quiz[,c(8:60)]
```

```
#use PCA to merge highly correlated variables
  quiz_PCA <- predict(preProc, data_quiz[, -53])

#prediction with rf model
  quiz_pred <- predict(mod_rf, newdata=quiz_PCA)
  #answers: B A A A A E D B A A B C B A E E A B B B
```