Human Activity Recognition

Peer Assessment: Practical Machine Learning Project (Coursera) – by: M Turk

Friday, May 22, 2015

Background

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

The test data are available here: URL:https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv

Require Libraries

```
require(caret)
```

Load and Preprocess data

Loading the data and updating the values of '#DIV/0!' to NA.

Further preprocessing of the NA's into meaningful values for machine learning models will be to assign the mean of the field to the NA's in that field and using the nearZeroVar function in caret to remove the near zero variance predictors.

```
# Load Data and preprocess na's.
training_set <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!",""))
testing_set <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!",""))

# Assigning the column mean to the NA values for each column.
for (col in names(training_set)){if(class(training_set[[col]]) %in% c("integer", "numeric")){
                avg<-mean(training_set[[col]], na.rm = TRUE)
                training_set[[col]][is.na(training_set[[col]])]<-avg}}

# Remove 1:5 columns of irrelivent data.
```

```
training_set <- training_set[,-c(1:5)]

# Remove near zero variance predictors.
training_set <- training_set[,-nearZeroVar(training_set)]
```

Create Train/Test Datasets

Create a training and testing dataset partition using the createDataPartition function in caret.

```
inTrain <- createDataPartition(y=training_set$classe, p=0.7, list=FALSE)
train <- training_set[inTrain,]
test <- training_set[-inTrain,]
```

Train Model

Use randomForest function in caret and the "train" dataset partition to train model.

```
cv <- trainControl(method="cv", number=3)
modelFit <- train(train$classe~., data=train, method="rf", trControl=cv)
```

Model Results

Use the model trained from the "train" dataset partition to predict "test" dataset partition and analyse the predicted results.

```
results <- predict(modelFit, test)
confusionMatrix(results, test$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    5    0    0    0
##          B    0 1130    2    0    0
##          C    0    3 1024    5    1
##          D    0    1    0  959    1
##          E    0    0    0    0 1080
##
## Overall Statistics
##
##                Accuracy : 0.9969
##                  95% CI : (0.9952, 0.9982)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9961
##  Mcnemar's Test P-Value : NA
```

```
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9921   0.9981   0.9948   0.9982
## Specificity            0.9988   0.9996   0.9981   0.9996   1.0000
## Pos Pred Value         0.9970   0.9982   0.9913   0.9979   1.0000
## Neg Pred Value         1.0000   0.9981   0.9996   0.9990   0.9996
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1920   0.1740   0.1630   0.1835
## Detection Prevalence   0.2853   0.1924   0.1755   0.1633   0.1835
## Balanced Accuracy      0.9994   0.9958   0.9981   0.9972   0.9991
```

Model Analysis

Analysis indicates that in the 3 cross-validation that have been made, the accuracy is between 97% and 99%. Therefore the out-of-sample error is between 1% and 3%, and will be sufficient to submit the results.

Prediction

Use model to predict testing set.

```
prediction <- predict(modelFit, testing_set, type="raw")
prediction
```

```
##  [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
```

Create files for submission

```
for (i in seq(20)){
  fileName<-paste("problem",i,".txt",sep="_")
  write.table(prediction[i],file=fileName,quote=FALSE,row.names=FALSE,col.names=FALSE)
}
```