Practical Machine Learning - Course Project

The training dataset contains about 20000 observations and 160 variables, most of them numerical. We remove observation id, date and time stamps, as well as all variables which have a lot of missing values. This leaves us with 53 predictors and the outcome variable. We clean the test dataset removing the same variables as for the training dataset.

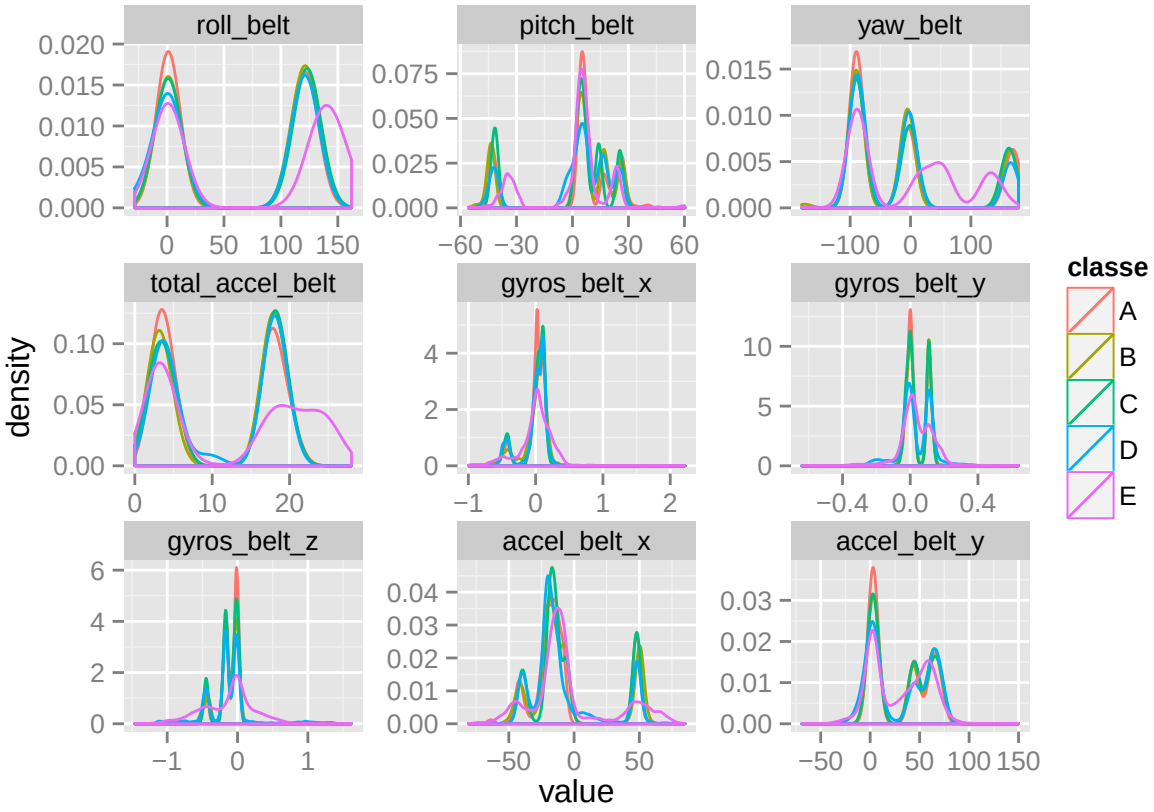
```
data.train = read.csv("D:/Courses/Coursera/Data Science - Practical Machine Learning/pml-training.csv",
data.numeric <- data.train[,sapply(data.train,function(x) is.numeric(x))]
data.count.na <- sapply(data.numeric, function(x) sum(is.na(x)))
data.no.na <- data.numeric[,data.count.na==0]
data.train.clean <- data.no.na[,5:56]
data.train.clean$user <- factor(data.train$user_name)
data.train.clean$classe <- factor(data.train$classe)
```

```
data.test = read.csv("D:/Courses/Coursera/Data Science - Practical Machine Learning/pml-testing.csv")
data.test.numeric <- data.test[,sapply(data.train,function(x) is.numeric(x))]
data.test.no.na <- data.test.numeric[,data.count.na==0]
data.test.clean <- data.test.no.na[,5:56]
data.test.clean$user <- factor(data.test$user_name)
data.test <- data.test.clean
```

Crossvalidation will be done as follows: The training dataset will be split into two parts. The first part is the training dataset for the model, using 60% of the observations. The second part is for validation of the model and estimating the out-of-sample error, using 40% of the observations.

```
library(caret)
data.partition <- createDataPartition(y=data.train.clean$classe,p=0.60,list=FALSE)
data.valid <- data.train.clean[-data.partition,]
data.train <- data.train.clean[data.partition,]
```

We have a look at the distribution of the first 9 variables to get an idea of the data.



We build a random forest prediction model on the training data set. Then we predict the outcome on both the training and the validation dataset and compute accuracies for both.

```
library(randomForest)
fit.rf <- randomForest(y=data.train$classe, x=data.train[,1:52], ntree=100)
pred.train <- predict(fit.rf, newdata=data.train[,1:52])
pred.valid <- predict(fit.rf, newdata=data.valid[,1:52])
acc.train <- sum(pred.train == data.train$classe)/length(data.train$classe)
acc.valid <- sum(pred.valid == data.valid$classe)/length(data.valid$classe)
acc.train
```

```
## [1] 1
```

```
acc.valid
```

```
## [1] 0.9941371
```

The accuracy on the training dataset is 100%. That indicates that we might be overfitting. The out-of-sample error for the model is computed as the accuracy on the validation dataset. The accuracy on the validation dataset is 99%, so the expected out-of-sample error is 1% and we do not seem to be overfitting.