Practical Machine Learning Class Assignment Write-up

Project Overview  
  
The goal of this project is to predict the manner in which participants in an exercise did the exercise. There are 20 observations to predict, after determining the best approach to prediction using a training data set. The variable to predict is the "classe" variable.

## Step 1: Inspect training data, remove unneeded rows and fields

The first step is to visually inspect the training and test data. The classe variable appears to have five possibilities: A, B, C, D, and E. This suggests a classification prediction algorithm.

In the training data set, records with a value of “yes” in the new\_window field appear to be summary statistics for a group of data. These may not be useful for predicting individual observations. When these records are filtered out of the dataset, many other fields are either all blank or have values of all “NA”. These fields will not be useful in predicting the classe variable. Removing these fields reduces the number of columns in the training set from 160 to 60.

## Step 2: Analyze remaining variables for suitability as predictors

With several variables removed, the next step is the review the remaining variables for suitability as predictors. The nearZeroVar function is used to analyze the remaining variables, with particular attention paid to the percent Unique output. Selected variables that have very few unique values are shown in Table 1 . These variables will initially be left in as possible predictors, but may need to be removed:

Table 1

freqRatio percentUnique

X 1.000000 1.000000e+02

user\_name 1.102460 3.122398e-02

raw\_timestamp\_part\_1 1.000000 4.355745e+00

raw\_timestamp\_part\_2 1.000000 8.568901e+01

cvtd\_timestamp 1.002041 1.040799e-01

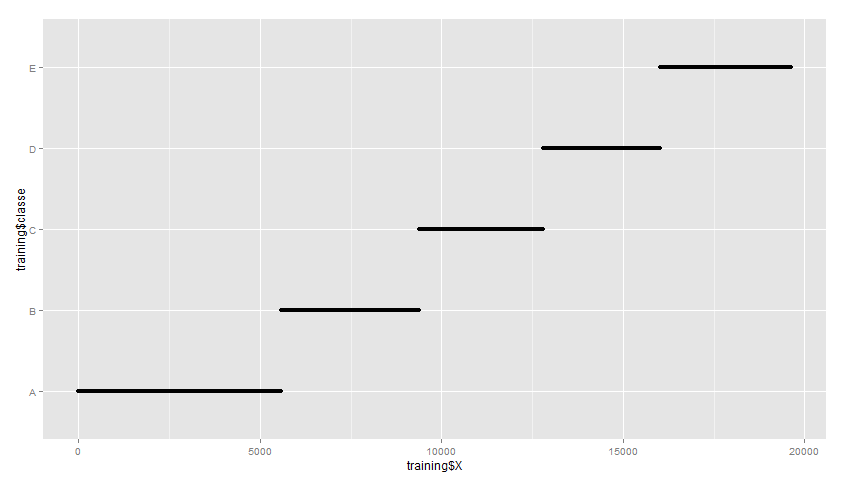
new\_window 0.000000 5.203997e-03

num\_window 1.000000 4.459825e+00

## Step 3: Run initial model to further analyze variables

With 60 variables remaining, an initial CART model is run on the full training data set using the rpart method in the caret package. The X variable (unlabeled in the .csv) is the sole variable used to create the tree. This variable appears to be a numbered index field. When plotted against the class variable in Figure 1, it can be seen that the class variable is sorted in order (A through E).

Figure 1 – index Variable vs. classe Variable



It appears probable that the dataset was originally constructed by sorting the data by the classe field and then adding the index variable. This index variable is removed from the data as it will not hold predictive value.

When a CART model is run without the “X” index field, the nodes are create by fields other than those listed in Table 1.

## Step 4: Cross-Validation Approach

One of the key outputs of the analysis is an estimate of the out-of-sample error. To accomplish this, a simple data partition is created in the training data. 90% of the data remains in the training data set. 10% is randomly selected to constitute a validation data set. The model will be trained on the training data, and used to predict the classe variable of the validation set. The accuracy of the validation set prediction will be the estimate of the out-of-sample error.

Note that a k-folds cross-validation approach was considered. However, it is unclear whether the accuracy results of the cross-validation method using caret would produce a true out-of-sample estimate.

## Step 5: Model Results

Three initial models are fitted to the training data, all using the caret package in R:

1. A classification and regression tree (CART) model
2. A random forest model
3. A gbm boosting model

Note that the seed was set to 1 in order to enable reproducible results.

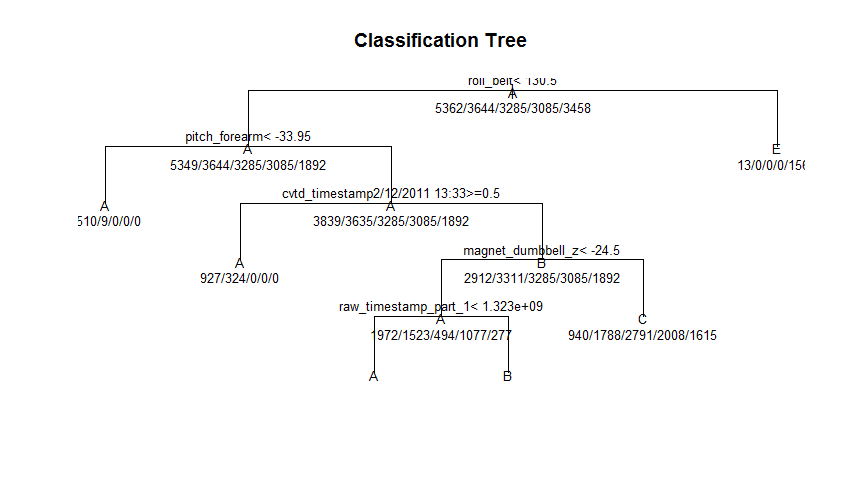
### CART Model Results

A CART model was fit to the training data using the rpart method in the caret package.

The in-sample accuracy using this method, reflecting the algorithm’s accuracy in predicting the classe variable of the training data set, is 49.86%. The out-of-sample accuracy, reflecting the algorithm’s accuracy in predicting the classe variable of the validation data set, is 48.68%. If the in-sample accuracy were substantially higher than the out-of-sample accuracy, the model may be affected by over-fitting. However, the similarity of the two accuracy rates provides evidence that the model is not affected by over-fitting.

The classification tree created using the rpart method on the training data is shown in Figure 2.

Figure 2 – rpart Classification Tree



### Random Forest Model Results

A random forest model was fit to the training data using the rpart method in the caret package.

However, this approach did not appear to resolve and was ended after six hours.

As a second option, the randomForest package was used to create a random forest model in R. This model resolved quickly. Both the in-sample and out-of-sample accuracy are 100%. The author has been unable to find an approach to plot the specific indicator variables and nodes within the randomForest package, but the overall statistics of both the training and validation set are shown in Figure 3 and Figure 4.

Figure 3 – Overall Statistics of randomForest Algorithm on Training Data

Overall Statistics

Accuracy : 1

95% CI : (0.9998, 1)

No Information Rate : 0.2847

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 1

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 1.0000 1.0000 1.0000 1.0000 1.0000

Specificity 1.0000 1.0000 1.0000 1.0000 1.0000

Pos Pred Value 1.0000 1.0000 1.0000 1.0000 1.0000

Neg Pred Value 1.0000 1.0000 1.0000 1.0000 1.0000

Prevalence 0.2847 0.1935 0.1744 0.1638 0.1836

Detection Rate 0.2847 0.1935 0.1744 0.1638 0.1836

Detection Prevalence 0.2847 0.1935 0.1744 0.1638 0.1836

Balanced Accuracy 1.0000 1.0000 1.0000 1.0000 1.0000

Figure 4 – Overall Statistics of randomForest Algorithm on Validation Data

Overall Statistics

Accuracy : 1

95% CI : (0.9903, 1)

No Information Rate : 0.2831

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 1

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: A Class: B Class: C Class: D Class: E

Sensitivity 1.0000 1.0000 1.0000 1.0000 1.0000

Specificity 1.0000 1.0000 1.0000 1.0000 1.0000

Pos Pred Value 1.0000 1.0000 1.0000 1.0000 1.0000

Neg Pred Value 1.0000 1.0000 1.0000 1.0000 1.0000

Prevalence 0.2831 0.1878 0.1799 0.1481 0.2011

Detection Rate 0.2831 0.1878 0.1799 0.1481 0.2011

Detection Prevalence 0.2831 0.1878 0.1799 0.1481 0.2011

Balanced Accuracy 1.0000 1.0000 1.0000 1.0000 1.0000

### GBM Model Results

Like the rf method within the caret package, the gbm method within caret was also not able to resolve to a model.

## Step 6: Model Selection

Given the results of the three models, the random forest model has been selected as the best approach to predict the classe variable in the test data set. The 100% accuracy of the random forest model on the validation data set suggests an accuracy of 100% when the model is applied to the test data, and an error rate of 0%.

## Step 7: Apply Model to Test Data

When the random forest model is applied to the 20 test data observations, the model predicts the classe variable for all 20 observations accurately.

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 :-).