

# Prediction Assignment

Prashant Sharma

24/01/2021

Loading training and test csv data files.

```
training = read.csv("pml-training.csv",na.strings = c("", "NA"))
testing = read.csv("pml-testing.csv",na.strings = c("", "NA"))
dim(training)
```

```
## [1] 19622 160
```

```
dim(testing)
```

```
## [1] 20 160
```

lets clean our dataset for NA entries.

```
training <- training[ , colSums(is.na(training)) == 0]
training_set <- training[,-c(1:8)] #dropping irrelevant features
testing_set <- testing[,names(training_set[, -52])]
dim(training_set)
```

```
## [1] 19622 52
```

```
dim(testing_set)
```

```
## [1] 20 51
```

Our task here is to classify or predict whether the dumb-bell curl done by the user is correct or not and what type of error he is doing that is in which class his exercise form belongs.

lets import few packages for our classification task.

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##  
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':  
##  
## margin
```

```
library(parallel)  
library(doParallel)
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
cluster <- makeCluster(detectCores() - 4)  
registerDoParallel(cluster)
```

Before proceeding further lets create a validation set

```
set.seed(1234)  
intrain <- createDataPartition(training_set$classe , p = 0.75)[[1]]  
new_training_set <- training_set[intrain,]  
  
validation_set <- training_set[-intrain,]  
dim(new_training_set)
```

```
## [1] 14718    52
```

```
dim(validation_set)
```

```
## [1] 4904    52
```

lets try Random forest. we will be applying cross validation method at 5 folds.

```
fit_control<-trainControl(method = "cv" ,number = 5,allowParallel=TRUE)  
  
model1 <- train(classe ~ . , method = "rf" ,trControl = fit_control, data = new_training_set, ve  
rbose = F )
```

```
model1
```

```
## Random Forest
##
## 14718 samples
##    51 predictor
##    5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 11774, 11774, 11776, 11774, 11774
## Resampling results across tuning parameters:
##
##  mtry  Accuracy   Kappa
##    2    0.9902841 0.9877086
##   26    0.9917790 0.9895999
##   51    0.9856641 0.9818630
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 26.
```

Stopping the cluster.

```
stopCluster(cluster)
registerDoSEQ()
```

Confusion Matrix

```
prediction1 <- predict(model1 , newdata = validation_set)
confusionMatrix(prediction1 , as.factor(validation_set$classe))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A    B    C    D    E
##           A 1395   10     0     0     0
##           B     0  937     7     0     0
##           C     0    1  846     5     0
##           D     0    1    2  799     2
##           E     0    0     0     0  899
##
## Overall Statistics
##
##           Accuracy : 0.9943
##           95% CI : (0.9918, 0.9962)
##           No Information Rate : 0.2845
##           P-Value [Acc > NIR] : < 2.2e-16
##
##           Kappa : 0.9928
##
##           Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##           Class: A Class: B Class: C Class: D Class: E
## Sensitivity         1.0000   0.9874   0.9895   0.9938   0.9978
## Specificity         0.9972   0.9982   0.9985   0.9988   1.0000
## Pos Pred Value       0.9929   0.9926   0.9930   0.9938   1.0000
## Neg Pred Value       1.0000   0.9970   0.9978   0.9988   0.9995
## Prevalence          0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate       0.2845   0.1911   0.1725   0.1629   0.1833
## Detection Prevalence 0.2865   0.1925   0.1737   0.1639   0.1833
## Balanced Accuracy    0.9986   0.9928   0.9940   0.9963   0.9989
```

Overall accuracy

```
confusionMatrix(prediction1, as.factor(validation_set$classe))$overall[[1]]
```

```
## [1] 0.9942904
```

Lets predict on our test set:

```
prediction_test<- predict(model1 , newdata = testing)
prediction_test
```

```
## [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
```