Machine Learning Prediction Assignment ~ Week 4

Sandip Khanvilkar

July 5, 2018

Introduction:

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Data:

The training data for this project are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv

The test data are available here:

https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

Step 1:Loading and Preprocessing of Data

```r
##install.packages("caret")
##install.packages("numDeriv")
library(caret)
library(rpart)
##install.packages("rpart.plot")
library(rpart.plot)
##install.packages("RGtk2")
library(RColorBrewer)
library(RGtk2)
##install.packages("rattle")
library(rattle)
##install.packages("rpart.plot")
library(rpart.plot)
#library(randomForest)
##install.packages("munsell")
##sessionInfo()
install.packages("fancyRpartPlot")
library(munsell)
```

```r
UrlTrain <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
UrlTest  <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

# download the datasets
dt_training <- read.csv(url(UrlTrain))
dt_testing  <- read.csv(url(UrlTest))
```

Step 2 Cleaning the data

```r
features <- names(dt_testing[,colSums(is.na(dt_testing)) == 0])[8:59]
# Only use features used in testing cases.
dt_training <- dt_training[,c(features,"classe")]
dt_testing <- dt_testing[,c(features,"problem_id")]
dim(dt_training); dim(dt_testing);
```

Removed all columns that contains NA and also removed features that are not in the testing dataset. The features containing NA are the variance, mean and standard devition (SD) within each window for each feature. Since the testing dataset has no time-dependence, these values are useless and can be disregarded. We will also remove the first 7 features since they are related to the time-series or are not numeric

Step 3 Partitioning the Datasets

Following the recommendation in the course Practical Machine Learning, we will split our data into a training data set (60% of the total cases) and a testing data set (40% of the total cases; the latter should not be confused with the data in the pml-testing.csv file). This will allow us to estimate the out of sample error of our predictor.

```r
set.seed(12345)
inTrain <- createDataPartition(dt_training$classe, p=0.6, list=FALSE)
training <- dt_training[inTrain,]
testing <- dt_training[-inTrain,]
dim(training)
dim(testing)
```

Step 4 Building the decision tree model

```r
modFitDT <- rpart(classe ~ ., data = training, method="class")
fancyRpartPlot(modFitDT)
```

Using Decision Tree, we shouldn't expect the accuracy to be high. In fact, anything around 80% would be acceptable.

Step 5 Predicting with the Decision Tree Model

Modeling based on only one predictor variable does not seem to be sufficient and good enough as we have other predictor variables that might affect MPG and therefore affect the difference in MPG by transmission. So the univariate model in this case is only part of the picture. Therefore in this part of the analysis we use multivariable linear regression to develop a model that includes the effect of other variables.

```
set.seed(12345)
prediction <- predict(modFitDT, testing, type = "class")
confusionMatrix(prediction, testing$classe)
```

Step 6 Building the Random Forest Model

Using random forest, the out of sample error should be small. The error will be estimated using the 40% testing sample. We should expect an error estimate of < 3%.

```
set.seed(12345)
modFitRF <- randomForest(classe ~ ., data = training, ntree = 1000)
```

Step 7 Predicting with the Random Forest Model

```
prediction <- predict(modFitRF, testing, type = "class")
confusionMatrix(prediction, testing$classe)
```

Step 8 Predicting on the Testing Data (pml-testing.csv) ~ Decision Tree Prediction

```
predictionDT <- predict(modFitDT, dt_testing, type = "class")
predictionDT
```

You can observe that all the variables now are statistically significant. This model explains 84% of the variance in miles per gallon (mpg). Now when we read the coefficient for am, we say that, on average, manual transmission cars have 2.94 MPGs more than automatic transmission cars. However this effect was much higher than when we did not adjust for weight and qsec.

Step 9 Random Forest Prediction

```
predictionRF <- predict(modFitRF, dt_testing, type = "class")
predictionRF
```

Step 10 Submission File

Prepare the submission.

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(predictionRF)
```

Conclusion:

As can be seen from the confusion matrix the Random Forest model is very accurate, about 99%. Because of that we could expect nearly all of the submitted test cases to be correct. It turned out they were all correct.