Practical Machine Learning Project

Introduction

This Practical Machine Learning assignment is using data recorded from various individuals. The data is capturing the performance of the Unilateral Dumbbell Biceps curl. Each participant performed the exercised 5 different ways. The first, A, is considered the standard correct way and the other four (4), B to E is the exercise performed with common mistakes.
The purpose of this exercise is to use the data provided in training dataset to develop a model to predict how well each activitiy was performed by the six (6) individuals in the test dataset.

```
training <- read.csv("pml-training.csv")
test <- read.csv("pml-testing.csv")
```

Predictor Selection

The datasets were downloaded previously and placed in the project folder for this assignment.
The assignment indicated that only the accelerometer attached to the belt, forearm, arm and dumbbell are to be used to predict how well the activities were performed. The predictors were then selected by using only that had “accel” in it’s name and this was found using the grep function.
The columns were subsetted doing the following:

```
training_s <- training[,c(grep("accel", colnames(training)),160)]
training_sub <- training_s[,-(grep("^var", colnames(training_s)))]
```

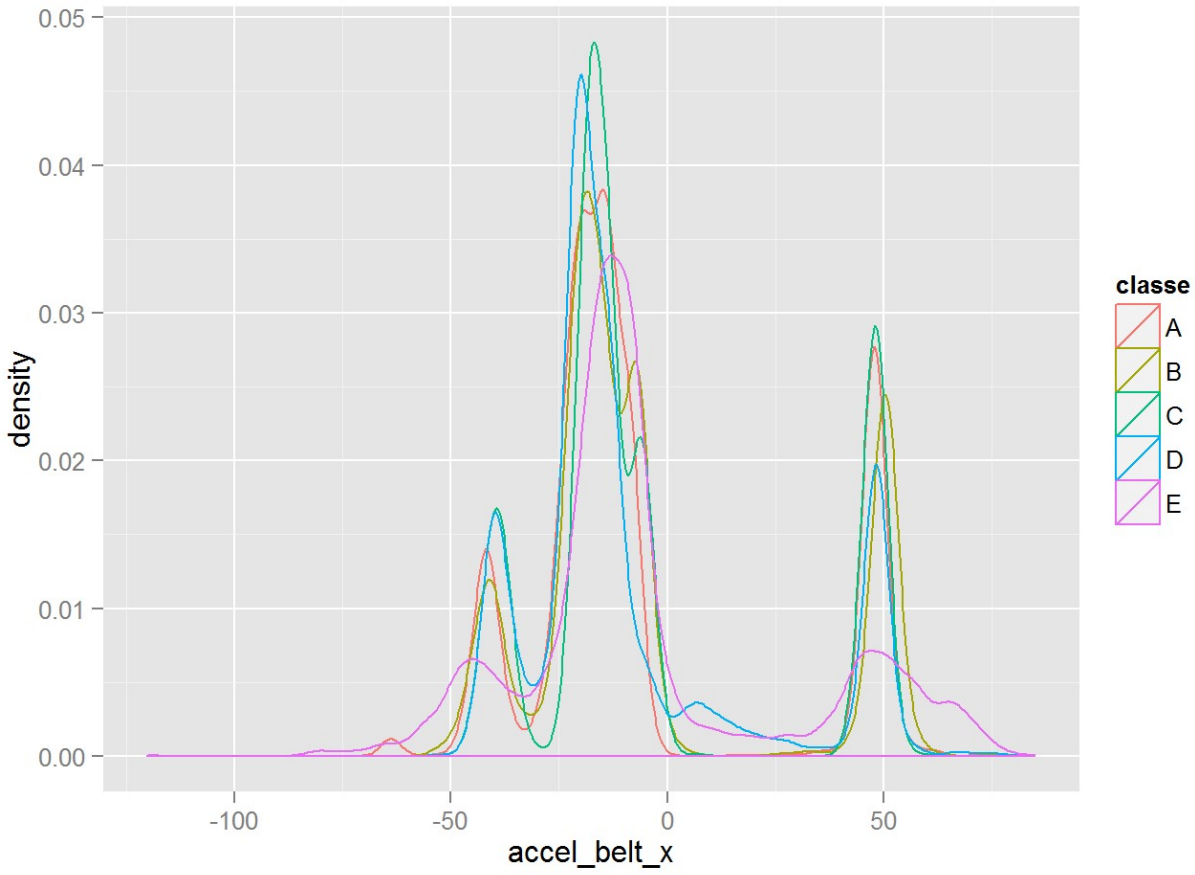
As demonstrated above, 17 columns were chosen from the dataset of 160. 16 of the columns would be used as predictors to predict the 17th, the column “Classe”. The columns names beginning with var were mostly populated with “NA” values which would affect the prediction of the classe variable so therefore it was removed.

Exploratory Analysis

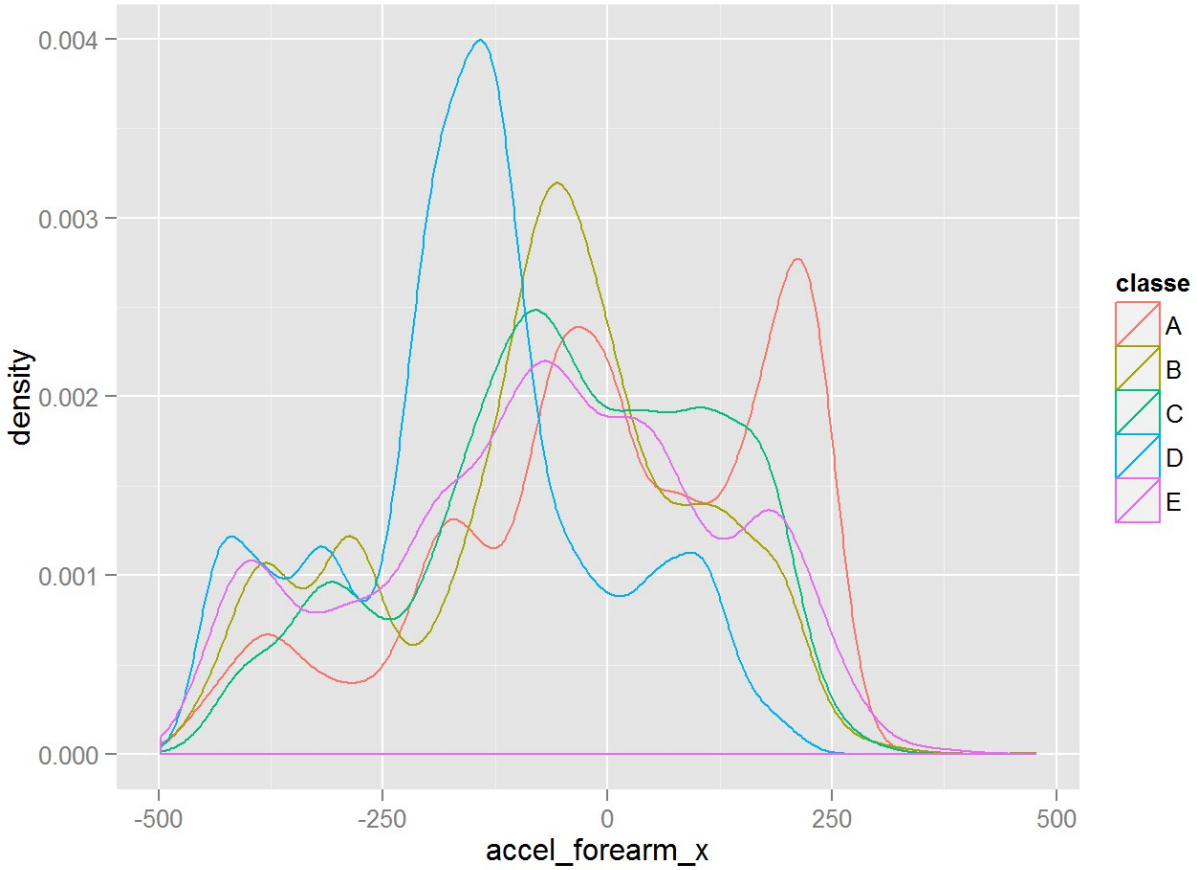
```
library(caret)
```

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

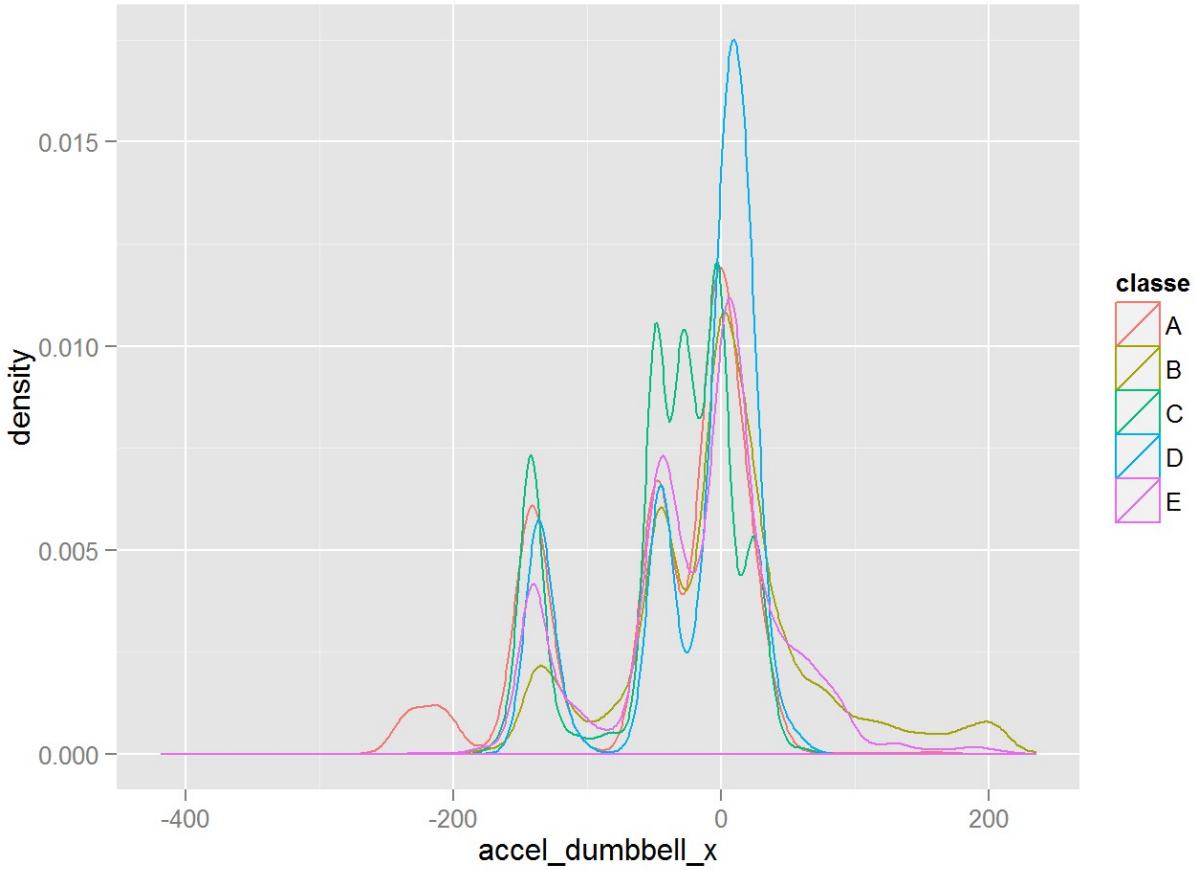
```
qplot(accel_belt_x, colour = classe, data=training_sub, geom = "density")
```




```
qplot(accel_forearm_x, colour = classe, data=training_sub, geom = "density")
```




```
qplot(accel_dumbbell_x, colour = classe, data=training_sub, geom = "density")
```




```
#do graphs
```

Model Selection and Training

Before the model can be developed and trained I ensured that reproducibility is possible. I ensured that the seed was set for the overall model and also created a vector of lists, seeds, to pass to the fitControl function to ensure that every step of the model training is reproducible.
Next, the training dataset was split, 75/25, into training and validation datasets using the createDataPartition function. The variable fitControl was created using the trainCOntrol function to help with the optimizing of the train function. oob was chosen as the method, the number of folds is 10 and this should be repeated 5 times.
Several algorithms were used for the assignment such as AdaBoost.M1 and AdaBag however the accuracy of those were very low, less than 50%. The random forest method was chosen to train the model because it was giving the best accuracy percentage. preProcessing in the training function was left out because the accuracy was better in the random forest model without it.

```
set.seed(1249)
seeds <- vector(mode = "list", length = 51)
for(i in 1:50) seeds[[i]] <- sample.int(1000, 22)
seeds[[51]] <- sample.int(1000, 1)
inTraining <- createDataPartition(training_sub$classe, p = .75, list = FALSE)
training_new <- training_sub[inTraining,]
validation <- training_sub[-inTraining,]
fitControl <- trainControl(method = "oob", number = 10, p = .75, seeds = seeds)
rfModelFit <- train(classe ~ ., data = training_new, method = "rf",trControl =
fitControl)
```

```
## Loading required package: randomForest
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
```

```
ans <- predict(rfModelFit, validation)
c_mat <- confusionMatrix(ans, validation$classe)
c_mat$table
```

```
##           Reference
## Prediction    A    B    C    D    E
##          A 1365   39   16   17    2
##          B    2  877   22    1    8
##          C   11   27  803   35    8
##          D   17    2   13  748    5
##          E    0    4    1    3  878
```

```
c_mat$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   9.524878e-01   9.398392e-01   9.461568e-01   9.582727e-01   2.844617e-01
## AccuracyPValue  McnemarPValue
##   0.000000e+00   3.880312e-08
```

After the model was trained into the variable rfModelFit and it was tested on the validation dataset, a confusionMatrix was done with the results, ans and the actual values in training_sub$classe. The model gave an in-sample Accuracy of 95%. Therefore an **out-of-sample Accuracy** would be less than **95%** and it's error rate would be greater than **5%**.