Practical Machine Learning course Project

Kike Sedes

Background

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. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (see the section on the Weight Lifting Exercise Dataset).

Data

- The training data for this project are available here:https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv
- The test data are available here:https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv
- The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har

Goal

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. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

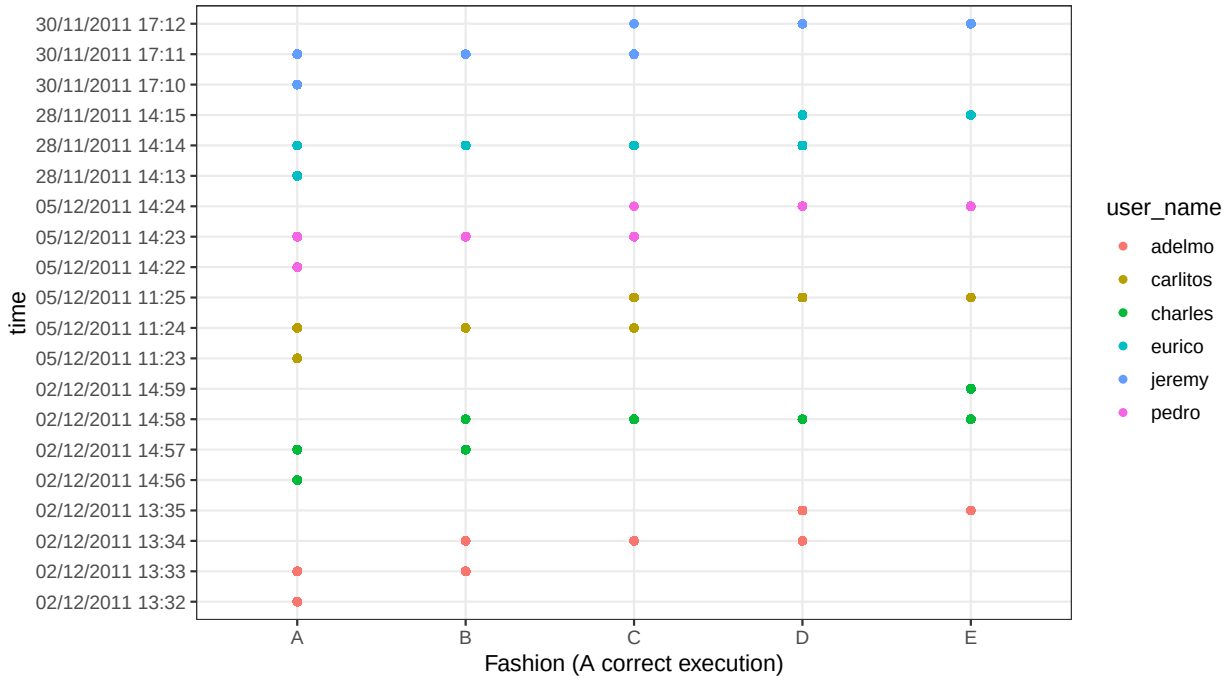
Data loading and exploring.

```
# Urls mentioned above
urlForTraining <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
urlForTesting  <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

# download datasets
training <- read.csv(url(urlForTraining))
validation  <- read.csv(url(urlForTesting))
str(training)
dim(training)
```

As we can see with "str", there are some variables with only NA's that we may exclude for our analysis. Let's create one plot with some significant factor variables identified in the str output

```
theme_set(theme_bw(base_size = 12))
plot1 <- qplot(classe, cvtd_timestamp, data=training, color=user_name,xlab = "Fashion (A correct execution)",
        ylab = "time")
plot1
```



It seems as if the participants in this study all performed these trials in temporal order. Moreover they started with exercice A (correct one)

Cleaning data & Cross Validation

As we saw in the lectures during the specialization, we will focus in cleaning our data in three different points:

- Variables with variance close to 0
- Variables with too many NA's in the validation dataset

Regarding Cross Validation, tet's rebember the key factor in Cross Validation:

- Use the trainingSet
- Split into training/test sets.
- Build the model on the training set.
- Evaluate on the test set.
- Repeate and average errors.

So, for this task, I am going to split the training dataset into "training" and "testing" sets. Moreover, I will only use "pml-testing.csv" at the end of the exercice,as a validation sample.

```
inTrain<- createDataPartition(training$classe, p=0.7, list=FALSE)
trainSet<- training[inTrain, ]
testSet<- training[-inTrain, ]

#remove variables with Nearly Zero Variance
zeroSigma_vars <- nearZeroVar(trainSet)
trainSet <- trainSet[, -zeroSigma_vars]
dim(trainSet)
```

```
## [1] 13737   108
```

```
#In order to predict classes in the validation sample, I will have to use features that are non-NA in the validation data set
all_zero_colnames <- sapply(names(validation), function(x) all(is.na(validation[,x])==TRUE))
nz_names <- names(all_zero_colnames)[all_zero_colnames==FALSE]

#remove ids & times
nz_names <- nz_names[-(1:7)]
nz_names <- nz_names[1:(length(nz_names)-1)]
trainSet <- trainSet[,c('classe', nz_names)]
dim(trainSet)
```

```
## [1] 13737    53
```

```
#I will set cross validation for each model with K = 3
fitControl <- trainControl(method='cv', number = 3)
```

ML model selection

I will try 3 differnt model algorithms and then I will check which provides the best accuracy:

- Decision trees with CART (rpart)
- Boosting(gbm)
- Random forest (rf)

Let's create or different models:

```
ml_model_cart <- train(classe ~ ., data=trainSet, trControl=fitControl,method='rpart')
save(ml_model_cart, file='./ModelFitCART.RData')

ml_model_gbm<- train(classe ~ ., data=trainSet, trControl=fitControl,method='gbm')
save(ml_model_gbm, file='./ModelFitGBM.RData')

ml_model_rf<-train(classe ~ ., data=trainSet,method='rf',ntree=100)
save(ml_model_rf, file='./ModelFitRF.RData')

testPredictionCART <- predict(ml_model_cart, newdata=testSet)
cMatrixCART <- confusionMatrix(testPredictionCART, testSet$classe)

testPredictionGBM <- predict(ml_model_gbm, newdata=testSet)
cMatrixGBM <- confusionMatrix(testPredictionGBM, testSet$classe)

testPredictionRF <- predict(ml_model_rf, newdata=testSet)
cMatrixRF <- confusionMatrix(testPredictionRF, testSet$classe)

AccuracyResults <- data.frame(Model = c('CART', 'GBM', 'RF'),Accuracy = rbind(cMatrixCART$overall[1], cMatrixGBM$overall[1], cMa
print(AccuracyResults)
```

Looking at the results, it is clear that the CART model does not work as well as boosting and random forest. As RF is the model with highest accuracy, I am going to select it for prediction with our validation dataset. Here are the complete results for the ramdom forest model:

```
cMatrixRF
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1670    7    0    0    0
##          B    2 1127    3    0    1
##          C    2    3 1019    4    3
##          D    0    2    4  959    2
##          E    0    0    0    1 1076
##
## Overall Statistics
##
##                Accuracy : 0.9942
##                  95% CI : (0.9919, 0.996)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9927
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9976   0.9895   0.9932   0.9948   0.9945
## Specificity            0.9983   0.9987   0.9975   0.9984   0.9998
## Pos Pred Value         0.9958   0.9947   0.9884   0.9917   0.9991
## Neg Pred Value         0.9990   0.9975   0.9986   0.9990   0.9988
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2838   0.1915   0.1732   0.1630   0.1828
## Detection Prevalence   0.2850   0.1925   0.1752   0.1643   0.1830
## Balanced Accuracy      0.9980   0.9941   0.9954   0.9966   0.9971
```

Prediction

Finally, it is time to work with our validation data sample ('pml-testing.csv') to predict a classe for each of the 20 observations.

```
testPredictionValidation <- predict(ml_model_rf, newdata=validation)
validationPredictionResults <- data.frame(problem_id=validation$problem_id,predicted=testPredictionValidation)
print(validationPredictionResults)
```

```
##    problem_id predicted
## 1           1         B
## 2           2         A
## 3           3         B
## 4           4         A
## 5           5         A
## 6           6         E
## 7           7         D
## 8           8         B
## 9           9         A
```

```
## 10          10          A
## 11          11          B
## 12          12          C
## 13          13          B
## 14          14          A
## 15          15          E
## 16          16          E
## 17          17          A
## 18          18          B
## 19          19          B
## 20          20          B
```