Practical Machine Learning Course Project

Andy Sullivan

5/25/2020

Overview

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).

Objective

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.
1. Use any of the other variables to predict with.
2. Create a report describing how the model was built, how cross validation was used, the expectation of the sample error, and the choices made.
3. Use the prediction model to predict 20 different test cases.

Data

Load and Edit data

Load the training and validation data sets from the files pml-training.csv and pml-testing.csv respectively.

When we look at the data, we notice that many of the columns in the training data set do not contain values (NA values). For simplicity sake we will eliminate these columns from our analysis. We also must eliminate the same columns in the validation data set.

Next, also looking at the data, we can determine that the first 7 variables have little to no bearing on the classe variable as they are more descriptive of the event (person and time) rather than a measurement.

```
train_data <- read.csv("data/pml-training.csv")
validation_data <- read.csv("data/pml-testing.csv")
# eliminate NA columns from our analysis.
# We also must eliminate the same columns in the test data set.
train_data = train_data[,!sapply(train_data, function(x) mean(is.na(x)))>0.5]
```

```r
keep <- as.vector(names(train_data))
validation_data = validation_data[,(names(validation_data) %in% keep)]
# eliminate the first 7 columns from data sets
train_data <- train_data[,-c(1:7)]
validation_data <- validation_data[,-c(1:7)]
```

Cross Validation

Training and Test Data Sets

Next, we will split the data into training a testing sets. Both sets will be based on train data

```r
set.seed(45034)
inTrain = createDataPartition(y=train_data$classe,p=0.7, list = FALSE)
training = train_data[ inTrain,]
testing = train_data[-inTrain,]
```

Removing Zero Covariates

Some variables have no variability at all. These variables are not useful when we want to construct a prediction model. We can use the nearZeroVar function to eliminate those variables that have no variability to them.

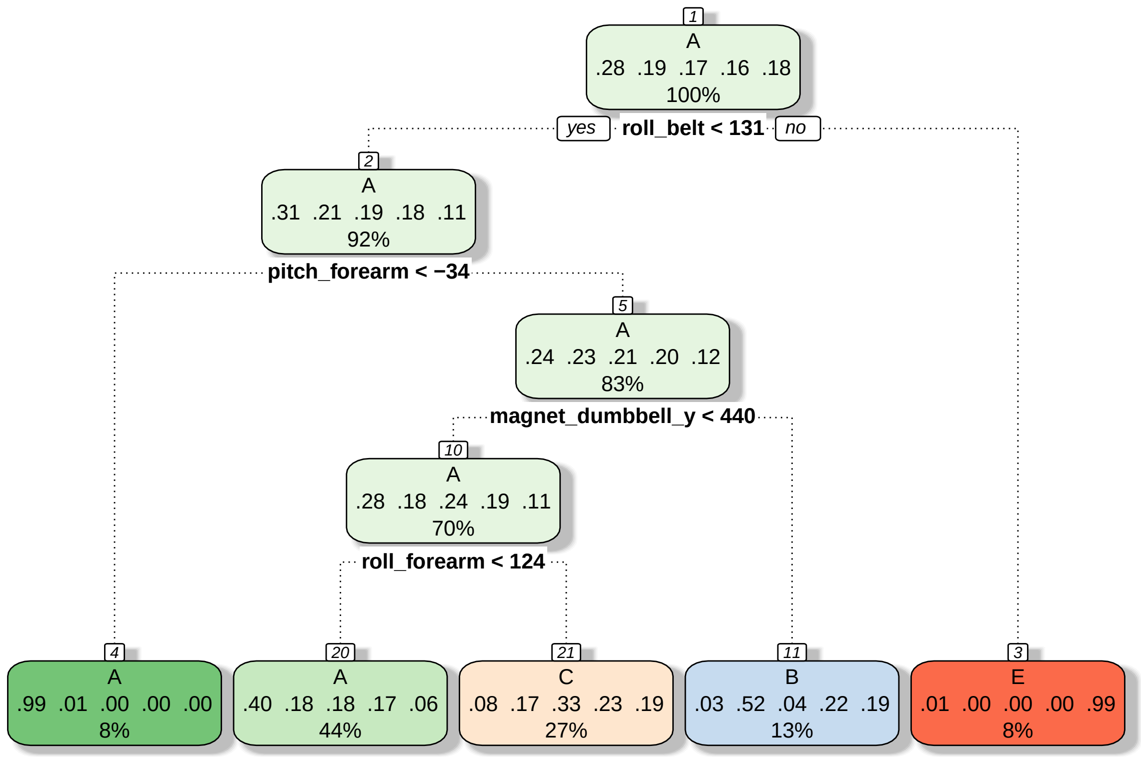
```r
omit_near_zero <-nearZeroVar(training,saveMetrics=FALSE)
training <- training[,-omit_near_zero]
testing <- testing[,-omit_near_zero]
```

Prediction Models

Regression and Classification Trees

The first model that we will attempt to utilize is the Regression and Classification Trees (or rpart). We train the data against the training data and then predict against the testing data. Using this model we can create a simple Classification Tree.

```r
# fit classification tree as a model
rpart_Fit <- train(classe ~ .,method="rpart",data=training)
# create the classification tree
fancyRpartPlot(rpart_Fit$finalModel)
```



Rattle 2020−May−27 14:18:20 AS16651

```
# predict against the testing data
prd_rpart_Fit <- predict(rpart_Fit,newdata=testing)
# create a confustion matrix to determine how well we fit the testing data
cm_rpart_Fit <- confusionMatrix(prd_rpart_Fit, testing$classe)
```

While the Classification tree looks very simple, this does not give us a high degree of Accuracy. The accuracy for the Regression and Classification Trees model is 49.58 percent

Bootstrap Aggregating (Bagging)

The next model that we will attempt to utilize is Bootstrap Aggregating or Bagging. In this specific example we will use the Bagging method called treebag. We train the data against the training data and then predict against the testing data.

```
# fit classification tree as a model
treebag_Fit <- train(classe ~ .,method="treebag",data=training)
# predict against the testing data
prd_treebag_Fit <- predict(treebag_Fit,newdata=testing)
# create a confustion matrix to determine how well we fit the testing data
cm_treebag_Fit <- confusionMatrix(prd_treebag_Fit, testing$classe)
```

The accuracy for the Bagging model is 98.5 percent

Boosting

The next model that we will attempt to utilize is Boosting. In this specific example we will use the Boosting method called gbm (boosting with trees) We train the data against the training data and then predict against the testing data.

```
# fit classification tree as a model
gbm_Fit <- train(classe ~ .,method="gbm",data=training, verbose=FALSE)
# predict against the testing data
prd_gbm_Fit <- predict(gbm_Fit,newdata=testing)
# create a confusion matrix to determine how well we fit the testing data
cm_gbm_Fit <- confusionMatrix(prd_gbm_Fit, testing$classe)
```

The accuracy for the Boosting model is 96.04 percent

Random Forest

The next model that we will attempt to utilize if Random Forest. In this specific example we will use the Random Forest method rf. We train the data against the training data and then predict against the testing data.

```
# fit classification tree as a model
rf_Fit <- train(classe ~ .,method="rf",data=training, prox=TRUE)
# predict against the testing data
prd_rf_Fit <- predict(rf_Fit,newdata=testing)
# create a confusion matrix to determine how well we fit the testing data
cm_rf_Fit <- confusionMatrix(prd_rf_Fit, testing$classe)
```

The accuracy for the Random Forest model is 99.2 percent

Accuracy of All Models

The table below shows results for the 4 prediction models. While Bagging and Boosting have very high accuracy levels, the highest accuracy level is the Random Forest method rf

```
AccuracyResults <- data.frame(
  Model = c('RPART', 'TREEBAG', 'GBM', 'RF'),
  Accuracy = rbind(round(cm_rpart_Fit$overall["Accuracy"] * 100,2),
                   round(cm_treebag_Fit$overall["Accuracy"] * 100,2),
                   round(cm_gbm_Fit$overall["Accuracy"] * 100,2),
                   round(cm_rf_Fit$overall["Accuracy"] * 100,2))
)
print(AccuracyResults)
```

```
##     Model Accuracy
## 1   RPART    49.58
## 2 TREEBAG    98.50
## 3     GBM    96.04
## 4      RF    99.20
```

Predicting the Validation Group

Finally, use the Random Forest (rf) model to predict the values of the 20 different test cases

```
prd_validation_data <- predict(rf_Fit, newdata=validation_data)
Predicted_values <- data.frame(c(1:20),prd_validation_data[1:20] )
colnames(Predicted_values) <- c("problem_number","answer")
print(Predicted_values)
```

```
##    problem_number answer
## 1               1      B
## 2               2      A
## 3               3      B
## 4               4      A
## 5               5      A
## 6               6      E
## 7               7      D
## 8               8      B
## 9               9      A
## 10             10      A
## 11             11      B
## 12             12      C
## 13             13      B
## 14             14      A
## 15             15      E
## 16             16      E
## 17             17      A
## 18             18      B
## 19             19      B
## 20             20      B
```