Course Project - Practical Machine Learning

Bert Lijnen

7 September 2017

1. Introduction

In this project, we will use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants and we will attempt to predict the manner in which the participants did the exercise (the variable `Classe`). More specifically, we will report how we have built our model and motivate the different decisions involved in developping this model (cross validation, expected out-of-sample error, ...). The model will be built using the training set and we will use the prediction model to predict 20 different test cases in the test set. At the end of our analysis we will report our key findings.

2. Data import and data cleaning

We start by importing the datasets and the packages we will use. The data comes from: 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.

Participants were asked to perform barbell lifts correctly and incorrectly in 5 different ways (Class A-E). Class A corresponds to the specified execution of the exercise, while the other 4 classes correspond to common mistakes.

```
library(pander)
library(psych)
library(ggplot2)
library(caret)
library(kernlab)
library(rattle)
library(randomForest)
pml_training_url<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
pml_test_url<-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
pml_training<-read.csv(url(pml_training_url), na.strings=c("NA",""), header=TRUE)
pml_test<-read.csv(url(pml_test_url), na.strings=c("NA",""), header=TRUE)
dim(pml_training)
```

```
[1] 19622   160
```

```
dim(pml_test)
```

```
[1]  20 160
```

We can see that the number of variables is equal in both datasets. However, many variables are useless for predicting the outcome variable since they contain too many NAs (verified with the function `describe()` from the package `psych`). Other variables are useless because the information that they provide will not affect the outcome. Still other variables have a variance near zero (verified with the function `nearZeroVar()` from the package `caret`). We decide to remove these irrelevant features both for simplicity, but also to avoid overfitting and to increase the predictive performance of our model.

```
training_keep<-pml_training[,c(8:11, 37:49, 60:68, 84:86, 102, 113:124, 140, 151:160)]
dim(training_keep)
```

```
[1] 19622    53
```

We have suppressed 107 features that would unnecessarily complicate the model development. Let's now check the number of observations for each outcome category:

```
pander(table(pml_training$classe), caption="Observations in each outcome category")
```

Table 1: Observations in each outcome category

A	B	C	D	E
5580	3797	3422	3216	3607

Class A (correct lifting) contains more observations than the other classes (incorrect lifting). But the data is largely sufficient to fit a prediction model. It is crucial not to assess performance on the same observations that were used to develop the model. Therefore, we will split our training data for **cross validation**. We will use 60% of the observations to develop our model and 40% for assessing the model.

```
set.seed(1150)
inTrain<-createDataPartition(training_keep$classe, p=0.6, list=FALSE)
training<-training_keep[inTrain,]
testing<-training_keep[-inTrain,]
dim(training)
```

```
[1] 11776    53
```

```
dim(testing)
```

```
[1] 7846   53
```

3. Prediction Model: Random Forest

We choose to develop a random forest model to predict the class of the lifting exercises because a random classifier does not attempt to fit a single model to data but instead builds an ensemble of models that jointly classify the data. By fitting a large number of classification trees, the random forest model makes use of an assortment of models by optimizing each tree to fit only some of the observations using only some of the predictors. When we will predict a new case, it will be predicted by every tree and by means of a majority voting, the new case is assigned to the class that is most often selected. In this way, we avoid dependencies on precise model specifications. In addition, random forest models perform well accross a wide variety of datasets and problems (cf. CHAPMAN, C. and McDONNELL FEIT, E. 2015. **R for Marketing Research and Analytics**. Springer). We prefer to work with the `randomForest()` function from the package of the same name, because we used this function before and it is much faster than the `train()` function in the `caret` package.

First we fit the model and print the confusion matrix. Next, we run the model on the 40% of the data we left out to test the model and print the confusion matrix for the predictions.

```
set.seed(1150)
modFit<-randomForest(classe~., data=training, ntree=501)
pander(modFit)
```

Call: randomForest(formula = classe ~ ., data = training, ntree = 501) Type of random forest: classification Number of trees: 501 No. of variables tried at each split: 7

OOB estimate of error rate: 0.7133%

Table 2: Confusion Matrix

	A	B	C	D	E	class.error
A	3344	3	0	1	0	0.001195
B	15	2261	3	0	0	0.007898
C	0	17	2036	1	0	0.008763
D	0	0	29	1899	2	0.01606
E	0	0	3	10	2152	0.006005

The confusion matrix shows that the **classification error** is very low for all outcome categories.

```
modFit_pred<-predict(modFit, testing, type="class")
modFit_conf<-confusionMatrix(modFit_pred, testing$classe)
print(modFit_conf)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 2231    6    0    0    0
         B    0 1507   17    0    0
         C    0    5 1351   19    0
         D    0    0    0 1264    1
         E    1    0    0    3 1441

Overall Statistics

               Accuracy : 0.9934
                 95% CI : (0.9913, 0.995)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9916
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9996   0.9928   0.9876   0.9829   0.9993
Specificity            0.9989   0.9973   0.9963   0.9998   0.9994
Pos Pred Value         0.9973   0.9888   0.9825   0.9992   0.9972
Neg Pred Value         0.9998   0.9983   0.9974   0.9967   0.9998
Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
Detection Rate         0.2843   0.1921   0.1722   0.1611   0.1837
Detection Prevalence   0.2851   0.1942   0.1752   0.1612   0.1842
Balanced Accuracy      0.9992   0.9950   0.9919   0.9914   0.9993
```

The accuracy of the prediction is as high as 0.9934 with a 95% confidence interval of [0.9913, 0.995]. This means that the **out-of-sample error** is $1 - 0.9934 = 0.0066$. Of course we had many observations to develop our model which significantly increases the likelihood of finding a highly performant model. The different statistics by class corroborate the success of predicting the class membership of the test observations.

4. Using the random forest model on new cases

We now use the second dataset with 20 completely new cases to predict their class membership.

```
pred_new_cases<-predict(modFit, pml_test, type="class")
tab_pred<-data.frame(case=pml_test$X, predicted_class=pred_new_cases)
tab_pred_trans<-t(tab_pred)
pander(tab_pred_trans[-1,], split.table=Inf)
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
B	A	B	A	A	E	D	B	A	A	B	C	B	A	E	E	A	B	B	B

```
pander(table(pred_new_cases), caption="Predicted class of 20 new cases")
```

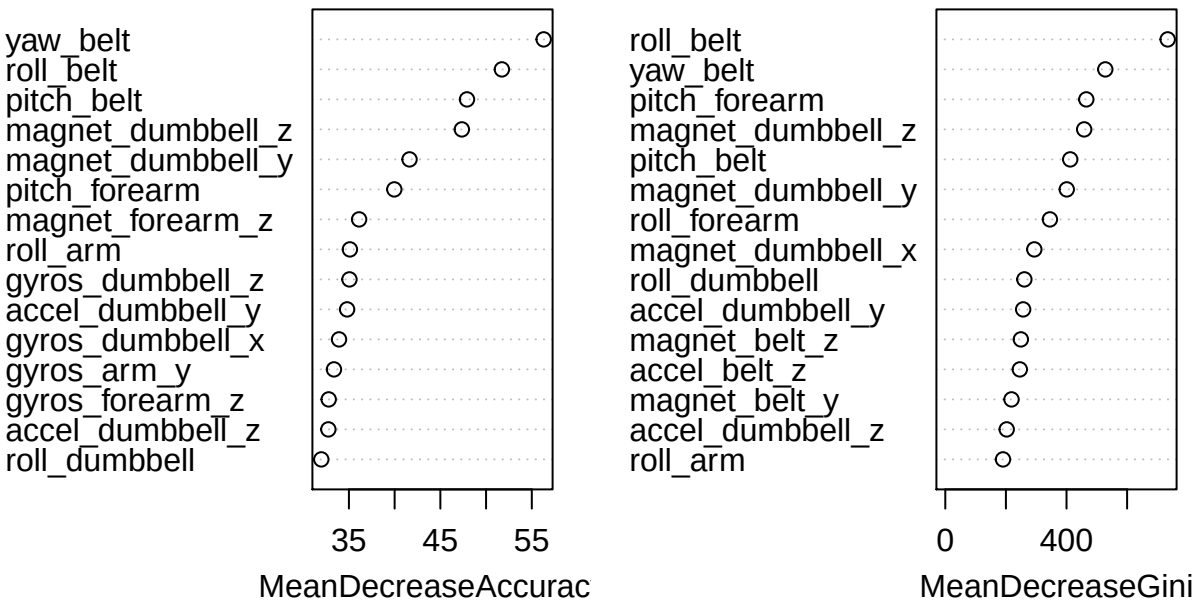
Table 4: Predicted class of 20 new cases

A	B	C	D	E
7	8	1	1	3

5. Random Forest Variable Importance

To conclude our analysis we will investigate the importance of a variable. To estimate the importance we run the `randomForest()` function but now include the argument `importance=TRUE`. We plot the 15 most important variables in order to keep the plot readable.

```
set.seed(1150)
var_imp<-randomForest(classe~., data=training, ntree=501, importance=TRUE)
varImpPlot(var_imp, main="Variable Importance", n.var=min(15, nrow(var_imp$importance)))
```



6. Conclusions

- In order to predict the class membership of the lifting exercises (class A = correct lifting and classes B-E represent incorrect liftings) we have developed a random forest model. The model predicts the class membership using 53 features.

- The prediction model has an out-of-sample error of only 0.0066 meaning that it can almost perfectly predict the class membership in the training dataset. To assess the model performance, we have used cross validation. Running the model on the test set corroborates the excellent predictive power (accuracy of 99%). We can therefore be confident to apply our model to predict new cases.

- The variables that contribute most to the prediction have been identified and plotted in section 5.