pml_YI.Rmd

YI HAN

Wednesday, April 22, 2015

Loading the data and basic setting

```
setwd("D:/YI_20150421/2_Coursera/Prac_ML/Projects/1")
TrainingData <- read.csv(file="pml-training.csv", head=TRUE, sep=",")
TestingData <- read.csv(file="pml-testing.csv", head=TRUE, sep=",")

library(lattice)
library(ggplot2)
library(caret)
library(randomForest)
```

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

```
set.seed(777)
```

Removing those cols which contain NAs and “”, and choosing some meaningful parameters as predictors

```
TrD <- TrainingData[,colSums(is.na(TrainingData))==0] #remove cols with NAs
TeD <- TestingData[,colSums(is.na(TrainingData))==0] #remove cols with NAs

TrainingData <- TrD[,colSums(TrD=="")==0] #remove cols with ""
TestingData <- TeD[,colSums(TrD=="")==0] #remove cols with ""

rm(TrD)
rm(TeD)

names(TrainingData)
```

```
##  [1] "X"                    "user_name"            "raw_timestamp_part_1"
##  [4] "raw_timestamp_part_2" "cvtd_timestamp"       "new_window"
##  [7] "num_window"           "roll_belt"            "pitch_belt"
## [10] "yaw_belt"             "total_accel_belt"     "gyros_belt_x"
## [13] "gyros_belt_y"         "gyros_belt_z"         "accel_belt_x"
## [16] "accel_belt_y"         "accel_belt_z"         "magnet_belt_x"
## [19] "magnet_belt_y"        "magnet_belt_z"        "roll_arm"
## [22] "pitch_arm"            "yaw_arm"              "total_accel_arm"
## [25] "gyros_arm_x"          "gyros_arm_y"          "gyros_arm_z"
## [28] "accel_arm_x"          "accel_arm_y"          "accel_arm_z"
## [31] "magnet_arm_x"         "magnet_arm_y"         "magnet_arm_z"
## [34] "roll_dumbbell"        "pitch_dumbbell"       "yaw_dumbbell"
## [37] "total_accel_dumbbell" "gyros_dumbbell_x"     "gyros_dumbbell_y"
## [40] "gyros_dumbbell_z"     "accel_dumbbell_x"     "accel_dumbbell_y"
## [43] "accel_dumbbell_z"     "magnet_dumbbell_x"    "magnet_dumbbell_y"
## [46] "magnet_dumbbell_z"    "roll_forearm"         "pitch_forearm"
## [49] "yaw_forearm"          "total_accel_forearm"  "gyros_forearm_x"
## [52] "gyros_forearm_y"      "gyros_forearm_z"      "accel_forearm_x"
## [55] "accel_forearm_y"      "accel_forearm_z"      "magnet_forearm_x"
## [58] "magnet_forearm_y"     "magnet_forearm_z"     "classe"
```

```
names(TestingData)
```

```
##  [1] "X"                    "user_name"            "raw_timestamp_part_1"
##  [4] "raw_timestamp_part_2" "cvtd_timestamp"       "new_window"
##  [7] "num_window"           "roll_belt"            "pitch_belt"
## [10] "yaw_belt"             "total_accel_belt"     "gyros_belt_x"
## [13] "gyros_belt_y"         "gyros_belt_z"         "accel_belt_x"
## [16] "accel_belt_y"         "accel_belt_z"         "magnet_belt_x"
## [19] "magnet_belt_y"        "magnet_belt_z"        "roll_arm"
## [22] "pitch_arm"            "yaw_arm"              "total_accel_arm"
## [25] "gyros_arm_x"          "gyros_arm_y"          "gyros_arm_z"
## [28] "accel_arm_x"          "accel_arm_y"          "accel_arm_z"
## [31] "magnet_arm_x"         "magnet_arm_y"         "magnet_arm_z"
## [34] "roll_dumbbell"        "pitch_dumbbell"       "yaw_dumbbell"
## [37] "total_accel_dumbbell" "gyros_dumbbell_x"     "gyros_dumbbell_y"
## [40] "gyros_dumbbell_z"     "accel_dumbbell_x"     "accel_dumbbell_y"
## [43] "accel_dumbbell_z"     "magnet_dumbbell_x"    "magnet_dumbbell_y"
## [46] "magnet_dumbbell_z"    "roll_forearm"         "pitch_forearm"
## [49] "yaw_forearm"          "total_accel_forearm"  "gyros_forearm_x"
## [52] "gyros_forearm_y"      "gyros_forearm_z"      "accel_forearm_x"
## [55] "accel_forearm_y"      "accel_forearm_z"      "magnet_forearm_x"
## [58] "magnet_forearm_y"     "magnet_forearm_z"     "problem_id"
```

```
TrainingData <- TrainingData[,7:60]
TestingData <- TestingData[,7:60]
# TrainingData$classe <- as.factor(TrainingData$classe) if they are not factor!!!
```

Applying 5-folds cross validation in 5 repetitions and training the data using random forest method, and checking how well this modelFit is.

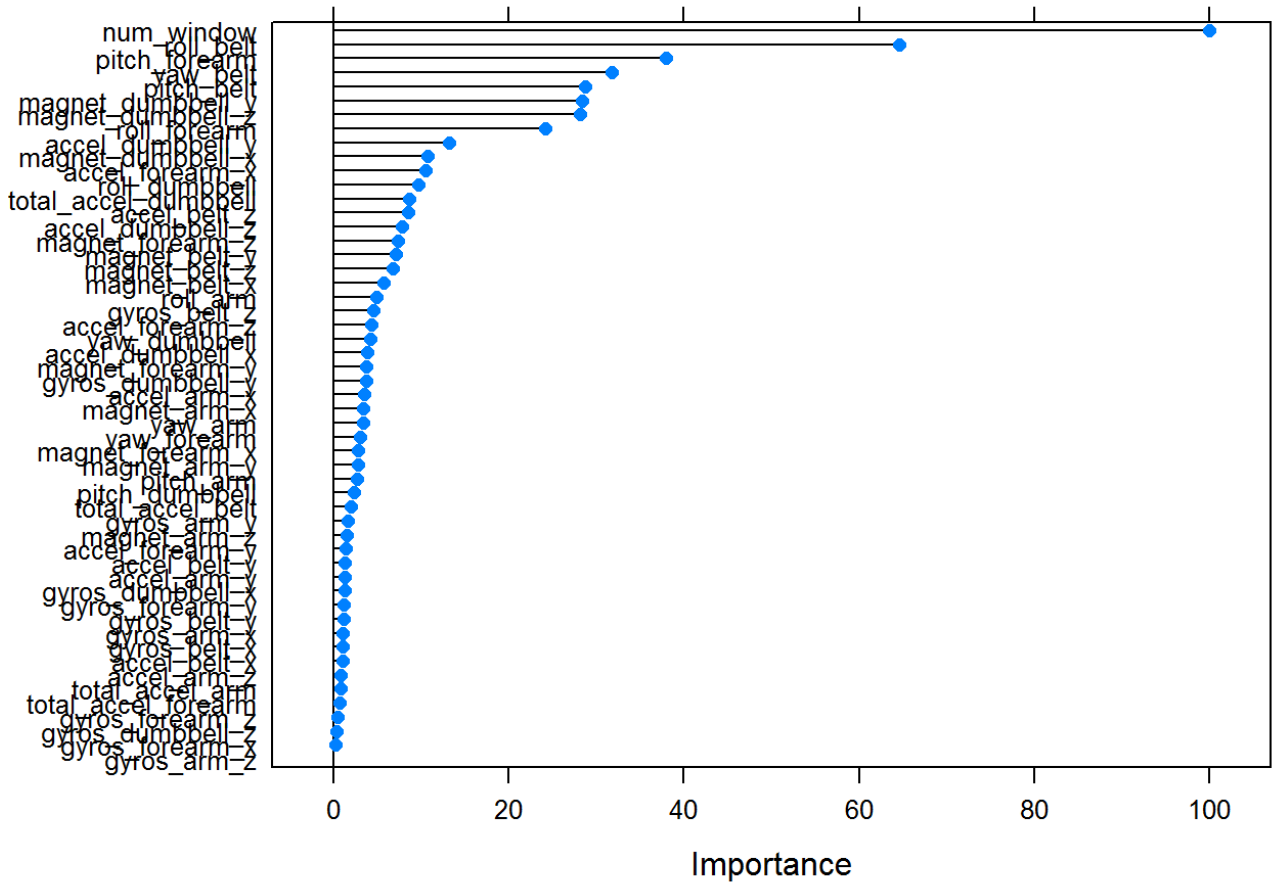
```
ctrl <- trainControl(method="repeatedcv", number=5, repeats=5)
# number: Either the number of folds or number of resampling iterations

modelFit <- train(classe ~ ., data=TrainingData, method="rf", trControl=ctrl)

modelFit
```

```
## Random Forest
##
## 19622 samples
##    53 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 5 times)
##
## Summary of sample sizes: 15699, 15698, 15697, 15696, 15698, 15697, ...
##
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa      Accuracy SD   Kappa SD
##    2    0.9961575  0.9951394  0.0007777574  0.0009838516
##   27    0.9980940  0.9975891  0.0007285632  0.0009215901
##   53    0.9962288  0.9952297  0.0013639941  0.0017254416
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 27.
```

```
summary(modelFit)
```

```
##                 Length Class      Mode
## call               4  -none-     call
## type               1  -none-     character
## predicted      19622  factor     numeric
## err.rate        3000  -none-     numeric
## confusion         30  -none-     numeric
## votes          98110  matrix     numeric
## oob.times      19622  -none-     numeric
## classes            5  -none-     character
## importance        53  -none-     numeric
## importanceSD       0  -none-     NULL
## localImportance    0  -none-     NULL
## proximity          0  -none-     NULL
## ntree              1  -none-     numeric
## mtry               1  -none-     numeric
## forest            14  -none-     list
## y              19622  factor     numeric
## test               0  -none-     NULL
## inbag              0  -none-     NULL
## xNames            53  -none-     character
## problemType        1  -none-     character
## tuneValue          1  data.frame list
## obsLevels          5  -none-     character
```

```
plot(varImp(modelFit))
```



Here shows hwo this model works for the training data (train set and validation set). As you can see, the accuracy is very high as around 99.6%. The reason I choose 5-folds cross validation is because it is 80% as train set and 20% as validation set which is very general or quite similar to default setting (70% and 30%). I use 5 repetitions trying to make the resuls more believable.

Calculating the errors using the TrainingData Set, generating data for the prediction vector for the Assigment Submission

```
predictionsTr <- predict(modelFit, newdata=TrainingData)
confusionMatrix(predictionsTr,TrainingData$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 5580    0    0    0    0
##          B    0 3797    0    0    0
##          C    0    0 3422    0    0
##          D    0    0    0 3216    0
##          E    0    0    0    0 3607
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9998, 1)
##     No Information Rate : 0.2844
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2844   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2844   0.1935   0.1744   0.1639   0.1838
## Detection Prevalence   0.2844   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

It shows that sensitiity and specificity for 5 classes are all 1. The kappa statistic of 1 reflects the out-of-sample error.

Predicting the rsults for the testing data

```
predict(modelFit, newdata=TestingData)
```

```
##  [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
```