

# Peer-graded Assignment: Practical Machine Learning

*Steven Michiels*

*4/22/2020*

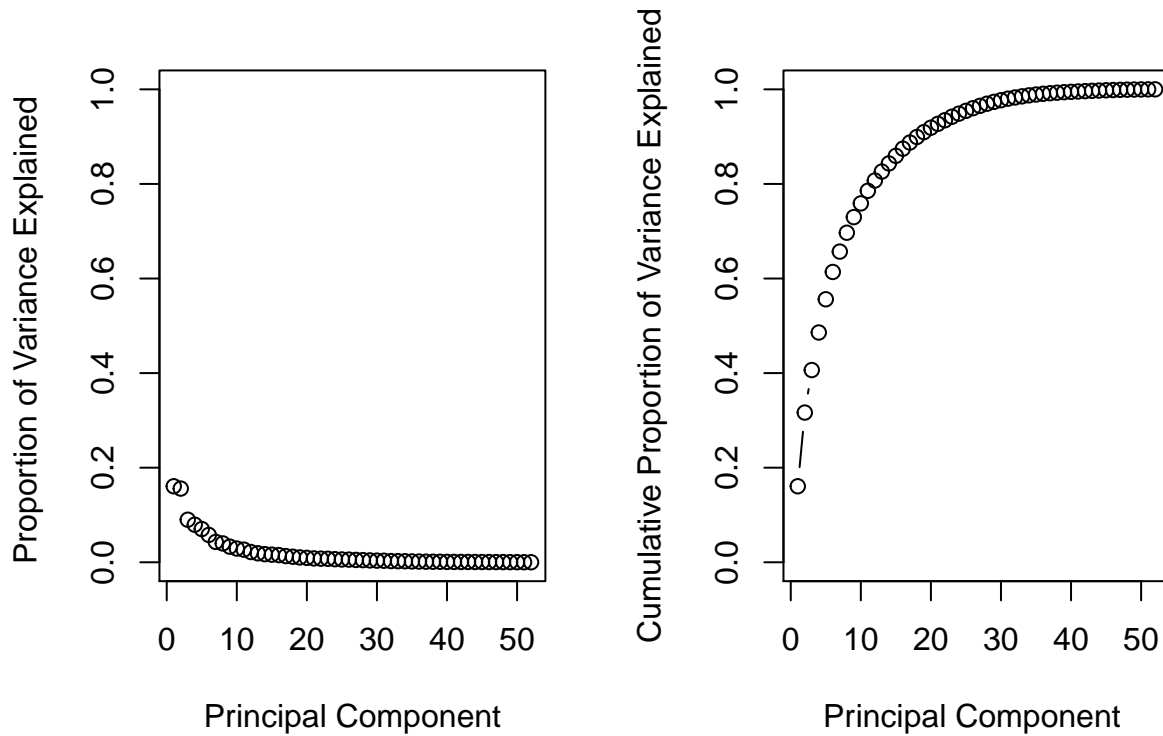
## Executive summary

The goal of this project is to **predict** the **correctness of training exercise**, based on accelerometer data from the belt, forearm, arm, and dumbbell of 6 participants. These were asked to perform barbell lifts correctly and incorrectly in 5 different ways. This training data was used to build a **random forest model** to **classify** the training exercise manner into **five different categories**: sitting-down, standing-up, standing, walking, and sitting. Five-fold cross-validation provided a **95% confidence interval for the out-of-sample accuracy of [99.3%, 99.5%]**. This is in line with the .4% out-of-bag error estimated from a bootstrapped random forest model. The random forest built using 5-fold cross validation was used to **predict the training exercise manner for 20 new observations**.

## Data cleaning and exploration

We load the training and testing data from the provided URL's. The training data consists of 19622 observations of 159 predictors and a 5-level factor outcome (A, B, C, D or E). A hold-out test set is already provided, consisting of 20 observations. Using the `is.na` command on all the columns, we see that many predictors barely contain any observation. We verify that these columns do not contain any useful information using the near-zero variance function and **filter out** these columns. We manually observe that the first seven columns neither contain any useful information, so we filter these out as well. This leaves us with **52 predictors**.

To get a feeling of the variance in the predictors, we perform a **principal component analysis**. The cumulative variance proportion is not dominated by few predictors, as instead many predictors make smaller contributions. Therefore, instead of explicit feature selection, we will make use of the **inherent feature importance selection** of the **random forest algorithm** during model building.



## Model building and evaluation

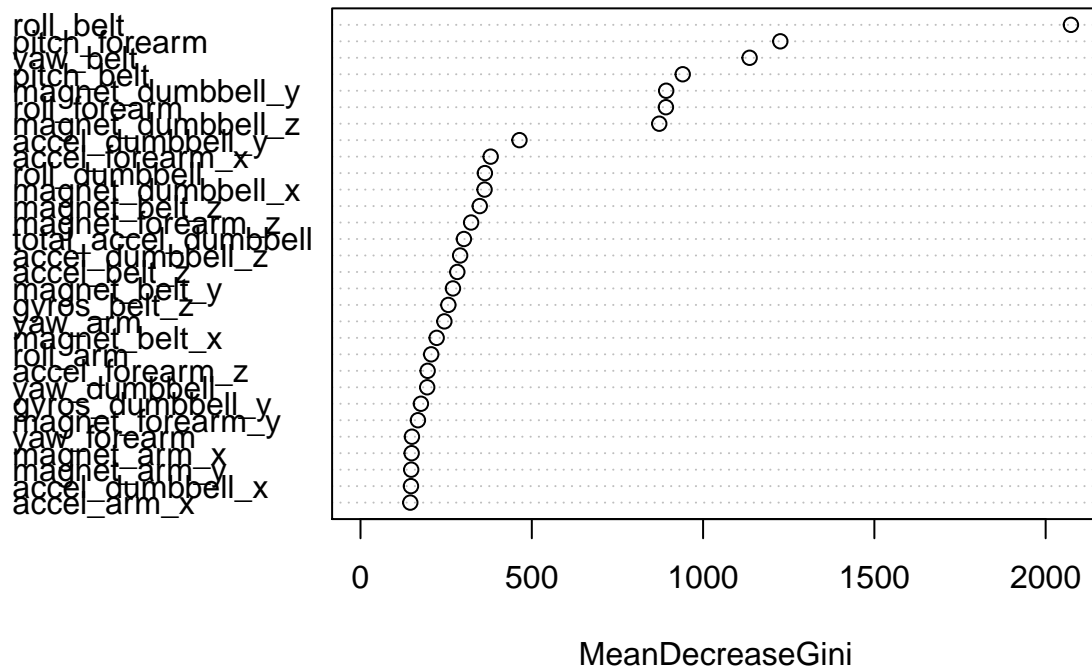
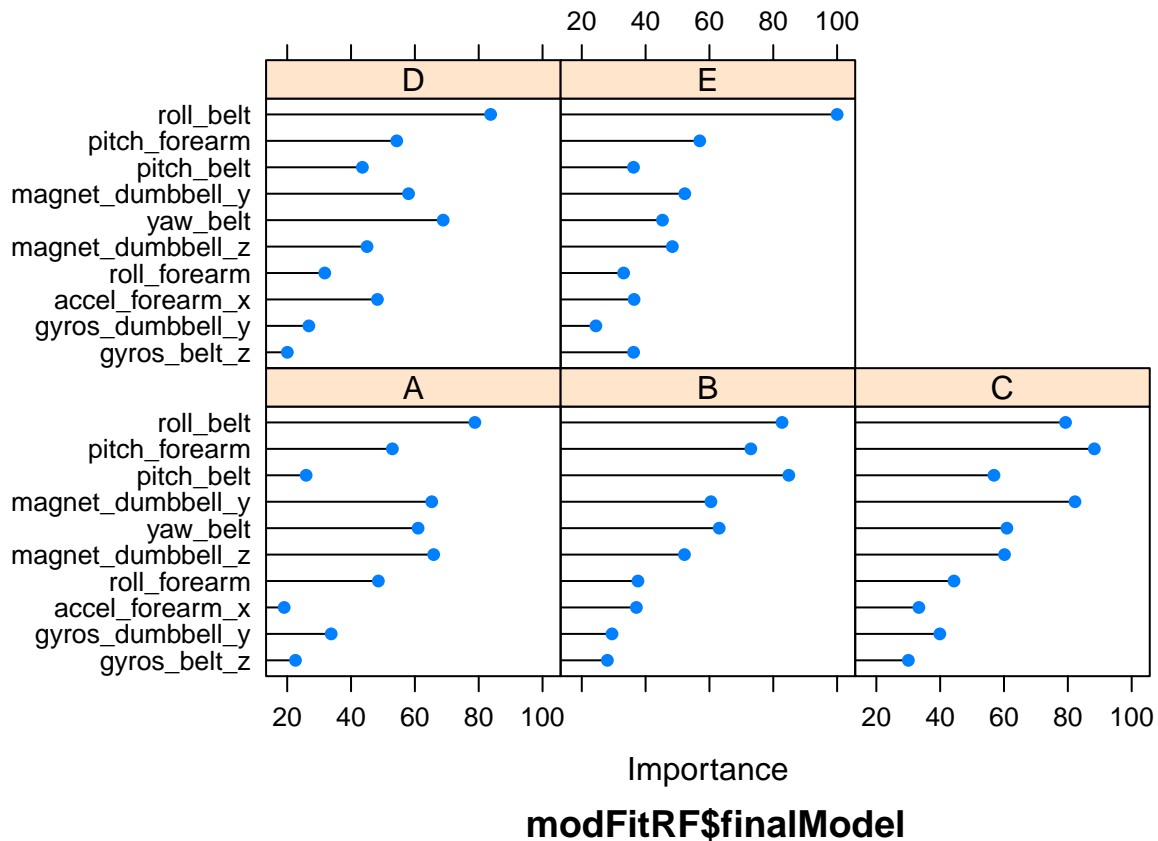
We create a random forest model using 5-fold cross validation, which will allow us to estimate the out-of-sample error. The training takes around 40 minutes on a MacBook Pro 2016 with CPU.

The random forest algorithm by default tries three different numbers of variables randomly sampled at each split, the `mtry`. We use the `CARET` `trainControl` function with option `savePredictions` to 'final'. As such, the predicted outcomes are saved for the cross-validation results of the model for the finally selected value for `mtry`, in this case 27. This allows us to create a **confusionMatrix** for the **cross-validated results**.

```
##           Reference
## Prediction    A    B    C    D    E
##           A 5574   23    0    0    0
##           B   3 3767   12    0    1
##           C   1   7 3400   34    3
##           D   0   0   10 3180   10
##           E   2   0   0   2 3593

##           Accuracy           Kappa  AccuracyLower  AccuracyUpper  AccuracyNull
##           0.9944960           0.9930376           0.9933586           0.9954828           0.2843747
## AccuracyPValue  McNemarPValue
##           0.0000000           NaN
```

We obtain an **accuracy within the 95% CI [99.3%,99.5%]**. We can **plot the most important predictors per class** and the **overall predictor importance**.



As we've obtained an **excellent cross-validated accuracy**, this model will be **used** for the **prediction of the test cases**.

It should be noted that strictly spoken the 5-fold cross-validation was not necessary to obtain an estimate for the out-of-sample error, since the default method of random forest uses bootstrapping. As such, only 63%

of the observations are effectively used for the model training. The remaining out-of-bag (oob) observation can be used as well to estimate the out-of-sample error. A random forest without cross-validation was built as well and the estimated oob-error was .42%, which is in line with the cross-validated out-of-sample error.

The **final predictions** using the **random forest built using 5-fold cross-validation** are:

```
test_predictions = predict(modFitRF, newdata=testing)
test_predictions
```

```
## [1] 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
```