Machine Learning Assignmnet

Kim Monks

Tuesday, March 17, 2015

Course Project Assignment

"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 Sources

The training and test data for this project are available here:

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

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

The original source of the data is: 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.

Project Objectives

The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases. 1. Your submission should consist of a link to a Github repo with your R markdown and compiled HTML file describing your analysis. Please constrain the text of the writeup to < 2000 words and the number of figures to be less than 5. It will make it easier for the graders if you submit a repo with a gh-pages branch so the HTML page can be viewed online (and you always want to make it easy on graders :-). 2. You should also apply your machine learning algorithm to the 20 test cases available in the test data above. Please submit your predictions in appropriate format to the programming assignment for automated grading. See the programming assignment for additional details.

Reproducibility

Libraries Needed

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.3
```

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

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.1.3
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.1.3
```

```
library(parallel)
library(doParallel)
```

```
## Warning: package 'doParallel' was built under R version 3.1.3
```

```
## Loading required package: foreach
## Loading required package: iterators
```

```
cl <- makeCluster(detectCores() - 1)
registerDoParallel(cl)
```

Seed for random generator

```
set.seed(12345)
```

Data Importing into R

```
train <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!",""))

test <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!",""))
```

Training-Set cleaning and pre-processing

Remove dataset NAs and low variance.

```
nearzero <- nearZeroVar(train, saveMetrics = TRUE)
train <- train[, !nearzero$nzv]
```

Variables with more than 30% missing values are removed

```
rem <- sapply(colnames(train), function(x) if(sum(is.na(train[, x])) > 0.30*nrow(train))    {return(TRU
}else{
return(FALSE)
}
)
train <- train[, !rem]
train <- train[, -(1:6)]
```

Correlation analysis:

```
corr <- caret::findCorrelation(cor(train[, -53]), cutoff=0.8)
names(train)[corr]
```

```
##  [1] "accel_belt_z"     "roll_belt"        "accel_belt_y"
##  [4] "accel_dumbbell_z" "accel_belt_x"     "pitch_belt"
##  [7] "accel_arm_x"      "accel_dumbbell_x" "magnet_arm_y"
## [10] "gyros_arm_y"      "gyros_forearm_z"  "gyros_dumbbell_x"
```

Many variables are highly correlated. PCA will be used in the pre-processing. Model Specification and Cross Validation

In order to avoid overfitting and to reduce out of sample errors, TrainControl is used to perform 3-fold cross validation.

```
tc <- trainControl(method = "cv", number = 3, verboseIter=FALSE , preProcOptions="pca", allowParallel=T
```

Two models are estimated: Random forest and a Logit Boosted model.

```
rf <- train(classe ~ ., data = train, method = "rf", trControl= tc)
logitboost <- train(classe ~ ., data = train, method = "LogitBoost", trControl= tc)
```

```
## Loading required package: caTools
```

```
nnet <- train(classe ~ ., data = train, method = "nnet", trControl= tc)
```

```
## Loading required package: nnet
```

```
## # weights:  179
## initial  value 35738.495760
## iter  10 value 29658.410187
## iter  20 value 29390.932130
## iter  30 value 29268.803133
## iter  40 value 29122.039330
## iter  50 value 28465.788633
## iter  60 value 28127.625809
## iter  70 value 27662.713379
## iter  80 value 27550.518749
## iter  90 value 27489.443764
## iter 100 value 27463.351873
## final  value 27463.351873
## stopped after 100 iterations
```

```
rpart <- train(classe ~ ., data = train, method = "rpart", trControl= tc)
```

Accuracy comparision

```
model <- c("Random Forest", "LogitBoost", "Neural Net", "Recursive Partitioning")
Accuracy <- c(max(rf$results$Accuracy),
        max(logitboost$results$Accuracy), max(nnet$result$Accuracy), max(rpart$result$Accuracy))


performance <- cbind(model,Accuracy)
knitr::kable(performance)
```

model	Accuracy
Random Forest	0.992814188118823
LogitBoost	0.901833895999734
Neural Net	0.399397229724908
Recursive Partitioning	0.498776649319801

Random forest provides the best results and will provide the predictions for the submission.

Prediction of "classe" variable for the test set

```
rfPred <- predict(rf, test)
logitboostPred <- predict(logitboost, test)
nnetPred <- predict(nnet, test)
rpartPred <- predict(rpart, test)
```

Checking if the models give same predictions - Boost and NNET do not

```
prediction <- data.frame(cbind(rfPred, rpartPred))
prediction$same <- with(prediction, rfPred == rpartPred)
colnames(prediction) <- c("Random Forest", "Recursive Partitioning", "Same Prediction")
knitr::kable(prediction)
```

Random Forest	Recursive Partitioning	Same Prediction
2	3	FALSE
1	1	TRUE
2	3	FALSE
1	1	TRUE
1	1	TRUE
5	3	FALSE
4	3	FALSE
2	1	FALSE
1	1	TRUE
1	1	TRUE
2	3	FALSE
3	3	TRUE
2	3	FALSE
1	1	TRUE
5	3	FALSE
5	1	FALSE
1	1	TRUE
2	1	FALSE
2	1	FALSE
2	3	FALSE

Conclusions

The random forest model provides accuracy and, accordingly, the predictions