Project: Practical Machine Learning

Analysis and Prediction of Data

Fong FH 19 May 2015

1. Synopsis

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. We make use any of the other variables in the dataset to predict the outcome of the **classe** variable through building a model. We used cross validation techniques to generate and compute the model efficiencies and out of sample error. Finally we test our prediction model through using it to predict 20 different test cases.

2. Introduction

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.

3. Data Processing

The data for this assignment come in the form of comma-separated-value files. They can be downloaded from the following web site:

Training Data: (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv). The file is about 12Mb in size.

Data for Test Cases: (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv). This data consist of 20 sets that we need to use to predict the outcome of the **classe** variable.

Detailed documentation of the data available at (http://groupware.les.inf.puc-rio.br/har). Here we can find details of how the tests were performed by the six test candidates.

3.1 Reading Data

To start the analysis, we read the training data files. We assume that the files are stored in the current R working directory.

```
# read in data file, replace the NA and blank spaces with NAs
pml_train <- read.csv("pml-training.csv", header=T, na.strings=c("","NA"))

library (rpart); library (rpart.plot);library (caret);library (caTools);library (randomForest)
options(scipen=9, digits=2)
```

3.2 Tidying Data

The data file contains many columns that were filled with NAs and blanks. For all of these we replace them with NAs throughout. In addition we remove the first seven columns of the data as they contain names, time and related informatio which are not useful for building the prediction model. Finally we remove all the columns of the training data set that contain NAs.

```
pml_train <- subset( pml_train, select = -c( 1 : 7 )) # remove columns 1 to 7
pml_train <- pml_train[,colSums(is.na(pml_train))<19216] # remove columns with NAs
```

The resultant training data frame consist of 19622 observations in 53 variables.

4. Prediction Models

4.1 Splitting Training Dataset

In order to test the accuracy of a prediction model we split the dataset into **training** and **testing** data sets.

```
set.seed (1000)
split = sample.split (pml_train$classe, SplitRatio=0.5)
train = subset(pml_train, split==TRUE)
test = subset(pml_train, split==FALSE)
```

4.2 CART Prediction Model

Next we extract only the relevant fields from the dataset and create two data frames for the total fatalities and injuries that arise from the environmental events.

```
# Generating CART model using pml_train
# pml_train_tree <- rpart (classe ~., data=pml_train, method="class", control=rpart.control(minsplit=20
pml_train_tree<-train(classe~.,data=train, method="rpart", trControl=trainControl(method="cv",number=20)
pml_train_tree
```

```
## CART
##
## 9811 samples
##   52 predictor
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (20 fold)
##
## Summary of sample sizes: 9320, 9319, 9321, 9320, 9321, 9320, ...
##
## Resampling results
##
##   Accuracy  Kappa  Accuracy SD  Kappa SD
##   0.92      0.89   0.015        0.019
##
## Tuning parameter 'cp' was held constant at a value of 0
##
```

The CART model can be used to predict the **classe** variable from the test data that we had split earlier.

```
predict_classe = predict (pml_train_tree, newdata=test)
```

We can compute the confusion matrix of the CART model as follows.

```
##    predict_classe
##        A    B    C    D    E
##   A 2687   44   11   30   18
##   B   72 1687   64   40   36
##   C   19   60 1570   47   15
##   D   25   23   59 1472   29
##   E   14   39   23   27 1700
```

The CART model was run with **20-fold cross-validation** on the training data set to improve model accuracy. The resultant accuracy on the testing data set can be obtained from the confusion matrix. We can see that the accuracy of the CART model is **92.92%** which also means that the **out-of-sample error** is **7.08%**.

4.3 Random Forest Prediction Model

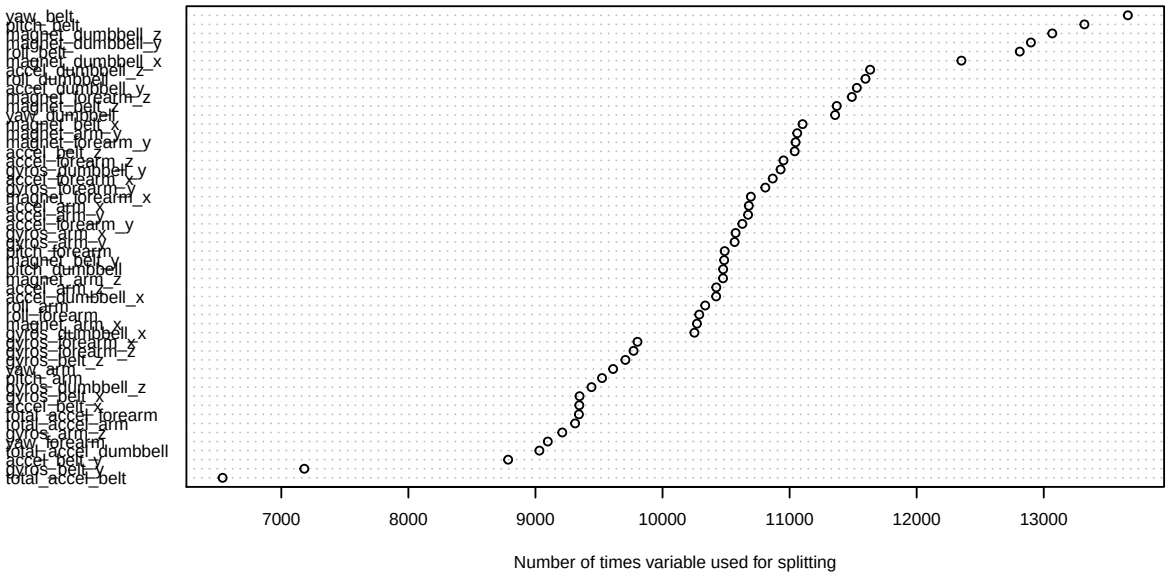
We can try to improve our prediction model through buiding a Random Forest model as follows.

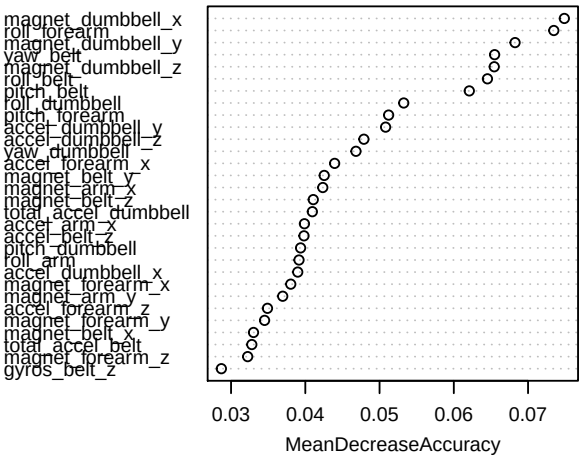
```
rf_model <- randomForest(classe ~ ., data = train, mtry = 2, importance = TRUE, do.trace = 100)
```

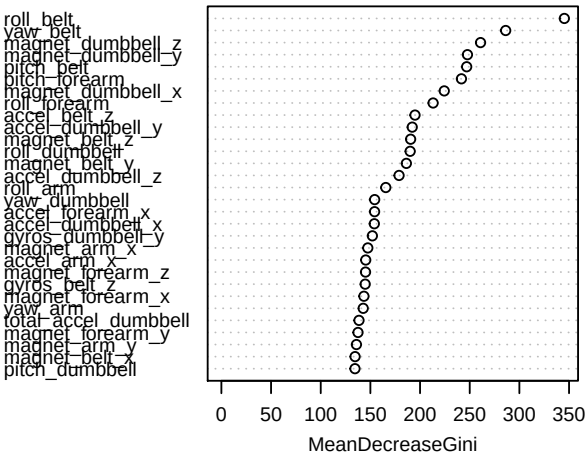
```
## ntree      OOB      1      2      3      4      5
##   100:   1.38%  0.25%  2.05%  1.69%  2.99%  0.67%
##   200:   1.15%  0.14%  1.53%  1.64%  2.74%  0.44%
##   300:   1.15%  0.18%  1.63%  1.40%  2.80%  0.44%
##   400:   1.05%  0.18%  1.42%  1.23%  2.61%  0.44%
##   500:   1.05%  0.18%  1.26%  1.34%  2.74%  0.39%

##
## Call:
##  randomForest(formula = classe ~ ., data = train, mtry = 2, importance = TRUE,      do.trace = 100)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 1.0%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 2785    4    0    0    1      0.0018
## B   17 1874    7    0    0      0.0126
## C    0   20 1688    3    0      0.0134
## D    0    0   42 1564    2      0.0274
## E    0    0    3    4 1797      0.0039
```

We note that from the nature of Random Forest models, there is no need for cross-validation to get an unbiased estimate of the test set error. It is estimated through the **OOB error** estimate of the rate (above).

To see the number of time each of the variables are selected for tree splitting, we can plot the following. We can see that **yaw_belt** is used most frequently, followed by **pitch_belt**, **magnet_dumbell_z** and so on. The next plot below shows the variable importance information in the random forest model.







With the random forest model, we can test it against the test data set we splitted in section 4.1. The resultant confusion matrix is below.

```
##    predictForest
##        A    B    C    D    E
##   A 2789    1    0    0    0
##   B   15 1864   20    0    0
##   C    0   13 1697    1    0
##   D    0    0   32 1575    1
##   E    0    0    0    3 1800
```

From the confusion matrix, we can see that the accuracy of the random forest model is **99.14%**. The **out-of-sample error** is **0.86%**.

This is a significant improvement over the CART model in terms of out-of-sample errors on the testing data set.

5. Predicting Test Cases

We can now proceed to use both the CART and Random Forest models to predict the **classe** variable for each of the 20 test cases in the *pml_test* dataset.

The test cases can be read in as follows.

```
pml_test = read.csv ("pml-testing.csv") # read in test cases
```

The test data set consist of 20 observations in 160 variables.

We can now do a prediction of the 20 test cases using the CART model we had generated.

```
predict_classe = predict (pml_train_tree, newdata=pml_test) # predict with CART
```

```
##  [1] B A B A A B D A A A B C B A E E A B B B
## Levels: A B C D E
```

Next, we perform predictions of the same 20 test cases using the Random Forest model.

```
predictForest = predict (rf_model, newdata=pml_test) # predict with random forest
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```

We can see that the CART and Random Forest models gave slightly different prediction result.

The Random Forest predicted values of **classe** was correct for all of the 20 test cases. The CART model resulted in two prediction errors which translates to an error rate of 10%.

Thus the Random Forest model is more accurate over the CART model.

6. Conclusion

This report showed that data prediction models can be built to accurately predict the outcomes of different data sets. In data prediction, we partioned the data in training and testing sets. The prediction model is developed using the training data while the testing data set is used to quantitfy the accuracy of the model on real data. Both **CART** and **Random Forest** models were built. For CART models, we used **multiple-folds cross-validation** to improve the model accuracy.

We have used both CART and Random Forest models to predict the 20 test cases. It was found that the Random Forest model built was able to predict all the 20 test cases accurately!