Machine Learning Course Project

David Halldorson

June 16, 2018

Background

Wearable devices have become much more common and they can be used to predict the activity the person is doing. The goal of this project is 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. For this project, we will use various machine learning algorithms to come up with an accurate prediction model.

Data Preperation

First we will load the data in for both the training and test data sets. Next we will clean up the data set by removing all columns that contain “NA”. We will also remove date and categorization columns that will not be used in the training. Finally we will split the training set into a set for training and another for validation.

```
#1. Load in the training and test sets
train<-read.csv('pml-training.csv', header=TRUE,sep=",", na.strings=c('NA',"#DIV/0!"))
test<-read.csv('pml-testing.csv', header=TRUE,sep=",", na.strings=c('NA',"#DIV/0!"))

#2. Cleanup the data sets by removing columns that have at least one NA
train<-train[, apply(is.na(train), 2, sum)==0]
test<-test[, apply(is.na(train), 2, sum)==0]

train<-train[,!names(train) %in% c('X','user_name', 'raw_timestamp_part_1', 'raw_timestamp_part_
2', 'cvtd_timestamp', 'new_window', 'num_window')]
test<-test[, !names(train) %in% c('X','user_name', 'raw_timestamp_part_1', 'raw_timestamp_part_
2', 'cvtd_timestamp', 'new_window', 'num_window')]

#3. Split into train and validate sets
inTrain = createDataPartition(train$classe, p = 0.75)[[1]]
trainset = train[ inTrain,]
validset= train[-inTrain,]
```

Training the Models

To perform the model training, we will attempt this with 2 different methods CART (rpart) and Bagged CART (treebag). After fitting the model on the training set we will use the model to predict the result of the validation set. Finally we will check the accuracy on each comparing the predicted value with the actual value in the validation set.

```
#4a. Fit using CART (rpart)
tic()
set.seed(12998)
modelFitRP<-train(classe~.,data=trainset, method="rpart")
predictRP<-predict(modelFitRP,newdata=validset)
cmRP<-confusionMatrix(predictRP, validset$classe)
cmRP$overall[1]
```

```
##  Accuracy
## 0.4934747
```

```
#4b. Fit using TreeBag
set.seed(12998)
modelFitTB<-train(classe~.,data=trainset, method="treebag")
predictTB<-predict(modelFitTB,newdata=validset)
cmTB<-confusionMatrix(predictTB, validset$classe)
cmTB$overall[1]
```

```
## Accuracy
## 0.983075
```

Of these two models, the Bagged Cart (treebag) produces much better performance with an accuracy of 0.983075. Therefore of these models we will choose the Bagged Cart

Prediction

Using the chosen Bagged Cart model we will now predict the category for each of the 20 test cases.

```
predictTest<-predict(modelFitTB,newdata=test)
predictTest
```

```
##  [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
```

Note on Additional Models

An attempt was also made to perform the same fitting using Random Forest (rf) and Stochastic Gradient Boosting (gdm), and some others. The performance of these were less than ideal on this many dependent variables. In some cases they failed and in others they worked but took much too long to complete. It is likely that one of these would give a better fit on a more powerful machine with more available memory.