Machine Learning Assigment

Endika

7 de julio de 2015

I will set the working directory and download the data.

```
setwd("/Users/endika/Downloads")
testing <- read.csv("~/Downloads/pml-testing.csv")
training <- read.csv("~/Downloads/pml-training.csv")
library(caret)
```

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

I will begin the PreProcesses: I will omit the variables with too many NA´s and Little variance.

```
#Remove near 0 variability

nsv <- nearZeroVar(training,saveMetrics=TRUE)
View(training)
training_no_na <- training[,!nsv$nzv]
testing_no_na <- testing[,!nsv$nzv]
View(training_no_na)
# Remove unnecessary columns
colRm_train <- c("X","user_name", "cvtd_timestamp","raw_timestamp_part_1","raw_timestamp
_part_2","cvtd_timesta
 mp","num_window")
colRm_test <- c("X","user_name","cvtd_timestamp","raw_timestamp_part_1","raw_timestamp_p
art_2","cvtd_timestam
 p","num_window","problem_id")
training_colRm <- training_no_na[,!(names(training_no_na) %in% colRm_train)]
testing_colRm <- testing_no_na[,!(names(testing_no_na) %in% colRm_test)]
View(testing_colRm)
#Remove Variables with too many NA´S

training_colRm<- training_colRm[ , colSums(is.na(training_colRm)) == 0]
testing_colRm <- testing_colRm[,(colSums(is.na(testing_colRm)) == 0)]
View(training_colRm)
View(testing_colRm)
```

Now its time to split the training set into training and validating set. Then we will test our predictor to the test set.

```
inTrain <- createDataPartition(y=training_colRm$classe, p=0.7, list=F)
training_clean <- training_colRm[inTrain,]
validation_clean <- training_colRm[-inTrain,];validation_clean  <- validation_clean
training_clean$row.names <- NULL
validation_clean$row.names <- NULL

View(validation_clean[,1])
```

In order to finf the best algorithm, I will test the correlation to see if there is a strongly correlated variable and use linear regresssion model. It´s not the case so i will use Random Forest

```
#lets chechk the corralation between the variables to see if "lm" is apropiate
cor <- abs(sapply(colnames(training_clean[, -ncol(training)]), function(x)
          cor(as.numeric(training_clean[, x]), as.numeric(training_clean$classe)
              , method = "spearman")))
head(cor,4)
```

```
##        roll_belt       pitch_belt         yaw_belt total_accel_belt
##       0.12844598       0.04485885       0.07794292       0.08816324
```

I will fit the model using the train function. I will do ***crossValidation to stimate the eroor***. I will do 4-folds cross validation as an argument in the train function.

```
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(1234)
#We will do cross validation in 4 times
rfFit <- train(classe ~  ., method = "rf", data = training_clean, importance = T, trCont
rol = trainControl(method = "cv", number = 4))
```

Its time to see the results obtained applying the predictor to the validationset to spect the ***out-sample error***

```
validation_pred <- predict(rfFit, newdata=validation_clean)
# Check model performance
confusionMatrix(validation_pred,validation_clean$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    8    0    0    0
##          B    0 1131    5    0    0
##          C    0    0 1019   23    0
##          D    0    0    2  940    2
##          E    0    0    0    1 1080
##
## Overall Statistics
##
##                Accuracy : 0.993
##                  95% CI : (0.9906, 0.995)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9912
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9930   0.9932   0.9751   0.9982
## Specificity            0.9981   0.9989   0.9953   0.9992   0.9998
## Pos Pred Value         0.9952   0.9956   0.9779   0.9958   0.9991
## Neg Pred Value         1.0000   0.9983   0.9986   0.9951   0.9996
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1922   0.1732   0.1597   0.1835
## Detection Prevalence   0.2858   0.1930   0.1771   0.1604   0.1837
## Balanced Accuracy      0.9991   0.9960   0.9942   0.9871   0.9990
```

Last I will test the model in the testing set, and create the tables

```
predict(rfFit,testing_colRm)
```

```
##  [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
```

```
pml_write_files = function(x){
          n = length(x)
          for(i in 1:n){
                    filename = paste0("problem_id_",i,".txt")
                    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names
=FALSE)
          }
}

#pml_write_files()
```