Practical Machine Learning Course Project

By Arvind Krishna

Executive summary

The goal of the project is to classify exercise behaviours of people using data from fitness devices like fitbit, jawbone up etc. A couple of model were built to classify - 1) Decision tree and (2) Random forest. The random forest model was found to be very accurate having an accuracy of classifying 995 out of 1000 cases correctly. The decision tree model's accuracy was somewhat lesser with the model classifying approximately 3 out of 4 cases correctly.

Thus the random forest model was chosen to predict the classification of 20 new cases. The model was able to classify all the cases correctly.

Solution approach

Reproducibility

A seed of 999 has been set for the work to be reproducible.

Modeling approach

Two models will be built to classify the exercising behaviour of people - 1) Using decision trees, (2) Using random forest. The model with higher accuracy will be chosen for prediction

Cross validation

The training data set will be split into training(75%) and testing(25%) data subsets. Model will be built on the training data and its accuracy will be tested on the testing data. The model with higer accuracy on the testing data will be chosen for prediction on the testing data provided.

Expected out-of-sample error

Accuracy is the proportion of correctly classified observation over the total sample in the subTesting data set. Expected accuracy is the expected accuracy in the out-of-sample data set (i.e. original testing data set). Thus, the expected value of the out-of-sample error will correspond to the expected number of missclassified observations/total observations in the Test data set

Data preparation

Libraries

Including all libraries required for the analysis. Note that library 'e1701' has been added since I am working on an older version of R (Version 0.98.507) and don't have the admin rights to update the version. 'e1701' is a dependancy of the 'caret' package.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.2
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.1.3
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 3.1.3
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.1.3
```

Data input

Taking testing and training datasets as input

```
trainUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
training <- read.csv(url(trainUrl), na.strings=c("NA","#DIV/0!",""))
testing <- read.csv(url(testUrl), na.strings=c("NA","#DIV/0!",""))
```

Cleaning data

Finding dimesntions of data

```
dim(training)
```

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

```
dim(testing)
```

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

By viewing the data we can see a lot of columns contain 'NAs', so removing those unnecessary columns

```
training<-training[,colSums(is.na(training)) == 0]
testing <-testing[,colSums(is.na(testing)) == 0]
```

Some variables like user_name, raw_timestamp_part_1 etc. are not useful in the model, so removing them from the data

```
training   <-training[,-c(1:7)]
testing <-testing[,-c(1:7)]
```

Now viewing the dimesions of the cleaned data

```
dim(training)
```

```
## [1] 19622    53
```

```
dim(testing)
```

```
## [1] 20 53
```

Partitioning data for cross validation

Setting a seed for the result to be reproducible

```
set.seed(999)
```

In order to perform cross-validation, the training data set is partionned into 2 sets: subTraining (75%) and subTest (25%). This will be performed using random subsampling without replacement.

```
subsamples <- createDataPartition(y=training$classe, p=0.75, list=FALSE)
subTraining <- training[subsamples, ]
subTesting <- training[-subsamples, ]
dim(subTraining)
```

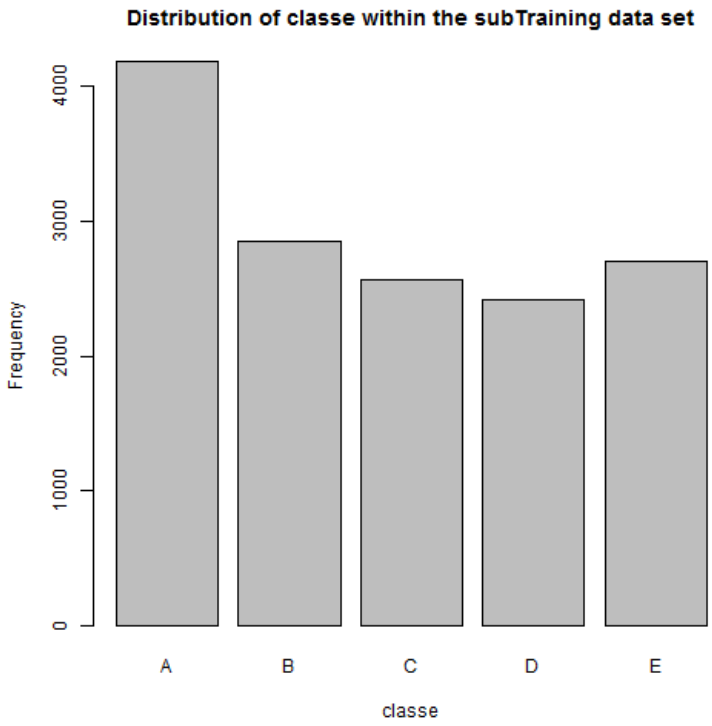
```
## [1] 14718    53
```

```
dim(subTesting)
```

```
## [1] 4904   53
```

Viewing the distribution of classe in our model training data set

```
plot(subTraining$classe, col="grey", main="Distribution of classe within the subTraining data set", xlab="classe",
ylab="Frequency")
```

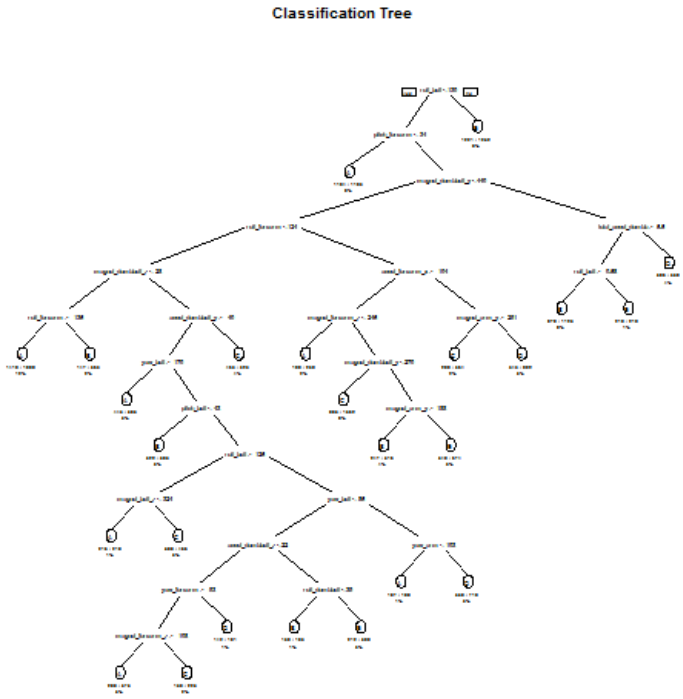


Data contains 5 classifications of exercise behaviours - A,B,C,D & E. The order of number of cases for all classifications is the same

Model development

Developing the 1st prediction model using decision trees

```
model1 <- rpart(classe ~ ., data=subTraining, method="class")
prediction1 <- predict(model1, subTesting, type = "class")
rpart.plot(model1, main="Classification Tree", extra=102, under=TRUE, faclen=0)
```



Viewing the model performance on the model test data

```
confusionMatrix(prediction1, subTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1280  155   18   47    4
##          B   42  566   98   62   93
```

```
##          C   41  127  662   60   76
##          D   19   62   57  559   69
##          E   13   39   20   76  659
##
## Overall Statistics
##
##                Accuracy : 0.7598
##                  95% CI : (0.7476, 0.7717)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6954
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9176   0.5964   0.7743   0.6953   0.7314
## Specificity            0.9362   0.9254   0.9249   0.9495   0.9630
## Pos Pred Value         0.8511   0.6574   0.6853   0.7298   0.8166
## Neg Pred Value         0.9662   0.9053   0.9510   0.9408   0.9409
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2610   0.1154   0.1350   0.1140   0.1344
## Detection Prevalence   0.3067   0.1756   0.1970   0.1562   0.1646
## Balanced Accuracy      0.9269   0.7609   0.8496   0.8224   0.8472
```

Developing the 2nd prediction model using random forest

```
model2 <- randomForest(classe ~. , data=subTraining, method="class")
prediction2 <- predict(model2, subTesting, type = "class")
confusionMatrix(prediction2, subTesting$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1393    3    0    0    0
##          B    1  943    4    0    0
##          C    0    3  850    5    0
##          D    0    0    1  797    0
##          E    1    0    0    2  901
##
## Overall Statistics
##
##                Accuracy : 0.9959
##                  95% CI : (0.9937, 0.9975)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9948
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9986   0.9937   0.9942   0.9913   1.0000
## Specificity            0.9991   0.9987   0.9980   0.9998   0.9993
## Pos Pred Value         0.9979   0.9947   0.9907   0.9987   0.9967
## Neg Pred Value         0.9994   0.9985   0.9988   0.9983   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2841   0.1923   0.1733   0.1625   0.1837
## Detection Prevalence   0.2847   0.1933   0.1750   0.1627   0.1843
## Balanced Accuracy      0.9989   0.9962   0.9961   0.9955   0.9996
```

Observations

We observe that the random forest model is more accurate than the decision tree model. Accuracy of random forest model is 0.995 (95% CI: (0.993,0.997)) while that of the decision tree model is 0.759 (95% CI: (0.747, 0.771)).

Conclusion

We chose the random forest model since it more accurate. The expected out-of-sample error is estimated at 0.005, or 0.5%. Thus in our testing data comprising of 20 cases, we can expect 0.005*20 i.e 0.1 cases to be missclassified. Thus, practically we should not get any missclassified cases in our testing data.

Prediction for testing data

Predicting classification of the 20 cases of testing data

```
predictfinal <- predict(model2, testing, type="class")
predictfinal
```

```
##  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
```

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
```

```
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}
pml_write_files(predictfinal)
```