Coursera Machine Learning Project

louissmith

Thursday, February 12, 2015

Classification of exercise

Load packages

```
library(caret, quietly = TRUE)
```

```
## Warning: package 'caret' was built under R version 3.1.2
```

Load data

Testing & training data files are loaded from https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv) and https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv) respectively.

```
setwd("C:/Users/Louis/SkyDrive/Documents/R-files/CourseraMachineLearning")
if (!file.exists("training.csv")) {
  dataset_url1 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
  download.file(dataset_url1, "training.csv")
}
if (!file.exists("testing.csv")) {
  dataset_url2 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
  download.file(dataset_url2, "testing.csv")
}
training <- read.csv("training.csv", stringsAsFactors=F)
testing <- read.csv("testing.csv", stringsAsFactors=F)
training$classe <- as.factor(training$classe)
```

Summary of training data

Looking at the summary of the training data.

Pre-Processing

A number of the dimensions such as name and date stamps will be excluded.
the remaining dimensions are cast as numeric.

```
training1 <- training[, !names(training) %in% c("user_name", "cvtd_timestamp", "new_window",
                                                "raw_timestamp_part_1", "raw_timestamp_part_2")]
for(i in 1:(ncol(training1)-1)){
  training1[,i] <- as.numeric(training1[,i])
}
```

A number of dimensions contain a large % of NA's.
Dimensions with more than 10,000 NA's are dropped.

```
drops <- NULL
for (i in 1:(ncol(training1)-1)) {
  if (sum(is.na(training1[ , i])) > 10000)
  {
    drops <- c(drops, names(training1)[i])
  }
}
training2 <- training1[,!(names(training1) %in% drops)]
```

The cleaned data is condensed to reduce the number of attributes using PCA to capture 80% of the variance
filtered for complete rows only.

```
complete.training <- training2[complete.cases(training2), ]
preProc <- preProcess(complete.training[,-ncol(complete.training)],method="pca", thresh = .8)
trainingPCA <- predict(preProc,complete.training[,-ncol(complete.training)])
trainingPCA <- cbind(trainingPCA, complete.training$classe)
names(trainingPCA)[ncol(trainingPCA)] <- "classe"
```

Finally the testing dataset is put throught the same transformation.

```
testing2 <- testing[ , (names(testing) %in% names(training2))]
testingPCA <- predict(preProc, testing2)
```

Training Models

Partitioning data for training & testing

The training data is partitioned into two datasets 60/40 split; to train and test prospective models.

```
inTrain <- createDataPartition(y=trainingPCA$classe, p=0.6, list=FALSE)
trainingOfTraining <- trainingPCA[inTrain,]
testingOfTraining <- trainingPCA[-inTrain,]
```

The testingOfTraining dataset will be used for cross validation.

Decision Tree

```
modelFitDT <- train(classe ~ .,method="rpart",  data=trainingOfTraining)
```

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

```
predict.DT <- predict(modelFitDT, testingOfTraining[,-length(testingOfTraining)])
matrix1<-confusionMatrix(testingOfTraining$classe, predict.DT)
```

The results of the cross validation indicate that the overall accuracy of the decision tree model is 0.4472.

Neural Net

```
modelFitNN <- train(classe ~ .,method="nnet",  data=trainingOfTraining)
predict.nnet <- predict(modelFitNN, testingOfTraining[,-length(testingOfTraining)])
matrix2<-confusionMatrix(testingOfTraining$classe, predict.nnet)
```

The results of the cross validation indicate that the overall accuracy of the neural net model is 0.6652.

Random Forest

Due the memory (RAM) limitations, the training dataset is sampled to give a more managable number of rows.

```
set.seed(123)
sample <- sample(nrow(trainingOfTraining), 5000)
sample <- sample[order(sample)]
modelFitRF <- train(trainingOfTraining$classe[sample] ~. , method = "rf", verbose=F, prox = T,
                   data = trainingOfTraining[sample, ])
predict.rf <- predict(modelFitRF, testingOfTraining[,-length(testingOfTraining)])
matrix3 <- confusionMatrix(testingOfTraining$classe, predict.rf)
```

The results of the cross validation indicate that the overall accuracy of the Random Forest model is 0.9416.

To further check the accuracy of the rf model, we test the model accuracy on the unused training data.

The indication accuracy when tested on the unused training data is 0.9435.

The results of the cross validation for the random forest of 0.9416 and 0.9435 indicate that this is the best fit model.

Testing on unseen data

```
predict(modelFitRF, testingPCA)
```

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