# Practical Machine Learning - Course Project

*Rodrigo Rezende*

*July 15, 2016*

## Introduction

Research on human activity recognition has traditionally focused on discriminating between different types of activities, that is, to predict which activity was performed at a specific point in time. The quality of execution of an activity has received little attention so far, even though it potentially provides useful information for a large variety of applications. In this analysis we are going to use the Weight Lifting Exercises Dataset [1] to try and predict the quality of execution of a specific exercise using data from wearable accelerometers. To generate this dataset a group of 6 participants wearing accelerometers on the belt, forearm, arm and dumbbell were asked to perform unilateral dumbbell biceps curls correctly and incorrectly in five different ways. The objective of this project is therefore to develop a predictive model capable of identifying the specific way in which the participant was performing the exercise (the *classe* variable in the dataset) using the data provided by the wearable accelerometers.

## Model Development

The Weight Lifting Excercises Dataset contains approximately 20,000 observations of 159 variables. These variables include the identification of the subject performing the exercise, a number of different timestamps and the readings from all the accelerometers at that specific point in time. For the observations in which the variable *new_window* has the value "yes" summary statistics like kurtosis, skewness, variance, standard deviation and average are also provided for each one of the accelerometer readings.

The first step in our analysis will be to split the dataset into the *training* and *testing* datasets that will be used for cross validation using approximately 80% of the observations for training and the balance for testing the model.

```
library(caret)

wleData = read.csv("pml-training.csv", row.names = 1)

set.seed(27182)
inTrain = createDataPartition(wleData$classe, p = 0.80, list = FALSE)

training = wleData[inTrain,]
testing = wleData[-inTrain,]
```

As already mentioned above, for the observations in which the variable *new_window* has the value "yes", and only for these specific observations, a large number of summary statistics is also provided. Considering that this type of observations comprise only a very small percentage of the dataset (approximately 2%) and that the summary statistics provided with them are effectively redundant because the raw data is available in the dataset, we can significantly reduce the number of variables in the problem by simply removing those observations. Once those observations have been removed, the variables related to the summary statistics can be easily removed using the nearZeroVar() function from the caret package, which will also remove a couple of other variables with low information content.

```
training = subset(training, new_window == "no")
nzv = nearZeroVar(training)
training = training[,-nzv]
testing = testing[,-nzv]
dim(training)
```

```
## [1] 15371    58
```

We were therefore able to reduce the number of variables in the problem from 159 to 58 by removing the small number of observations that contained redundant summary statistics. Now, it is reasonable to assume that many of the remaining variables will be highly correlated since they all represent concurrent measurements of the same body movement. This type of problem can be very efficiently addressed using Principal Component Analysis and for that we can use the preProcess() function of the caret package.

```
indClasse = ncol(wleData) - length(nzv)

preProc = preProcess(training[,-indClasse],
                     method = c("pca","center","scale"),
                     thresh = 0.98)

trainPC = predict(preProc, training[,-indClasse])
dim(trainPC)
```

```
## [1] 15371    35
```

Using PCA we were able to further reduce the number of predictors to 35 retaining 98% of the variance of the original data. We can now efficiently fit a Support Vector Machine to classify the types of movement being performed.

```
library(e1071)

modelFit = svm(training$classe ~ ., data = trainPC)
```

Once the model has been fitted we can then use our *testing* dataset to generate a confusion matrix and estimate the expected out of sample error.

```
testPC = predict(preProc, testing[,-indClasse])

confusionMatrix(testing$classe, predict(modelFit, testPC))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1107    9    0    0    0
##          B   55  688   15    0    1
##          C    0   27  654    3    0
##          D    0    0   54  588    1
##          E    0    0    0   27  694
##
## Overall Statistics
```

```
##
##                Accuracy : 0.9511
##                  95% CI : (0.9438, 0.9576)
##     No Information Rate : 0.2962
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.938
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9527   0.9503   0.9046   0.9515   0.9971
## Specificity            0.9967   0.9778   0.9906   0.9834   0.9916
## Pos Pred Value         0.9919   0.9065   0.9561   0.9145   0.9626
## Neg Pred Value         0.9804   0.9886   0.9787   0.9909   0.9994
## Prevalence             0.2962   0.1846   0.1843   0.1575   0.1774
## Detection Rate         0.2822   0.1754   0.1667   0.1499   0.1769
## Detection Prevalence   0.2845   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9747   0.9640   0.9476   0.9674   0.9944
```

Given the classification results above we would expected an accuracy of around 95% for out of sample observations.

It is important to note that a number of other classification models were tried, namely Random Forest, Linear Discriminant Analysis and Stochastic Gradient Boosting, but the Support Vector Machine was by far the fastest while still achieving very good accuracy results. For example, it took a couple of hours to train a Random Forest model to fit our training data while a Support Vector Machine was able to achieve similar levels of accuracy taking only 40 seconds to be trained.

## References

[1] Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013.