Practical Machine Learning Course Project

Hasan Shojaei

19 October 2018

The main goal of this project is to fit different models to data from accelerometers used on 6 participants, and predict the manner in which they did the exercise. We will use k-fold cross-validation to optimize tuning parameters for each model. To estimate the expected out-of-sample accuracy for each model, we divide the training data into two subsets (training1 & training2), train the model (including cross validation) using training1 data set, and then test the skill of the model using training2 data set. At the end, the best performing model will be used to perform prediction for 20 different test cases.

Out of the 4 different models used (decision tree, generalized boosted regression, random forest, support vector machine) the **random forest** model gave the highest out-of-sample accuracy followed very closely by **generalized boosted regression**. Therefore the random forest model was used to perform the classification on the test set.

Loading Required Libraries

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.4
```

```
library(rattle)
```

```
## Warning: package 'rattle' was built under R version 3.4.4
```

Loading Data

```
training <- read.csv(url("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"))
testing <- read.csv(url("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"))
```

The dimensions of the *training* and *testing* data frames are:

```
dim(training)
```

```
## [1] 19622   160
```

```
dim(testing)
```

```
## [1]  20 160
```

There are 160 columns in both *training* and *testing* data frames, while there are 19622 and 20 rows in the *training* and *testing* data frames respectively.

Cleaning Data

Many of the columns in the data set have mostly *NA* values. IWe will remove those columns, along with the columns that have nearly zero variability because they will not be useful for machine learning. The first two columns (X & user_name) are also not useful for machine learning and will be removed.

```
# removing columns with mostly NA values
mostlyNA <- sapply(training, function(x) mean(is.na(x))) > 0.95
training <- training[, !mostlyNA]
testing <- testing[, !mostlyNA]
dim(training)
```

```
## [1] 19622    93
```

```
# removing columns with nearly zero variability
n0v <- nearZeroVar(training)
training <- training[, -n0v]
testing <- testing[, -n0v]
dim(training)
```

```
## [1] 19622    59
```

```
# removing first 2 columns
training <- training[, -c(1,2)]
testing <- testing[, -c(1,2)]
dim(training)
```

```
## [1] 19622    57
```

We ended up with 57 columns (56 predictors and 1 response variable) after cleaning up the data.

Model Building

We will train and test four different models: Decision Tree, Generalized Boosted Regression, Random Forest, and Support Vector Machine. But before building any model, we divide the *training* data into 2 subsets: *training1* to train the models on, and *training2* to do an internal test of the models before choosing and applying the best model to the *testing* data set. This will allow us to have an estimate of the out-of-sample accuracy before applying the model to the test set.

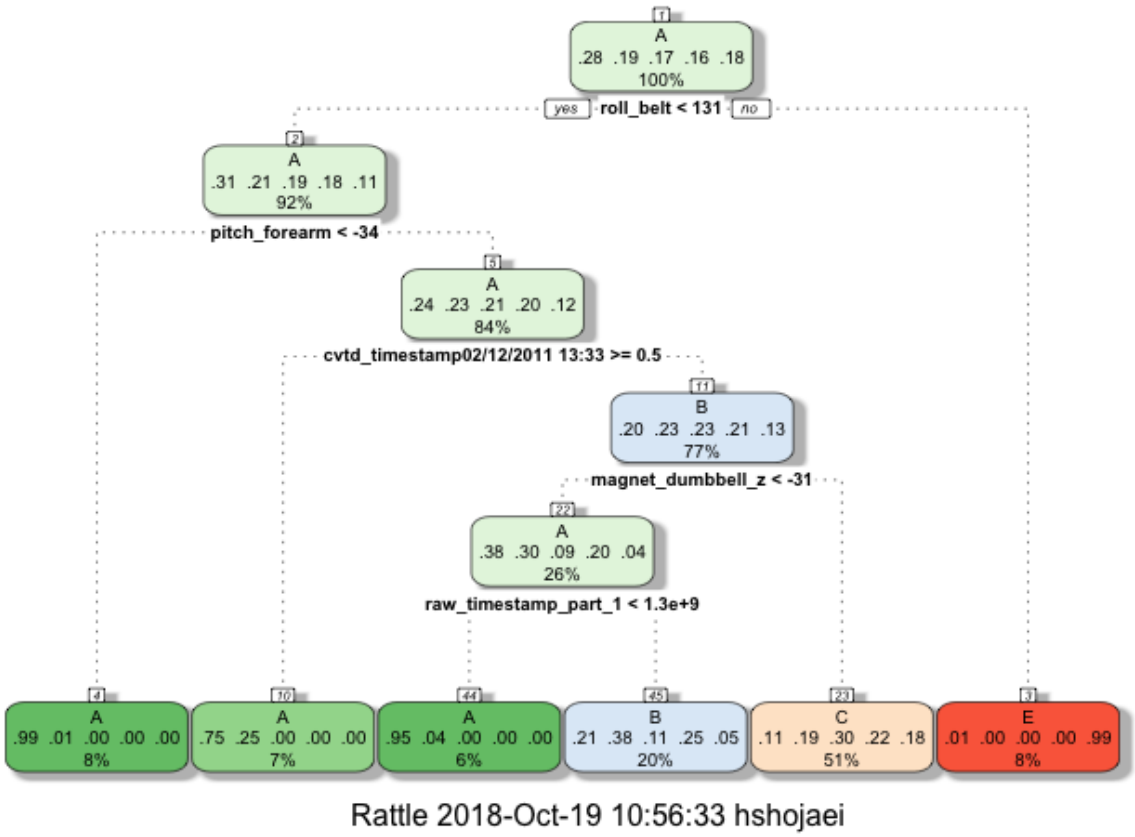
```
set.seed(1000)
inTrain <- createDataPartition(y=training$classe, p=0.7, list = FALSE)
training1 <- training[inTrain,]
training2 <- training[-inTrain,]

# use 5-fold cross-validation to optimize tuning parameters
fitControl <- trainControl(method="cv", number=5)
```

Decision Tree

```
modFit_dt <- train(classe ~ ., data=training1, method="rpart", trControl=fitControl)
pred_dt <- predict(modFit_dt, newdata = training2)
conf_mat_dt <- confusionMatrix(training2$classe, pred_dt)
```

The decision tree is visualized below:

```
fancyRpartPlot(modFit_dt$finalModel)
```



To examine the out-of sample accuracy of the model, we investigate the confusion matrix:

```
conf_mat_dt$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1094  230  347    0    3
##          B  111  434  594    0    0
##          C    2  136  888    0    0
##          D    0  281  683    0    0
##          E    0   80  514    0  488
```

```
conf_mat_dt$overall[1]
```

```
##  Accuracy
## 0.4934579
```

We can see that the overall accuracy of the descision tree model is only 0.4935, which is not satisfactory.

Generalized Boosted Regression

```
modFit_gbm <- train(classe ~ ., data=training1, method="gbm", trControl=fitControl, verbose=FALSE)
pred_gbm <- predict(modFit_gbm, newdata = training2)
conf_mat_gbm <- confusionMatrix(training2$classe, pred_gbm)
```

To examine the out-of sample accuracy of the model, we investigate the confusion matrix:

```
conf_mat_gbm$table
```

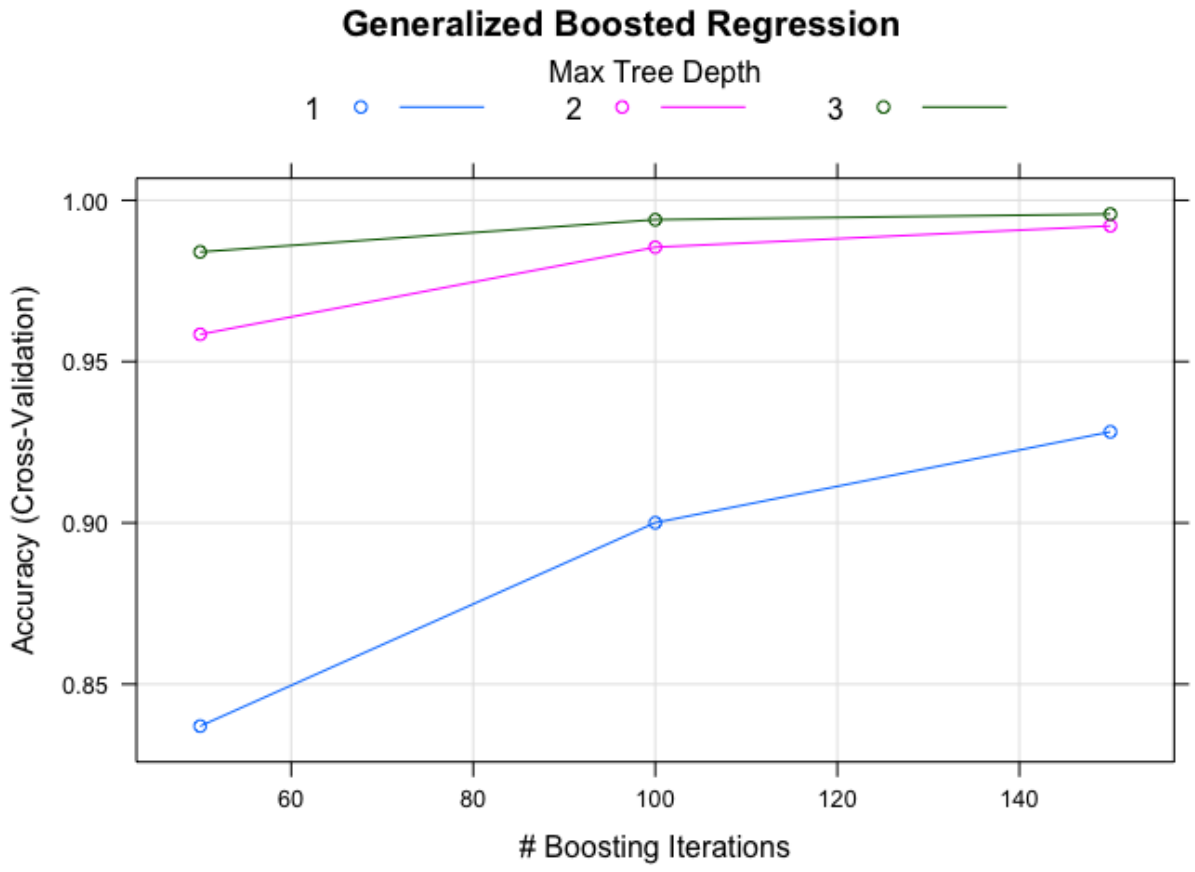
```
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    1    0    0    0
##          B    1 1135    3    0    0
##          C    0    1 1017    8    0
##          D    0    0    4  957    3
##          E    0    0    0    2 1080
```

```
conf_mat_gbm$overall[1]
```

```
##  Accuracy
## 0.9960918
```

The overall accuracy of the generalized boosted regression model is 0.9961 which is very high.

```
plot(modFit_gbm, main="Generalized Boosted Regression")
```



We see that the best model performance is obtained with 150 boosting iterations and a maximum tree depth of 3.

Random Forest

```
modFit_rf <- train(classe ~ ., data=training1, method="rf", trControl=fitControl, verbose=FALSE)
pred_rf <- predict(modFit_rf, newdata = training2)
conf_mat_rf <- confusionMatrix(training2$classe, pred_rf)
```

To examine the out-of sample accuracy of the model, we investigate the confusion matrix:

```
conf_mat_rf$table
```

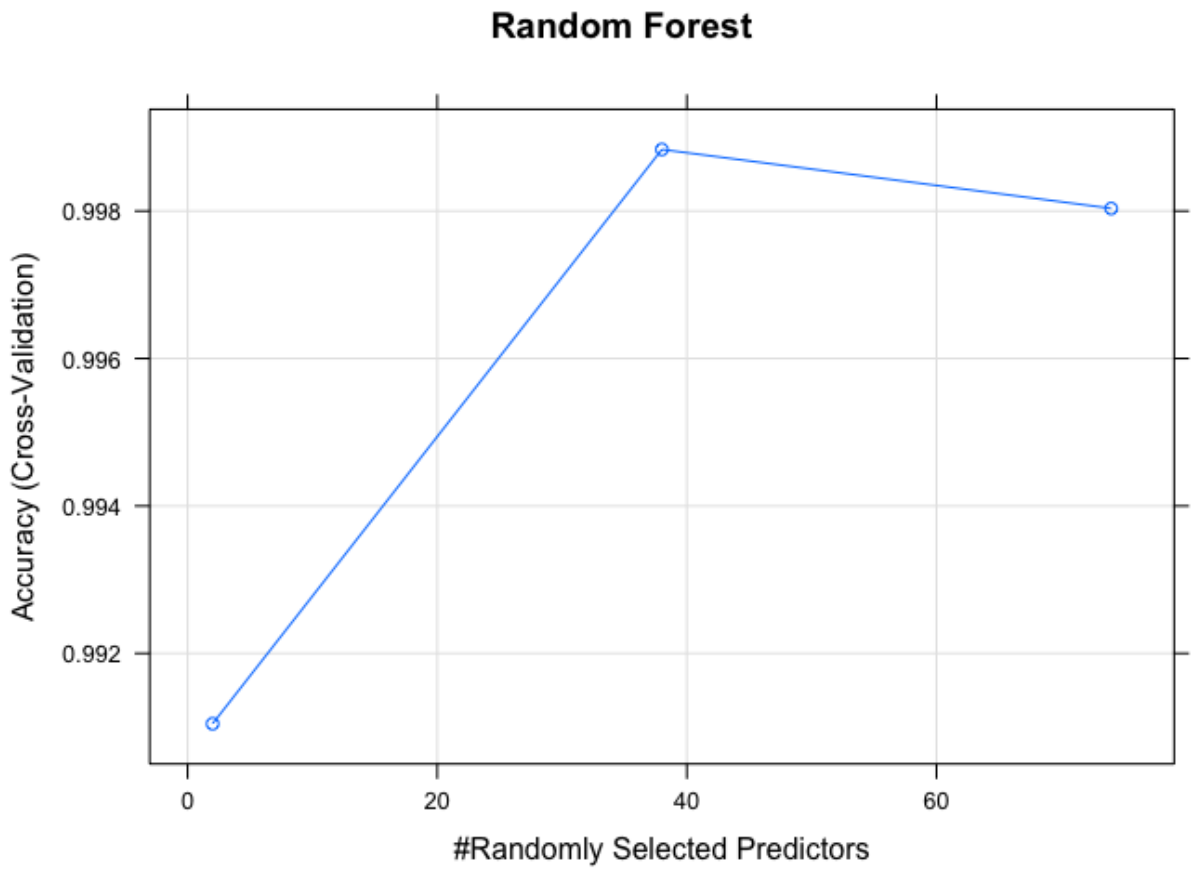
```
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B    1 1138    0    0    0
##          C    0    1 1024    1    0
##          D    0    0    0  964    0
##          E    0    0    0    0 1082
```

```
conf_mat_rf$overall[1]
```

```
##  Accuracy
## 0.9994902
```

The overall accuracy of the random forest model is 0.9995 which is very high.

```
plot(modFit_rf, main="Random Forest")
```



The highest accuracy is obtained with 38 randomly selected predictors. It should be noted that predictions

with 2, 38 and 74 randomly selected predictors all give accuracies above 99%.

Support Vector Machine

```
modFit_svm <- train(classe ~ ., data=training1, method="svmLinear", trControl=fitControl)
pred_svm <- predict(modFit_svm, newdata = training2)
conf_mat_svm <- confusionMatrix(training2$classe, pred_svm)
```

To examine the out-of sample accuracy of the model, we investigate the confusion matrix:

```
conf_mat_svm$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1623   42    9    0    0
##          B  129  940   70    0    0
##          C    4   74  940    8    0
##          D    1    3   89  834   37
##          E    0    0    4   72 1006
```

```
conf_mat_svm$overall[1]
```

```
##  Accuracy
## 0.9079014
```

The overall accuracy of the support vector machine model is 0.9079 which acceptable, but not as good as random forest or generalized boosted regression.

Model Selection

Random Forest model gave the highest out-of-sample accuracy out of the 4 tested models. Therefore we will move forward with random forest method for prediction. We don't need to re-train the model using all training data (training1 + training2) because the estimated out-of-sample accuracy is already very hight ($>$ 99%).

Prediction

In this section we predict the type of excercise for the test set using our trained random forest model:

```
pred_rf_testdata <- predict(modFit_rf, newdata = testing)
print(pred_rf_testdata)
```

```
##  [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
```

Summary

We fit 4 different models to data from accelerometers used on 6 participants, and predicted the manner in which they did the exercise. We used 5-fold cross-validation to optimize tuning parameters for each model. We divided the training data into training1 & training2 subsets, trained the model using training1, and tested the skill of the model using training2 data set.

Our analysis showed that the **random forest** model gave the highest out-of-sample accuracy followed very closely by **generalized boosted regression**. Therefore the random forest model was used to perform the classification on the test set.